Tracking Ballistic Vehicles during Boost

Development and Performance Analysis of Tracking Filter Algorithms

by Vicente Aycart Avila

Tracking Ballistic Vehicles during Boost

Development and Performance Analysis of Tracking Filter Algorithms

By

Vicente Aycart Avila

Master Thesis Report

to obtain the degree of Master of Science at the Delft University of Technology, to be defended in Delft in January 17, 2020

Non-Disclosure Agreement

This academic thesis contains sensitive information belonging to Airbus Defense and Space GmbH. In accordance with the legal obligations of the university, the work will be placed under embargo in the library repository of the university for five years following its publication. During the embargo period, this work will only be viewed by experts designated by Airbus as well as authorized members of the examination board designated by TU Delft. Some data of this work is not available for public research.

Submission Date:	December 8, 2019
Student Name:	Vicente Aycart Avila
Student Number:	4743512
Supervisor:	Dr. Ir. E.J.O. Schrama
Defense Committee:	Pieter Visser EJO Schrama Erik-Jan van Kampen

Cover image retrieved from *www.esa.int/ES*

TU Delft Delft University of Technology

This page is intentionally left blank

Abstract

Accurately tracking the trajectory of launching vehicles during the boost phase is crucial for a variety of reasons. First, an accurate burnout point location and velocity of the launch vehicle will facilitate the estimation of its trajectory during the following phases of ballistic flight and the impact location. This has important implications for safety purposes. The first stage of many launcher systems such as the Russian Soyuz or the Chinese Long March rockets commonly impact on land, sometimes on populated areas. Furthermore, the proliferation of suborbital rocket launches presents a hazard if regulations are not properly followed. Therefore, it becomes critical to estimate the impact point of any vehicle that reenters the lower layers of the atmosphere beforehand, during the boost phase, in order to prevent personal and material damage. Second, a critical aspect in any space mission is to insert the payload into the required orbit. An accurate tracking of the launcher system during the boost phase allows carrying out early orbit determination in order to successfully complete the payload insertion.

In this Master Thesis, an analysis and performance comparison of different tracking algorithms for launching vehicles during boost (e.g. launcher systems or suborbital rockets) has been carried out. This type of tracking problem has several difficulties that must be overcome. First, the unknown thrust profile – thrust magnitude and orientation – of a launching vehicle makes difficult to develop motion models that can accurately describe its behavior. Second, the observations of the plume of the launching vehicle obtained from two line-of-sight passive sensors located in geostationary satellites are used. The way these observations are obtained does not allow to accurately measuring the initial state and trajectory of the vehicle, complicating the initialization and the trajectory estimation processes of the tracking filters. Finally, the nonlinearities present in the system and measurements models of the tracking filters compel us to devise linearization schemes or the implementation of alternative filtering methods.

The work presented in this document has been developed at the Military and Security department of Airbus Defense and Space, as a continuation of an ongoing project. To solve the aforementioned difficulties of the tracking problem, a tracking system – EKF-Tool – was developed at ADS previous to the start of this Thesis. Nevertheless, this tool shows several limitations that hinder its performance, even preventing from successful tracking. In this Thesis, the limitations and potential areas of improvement of the EKF-Tool have been analyzed. Based on the results of this analysis, an alternative tracking algorithm – PKF-Tool – is introduced with the aim of overcoming the limitations of the EKF-Tool. The tracking performance of both tools has been tested using several indicators: stability, consistency, credibility, accuracy, precision, and versatility.

This page is intentionally left blank

Preface

I would like to thank every classmate, professor, and member of the academic community that has helped me during my undergraduate and graduate studies in Spain, the United States, and in The Netherlands. From them I have learnt that hard-work, honesty, and commitment; rather than talent are the actual keys to a successful professional career. I would also like to express my gratitude to the staff at Spacebel in Brussels, who introduced me to the space sector at a professional level; and the staff at Airbus in Friedrichshafen, especially to Garima Pandey and Tilman Binder: without their help this Thesis would have not existed. Thanks to my Thesis supervisor, Ernst Schrama, for his feedback during this project. Finally, I would like to thank my family, girlfriend, and friends for their unconditional support and understanding, not only for these last few months but also since the start of this journey.

Table of Contents

List of Figures

List of Tables

Symbols and Acronyms

Symbol	Description [units]
θ	azimuth
ε	elevation
α	true/mean anomaly (circular orbit)
ϕ	latitude
λ	longitude
σ_{pos}	standard deviation in position
σ_{vel}	standard deviation in velocity
σ_n	standard deviation of measurement noise for azimuth and elevation coordinates

Acronym	Description
AA	Access Area
Airbus / ADS	Airbus Defense and Space GmbH
CB	Cloud Break Altitude
CDF	Cumulative Distribution Function
DOF	Degrees of Freedom
ECEF	Earth Centered Earth Fixed
EKF	Extended Kalman Filter
GA	Genetic Algorithm
GEO	Geostationary
GT	Gravity Turn
IR	Infrared
LP	Launch Point
LV	Launch Vehicle
LOS	Line of Sight
LHZ	Local Horizontal Frame
MAE	Mean Absolute Error
MSE	Mean Square Error
MC	Monte Carlo
NCI	Non-Credibility Index
NEES	Normalized Estimation Error Squared

NIS	Normalized Innovation Squared
PKF	Polynomial Kalman Filter
PKFCC	Polynomial Kalman Filter Coupled Cartesian
PKFCP	Polynomial Kalman Filter Coupled Polar
PKFD	Polynomial Kalman Filter Decoupled
PKFM	Polynomial Kalman Filter with Markov process model
PSD	Power Spectral Density
RAAN	Right Ascension of the Ascending Node
RMSE	Root Mean Square Error
SE	Simulation Environment
SNC	State Noise Compensation

Chapter 1

Introduction

In this chapter, an introduction to the Thesis project is provided. The context in which this Thesis takes place is analyzed, together with the scope and purposes of the project. An overview of the different topics to be analyzed throughout the entire document is also provided.

1.1. Space-Based Object Tracking

There is a wide variety of applications for which tracking filters are useful. Within the aerospace sector, filters are used for defense, space, and civil aviation applications. For example, the orbit determination of artificial satellites for precise altimetry applications [1] and the estimation of the flight path of aircrafts using radars [2] are performed using tracking systems. In this Thesis, an analysis and comparison of tracking filters will be carried out. These filters will be applied to estimate the trajectory of flying objects, or vehicles, such as ballistic rockets or space launcher systems that are exiting the Earth's atmosphere.

A ballistic vehicle is a rocket launched from Earth that usually has three different flight phases. The first one corresponds to the phase in which the vehicle is under the influence of thrust. This phase is commonly referred in the literature as the boost phase. Once the launcher has reached a certain altitude and velocity, usually after the thicker, lower part of the atmosphere has been overcome, the launcher's engines stop, and the vehicle enters in a coasting phase in which the main acting force is the gravity. The last phase corresponds to the re-entry phase, in which the vehicle returns to the thicker part of the atmosphere and the aerodynamic forces are predominant (besides gravity) if no decelerating thrust is present [3] [4].

Tracking such objects is useful for a wide variety of reasons. Among many possible applications in the space domain, tracking powered launching vehicles (LV) is used for early orbit determination and prediction [5] and for defense applications such as identifying whether the launcher system corresponds to a civil or military organization [6]. More specifically, obtaining an accurate trajectory of launch vehicles is fundamental for safety reasons. Commonly, launchers' first stages re-enter the lower atmosphere and impact on the ground. Therefore, it is essential to determine precise impact points in order to specify the safety of a given mission [7]. In the given tracking problem, this is done by estimating accurately the trajectory of the vehicle during the boost phase and more specifically the vehicle's burnout state, as measurements are only available during this phase due to the specific characteristics of the observing sensors. If the vehicle's burnout state is accurate, the propagation of the state for the subsequent flight phases without available observations can be done more precisely, finally obtaining a better estimation of the impact point [3].

1.2. Research Scope

In this section, the specific areas that this Thesis study will explore will be provided. To do this, the project context introduced in the previous section will be broadened. Then, a series of constraints and assumptions needed to define the boundaries of the study will be established. Finally, the thesis objectives will be set.

1.2.1. Project Context

This thesis is developed using the conclusions of the Literature Review [3] as a starting point. A tracking tool named EKF-Tool – in short, EKF – was previously developed at Airbus with the goal of estimating the trajectory of launching vehicles from Earth using a pair of infrared line-of-sight sensors located in geostationary satellites. This Tool is based on an extended Kalman filter and uses a gravity-turn dynamic model for the propagation of the state.

The EKF-Tool is used within a Simulation Environment (SE) in which in-house (true) trajectories are generated in order to simulate the observations needed by the filter. After the study carried out for the Literature Review, it was seen that the performance – accuracy – of the EKF-Tool was relatively high in comparison with other trackers found in the literature [8] for the levels of measurement noise contained in the observations. That is, the estimated results produced by the EKF-Tool were relatively closer to the truth than the results of other filters.

Nevertheless, these results were obtained with observations simulated just from the in-house trajectories. It has been found that the EKF-Tool presents several issues when realistic trajectories such as the ones from a civil launcher vehicle are used. First, the EKF-Tool is not capable of filtering at all when external trajectories are tracked and the filter's input parameters – process noise – are the same ones used for in-house trajectories. Namely, filter divergence is obtained because the level of process noise used to compensate the filter's dynamic model mismatch with the reality is too low. If process noise is increased, the performance – accuracy – of the filter is reduced, although divergence is avoided. Second, the amount of process noise necessary to avoid filter divergence cannot be known beforehand. There are as many different values of process noise that allow filter convergence as rockets launching from Earth whose trajectory is to be tracked. Therefore, large amounts of process noise must be input in order to avoid filter divergence when tracking any possible trajectory. However, tool's performance indicators such as accuracy and credibility (see Chapter 5) will show poor results. Thus, a tracking tool that does not rely so heavily on its input parameters should be sought in order to achieve trajectory-tracking of any rocket launching from any location on Earth and still obtain good performance results.

In order to find an alternative filtering approach to the EKF-Tool that could overcome its limitations, several filtering methods were studied during the Literature Review and the analysis carried out at the beginning of the thesis work. Two different aspects were analyzed.

On one hand, the utilization of filtering methods different from the extended Kalman filter was studied. In this manner, nonlinear approaches that do not rely on the linearization methods of the extended Kalman filter such as the unscented Kalman filter (UKF) or the particle filter (PF) were reviewed. Furthermore, simpler, linear filtering methods were also considered such as the polynomial Kalman filter. On the other hand, the implementation of different types of dynamic models was analyzed. In this way, more general models than the gravity turn of the EKF-Tool that considered more complex trajectories were studied. Some of these dynamic models take into account possible lateral motion of the vehicle, for example. Some other dynamic models were based on stochastic processes. After carrying out the review of alternative filtering methods and dynamic models, the polynomial Kalman filter seemed the most promising one due to its simplicity, which made possible to use relatively complex dynamic models in a simpler manner. Thus, stochastic processes could be included in order to model the dynamics of the tracked vehicle. Moreover, such type of filters has already been used for tracking ballistic vehicles in the literature [8] [9] so good filtering performance is expected to be ensured, at least to a certain extent.

Therefore, a polynomial Kalman filter using stochastic models is the preferred solution to overcome the deficiencies of the EKF-Tool. Such a filter is named PKF-Tool – in short PKF – in analogy to the EKF filtering tool developed at Airbus[*]. Thus, the whole thesis work contained in this document explains the development and implementation of the PKF-Tool within the same SE in which the EKF-Tool is built in and displays the performance comparison results for both of the filtering tools. The following parts of this section will explain the constraints that exist and assumptions that are made within this thesis. Moreover, the main hypothesis posed regarding the tracking problem at hand will be provided. Finally, the objectives of this thesis will be stated.

1.2.2. Constraints and Assumptions

A series of constraints and assumptions are made in order to limit the scope of the problem and define the boundaries of the launch scenario in which tracking is allowed. A launch scenario is defined as all the different aspects that can be simulated within the SE in order to determine a specific tracking problem. These aspects range from the trajectory direction and the launch point location of the actual vehicle to the observing satellites' setup. In any case, the assumptions and constraints defined for this tracking problem are itemized as follows.

- Any true trajectory profile of a tracked vehicle consists on a two-stage rocket. An inter-stage coasting phase might exist or not.
- Observations are only available after the vehicle has surpassed the cloud tops. These clouds are assumed to be at a height of 10 km, unless otherwise indicated.

[*] Within the context of this Thesis, note the distinction between "extended Kalman filter" and "EKF": The former is used for referring to any general extended Kalman filter whereas the latter EKF nomenclature is used for referring to the EKF-Tool that is based on the specific extended Kalman filter described in the following chapters. The same applies for polynomial Kalman filter and PKF.

- The bias of the sensors that obtain the measurements is assumed to be zero.
- Sensors' noise is white random noise (the noise is Gaussian, so it is assumed that other types of noise such as glint noise is not present). Its value is defined by the stakeholder (Airbus).
- The tracking tool shall be capable of estimating the trajectory of the vehicle in real-time.
- Computational performance shall not be considered in the design of a solution.

1.2.3. Main Hypotheses

Once the limitations and issues of the EKF-Tool have been identified, it is proposed for this thesis the development of an alternative filtering tool capable of performing tracking without the limitations of the EKF-Tool. It is thus, hypothesized, that such a tracking tool should be capable of the following:

- Estimating the vehicle's state (at least at burnout) for any possible launch scenario with an accuracy and precision at least of the same order of magnitude as the EKF-Tool.
- Being able to perform tracking of different trajectories without the need of modifying its input parameters – i.e. process noise – and initialization method.
- Perform equally for any possible launch scenario in terms of consistency and credibility.

After the study carried out in Chapter 2 regarding the alternative filtering method and the problem background, a series of more specific hypotheses to the new filter are proposed in addition to the ones described above.

1.2.4. Thesis Objectives

In summary, the EKF-Tool is only capable of tracking one specific trajectory. For tracking other trajectories, its inputs have to be modified beforehand. However, since any tracked vehicle's trajectory is unknown, it is impossible to find an input value that ensures the EKF-Tool will perform well (i.e. avoid filter divergence). Thus, an alternative filtering method called PKF-Tool is envisaged so these limitations can be overcome. In order to test that this new tool behaves in a better way than the EKF-Tool, a performance comparison of their outputs is performed for a variety of launching scenarios.

Therefore, based on this problem definition, the objective of this Master Thesis Project is *to develop a filtering tool capable of solving the defined trajectory tracking problem by carrying out a performance analysis and comparison of its output with a filtering tool already developed at ADS.*

This main objective will be achieved by completing the following sub-goals:

1) Carrying out an analysis of suitable tracking-filter algorithms, using the Literature Review as starting point.
2) Developing a filtering tool based on the analysis from sub-goal 1
3) Perform any necessary modification to the Simulation Environment so a larger variety of trajectories can be tracked by

   a. allowing multi-stage rockets
   b. allowing coasting
   c. allowing the use of externally developed trajectories (i.e. from real missions)
4) Finding suitable performance comparison methods
5) Carrying out a performance comparison analysis using the methods from sub-goal 4 and the SE from sub-goal 3

1.3. Methodology

The methodology used for this thesis must be understood within a larger context in which the Literature Review is included. The analysis carried out in the Literature Review [3] led to the development of this thesis. Both the Literature Review and the Thesis Project comprise the necessary steps to understand and tackle the problem of tracking ballistic vehicles during boost. As this whole project aims to compare the outcome of several tracking implementations in a shared Simulation Environment that includes a Kalman filtering phase, it was decided to undertake a mixed methods approach. This involved analyzing the problems of the EKF-Tool and potential solutions from a qualitative point of view. These tasks are part of the work explained in the Literature Review. Then, the development of an additional tracker, the PKF-Tool, and a study of the performance of both filtering tools are assessed using a quantitative approach. The development of the PKF-Tool and the filters' performance comparison are carried out throughout this thesis.

The methodology followed in the Literature Review and the Thesis Project is summarized in Figure 1. In the Literature Study, an analysis of the EKF-Tool was performed, together with a preliminary study of the tracking problem. Then, during the thesis period, the conclusions obtained in the Literature Review were used as a stepping-stone for a more thorough analysis for potential solutions of the tracking problem. This analysis led to the development of the PKF-Tool, in analogy to the EKF-Tool previously developed at Airbus. Furthermore, from these analyses it was concluded that in order to perform a fair performance comparison, different launch scenarios were necessary to be tested. That is, external and more complex true trajectories should be used for assessing filter performance. Due to the increase in complexity of these trajectories, the Simulation Environment (SE) had to be modified accordingly, allowing the reading of external trajectories and the possibility of tracking trajectories with staging events and inter-stage coasting phases. Finally, several performance indicators were developed within the SE in order to test both tools' outputs for the different scenarios.

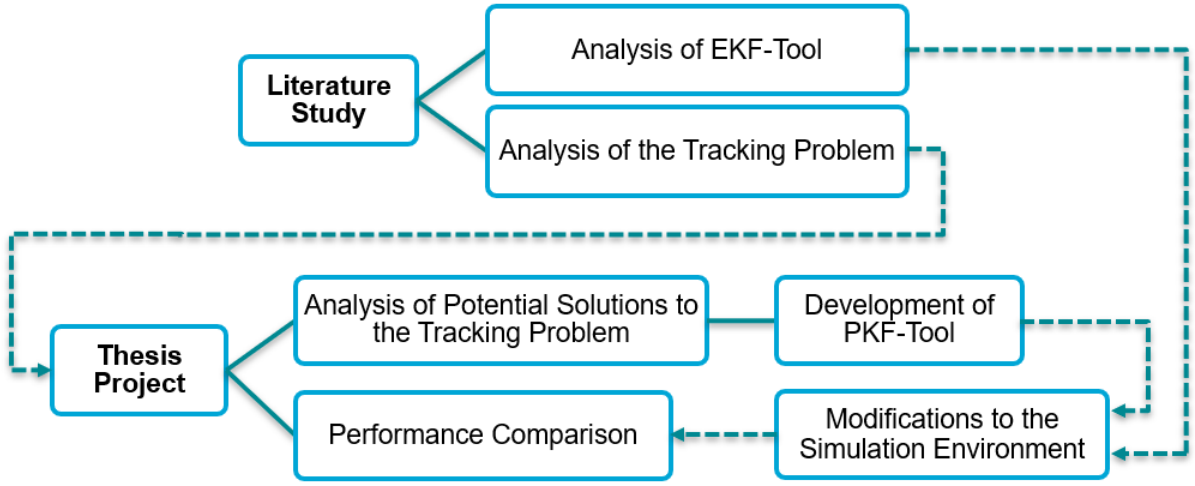

Figure 1: Thesis methodology. The work presented here is the outcome to the Literature Study and the Thesis project carried out at ADS.

The code implemented within the SE is written in Fortran 77/95. The IDE used for developing code was MS Visual Studio. As a final note, parallel verification of the FORTRAN code developed throughout this thesis was performed using MATLAB.

1.4. Organization of the Thesis

The organization of the thesis follows a process that answers to each of the sub-goals from Section 1.2.4 so the thesis objective can be achieved. More specifically, the problem background is further explained in Chapter 2. The problem setup is exhaustively explained there by analyzing the way observations are obtained and by providing a general overview of the characteristics of the motion of launching vehicles. Then, a mathematical description of the estimation problem and the filtering tools analyzed in this thesis are provided. In Chapter 3, the Simulation Environment in which this thesis is carried out is described. This includes an explanation of the way trajectories are generated or read, as well as how they are used to simulate observations, since real observations are not currently available. The chapter is concluded by providing the modifications made to the SE in order to carry out a performance comparison. Then, in Chapter 4, the specific characteristics of the dynamic models and measurement models of the tracking filters are described. This is followed by Chapter 5 and Chapter 6 in which the results of this thesis are presented. In Chapter 5 several performance indicators are introduced together with the results obtained for each of the filtering tools for external and internal trajectories. Chapter 6 carries out a parametric performance analysis by analyzing different launch scenarios using the performance indicators introduced in Chapter 5. Finally, conclusions and further lines of work regarding this thesis work are provided in Chapter 7.

Chapter 2

Problem Background

Tracking the trajectory of powered launch vehicles is a highly nonlinear problem. As it is pointed out in [10], in an actual flight the initial state of the vehicle is never known perfectly and there are disturbances in the motion derived from environmental aspects of the vehicle that make the differential equations to be defined only approximately. In this way, the differential equations that describe the motion of launch vehicles exiting the atmosphere are usually unknown, since their flight profile can greatly vary from one vehicle to another. Even though the flight path that follow most of these launch vehicles can be understood in general terms, there are uncertainties derived from trajectory loft or depression, thrust profile management, etc. that increase the complexity of the vehicle's trajectory estimation [11]. Furthermore, nonlinear disturbances due to aerodynamic or gravitational forces are also present during the motion of a ballistic vehicle. Most of the techniques used for tracking ballistic vehicles are model based: they depend on the description of the behavior of the vehicle through a motion model and they rely on the observations made to track the vehicle through an observation model [12].

The system in which this thesis is based on consists of two parts: launcher vehicle (LV) and mission side, which in turn is divided in ground segment and space segment. The LV side is the part of the system that cannot be controlled since it is the rocket in a boost phase trajectory to be tracked. The mission side is the part of the system that can be controlled. On one hand, the space segment includes two LOS passive optical (IR) sensors located in a GEO orbit that can measure the location of the LV. The design and analysis of this part of the mission side is out of the scope of this thesis and the necessary inputs are provided by the stakeholder. On the other hand, the ground segment includes all the tools used for the post-processing of the raw IR observations obtained by the sensors such as the computing unit that carries out the filtering. The focus of this thesis, thus, is put on this part of the mission side of the system.

The purpose of this chapter is to provide the necessary background to understand the trajectory-tracking problem and to provide an introductory explanation on the filtering part of the system as well as initial solutions to the tracking problem. To do this, three main aspects that have an influence on the results produced by any tracking tool will be analyzed: observations, motion of boosting vehicles, and estimators. First, the problem setup and the way observations are produced will be provided. Second, the main characteristics of the types of vehicles to which the tracking tools studied in this thesis are applicable will be briefly introduced. Then, a general overview of Kalman filtering will be given. Finally, the two tracking tools analyzed and developed for this thesis, the PKF-Tool and EKF-Tool, will be presented.

2.1. Obtaining Observations

Without accounting for the nonlinearity and uncertainty derived from the vehicle's motion, the tracking problem of ballistic vehicles during boost face the challenge of obtaining accurate observations of the flying vehicle. Several different types of approaches for obtaining measurements are currently used to solve the tracking problem analyzed in this thesis. On one hand, radar-based observations allow accurate tracking from ground of the whole trajectory of ballistic vehicles, even during non-powered phases of flight [13]. On the other hand, among other infrared detection methods, space-based sensors are gaining more attention in the last years due to their capability of analyzing larger areas on Earth's surface from where a vehicle could be launched [14].

Sensor Characteristics

The case studied in this thesis corresponds to observations obtained from a pair of geostationary satellites that are separated by a certain true anomaly $\Delta\alpha$, each equipped with an infrared (IR) Line-Of-Sight (LOS) sensor. These sensors measure the azimuth θ and elevation ε of the tracked vehicle with respect to their local horizon (LHZ) reference frame, as shown in Figure 2.

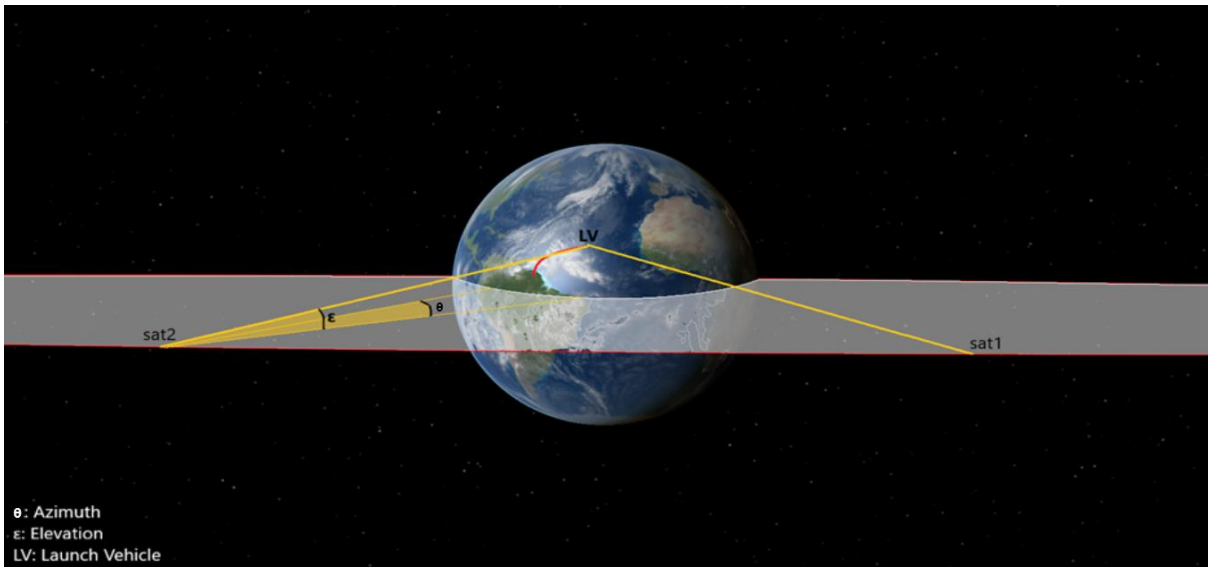

Figure 2: Observation satellites setup. These measurements are carried out by detecting the specific thermal signature of the plume produced by the boosting vehicle, in other words, the sensors detect the vehicle only when the engine is running, and exhaust gases are being produced.

Furthermore, due to cloud cover, the sensors are only able to detect the thermal signature when the vehicle has already been flying for a certain amount of time, usually when the vehicle has reached an altitude enough to surpass the cloud tops [15], as shown in Figure 3. In this way, the kick angle that initiates the gravity turn maneuver of the vehicle (see Section 2.2) is assumed to take place before the first thermal signature of the vehicle has been obtained, so the vertical ascent phase of flight is not detected by the sensors.

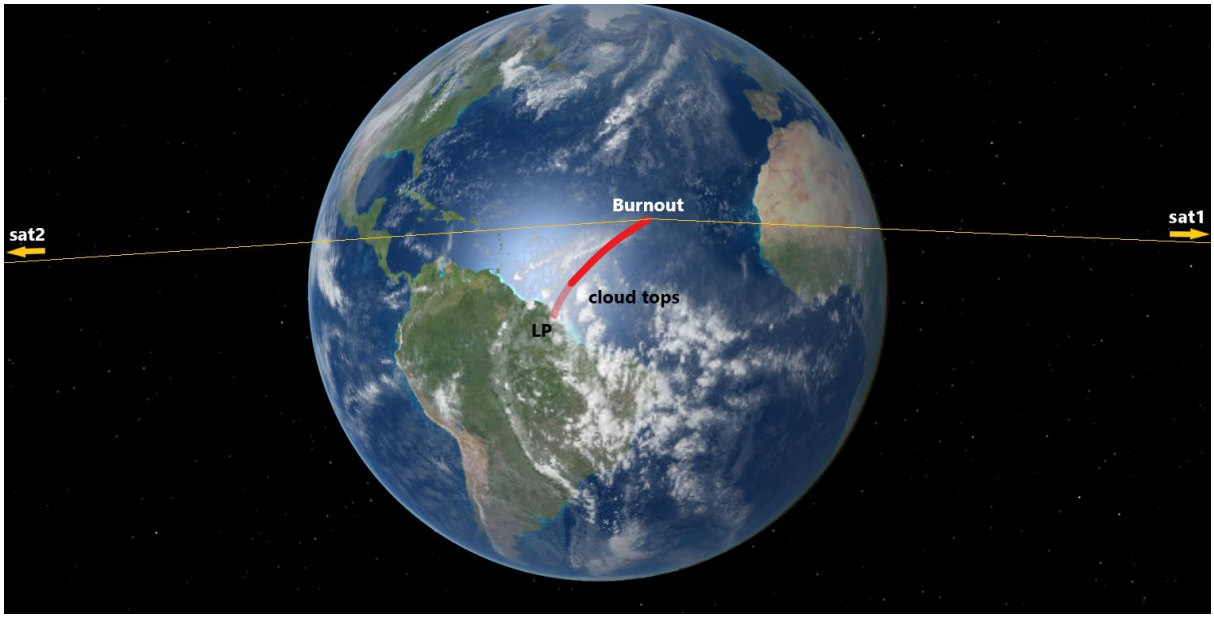

Figure 3: Observability region. The two observing satellites are only capable of tracking the launching vehicle once it has surpassed the cloud tops and until engine burnout since only the thermal signature of the exhaust gases can be seen

The sensors' noise and bias are reflected in the obtained observations. These observations made by the sensors are sent to ground for processing the trajectory of the vehicle. Furthermore, it is assumed for this thesis work that the measurements obtained by the sensors are synchronized, although some of the tracking filters developed and analyzed throughout this document can also handle asynchronous measurements (see Section 4.4).

Measurement Noise

One of the greatest challenges of the tracking problem resides in the geometric setup of the sensors with respect to the vehicle, or target. The observing satellites are located in a GEO orbit, far from the tracked vehicle. Therefore, a small amount of sensor noise traduces into a large error of the measurements after processing the IR signal. Instead of obtaining the precise location of the target, an ellipsoidal region where it is possible to find the vehicle is observed to a certain confidence level. The purpose of using a filter is to reduce the volume of this ellipsoid, or in other words, to filter the noise produced by the sensors. In any case, the sensor noise is a parameter intrinsic to the way the sensors are produced and can only be calibrated in certain cases. For the work of this thesis, measurement, or sensor, noise is considered to be a fixed value that is provided by Airbus. Moreover, it is assumed that the performance of the sensors is the same for tracking in azimuthal or elevation directions. Sensor bias is assumed to be zero.

2.2. Motion of Boosting Vehicles

To carry out tracking of boosting vehicles, it is necessary to know, at least to a certain extent, the main characteristics of its motion in order to develop accurate dynamic models that help in this process. Usually, rockets launching from Earth's surface follow a determined type of maneuver called Gravity Turn that allows optimizing the velocity gain due to the thrust force

[16]. In a GT maneuver, the launch vehicle flies with almost zero angle of attack. That is, the direction of the velocity vector is parallel to the direction of the thrust force. Thus, the net acceleration (the acceleration due to thrust force minus drag) is determined by the following vector equation

$$\boldsymbol{a_T} = a_T \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \qquad (1)$$

where a_T is the magnitude of the net acceleration and $\boldsymbol{v}$ is the velocity vector of the flying vehicle.

The net force $\boldsymbol{F_T} = m\boldsymbol{a_T}$ is the main source of acceleration of the ballistic vehicle. Secondary forces that act on the flying vehicle are the gravitational force and the Coriolis and centrifugal forces caused due to the utilization of a non-inertial reference frame such as the ECEF for the case studied in this thesis. Nevertheless, the effect of non-inertial forces is negligible in comparison with thrust and drag and it is not included in the representation of the vehicle's motion for the sake of simplicity. Figure 4 shows a simplified representation of the forces acting on the vehicle.

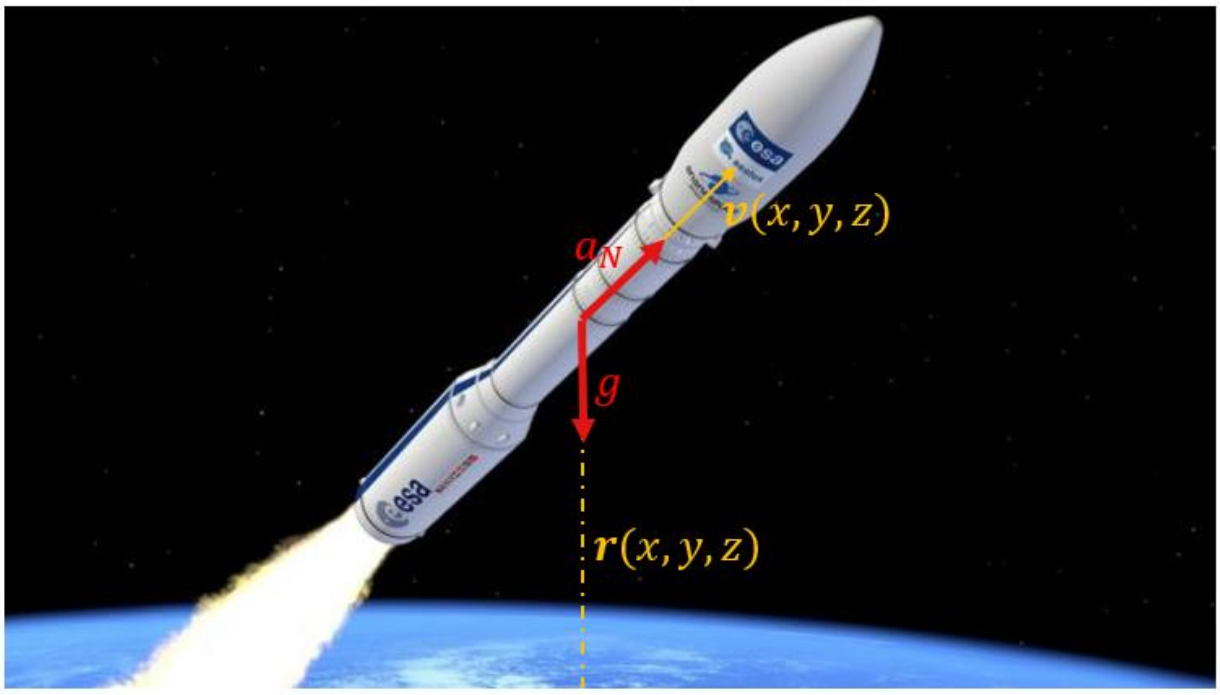

Figure 4: Forces on the boosting vehicle. The main accelerations present on the motion of a rocket during the boost phase are the acceleration due to thrust and the acceleration due to the gravitational forces. Accelerations associated to the influence of forces such as Coriolis, centrifugal, and drag can be neglected during this phase due to their smaller impact.

The model that represents the full motion of the vehicle under study is the following

$$\boldsymbol{a} = \boldsymbol{a_T} + \boldsymbol{a_G} \qquad (2)$$

where $\boldsymbol{a_G}$ is the gravitational acceleration and can be modeled according to any specific degree and order of spherical harmonics. Note the equivalency of Equation (2) with

Equation (45) in Chapter 4. As it was explained in [3] and briefly restated in Chapter 1, the motion model used to create in-house trajectories is in essence the same one used by the EKF filtering tool. Thus, the results generated by this tracker tend to be better than what it should be actually expected for real applications. In order to obtain more realistic results of the performance of the EKF filtering tool, external trajectories are included in the analysis carried out in the following chapters, see Table 3. Moreover, it might be the case that the flight profile of the gravity turn maneuver is also modified from the minimum energy (maximum range) trajectory. This can be done by using a smaller (lofted trajectory) or a larger (depressed trajectory) delta-kick angle. In these cases, tracking of the launching vehicle could become more challenging, as the trajectory is not as expected. In this way, three types of in-house trajectories are studied in the performance analysis of this thesis, as shown in Table 2.

Finally, besides the trajectory profile that follows the tracked vehicle, one of the most important aspects to properly model in the trackers is the thrust template – i.e. the behavior of the (thrust) acceleration with respect to time during boost. As it is explained in Section 2.2, many different types of thrust templates exist and the one used to generate the in-house trajectories is relatively simple. Thus, the utilization of external trajectories with a more complex template allows characterizing more accurately the performance of the studied filters in this thesis.

2.3. Sequential Filtering

Filtering, in a broad sense, is concerned with the extraction of signals from noise. In other words, filtering is a process used to estimate a signal that is based on noisy measurements [17]. Any signal $s(t+\alpha)$ is correlated with the measurements $y(t)$. This is such the case of a signal that is corrupted by noise. The measurements $y(t)$ are then input into the 'filter' in order to produce an output estimate $g(t)$. Any true signal can be observed in order to produce a noisy measurement to be used in a filter with the goal of obtaining a more precise estimate. The importance of filtering resides in developing a robust filter that combines information from different sources such as system and measurement models in order to provide an accurate estimate of the state of a moving object [18].

Since the tracking process must be carried out in real-time, observations are obtained *sequentially* – i.e. one after another. Therefore, this makes necessary to use a sequential filter, in opposition to a batch filter [19]. There are many different types of sequential filters but the most commonly used is the Kalman filter [20], which in turn has also many different possible formulations. In any case, a Kalman filter is a method for sequential, or recursive, parameter estimation. This method considers a state vector and its corresponding covariance matrix for a given time, or epoch[†]. Then, the algorithm that defines the Kalman filter has two steps: propagation and update of the state. In the propagation step, the state is propagated from

[†] An epoch is a specific time of the tracking filter. One epoch and the following are separated by a number of seconds that define the timestep size. Thus, usually an epoch is the time in which a new observation becomes available. Nevertheless, if no observations are available but the filtering process is still taking place, the epoch time is obtained from the previous epoch time plus the duration of the timestep.

epoch t_1 to epoch t_2 using a dynamic model. In the update step, observations are combined with the propagated state in order to account for the possible deviations due to a poor dynamic model or a large initial uncertainty. On the other hand, these observations are influenced by measurement noise. Therefore, with the Kalman filter algorithm it is expected that both the deficiencies of the propagation models and the inaccurate observations are (sub)optimally reduced in order to obtain an estimation of the target. Through a state noise compensation algorithm, system noise is added to the propagation step in order to avoid the Kalman filter to become inert to new observations. The general equations that define the Kalman filter are provided below. Note that several deviations from the typical Kalman filter equations have been used in order to improve the robustness of the Kalman filter. The equations presented are used for both the PKF Tool and the EKF Tool (explained in Section 2.4), unless otherwise indicated. For a specific formulation of the Kalman filter equations for each of the Tools the reader is referred to Chapter 4.

Mathematical Description of the Kalman Filter

A dynamical system can provide the future state of a system with known initial state. In this way, in continuous processes, a differential equation contains the dependence of the current state $\boldsymbol{S}$ from a previous one [21]. This relationship is provided in the system model

$$\dot{\boldsymbol{S}}(t) = F\big(\boldsymbol{S}(t)\big) \tag{3}$$

The propagation of the state is performed with the previous equation. However, an error in the initial estimate $\boldsymbol{S}(t_0)$ or in the system dynamics model F will divert future estimates from reality. Therefore, a state error covariance matrix is introduced to account for these imperfections. Furthermore, measurements of the target are included into the system to increase the accuracy of the propagated state. This can be done by including a measurement model that contains the relationship between the target state and the observations $\boldsymbol{y}(t)$ made by the sensor(s)

$$\boldsymbol{y}(t) = C\big(\boldsymbol{S}(t)\big) + \boldsymbol{\varepsilon} \tag{4}$$

where $\boldsymbol{\varepsilon}$ is a vector of residuals containing the measurement error due to sensor noise. Thus, the purpose of the Kalman filter is to solve the system given by Equations (3) and (4).

After the filter has been initialized, real-time recursive estimation of the target (in this case a launching vehicle) state starts. The filtering process is explained in Section 3.4. A more detailed derivation of the Kalman filter algorithm can be found in Gelb [20]. The rest of this section describes in more detail some of the Kalman filter equations.

Propagation Step

The propagation of the state is performed differently for the EKF-Tool and the PKF-Tool. In an extended Kalman filter, linearization of the previous system of equations is carried out.

The equations of an EKF are provided by Tapley et al. [10]. Numerical integration of the state and covariance is used for the propagation of the EKF using a Shampine-Gordon integrator [22]. More specifically, propagation of the error covariance of the state is carried out with the following State Noise Compensation (SNC) algorithm [10]

$$\dot{P}_k = AP_k + P_k A^T \quad (5)$$

where A is the linearized dynamics matrix of the model, as provided in Section 4.1. Process noise Q is added after propagation of the error covariance. Process noise is added in the SNC algorithm in order to reduce the optimism in $\bar{P} = P_{k|k-1}$, so the Kalman filter does not 'stall', meaning that it becomes inert to new observation information added at t_k.

$$\bar{P}_Q = \bar{P} + Q \quad (6)$$

Propagation of the PKF state and covariance are carried out using Euler integration [17]

$$\bar{S}(t) = \Phi(\mathrm{t} - \mathrm{t}_0) S(t_0) \quad (7)$$

$$\bar{P}_Q(t) = \Phi(\mathrm{t} - \mathrm{t}_0)\mathrm{P}(\mathrm{t}_0)\Phi^{\mathrm{T}}(\mathrm{t} - \mathrm{t}_0) + Q_k \quad (8)$$

where $\bar{S}$ and $\bar{P}$ are the state and covariance at time t, Φ is the state transition matrix as defined in Section 4.2, and Q_k is the following discrete process noise matrix [17]

$$Q_k = \int_0^{T_s} \Phi(\tau) Q \Phi^T(\tau)\, d\tau \quad (9)$$

where Q is the continuous process noise matrix whose elements are obtained from the power spectral density PSD as explained in Section 4.2 and T_s is the timestep length $T_s = t_k - t_{k-1}$.

Update Step

The relation between the state and the observations is contained in the measurement transition matrix H. The Kalman gain matrix at epoch k is calculated as

$$K_k = \bar{P}_Q H_k^{\mathrm{T}} S_k^{-1} \quad (10)$$

where the innovation matrix S is defined as

$$S_k = H_k \bar{P}_Q H_k^{\mathrm{T}} + R_k \quad (11)$$

being R the measurement noise covariance matrix whose formulation is explained in Section 4.3. For the update of the error covariance, the Joseph form [23] is used in order to reduce numerical errors and thus ensure that the resulting matrix is positive semidefinite, as this will

be necessary for future post-processing calculations – e.g. in the computation of the NCI index, Section 5.4.

$$\hat{P} = (I - K_k H_k)\bar{P}_Q(I - K_k H_k)^T + K_k R_k K_k^T \quad (12)$$

In the previous equation, I is an identity matrix of size defined by the product of the Kalman gain and the measurement transition matrices. Finally, the state is updated as

$$\hat{S} = \bar{S} + K_k \delta_k \quad (13)$$

where δ_k is the measurement residual defined as

$$\delta_k = y_k - H_k \bar{S} \quad (14)$$

Being y_k the observations at the current epoch.

2.4. Linear and Nonlinear Kalman Filters

In the previous section, the Kalman filter was introduced and its general equations were provided. The importance of the Kalman filter within the context of this thesis is that the developed and analyzed tracking tools are based on this filter. On one hand, a tracking tool previously developed at Airbus based on an extended Kalman filter has been analyzed. This version of Kalman filter is used when the system to be tracked is highly nonlinear. On the other hand, a second tracking tool based on a polynomial Kalman filter has been developed in order to compare its performance with the EKF-Tool. The polynomial Kalman filter is a linear filter so the tracked system is assumed to be linear. In general, the Kalman filter equations are not affected if the KF version used is linear (polynomial) or nonlinear (extended). The main differences reside in the way the KF matrices – dynamics matrix in the propagation step, measurement transition matrix in the update step, etc. – are formulated, as shown in Chapter 4.

In this section, a description of the extended Kalman filter is provided. Then, a review of the polynomial Kalman filter is given. Finally, the reasons and hypotheses of why the developed PKF-Tool is considered to overcome in performance the EKF-Tool are provided.

Extended Kalman Filter

The filter that is currently implemented in the EKF-Tool is an extended Kalman Filter. Unlike regular Kalman filters, an extended Kalman filter can handle nonlinear systems by using approximate expressions for propagating the conditional mean of the state and its associated covariance matrix (linearization of the system models) [20]. More specifically, the gravity turn dynamic model of the EKF-Tool makes necessary to employ nonlinear filters such as the extended Kalman filter. This filter linearizes the dynamic model in each propagation step in order to propagate the error covariance of the state. Linearization of the

model is not used for propagation of the state since this can be done with numerical integration.

Polynomial Kalman Filter

A polynomial Kalman filter is a simpler version of the extended Kalman filter. In fact, the EKF is usually derived from the PKF when the complexity of the system requires the use of nonlinear filters. Therefore, extended Kalman filters are used when a high accuracy is possible to be obtained. In this way, extended Kalman filters are commonly used for orbit determination applications [10] when accurate observations are available and offline initialization methods are possible, since real-time tracking is not required. Nevertheless, for tracking launching vehicles the utilization of extended Kalman filters seems to overcomplicate the tracking process [9]. This is because the initial conditions of the launch vehicle are usually unknown, the dynamics of the vehicle's motion – thrust template and direction – are also unknown, and the observations are obtained sequentially, so only online tracking is possible for the goal intended within the context of this thesis.

In a polynomial Kalman filter the system dynamics F is a linear model thus, Equation (3) can be rewritten as

$$\dot{S} = FS \tag{15}$$

For this case, the system dynamics matrix F is time invariant so a transition matrix Φ can be used to exactly propagate the state forward from any time t_0 to time t [17] using Equation (7).

This fundamental matrix can be computed either using the Laplace transform or a Taylor series expansion, being the second option preferred for the development of the PKF-Tool within the work performed for this thesis as the system dynamics is (almost) time invariant

$$\Phi(\mathrm{t}) = I + Ft + \frac{F^2t^2}{2!} + \cdots + \frac{F^nt^n}{n!} + \cdots \tag{16}$$

If the differential equations describing the system behavior yield polynomial signals – that is, the true motion of the tracked vehicle follows a polynomial curve – the resultant polynomial Kalman filter will have a special type of state transition matrix. For example, for a first-order system – a system S with a constant derivative $\dot{S}$ in time – the following dynamics matrix is obtained

$$\dot{S} = FS \rightarrow \begin{bmatrix} \dot{s} \\ \ddot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s \\ \dot{s} \end{bmatrix} \tag{17}$$

The same approach is followed to develop the state transition matrices of the (linear) dynamic models used in this thesis, as shown in Section 4.2.

In this special case, the system is completely deterministic – the tracked vehicle's true motion is constant ($\ddot{s} = 0$). In reality, the motion of the vehicle cannot completely be determined as there are many factors that have influence on its trajectory such as drag, wind gusts, misalignment of the rocket engine, etc. Moreover, if the thrust template of the vehicle is unknown, it is impossible to model deterministically the system's behavior. For these cases, it is very useful to consider the system behavior – e.g. the motion of the vehicle – as a stochastic process. In this way, it is assumed that the differential equations that describe the system behavior yield *almost* polynomial signals. The divergence from a true polynomial is modelled by considering the last element of the state as a random process. This randomness is included in the process noise matrix as a zero-mean element with a standard deviation value given by the power spectral density.

For the work performed in this thesis, two types of stochastic processes are considered: Wiener and Markov processes [19]. A Wiener process $s = W$ is regarded as the integral form of a white noise Gaussian process[‡] dW

$$W(t) = \int_0^t \frac{dW(\tau)}{dt} d\tau \qquad (18)$$

That is, whereas a white noise process is produced by randomly choosing each sample independently, a Wiener process is produced by adding a random offset to each sample in order to obtain the next sample. Therefore, the stochastic variable is a process with independent increments as shown in Equation (19), where w is the process noise (white Gaussian noise). Note that, in truth, the derivative of the Wiener process is nonexistent and the above explanation is not formally rigorous [24].

$$\dot{s}(t) = w \qquad (19)$$

In fact, a Wiener process is a special case of Markov stochastic process. As it was explained before, a stochastic process is one whose value changes over time in an uncertain way and we only know the distribution of the possible values of the process at any time point. In the previous example of Equation (17), if the process is assumed stochastic instead of deterministic, the velocity $\dot{s}$ of the system at time t_k would vary from the previous time t_{k-1} instead of remaining constant. A Markov process claims that the likelihood of the state at any future time depends only on its present state but not on any past states [25].

In this thesis, an additional type of Markov process is defined for a dynamic model different from the Wiener-based process models. For this type of Markov process model, there exists time correlation of the stochastic variable; unlike for Wiener process models, in which the

[‡] Note the reader the naming convention of the different PKF dynamic models that are described in the next chapters: a White Noise Jerk process is equivalent to a Wiener acceleration process, as a White Noise process is the derivative (jerk) of a Wiener process.

white noise is isolated in time since its value at one time is uncoupled of any other time. In this way, in the Markov process model defined for this thesis the value of the stochastic variable – e.g. the acceleration – at one time depends on the value at the previous time

$$\dot{s}(t) = Ks(t) + w \tag{20}$$

2.5. Kalman Filtering for Tracking Boosting Vehicles

After analyzing the problems and limitations of the EKF-Tool previously developed at ADS during the Literature Study [3] that led to the development of this thesis, it was concluded that a filtering method based on the stochastic models explained above was the best approach to tackle this tracking-problem. More specifically, it is argued that a simpler method based on linear polynomial Kalman filters in the same way as it was done by Zarchan [9] can provide a more robust estimator that overcomes the limitations of the EKF-Tool. Similarly as the EKF-Tool, a PKF-Tool has been developed based on this polynomial Kalman filter for this thesis in order to compare their performance.

The purpose of this section is to provide a list with the reasons why the PKF-Tool is believed to have a better performance than the EKF-Tool. Each of these reasons, thus, leads to the development of a hypothesis that will be tested in the performance analysis carried out in Chapter 5 and Chapter 6.

Besides measurement-origin uncertainty, the following are the main factors that have an influence on the problem of tracking ballistic vehicles: uncertain initial state, unknown vehicle's dynamics and thrust template, influence of environmental factors – e.g. drag –, and need for real-time estimation. All these factors can be controlled by choosing a certain type of filter. For example, in order to carry out real-time estimation, a recursive filter shall be chosen in contrast to a batch filter, as it was seen in Section 2.3. The following discussion summarizes for each of the above factors why it is hypothesized that the PKF has a better performance than the EKF within the tracking problem analyzed in this thesis.

Unknown Initial State

There is no guarantee that an extended Kalman filter will perform better or even achieve convergence when realistic initialization errors are considered [9]. Moreover, the need of using a numerical integrator for the propagation step makes necessary to use an initial state that is close to what the filter's dynamic model would expect in order to avoid divergence of the integrator. Therefore, an accurate state and covariance initialization is required for the EKF. On the contrary, linear polynomial Kalman filters are very robust and are insensitive to filter initialization issues. Furthermore, a decoupled filter that uses pseudomeasurements (see Section 4.3) will also be more insensitive to initialization errors in comparison to a coupled filter such as the EKF.

Therefore, it is hypothesized that the PKF-Tool will not need any robust initialization scheme and any initial state and covariance will rapidly update so the filter achieves convergence. This hypothesis is tested in Section 5.3.

Real-time tracking

Both of the Kalman filters studied in this thesis allow real-time tracking as both of them are recursive filters. Nevertheless, the initialization method necessary for the EKF in order to overcome the initialization issues described above uses a least-squares-based (polynomial fit) algorithm. Such an algorithm is of batch type; that is, the EKF needs the first available k initial observations in order to perform its initialization. Therefore, real-time tracking for the EKF is not possible until epoch k+1. Conversely, the simple initialization method of the PKF only uses the first available observation so real-time tracking is always possible. The reader is referred to Chapter 4 for an in-depth explanation of the initialization methods of both filtering tools.

Vehicle Dynamics and Environmental Factors

The main reason why filtering needs to be carried out is that the motion of the launching vehicle is completely unknown and only noisy observations of its trajectory are available. The largest constraint of the tracking problem at hand is that the launching vehicle is completely unknown. This means that its characteristics such as vehicle dynamics and exact launch location cannot be identified before performing tracking and only hypotheses can be made regarding the dynamics of the launching vehicle. If the vehicle's dynamics and the effect of atmospheric drag, wind gusts and any other forces affecting to the dynamics of the vehicle were identified, it would be possible to know its full trajectory, provided that the initial state was accurately estimated, so no filtering would be necessary. Nevertheless, the thrust template of the launching rocket is not available, and the effect of environmental factors could be of any kind.

Therefore, the selected dynamic model for the filtering tool becomes of greatest importance. A poorly designed dynamic model will not be able to properly estimate the trajectory of the vehicle, unless large amounts of process noise, which accounts for the deviations in the motion from the filter's model, are used in order to avoid filter divergence. Nonetheless, a large value of process noise will result in a very low accuracy of the filter. As it has already been introduced and will be discussed in the next chapters, the EKF performs quite well for in-house trajectories since the filter's dynamic model is equivalent to the model used to generate the true motion of the vehicle. However, when external, real-missions, trajectories are used, the process noise must be increased in order to achieve a stable filter and thus make tracking possible. This happens due to the large mismatch between the filter's dynamic model and the true motion of the external vehicle, which is unknown.

The main purpose of introducing a simpler Kalman filter such as the polynomial Kalman filter is that its dynamic model, although inaccurate in general, can be used to track any type of trajectory obtaining a similar performance, regardless of the vehicle's motion. Moreover,

since this type of filter is linear, it allows using stochastic processes within its dynamic model allowing to model process noise in a more effective way than for a nonlinear filter such as the one used for the EKF-Tool. Thus, it is hypothesized that the combination of simpler dynamic models based on stochastic processes and the more effective process noise modelling approach used for the PKF-Tool will provide better performance in general in comparison with the EKF-Tool. This performance hypothesis is the main subject studied in Chapter 4 and Chapter 5, in which both tools' performance is analyzed for external and in-house trajectories.

Chapter 3

Simulation Environment

The purpose of this chapter is to explain the Simulation Environment (SE) in which the tracking tools introduced in the previous chapter are developed. In this way, the information provided in this chapter is exclusive to the SE used for the development of the work of this thesis. Thus, the filters analyzed in this thesis are standalone tools that can be also used outside of this SE, provided that the input observations from Section 2.1 are available and the filters are properly initialized. Due to the limited access to real data of launching vehicles, it is necessary to generate "in-house" trajectories that are used to generate filter inputs and to analyze the performance of the tracking tools. Moreover, the trajectories during boost of the Vega rocket for three different real missions are available, which are also used within the SE by the tracking-tools for output generation and performance analysis. The way in-house trajectories are generated, and external trajectories are read by the SE is provided in Section 3.1.

Once the actual trajectories have been generated within the SE, the noisy measurements obtained by the observing satellites are randomly simulated from the trajectory data, as explained in Section 3.2. These observations are used as input to the EKF and PKF filtering tools. The tools carry out the tracking process and an estimated trajectory is generated (see Section 3.4). Based on the random generation of the observations, both of these steps are performed within a Monte Carlo environment in order to obtain meaningful statistics from the tools' output.

Finally, the generated output for nMC runs together with the "true" trajectories (external or in-house) are used for performance analysis and post-processing, as explained in Section 3.5. The SE environment is summarized in the following Figure 5.

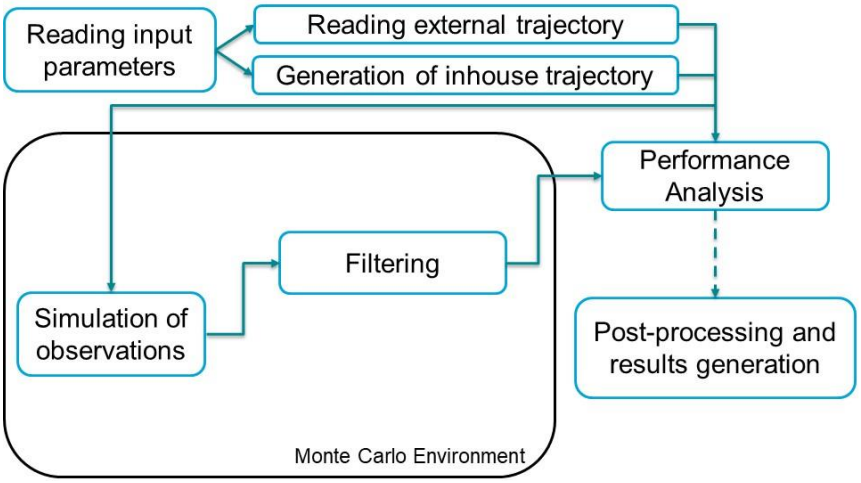

Figure 5: Simulation Environment. The simulation environment consists of three different main parts: the input and trajectory generation step, the filtering step (within a Monte Carlo environment) and the post-processing step, which includes the performance analysis

Simulation Environment Input Parameters

A special attention must be put to explain the different input parameters that are used by the SE in general, or by any of the specific steps within it. In this way, three main groups of input parameters to the SE are considered. The first group corresponds to those SE global inputs used for the generation of the trajectories and the simulation of the observations that are common to the two tracking tools. The second group encloses the specific tracking-tool inputs in the filtering step of Figure 5. Finally, the third group encloses those filter inputs or environmental conditions that affect to filter performance independently of the tracking tool used. The reader is referred to Section 5.3 for a more detailed explanation of the second and third groups of input parameters. In the following Table 1, the global input parameters of the SE are provided.

	Name	Description	Value
General	nMC	Number of MC runs within the SE	$[1, 100]$
In-house Trajectory Generation	n_g	Degree and order of geopotential coefficients	2
	LP	Launch point of the rocket (see Table 2)	NA
	Trajectory	see Table 2	NA
	Template	see Table 2	NA
Satellites Observation Simulation	$nSat$	Number of observing satellites	2
	$orbParam$	Orbital parameters	NA
	Noise	Noise of the measuring sensors	NA
	Bias	Bias of the sensors (not used, see Chapter 1)	NA

Table 1: SE global input parameters

3.1. Generation of Trajectories

The trajectories used by the SE are from two different sources: in-house and external (more specifically, from the Vega rocket). A series of input parameters are necessary for the generation of the in-house trajectories. For the external Vega rocket trajectories, a file containing the time and x, y, z position and velocity in ECEF coordinates is read.

In-house Trajectories

The SE is capable of generating trajectories using a series of input parameters. These in-house trajectories are based on a simple gravity turn (GT) maneuver (see Section 2.2). The launch coordinates λ and ϕ (longitude, latitude) and the launch azimuth angle β[§] are selected input-user values, then a delta-kick δ is performed after a certain time t_v from launch in order to enter in the GT. The rocket is accelerating until burnout t_b[**]. The initial acceleration and exhaust velocity parameters are also provided as inputs. For the performance analysis carried

[§] A launch azimuth of zero degrees indicates a purely eastwards launch. The right-hand rule is used to determine the direction of the launch. For example, a launch azimuth between 90° and 180° corresponds to a trajectory in the northeast quadrant.

[**] t_b is the time from start of the staging phase (or launch) until end of the staging phase.

out in Chapter 5 and Chapter 6, three different in-house trajectories have been used depending on the type of GT maneuver performed (nominal, lofted, and depressed) [16]. In Chapter 6 the launch coordinates and launch azimuth are varied. The three in-house trajectories used in this thesis have two stages without inter-stage coasting. In the following Table 2 the characteristics of the three in-house trajectories used for the performance analysis are provided.

Parameter		Value
Launch longitude λ		-52.775°
Launch latitude ϕ		5.201°
Launch Azimuth β		138.000°
Burnout time, stage 1 t_{b_1}		110 s
Initial acceleration, stage 1 a_{0_1}		17.0 m/s^2
Exhaust velocity, stage 1 U_1		2500.0 m/s
Burnout time, stage 2 t_{b_2}		90 s
Initial acceleration, stage 2 a_{0_2}		17.0 m/s^2
Exhaust velocity, stage 2 U_2		1900.0 m/s
Nominal (minimum energy) Trajectory	Time in vertical phase t_v	25 s
	Gamma kick angle δ	74.0°
Lofted Trajectory	Time in vertical phase t_v	15 s
	Gamma kick angle δ	69.7°
Depressed Trajectory	Time in vertical phase t_v	25 s
	Gamma kick angle δ	79.0°

Table 2: Characteristic of the in-house trajectories

Using these input parameters, the initial position (at launch) in ECEF coordinates of the vehicle can be obtained for the propagation of the state (velocity is zero). The motion model used for the numerical integration of the trajectory is obtained from the rocket Thrust Equation $ma = -\dot{m}U$, from which the following ode is derived [26]

$$\frac{da}{dt} = \frac{a^2}{U} \tag{21}$$

where U is the exhaust velocity of the rocket. Using the simplifications given by [27] on Equation (21) the thrust acceleration as a function of time is obtained

$$a_T = \frac{1}{\frac{1}{a_0} - \frac{t}{U}} \tag{22}$$

Therefore, the thrust acceleration can be plotted against time using the previous expression. As an example, Figure 6 shows the thrust profile obtained with Equation (22) for two

sample values of initial acceleration and exhaust velocity. It can be seen that with this type of model the acceleration increases rationally, achieving an asymptote for $t = U/a_0$.

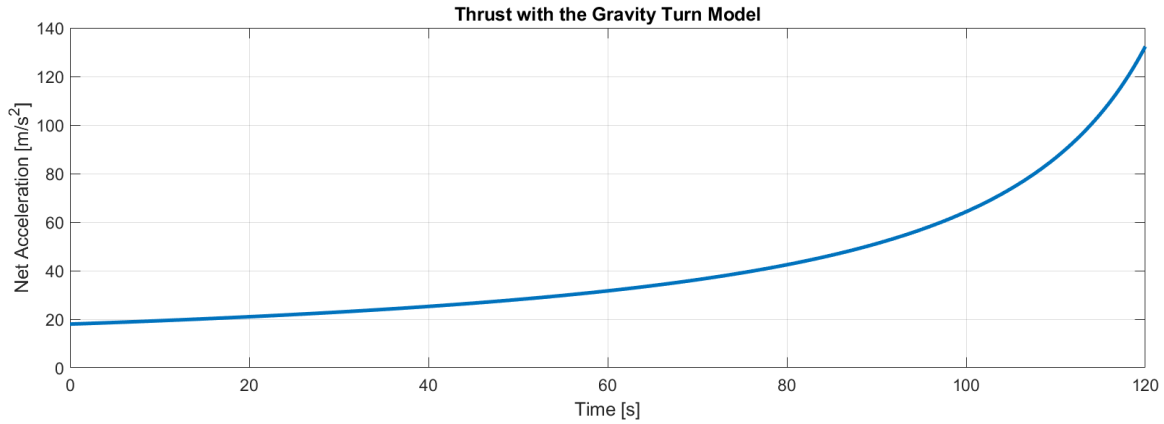

Figure 6: Thrust profile for an in-house generated trajectory. For in-house trajectories, the thrust curve is relatively simple. The thrust follows a rational curve with a defined asymptote. That is, if the thrust time is long enough, the acceleration of the launching vehicle will tend towards infinite. However, this is not possible in real life.

Using this net thrust acceleration magnitude and the GT Expression (1) given in Section 2.2, the position and velocity components of the launching vehicle in ECEF coordinates are obtained after numerical integration (a Shampine-Gordon [22] integrator is used by the SE for propagation purposes) with the dynamic model given by Equation (2) using a degree and order $n_g = 2$ for the gravitational acceleration as follows

$$\boldsymbol{a_G} = -\mu \frac{\boldsymbol{r}}{|\boldsymbol{r}|^3} \qquad (23)$$

where $\boldsymbol{r}$ is the position vector in ECEF coordinates and μ the standard gravitational parameter. Note that, for generating in-house trajectories, the effect of drag and any other forces (e.g. Coriolis or centrifugal) are not taken into account. Therefore, the in-house trajectories generated within the SE are relatively simple compared to trajectory profiles from any actual mission. Moreover, note the equivalency of Equation (2) with the dynamic model used for the EKF-Tool – Equation (45). This equivalency suggests that the results obtained by the EKF will be quite accurate to the tracked in-house trajectory (only deviated from the truth due to the random component of the observations and the initial estimate). Thus, in order to carry out a fairer performance comparison (i.e. how close to the truth the filter's estimates are) of the filtering tools and to analyze their behavior tracking real vehicles with a more complex thrust template, three different external trajectories based on the Vega rocket are used within the SE.

External Trajectories

Three different external trajectories are used within the context of this thesis. These trajectories are obtained from the original, realistic 6-DOF Vega simulator available to Airbus in Bremen. Although the thrust template is relatively similar and the launch location (Kourou) is the same for the three of them, their differences reside in different launch

azimuths. Moreover, a Vega rocket typically has four stages until orbit insertion. However, for the purpose of this thesis only the two first stages of the rocket are taken into account in the analysis (i.e. it is assumed that the rocket enters in ballistic phase after burnout of the second stage) in order to increase simplicity and reduce the computation time within the SE. Unlike in-house trajectories, external trajectories are based on real missions and, thus, the thrust template (see Figure 7) is more complex than the profile of the trajectories generated by the model shown in Figure 6.

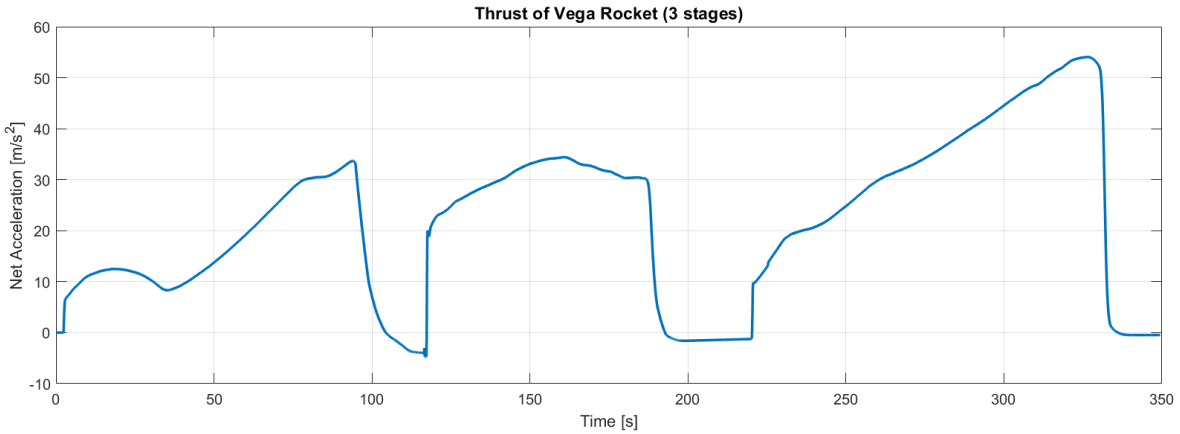

Figure 7: Thrust profile for an external launch vehicle calculated by finite differences from velocities. Unlike with in-house trajectories, the thrust curve of an external vehicle such as the Vega rocket has a more complex profile, having phases in which the thrust increases (linearly or exponentially, for example), decreases, or is zero (coasting).

The name and launch direction of the external trajectories used for the performance analysis of this thesis are provided in Table 3. All of them are launched from Kourou $(-52.775°, 5.201°)$ and have two stages, with a stage 1 burnout time of $t_{b_1} = 116.0$ s and a stage 2 burnout time of $t_{b_2} = 88$ s.

External Trajectory	Description
Vega E	Launched eastwards for payload insertion in an equatorial orbit
Vega NE	Launched northeastwards for insertion in sun-synchronous orbit
Vega N	Launched northwards for payload insertion in polar orbit

Table 3: Description of the external trajectories

3.2. Simulation of Observations

Once the trajectories have been read or generated by the SE, this data is used for simulating the measurements obtained by the observing satellites. Each of the sensors obtain measurements in the LHZ frame. Thus, the x, y, z position in ECEF coordinates obtained in the previous step undergoes a frame transformation into the local horizontal frame of each satellite [28]. To do this, the location of the satellites must be known. It is assumed that the satellites location is purely in a GEO orbit. Therefore, their location is fixed in the ECEF reference frame. Note that for the case studied in this thesis, there are two observing satellites, although the following equations can be used for n satellites. The azimuth coordinate in the satellite i reference frame is

$$\theta^{(i)} = \text{atan}\left(\frac{\Delta r_y^{(i)}}{\Delta r_x^{(i)}}\right) \quad (24)$$

and the elevation coordinate as seen by the sensor in the satellite i LHZ frame from the ECEF position of the vehicle is given by

$$\varepsilon^{(i)} = \text{atan}\left(\frac{\Delta r_z^{(i)}}{|\Delta \boldsymbol{r}^{(i)}|}\right) \quad (25)$$

where $\Delta r_j^{(i)}$ is the distance along the $j = x, y, z$ axis from the launch vehicle r_{LV} and the i observing satellite $r_{sat}^{(i)}$

$$\Delta r_j^{(i)} = r_{j,LV} - r_{j,sat}^{(i)} \quad (26)$$

and $\left|\Delta \boldsymbol{r}^{(i)}\right|$ is the magnitude of the distance between the launch vehicle and satellite i.

Note that the previous procedure is carried out in a batch form within the SE before entering in the filtering step. That is, all the epochs from first observation time (after the launcher surpasses the cloud tops, see Section 2.1) until burnout of the last stage (including the coasting periods, although in reality there are no available observations then, this is done to simplify the code) are converted from ECEF to LHZ coordinates at once.

The obtained values above are the true azimuth and elevation coordinates. Therefore, it is necessary to perform a final calculation to simulate real observations. This is done by adding a random component drawn from the standard normal distribution $\mathcal{N}(\mu, \sigma^2)$

$$\varepsilon_n^{(i)} = \varepsilon^{(i)} + \mathcal{N}(0, \sigma_n^2) \quad (27)$$

being σ_n the standard deviation of the measurement noise, which is a user input with a value specified by Airbus (see Section 2.1).

3.3. Triangulation of Observations

Before entering in the filtering step, a final transformation back to the ECEF frame from LHZ must be carried out. This triangulation of the observations is done for two reasons. First, the EKF-Tool needs to use the first n available measurements for initialization purposes and they must be in ECEF coordinates. Second, there are two versions of the PKF-Tool that use a Cartesian measurement model (see Section 4.3) so the observations must be in the ECEF frame. The triangulation is performed with the following "composite measurements" technique provided in [14] and [29]. Note that the sensors read the azimuth and elevation coordinates of the launch vehicle. Therefore, it is necessary to use a pair of sensors (four coordinates) in order to compute the x, y, z cartesian coordinates by triangularization. The

additional coordinate from the pair of observations provides a double solution. In order to have just one, the averaged value from the two solutions is obtained.

First, the difference vector between satellite $i = 1$ and $j = 2$ is obtained

$$\Delta \boldsymbol{r}_{sat} = \boldsymbol{r}_{sat}^{(2)} - \boldsymbol{r}_{sat}^{(1)} \tag{28}$$

Then, the unitary direction vector from each satellite i to the launch vehicle is computed

$$\boldsymbol{u}^{(i)} = \left[\cos\left(\varepsilon^{(i)}\right) \cdot \cos\left(\theta^{(i)}\right) \quad \cos\left(\varepsilon^{(i)}\right) \cdot \sin\left(\theta^{(i)}\right) \quad \sin\left(\varepsilon^{(i)}\right)\right]^T \tag{29}$$

The scalar distance from the launch vehicle to satellites $i = 1$ and $i = 2$ is

$$\left|\Delta \boldsymbol{r}^{(1)}\right| = \frac{\left(\boldsymbol{u}^{(1),T} \Delta \boldsymbol{r}_{sat} - \boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)} \boldsymbol{u}^{(2),T} \Delta \boldsymbol{r}_{sat}\right)}{1 - (\boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)})^2} \tag{30}$$

$$\left|\Delta \boldsymbol{r}^{(2)}\right| = \frac{\left(\boldsymbol{u}^{(2),T} \Delta \boldsymbol{r}_{sat} - \boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)} \boldsymbol{u}^{(1),T} \Delta \boldsymbol{r}_{sat}\right)}{1 - (\boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)})^2} \tag{31}$$

With this, a double solution of the measured location of the launch vehicle is obtained

$$\boldsymbol{r}_{meas}^{(1)} = \boldsymbol{r}^{(1)} + \left|\Delta \boldsymbol{r}^{(1)}\right| \boldsymbol{u}^{(1)} \tag{32}$$

$$\boldsymbol{r}_{meas}^{(2)} = \boldsymbol{r}^{(2)} + \left|\Delta \boldsymbol{r}^{(2)}\right| \boldsymbol{u}^{(2)} \tag{33}$$

And the final triangulated ECEF position of the launch vehicle is obtained by averaging Equations (32) and (33)

$$\boldsymbol{r} = \frac{\boldsymbol{r}_{meas}^{(1)} + \boldsymbol{r}_{meas}^{(2)}}{2} \tag{34}$$

Finally, note that the obtained vector $\boldsymbol{r}$ should be equivalent to the generated or read position vector of the launch vehicle plus a random component that follows a normal distribution (the normal distribution is applicable only to the raw measurement errors from the sensors, thus this is only true if the angles – i.e. measurement noise – are small). Therefore, this triangulation of the measurements was also used to validate that the simulation of observations was properly implemented.

3.4. Filtering: Boosting, Staging, and Coasting Launch Vehicles

In this section, a brief summary of the filtering step is provided. This step, and the way the filtering tools introduced in Section 2.4 are implemented in the SE is thoroughly analyzed in

Chapter 4. In short, the filtering step is sequentially carried out until burnout time of the tracked vehicle. Three main phases comprise this step: filter initialization, propagation, and update (see Figure 8). The purpose of this section is to provide a general overview of the filtering step and the implementations that were carried out within the context of this thesis.

Since the used true trajectories for the performance analysis carried out in this thesis corresponds to multistage rockets with possibility of staging, the SE was upgraded in order to allow a correct simulation for tracking this kind of vehicles. That is, since during coasting phases no observations are available, the SE should be able to detect these phases in order to not provide observations to the filter. Moreover, the filters had to be modified to take into account staging. In this way, two reinitialization methods are implemented, depending whether there is coasting or not after the staging event.

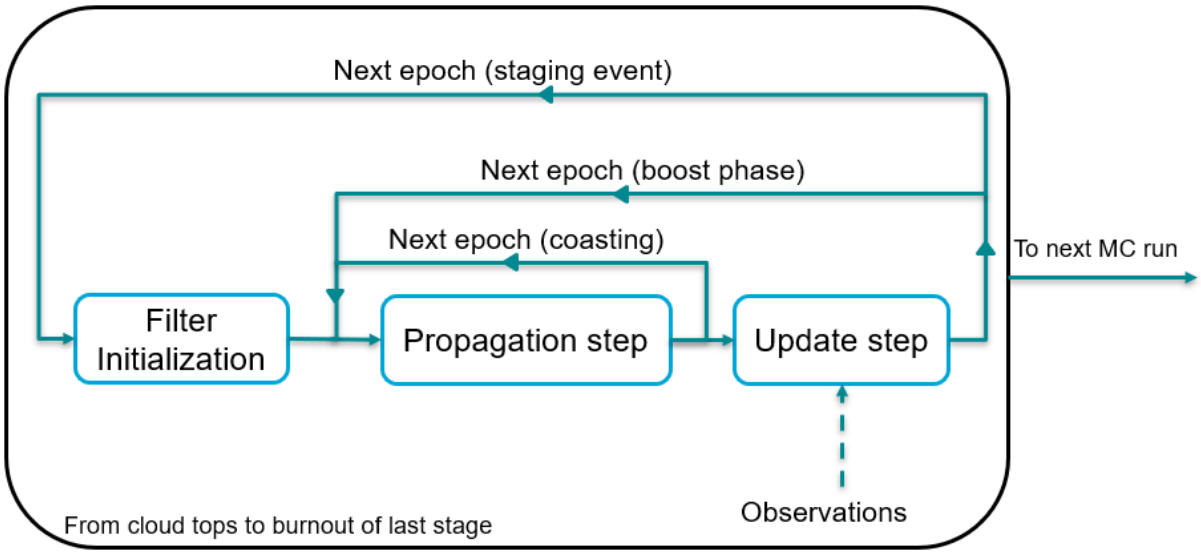

Figure 8: Filtering step within the SE. The filtering step consists of three different sub-phases that are iterated during the tracking period, depending on the conditions of the tracked trajectory.

The filtering step works as follows:

1) The filter (EKF-Tool or PKF-Tool) is initialized. Any input parameters are used, and the state vector for the first epoch is obtained. The methods used for initialization can be found in Chapter 4.
2) Once the filter is initialized, the propagation step is entered in which the state is predicted for the next epoch using numerical integration (EKF-Tool) or simpler Euler integration (PKF-Tool). The same applies for the associated state error covariance matrix of the state (see Section 2.3).
3) The update step of the estimator is entered, in which observations are incorporated to correct the predicted value of the state.
4) Steps 2-3 are sequentially performed during the boost phase from cloud tops or beginning of stage until burnout time t_b, when there are available observations. If burnout of the stage has been reached (t_b is provided as an input for the SE), two options are possible.

a. There is no coasting between stages: in this case, the position and velocity elements of the filter after staging are the obtained ones for the previous epoch. The remaining elements of the state are reinitialized as in step 1 since the e.g. acceleration will abruptly change after staging. The same applies to the state error covariance matrix. This is the method used for the in-house trajectories.
b. There is an inter-stage coasting phase: in this case, the state vector is propagated using a coasting dynamic model (see Chapter 4) until observations become available again. The error covariance of the state is also propagated so no abrupt reinitialization is performed. This is the method used for the external trajectories, as they all have a coasting phase of about 2 seconds.

3.5. Output Generation

Once the filtered position and velocity have been obtained for nMC runs (see Figure 5), the results are stored to obtain statistical information about the performance of each tracking tool. The following parameters within the SE are stored in output files.

- State vector elements for each epoch and MC run
- State error covariance matrix elements for each epoch and MC run
- True trajectory (position and velocity)
- Simulated observations in LHZ reference frame
- Triangulated observations (ECEF reference frame)

The data in these files is then post-processed for performance analysis. The results of this performance analysis are provided in Chapter 5. The parameters that are obtained after post-processing are the following.

- Normalized Inversion Squared as well as the results of the NIS test (pass/fail) for each epoch
- Root Mean Squared Error at each epoch
- Non-Credibility Index at each epoch
- Eigenvalues and eigenvectors at burnout of position and velocity used to create the error ellipsoid
- Standard deviation of position and velocity at each epoch

Chapter 4

Tracking Boosting Vehicles

This chapter provides a detailed explanation of the filtering step from Section 3.4 and Figure 8. It serves as an introduction for the reader to the estimator-model on which this thesis is based. The measurements are assumed to be available at all times in which the launcher vehicle (LV) or rocket is in powered flight, regardless how these observations are obtained. Therefore, the model types explained in this chapter are perfectly valid for real mission applications, not limited only to simulated data.

In Chapter 2, the specific characteristics and differences of polynomial and extended Kalman filtering were explained. Then, an EKF-Tool and PKF-Tool based on these two types of Kalman filters were introduced. The purpose of this chapter is to provide a detailed analysis of the characteristics of the EKF and PKF. Namely, the goal of the description that will be carried out in this chapter is to completely define the filters used within the Simulation Environment (or for any potential real mission) for tracking ballistic vehicles launching from Earth.

First, the EKF-Tool and PKF-Tool will be exhaustively studied, focusing on the dynamic models of the filters including their differences, advantages and disadvantages. To this end, the inputs and outputs of the filtering tools will be investigated. Then, the different measurement models used for each filter (including the versions of the PKF) will be thoroughly explained. The focus will be put on analyzing the characteristics of the PKF, as the EKF used during the thesis is based on an extended Kalman filter developed by Airbus. In any case, the specific modifications performed to the EKF for this thesis will also be explained.

4.1. The EKF

The EKF is a tracking tool based on an Iterated Extended Kalman Filter (IEKF) developed by Airbus. Previous work showed that the best results of this filter were obtained with just one iteration, hence the naming to just EKF. Moreover, the state vector of the IEKF contained measurement bias as an element to be estimated. For this thesis work, these elements have been removed from the state for the final version of the EKF used for the thesis since it has been analyzed that estimations are not improved with this augmented state.

Thus, the implemented EKF uses two different dynamic models. The first one is based on a gravity turn motion. It is used for the propagation step of the tracked vehicle during boost. The second one, used for tracking coasting phases during inter-stage periods in the motion of the vehicle, is just a propagator with gravity as the only force acting on the vehicle (see

Chapter 3). For the update step, a polar measurement model that relates the Cartesian state vector with the polar LHZ measurements is used.

In the following subsections, the elements of the state vector will be defined together with the necessary initialization techniques for the EKF. The type of measurements used, and the way process noise is added will also be discussed. Then, a detailed analysis of the dynamic models used in the propagation step will be carried out.

4.1.1. Inputs and Outputs

EKF State Vector

The state vector of the EKF estimator is

$$S = [\boldsymbol{r^T} \quad \boldsymbol{v^T} \quad \alpha \quad \beta]^T \tag{35}$$

where

$$\boldsymbol{r} = [r_x \quad r_y \quad r_z]^T \tag{36}$$

is the position vector of the launcher vehicle in km, expressed in the ECEF coordinate system [29]; and

$$\boldsymbol{v} = [v_x \quad v_y \quad v_z]^T \tag{37}$$

is the velocity vector of the rocket in km/s, also expressed in ECEF coordinates. α is the magnitude of the thrust acceleration in km/s^2 and β is the velocity parameter in s^{-1}. Note here that α is, in fact, the net acceleration magnitude. That is, the thrust acceleration minus drag divided by the mass at time t. The velocity parameter is derived from the relation between mass flow rate and instantaneous mass of the rocket [30]. Actually, in view of the rocket thrust equation $T = m\alpha = -\dot{m}U$, β is the relation between thrust acceleration and rocket exhaust velocity U.

$$\beta = \frac{-\dot{m}}{m(t)} = \frac{\alpha}{U} \tag{38}$$

The covariance of the estimation error is given by[††]

$$P = E\left[(S - \hat{S})(S - \hat{S})^T\right] \tag{39}$$

[††] This is the real covariance; the estimated covariance is obtained using the Kalman Filter equations provided in Chapter 2. Note that Equation (39) also applies for PKF.

EKF Initialization

Initialization of the EKF is carried out using observations and predefined input values by the user. The explanation of state and covariance error initialization is explained hereafter.

For the position and velocity elements of the state, a polynomial fit based on a batch initialization method [31] containing the first n triangulated x, y, z measurements (see Chapter 3), or epochs, is used. The remaining elements of the state α and β are initialized with the following equations [32] [30]

$$\alpha_0 = -k|\boldsymbol{g}(\boldsymbol{r_0})| \quad (40)$$

$$\beta_0 = \frac{\alpha_0}{U_0} \quad (41)$$

where k is a user input chosen as $k = 2.5$, that is, the excess acceleration due to thrust; and U_0 is the initial exhaust velocity, also a user input parameter chosen as $U_0 = 2.5\ km/s$ since the exhaust velocity is unknown. These two input parameters for initialization are rough estimations and do not have to be excessively accurate as they have a low impact on filter output since the state is rapidly updated once more observations become available.

Note that this defined initial state is obtained after the rocket surpasses the cloud tops, when the first observation is available; and do not correspond to the state at launch. Moreover, the batch initialization approach of the filter for position and velocity do not allow performing real-time tracking of the target, and a total of $n\Delta t$ seconds, being Δt the timestep size, has to be awaited until real-time tracking is achieved.

For the initial error covariance matrix, the parameters are chosen according to the method described in Section 5.3. In any case, values that are too large or small will have a critical effect on filter output, as it may be possible that the integrator diverges in the propagation step if the value of the initial error covariance is too inaccurate. If the values are approximately close to what could be expected, the integrator will not diverge. Finally, the reader is referred to Section 2.4 to find a comparison of the potential disadvantages of this initialization method over the one used for the PKF.

4.1.2. State Propagation Models

This section focuses on the propagation step of the filtering process. There are two different dynamic models used in the EKF. First, a Gravity Turn dynamic model used for the boosting phases of the trajectory and then, a gravitational model used for the inter-stage coasting phases of the tracked vehicle will be analyzed.

Gravity Turn Model during Boost

The model used to propagate the state vector in time is expressed as

$$\dot{S} = [\dot{\boldsymbol{r}} \quad \dot{\boldsymbol{v}} \quad \dot{\alpha} \quad \dot{\beta}]^T = f(S,t) \tag{42}$$

This model is split into different parts. First, the propagation of the velocity parameter is obtained from the rocket thrust equation, or conversely, from Equation (38). Assuming that the net acceleration (thrust minus drag) is constant and that the rocket's propellant mass decreases linearly at a constant rate, differentiation of Equation (38) gives the state propagation model of β [30]:

$$\dot{\beta} = \beta^2 \tag{43}$$

Similarly, the propagation of the net thrust acceleration is also computed from the rocket thrust equation, including the definition of the velocity parameter:

$$\dot{\alpha} = \alpha\,\beta \tag{44}$$

Equation (44) shows the benefit for the dynamic model. The derivatives only depend on the state values themselves, so it is a closed system with only a limited number of unknowns included in the system of equations. For the propagation of the velocity, its derivative equals the sum of all the accelerations that affect the rocket's motion. The original model of the EKF developed by Airbus considers engine axial propulsion, gravitational 2-body acceleration, and non-inertial accelerations due to the Coriolis effect and the centripetal pull. Nevertheless, it has been studied that the influence of the Coriolis effect and the centripetal pull are negligible for the cases studied in this thesis. That is, the effect of the non-inertial forces is very small in comparison with the remaining forces (thrust and gravity) during the powered phase of the trajectory. Moreover, the loss of information contained in the observations made by the sensors and the initial uncertainty of the state are greater than the effect of Coriolis and centripetal forces. In other words, the effect of noise (measurement noise and process noise) is greater than the influence of the non-inertial forces. All things considered, the propagation of the velocity is expressed with the following equation

$$\dot{\boldsymbol{v}} = \boldsymbol{a_T} + \boldsymbol{a_G} = \alpha \frac{\boldsymbol{v}}{|\boldsymbol{v}|} - \mu \frac{\boldsymbol{r}}{|\boldsymbol{r}|^3} \tag{45}$$

where $\boldsymbol{a_T}$ and $\boldsymbol{a_G}$ are the components of acceleration due to thrust and gravity, respectively. This first term of this equation of motion describes a gravity turn.

Note from Equation (45) that this motion is highly nonlinear, as there are elements of the state that are present in a higher order as it can be seen in the norm of the position and velocity. Such nonlinearities make necessary to utilize nonlinear filters or approximations of the model in order to linearize it. This is the reason why an Extended Kalman Filter is used. As it was already discussed in Section 2.4, this is done in an EKF by computing the Jacobian of the state with respect to the derivative. For the case at hand, numerical propagation of the model given by (42) is feasible. Nevertheless, linearization is needed in order to propagate

the state error covariance used for the estimator in the update step of the filtering process. Therefore, the model is linearized around the true state, $\hat{S}$, for each epoch using the following Taylor series expansion [10]

$$\dot{S}(t) = f(S,t) = f(S,t) + \left(S(t) - \hat{S}(t)\right)\frac{\partial f}{\partial S}\Big|_{S=\hat{S}} + O_F\left(S(t) - \hat{S}(t)\right) \quad (46)$$

where $O_F\left(S(t) - \hat{S}(t)\right)$ represents the second and higher order terms of the expansion. The partial derivative term is the gradient $\frac{\partial v}{\partial u} \equiv \nabla_{\mathrm{u}} v(u)$, which for this case is the following 8x8 partial derivative matrix

$$A(S,t) = \nabla_S f(S,t) = \begin{array}{c} \begin{matrix} 1\cdots3 & 4\cdots6 & 7 & 8 \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \boldsymbol{a} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{b} & \boldsymbol{c} & \boldsymbol{d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & e & f \\ \mathbf{0} & \mathbf{0} & 0 & g \end{bmatrix} \end{array} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7 \\ 8 \end{matrix} \quad (47)$$

that represents the linearized dynamic matrix of the model, or state transition matrix, obtained from partial differentiation of Equation (42). Bold $\mathbf{0}$ are zero matrices of dimension (rows, columns) given by the indices in grey. The terms ranging from $\boldsymbol{a}$ to $\boldsymbol{g}$ are the following expressions

$$\boldsymbol{a} = \boldsymbol{I_3} \quad (48)$$

$$\boldsymbol{b} = \frac{\partial \boldsymbol{a_G}}{\partial \boldsymbol{r}} \quad (49)$$

$$\boldsymbol{c} = \frac{\partial \boldsymbol{a_T}}{\partial \boldsymbol{v}} \quad (50)$$

$$\boldsymbol{d} = \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \quad (51)$$

$$e = \beta \quad (52)$$

$$f = \alpha \quad (53)$$

$$g = 2\beta \quad (54)$$

whose dimensions are also given by the indices in grey. Note that $\boldsymbol{I_3}$ is an identity matrix of dimension 3x3. The correct implementation of this matrix (the partial derivatives) has been verified with a variational mechanism in which the propagation of the state numerically and with the state transition matrix above produced the same result after including a small perturbation.

Gravitational Model during Inter-Stage Coasting

The trajectories that are tracked for the analysis carried out in this thesis have more than one stage with possibility of inter-stage coasting. Furthermore, in a real application it might be possible that the observations are not available even during the powered phase of flight due to issues with the measurements. Therefore, a gravitational model is developed for these phases of flight. This model is based on the previous gravity turn model, without the influence of the acceleration due to thrust. As it was explained in Section 3.4, the filter automatically shifts to this model when no observations are being received and shifts back to the gravity turn model once they become available again. Expression (55) provides this dynamic model

$$\dot{S} = [\dot{\boldsymbol{r}} \quad \dot{\boldsymbol{v}} \quad 0 \quad 0]^T = f(S,t) \tag{55}$$

where the state vector has the same length, but the derivatives of the acceleration and the velocity parameter are zero, since the vehicle is in a free fall motion during coasting. Propagation of the velocity is based on Equation (45) as

$$\dot{\boldsymbol{v}} = \boldsymbol{a_G} = -\mu \frac{\boldsymbol{r}}{|\boldsymbol{r}|^3} \tag{56}$$

Finally, the linearized dynamic matrix is given by

$$A(S,t) = \nabla_S f(S,t) = \begin{array}{c} \begin{matrix} 1\cdots3 & 4\cdots6 & 7 & 8 \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \boldsymbol{a} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{b} & \mathbf{0} & \boldsymbol{d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7 \\ 8 \end{matrix} \tag{57}$$

which is the same matrix as from Equation (47), with the terms related to thrust acceleration equal to zero.

Addition of Process Noise

Process noise is added by just multiplying the propagated covariance matrix by a factor input by the user.

$$\bar{P}_Q = \left(1 + \frac{q}{100}\right)\bar{P} \quad ; \quad q \in [0, \infty) \tag{58}$$

Therefore, there is a certain percentage of covariance increase that allows the filter to not become too insensitive to new observations. In any case, this is a very simple method of tuning the covariance matrix with process noise. Previous work at Airbus implemented a process noise matrix that was added to the propagated covariance matrix and was dependent on the timestep size. However, it was found that this procedure of addition of process noise

does not perform very well when external trajectories such as the Vega rocket trajectories studied in the following chapters (see also Section 2.2) are tracked. More complex methods, such as adaptively changing the value of noise based on whether the results follow a certain probability distribution (see Section 5.4) are proposed throughout the following chapters. However, it is left for future work its implementation and analysis for the EKF.

4.2. The PKF

The PKF is tracking tool based on a linear, polynomial Kalman filter. The PKF-Tool shares the same Simulation Environment with the EKF. Therefore, the only differences between the EKF and PKF are found within the filtering step of Figure 8: Filtering step within the SE; specifically in the propagation and, in some cases (see Section 4.3), in the update of the Kalman filter. The purpose of this section is to perform a thorough analysis of the specific characteristics of the developed PKF, focusing on to the different types of dynamic models and thus, on the state transition matrices used for the propagation step of the filtering process. Unlike the EKF, a critical parameter for a correct functioning of the PKF is the process noise matrix. This matrix will be fully studied using as a starting point the general equations of a polynomial Kalman filter provided in Chapter 2.

In this section, first a general analysis of the PKF, its inputs and initialization methods will be carried out. Then, several types of dynamic models will be introduced, based on stochastic processes that are used for the propagation of the state during the filtering step. In general, these models are classified as Wiener and Markov process models. These models will be written into the form of state transition matrices. Each of these matrices will be used for a different phase of flight of the tracked rocket based on the thrust template characteristics. Finally, the process noise matrices for each of the types of PKF dynamic models will be provided.

4.2.1. Inputs and Outputs

PKF State Vector

All the different versions of the PKF, each with different types of propagation models, have the same elements in the state vector. In some of the case types, the elements have certain characteristics that differ from each other. The state vector of the PKF estimator is

$$S = [\boldsymbol{r}^T \quad \boldsymbol{v}^T \quad \boldsymbol{a}^T \quad \boldsymbol{j}^T] \tag{59}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are given by Equations (36) and (37), and $\boldsymbol{a}$ and $\boldsymbol{j}$ are the acceleration and jerk (first derivative of the acceleration) vectors given by

$$\boldsymbol{a} = [a_x \quad a_y \quad a_z]^T \tag{60}$$

$$\boldsymbol{j} = [j_x \quad j_y \quad j_z]^T \tag{61}$$

The error covariance associated to the estimated state is given by Equation (39).

PKF Initialization Method

Unlike the EKF, in which the initialization method was based on a polynomial fit of the first n measurements, the PKF is initialized by just utilizing the first available observation. Therefore, the position at epoch zero will be equal to the ECEF measurements obtained after triangularization of the satellite sensors' observations. The remaining elements of the state are set equal to zero for this initial epoch. Moreover, as it was argued in Section 2.4, one of the main advantages of polynomial Kalman filters are their robustness to relatively high initialization errors [9]. Thus, it is argued that this simple initialization method will be sufficient to achieve filter stability, as far as the remaining initial conditions are suitable for tracking – see Section 5.3. The initial elements of the error covariance matrix are the same chosen for the EKF. Note that, for the same reasons, the PKF is less insensitive than the EKF to initialization differences in the covariance matrix.

Polynomial Kalman Filter Dynamic Models

Throughout this entire thesis, different versions of PKF are used, based on the way the state vector is updated. These versions will be exhaustively addressed in Section 4.3. Nonetheless, any version of the PKF is implemented in such a way that several dynamic models can be used, independently on how the state is updated. In this section, the different types of dynamic models used by each of the versions of the PKF will be analyzed. Each of them will have a characteristic state transition matrix and a process noise matrix. In this way, two branches of dynamic models are studied based on the information that can be inferred from the motion of the flying vehicle and in the type of stochastic models that are utilized to develop the state transition matrices. Therefore, the following dynamic models have been implemented within the SE framework for the performance analysis and comparison carried out in this thesis [17] [8] [24] [4].

1) Wiener process models
    a. Wiener Process Acceleration Model
    b. Wiener Process Jerk Model
2) Markov process models
    a. Exponentially Autocorrelated Acceleration Model

4.2.2. State Propagation Models

The state propagation models used for the PKF are obtained from the Taylor series expansion given by Expression (16) truncated at the n^{th} order term [17]. The truncation is determined by the highest time derivative of position within the state vector. In this way, since the state vector contains the jerk of the vehicle (fourth time derivative of position), the truncation is done at $n - 1 = 3$, and the Taylor series expansion used to compute the state transition matrix Φ is the following

$$\Phi(t) = I + Ft + \frac{F^2 t^2}{2!} + \frac{F^3 t^3}{3!} \quad (62)$$

where I is an identity matrix and F is the system dynamics matrix obtained from the state vector given by Equation (15)

The system dynamics matrix is, thus, the relation between each of the elements of the state and their corresponding derivative. The elements of this matrix will have a different value depending on the PKF dynamic model used.

Time-Variant PKF

As it was introduced before, the PKF estimator algorithm will use a specific type of dynamic model depending on the phase of flight. For phases of flight in which the net acceleration (thrust minus drag) is dominated by thrust, a strong temporal correlation exists. In these cases, it is convenient to model the acceleration as a Markov process or to augment the state in order to include jerk, the derivative of the acceleration, as a Wiener process [4]. In other words, it is desirable to increase the stochasticity of the models (state augmentation, Markov models) when the temporal correlation of the vehicle's engine thrust is higher. In phases of flight in which the thrust force is relatively lower or zero, such as towards stage burnout (tail-off thrust) or during coasting, the net acceleration can be assumed to be (almost) constant and modelled by Wiener processes.

Therefore, the elements of the state vector given by Equation (59), and more specifically the acceleration and jerk terms, will have different values depending on the phase of flight of the launching vehicle. These values are provided subject to the type of stochastic process that is more suitable to be used. Moreover, it is possible to develop a time-variant method to choose the type of dynamic model based on the information provided by the thrust template of the external trajectories (Vega rocket, see Figure 7 in Section 3.1) analyzed in this thesis. By observing Figure 7, it can be seen that there is a period of about 20 seconds from maximum thrust until burnout of each of the stages in which the thrust rapidly decreases. In this period, it can be assumed that the time-correlation of the thrust force is lower and thus, a Wiener process acceleration model can be used (integral of a white noise process, see Chapter 2).

Furthermore, the infrared thermal signature of the vehicle obtained by the observing sensors can provide additional information regarding whether the vehicle is flying in full-thrust boost, tail-off thrust, or coasting phases. In this way, if no thermal signature is obtained at all, the vehicle will be coasting without powered propulsion (inter-stage coasting or after burnout, during the ballistic phase). If the thermal signature starts to decrease compared with the acquired values during the first available observations, it can be said that the vehicle has entered in the tail-off boost phase in which the thrust is relatively low. Otherwise, the vehicle will be flying under full thrust. Based on these ideas, each of the introduced stochastic models are used for the different phases of flight, as provided in Table 4. Therefore, it can be generalized that every flying vehicle under the thrust force will have a similar motion to the

one provided in Figure 7 so a time-variant PKF estimator which contemplates three different types of dynamic models is suitable for tracking.

Phase of Flight	Stochastic Model
Full-thrust boost	Wiener Process Jerk or Exponentially Autocorrelated Acceleration
Tail-off boost	Wiener Process Acceleration
Coasting	Wiener Process Acceleration (zero process noise)

Table 4: PKF dynamic models

From the information provided in Table 4, two main PKF types based on the dynamic models are obtained. On one hand, a PKF type based purely on Wiener process models. On the other hand, a PKF type that uses a Markov model in the full-thrust boost phase. The remaining chapters of this thesis mainly focus the analysis on the Wiener-based PKF due to several issues regarding tracking performance with Markov processes (see Section 5.1 and Annex B.2). In any case, each of the different types of dynamic models will be provided in the remaining parts of this section. Finally, as it was discussed in Section 2.4, process noise is added by adding a term of white Gaussian noise to the last element of the state vector. In this section, the process noise matrices that are obtained from the state transition matrices of each dynamic model will be obtained using Equation (9) as a starting point.

Wiener Process Acceleration Model and Noise

For this type of dynamic model, the elements of jerk of the state vector from Equation (61) have a value of zero. Therefore, the state vector becomes

$$S = [\boldsymbol{r}^T \quad \boldsymbol{v}^T \quad \boldsymbol{a}^T \quad \boldsymbol{0}^T] \tag{63}$$

and its corresponding system dynamics matrix is

$$F = \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \tag{64}$$

where $\boldsymbol{I}$ is a $3x3$ identity matrix and the remaining elements of the matrix are zero. Thus, using Equation (62) and this system dynamics matrix, the state transition matrix of the Wiener Process Acceleration Model will be

$$\Phi = \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\alpha} & \boldsymbol{\beta} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{\alpha} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \tag{65}$$

where the terms $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the following $3x3$ matrices

$$\boldsymbol{\alpha} = \begin{bmatrix} T_s & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix} \tag{66}$$

$$\boldsymbol{\beta} = \begin{bmatrix} T_s^2/2 & 0 & 0 \\ 0 & T_s^2/2 & 0 \\ 0 & 0 & T_s^2/2 \end{bmatrix} \tag{67}$$

being $t = T_s$ the timestep size (in seconds) between two consecutive epochs.

As it was stated in Section 2.4, process noise is modelled for the stochastic models analyzed in this thesis by considering the last element of the state vector as a random process [17]. In this case, the acceleration terms are assumed to be random, stochastic, processes as they are the last elements of the state on each Cartesian ECEF direction (jerk is set to zero). Therefore, a continuous process noise matrix Q is defined as follows

$$Q = \begin{array}{c} \begin{matrix} 1 \cdots 3 & 4 \cdots 6 & 7 \cdots 9 & 10 \cdots 12 \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{q_S} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{array} \begin{matrix} 1 \cdots 3 \\ 4 \cdots 6 \\ 7 \cdots 9 \\ 10 \cdots 12 \end{matrix} \tag{68}$$

being $\boldsymbol{q_S}$ the following $3x3$ matrix

$$\boldsymbol{q_S} = \begin{bmatrix} \phi_{a_x} & 0 & 0 \\ 0 & \phi_{a_y} & 0 \\ 0 & 0 & \phi_{a_z} \end{bmatrix} \tag{69}$$

where ϕ_{a_x}, ϕ_{a_y}, and ϕ_{a_z} are the power spectral densities (PSD) of the random process $S_{XX}(t)$ in which $S_{XX}(t) = \boldsymbol{a}$ is the acceleration vector for this case. The PSD is defined as the power density function of the random process in the frequency domain and is given by

$$S_{XX}(f) = \lim_{t_0 \to \infty} \frac{\mathrm{E}\left[\left|X_{t_0}(f)\right|^2\right]}{2t_0} \tag{70}$$

where $X_{t_0}(f)$ is the Fourier transform of the truncated version of the process as described by [33]. The value of the PSD for each of the directions is a user input parameter to be tuned with the procedure explained in Section 5.2.

From the continuous process noise matrix (68), the discrete process noise matrix Q_k used in the propagation step of the PKF is obtained with Equation (10) as a starting point (for one generic coordinate) and resulting in the following matrix[‡‡]

$$Q = \begin{array}{c} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{A} & \boldsymbol{D} & \boldsymbol{F} & \mathbf{0} \\ \boldsymbol{D} & \boldsymbol{B} & \boldsymbol{E} & \mathbf{0} \\ \boldsymbol{F} & \boldsymbol{E} & \boldsymbol{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{array} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \quad (71)$$

where each of the submatrices $\boldsymbol{A}$ to $\boldsymbol{F}$ are defined as follows

$$\boldsymbol{A} = T_s^5/20 * \boldsymbol{q_S} \quad (72)$$

$$\boldsymbol{B} = T_s^3/3 * \boldsymbol{q_S} \quad (73)$$

$$\boldsymbol{C} = T_s * \boldsymbol{q_S} \quad (74)$$

$$\boldsymbol{D} = T_s^4/8 * \boldsymbol{q_S} \quad (75)$$

$$\boldsymbol{E} = T_s^2/2 * \boldsymbol{q_S} \quad (76)$$

$$\boldsymbol{F} = T_s^3/6 * \boldsymbol{q_S} \quad (77)$$

and $\boldsymbol{q_S}$ is a diagonal matrix that contains the power spectral densities in each of the directions given by Expression (69).

Wiener Process Jerk Model and Noise

The Wiener Process Jerk Model definition is analogous to the Wiener Process Acceleration Model. Nevertheless, unlike the acceleration, in this model the jerk is selected as the random process (stochastic variable). In this case, since all the elements of the state given by Expression (59) are different from zero, the system dynamics matrix obtained is

$$F = \begin{array}{c} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{array} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \quad (78)$$

[‡‡] The computation is performed for one generic coordinate and a $3x3$ matrix is obtained. Nevertheless, it is a straightforward process to map this generic matrix into the full $12x12$ matrix that contains each of the elements of the state vector. In this way, the first element of the (diagonal) submatrices of Q correspond to the x-direction, the second element to the y-direction, and the third element to the z-direction. This procedure has been done for all the analyzed models. Note that the PSD within the Q matrix remain decoupled, so the tuning process becomes simpler as it can be done for each direction separately.

By the definition of a Wiener process [34], a stochastic variable is normally distributed with zero mean (hence zero value in the system dynamics matrix) and variance u, which is related to the power spectral density. Thus, the derivative of the last element of the state (jerk) is zero (rows 10 to 12 in F). In any case, the state transition matrix is computed from the Taylor series expansion given by Equation (62) resulting in

$$\Phi = \begin{matrix} \\ 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \begin{matrix} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\alpha} & \boldsymbol{\beta} & \boldsymbol{\gamma} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{\alpha} & \boldsymbol{\beta} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{\alpha} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \end{matrix} \tag{79}$$

being $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ Expressions (66) and (67) and $\boldsymbol{\gamma}$

$$\boldsymbol{\gamma} = \begin{bmatrix} T_s^3/6 & 0 & 0 \\ 0 & T_s^3/6 & 0 \\ 0 & 0 & T_s^3/6 \end{bmatrix} \tag{80}$$

Process noise is computed for the Wiener Process Jerk model similarly to the process noise of the Wiener Acceleration Process model. In this way, the discrete process noise matrix is given by the following symmetric matrix

$$Q = \begin{matrix} \\ 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \begin{matrix} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{A} & \boldsymbol{E} & \boldsymbol{H} & \boldsymbol{J} \\ \boldsymbol{E} & \boldsymbol{B} & \boldsymbol{F} & \boldsymbol{I} \\ \boldsymbol{H} & \boldsymbol{F} & \boldsymbol{C} & \boldsymbol{G} \\ \boldsymbol{J} & \boldsymbol{I} & \boldsymbol{G} & \boldsymbol{D} \end{bmatrix} \end{matrix} \tag{81}$$

where the matrix terms ranging from $\boldsymbol{A}$ to $\boldsymbol{J}$ are defined as

$$\boldsymbol{A} = T_s^7/252 * \boldsymbol{q_S} \tag{82}$$

$$\boldsymbol{B} = T_s^5/20 * \boldsymbol{q_S} \tag{83}$$

$$\boldsymbol{C} = T_s^3/3 * \boldsymbol{q_S} \tag{84}$$

$$\boldsymbol{D} = T_s * \boldsymbol{q_S} \tag{85}$$

$$\boldsymbol{E} = T_s^6/72 * \boldsymbol{q_S} \tag{86}$$

$$\boldsymbol{F} = T_s^4/8 * \boldsymbol{q_S} \tag{87}$$

$$\boldsymbol{G} = T_s^2/2 * \boldsymbol{q_S} \tag{88}$$

$$\boldsymbol{H} = T_s^5/30 * \boldsymbol{q_S} \quad (89)$$

$$\boldsymbol{I} = T_s^3/6 * \boldsymbol{q_S} \quad (90)$$

$$\boldsymbol{J} = T_s^4/24 * \boldsymbol{q_S} \quad (91)$$

and the matrix $\boldsymbol{q_S}$ is the equivalent to the PSD matrix given by Expression (69) but using jerk instead of acceleration (i.e. ϕ_{j_x}, ϕ_{j_y}, ϕ_{j_z}).

Exponentially Autocorrelated Acceleration Model and Noise

The Exponentially Autocorrelated Acceleration Model is based on a Markov process and it is analogous to the Singer model used for aircraft maneuvers. Unlike Wiener processes, Markov processes include correlation information of the last element of the state vector [4]. In this case, the highest element of the state vector is the acceleration, or equivalently, the jerk is set to zero, as given by Equation (63). The relation in acceleration (i.e. its derivative) given by the Markov process is the following

$$\dot{\boldsymbol{a}} = [\dot{a}_x \quad \dot{a}_y \quad \dot{a}_z]^T = \left[\frac{-1}{\tau_x}\, a_x \quad \frac{-1}{\tau_y}\, a_y \quad \frac{-1}{\tau_z}\, a_z\right]^T \quad \text{in continuous time} \quad (92)$$

$$\dot{\boldsymbol{a}}_{\boldsymbol{k}} = [\dot{a}_{x,k} \quad \dot{a}_{y,k} \quad \dot{a}_{z,k}]^T = \left[e^{-\frac{T_S}{\tau_x}}\, a_x \quad e^{-\frac{T_S}{\tau_y}}\, a_y \quad e^{-\frac{T_S}{\tau_z}}\, a_z\right]^T \quad \text{in discrete time} \quad (93)$$

where the terms τ_x, τ_y, and τ_z are the correlation time constants for each of the directions whose value is a user input parameter. The correlation time constant is thus a design parameter specific to this model, dependent on the booster characteristics and has to be tuned with respect to the possible launch scenarios [30]. The conversion from continuous to discrete domains is performed using the Laplace transform and a detailed description of this specific transformation can be found in [17]. From Equations (92) and (62), the following system dynamics matrix is obtained (in discrete time)

$$F = \begin{array}{c} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\alpha} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \end{array} \begin{matrix} 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \quad (94)$$

where $\boldsymbol{\alpha}$ is a diagonal matrix containing the discretization formula of the correlation time constants for each Cartesian direction

$$\boldsymbol{\alpha} = \begin{bmatrix} e^{-\frac{T_S}{\tau_x}} & 0 & 0 \\ 0 & e^{-\frac{T_S}{\tau_y}} & 0 \\ 0 & 0 & e^{-\frac{T_S}{\tau_z}} \end{bmatrix} \tag{95}$$

Using the above system dynamics matrix and the Taylor series expansion given by Expression (62), the following state transition matrix is obtained [34]

$$\Phi = \begin{matrix} \\ 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \begin{matrix} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\beta} & \boldsymbol{\delta} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \boldsymbol{\gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\alpha} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{matrix} \tag{96}$$

where the $3x3$ diagonal matrix terms ranging from $\boldsymbol{\beta}$ to $\boldsymbol{\delta}$ are the following expressions – note $\boldsymbol{\alpha}$ is given by Equation (95)

$$\boldsymbol{\beta} = T_S \boldsymbol{I} \tag{97}$$

$$\boldsymbol{\gamma} = \mathbf{T}(\boldsymbol{I} - \boldsymbol{\alpha}) \tag{98}$$

$$\boldsymbol{\delta} = \boldsymbol{\beta}\mathbf{T} - \mathbf{T^2} + \mathbf{T^2}\boldsymbol{\alpha} \tag{99}$$

being $\mathbf{T}$ the following $3x3$ diagonal matrix that contains the correlation time constants

$$\mathbf{T} = \begin{bmatrix} \tau_x & 0 & 0 \\ 0 & \tau_y & 0 \\ 0 & 0 & \tau_z \end{bmatrix} \tag{100}$$

Once the state transition matrix is obtained, process noise is computed in the same way used for the Wiener process models. The process noise matrix obtained for the analyzed Markov process is [34]

$$Q = 2\boldsymbol{W} \begin{matrix} \\ 1\cdots3 \\ 4\cdots6 \\ 7\cdots9 \\ 10\cdots12 \end{matrix} \begin{matrix} \begin{matrix} 1\cdots3 & 4\cdots6 & 7\cdots9 & 10\cdots12 \end{matrix} \\ \begin{bmatrix} \boldsymbol{A} & \boldsymbol{D} & \boldsymbol{F} & \mathbf{0} \\ \boldsymbol{D} & \boldsymbol{B} & \boldsymbol{E} & \mathbf{0} \\ \boldsymbol{F} & \boldsymbol{E} & \boldsymbol{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{matrix} \tag{101}$$

where the terms ranging from $\boldsymbol{A}$ to $\boldsymbol{F}$ are the same ones obtained for the Wiener Process Acceleration Model – Equations (72) to (77) and matrix $\boldsymbol{W}$ is given by

$$W = \begin{matrix} \begin{matrix} 1 \cdots 3 & 4 \cdots 6 & 7 \cdots 9 & 10 \cdots 12 \end{matrix} \\ \begin{bmatrix} \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{0} \\ \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{0} \\ \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{T^{-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{matrix} \begin{matrix} 1 \cdots 3 \\ 4 \cdots 6 \\ 7 \cdots 9 \\ 10 \cdots 12 \end{matrix} \qquad (102)$$

4.3. Measurement Models – PKF Versions

In the previous section, several different types of state transition matrices that are used in the propagation step within the Simulation Environment are proposed for both EKF and PKF. For the PKF, this resulted in the development of two versions: the first one based on Wiener processes and the second one (mainly) based on Markov processes. Each of these two versions uses three different types of dynamic models based on information obtained by the observing sensors about the thrust template of the tracked launching vehicle.

In this section, the focus is shifted to the update step within the SE. Three different types of measurement models will be proposed, which potentially can be applied to any of the two versions – Wiener- or Markov-based, of the PKF. Nevertheless, only one of the measurement models can be used for the EKF, as it will be explained below. Two of these measurement models use a Cartesian reference frame and the third proposed model uses a polar (or spherical) reference frame. There are several advantages and inconveniences of using one type of model or another, so a trade-off can be performed to analyze the strengths and weaknesses of each of the proposed measurement models. On one hand, the observations are obtained in the LHZ spherical frame of the sensors. On the other, the state vector of each of the filtering tools is in the Cartesian ECEF reference frame. Transformation from one frame to another increases the complexity of measurement noise modelling but allows the utilization of linear models [12]. Therefore, in the update step two approaches can be carried out:

1) Converting the propagated state into the spherical reference frame or finding a relation between the state and the observations within the measurement model contained in the measurement transition matrix H (see Chapter 2).
2) Converting the observations into the Cartesian ECEF frame so a linear relation between the state and the observations can be obtained. This implies to transform the measurement noise of the sensors from polar coordinates (i.e. azimuth and elevation) to Cartesian x, y, z coordinates.

In any case, a frame transformation of certain elements of the SE is needed, which can be a complex task.

The three proposed models are the following:

1) Cartesian Decoupled Measurement Model
2) Cartesian Coupled Measurement Model
3) Polar Measurement Model

The three measurement models originate three different versions of the PKF-Wiener based plus one version of the PKF-Markov based. The EKF can only use the polar measurement model. In this way, the different versions analyzed of the PKF-Tool are gathered in the following table, according to the types of measurement models and dynamic models.

PKF Version	Measurement Model	Dynamic Models
PKF Decoupled (PKFD)	Cartesian Decoupled	Wiener: boost, tail-off, coasting
PKF Coupled Cartesian (PKFCC)	Cartesian Coupled	Wiener: boost, tail-off, coasting
PKF Coupled Polar (PKFP)	Polar (always coupled)	Wiener: boost, tail-off, coasting
PKF with Markov models (PKFM)	Polar	Wiener: tail-off, coasting Markov: Boost

Table 5: PKF Versions. Measurement Models and Dynamic Models

Unlike the decoupled version, coupled PKF versions contain information about the correlation between the different elements of the state vector. For the polar measurement model, since the relation between state and observations is nonlinear, it will always be coupled. Finally, although the PKFM can be implemented for decoupled measurement models, its relatively higher complexity makes unnecessary to analyze this version for different cases besides the polar one, as it is explained in Section 5.1 and Annex B.2.

4.3.1. Cartesian Measurement Model

The purpose of this type of measurement model is to obtain a linear relationship between the state and the observations. To obtain this linearity, a frame transformation from spherical (polar) to Cartesian coordinates is necessary to be performed to the measurement noise matrix R. The triangulated measurements in ECEF are used as pseudomeasurements for this model. The measurement transition matrix H, thus, has the following linear relation for the coupled and decoupled measurement models

$$H = \underset{1 \cdots 3}{[\,\boldsymbol{I_{3x3}}} \quad \underset{4 \cdots 12}{\mathbf{0}}\,] \begin{matrix} \\ 1 \cdots 3 \end{matrix} \qquad (103)$$

For bot Cartesian measurement models, the measurement noise matrix R is a diagonal matrix of dimension $3x3$ with the cartesian measurement noises x, y, z in its diagonal.

$$R = \begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \qquad (104)$$

Cartesian Measurement Model - Decoupled

The simplest Cartesian measurement model is decoupled case, in which it is assumed that there is no correlation among the elements of the state. The advantage of this model is that allows to use three different filters, one on each Cartesian direction, so values of e.g. process noise in one direction do not affect to the other two. On the contrary, the loss of information due to neglecting correlation will cause that i.e. the orientation of the error ellipsoid at burnout will remain unknown.

The decoupled Cartesian measurement model is used for the PKFD version. The purpose of this model is to have the simplest possible filter update step by having a linear relation between the state vector and the measurements. Nevertheless, measurement noise σ_n is contained in the R matrix (11) in LHZ polar coordinates. In order to use the Cartesian measurement model, a frame transformation of the measurement noise is performed. The general idea of measurement noise transformation is given by [17]

$$\sigma_r = f(E[\Delta r^2])\sigma_n \qquad (105)$$

where σ_r is the measurement noise in cartesian ECEF and σ_n the measurement noise in the polar LHZ frame. The component r represents any generic cartesian coordinate. The derivation of the Cartesian measurement noise for the decoupled case is provided in Annex A. The following are main remarks of this transformation (this is also applicable for the coupled Cartesian measurement model explained below):

1) The triangulated ECEF positions at each epoch are necessary
2) The expectations $E[x]$ of the relation between polar coordinates and cartesian are taken, assuming that there is no correlation between the LOS measurements (azimuth and elevation) within each sensor and (logically) between sensors.

Specific to the decoupled measurement model, by definition,

3) There is no correlation between the elements of the state: $E(\Delta u \Delta v) = 0$, being u, v any pair of Cartesian coordinates x, y, z; thus, not taken into account for the above computation.

Note that [12] proposes a pseudolinear measurement model with a simpler computation of the elements of the measurement noise matrix R. Nevertheless, it was realized that the filter loss of consistency is relatively high (see Section 5.4) and thus this method was not selected.

Cartesian Measurement Model – Coupled

This measurement model is rather similar to the decoupled case, with the main difference that the correlation terms between the elements of the state (in position) are also taken into account. This issue results in a more complex frame transformation of measurement noise, as provided in Annex A. On the other hand, the linearity between state and measurements is still

maintained with the advantage that correlation information is available. This results into e.g. a known orientation of the error ellipsoid at burnout (see Section 5.6) and potentially a more precise burnout state due to the increase in information.

4.3.2. Polar Measurement Model

The polar measurement model is the option used by the EKF, since the nonlinearities present in the dynamic model of this filtering tool do not allow carrying out the noise transformation used with the PKF Cartesian versions. Furthermore, this model can also be implemented in a polynomial Kalman filter, so an additional version of the PKF is also developed with this measurement model, as shown in Table 5. The model is nonlinear, since it uses the LHZ spherical coordinates of the sensors and the state vector remains in Cartesian ECEF. Thus, the gradient of this nonlinear relation ($z = h(x,t)$ being z the spherical observations and x the state vector) is needed in the measurement transition matrix H. The gradient of the observations with respect to the position is

$$\nabla_r h = \frac{\partial h}{\partial \boldsymbol{r}} = \begin{bmatrix} \frac{\partial \theta_1}{\partial r_x} & \frac{\partial \theta_1}{\partial r_y} & \frac{\partial \theta_1}{\partial r_z} \\ \frac{\partial \varepsilon_1}{\partial r_x} & \frac{\partial \varepsilon_1}{\partial r_y} & \frac{\partial \varepsilon_1}{\partial r_z} \\ \frac{\partial \theta_2}{\partial r_x} & \frac{\partial \theta_2}{\partial r_y} & \frac{\partial \theta_2}{\partial r_z} \\ \frac{\partial \varepsilon_2}{\partial r_x} & \frac{\partial \varepsilon_2}{\partial r_y} & \frac{\partial \varepsilon_2}{\partial r_z} \end{bmatrix} \quad (106)$$

where θ_i and ε_i are the azimuth and elevation measurements for satellites $i = 1,2$. The partial derivatives of Equation (106) are computed as follows (for satellite i)

$$\begin{bmatrix} \frac{\partial \theta_i}{\partial r_x} & \frac{\partial \theta_i}{\partial r_y} & \frac{\partial \theta_i}{\partial r_z} \end{bmatrix} = \frac{1}{\Delta r_{x_i}^2 + \Delta r_{y_i}^2} [-\Delta r_{y_i} \quad \Delta r_{x_i} \quad 0] \quad (107)$$

$$\begin{bmatrix} \frac{\partial \varepsilon_i}{\partial r_x} & \frac{\partial \varepsilon_i}{\partial r_y} & \frac{\partial \varepsilon_i}{\partial r_z} \end{bmatrix} = \frac{1}{\Delta r_{x_i}^2 + \Delta r_{y_i}^2 + \Delta r_{z_i}^2} \begin{bmatrix} \frac{\Delta r_{z_i} \Delta r_{x_i}}{\Delta r_{x_i}^2 + \Delta r_{y_i}^2} & \frac{\Delta r_{z_i} \Delta r_{y_i}}{\Delta r_{x_i}^2 + \Delta r_{y_i}^2} & \Delta r_{x_i}^2 + \Delta r_{y_i}^2 \end{bmatrix} \quad (108)$$

In the above equations, the parameter Δr_i is the vector between the satellite i location and the observed target. These partial derivatives have been computed using the relation between the observations and the target coordinates provided in Section 2.1.

Since the EKF and PKF have state vectors with different lengths, matrix H size is different for EKF and PKF. Nevertheless, its elements are the same, obtained from the computation of the gradient of Expression (106). For the PKF case H has dimension $4x12$; for the EKF case, H has dimension $4x8$:

$$H = \begin{matrix} 1 \cdots 3 & 4 \cdots 12 \\ [\nabla_r h & \mathbf{0}] \end{matrix} \begin{matrix} \\ 1 \cdots 4 \end{matrix}$$ PKF polar measurement transition matrix (109)

$$H = \begin{matrix} 1 \cdots 3 & 4 \cdots 8 \\ [\nabla_r h & \mathbf{0}] \end{matrix} \begin{matrix} \\ 1 \cdots 4 \end{matrix}$$ EKF polar measurement transition matrix (110)

As a final remark, the measurement noise matrix R does not need to undergo any type of transformation, unlike for the cartesian measurement models. For this case, the R matrix has dimension $4x4$ with the measurement noise spherical values σ_{θ_i}, σ_{ε_i} in its diagonal (see also Sections 2.1 and 3.2).

$$R = \begin{bmatrix} \sigma_{\theta_1} & 0 & 0 & 0 \\ 0 & \sigma_{\varepsilon_1} & 0 & 0 \\ 0 & 0 & \sigma_{\theta_2} & 0 \\ 0 & 0 & 0 & \sigma_{\varepsilon_2} \end{bmatrix} \quad (111)$$

4.4. Conclusions

In this chapter, a review of the particular characteristics of the EKF-Tool and the PKF-Tool tracking estimators has been carried out. Moreover, several PKF versions based on the type of measurement model used have been provided. As a summary, the EKF-Tool is an extended Kalman filter tracking tool previously developed at Airbus that uses a gravity turn nonlinear dynamic model and a spherical measurement model. On the other hand, the proposed PKF-Tool is based on a polynomial Kalman filter that allows using measurement models in Cartesian or spherical coordinates. Furthermore, the simpler, linear, formulation of its dynamic model makes possible the development of more complex process noise models that potentially allows tracking in a broader number of launch scenarios and trajectories.

Regarding the PKF, two broad forms of dynamic models have been introduced based on Wiener or Markov processes. Furthermore, for each of these PKF forms, three different versions of PKF can potentially be used based on the measurement model type: Cartesian (decoupled and coupled) and spherical models. With this, four PKF versions have been fully developed, as seen in Table 5. In Chapter 5 and Annex B a thorough comparison of the PKF versions is given, although an initial comparison is provided as follows.

As a summary, the decoupled PKF is not able to provide information regarding the correlation between the different Cartesian directions (as shown for the error ellipsoids in Section 5.6) but allows using a different PKF filter for each direction. One of the main advantages that such a model could have is that the main direction of flight can be tracked using one type of dynamic model and the perpendicular directions can be modelled as perturbances using a different type of dynamic model, thus increasing the overall accuracy of the estimation. The coupled polar and Cartesian PKF have differences only in the measurement model. The polar measurement model is the most complex one because the observations are in the local horizontal frame (LHZ) and the state is in ECEF coordinates.

Since observations and state are both in ECEF coordinates, their relation in the measurement model is linear. Finally, the greatest advantage of the PKF Polar is that it allows to use asynchronous sensors (i.e. each sensor obtain measurements at different epochs), that is, more than two observing satellites can be used and they do not have to be necessarily time-synchronized [35]. Using more than two sensors provides higher observability. On the other hand, the triangulation carried out for the Cartesian PKF filters (see Section 3.3) provides only one x, y, z ECEF measurement, so observations can only be formed using pairs of satellites and they all must be synchronized in time (all observations obtained at the same epoch). It is left for further work an in-depth analysis of the characteristics of sensor synchronicity and its impact on ballistic target tracking using the analyzed filters in this thesis.

	PKF Decoupled	**PKF Coupled**	**PKF Polar**
Asynchronous Sensors	No	No	Yes
Correlation	No	Yes	Yes
Simplicity	High	High	Low

Table 6: PKF capabilities comparison

Chapter 5

Filter and Performance Analysis

The purpose of this chapter is to analyze the behavior of the EKF and PKF under different input conditions in order to assess their strengths and drawbacks. The different nature of the two types of filters introduced in the previous chapter, based on their suitability for tracking nonlinear and linear systems, hints that they will show a different performance depending on the filter's inputs and the specific conditions of the Simulation Environment. Furthermore, each of the introduced PKF versions could potentially display a behavior that would make more suitable their utilization to solve a specific type of tracking problem.

Within the scope of this thesis, the performance of a filter indicates its suitability for tracking a launching vehicle. Many different types of performance indicators could be used to assess filter performance, depending on the specific interest of the research. The most fundamental performance indicator is filter stability. A stable filter displays a steady state behavior once it has been updated with enough observations [36]. Any of the analyzed filters can be deemed stable even if towards engine burnout divergence is observed (since this is due to a higher mismatch between the dynamic model and the vehicle's true motion, and the effect of process noise [37], see Chapter 3), as far as there is a long enough period of time showing a steady behavior.

Once the studied filters' estimations achieve a stable behavior, the next performance indicator to analyze is their accuracy. The accuracy of a tracking filter indicates how good it can estimate the true trajectory of a specific launching vehicle. This means that the difference between the true trajectory and the filter's estimated trajectory at each epoch, or error, must be as small as possible to consider the filter to have high accuracy. Furthermore, the precision of the filter, the closeness of all the filter's estimations for a specific epoch within a Monte Carlo environment, must also be high to deem a filter to have high performance [38]. In addition, the filter shall also produce statistics that are in accordance with the reality (filter consistency). That is, the state error covariance matrix produced by the filter and the error obtained from a certain number of iterations within a Monte Carlo environment should be somehow closely related [39].

However, accuracy, precision, and consistency alone are insufficient to conclude the suitability of a filter in terms of performance for the tracking problem studied throughout this thesis. Back in Chapter 1, the drawbacks of the EKF previously developed at Airbus were introduced. The main issue of this EKF resides in its incapability of adapting to different launch scenarios without modifying its input parameters. In this way, a filter that produces very accurate estimates and realistic statistics for a given launch scenario could have a relatively low performance if it is necessary to modify its inputs every time that the (a priori

unknown) vehicle's true trajectory changes. Therefore, an additional performance indicator is introduced to account for the versatility of the filter. That is, the capability of the filter to automatically adapt to a new launching scenario without the need of modifying the filter's input parameters.

In addition, variation of filter inputs could also have an impact on filter performance. If more than one launch scenario is used to analyze filter versatility, it must be necessary to first assess the influence of the input parameters for one specific launch scenario. In this chapter, two trajectories, one internal and one external to Airbus (see Chapter 3), are used for evaluating filter performance when the input parameters are modified. Modifying filter inputs could, for example, achieve faster filter stability but at expenses of high accuracy or consistency.

In order to account for all the different parameters that affect filter performance, several performance indicators are introduced in this chapter with the goal of determining which filter is more suitable in general and what are the optimal input values that each filter should have. In this way, performance indicators for filter consistency, accuracy, and precision are presented and then applied to the outputs produced by the EKF and the PKF. First, the effect of the arguably most important filter input parameter for filter stability (and overall performance), process noise, will be studied. This study will consist on a proposed manual tuning of process noise based on the steady state value of the root mean square error (RMSE). Then, several input parameters used for the initialization of the Simulation Environment (SE) that will remain constant to all the subsequent produced tests for performance analysis will be provided, together with their influence on filter output. Once a nominal value of process noise that achieves filter stability has been found for each type of trajectory (in-house and external), and the general input parameters used for initialization of the SE have been set, the aforementioned performance indicators will be thoroughly analyzed in the remaining parts of this chapter to determine which filter (EKF or PKF) is more convenient for each type of application (in-house and external trajectories). Together with this performance comparison of both filters, the specific effects of process noise on filter performance will also be analyzed. The last performance parameter, filter versatility, will be examined in depth in Chapter 6.

5.1. Comment on PKF Filter Versions

Throughout this chapter, a comparison between the PKF and the EKF is carried out. However, there is no mention to which of the presented versions of the PKF (see Chapter 4) is used. For the analysis performed in the following sections, the results displayed for the PKF are obtained with just one version of the PKF, since it has been studied that all the PKF versions display similar results for the performance indicators utilized in this thesis (except the one that uses Markov process models, see below). Therefore, there is no need to compare the EKF with all the PKF versions and the PKFD version is chosen for comparison purposes. In Annex B.1, a detailed assessment between each of the versions (PKFD, PKFCC, and PKFCP) is provided based on the performance indicators that are used in this chapter. As a

summary, Figure 9 shows a comparison of the different PKF versions (Wiener process models) using the Mean Absolute Error (MAE, see Section 5.5).

Developing a robust PKF filter based on Markov process models is a difficult task. During the thesis work, it has been seen that introducing the correlation time constant as a filter input greatly increases the work necessary for tuning the filter's process noise (more on filter tuning in Section 5.2). For one specific trajectory, it has been found that it is possible to find an optimal value of process noise using Markov models for one given value of the correlation time constant. However, unlike when Wiener process models are used, the value of either process noise or the correlation time constant must be modified if a different type of trajectory is to be tracked, which defeats the purpose of finding a versatile filter that allows tracking any type of trajectory without modifying its input parameters. This analysis is supported with the figures presented in Annex B.2.

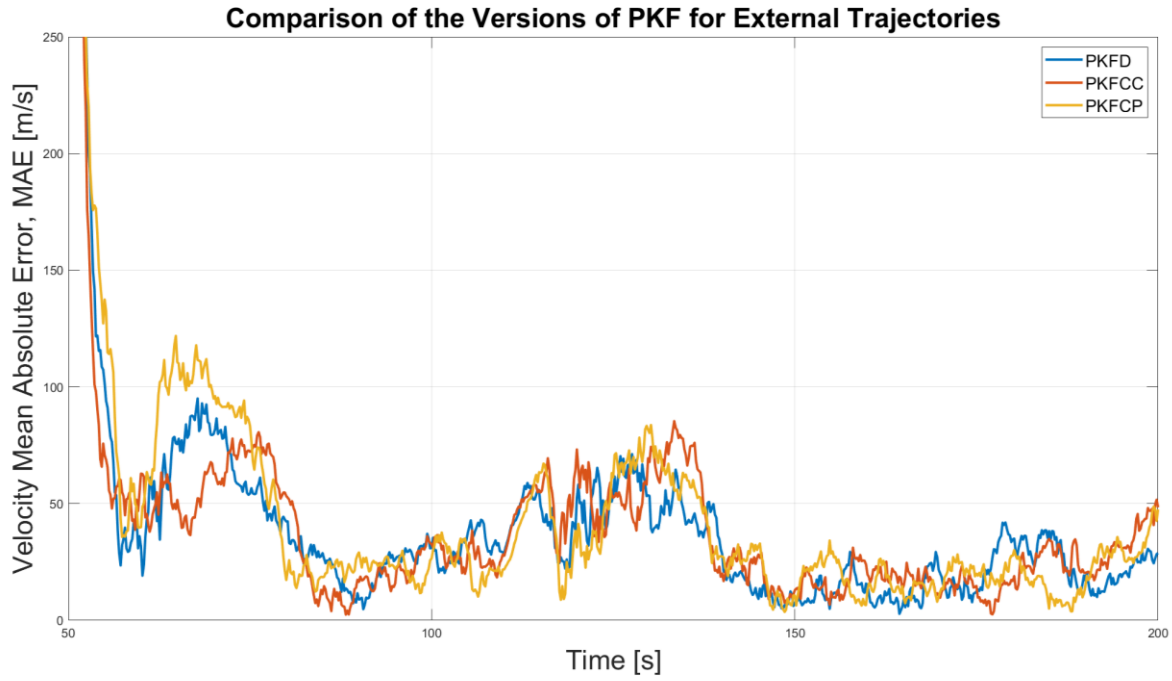

Figure 9: Comparison of PKF versions. The mean absolute error (computed for 100 MC runs) between the real trajectory and the estimated one by each of the PKF filters have been computed. It can be seen from this figure that all the PKF filter show relatively the same error.

5.2. Filter Tuning of Process Noise

In Kalman filtering, process noise is added to avoid filter output divergence, since the filter tends to trust more the dynamic models as more observation are added and in order to account for the differences between the filter's model and the reality (see Chapter 2) [37]. On the other hand, higher process noise values increase filter steady state error. Therefore, it is necessary to find a proper input value of process noise for the analyzed filters (the input process noise is the power spectral density for the PKF and the percentage of covariance increase for the EKF, see Chapter 4), so a minimum value of steady state error is obtained, and filter divergence is avoided. This means that filter stability can be reached for this value

of process noise. In terms of performance, the way process noise is modeled is also important as it can affect parameters such as filter credibility (or consistency), precision, and accuracy. The effect of process noise on filter performance is studied in the last sections of this chapter.

Therefore, this section just focuses on analyzing the effect of process noise on filter output. The main goal will be to introduce a method to obtain an optimal value of process noise that allows achieving filter stability and reduces steady state error. In this way, it is assumed that an optimal value of process noise can be obtained for one given launch vehicle's trajectory. A method to find this optimal value will be proposed using the Root Mean Square Error (RMSE). Furthermore, it is assumed that the proposed method to find an optimal value of the EKF process noise can also be applied for the PKF and the results of this method for PKF will also be provided. This section will not, however, address a thorough comparison of the EKF and PKF in terms of performance for different values of process noise. This study will be provided in the remaining sections of this chapter. Nonetheless, a comparison of different process noises applied for tracking in-house and external trajectories will be provided together with the behavior of each of the filters regarding tracking in-house and external trajectories.

5.2.1. RMSE Computation

The main tool used to compare filter output is the Root Mean Square Error (RMSE) [40]. This tool provides an absolute value of the error between the truth and the filter estimation, for a certain number of Monte Carlo runs. In this section, the RMSE is used to analyze filter stability, although the RMSE is also useful to determine the accuracy of estimators. In the terms defined for this thesis, filter stability is a qualitative value that explains whether steady state behavior is observed in the filter's output and divergence is avoided. Furthermore, once filter stability has been achieved, the RMSE can also provide a steady state error value, that is, the mean difference between n Monte Carlo estimations of the filter and the truth, squared. The square root is taken so the RMSE has the same units as the parameter that is being evaluated (position or velocity). In the analysis carried out in this section, the focus will be set on tuning process noise by analyzing the filters' RMSE values for achieving filter stability. The reader is referred to Section 5.5 for an analysis on the value of the steady state error (filter accuracy). The RMSE (in position or velocity magnitude) is computed using the following equations

$$RMSE_S = \sqrt{RMSE_{S_x}^2 + RMSE_{S_y}^2 + RMSE_{S_z}^2} \quad (112)$$

$$RMSE_{S_i} = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{MC} \left(|\hat{S}_i| - |S_i| \right)^2} \quad (113)$$

where N_{MC} is the total number of MC runs and $\hat{S}_i$ and S_i the filter's truth and estimated state (position or velocity) components ($i = x, y, z$), respectively.

5.2.2. Process Noise Optimal Value

Finding the optimal value of process noise is a rather complex task. There are many possible ways to find this optimum, which range from adaptively tuning the process noise based on the covariance matrix produced by the estimator, for example, to carrying out an offline optimization process, as it was done previously at ADS for this project [41] [42]. Moreover, for the work presented in this document, it is assumed that the optimal value of process noise is the one that achieves filter stability, avoids high divergence, and provides the smallest possible steady state error. A trade-off between divergent behavior and steady state error must be sought, as each of these parameters is inversely proportional to the other. On one hand, a low value of process noise provides a smaller steady state error. On the other hand, if the value of process noise is too small, the risk of divergent behavior of the filter's output increases.

Furthermore, it might be well possible that an optimal value of process noise used to track one given trajectory is not the same one used for a different trajectory. This concern will be also explored in this section by analyzing an in-house (minimum energy gravity turn – nominal GT –) and an external (Vega E) trajectory. In truth, finding a suboptimal value of process noise that works as good as possible for any trajectory seems a more suitable approach. Many different approaches can be used to find an optimum. In previous work carried out at Airbus, a Genetic Algorithm (GA) was envisaged for the EKF which made possible to obtain an optimum for one specific trajectory. The issue here is that this optimal value works well for tracking one trajectory (in terms of filter output) but when the same value of process noise is applied to a different one (e.g. for tracking a Vega rocket), high filter divergence occurs, and the optimization algorithm must be run again for the new trajectory. This GA has also the drawback of not being able to be implemented for real-time tracking. It can only be used a-posteriori, once the trajectory type is known. A solution would be to carry out offline optimization of all types of trajectories that are known and save them as a filter database bank, as it has largely been studied in the literature [3]. In any case, it is not possible to tune process noise on-line and the EKF becomes useless. This is the main reason why a suboptimal value of process noise becomes very convenient to be implemented (at least for the EKF, see more information regarding EKF downsides for complying with the stakeholder's needs in Chapter 1). Regarding the PKF, one of its most interesting characteristics is its simplicity in terms of modeling the dynamics of the vehicle's motion but its well-defined method of implementing process noise. These characteristics of the PKF lead to the following hypothesis: there exists a global optimal value of process noise that make possible to track any kind of trajectory.

Additionally, using a GA for process noise tuning seems to overcomplicate an already complex problem. Therefore, a simpler, yet almost equally effective method of process noise tuning is presented in this section. A manual tuning method is presented here, in which the

process noise is modified depending on the output obtained by the RMSE for a given trajectory. This manual tuning method is simpler but allows the user to realize the effect of an increasing/decreasing value of process noise. Furthermore, once this method has been used for one trajectory that is known (e.g. in-house rocket trajectory), one can think that it can work well for any other trajectory. If this is not the case, the same manual tuning can be performed for a different type of trajectory (e.g. Vega rocket). Then, a trade-off is performed between the two and a suboptimal value of process noise is selected by visual inspection. This is the procedure that it is used for tuning the process noise of both the EKF and PKF.

In-House Trajectories

First, an optimal value of process noise for the EKF is sought. This is done by checking the RMSE value for different process noises using the in-house nominal GT trajectory. The optimum is selected by checking one by one the RMSE produced by a certain value of process noise. This is done as follows: the RMSE for each epoch is obtained. If the RMSE reaches steady state, it is possible that the selected value of process noise is too high and a better (smaller) value of process noise is chosen. Then, the RMSE is again computed. If for the new value of process noise divergence is seen or the steady state behavior is not observed, it means that this value of process noise is too small and a higher one must be selected. This process is repeated iteratively until a minimum value that achieves steady state filter output is found. Figure 10 shows the above description of manual tuning.

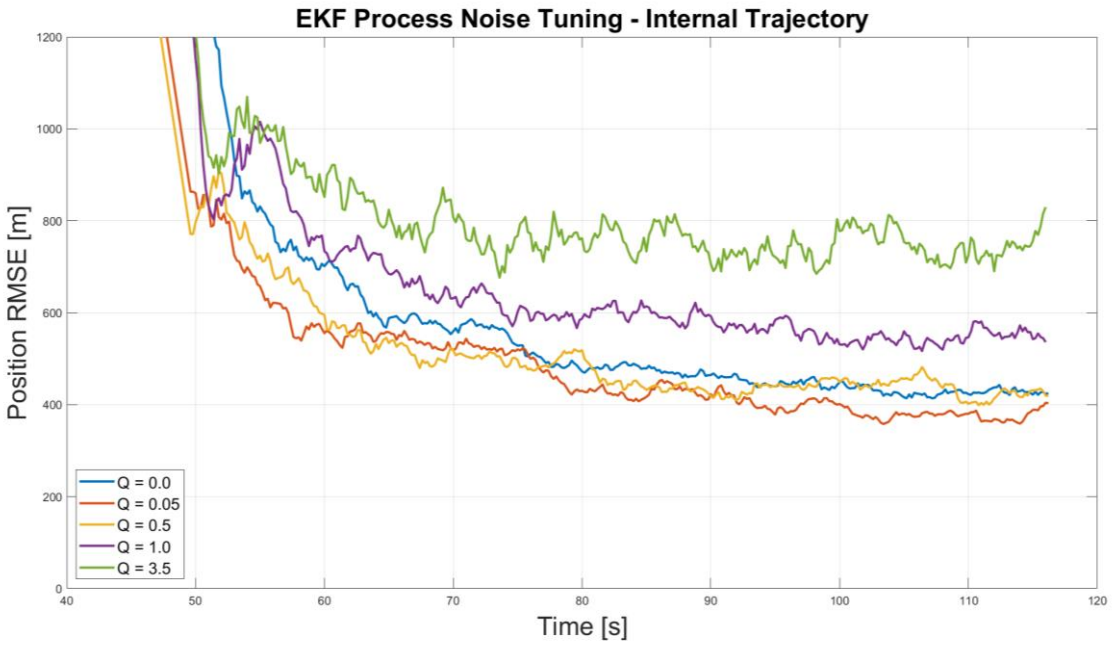

Figure 10: EKF process noise tuning for in-house trajectories. The manual tuning process is relatively simple: process noise is increased/decreased until finding the smallest value in which steady state convergence is achieved

Figure 10 displays the EKF RMSE-position for the 1st stage of the in-house nominal GT trajectory for different values of process noise q that range from $[0, 3.5]$. As it was explained in Chapter 4, process noise is added to the EKF by just multiplying the propagated covariance matrix by Q, which is calculated with Equation (58) using this constant q value. For a

$q = 0.0$, it can be seen that although divergence is not observed (the filter's dynamic model and true motion model are essentially the same, see Chapter 1 and Chapter 4), steady state behavior is not achieved. A higher value is then chosen: $q = 3.5$. For this value, a steady state behavior is observed but it could be possible that a smaller value of steady state error can be obtained. Following this idea, a final optimal value of $q = 0.05$ is selected. Another interesting feature can be seen from Figure 10: higher values of process noise reduce the settling time of the RMSE, e.g. the time the filter output takes to reach steady state behavior. On the other hand, a higher value of process noise increases filter steady state error, and that is the reason why $q = 0.05$ is preferred over $q = 0.5$, even if both of them show very similar RMSE trends.

Regarding the PKF, it was argued that due to its simplicity an optimal value of process noise can be obtained. The reader is referred to Chapter 4 for an in-depth explanation of process noise modeling for the PKF. Figure 11 shows the tuning process of process noise for in-house trajectories using the PKF. The procedure followed is the same as for the EKF. It can be clearly seen that for lower values of process noise, filter divergence exists. On the other hand, steady state behavior is obtained when process noise is high, achieving an optimal value of process noise equal to $q = 5 * 10^{-4}$, since higher process noises increase the steady state error excessively. Note that for simplicity's sake in the tuning process, the same value of process noise is used for the x, y, and z terms. As the purpose of the thesis is to compare both the EKF and PKF, finding a more accurate optimal PKF process noise is not studied in depth. Furthermore, spending too much time in obtaining an accurate optimum is unnecessary, as it will be seen that the filter output will change (at least slightly) depending on the tracked trajectory and no absolute global optimum exists neither for EKF nor PKF. In terms of settling time, the same stated for the EKF holds true for the PKF.

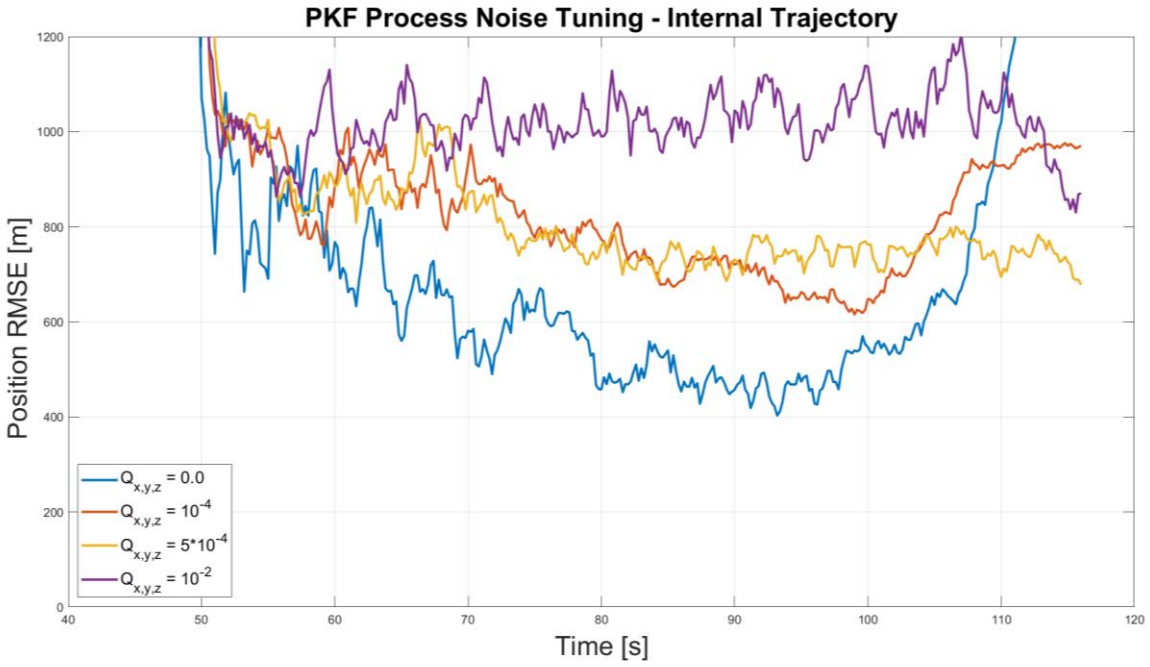

Figure 11: PKF process noise tuning for in-house trajectories. Similarly as it was done with the EKF, the manual method of process noise tuning for the PKF is performed until steady state convergence is achieved

External Trajectories

All things considered, an approximated optimal value of process noise has been found for both the EKF and PKF using a manual tuning method. These optimums have been obtained using one specific in-house trajectory (more on different types of in-house/external trajectories in Chapter 6). Therefore, it is necessary to test if the filters also work with these optimal values of process noise when an external trajectory is to be tracked or, on the contrary, a new value of process noise must be sought. If this is the case, a trade-off has to be carried out between the optimum found for in-house and external trajectories in order to select a suboptimal value of process noise. Figure 12 and Figure 13 show the manual tuning procedure for external trajectories. The followed procedure is the same as the one carried out for the in-house trajectories. The idea is as follows: first, the obtained optimal value of process noise is used in the filter and tracking is carried out. Then, if divergence is obtained a higher value of process noise is chosen. On the contrary, if RMSE steady state behavior is observed with the optimum obtained for in-house trajectories, a smaller value of process noise is used to verify if this optimum is global or a smaller value of process noise can be used for tracking external trajectories.

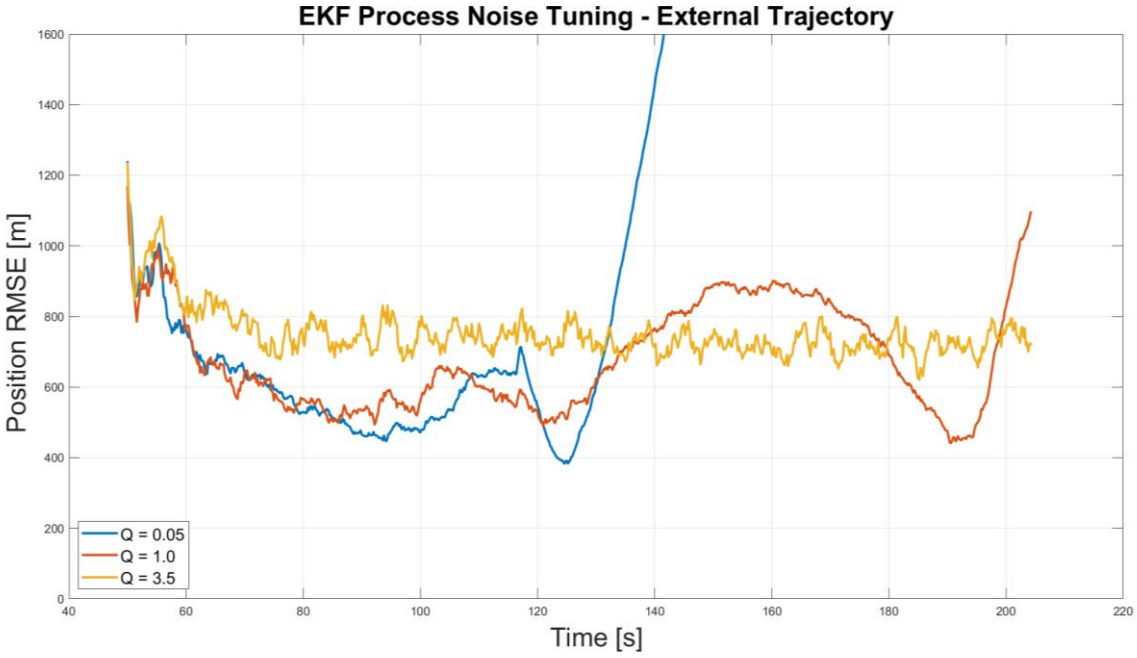

Figure 12: EKF process noise tuning for external trajectories. Unlike with in-house trajectories in which convergence was achieved for any value of process noise, for external trajectories process noise must be increased until steady state convergence is obtained. This value of process noise is higher than with in-house trajectories

Figure 12 shows that the filter loses stability when tracking external trajectories with the optimal value of process noise found for in-house trajectories. It can be seen that divergence occurs after the staging event has taken place (see Chapter 3). In order to avoid divergence, process noise must be increased. For a $q = 1.0$, steady state behavior is neither obtained and a final value of $q = 3.5$ is chosen as it is the smallest possible value of process noise that avoids filter divergence (plus a margin) using external trajectories. Therefore, it can be

clearly seen that a global optimum does not exist for the EKF and a trade-off must be used. Since $q = 3.5$ is limiting for external trajectories (a smaller value causes divergence), this value of process noise will be chosen as the suboptimum process noise for the EKF. This relatively high value of process noise will have an impact on filter performance, as it will be seen in the following sections of this chapter.

On the other hand, Figure 13 shows the effect of three different values of process noise for tracking an external trajectory using the PKF. Starting with the optimum of $q = 5 * 10^{-4}$ that was obtained for inhouse trajectories, steady state behavior is obtained. Nevertheless, if this value of process noise is reduced, divergence starts to be observed. Therefore, it can be concluded that the optimal value of process noise found for in-house trajectories is effective for external trajectories as well, so a global optimum is obtained. That means that every type of trajectory can be tracked using the value of process noise found with in-house trajectories, and it can be expected that the filter output will be stable for all cases.

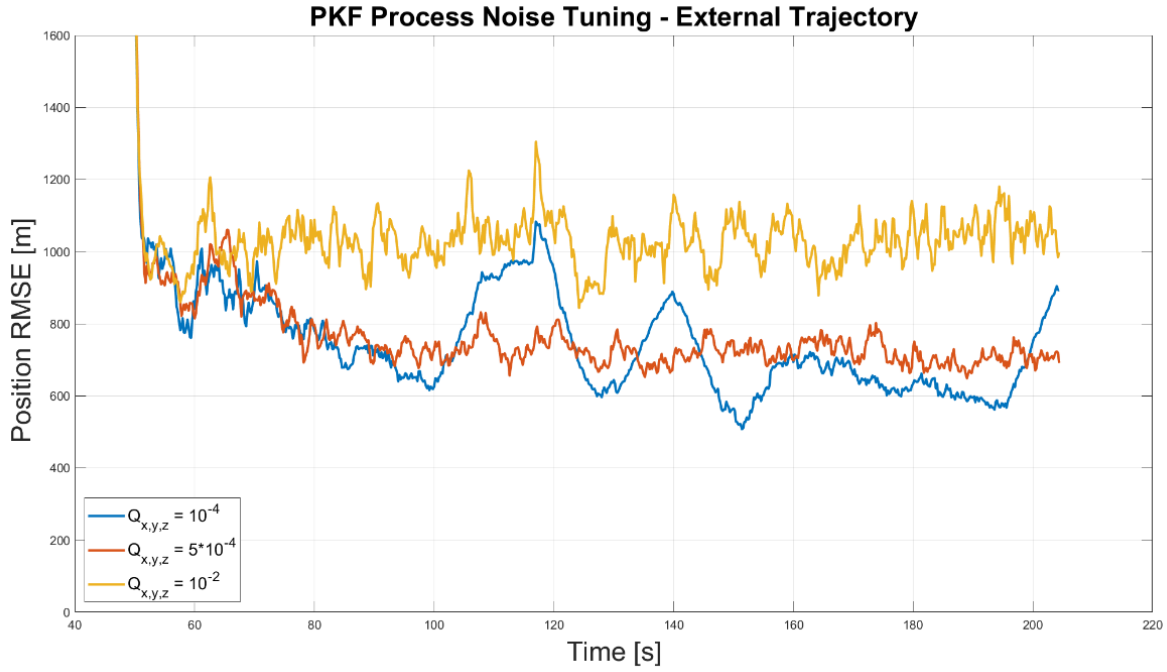

Figure 13: PKF process noise tuning for external trajectories. The manual tuning process using the RMSE is repeated. If the value of process noise is too high, steady state will be achieved but at cost of a lower accuracy. Therefore, process noise must be reduced so steady state is still achieved with a higher accuracy

It is important to notice that whereas the EKF needs a different optimal value of process noise depending on the tracked trajectory, the PKF succeeds tracking in-house and external trajectories with just one optimal value of process noise. That is, the PKF process noise can be tuned for one known trajectory and any other trajectory can potentially be tracked with this value of process noise. The advantages of PKF over EKF are clear in this matter: potentially, the PKF allows real time tracking of any possible trajectory ensuring filter stability once its process noise has been tuned using, for example, a known in-house trajectory. On the other hand, the EKF needs to be tuned for every trajectory so real-time tracking is not possible, which defeats the purpose of online tracking. Nevertheless, if the process noise for EKF is high enough, any possible trajectory can potentially be tracked

although this has consequences on filter performance, not to mention that the amount of process noise needed for the EKF is a-priori unknown. Despite if a (sub)optimal value of process noise has been found for each filter, in the subsequent performance analysis three values of process noise will be used: the (sub)optimal one, a higher one, and a smaller one than this optimum.

5.3. Effect of Environmental Conditions on Filter Output

The output produced by each of the tracking filters analyzed in this thesis is affected by many different factors. Before studying the performance of the EKF and PKF, it is necessary to select and define a group of input parameters that will remain fixed throughout all the tests performed in this chapter and the following. Furthermore, it also crucial to characterize all those parameters whose variation will affect filter output and, thus, will play a role in the performance analysis carried out in the subsequent sections of this chapter. The goal of this section is to provide the reader with the effect on filter output of a series of input parameters common to the Simulation Environment. These parameters are not, thus, intrinsic to each of the analyzed filters but general to the whole SE. Therefore, once their effect on filter output has been investigated, a fixed value that will be used for the subsequent performance analyses will be selected for each of them.

The input parameters of the Simulation Environment can be classified in three groups. More information about the characteristics of the input parameters can be found in Chapter 3. The first group includes all those parameters used for initialization of the SE that are common to both filters, such as satellite location and launch vehicle true trajectory. With the exception of vehicle's trajectory (which one sample external, Vega E, and internal, nominal GT, trajectories have been selected for the current chapter, see also Chapter 2), all the SE initialization parameters will remain fixed during the performance analysis (satellite location and vehicle trajectories are modified in Chapter 6 for versatility performance analysis).

The second group comprises the input parameters specific to each filter. Some of them have been studied in previous work at Airbus and their impact on filter performance will not be further addressed in this thesis. Furthermore, it has been proved from previous work at ADS that their effect on performance is considerably much smaller than the one produced by filter process noise, which is thoroughly studied in Section 5.2. Besides filter process noise, these parameters are the EKF acceleration factor, EKF initial exhaust velocity, EKF integrator tolerances, and number of iterations of the IEKF (see Chapter 1 and Chapter 2); and the initial covariance matrix for both EKF and PKF. These parameters will thus remain unchanged during the whole analysis and their value is shown in the following table.

Parameter [Units]	**Value**
Initial State Error Covariance Matrix (x, y, z position), $P_{0,p}$ [km]	2.0
Initial State Error Covariance Matrix (x, y, z velocity), $P_{0,v}$ [km/s]	0.2
Initial State Error Covariance Matrix (x, y, z acceleration), $P_{0,a}$ [km/s^2]	0.02

Initial State Error Covariance Matrix (EKF velocity parameter), P_{0,a_1} [km/s]	0.01
EKF acceleration factor, f_a [-]	1.8
EKF initial exhaust velocity, $a_{1,init}$ [m/s]	2500
(I)EKF iterations [-]	1
ODE absolute local error tolerance [-]	10^{-16}
ODE relative local error tolerance [-]	10^{-8}

Table 7: Fixed Input Parameters of the Filter

The values chosen for the parameter specific to the EKF have been proposed by ADS after carrying out an optimization process. In the case of the state error covariance matrix P_0, their initial values have been chosen according to the following idea. The 1-σ standard deviation error obtained from the difference between the triangulated and the true position for 100 MC runs is in most cases around 2 km. Therefore, the position elements of the initial covariance matrix are chosen to have this value. In terms of velocity and acceleration, a value different from zero (in order to avoid integrator divergence) that is reasonably close to what it should be expected is chosen. As long as the initial state covariance is within a reasonable value close (and greater than) the real initial error, the EKF will be able to obtain estimates: propagation will be carried out without integrator divergence. Further, in Chapter 4 it was argued that the PKF is less sensitive than the EKF to initialization errors. Therefore, using the P_0 values of Table 7 obtained with the aforementioned logic will ensure filter convergence (at least for the nominal case with (sub)optimal process noise, see Section 5.2).

Finally, the third group of input parameters includes all the SE input parameters that directly affect to filter output, with independency of the filter type used. In this way, parameters such as the observation rate or observation timespan will directly affect to the filters' output since any estimator generally performs better when more information is available [1]. Thus, the goal of this section is to analyze the effect of varying SE input parameters of this third group and to choose nominal values for the remaining of the performance analysis. The values of these input parameters will be modified within a determined range and their effect on filter output will be displayed by computing the velocity RMSE for $MC = 100$ MC runs using Equations (112) and (113).

The RMSE in velocity is preferred for displaying purposes over position as the effect on filter output is better seen in velocity due to their higher order of magnitude. Nonetheless, the same behavior is also obtained for position. Moreover, the RMSE computation is chosen for analysis purposes since it is a very useful tool to realize filter's output characteristics such as settling time, steady state error, and convergence/divergence. The SE input parameters analyzed in this section are gathered in the following table together with the range for which this analysis is carried out.

Parameter [Units]	**Range**
Number of measurements for EKF initialization [-]	$3, 30$
Cloud Break altitude [km]	$4, 11$

Timestep size (observation frequency) [s]	[0.2, 1.0]

Table 8: Analyzed SE Input Parameters

The number of measurements for EKF initialization is specific to the EKF and the two extreme values (minimum and maximum possible number of measurements) have been selected to see their effect on filter output. The time step size is increased from 0.2 to 1.0 seconds in increments of 0.2 seconds for both filters.

Analysis of SE Input Parameters

In the analysis carried out in this section, an optimal value of process noise for the EKF and PKF is used as a fixed input. This value is derived from a sensitivity analysis performed using the RMSE. An exhaustive description of this analysis together with the found optimal values of process noise is presented in Section 5.2 and no further comment shall be made here. Furthermore, the nominal GT trajectory (used for the analyses for EKF initial measurements and timestep size) and Vega E (used for the analysis for cloud break altitude) trajectory have been used. The description of these trajectories is provided in Chapter 2. Finally, the fixed input values provided in Table 7 are also used together with the SE general input parameters whose value is given in Table 1.

First, the number of measurements for EKF initialization is analyzed. The RMSE-velocity results are shown in Figure 14. One measurement is defined as the triangulated x, y, z values obtained from the observations at one epoch. It can clearly be seen that the filter output is greatly improved when the number of initial measurements is higher. For the low number of initial measurements case, the settling time is higher and the EKF filter struggles to achieve steady state behavior, incurring into divergence towards burnout.

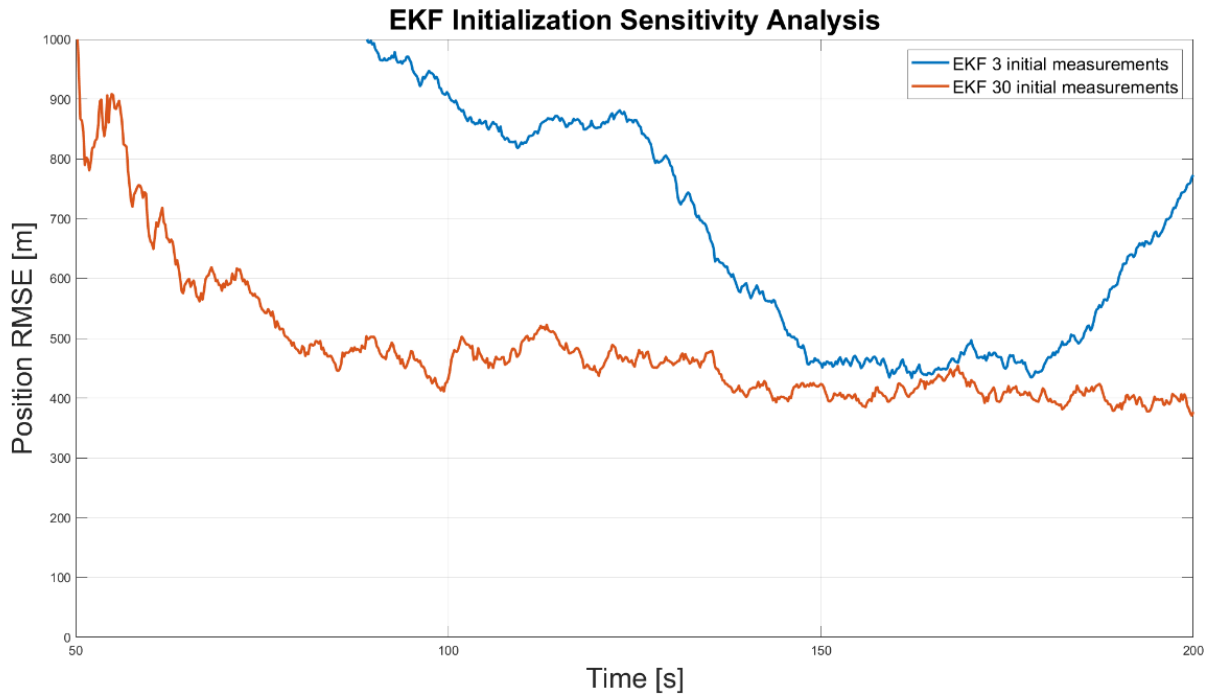

Figure 14: Sensitivity analysis for EKF number of initial measurements. If the number of initial measurements used for EKF initialization is too low, filter convergence will not be achieved. Since the length of a timestep (epoch) is relatively small (less or equal than a second), a higher number of initial measurements can be chosen without affecting to the main goal, which is to find the best estimate at burnout (usually several tens of seconds after the first observation is available)

It can be thus concluded that the number of initial measurements have a great impact on filter output. The maximum number of measurements allowed by the EKF filter initialization algorithm (polynomial fit) is 30. For this value, a steady state behavior is achieved. Therefore, this is the number of initial measurements used for the performance tests in the remaining parts of this chapter and the following.

As it was explained in Chapter 2, the IR LOS passive sensors are only capable of observing the launching vehicle plume after it has surpassed the cloud tops. Therefore, the higher the clouds top is, the less number of observations that will be available. As the estimators perform better when more observations are available, it is therefore hypothesized that a higher clouds top altitude will hinder filter output. Table 9 shows the RMSE at burnout in position and velocity for both the EKF and PKF from the analysis carried out for Cloud Break altitude.

	EKF		**PKF**	
CB Altitude	*RMSE Position*	*RMSE Velocity*	*RMSE Position*	*RMSE Velocity*
4 km	721.2839 m	165.4644 m/s	629.1236 m	140.8256 m/s
11 km	739.6344 m	158.2152 m/s	675.2148 m	140.6870 m/s

Table 9: Sensitivity Analysis for Cloud Break Altitude

From Table 9, it can be seen that the difference in RMSE is minimal between the high and low CB altitude cases for both of the filters. Therefore, the impact of cloud break altitude on

observability is low (if the thrust phase duration is long enough after the cloud break altitude) and the effect on filter output can be assumed negligible. In any case, the worst-case scenario of a cloud break altitude of 11 km is chosen for all the performance tests that are carried out in the following sections.

The last SE input parameter analyzed in this section is the timestep size. This parameter is related to the observation frequency and provides how often new measurements are received in the filter. The smaller the timestep size is, the more observations that will be available. Therefore, it is hypothesized that a small timestep size will provide better filter results (even though if this causes the addition of more process noise for the EKF, this effect is smaller in comparison with having more observations available). Figure 15 shows the EKF and PKF RMSE in velocity for different timestep sizes.

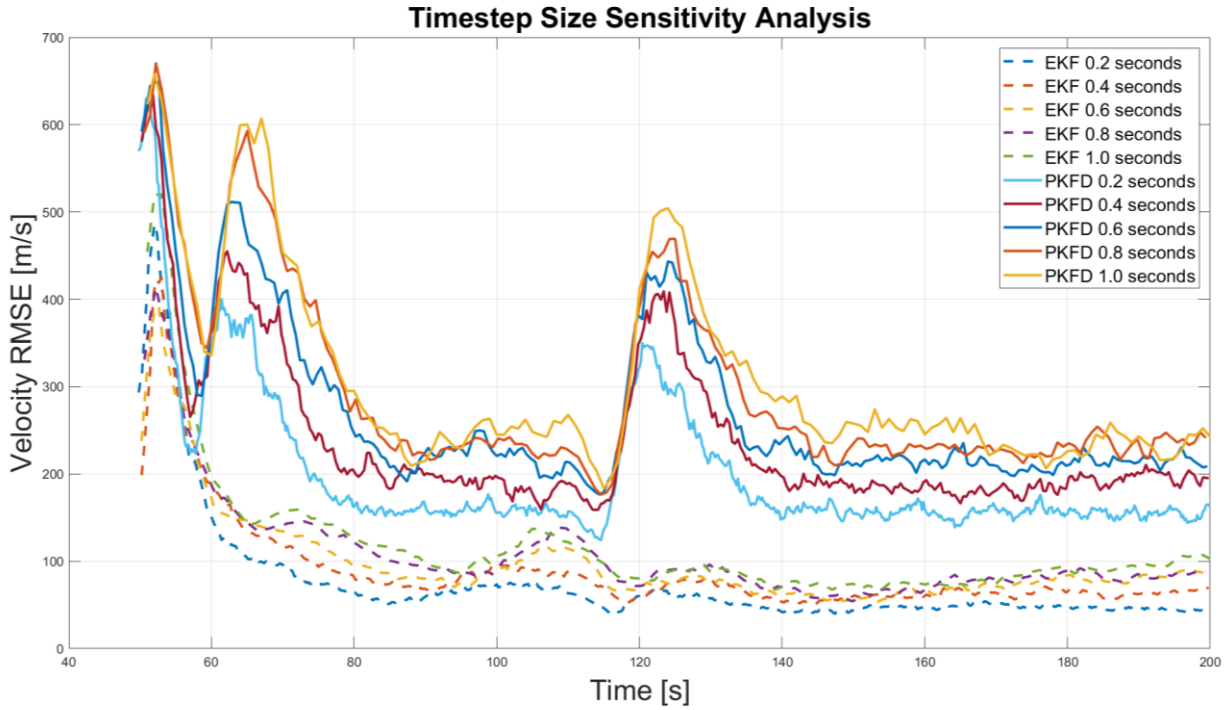

Figure 15: Sensitivity analysis for timestep size. For both EKF and PKF a smaller timestep size means a faster convergence rate and a smaller error at steady state.

Figure 15 shows that the timestep size affects mainly to two parameters of the RMSE: settling time and steady state error. As expected, the best results are obtained for smaller timestep sizes. Moreover, it can also be seen that the EKF is more affected than the PKF by the variation of the timestep size. Thus, it can be observed that the EKF struggles to reach steady state towards burnout of the 2nd stage and divergence is seen, especially for higher timestep sizes. In any case, the best RMSE results are obtained for a timestep size of 0.2 seconds (for both settling time and steady state error). Since there is no requirement regarding the frequency at which the observations are obtained, it is assumed that the IR sensors are capable of obtaining observations every 0.2 seconds and this will be the timestep value chosen for the performance analysis of the following sections.

As a final note, from the previous analyses it can be hinted that the EKF performs better for in-house trajectories (see the timestep analysis test) and the PKF performs better for external trajectories (see the CB altitude test). No comment is made regarding these characteristics as the purpose of this section is to provide the reader with the general effect of the aforementioned input parameters on filter output. For a thorough comparison of the results obtained by the EKF and PKF, the reader is referred to the performance analysis carried out in the remaining sections of this chapter and the following, more specifically to the description done in Section 5.2.

Selection of SE Input Parameters

All things considered, Table 10 together with Table 1 and Table 7 summarize the values that are used for all the performance analysis tests that are carried out in the following sections and in Chapter 6. In the remaining sections of the chapter, process noise will be modified as introduced in Section 5.2. The (sub)optimal value of process noise will be fixed for the analysis in the following chapter.

Parameter [Units]	Value
Number of measurements for EKF initialization [-]	30
Cloud Break altitude [km]	11
Timestep size (observation frequency) [s]	0.2

Table 10: Chosen SE Input Parameters

Now that the effect of process noise on filter output has been studied and the Simulation Environment input parameters have been clearly defined, a performance analysis can be carried out using these values. The performance analysis of the remaining sections of this chapter is based on analyzing how process noise affects filter output, as it is the input parameter whose variation has the greatest impact on filter performance. Moreover, a further comparison of the advantages and disadvantages of the EKF and PKF will be carried out using different performance indicators. The effect of versatility on performance is left out for Chapter 6.

5.4. Filter Consistency

A filter is said to be consistent if its estimated error is a reliable indicator of the actual error [43]. Many different types of tests can be used for evaluating filter consistency [44], such as the Normalized Estimation Error Squared (NEES) and the Normalized Innovation Squared (NIS) [34]. These types of tests analyze consistency from a qualitative point of view by assessing whether the Gaussian estimations produced by a filter follow a certain distribution. For the work performed in this thesis, the NIS test is preferred over the NEES test due to its potential advantages for real-time applications (see Chapter 7).

The consistency of a filter can also be measured from a quantitative point of view. This can be achieved by analyzing whether the filter's self-assessment about its own estimation error is realistic and trustable [45]. More specifically, the consistency (or credibility) of a filter can

be quantitatively measured by determining how similar to the real error is the filter's output covariance error. This closeness will be analyzed utilizing the Non-Credibility Index (NCI) performance indicator introduced by [45].

In this section, a consistency check will be applied to the Normalized Innovation Squared by testing if the NIS is within the boundaries defined by a χ^2 probability distribution. This test will determine whether the output produced by the EKF and the PKF is consistent from a qualitative point of view. Then, the credibility of each of the filters will be measured from a quantitative standpoint using the Non-Credibility Index. These analyses will be carried out for several different values of process noise.

5.4.1. Consistency Check

Consistency is measured by performing an analysis of the innovations produced by the filters. The innovation of a filter is the residual obtained from the difference between the predicted filter state and the obtained measurements, for a given epoch. If these residuals are too large, it can be expected that the consistency of the filter will be low. In this consistency analysis, it is crucial to assume that the estimate is unbiased and follows a Gaussian distribution. By using the innovation and the innovation matrix S_k -see Equation (114)-, the Normalized Innovation Square (NIS) performance indicator is obtained. The NIS measures the information provided by a new observation added to the estimation process [46]. Note that the definitions of the matrices used for the innovation matrix are provided in Chapter 2.

$$S_k = H_k \bar{P}_k H_k^T + R_k \tag{114}$$

The NIS test uses the value obtained from computing the normalized innovations squared according to Equation (115), for n Monte Carlo runs (i is the i^{th} MC iteration). ν_k is the residual ($\bar{x}_k - z_k$) at epoch k.

$$q_k(i) = \nu_k^T(i) S_k^{-1}(i) \nu_k(i) \tag{115}$$

The average value $\bar{q}_k$ of all the MC iterations is computed. Since ν_k is (assumed) normally distributed, $\bar{q}_k$ should follow a χ^2 distribution. Further, $\bar{q}_k$ should be a random variable with $n \cdot N_{MC}$ degrees of freedom, where n is the number of elements in the state vector.

Let α be a real positive number $\alpha < 1$. Then, one can say that the filter is consistent with probability $1 - \alpha$ if [47]

$$\chi^2_{nN_{MC}}\left(p < \frac{\alpha}{2}\right) < \bar{q}_k < \chi^2_{nN_{MC}}\left(p < 1 - \frac{\alpha}{2}\right) \tag{116}$$

That is, if the NIS value lies within the upper and lower boundaries defined by a χ^2 distribution with significance $\alpha = 0.05$, the test is passed. Moreover, a sliding window for several epochs can be introduced [48] to provide more information to the result of the test,

since the result of a single NIS value outside of the threshold defined by the χ^2 distribution would trigger a failed test case. This means that the following Equation (117)

$$J_k = \sum_{i=k-l}^{k} \frac{\bar{q}_i}{w} \qquad (117)$$

can be applied to Equation (116) from epoch $k - w - 1$ to epoch k, substituting $\bar{q}_k$ by J_k. w is the length of the sliding window and $l = w - 1$. Using this sliding window will ensure that divergence is avoided by a single NIS value outside of the threshold. Note that in this case, the degrees of freedom of the Chi-Square distribution will be $w \cdot n \cdot N_{MC}$.

All things considered, the following Table 11 shows the results of the NIS test for two cases of process noise (low and high). For the EKF, the process noise low value is the optimal one obtained for in-house trajectories and the high value is the optimal one obtained for external trajectories. For the PKF case, low process noise indicates the optimal value obtained for both in-house and external trajectories. High process noise is just a value two orders of magnitude higher than the optimal one. The sliding window used is two seconds ($w = 10$ epochs, i.e. timesteps).

NIS TEST		**EKF**	**PKF**
In-house	Low Noise	Passed	Passed
	High Noise	Passed	Passed
External	Low Noise	Failed	Passed
	High Noise	Passed	Passed

Table 11: NIS tests

From the NIS test, it is clearly seen from a qualitative point of view that the PKF works in all cases of noise and for external and in-house trajectories. On the other hand, the NIS test fails for the EKF when a low value of process noise is used for tracking external trajectories. These values are in accordance with the results obtained from the stability analysis of Section 5.2 in which filter divergence was obtained.

A very interesting remark can be made regarding this previous assessment. It is possible to implement the NIS test real-time, so process noise (EKF) can be adaptively changed based on the result of the test. Furthermore, it is important to notice that no information from the true trajectory is necessary for performing the NIS test, unlike other types of tests (e.g. NEES test). Therefore, the NIS test cannot only be used as a tool for assessing filter consistency but also as a tool for online (real-time) tuning of process noise.

Finally, note that the results obtained from the NIS test only produce a rough qualitative assessment of the performance of the filter. No information regarding the consistency of each filter, i.e. as a function of time, can be inferred from the NIS test. Furthermore, even though the NIS test allows us to realize whether the filter is consistent or not, questions such as

which filter is more consistent -credible- and for which cases might arise. Therefore, another consistency performance indicator is introduced in order to answer such questions.

5.4.2. Non-Credibility Index

In order to know *by how much* a filter is reliable, the Non-Credibility Index (NCI) is used [45]. The NCI provides a quantitative assessment of the consistency of a filter. Indeed, the NCI allows addressing the issue of reliability of an estimator's self-assessment from a quantitative point of view. Therefore, the term credibility is preferred to be used in this case over consistency, since the NCI will provide an answer to which filter can be more credible and for which types of (launch) scenarios.

The working principle of the NCI is very simple. The NCI is a measure -ratio- of the true and expected errors of the filter. Therefore, the NCI must be implemented in a Monte Carlo environment, and the true trajectory of the launch vehicle must be known. The NCI is actually a NEES ratio: the actual NEES normalized by the ideal NEES [49]. The definition of the (actual, averaged) NEES is

$$\overline{NEES}_k = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \tilde{x}_k^T(i) \hat{P}_k^{-1}(i) \tilde{x}_k(i) \qquad (118)$$

where P_k^{-1} is the inverse of the covariance and $\tilde{x}_k = x_k - \hat{x}_k$ the error (estimate minus truth) at epoch k for N_{MC} Monte Carlo runs. The ideal NEES is computed similarly, but using the Mean Squared Error (MSE) matrix instead of the estimated error covariance matrix P_k since the ideal error covariance matrix is not known in practice [45]. The MSE matrix is approximated by the outer product $\Sigma_k = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \tilde{x}_k \tilde{x}_k^T$.

Finally, the NCI indicator is given by

$$NCI_k = \frac{10}{N_{MC}} \sum_{i=1}^{N_{MC}} \log_{10} \rho_k(i) = \frac{10}{N_{MC}} \sum_{i=1}^{N_{MC}} \log_{10} \frac{\tilde{x}_k^T(i) \hat{P}_k^{-1}(i) \tilde{x}_k(i)}{\tilde{x}_k^T(i) \Sigma_k^{-1}(i) \tilde{x}_k(i)} \qquad (119)$$

An important remark is that ρ_k must remain always positive. Therefore, P_k and Σ_k must be always symmetric positive definite matrices so their inverse is also symmetric positive definite [50]. In this way, it is necessary to run the MC environment at least a number of times higher than the length of the state vector (degrees of freedom) in order to ensure that the Σ_k matrix is invertible (positive definite): $N_{MC} > n$. Regarding the error covariance matrix obtained by the filter P_k, the symmetry and positive semi-definiteness is ensured by using the Joseph Formulation in the update step of the Kalman Filter (see Chapter 2). Also note that the NCI is multiplied by an amplification factor in analogy to the signal-to-noise ratio.

The obtained value of NCI gives a measure of the relation between the actual and ideal output errors produced by the filters. In this way, a perfectly credible filter would have an NCI value equal to zero. NCI greater than zero indicates that the estimator is overly optimistic, that is, the filter outputs an error smaller than the actual error. NCI smaller than zero indicates that the estimator is overly pessimistic so the error output by the filter is greater than the actual error. In any case, overly optimistic results must be avoided as an error greater than the reality means that the $1-\sigma$ confidence region obtained by the filter is smaller than the actual one (so in reality less than 68% of the results would lie within the $1-\sigma$ boundaries). As an example, the definitions of optimism and pessimism can be clearly seen by checking the standard deviation obtained from the MSE matrix and the error covariance matrix as shown in Figure 16. It can be seen that the actual error of the overly optimistic filter is lower than the ideal one. This means that one cannot assert that the actual e.g. burnout location of the launch vehicle will be within the values obtained by the error covariance of the filter with a probability of 68%. In truth, this probability will be an (unknown) smaller value. On the other hand, for an overly pessimistic filter, the probability of finding the actual burnout location of the vehicle within the $1-\sigma$ error boundaries will be an (unknown) value higher than 68%. This second option is thus preferred since we can assure that the $1-\sigma$ confidence region given by the error covariance of the filter is at least achieved.

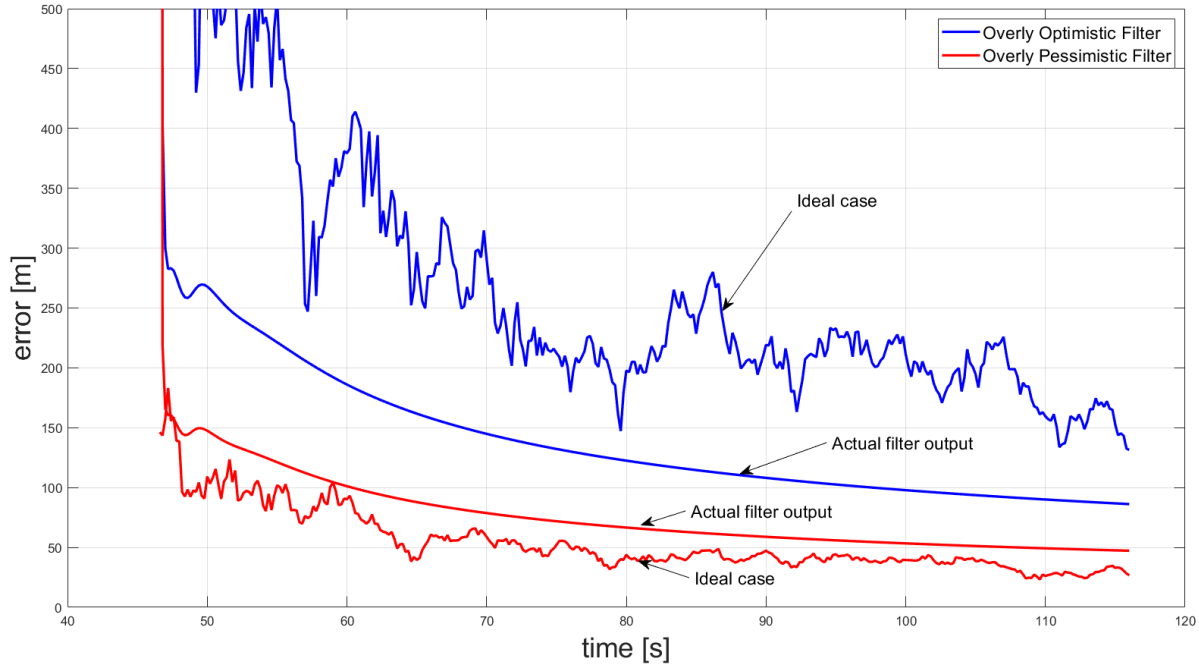

Figure 16: NCI explanation. An overly optimistic filter will have a covariance that provides an error (standard deviation) that is smaller than the reality. On the other hand, if the filter is overly pessimistic, its output covariance error will be worse than reality

All things considered, the NCI value with respect to time has been obtained for both the EKF and the PKF. The same trajectories used for the filter stability analysis carried out in Section 5.2 (nominal GT and Vega E) have been used. Moreover, the effect of process noise on filter consistency has been analyzed by using three different values as shown in Table 12.

Process Noise	EKF	PKF	
Low[§§]	[0.05]	$[10^{-4}]$ in x, y, z (full thrust)	$[1, 0.5, 1] \cdot 10^{-3}$ in x, y, z (tail-off thrust)
Medium[***]	[1.0]	$[5 \cdot 10^{-4}]$ in x, y, z (full thrust)	$[7, 7, 7] \cdot 10^{-3}$ in x, y, z (tail-off thrust)
High[†††]	[3.5]	$[0.01]$ in x, y, z (full thrust)	$[0.1, 0.1, 0.1]$ in x, y, z (tail-off thrust)

Table 12: Values of process noise for all tests

In-House Trajectories

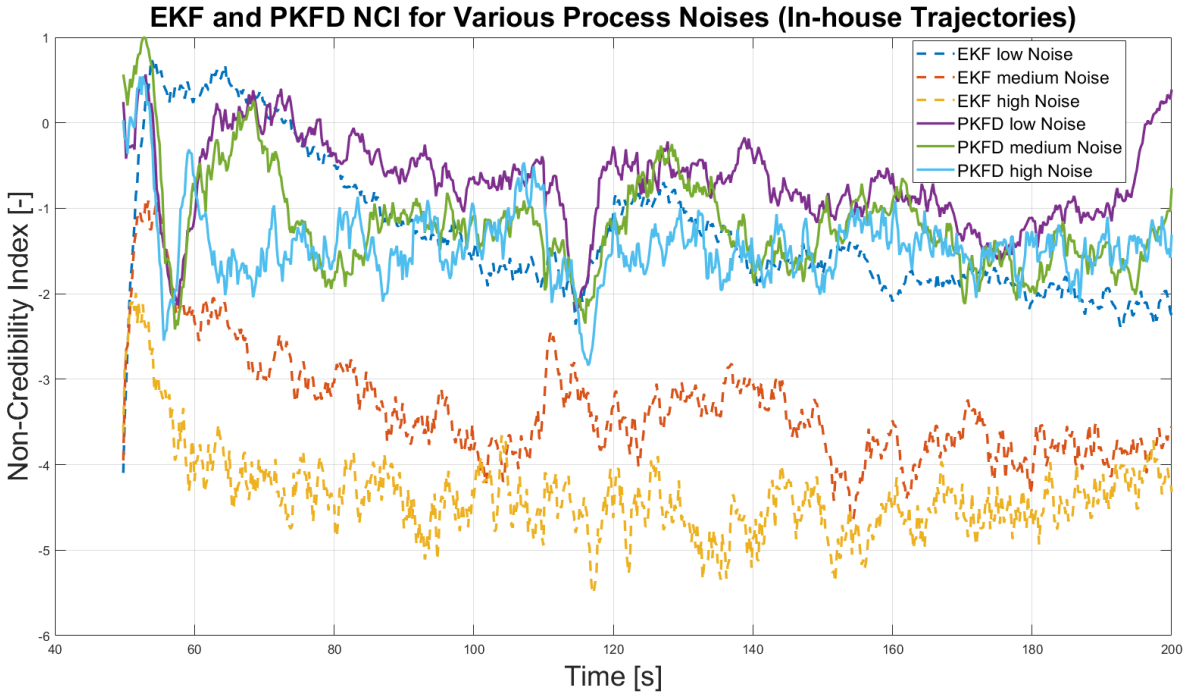

Figure 17: NCI for in-house trajectories. The NCI for EKF is greatly dependent on the amount of process noise, whereas for PKF stays relatively at the same level, no matter the amount of process noise. In absolute terms, the PKF tends to show a better credibility, as the PKF outputs are closer to 0 than the EKF outputs

Figure 17 shows the computed NCI for in-house trajectories for both EKF (dashed line) and PKF (continuous line). In general terms, it can be seen that both filters tend to have a negative value of NCI, which is desirable. Regarding the EKF, higher values of process noise increase the absolute value of NCI. Therefore, adding process noise to the EKF decreases filter credibility. On the other hand, variation of process noise for the PKF does not display great differences. However, if the PKF process noise is too low, filter divergence occurs, and credibility is lost (positive NCI value at burnout for PKF low noise). It can be concluded that the PKF is not so sensitive to the variations of process noise in terms of filter consistency/credibility. This fact was already shown in the previous section using the NIS test. Moreover, the optimal noise case (low process noise, in-house trajectories) of EKF has a

[§§] Note that the low case of EKF process noise corresponds to the suboptimal process noise of EKF (in-house trajectories)
[***] Note that the medium case of PKF process noise corresponds to the optimal process noise of PKF
[†††] Note that the high case of EKF process noise corresponds to the suboptimal process noise of EKF (external trajectories)

similar credibility to all noise cases in the PKF. Therefore, for the optimum process noise, the EKF performance is as good as the PKF, at least in terms of filter credibility. Nevertheless, this conclusion can only be made for in-house trajectories since, as it was previously seen, it is necessary to increase the EKF process noise when external trajectories are being tracked.

External Trajectories

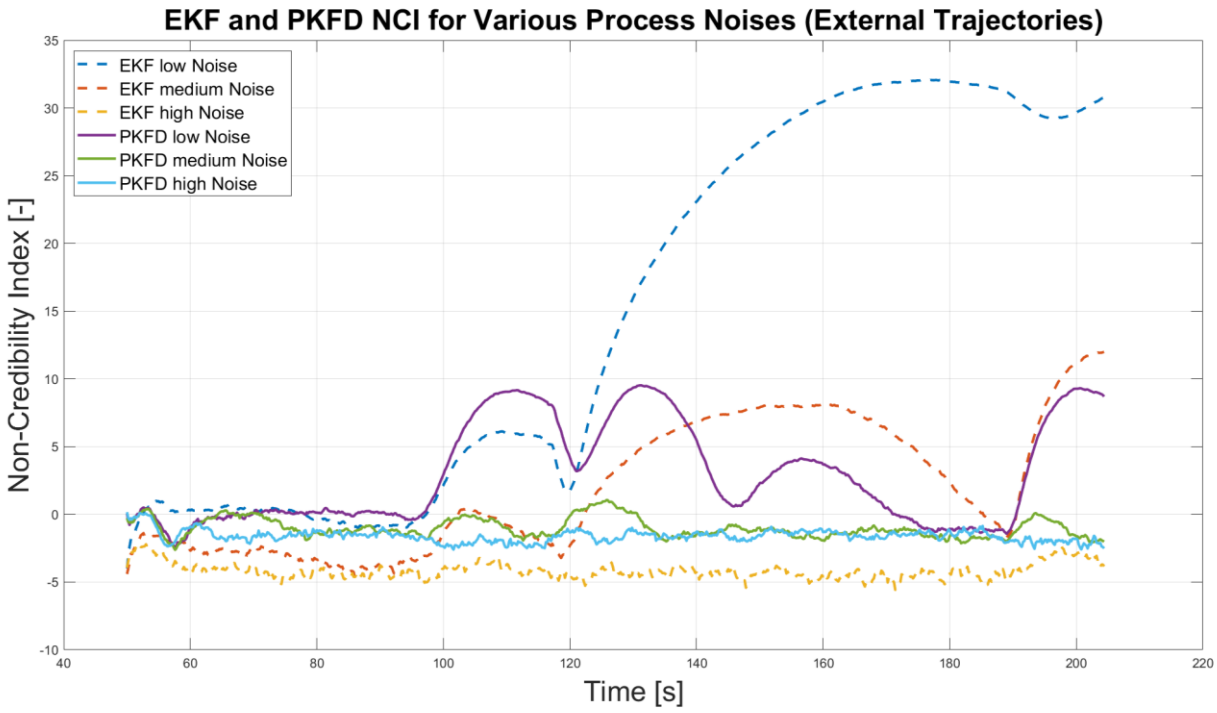

Figure 18: NCI for external trajectories. In the same manner as for in-house trajectories, the PKF behaves better than the EKF in terms of credibility. The only level of process noise in which steady state is achieved with the EKF is one order of magnitude than the NCI values obtained with the PKF

Figure 18 displays the NCI values for the three cases of noise for both the EKF and PKF. Similarly, as it occurred for in-house trajectories, if the value of process noise of the PKF is sufficient so divergence is avoided, filter credibility remains approximately at the same level, no matter the amount of noise (at least for the studied values, presumably for extremely high values of process noise this fact does not hold true). On the other hand, the credibility of the EKF is lost for low values of process noise due to filter divergence. For the higher level of EKF noise, credibility is maintained (a steady state level is reached) but the NCI value is lower than the one obtained for the PKF.

From the consistency analysis carried out for the EKF and PKF, it can be concluded that both consistency and credibility for the PKF is higher than for the EKF, in general. Moreover, there is not a single case for which the credibility of the EKF is higher than for the PKF. Even for in-house trajectories (in which the rocket trajectory is generated using essentially the filter's dynamic model), the EKF best case of noise provided the same results as the PKF in terms of credibility. Furthermore, an interesting remark can be made regarding process noise: higher levels of process noise decrease filter credibility, but always towards the negative zone. Thus, the filter always remains overly pessimistic, which is desirable. Finally, the NCI

value with respect to time could be used for process noise tuning, in the same way the RMSE was used in Section 5.2. This can clearly be applied for tuning the EKF process noise, although it is not so straightforward for the PKF since different values of process noise do not represent high changes of the NCI value. For values of process noise that are too low, NCI diverges as the filter is not able to properly track the vehicle's trajectory. If the value of process noise is too high, the credibility will be low. Therefore, a tradeoff can be performed, in a way that an optimum value of process noise is found (as close as possible to zero from the negative side but avoiding divergence).

5.5. Filter Accuracy

Accuracy is defined as the overall distance between the estimated values by the filter and the truth [38]. Several different accuracy indicators can be used to analyze the performance of the EKF and PKF such as the Mean Absolute Error (MAE), the Mean Square Error (MSE), or the Root Mean Square Error (RMSE). One of the main drawbacks of using the RMSE for evaluating accuracy is that it tends to be dominated by outlying estimates far away from the true value because the squared of the differences is used to compute its value. Nonetheless, for the work presented in this thesis, the RMSE is chosen as the performance indicator for filter accuracy since the scale it uses is the same as the original measurements (unlike the MSE) and it is widely used in the target tracking literature for evaluating filter output. Moreover, the RMSE can be used for other applications besides analyzing filter accuracy. The reader is referred to Section 5.2 for an in-depth explanation of the different applications for which the RMSE is used in this thesis, as well as a description of its computation.

In this section, the RMSE is computed for $N_{MC} = 100$ Monte Carlo runs, as introduced in Chapter 3, and for each epoch in which there is available an estimation of the filter. It is interesting to note that the RMSE is equal to the standard deviation if bias is zero [51]. In this way, information regarding the bias of the estimator is contained in the RMSE. By comparing the differences between the standard deviation and the RMSE, it could be seen whether the EKF or PKF estimates produce a higher bias, for example. Nonetheless, the main focus of this section will be set on analyzing the differences in RMSE accuracy obtained between the PKF and EKF, for different values of process noise (see Table 12).

In-House Trajectories

The RMSE in position and velocity is computed for the EKF and PKF in Figure 19 and Figure 20. The RMSE in velocity for the PKF using low process noise shows divergence. Therefore, only the medium and high values of process noise for the PKF can be compared, as they are the only two that achieve filter stability (steady state). In any case, the PKF presents higher accuracy errors than the EKF for all the stable values of process noise (the highest analyzed value of process noise for the EKF has an RMSE similar to the PKF with optimal process noise). The optimal value of process noise for in-house trajectories with the EKF shows the lowest RMSE, as expected since the tuning of process noise was carried out using the RMSE.

Moreover, for the tracked launch vehicle, the staging event that takes place around 116 seconds can be seen in both figures. The effect of the staging event has a larger impact on the RMSE results obtained by the PKF. On the other hand, the EKF is almost not disturbed by this staging event. The explanation of this is based on the way the Simulation Environment carries out the estimation when a staging event without coasting occurs. The reinitialization of the filters' states that are different from position and velocity when staging occurs have a greater influence on the PKF than the EKF since the dynamic model of the PKF relies heavily on these states (acceleration and jerk) for propagation of the state vector. For the EKF, as the filter propagation depends on the value of velocity for computing the acceleration (gravity turn), the reinitialization after staging is affected in a much lesser extent. The reader is referred to Chapter 3 for more information about the SE. Finally, the decrease in RMSE seen after staging for the high noise case of PKF (Figure 20) is caused since the value of process noise is too high. In fact, the steady state estimates obtained with this value of noise have a higher error than the ones obtained after reinitialization due to a staging event.

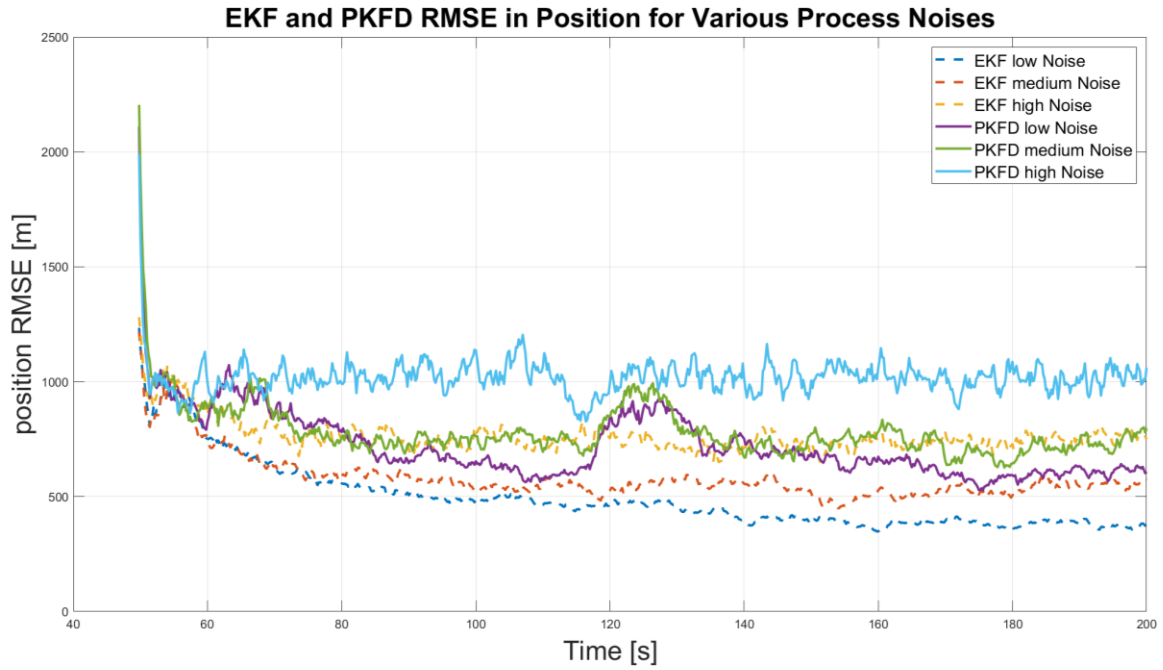

Figure 19: RMSE in position for in-house trajectories. In general, EKF shows a higher accuracy than PKF for in-house trajectories

All things considered, by inspecting Figure 19 and Figure 20 it can be concluded that the effect of varying process noise is better seen in RMSE velocity than position (i.e. divergence due to low noise is seen for PKF in velocity but not in position). This is the reason why RMSE velocity was chosen for tuning the process noise in Section 5.2.

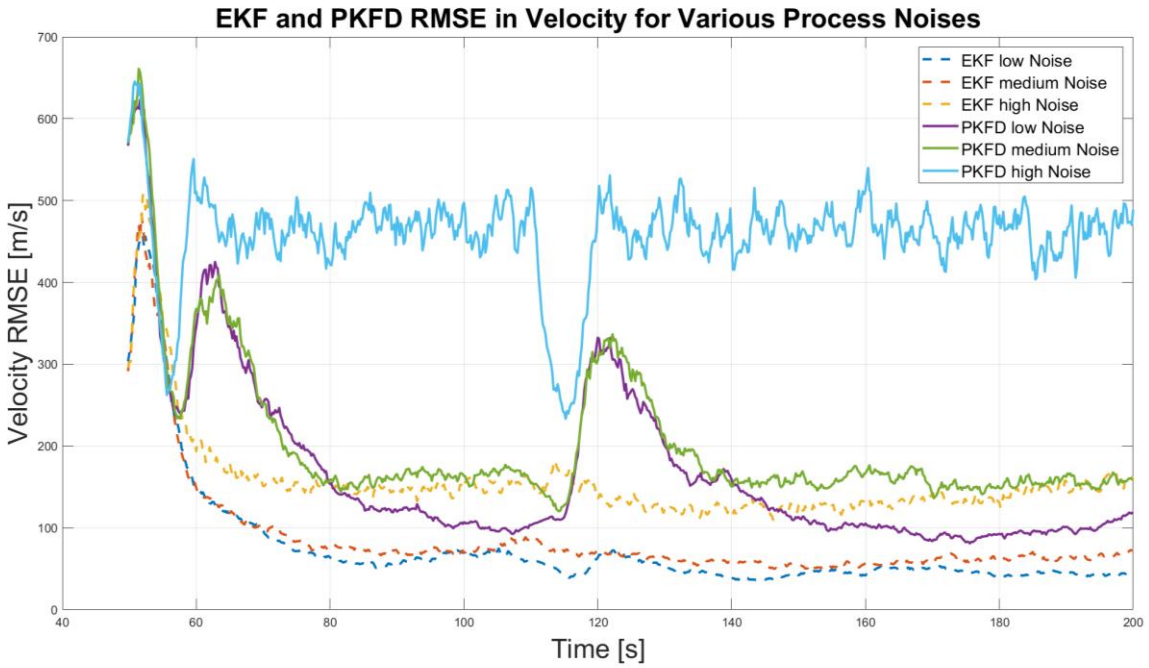

Figure 20: RMSE in velocity for in-house trajectories. The same behavior as for position RMSE is seen for velocity RMSE. EKF shows higher accuracy than PKF with in-house trajectories. Moreover, faster convergence is seen by EKF than PKF

External Trajectories

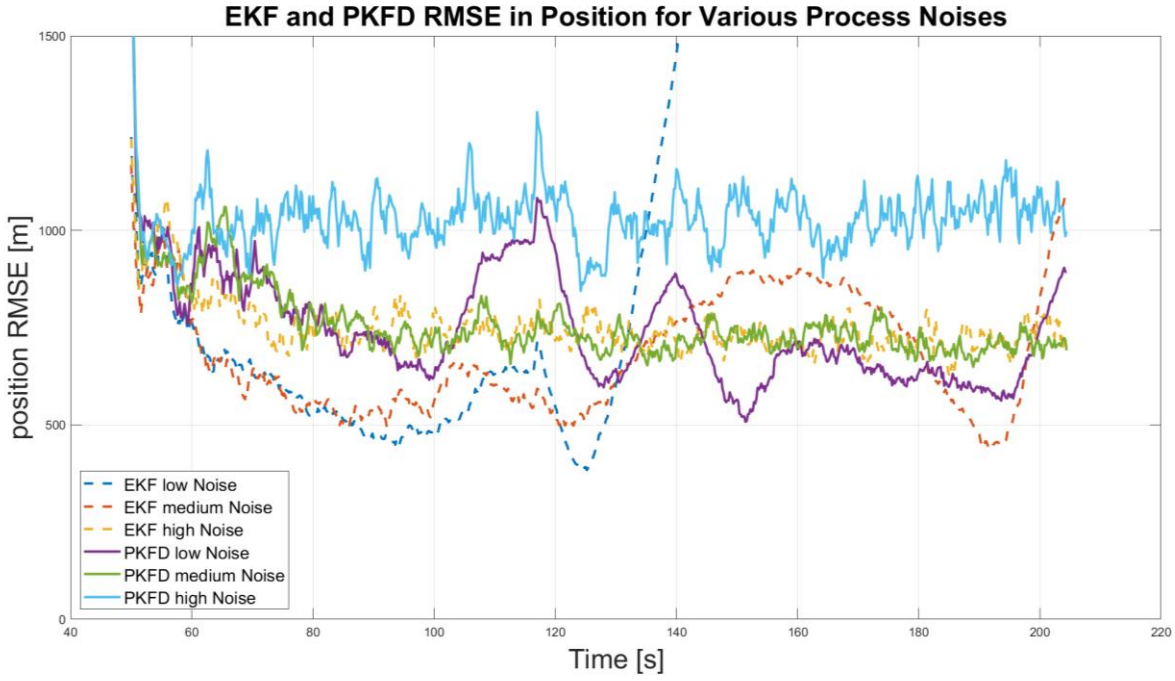

Figure 21: RMSE in position for external trajectories. Unlike with in-house trajectories, PKF shows a better accuracy performance in general than the EKF since divergence is seen for EKF for all cases of process noise except one. The case in which convergence is obtained (highest process noise), provides results close to optimal case of the PKF

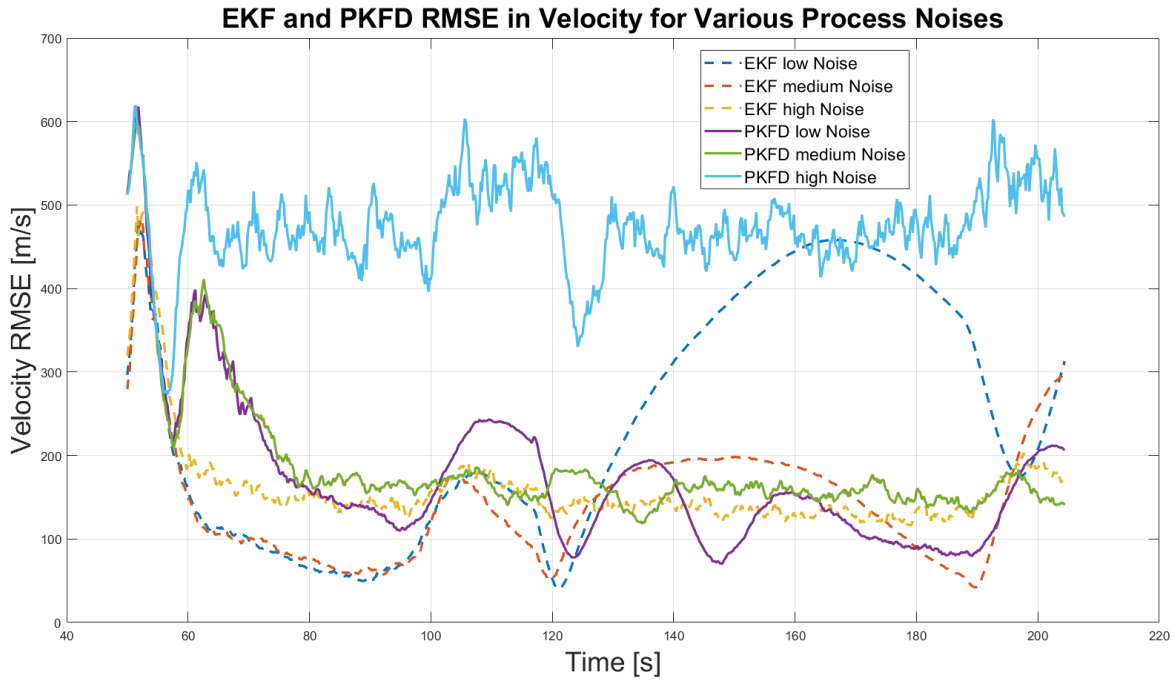

Figure 22: RMSE in velocity for external trajectories. The same behavior as for RMSE position with external trajectories can be inferred for RMSE velocity. The high noise case of the EKF shows similar results as the optimal noise case of the PKF

The accuracy results obtained for external trajectories in Figure 21 and Figure 22 show a great difference in comparison with the RMSE values obtained for in-house trajectories. Moreover, the RMSE trends found for external trajectories are quite similar in terms of steady state behavior to those obtained with the NCI indicator in Section 5.4. In this way, it can be seen that the EKF is only able to achieve filter convergence for the high noise case. The accuracy results obtained for this high noise case are quite similar to those obtained with the medium process noise for the PKF. This is in accordance with the tuning carried out in Section 5.2 in which the (sub)optimal values of process noise for EKF and PKF where found. An important remark in terms of filter comparison can be made from this result: for one given type of external trajectory, the optimal value of process noise for EKF and PKF provide similar performance in terms of accuracy.

Regarding the transition phase before steady state is achieved, the EKF provides better results than the PKF for both in-house and external trajectories (steady state is achieved faster), since each filter type has a different initialization method (see Chapter 4). Nevertheless, this parameter is not critical in terms of comparison, as the main goal of this thesis project is to obtain the best possible burnout estimate, as it was stated in Chapter 1.

The following table summarizes the burnout RMSE error with each filter for in-house and external trajectories for the best case of process noise (note that for the PKF the best case of process noise is always the same). It can be seen that the accuracy of the PKF filter is similar for both trajectories, so the accuracy is mainly affected by the optimal value of process noise. For the EKF, process noise also has a great impact on filter accuracy: increasing the value of noise for external trajectories (to avoid filter divergence) also reduces the accuracy of the

filter. As explained before, the burnout accuracy of the filters for the external trajectories is quite similar. Nevertheless, the EKF performance in accuracy is improved by a factor of ~3 when in-house trajectories are tracked.

	RMSE Position [m]		RMSE Velocity [m/s]	
	EKF	**PKF**	**EKF**	**PKF**
In-house	363.4	797.1	44.1	159.7
External	719.4	691.4	161.8	140.7

Table 13: Summary of filter accuracy performance

5.6. Filter Precision

The last performance indicator analyzed in this chapter is filter precision, which is a measure of the spread of the obtained estimates within a Monte Carlo environment. Note that precision is the same as the RMSE if the filter is perfect (no bias and an NCI value of zero for all the epochs). Since this is never the case, the analysis of the precision of the filter becomes very useful to consider the imperfections of the filters. The precision of the studied filters is based on the random errors that are contained in the observations obtained by the sensors [38]. Therefore, in order to increase filter precision, the measurement error should be as low as possible. Nevertheless, since the measurement error is a fixed value defined by the stakeholder (see Chapter 2), the precision of the filter can only be improved if the combination of dynamic and measurement models improves filter output.

In any case, the performance indicator used to analyze the precision of the studied filters is the standard deviation. The real standard deviation that is obtained by the filter appears squared in the trace of the error covariance matrix. The standard deviation computed from the error covariance matrix (after the computation of the eigenvalues, see below) provides a measure of the uncertainty of the obtained state. Furthermore, the ideal standard deviation can also be computed from N_{MC} Monte Carlo runs, the comparison between both parameters would provide a consistency indicator, as it was done in Section 5.4. In this section, however, filter precision will be investigated by purely analyzing the error covariance matrix of the filter at burnout. For $N_{MC} = 100$ MC runs, the average error covariance matrix has been obtained.

Based on the error covariance matrix obtained by the filter, there is an uncertainty region defined by an n-dimensional ellipsoid (being n the dimension of the state vector) around the obtained estimate [52]. In order to present a more meaningful performance parameter, only the position elements are used (from both the state vector and its associated error covariance matrix). More specifically, a 3-dimensional error ellipsoid around the x, y, z coordinates of the burnout position can be obtained if confidence intervals are used – obtained from a χ^2 distribution with 3 degrees of freedom [53], as shown in Figure 23. This error ellipsoid shows that the true burnout position of the rocket has a probability of $p = 95\%$ (of lying within its limits. Moreover, the orientation of the error ellipsoid cannot be directly computed from the

trace of the error covariance matrix. The orientation (i.e. the angles of the ellipsoid axes with respect to the Cartesian directions) of the ellipsoid in the 3-dimensional space is given by the eigenvectors of the covariance matrix. The length of the ellipsoid axes is given by the eigenvalues. If the non-diagonal elements of the error covariance matrix are zero, the main directions of the ellipsoid are the Cartesian coordinates. Therefore, for the decoupled version (PKFD) of the PKF a loss of information exists, since the correlation between the different directions is not computed.

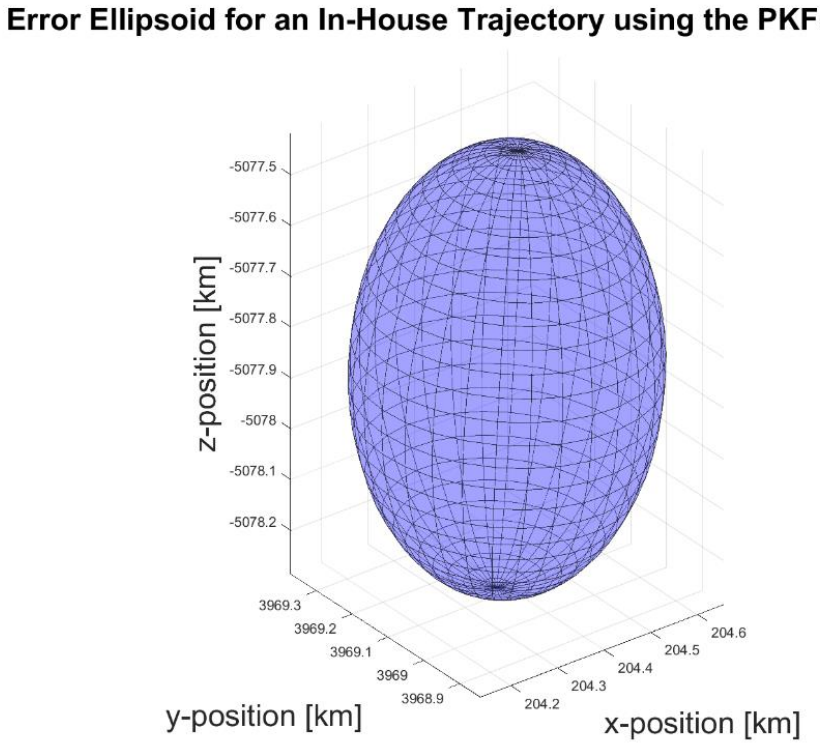

Figure 23: Example of error ellipsoid. This error ellipsoid does not correspond to any case in particular. However, it can be seen its orientation (given by the eigenvectors) and its overall size (which in turn will give its volume), given by its eigenvalues

Volume of the Error Ellipsoid

There are many approaches to analyzing the differences between the EKF and PKF by using the information provided by the error ellipsoid. In this thesis, the precision performance indicator used is the volume of the error ellipsoid. In this way, a higher volume means a larger error ellipsoid and thus, a larger uncertainty in the estimation of the true trajectory of the launching vehicle. The error ellipsoid volume is computed with the following equation [54]

$$V_n = \frac{2}{n}\frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)}(a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_n) \qquad (120)$$

where $n = 3$ is the dimension of the ellipsoid, $\Gamma(x)$ is the Gamma function and a_i is the length of the i^{th} semimajor axis, which is computed from the eigenvalues using the inverse of the chi-squared CDF for a probability of $p = 95\%$ and $n = 3$ DOF ($chi2inv$ function in MATLAB) as

$$a_i = 2\sqrt{F^{-1}(p,n)\sigma_i} \quad (121)$$

where $F(p,n)$ is the chi squared CDF and σ_i the eigenvalue in each of the main directions.

The volume (in m^3) of the error ellipsoid at burnout for inhouse and external trajectories using all the PKF versions and the EKF has been computed. For easier readability, a ratio between the obtained volume and the volume of a sphere of 100 meters in radius is provided (reference sphere). This will help the reader to compare the volume obtained by each of the filters. In any case, the volume of each ellipsoid can be re-computed with

$$V = \eta \frac{4 * 100^3}{3} \pi \quad (122)$$

where η is the volume ratio provided in Table 14.

Volume ratio, η [-]	**In-house Trajectories**	**External Trajectories**
EKF	0.0007	2.3550
PKFD	0.2001	0.2840
PKFC Cartesian	0.1732	0.2521
PKFC Polar	0.1721	0.2510
PKFM	0.1721	-

Table 14: Volume ratio of the error ellipsoid for PKFs and EKF

From Table 14, several important ideas can be inferred. First, the EKF performs extremely well in terms of precision for in-house trajectories, as the volume of the obtained ellipsoid is approximately 1,000 times smaller than the reference sphere (or equivalently, the volume of the ellipsoid is approximately proportional to a reference sphere of $10\,m$ in radius). For external trajectories, the EKF is the filter that performs the worst, obtaining more than two times the volume of the sphere of reference; that is, 10 times more the volume obtained with any version of the PKF.

Regarding the PKF, all the versions obtain values of volume of the same order of magnitude for both in-house and external trajectories. Moreover, it seems that the decoupled version obtains slightly worse results, whereas all the coupled versions obtain very similar results, meaning that all these versions have a similar performance in terms of precision (see also Section 5.1 and Annex B.1). Interestingly, the PKFM with the tuned correlation time specifically for this launch scenario shows very similar results to the PKFCP. The volume ratio of the PKFM is not displayed for external trajectories since filter stability is not achieved, which is a necessary requirement for performing the comparison. More information

regarding the PKFM can be found in Annex B.2. Moreover, from the analysis carried out for the PKFCC and PKFCP, whose only difference is the measurement model – Cartesian or polar), it can be concluded that the polar measurement model performs slightly better than the Cartesian one, probably due to the way measurement noise is converted for the Cartesian model (see Annex A).

Rotation Angles of the Error Ellipsoid

Using the eigenvectors from the error covariance matrix, the rotation angles of the error ellipsoid can be computed with the eigenvector that corresponds to the largest eigenvalue [53] with the following equation

$$\alpha_k = atan2\left(\frac{v_i}{v_j}\right) \quad (123)$$

where the i, j, k indices denote any of the three elements of the eigenvector v associated to the largest eigenvalue in the covariance matrix and α is the rotation angle.

Using Equation (123), the following Table 15 is obtained. The rotation angles do not provide any specific information in terms of filter performance. Nevertheless, it is interesting to realize whether the error ellipsoids have any relation between the different filters and if so, why this relation exists.

Rotation Angle [deg]	In-house Trajectories			External Trajectories		
	α_x	α_y	α_z	α_x	α_y	α_z
EKF	126.04	86.20	275.22	286.19	184.30	178.75
PKFD	0.00	0.00	0.00	0.00	0.00	0.00
PKFC Cartesian	2.04	1.47	35.68	2.26	1.62	35.66
PKFC Polar	2.04	1.47	35.68	2.26	1.62	35.66
PKFM	2.04	1.47	35.68	-	-	-

Table 15: Rotation angles of the error ellipsoid

From Table 15 it can be observed that the rotation angles of the EKF and PKF are especially different. Moreover, even for different types of trajectory, the results obtained from the rotation angles with the EKF differ. On the other hand, the results (for any version) of the PKF show that this difference is not so large between in-house and external trajectories. This small variation could have interesting implementations in order to predict the shape of the error ellipsoid at burnout when a new trajectory is being tracked even before the vehicle has reached burnout. Finally, the rotation angle for the PKFD is zero since for this filter there is no information regarding the correlation among each of the x, y, z directions.

In conclusion, the precision performance of the EKF is the highest when in-house trajectories are used, since the dynamic model of this filter is essentially the same as the model used to generate the in-house trajectories. Overall, the PKF (any version that uses Wiener process

models) provides better results than the EKF since its precision when tracking potentially any external trajectory is one order of magnitude better than the EKF. In the analysis carried out in this section, the precision of the filters has been derived from the volume of the error ellipsoid. No further analysis in terms of precision has been performed. Finally, the orientation of the error ellipsoid has been slightly discussed.

5.7. Conclusion: Filter Comparison

Several performance indicators have been presented in this Chapter. They have been used throughout this chapter to analyze the output produced by two trajectory-tracking filtering tools: the EKF and the PKF. The performance of each of these filters has been obtained according to four different performance indicators: stability, consistency (credibility), accuracy, and precision. The remaining performance indicator, filter versatility, has also been studied throughout the whole chapter, since two different trajectories (one in-house, nominal GT, and one external, Vega E) were used for each of the performance tests. Nevertheless, a more thorough parametric analysis will be performed in Chapter 6. Furthermore, a manual tuning method to find the (sub)optimal value of process noise for each filter has also been provided, together with an analysis of the effect of several SE input parameters on filter output/performance. This method, yet simple, allows to easily finding a value of process noise capable of achieving filter stability, which is necessary for carrying out the performance comparison of the PKF and EKF estimation tools. More complex tuning procedures could be implemented, albeit it will be left for future work, see Chapter 7. In this way, analytical methods for tuning the (input) power spectral density of the PKF based on the Wiener–Khintchine – Einstein Theorem [33], for example, could be implemented. For the EKF, adaptive noise changes based on the information of the residual seem to show promising results as well.

A qualitative comparison of each filter regarding each of the performance indicators can be performed, based on the results obtained from the performance analysis carried out in this chapter. Table 16 shows this comparison. The greatest difference between filters is found from the consistency analysis, in which the PKF extensively outperforms the EKF. In this way, the consistency of the EKF was lost if a non-optimal value of process noise was used for tracking whereas the consistency of the PKF was always maintained (provided the process noise was high enough). Furthermore, the credibility of the PKF was much higher than the EKF as the NCI value was only slightly affected when the PKF process noise was varied. For the EKF, the NCI became far from zero (on the negative side) if process noise was too high.

	EKF	**PKF**
Consistency/Credibility	Low	High
Accuracy	Medium	Low
Precision	Medium	Medium

Table 16: Qualitative comparison of EKF and PKF

In terms of accuracy, as seen in Table 16, both of the filters performed likewise for external trajectories using the RMSE. Nonetheless, the EKF showed a much better performance than the PKF for in-house trajectories, due to the similarity of its dynamic model with the true motion of the vehicle. Finally, the precision of the EKF was relatively high for in-house trajectories, but relatively low for external trajectories. Conversely, the PKF (all versions) showed comparable performance results for both external and in-house trajectories.

Chapter 6

Parametric Launch Diagnosis

In the previous chapter, several performance characteristics of the PKF and EKF have been analyzed. The purpose of this chapter is to measure another performance parameter: filter versatility. In the context of this thesis, versatility is the ability of a filter to perform a consistent, precise, and accurate tracking for any possible trajectory. The vehicle could be launched from any location (for which observability conditions exist) on Earth and could have any type of thrust profile – e.g. being a multistage rocket or having a lofted or depressed GT trajectory type.

In the previous chapter, an introductory analysis to filter versatility was already provided since two different trajectories (one in-house and one external) were used for each type of performance indicator. Nevertheless, this analysis is insufficient. The purpose of this chapter is to exhaustively study the effect of versatility on filter output. In this way, more than one different type of in-house and external trajectory will be used for the analysis. Moreover, different observability conditions will be provided (change of launch location or observing satellites location). To do this, a parametric analysis consisting of a total of 4176 tests will be performed.

Since the main objective of this chapter is to analyze filter versatility as a whole attending to varying characteristics of the true trajectory of the launching vehicle, these tests will be grouped in several different categories. That is, it is not possible to compare the effects of tracking in-house and external trajectories together, as the nature of these trajectory profiles is very different from each other. In the previous chapter it was seen that the EKF tends to perform in general much better than the PKF for in-house trajectories. The goal of carrying out a parametric analysis using different in-house trajectories will be to assess whether this difference in performance between the two filters for in-house trajectories is maintained. Moreover, as it was seen in the previous chapter, it is not possible to use just one value of process noise for the EKF, as filter divergence occurs if the value is too low with external trajectories. Therefore, the parametric analysis performed in this chapter will be based on the optimal values of process noise obtained in Chapter 5. Thus, two values of noise will be used for the EKF (one for external and one for in-house trajectories) whereas only one optimal value of process noise will be used for the PKF in this parametric diagnosis, see Table 17.

Process Noise (q or PSD)	**In-house Trajectories**	**External Trajectories**
EKF	0.05	3.5
PKF (boost/tail-off boost)	$[5 \cdot 10^{-4}]$ / $[7,7,7] \cdot 10^{-3}$	$[5 \cdot 10^{-4}]$ / $[7,7,7] \cdot 10^{-3}$

Table 17: Process noise values for parametric analysis

In general, two main types of trajectories are used for this parametric diagnosis. It is possible to generate in-house trajectories for different launch locations (longitude, latitude) on Earth. However, this is not possible to be performed for the external Vega trajectories since their launch location is always from Kourou, in French Guyana. To allow the parametrization of this type of trajectories in order to have an analysis similar to the one to be carried out for in-house trajectories, the observing satellites' location is modified. As the importance of the observations used for tracking resides in the relative distance between the launch vehicle and the satellites, this procedure is a workaround to simulate diverse launch locations of the Vega rocket for the parametric diagnosis.

Therefore, the tests to be performed are classified in two main groups: external and in-house trajectories. For each of these two groups, three different types of trajectories based on the thrust profile will be used. These are a Nominal GT, Lofted GT, and Depressed GT for the in-house trajectories and the Vega E, NE, and N for the external trajectories. In Table 2, the characteristics of these six trajectories are explained. Furthermore, for each of the different launch locations and in-house trajectory types, 10 different launch azimuths ranging from $[0°, 360°)$ have also been analyzed. The following Table 18 summarizes the tests that were performed for the parametric analysis that is provided in the following sections. Each of these tests corresponds to tracking one trajectory within the Simulation Environment for $N_{MC} = 30$ Monte Carlo runs; unlike in Chapter 5, in which $N_{MC} = 100$ were used. This value is preferred for the parametric diagnosis since it improves simulation speed. Furthermore, the precision of the obtained results for $N_{MC} = 30$ can be considered to be enough, as it increases linearly with the square root of the number of MC runs [38] (so using 30 instead of 100 MC runs only decreases result precision by less than half).

		Number of Tests
Parametric Diagnosis: In-house Trajectories	Nominal GT	$62 * 10$
	Lofted GT	$62 * 10$
	Depressed GT	$62 * 10$
Parametric Diagnosis: External Trajectories	Vega E	76
	Vega NE	76
	Vega N	76
	Total tests per filter type	2088
	Total Tests	**4176**

Table 18: Summary of the parametric diagnosis

As a final remark, this parametric analysis is mainly focused on studying the differences between the EKF and the PKF using the indicators introduced in the previous chapter. Nevertheless, since an analysis and comparison between the filters in terms of precision, accuracy, and consistency was already performed, these indicators will only be used for observing (if any) the relative differences between them when the trajectory conditions change. In other words, it could be possible that the versatility of one filter over the other is higher in general even if the performance indicators used to check this versatility (RMSE,

NCI) show worse results for this filter in each specific test. Moreover, a similar parametric diagnosis of the EKF was carried out in previous work at ADS to analyze the effects of the modification of the LP, trajectory type, etc. using in-house trajectories. This work showed that the EKF obtains worse results for specific observability conditions. That is, the obtained estimates of the filter are further from the truth when the tracked vehicle is flying towards one of the observing satellites, as the variation in LHZ elevation or azimuth is smaller and the measurement noise stays the same. Therefore, it is not necessary to study this again but to just evaluate possible differences between the filters using the introduced performance indicators. In the following analysis, only the burnout performance values (NCI, RMSE) are shown.

6.1. In-House Trajectories

The parametric diagnosis for in-house trajectories has been carried out by selecting different launch locations based on the common observability region of the two geostationary satellites. Both satellites are in the same GEO orbit separated by a fixed anomaly of $\nu = 49.8°$. Using the azimuthal and elevation coordinates of Kourou as the subsatellite location (nadir) of the middle point between the two satellites on their GEO orbit, the projection on Earth's surface of the location (anomaly) of each satellite can be obtained. The coordinates of Kourou and each of the satellites (azimuth and elevation) are provided in Table 19. Note that for simplification, the inclination used for the GEO orbit is not zero, but equal to the elevation coordinates of Kourou. This has not major effect on the parametric analysis.

	Azimuth [°]	**Elevation [°]**
Kourou	-52.775	5.236
Satellite 1 (west of Kourou)	-77.675	5.236
Satellite 2 (east of Kourou)	-27.875	5.236

Table 19: Kourou and subsatellite points coordinates

Knowing the location of the satellites, the combined access area (the region for which the sensors located in both satellites are able to observe – note that for this work it is assumed that the sensors' footprint is equal as the access area [55]) is obtained. Then, a grid with 76 different launch locations within this access area is created. The formulae to calculate the launch locations and the combined access area of the satellites is provided in the work of Wertz [55].

Figure 24 shows the access areas of each satellite and their combined access area, with the different launch locations that have been tested. The limit cases (those points that are on the edges of the access area) have not been studied as the purpose of this analysis is to just compare both filters in terms of versatility. It is left for further work a parametric analysis with the specifics of filtering trajectories in the limit cases. Moreover, there could be trajectories close to the access area' edges whose burnout point is outside, so observations would not be available. These cases have not been considered in this study for the same reasons as above. In Figure 24, the launch points range from an azimuth $[-97.996°, -7.554°]$ and an elevation $[-56.659°, 68.289°]$. The maximum and minimum

elevations are not equal in absolute terms due to the inclination of the satellites' orbit. This is the reason why the access areas of Figure 24 are not symmetric with respect to $\varepsilon = 0°$.

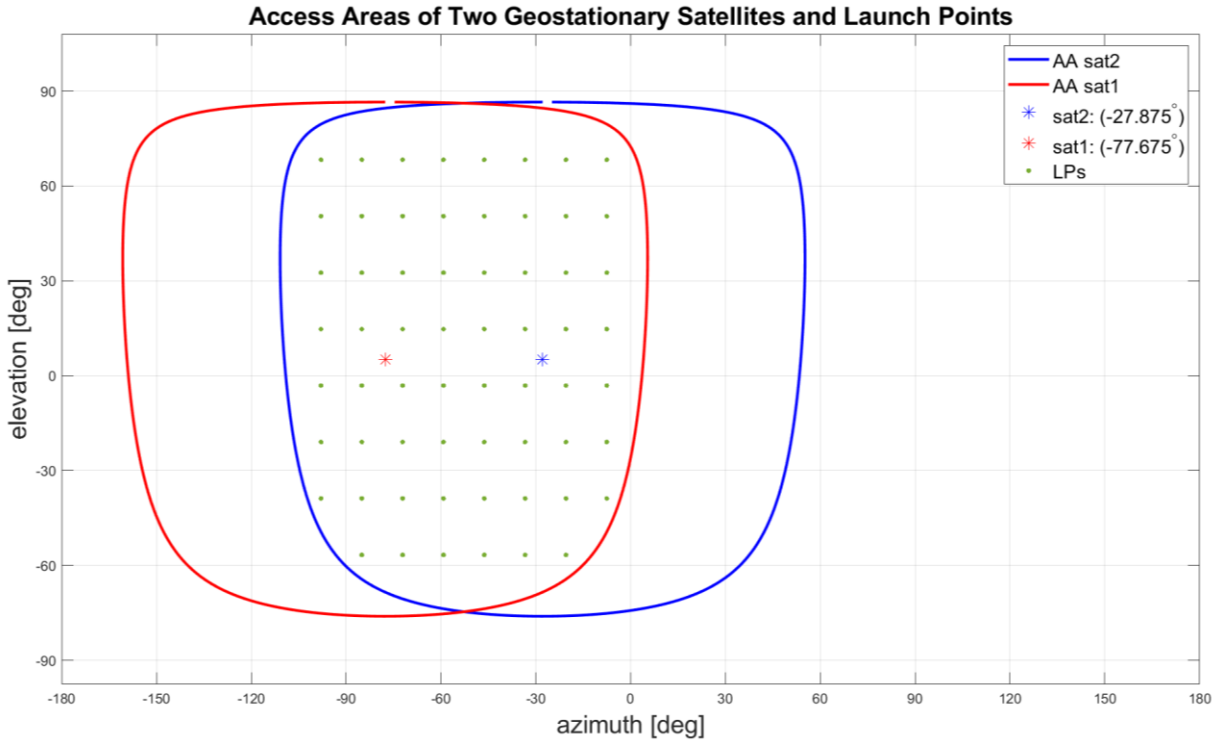

Figure 24: Launch points for the parametric analysis. Different launch locations on Earth's surface can be seen from two GEO satellites separated by a certain true anomaly. Each of these satellites will have a certain access area, and the combination of the access areas gives the region in which the launching vehicle is observable

All things considered, several types of studies can be carried out to compare the differences of the EKF and PKF using the performance indicators (NCI and RMSE) introduced in the previous chapter. First, a global comparison of the filters based on the NCI indicator is performed. Then, the effect of different launch azimuths is calculated (by averaging the indicator's results obtained for each of the launch points) for each filter. Finally, the effect on filter performance of tracking trajectories from different launch locations is studied for both the EKF and PKF. These analyses are carried out with the goal of potentially detecting any local and global trends within and between the results from each filter.

NCI Analysis on Filter Versatility

It was already discussed that NCI values greater than zero should be avoided, as this means that the filter is overly optimistic. It would be interesting to analyze which filter perform better within this parametric analysis in terms of credibility. In this way, the filter with the largest number of negative NCI values that are closer to zero can be considered to be more versatile, since this filter is more credible in general.

Figure 25 shows the percentage of tests that are above a certain minimum NCI value. It can be seen that for a certain minimum NCI value, the percentage of PKF tests is always greater

than the EKF tests. This means that the credibility of the PKF is in general greater than the EKF. For example, for a minimum NCI of -1.5, there are about 30% of EKF tests and 80% of PKF tests above this value. Moreover, some of the tests show an NCI value above 0. More specifically, a total of 32 PKF tests compared to 3 EKF tests have an NCI above zero (from a total of 1860 tests for each of the filters). The small amount of overly optimistic tests for the EKF is due to the high process noise (3.5% of covariance increase) that has to be used for the EKF. In any case, none of the NCI values for EKF and PKF is above 0.6 (maximum of $NCI_{PKF} = 0.471$ and $NCI_{EKF} = 0.560$).

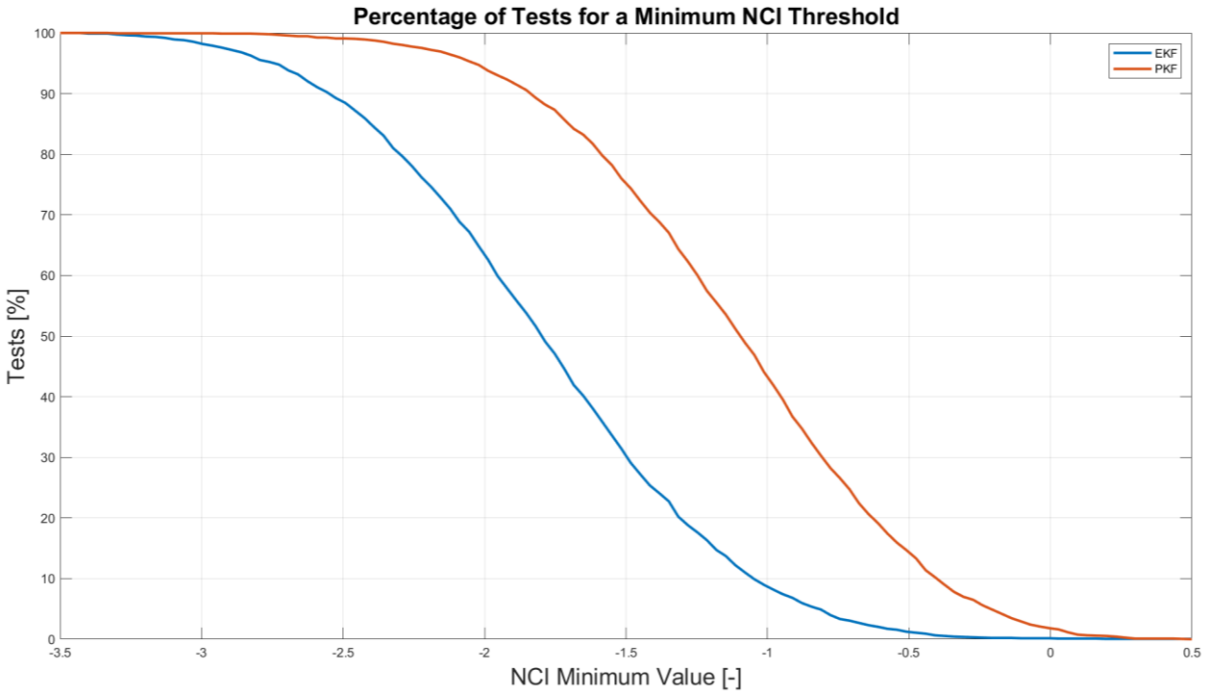

Figure 25: Parametric credibility comparison. For a certain minimum NCI threshold, the percentage of tests that are closer to zero from the parametric versatility analysis is higher for the PKF than for the EKF. This means that the PKF is in general a more credible filter than the EKF

The EKF test cases with an NCI value greater than zero are always found for launch locations close to the edges of the observability region. No relation between credibility and type of trajectory (lofted, depressed, and nominal GT) can be seen. On the other hand, for the PKF, all but one of the NCI tests above zero that are not on the edges of the observability region correspond to lofted or depressed trajectories. It will be left for future work a more in-depth analysis of this behavior.

Figure 26 shows the ratio of PKF to EKF tests for a minimum NCI threshold. It can be seen that this ratio is (almost) always greater than one, meaning that for a given minimum NCI value, the number of tests greater than that threshold is larger for the PKF than EKF. This can also be inferred from Figure 25, as the curve shown in Figure 26 is obtained by just dividing the two curves from Figure 25. The maximum is achieved for an NCI value around -0.25. That is, the overall performance of the PKF is better than the EKF as this filter has more tests with an NCI closer to 0 (on the negative side).

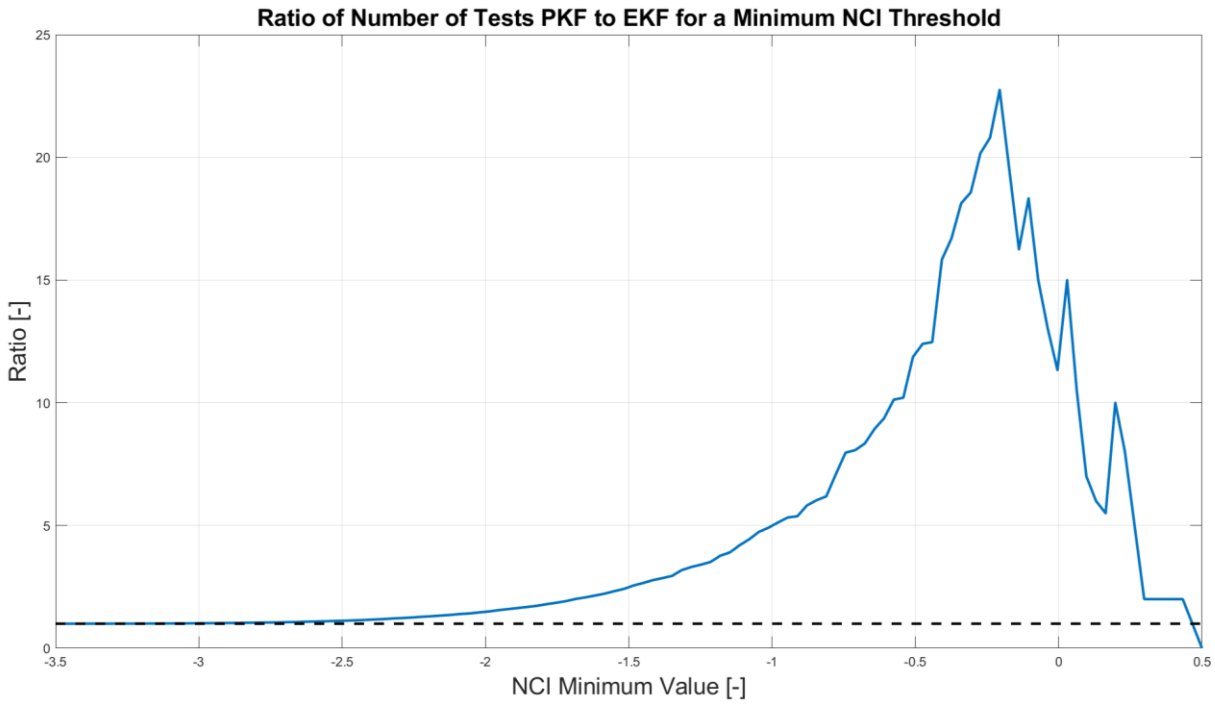

Figure 26: Ratio of PKF to EKF test regarding minimum NCI. In line with the results from the previous figure, it can be seen that for a certain number of tests, there are more PKF tests than EKF tests with a minimum value (closer to zero). The largest amount of tests using the PKF is found for a NCI minimum value of around -0.25. This confirms that the PKF is overall a more credible filter than the EKF, for any possible launch scenario

Effect of Trajectory Type on Filter Versatility

The previous analysis is a global study of the versatility performance of each filter for all the tests carried out. Nevertheless, it could be interesting to analyze whether there are specific trends with respect to the type of trajectory (lofted, depressed, and nominal GT) or the launch azimuth. It was seen from the global NCI versatility analysis that the PKF had a lower credibility for the lofted and depressed trajectory types (even with some of NCI values being above zero). In effect, the following Figure 27 shows that for the PKF optimal value of process noise, the NCI indicator shows values further from zero for the nominal GT trajectory. Conversely, this behavior is not displayed by the EKF. This is probably due to the large value of process noise used (see Table 12), which makes the filter be more insensitive to changes in terms of acceleration profile. In general, Figure 27 shows that the credibility of the PKF is higher than the EKF, supporting the assumption that the EKF has a lower credibility than the PKF that was already studied in Chapter 5.

Figure 28 shows the same analysis using the RMSE performance indicator for position (top plot) and velocity (bottom plot). In terms of accuracy, the EKF has smaller errors than the PKF. This was already seen in Chapter 5: for in-house trajectories the EKF has a better accuracy than the PKF. Further, the obtained values of RMSE for PKF show that the effect of varying launch azimuth is almost negligible. On the other hand, tracking trajectories with launch azimuths closer to westwards or eastwards trajectories provide slightly worse RMSE

results for the EKF. Finally, the RMSE obtained for the nominal GT trajectory for both EKF and PKF are, to some extent, better than for the other two types of in-house trajectories.

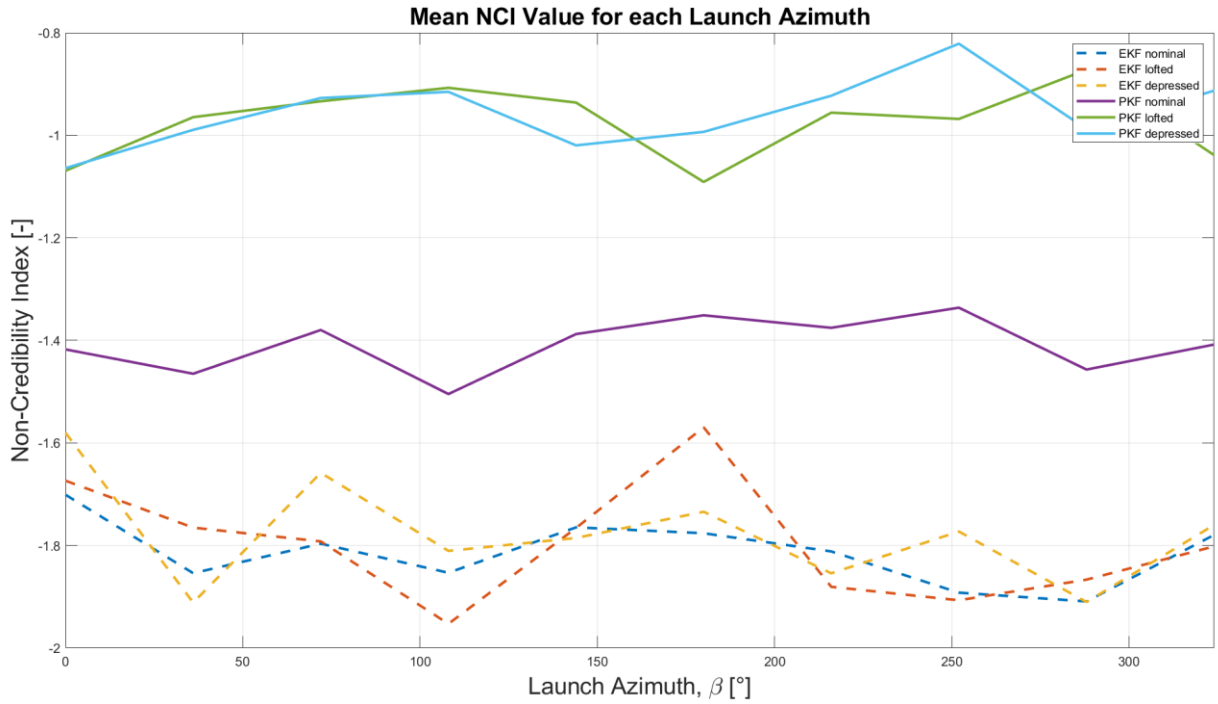

Figure 27: Comparison of launch azimuth and trajectory types – NCI. Tracking vehicles with different trajectories (lofted, depressed, minimum energy) affects more to the PKF than the EKF. In any case, the credibility of the PKF is always better for the PKF (closer to zero) than for the EKF, independently of the launch azimuth or the type of trajectory.

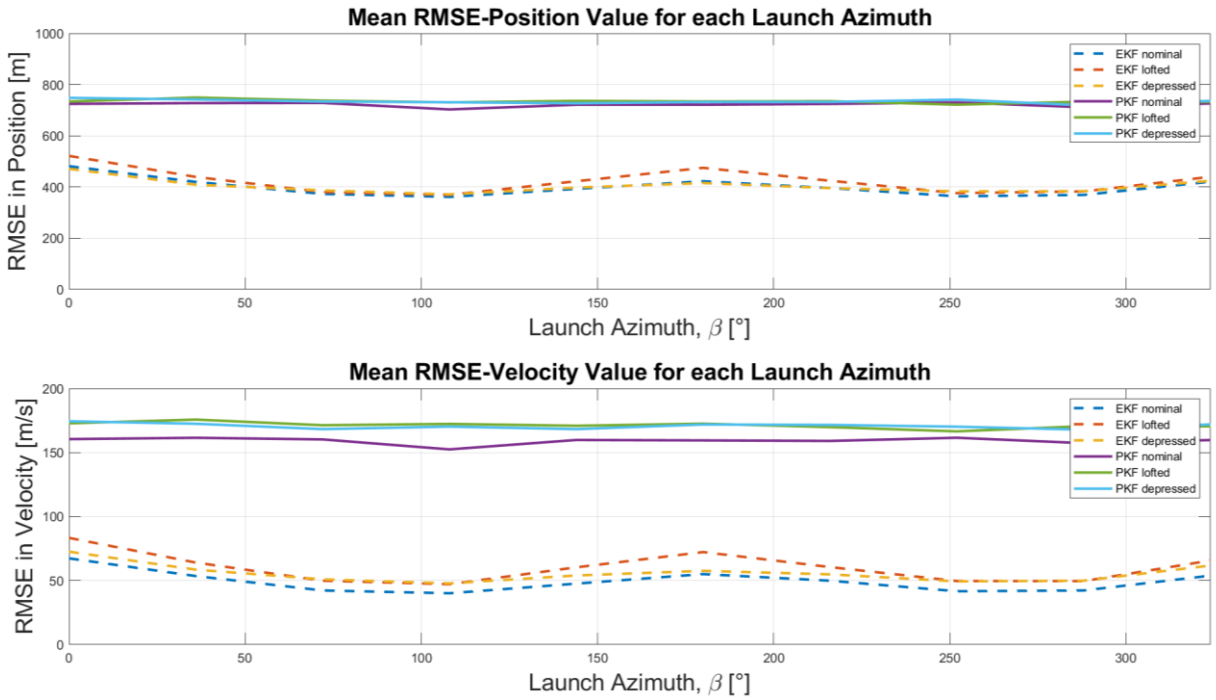

Figure 28: Comparison of launch azimuth and trajectory types – RMSE. In terms of RMSE, the EKF is more sensitive to the launch azimuth and the type of trajectory than the PKF, as the latter shows an almost constant behavior with respect to the launch azimuth.

Effect of Launch Location on Filter Versatility

Finally, the effect of different launch locations on filter performance will be studied. In this study, we are only interested in obtaining the relative terms of performance in order to compare both filters. The normalized RMSE indicator will be used for this purpose. In this way, the magnitude of RMSE obtained for both EKF and PKF can be compared. This has been done by averaging the RMSE in position for the different launch azimuths (results in velocity are similar to the ones in position, so it has been omitted for clarity's sake). Then, the obtained RMSE for each position is normalized by the norm of all the RMSE values.

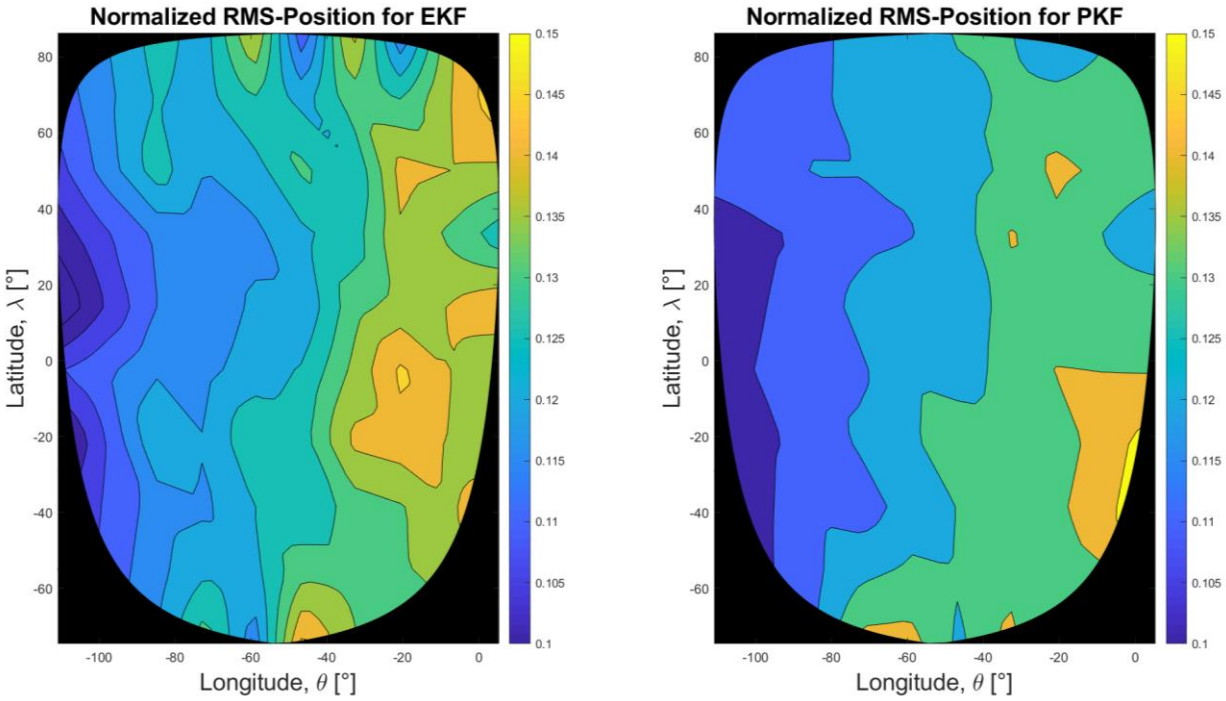

Figure 29: LP versatility analysis (in-house trajectories) using RMSE in position. The RMSE in position for both filters shows similar characteristics when the launch location is varied. These results are in accordance with those obtained in previous work at ADS

Figure 29 displays the normalized RMSE-position for EKF and PKF, for different launch locations. It can be seen that the RMSE tends to be smaller for vehicles launching from locations to the west of the observing satellites. No trend can be observed in terms of launching latitude. In any case, the focus for this thesis is on the differences between the EKF and PKF. From the results in Figure 29, it is noticeably that both filters display the same behavior. As a final remark, note the reader that a more elaborate analysis was already carried out at ADS to find the specific peculiarities of tracking rockets launching from different locations, so this aspect is not further investigated in this work. In any case, that analysis is in accordance with the results shown in Figure 29.

For in-house trajectories, it can be concluded that the PKF is less sensitive to variations of the trajectory type (Figure 28) or the launch location (Figure 29). Moreover, it has been seen that

the credibility of the PKF is better than the EKF, for the utilized values of process noise, similarly to the results obtained in Chapter 5. Table 20 summarizes the results obtained in this section. In general, the global NCI for the 1860 tests performed (for each filter) is better for the PKF than for the EKF, as it is closer to zero on the negative side. On the other hand, the PKF shows a lower (two to three times) accuracy and precision than the EKF. Furthermore, it can be concluded that the filters (both EKF and PKF) are not excessively biased, as the RMSE and 1-σ values for position and velocity are relatively similar, even though the EKF seems less biased than the PKF in general.

Parameter	**EKF**	**PKF**
Mean NCI [-]	-1.7963	-1.1123
Mean RMSE-Position [m]	408.9656	730.2461
Mean RMSE-Velocity [m/s]	55.1139	167.0975
Mean σ_{pos} [m]	418.4549	769.5158
Mean σ_{vel} [m/s]	54.6454	195.1716

Table 20: Comparison of EKF and PKF - parametric in-house trajectories

6.2. External Trajectories

A similar versatility analysis as the one performed for the in-house trajectories will be carried out in this section. The main difference resides in that different launch locations cannot be parametrized for the external trajectories since there are only three available (Vega E, NE, and N). Therefore, an analogous parametrization can be envisaged by modifying the satellite locations. This, of course, is a simplification and in reality, the effect of changing the orbital parameters of the satellites could have undesirable effects on filter performance. Nevertheless, for the purpose of this thesis, comparing the output (estimations and error covariance) of the EKF and PKF, this modification does not have any critical impact.

Figure 30 shows the different satellite locations used for this parametric analysis. These locations have been obtained using Kourou (the Vega rockets launch location) as starting point. Then, an access area using the coordinates of Kourou as a virtual subsatellite point can be obtained since the altitude of the satellites (GEO orbit) is known. Any satellite whose subsatellite point is within this access area can observe the launching point. Therefore, a grid search with a total of 76 nodes (azimuth and elevation) is used to obtain the subsatellite point of satellite 1. Then, the orbital parameters (anomaly and inclination, since the remaining ones are fixed for a geostationary orbit) of this satellite are obtained by performing a frame transformation. Since satellite 2 is in the same orbit separated by a fixed difference in anomaly (49.8°), its orbital characteristics can also be obtained. The algorithm used to follow this procedure together with the spherical trigonometry calculations necessary to obtain the access area for Kourou is calculated using the equations provided in [55].

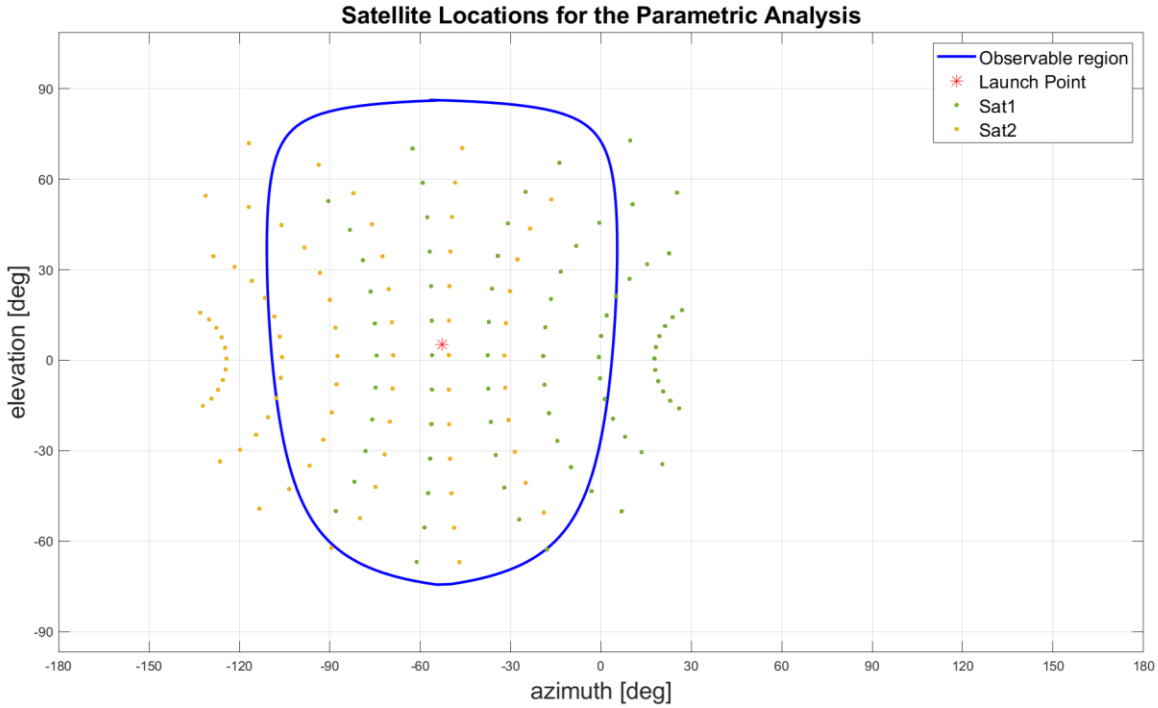

Figure 30: Satellite locations for the parametric analysis. The parametric analysis for different satellite locations has been carried out by obtaining the access area of the Vega rocket launch location, Kourou. Then, all the possible orbits (in terms of inclination and true anomaly) in which there is observability by both satellites (separated by a certain true anomaly) are obtained. It can be seen that satellites closer to the node line will have a smaller variation in terms of latitude, as expected

Orbital Parameter of Satellite 1	Value
Eccentricity, e	0.0
RAAN, Ω	$-142.775°$
Semimajor axis, a	$42164.0\ km$
Argument of periapsis, ω	0.0°
Inclination, i	$[-67.082°, 81.887°]$
True anomaly[‡‡‡], θ	$[68.249°, 160.495°]$

Table 21: Orbital parameters of satellite 1

In the same way as it was done for the in-house trajectories, the averaged NCI and RMSE for all the satellite locations will be obtained for each of the Vega trajectories. Moreover, the standard deviation will be also provided. Then, the normalized RMSE-position value (averaged for the three Vega trajectories) for each satellite location will be given in an isoline map. See Annex B.3 for the specific isoline maps of each Vega trajectory. Finally, the global performance results for the EKF and PKF will be presented and compared in order to realize which filter is in general better in terms of versatility for tracking external trajectories.

‡‡‡ Note that for a circular orbit, the true and mean anomaly are equal

Performance Comparison for External Trajectories

Parameter	Vega E		Vega NE		Vega N	
	EKF	**PKF**	**EKF**	**PKF**	**EKF**	**PKF**
Mean NCI [-]	-3.570	-2.086	-3.731	-1.957	-3.182	-1.971
Mean RMSE-Position [m]	619.827	603.946	616.020	617.954	628.062	610.053
Mean RMSE-Velocity [m/s]	153.834	134.256	150.385	134.600	153.313	134.702
Mean σ_{pos} [m]	717.523	650.102	715.393	650.639	721.703	650.581
Mean σ_{vel} [m/s]	249.458	182.701	247.861	182.788	254.239	182.770

Table 22: Detailed performance comparison for external trajectories

Table 22 shows a comparison of several performance indicators for the three Vega external trajectories tracked with the EKF and PKF. The results show that the credibility of the PKF is better than the EKF, in all cases, since the NCI value is closer to zero on the negative side. This is equivalent to the credibility results obtained for the versatility analysis with in-house trajectories. Moreover, none of the PKF tests show an NCI value greater than zero, which is desirable as the estimator stays at all times overly pessimistic (see Chapter 5). On the contrary, two test cases for the EKF filter have a value of NCI greater than zero. These two tests were carried out using the Vega N trajectory for the highest possible inclination of the observing satellites and the two highest values of azimuth (so the observing satellites are located on the furthest northeast region of the access area of Kourou). It is left for future work a more thorough analysis of the specific characteristics of this launch position with respect to the satellites' relative configuration.

Furthermore, the obtained RMSE of position and velocity for both the EKF and the PKF are very similar, although it seems that the PKF obtains slightly better results in general. The same trend can be seen for the results in standard deviation. The PKF obtains better results than the EKF. In any case, for both filters the values of standard deviation are very close for the three studied trajectories. Furthermore, it can be seen that the bias (i.e. the difference between RMSE and standard deviation) is larger for the EKF than for the PKF, unlike for the in-house trajectories. The bias of the PKF is quite similar for in-house and external trajectories, see Table 20.

Averaged Parametric Diagnosis for External Trajectories

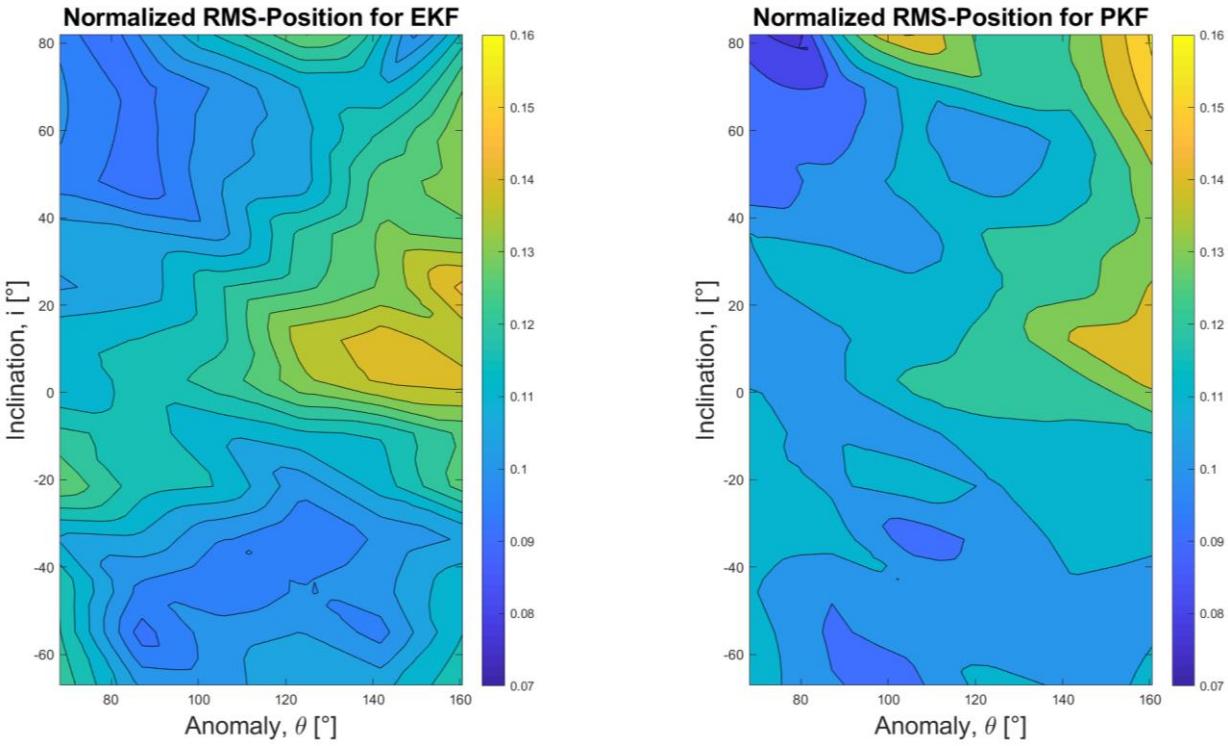

Figure 31: Launch location versatility analysis (external trajectories). Similar to the in-house trajectories, the EKF and PKF provide similar results in terms of accuracy for different observability conditions. It can be concluded that the external conditions to the filter affect in similar way to the filter output, no matter the type of filter chosen.

Figure 31 shows the normalized RMSE in position for EKF and PKF (compare with the analysis for Figure 29). This isoline map has been obtained for different inclinations and anomalies of satellite 1 (the satellite located to the west of the launch location, in the nominal configuration). The figure shows that the worst results for both PKF and EKF are obtained for satellite locations close to the furthest northeast and east regions of the access area. This comes as no surprise as from the credibility analysis it was seen that in this area the worst results were obtained (NCI value greater than zero for the EKF in two cases). In general, results for both filters are worse when satellite 1 is located to the east of Kourou and tend to improve for tests in which the satellite location is moved towards west. This is actually the opposite to the results obtained for the parametric launch diagnosis of in-house trajectories: worst results were obtained when satellite 1 was to the west of the launch point. Finding an answer to this issue will be left for further work since this is not the main goal of the thesis. Therefore, it can be concluded the PKF is less sensitive than the EKF to variations of the configuration between satellites and launch vehicle.

The global results obtained from the analysis of external trajectories are provided in Table 23. The results show that the PKF overcomes the EKF for all the performance indicators: credibility, accuracy, and precision. In any case, the greatest differences are seen in credibility, in the same way as it was seen for in-house trajectories. One of the major differences with respect to the in-house trajectories is that the standard deviation (computed

from the error covariance matrix) of the EKF is much worse for external trajectories in comparison with the PKF. This has an effect in bias: the greater difference between the RMSE and the standard deviation for the EKF implies a larger bias of this filter when tracking external trajectories, unlike for in-house trajectories, in which the bias of the EKF was the smallest. The bias for external trajectories of the PKF remains quite similar to the one obtained for in-house trajectories, supporting the hypothesis that the PKF is less sensitive than the EKF to changes in the trajectory that is being tracked.

Parameter	**EKF**	**PKF**
Mean NCI [-]	-3.4945	-2.0046
Mean RMSE-Position [m]	621.3030	610.6510
Mean RMSE-Velocity [m/s]	152.5107	134.5194
Mean σ_{pos} [m]	718.2065	650.4407
Mean σ_{vel} [m/s]	250.5194	182.7530

Table 23: Overall comparison of EKF and PKF - parametric external trajectories

All things considered, from the analysis carried out in this chapter it can be said that the PKF has a better performance than the EKF in terms of the versatility. Even for in-house trajectories, in which the motion model of the tracked vehicle is analogous to the EKF dynamic model, the PKF seemed to obtain in general a better credibility, without providing much worse results on precision and accuracy compared to the EKF. Moreover, the changes in launch azimuth and location affect less to PKF than to EKF. Finally, the advantages of using PKF for tracking external trajectories over the EKF have also been analyzed. All the performance indicators displayed better global results for the PKF.

Chapter 7

Conclusions and Further Work

As it was introduced in Chapter 1, the objective of this thesis was to develop a filtering tool capable of achieving tracking for any possible launch scenario with available observability. The main motivation to perform this thesis was the need to improve the output of a previously developed tracking tool, based on an extended Kalman filter, called EKF-Tool. The limitations of this tool were the necessity to modify its input parameters – more specifically, the level of process noise – before starting the tracking process in order to achieve filter convergence. It was seen that the (sub)optimal value of process noise greatly depends on the type of trajectory being tracked. Nevertheless, the type of trajectory cannot be known beforehand so this approach of tuning process noise is not effective in some cases. Moreover, a large value of process noise translates into worse results in terms of filter accuracy, precision, and consistency, among others. Therefore, an alternative tracking filter tool, PKF-Tool that could overcome these limitations was sought.

This PKF-Tool is developed based on one idea: finding a simpler dynamic model that allows a more complex addition of process noise. In this way, linear dynamic models based on stochastic processes were used so a polynomial Kalman filter could be used without the need for linearization and numerical integration that the EKF-Tool uses. Moreover, the addition of process noise was done based on the information obtained from the state transition matrix of the dynamic model, unlike the EKF-Tool which used a simpler scheme to include process noise in its algorithm. Using information from the state transition matrix is suitable to achieve high filter consistency so any possible launch scenario can be tracked without the need of modifying the inputs of the PKF-Tool.

The developed PKF-Tool and the EKF-Tool were tested against a variety of different launch scenarios using several performance indicators. The performance characteristics measured, and the tools used for measuring them were the following:

Performance Indicator	Performance Tool
Stability	Root Mean Square Error – RMSE
Consistency	Normalized Innovation Square – NIS test
Credibility	Non-credibility index – NCI
Accuracy	Root Mean Square Error – RMSE
Precision	Volume of Error Ellipsoid
Versatility	Parametric analysis of launch scenarios

Table 24: Summary of performance indicators and tools

These indicators were used to test overall performance of the filters. First, one specific in-house trajectory generated by the Simulation Environment and one specific external

trajectory were used as reference in order to compare each of the performance indicators with each filtering tool. Then, a parametric analysis of different types of in-house and external trajectories with different observability conditions such as launch point location and launch azimuth angle was carried out. This parametric analysis was performed in order to study the versatility of each of the filters. That is, to measure the performance of each filter for the others indicators without the need of modifying their input parameters. Considering the whole analysis, the following Table 25 summarizes the conclusions achieved.

Performance Analysis	**EKF-Tool**	**PKF-Tool**
Precision	Medium	Medium
Accuracy	High	Medium
Credibility	Low	High
Versatility	Low	High

Table 25: Conclusions of filter performance analysis

In terms of precision, both of the tools perform differently. On one hand, the PKF obtains the similar precision results (volume of the error ellipsoid in position) for both in-house and external trajectories. On the other hand, the EKF performs much better than the PKF for in-house trajectories but much worse for external trajectories. Therefore, it cannot be concluded which one has the best precision performance. Regarding accuracy, the EKF obtains relatively good results for in-house trajectories compared with the PKF. The accuracy of the PKF remains similar for any type of trajectory and about at the same level as the EKF for external trajectories. Therefore, it can be concluded that the EKF has an overall better performance than the PKF for accuracy, although the results for any of them are in accordance with what it is expected by the stakeholders (ADS).

The largest difference in performance between EKF and PKF is found in terms of credibility (consistency) and versatility. In this way, it was seen that the credibility of the EKF is much worse than the PKF for values of process noise that allow filter convergence. Moreover, the PKF credibility remains constant using any amount of process noise (within nominal ranges) for both in-house and external trajectories whereas EKF credibility is lost when the value of process noise is too low or too high. Finally, after performing a parametric analysis with over 4000 different launch scenarios, it was concluded that the versatility of the PKF is much higher than the EKF. This large difference in versatility is not only seen by how relatively low overall performance shows the EKF for any external trajectory but also by the influence that cause different launch scenarios in the filter output.

All things considered, it can be concluded that the developed filtering tool within the context of this thesis, the PKF-Tool, shows an overall better performance with the defined indicators than the previously developed tool, the EKF-Tool, at Airbus. In this way, it can be said that the thesis objective has been successfully achieved. An important remark is the effect of process noise on each of the analyzed filters. Varying process noise to a lower or higher level than the (sub)optimally obtained one affects both to EKF's stability and credibility. On the other hand, varying the level of process noise has a smaller effect in the performance of the

PKF. In other words, the PKF is more robust than the EKF to variations of process noise. Nevertheless, there are several issues and limitations of the PKF-Tool that should be further studied in order to improve its usefulness. Moreover, further lines of work can be envisaged in order to increase the EKF performance so its limitations can be overcome.

Further Work

The main issue that arises regarding the EKF-Tool resides in the way process noise is managed. During the thesis, it was discussed that the relatively simple method of process noise addition by increasing the propagated state error covariance matrix with an input factor caused most of the main issues of the EKF of stability and consistency (credibility). In this way, it is proposed for future work to investigate alternative methods of addition of process noise to the EKF. A potentially suitable method is to model process noise using the state transition matrix of the EKF, which can be obtained after linearization. This method is analogous to the approach used to manage process noise in the PKF. Nevertheless, since the gravity turn dynamic model of the EKF is highly nonlinear, unlike the stochastic models of the PKF, it might be possible that this approach does not work effectively. Therefore, a second approach is also proposed for handling EKF process noise [56] [57]. This approach consists on adaptive modification of process noise during the tracking process by tuning its value in real-time. In this way, in Chapter 5 an adaptive method was introduced which was based on the Normalized Innovation Square (NIS). Nevertheless, alternative and more powerful methods could be rehearsed in line with this idea.

Regarding the PKF-Tool, the only input parameter to be tuned is the power spectral density PSD that is used to determine the level of process noise. A manual approach of optimization of the PSD was proposed in this thesis. Nevertheless, alternative and more efficient methods can be sought such as the analytical approach using the Wiener-Khintchine-Einstein Theorem [33]. In any case, such methods seem to offer small estimation improvement since it was seen that the PKF performance is not very sensitive to variation of its input parameters and the filter itself is probably governed in a higher order of magnitude by the effect of the dynamic models based on stochastic processes.

Furthermore, more general lines of action for future work can also be proposed. In this way, it could be investigated whether the developed filter would work and what its performance would be if only one sensor (mono-tracking) was used to obtain observations. Furthermore, the possibility of including sensor bias should also be investigated, as most of the real sensors have a component of bias. Moreover, it is left for further work the modification of the Simulation Environment in order to allow asynchronous sensors; that is, observations being taken at different times by each of the observing satellites. Finally, potential applications of the PKF versions developed could also be studied such as the ones explained for the PKFD in Section 4.4, in which different dynamic models could be implemented for each direction of the flight.

Bibliography

[1] O. Montenbruck and E. Gill, Satellite Orbits: Models, Methods, and Applications, Wessling, Germany: Springer, 2001.

[2] Krause S.S, Avoiding Mid-air Collisions, TAB Books, 1995.

[3] V. Aycart, "Tracking Ballistic Vehicles during Boost. A Review of Target-Tracking Techniques," Faculty of Aerospace Engineering , Delft, 2019.

[4] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, vol. 46, no. 1, pp. 96-119, 1 2010.

[5] Y. Chen, S.-L. Wen and Z. Cheng, "Method for tracking ballistic missile on boost phase," *Xitong Fangzhen Xuebao / Journal of System Simulation*, vol. 24, no. 5, pp. 1063-1067, 2012.

[6] J. Ralph, M. Smith and J. Heather, "Identification of missile guidance laws for missile warning systems applications," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2006.

[7] H. Song and Y. Han, "Comparison of space launch vehicle tracking using different types of multiple models," in *FUSION 2016 - 19th International Conference on Information Fusion, Proceedings*, 2016.

[8] R. Tharmarasa, T. Kirubarajan, N. Nandakumaran and Y. Bar-Shalom, "Profile-Free Launch Point Estimation for Ballistic Targets Using Passive Sensors," *Journal of Advances in Information Fusion*, vol. 7, no. 1, 2012.

[9] P. Zarchan, "Boost-Phase Filtering Options: Is Simpler Better?," *JOURNAL OF GUIDANCE CONTROL AND DYNAMICS*, vol. 33, no. 6, pp. 1724-1731, 11 2010.

[10] B. Tapley, B. Schutz and G. Born, Statistical Orbit Determination, Elsevier, 2004.

[11] N. Danis, "Space-Based Tactical Ballistic Missile Launch Parameter Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 2, pp. 412-424, 1993.

[12] X. Li and V. Jilkov, "A survey of maneuvering target tracking - Part III: Measurement models," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2001.

[13] M. Sankowski, "Continuous-discrete estimation for tracking ballistic missiles in air-surveillance radar," *IET Radar, Sonar and Navigation*, vol. 5, no. 9, pp. 978-986, 2011.

[14] K.-H. Keil and W. Hupfer, "Simulation of signal and data processing for a pair of GEO IR sensors," in *SIGNAL AND DATA PROCESSING OF SMALL TARGETS 2007*, 2007.

[15] Y. Li, T. Kirubarajan, Y. Bar-Shalom and M. Yeddanapudi, "Trajectory and launch point estimation for ballistic missiles from boost phase LOS measurements," *IEEE Aerospace Applications Conference Proceedings*, vol. 4, pp. 425-442, 1999.

[16] H. Wittenberg, B. A. C. Ambrosius, M. C. Naeije and Mooij E, "AE4870A Lecture Notes: Rocket Motion," Faculty of Aerospace Engineering , Delft, 2014.

[17] P. Zarchan and H. Musoff, Fundamentals of Kalman FIltering, Cambridge, Massachusetts: American Institute of Aeronautics and Astronautics, 2000.

[18] G. Goodwin and K. Sang Sin, Adaptive Filtering Prediction and Control, Dover Publications, 1984.

[19] S. M. Challa, Mark R.Mušicki, Darko Evans and Robin J., Fundamentals of Object

Tracking, Cambridge University Press, 2011, pp. 62-102.
[20] A. Gelb, Applied Optimal Estimation, Cambridge, MA: The MIT Press, 1974.
[21] A. J. Haug, Bayesian Estimation and Tracking : A Practical Guide, Wiley, 2012.
[22] L. Shampine and M. K. Gordon, Computer solution of ordinary differential equations : the initial value problem, San Francisco, CA: Freeman, 1975.
[23] R. Bucy and P. Joseph, Filtering for Stochastic Processing with Applications to Guidance, 2 ed., AMS Chelsea Publishing, 2005.
[24] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS*, vol. 39, no. 4, pp. 1333-1364, 10 2003.
[25] Y. Wang, *Wiener Process (Brownian Motion)*, Singapore: Nanyang Technological University, 2019.
[26] J. Van Zandt, "Boost phase tracking with an unscented filter," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2002.
[27] J. Cornelisse, H. Schoyer and K. Wakker, Rocket propulsion and spaceflight dynamics, London: Pitman, 1979.
[28] E. D. Kaplan and C. J. Hegarty, Understanding GPS/GNSS - Principles and Applications, 3rd ed., Artech House, 2017.
[29] D. A. Vallado, Fundamentals of Astrodynamics and Applications, 4 ed., Microcosm Press, 2013.
[30] V. P. Jilkov, X. R. Li and J. Ru, "Modeling ballistic target motion during boost for tracking," in *SIGNAL AND DATA PROCESSING OF SMALL TARGETS 2007*, 2007.
[31] M. Hough, F. Daum and J. Huang, "Nonlinear recursive estimation of boost trajectories, including batch initialization and burnout estimation," in *Collection of Technical Papers - AIAA/AAS Astrodynamics Specialist Conference*, 2004.
[32] M. E. Hough, "Nonlinear recursive filter for boost trajectories," *JOURNAL OF GUIDANCE CONTROL AND DYNAMICS*, vol. 24, no. 5, pp. 991-997, 9 2001.
[33] S. Millers and D. Childers, Probability and random processes, Academic Press, 2012.
[34] Y. Bar-Shalom, X. Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation, 2001.
[35] I. Klein, Y. Lipman and Y. Bar-Shalom, "Asynchronous Passive Multisensor System Observability with Unknown Sensor Position," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 369-375, 2018.
[36] T. Karvonen, "Stability of linear and non-linear Kalman filters," 2014.
[37] E. Schrama, "Lecture on Planetary Sciences and Satellite Orbit Determination," Delft, 2017.
[38] B. A. Walther and J. L. Moore, "The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature review of estimator performance," *Ecography*, vol. 28, no. 6, pp. 815-829, 2005.
[39] S. Houts, S. Dektor and Rock S, "A ROBUST FRAMEWORK FOR FAILURE DETECTION AND RECOVERY FOR TERRAIN-RELATIVE NAVIGATION," 2013.
[40] M. G. S. Bruno and A. Pavlov, "Improved sequential Monte Carlo filtering for ballistic target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 1103-1108, 2005.
[41] P. Matisko and V. Havlena, "Noise covariances estimation for Kalman filter tuning," in

IFAC Proceedings Volumes (IFAC-PapersOnline), 2010.

[42] T. D. Powell, "Automated tuning of an extended Kalman filter using the downhill simplex algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 5, pp. 901-908, 2002.

[43] P. Ivanov, S. Ali-Loytty and R. Piche, "Evaluating the consistency of estimation," in *Proceedings of 2014 International Conference on Localization and GNSS, ICL-GNSS 2014*, 2014.

[44] R. Piche, "Online tests of Kalman Fiter consistency," Tampere, 2014.

[45] X. R. Li and Z. Zhao, "Measuring estimator's credibility: Noncredibility index," in *2006 9th International Conference on Information Fusion, FUSION*, 2006.

[46] I. Reid, *Estimation II - Introduction to Target Tracking*, Oxford: Oxford University, 2001.

[47] V. P. Panakkal and R. Velmurugan, "An improved measurement model for target tracking under measurement origin uncertainty," *Journal of Advances in Information Fusion*, vol. 10, no. 2, pp. 143-162, 2015.

[48] M. Scalzo, G. Horvath, E. Jones, A. Bubalo, M. Alford, R. Niu and P. K. Varshney, "Adaptive filtering for single target tracking," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2009.

[49] X. R. Li, "Evaluation of estimation algorithms part I: Incomprehensive measures of performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1340-1358, 2006.

[50] R. Mittal, "Lecture 7: Positive Semidefinite Matrices," in *Linear algebraic tools in TCS*, 2018.

[51] K. Saho, "Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design," in *Kalman Filters - Theory for Advanced Applications*, InTech, 2018.

[52] B. Wang, W. Shi and Z. Miao, "Confidence Analysis of Standard Deviational Ellipse and Its Extension into Higher Dimensional Euclidean Space," *PLoS ONE* , 2015.

[53] S. B. Pope, *Algorithms for Ellipsoids*, Ithaca: Cornell University, 2008.

[54] A. J. Wilson, "Volume of n-dimensional ellipsoid," *Sciencia Acta Xaveriana*, vol. 1, no. 1, pp. 101-106, 2009.

[55] Wertz JR, Orbit and Constellation Design and Management, The Space Technology Library, 2009.

[56] A. K. Anilkumar, M. R. Ananthasayanam, N. K. Philip and P. V. Subba Rao, "Adaptive kalman filter tuning and controlled random search for mmle with process noise," in *24th Atmospheric Flight Mechanics Conference*, 1999.

[57] A. G. Gelen and A. Atasoy, "A New Method for Kalman Filter Tuning," in *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, 2019.

Annex A: Measurement Noise in Cartesian Frame

This annex contains the derivation of the transformation equations used to convert measurement noise from spherical coordinates (as it is provided by the stakeholder ADS) to Cartesian coordinates. The measurement noise in Cartesian coordinates is used by the PKF Decoupled Cartesian (PKFD) and PKF Coupled Cartesian (PKFCC). The procedure to obtain the equations of measurement noise for the PKFCC is very similar to the PKFD, albeit more complex, as the effect of the correlated terms is taken into account.

The derivations of this annex are based on the work carried out by Zarchan [17]. The results have been obtained using the *symbolic* MATLAB Toolbox. The starting point for the derivation of the measurement noise is the equations used to obtain the triangulated measurements in Cartesian frame from the spherical observations of the sensors located in the GEO satellites. These equations provide the necessary relations to obtain the frame transformation spherical-to-Cartesian. The transformation equations are provided in Chapter 3 – Equations (28) to (34) – and are summarized as follows.

$$\Delta \boldsymbol{r}_{sat} = \boldsymbol{r}_{sat}^{(2)} - \boldsymbol{r}_{sat}^{(1)} \tag{A.1}$$

$$\boldsymbol{u}^{(i)} = \left[\cos\left(\varepsilon^{(i)}\right) \cdot \cos\left(\theta^{(i)}\right) \quad \cos\left(\varepsilon^{(i)}\right) \cdot \sin\left(\theta^{(i)}\right) \quad \sin\left(\varepsilon^{(i)}\right)\right]^{T} \tag{A.2}$$

$$\left|\Delta \boldsymbol{r}^{(1)}\right| = \frac{\left(\boldsymbol{u}^{(1),T} \Delta \boldsymbol{r}_{sat} - \boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)} \boldsymbol{u}^{(2),T} \Delta \boldsymbol{r}_{sat}\right)}{1 - (\boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)})^2} \tag{A.3}$$

$$\left|\Delta \boldsymbol{r}^{(2)}\right| = \frac{\left(\boldsymbol{u}^{(2),T} \Delta \boldsymbol{r}_{sat} - \boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)} \boldsymbol{u}^{(1),T} \Delta \boldsymbol{r}_{sat}\right)}{1 - (\boldsymbol{u}^{(1),T} \boldsymbol{u}^{(2)})^2} \tag{A.4}$$

$$\boldsymbol{r}_{meas}^{(1)} = \boldsymbol{r}^{(1)} + \left|\Delta \boldsymbol{r}^{(1)}\right| \boldsymbol{u}^{(1)} \tag{A.5}$$

$$\boldsymbol{r}_{meas}^{(2)} = \boldsymbol{r}^{(2)} + \left|\Delta \boldsymbol{r}^{(2)}\right| \boldsymbol{u}^{(2)} \tag{A.6}$$

$$\boldsymbol{r} = \frac{\boldsymbol{r}_{meas}^{(1)} + \boldsymbol{r}_{meas}^{(2)}}{2} \tag{A.7}$$

In the Cartesian frame, the model of the real world is linear but the observations are nonlinear as seen from Equations A.1 to A.7. The purpose of this derivation is to obtain a relation between measurement noise in spherical frame σ_n (note the reader that the value of noise is assumed to be the same for azimuth and elevation coordinates throughout this text) and measurement noise in the Cartesian frame σ_x, σ_y, σ_z. The goal is to find this measurement noise relation by using the aforementioned equations and by computing the expectation.

It can be seen that Equation A.7, which is provided in $x, y,$ and z coordinates, is a function of the azimuth θ and elevation ε angles of each sensor. Thus, the Cartesian location of the launch vehicle can be rewritten as

$$\boldsymbol{r} = f(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2) \tag{A.8}$$

Using Equation A.8, the total differential can be obtained from calculus as follows

$$\Delta \mathrm{x} = \frac{\partial \mathrm{x}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{x}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{x}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{x}}{\partial \varepsilon_1}\Delta\varepsilon_2 \tag{A.9}$$

$$\Delta \mathrm{y} = \frac{\partial \mathrm{y}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{y}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{y}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{y}}{\partial \varepsilon_1}\Delta\varepsilon_2 \tag{A.10}$$

$$\Delta \mathrm{z} = \frac{\partial \mathrm{z}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{z}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{z}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{z}}{\partial \varepsilon_1}\Delta\varepsilon_2 \tag{A.11}$$

Equations A.9 to A.11 are squared to be able to obtain expectations.

$$\Delta \mathrm{x}^2 = \left(\frac{\partial \mathrm{x}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{x}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{x}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{x}}{\partial \varepsilon_1}\Delta\varepsilon_2\right)^2 \tag{A.12}$$

$$\Delta \mathrm{y}^2 = \left(\frac{\partial \mathrm{y}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{y}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{y}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{y}}{\partial \varepsilon_1}\Delta\varepsilon_2\right)^2 \tag{A.13}$$

$$\Delta \mathrm{z}^2 = \left(\frac{\partial \mathrm{z}}{\partial \theta_1}\Delta\theta_1 + \frac{\partial \mathrm{z}}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial \mathrm{z}}{\partial \theta_2}\Delta\theta_2 + \frac{\partial \mathrm{z}}{\partial \varepsilon_1}\Delta\varepsilon_2\right)^2 \tag{A.14}$$

Equations A.12 to A.14 provide the relations between spherical and Cartesian coordinates in the decoupled case, so a diagonal measurement noise matrix is obtained, as it is shown in Section A.1. Nevertheless, for the coupled case it is also possible to find the correlated terms $\Delta x \Delta y$, $\Delta x \Delta z$, and $\Delta z \Delta y$ by multiplying Equations A.9 to A.11 accordingly in order to obtain the relation between the non-diagonal elements of the measurement noise matrix, as it is shown in Section A.2.

A.1. Derivation of Measurement Noise for the PKFD

The right-hand side of Equations A.12 to A.14 contains the measurement noise in the spherical frame, since measurement noise is defined as

$$\mathrm{E}[\Delta\theta_i^2] = \sigma_n^2 \tag{A.15}$$

for the azimuth measurement for satellites $i = 1, 2$ and

$$\mathrm{E}[\Delta\varepsilon_i^2] = \sigma_n^2 \tag{A.16}$$

for the elevation measurement for satellites $i = 1, 2$. Note that the expectation of the terms of the right-hand side of Equations A.12 to A.14 that contain the multiplications $\Delta\theta_i\Delta\varepsilon_i$ – being $i = 1, 2$ – are equal to zero, since there is no correlation in measurement noise between each satellite and each coordinate. On the other hand, the left-hand side of Equations A.12 to A.14 contains the measurement noise in Cartesian coordinates:

$$\mathrm{E}[\Delta x^2] = \sigma_x^2 \quad \mathrm{E}[\Delta y^2] = \sigma_y^2 \quad \mathrm{E}[\Delta x^2] = \sigma_y^2 \tag{A.17}$$

Thus, after taking expectations, Equations A.12 to A.14 can be rewritten as

$$\sigma_x^2 = g_x(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)\sigma_n \tag{A.18}$$

$$\sigma_y^2 = g_y(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)\sigma_n \tag{A.19}$$

$$\sigma_z^2 = g_z(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)\sigma_n \tag{A.20}$$

Finally, the measurement noise matrix in Cartesian frame (decoupled case) is written as

$$\mathbf{R_{xyz}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_z \end{bmatrix} \sigma_n^2 \tag{A.21}$$

Note the reader that the computation of Expressions $g_x(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)$, $g_y(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)$, and $g_z(\theta_1, \varepsilon_1, \theta_2, \varepsilon_2)$ have been directly computed using the MATLAB Symbolic toolbox from Equations A.1 to A.7. The value of these expressions is omitted from this text for clarity's sake, as each of these expressions contain more than 200 terms (additions and subtractions).

A.2. Derivation of Measurement Noise for the PKFCC

For the coupled case, the derivation performed in Section A.1 applies to the diagonal terms of the Cartesian measurement noise matrix. Starting from Equations A.9 to A.11, the following correlated terms are obtained

$$\Delta x\Delta y = \left(\frac{\partial x}{\partial \theta_1}\Delta\theta_1 + \frac{\partial x}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial x}{\partial \theta_2}\Delta\theta_2 + \frac{\partial x}{\partial \varepsilon_1}\Delta\varepsilon_2\right)\left(\frac{\partial y}{\partial \theta_1}\Delta\theta_1 + \frac{\partial y}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial y}{\partial \theta_2}\Delta\theta_2 + \frac{\partial y}{\partial \varepsilon_1}\Delta\varepsilon_2\right) \tag{A.22}$$

$$\Delta y\Delta z = \left(\frac{\partial y}{\partial \theta_1}\Delta\theta_1 + \frac{\partial y}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial y}{\partial \theta_2}\Delta\theta_2 + \frac{\partial y}{\partial \varepsilon_1}\Delta\varepsilon_2\right)\left(\frac{\partial z}{\partial \theta_1}\Delta\theta_1 + \frac{\partial z}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial z}{\partial \theta_2}\Delta\theta_2 + \frac{\partial z}{\partial \varepsilon_1}\Delta\varepsilon_2\right) \quad \text{A.23}$$

$$\Delta x\Delta z = \left(\frac{\partial x}{\partial \theta_1}\Delta\theta_1 + \frac{\partial x}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial x}{\partial \theta_2}\Delta\theta_2 + \frac{\partial x}{\partial \varepsilon_1}\Delta\varepsilon_2\right)\left(\frac{\partial z}{\partial \theta_1}\Delta\theta_1 + \frac{\partial z}{\partial \varepsilon_1}\Delta\varepsilon_1 + \frac{\partial z}{\partial \theta_2}\Delta\theta_2 + \frac{\partial z}{\partial \varepsilon_1}\Delta\varepsilon_2\right) \quad \text{A.24}$$

Taking expectations as it was done in the previous section, expressions of the following form are obtained

$$E[\Delta x\Delta y] = h_1(\theta_1)E[\Delta\theta_1^2] + h_2(\theta_2)E[\Delta\theta_2^2] + h_3(\varepsilon_1)E[\Delta\varepsilon_1^2] + h_4(\varepsilon_2)E[\Delta\varepsilon_2^2] \quad \text{A.25}$$

$$E[\Delta z\Delta y] = h_5(\theta_1)E[\Delta\theta_1^2] + h_6(\theta_2)E[\Delta\theta_2^2] + h_7(\varepsilon_1)E[\Delta\varepsilon_1^2] + h_8(\varepsilon_2)E[\Delta\varepsilon_2^2] \quad \text{A.26}$$

$$E[\Delta x\Delta z] = h_9(\theta_1)E[\Delta\theta_1^2] + h_{10}(\theta_2)E[\Delta\theta_2^2] + h_{11}(\varepsilon_1)E[\Delta\varepsilon_1^2] + h_{12}(\varepsilon_2)E[\Delta\varepsilon_2^2] \quad \text{A.27}$$

Since the expectations – or measurement noise – of the measurements are equal among them (Expressions A.15 and A.16), the above equations can be rewritten as

$$E[\Delta x\Delta y] = \sigma_{xy} = [h_1(\theta_1) + h_2(\theta_2) + h_3(\varepsilon_1) + h_4(\varepsilon_2)]\sigma_n^2 = h_{xy}\sigma_n^2 \quad \text{A.28}$$

$$E[\Delta z\Delta y] = \sigma_{yz} = [h_5(\theta_1) + h_6(\theta_2) + h_7(\varepsilon_1) + h_8(\varepsilon_2)]\sigma_n^2 = h_{yz}\sigma_n^2 \quad \text{A.29}$$

$$E[\Delta x\Delta z] = \sigma_{xz} = [h_9(\theta_1) + h_{10}(\theta_2) + h_{11}(\varepsilon_1) + h_{12}(\varepsilon_2)]\sigma_n^2 = h_{xz}\sigma_n^2 \quad \text{A.30}$$

The above expectations correspond to the non-diagonal terms of the measurement noise matrix of the coupled Cartesian PKF.

$$\mathbf{R_{xyz}} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} g_x & h_{xy} & h_{xz} \\ h_{xy} & g_y & h_{yz} \\ h_{xz} & h_{yz} & g_z \end{bmatrix}\sigma_n^2 \quad \text{A.31}$$

Annex B: Performance Analysis Further Results

In this annex, additional results regarding the different PKF versions are provided in order to complement the information gathered in Chapter 5 and Chapter 6. This annex, thus, will not provide an exhaustive analysis of the PKF versions, as this was already performed in the main text of the thesis. The purpose of this annex is to provide the reader with a brief performance comparison of the different PKF versions, including the one developed using Markov processes, together with some additional results of the parametric performance analysis.

B.1. Comparison of PKF Versions

For each PKF filter (excluding the Markov process based PKF), the NCI index and RMSE for the three values of process noise using external trajectories are provided in this section. The reason to only show the results for the external trajectories is that the performance of the PKF versions is very similar when in-house or external trajectories are used. Therefore, for clarity's sake the results with in-house trajectories have been omitted. An exhaustive explanation of the behavior of the PKF in terms of performance can be found throughout the main text (Chapter 5 and Chapter 6). Next table summarizes the three values (levels) of process noise that were used in the main text and for the performance comparison of the PKF versions.

Process Noise	**PKF**	
Low (1)	$[10^{-4}]$ in x, y, z (full thrust)	$[1, 0.5, 1] \cdot 10^{-3}$ in x, y, z (tail-off thrust)
Medium (2)	$[5 \cdot 10^{-4}]$ in x, y, z (full thrust)	$[7, 7, 7] \cdot 10^{-3}$ in x, y, z (tail-off thrust)
High (3)	$[0.01]$ in x, y, z (full thrust)	$[0.1, 0.1, 0.1]$ in x, y, z (tail-off thrust)

Table 26: Summary of different process noise levels used for each PKF version

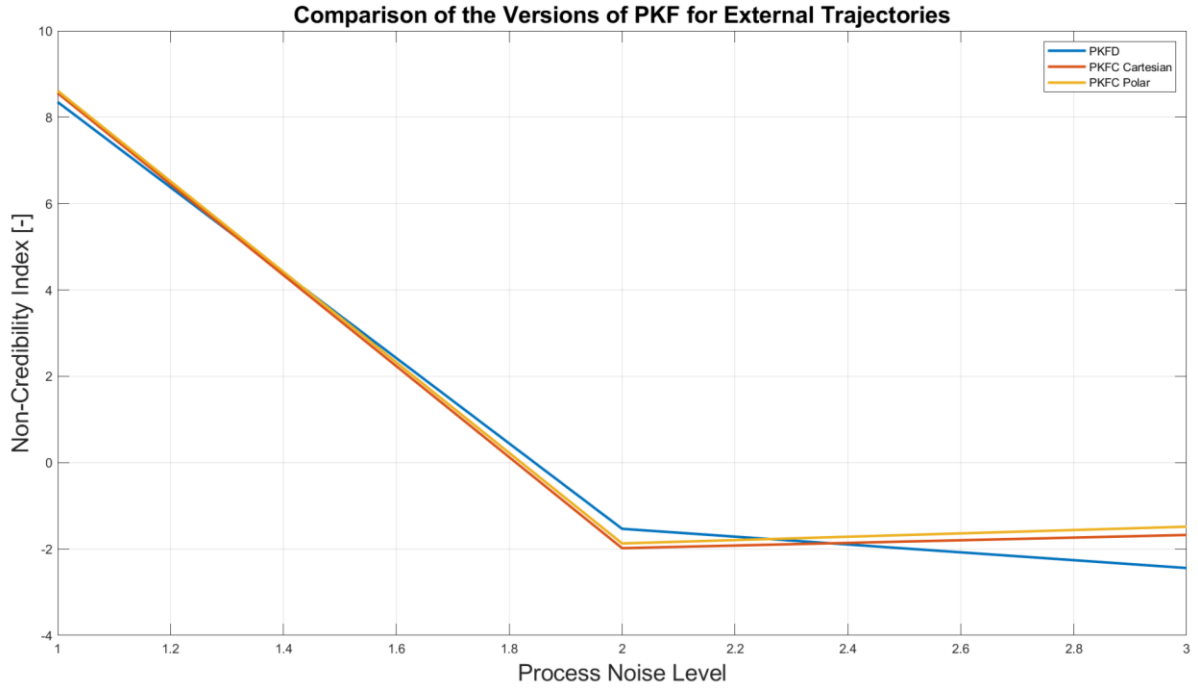

Figure 32: The value of the NCI index vs the amount of process noise (levels 1 to 3) is shown for each of the PKF versions based on Wiener processes. It can be seen that the behavior of the three filters in terms of NCI is very similar, although PKFD seems to show the best results for the optimal case of process noise (level 2), rapidly worsening for higher levels of process noise in comparison to the other two PKF versions. Both coupled filters (Cartesian and polar) show a very similar behavior, despite using each a different reference frame in the measurement model. This concludes that the effect of the reference frame chosen for the measurement model is smaller in comparison with the effect of process noise.

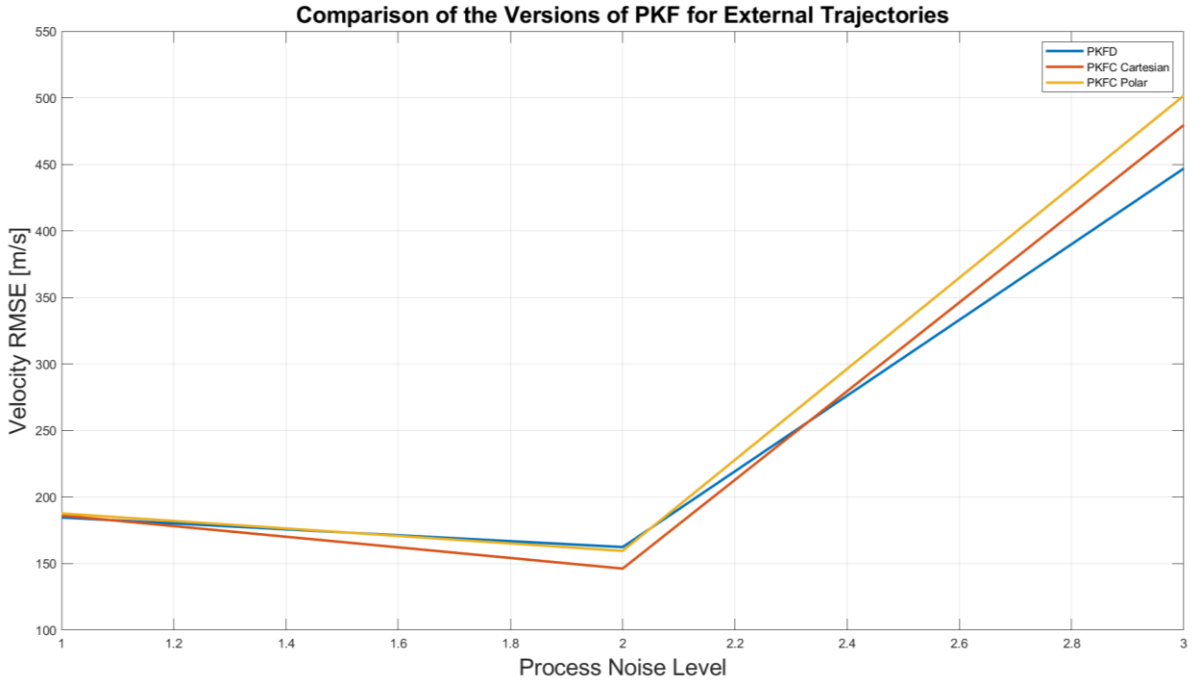

Figure 33: The RMSE in velocity (the behavior in RMSE for position is similar so it is omitted for clarity's sake) vs the amount of process noise (levels 1 to 3) is shown for each of the PKF versions based on Wiener processes. The best accuracy results are obtained with the PKFC Cartesian in general, although the PKFD shows better behavior for higher levels of process noise. The PKFC Polar displays the worst results for all cases. In any case, the difference in results among filters is small in relative terms in comparison to the differences that are obtained when process noise is increased or reduced.

B.2. Markov Models

In this section, a comparison of the results obtained using the Markov process-based filter (PKFM) compared with the Wiener process-based filters is provided. More specifically, the results of the PKFD are compared, as it was seen from the previous section and the main text that the Wiener-based filters have relatively similar results in terms of performance. This comparison is performed with the three levels of process noise used in the previous section.

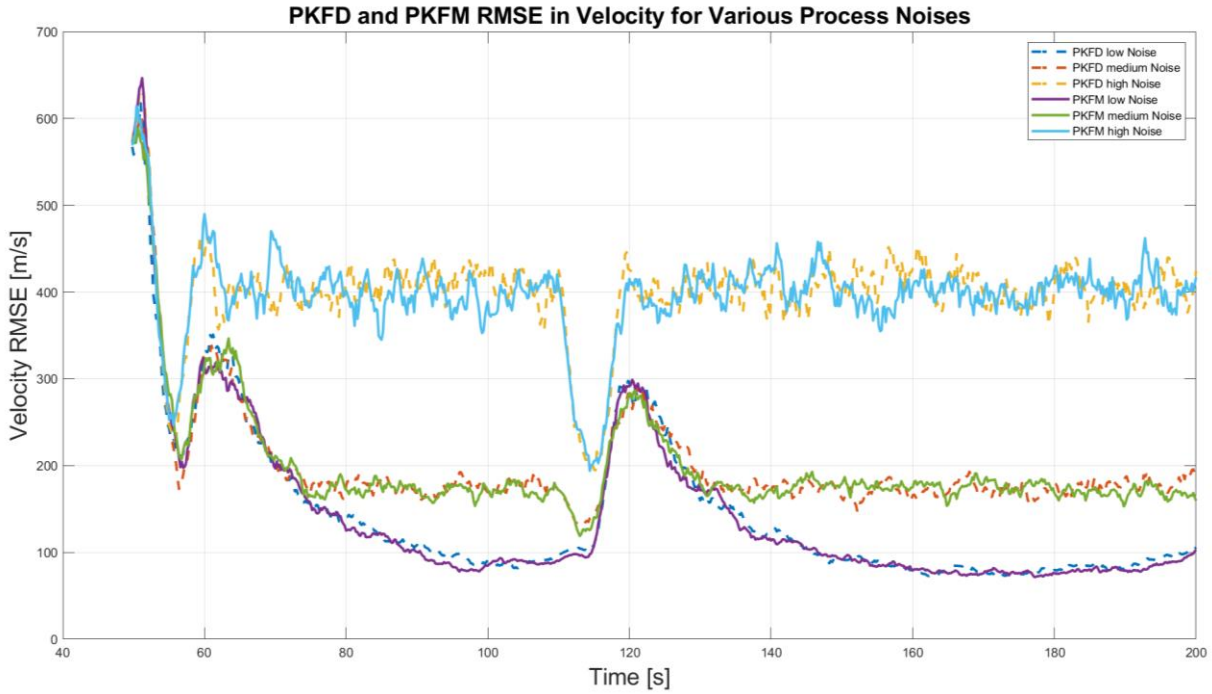

Figure 34: RMSE velocity results with in-house trajectories for PKFM and PKFD for three levels of process noise using in-house trajectories. It can be seen that the results of the PKFM and the PKFD are almost the same with respect to time for the three levels of noise. As it was discussed in the main text, the PKFM needs an additional input parameter (correlation time constant). The results provided in this figure have been obtained after manually tuning process noise together with the correlation time constant for the PKFM. It is observed that for a certain value of the correlation time constant, the PKFM is capable of obtaining results in accuracy of the same level that the Wiener-based PKF.

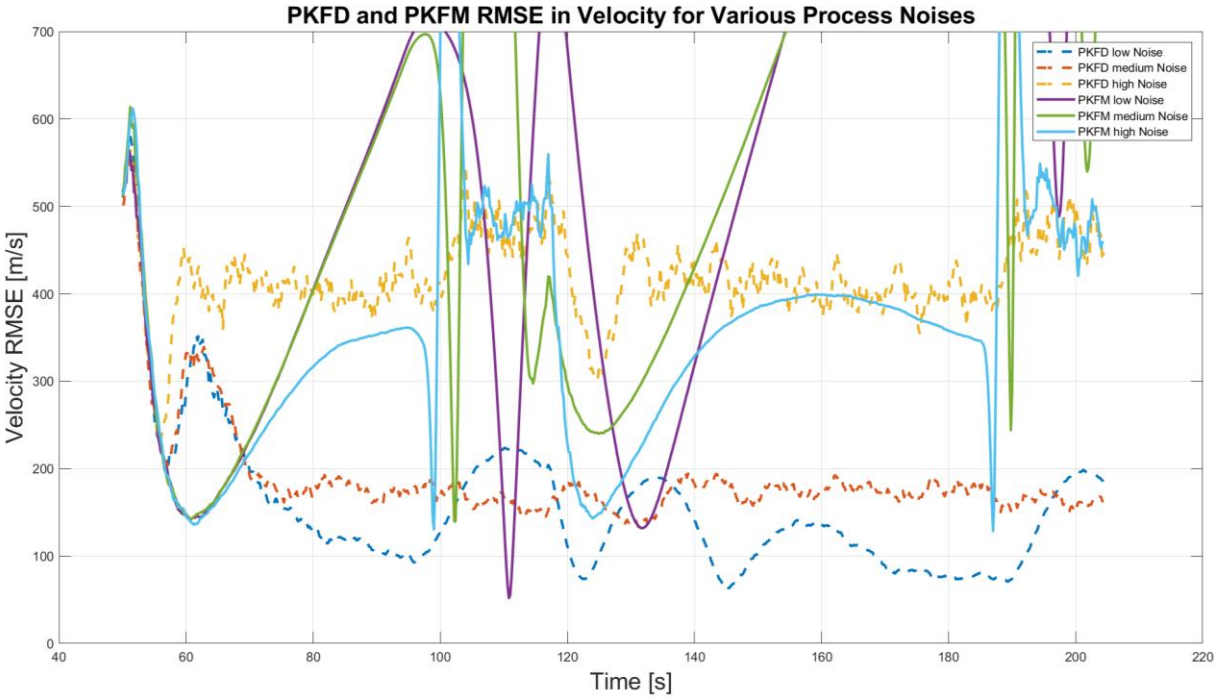

Figure 35: RMSE velocity results with external trajectories for PKFM and PKFD for three levels of process noise using external trajectories. The results shown in this figure have been obtained using the same value of process noise and correlation time constant used for the in-house trajectories of the previous figure. It can be concluded thus, that the correlation time constant is greatly dependent on the type of tracked trajectory. In this way, the PKFM must be tuned before tracking any new launching vehicle with unknown trajectory, thus defeating the purpose of the simplicity of the PKF. Therefore, it can be concluded that the PKFM is not capable of achieving the same versatility (the capability of not modifying filter inputs for new trajectories) achieved by the PKF based in Wiener processes, as divergence is observed when an alternative trajectory is to be tracked.

B.3. Parametric Diagnosis: Vega Trajectories

In this section, the results of the parametric analysis using external (Vega) trajectories are presented, in analogy with the results shown for in-house trajectories in the main text. The main conclusions that can be drawn from the results presented here is that the behavior of both filters EKF and PKF is similar for in-house (given in Chapter 6) and external trajectories regarding the satellite locations. There are no main differences between each of the Vega trajectories used. The results provided below show that the behavior of both filters when the location of the observing satellites is changed is similar for each of these external trajectories. The reader is referred to the main text for a more in-depth analysis of the parametric performance study.

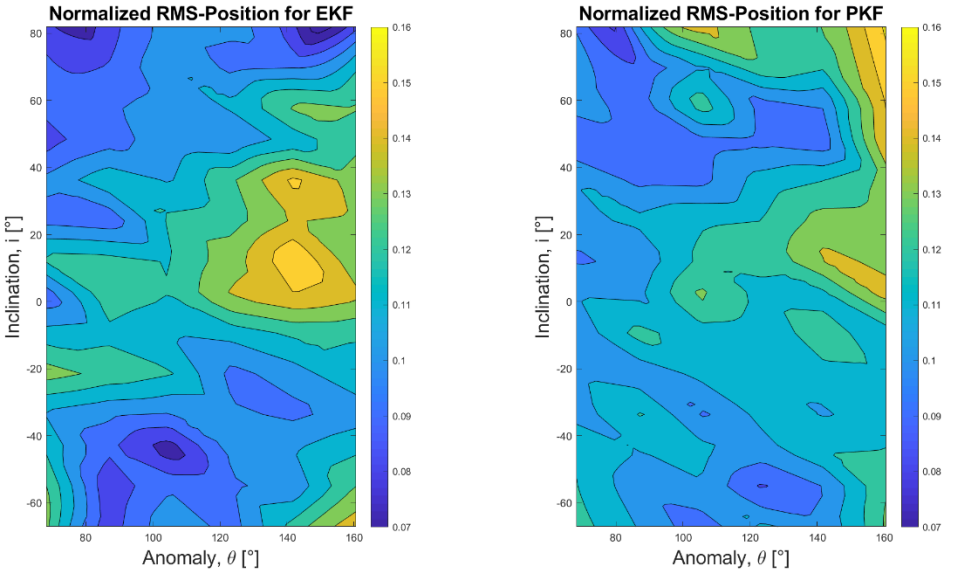

Figure 36: Parametrized satellite locations for Vega E

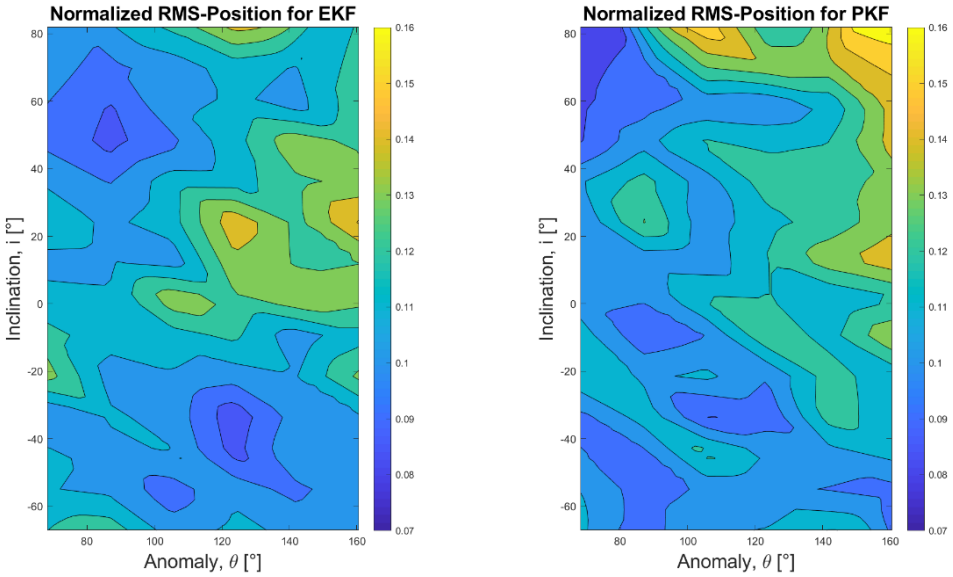

Figure 37: Parametrized satellite locations for Vega NE

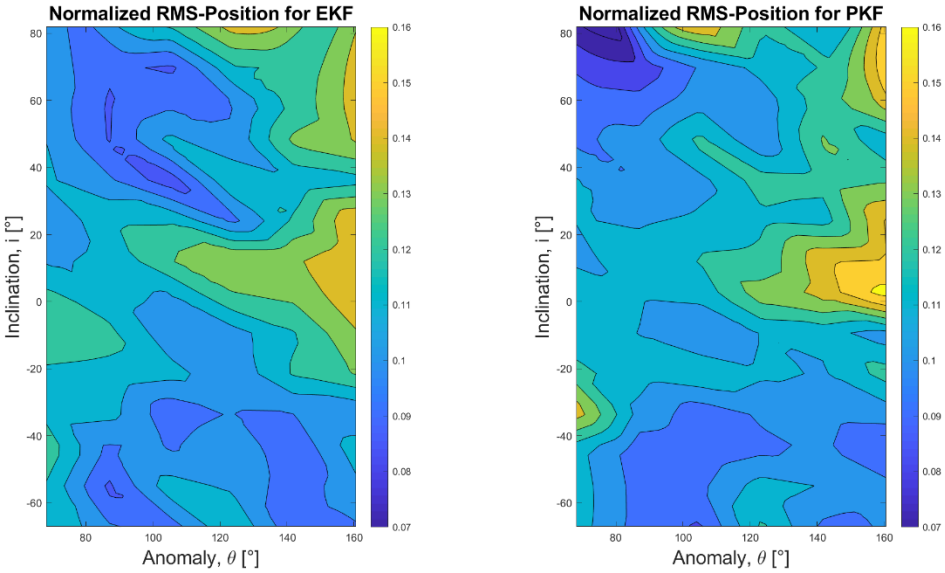

Figure 38: Parametrized satellite locations for Vega N