RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Final Report

Last update: December 10, 2025

Integrity Monitoring for High-Precision Navigation in Multi-Fault Scenarios

---

Presented on December, 2025 by

Shang-Ping Weng

Promotion:

ASNAT23

Supervised at **German Aerospace Center (DLR)** by

Andrea Bellés Ferreres

And at **École Nationale de l'Aviation Civile (ENAC)** by

Carl Milner

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Andrea and Carl. Despite the challenges and tight timeline during this internship, Andrea, thank you for your insightful suggestions and continuous support. Whenever I felt lost, you reminded me of the right direction and helped me stay on track. Your patience and guidance made this work possible. Thank you, Carl, for your feedback during the mid-term presentation and for helping me communicate with ENAC to ensure the internship proceeded smoothly. I would also like to thank our group leader, Daniel. Thank you, Daniel, for taking the time during the final stage of the internship to offer guidance and fight for the resources needed to complete this work.

To my lovely colleagues at DLR, thank you for being with me during my stay in Neustrelitz. The advice I received during coffee breaks, group meetings, and even on the bus helped me a lot. I would like to thank Michailas, Lucas, Filippo, Shradha, Christoph, Hakan, Alonso, Iulian, Ai-Chi, Carles, Malek, Jose, Esteban, Ashwin, and many others for creating such an wonderful environment and making my daily life during this internship so vibrant.

Finally, I would like to thank my dear family and Zhen Hui. Thank you for standing by me and supporting me during the most challenging moments. Your companionship and encouragement helped me stay strong and focused in completing this work.

Abstract

High-precision positioning is fundamental to ensure safe navigation in the field of autonomous transportation, where Global Navigation Satellite System (GNSS) play a pivotal role. However, navigational demands for Safety-of-Life (SoL) applications go beyond high accuracy, with integrity representing a critical aspect. This involves the integrity monitoring of the GNSS systems to ensure the high demanding requirements in terms of accuracy, integrity, and continuity are met. The use of GNSS carrier phase measurements is fundamental to obtain high-precision positioning solutions such as Precise Point Positioning (PPP), however it entails dealing with the complexity of integer phase ambiguities resolution. Furthermore, there is a lack of integrity concepts today accounting for absolute precise carrier phase positioning with integer ambiguity resolution and able to perform under multiple simultaneous faults, as it is the case for urban environments. This thesis aims at developing a methodological approach to assess the performance and detectability of integrity threats using robust estimators while exploiting carrier phase information. On the one hand, Multiple hypothesis solution separation (MHSS) schemes are well-established methods for integrity monitoring which involve considering different fault hypotheses (or fault modes) a-priori. Thus, it requires to run in parallel a bank of Kalman filters, one for each fault mode considered. This approach suffers from a high computational load and becomes increasingly complex when multiple faults are considered. On the other hand, Robust Statistics-based estimators have shown improved performance and resilience to outliers with respect to classical estimators under nominal and non-nominal conditions (i.e., when the Gaussian assumption is not hold). However, their applicability to integrity monitoring is still a field to be explored. It has been shown that the use of M-estimators for filtering can achieve higher accuracy performance of PPP float-ambiguity solution to a standard Kalman Filter even when outliers exist in measurements. To further explore the ability of robust filtering technique in PPP integrity monitoring, this thesis will focus on applying a robust estimator to achieve PPP fix-ambiguity solution, provide associated integrity information and compare its performance to the conventional MHSS scheme.

Keywords: Global Navigation Satellite System (GNSS), Precise Point Positioning (PPP), Integrity Monitoring (IM), Fault Detection and Exclusion (FDE), Multiple Hypothesis Solution Separation (MHSS), robust estimation, Robust Statistics, M-estimators.

Résumé

Le positionnement de haute précision est fondamental pour garantir une navigation sûre dans le domaine du transport autonome, où les systèmes mondiaux de navigation par satellite (GNSS) jouent un rôle essentiel. Cependant, les exigences de navigation pour les applications liées à la sécurité des personnes (SoL) vont au-delà de la haute précision, l'intégrité représentant un aspect critique. Cela implique la surveillance de l'intégrité des systèmes GNSS afin de garantir le respect des exigences élevées en termes de précision, d'intégrité et de continuité. L'utilisation des mesures de phase de porteuse GNSS est indispensable pour obtenir des solutions de positionnement de haute précision telles que le positionnement ponctuel précis (PPP), mais elle implique de gérer la complexité liée à la résolution des ambiguïtés entières de phase. En outre, il existe aujourd'hui un manque de concepts d'intégrité prenant en compte le positionnement absolu et précis basé sur la phase de porteuse avec résolution des ambiguïtés entières, tout en étant capable de fonctionner en présence de défaillances multiples simultanées, comme c'est le cas dans les environnements urbains. Ce rapport vise à développer une approche méthodologique permettant d'évaluer les performances et la détectabilité des menaces d'intégrité à l'aide d'estimateurs robustes tout en exploitant les informations de phase de la porteuse. D'une part, les méthodes de séparation de solutions à hypothèses multiples (MHSS) sont des approches bien établies pour la surveillance de l'intégrité, qui consistent à considérer a priori différentes hypothèses de défaillance (ou modes de défaillance). Ainsi, il est nécessaire d'exécuter en parallèle un ensemble de filtres de Kalman, un pour chaque mode de défaillance. Cette approche souffre d'une charge de calcul élevée et devient de plus en plus complexe lorsque plusieurs défauts sont pris en compte. D'autre part, les estimateurs basés sur la statistique robuste ont montré des performances et une résilience améliorées face aux valeurs aberrantes par rapport aux estimateurs classiques, dans des conditions nominales et non nominales (c'est-à-dire lorsque l'hypothèse gaussienne n'est pas respectée). Cependant, leur applicabilité à la surveillance de l'intégrité reste un domaine encore à explorer. Il a été démontré que l'utilisation d'estimateurs M pour le filtrage permet d'obtenir une meilleure précision dans la solution PPP à ambiguïtés flottantes qu'avec un filtre de Kalman standard, même en présence de valeurs aberrantes dans les mesures. Afin d'explorer plus en détail la capacité des techniques de filtrage robuste pour la surveillance de l'intégrité en PPP, ce rapport se concentrera sur l'application d'un estimateur robuste permettant d'obtenir une solution PPP à ambiguïtés fixées, de fournir les informations d'intégrité associées et de comparer ses performances à celles du schéma MHSS conventionnel.

Mots clés : Système mondial de navigation par satellite (GNSS), Positionnement ponctuel précis (PPP), Surveillance de l'Intégrité (IM), Détection et exclusion des défauts (FDE), Séparation des solutions à hypothèses multiples (MHSS), estimation robuste, Statistiques Robustes, estimateurs M.

Acronyms

AL Alert Limit. 39

AR Ambiguity resolution. 23

ARAIM Advanced Receiver Autonomous Integrity Monitoring. 17

CDF Cumulative Distribution Function. 41, 43, 46, 48, 63

CRAIM Carrier phase-based RAIM. 17

ECEF Earth Center Earth Fixed. 23

EKF Extended Kalman Filter. 21, 29

EKF-FDE EKF with Innovation based Fault Detection and Exclusion. 43

ENU East-North-Up. 37, 42

FDE Falut Detection and Exclusion. 17

GLONASS Global'naya Navigatsionnaya Sputnikovaya Sistema. 24

GNSS Global Navigation Satellite System. 17

GPS Global Positioning System. 24

HMI Hazardously Misleading Information. 39, 40

HPL Horizontal Protection Level. 42, 63

IAR Integer Ambiguity Resolution. 33

IGS International GNSS Service. 23

ILS Integer Least Square. 34

IM Integrity Monitoring. 11

ITS Intelligent Transportation Systems. 17

LAMBDA Least-squares AMBiguity Decorrelation Adjustment. 34

LS Least Square. 18

MADOCA Multi-GNSS ADvanced Orbit and Clock Augmentation. 19

MHSS Multiple Hypothesis Solution Separation. 18, 44

MI Misleading Information. 40

NEES Normalized Estimation Error Square. 31

NIS Normalized Innovation Square. 32

NLOS Non-Line-Of-Sight. 17

NO Normal Operation. 63

PAR Partial Ambiguity Resolution. 70

PE Position Error. 40

PL Protection Level. 17, 39

PPP Precise Point Positioning. 17, 18

PPP-AR PPP with Ambiguity Resolution. 19

QZSS Quasi-Zenith Satellite System. 19

RAIM Receiver Autonomous Integrity Monitoring. 17, 41

RINEX Receiver INdependent EXchange. 24, 50

RKF Robust Kalman Filter. 47

RMSE Root Mean Square Error. 37

RTK Real-Time Kinematic. 17, 23

SISRE Signal-In-Space Ranging Error. 24

SoL Safety-of-Life. 11

SU System Unavailable. 63

TTA Time To Alert. 39

Chapter 1

Introduction

1.1 Context

The demand for precise positioning is rapidly increasing across various domains, especially for the next generation of Intelligent Transportation Systems (ITS), which already pose much more stringent navigational requirements. Among the enabling technologies, Global Navigation Satellite System (GNSS) based on carrier phase measurements stand out for their ability to provide centimeter-level positioning accuracy. However, the conventional high-precision techniques (such as Real-Time Kinematic (RTK) or Precise Point Positioning (PPP)) come with significant challenges, for instance the integer nature of the ambiguities in the carrier phase measurements. In this context, integrity monitoring plays a central role in ensuring that the navigation solution is not only accurate, but also trustworthy, especially in safety-critical applications.

Initially, the concept of integrity was first developed for the aviation domain. In the context of GNSS, one typically refers to integrity monitoring at the user-level due to the popularity of Receiver Autonomous Integrity Monitoring (RAIM), which was limited to the use of pseudorange measurements and single fault detection (mainly errors coming from a satellite service failure). This assumption worked in the aviation domain since the chances of faults occurring simultaneously was rather low; however, these assumptions do not hold for high-precision applications in more challenging environments, such as urban canyons, where multiple simultaneous faults, Non-Line-Of-Sight (NLOS) conditions, and degraded satellite visibility occur regularly.

With the rise of multiple constellations, Advanced Receiver Autonomous Integrity Monitoring (ARAIM) was developed to address multiple simultaneous faults and to exploit the use of multiple frequencies across multiple constellations [1]. For instance, one can take advantage of dual-frequency combinations to eliminate ionospheric effects. ARAIM's Falut Detection and Exclusion (FDE) process is based on multi-hypothesis testing to detect and exclude multiple faults. It evaluates all potential fault modes in the position domain to carry out FDE and Protection Level (PL) calculations, while ensuring that the integrity risk allocation is based on a threat model that accounts for multiple faults. However, one important drawback of this methodology is its high computational load due to the need to check every possible fault modes, that is each possible combination or subset of observations [2, 3, 4]. In order to exploit the carrier phase measurements, another promising algorithm called Carrier phase-based RAIM (CRAIM) was later on developed [5], nevertheless it also relies on the assumption of single fault and it assumes the prior fix of the carrier phase ambiguities to be true. Some attempts to account for the latter assumption by providing validation for the ambiguity fixes based on ratio tests have been carried

out; but more extensive testing of such algorithm under more challenging conditions and the extension to account for multiple faults is required.

This thesis addresses the problem of integrity monitoring for carrier-phase-based high-precision navigation developing robust mechanisms to detect and mitigate multiple simultaneous faults. In contrast to the state-of-the-art, which often relies on Multiple Hypothesis Solution Separation (MHSS) schemes (i.e., bank of parallel filters), this work proposes the use of M-estimators based on robust statistics. This type of robust Kalman Filters can mitigate the impact of outliers within the estimation process itself by penalizing the residuals coming from the faulty measurements while preserving the benefits of precise carrier-phase positioning [6, 7]. Unlike MHSS, the robust Kalman Filter does not relay on the assumption of fault models so that it can avoid the time consuming process of running through the bank of filters. However, the robust Kalman Filter also does not have any assumption on the measurement noise which is usually assumed to be white Gaussian distributed. This means that there is no formal statistical bound available to ensure the reliability of the estimated solution.

A key contribution of this work is the investigation of integrity monitoring concept for PPP applications with ambiguity resolution including:

- Implementation and validation of a MHSS scheme with FDE based on the χ^2 test statistics,
- Integration of ambiguity resolution into the MHSS scheme,
- Computation of PLs for the MHSS scheme,
- Performance comparison between MHSS and M-type based robust filters under nominal, single- and multi-fault scenarios.

The proposed methods are evaluated through a combination of Monte Carlo simulations, focusing on their ability to maintain accuracy and integrity in the presence of multiple fault scenarios.

Building on a robust estimation scheme proposed in [8], this internship focuses on developing an algorithm to provide integrity information for ambiguity-fixed solutions in PPP. Popular techniques such as Multiple Hypothesis Solution Separation (MHSS) will be implemented and compared to the performance of robust filters leveraging on Robust Statistics.

1.2 Related works

RAIM has been widely used in aviation [9]. This method can detect a single fault in GNSS pseudornage measurement based on the Least Square (LS) estimator. It can also isolate a single fault based on the test statistic calculated from the normalized measurement residual. RAIM only consider single fault with the assumption that multiple fault events are very unlikely to happen, especially in the aviation domain. With the increasing number of satellites and multiple GNSS constellations, the ARAIM technique is introduced to consider multiple fault events. The fault detection and exclusion (FDE) is usually performed with the measurement residual or solution separation method. Based on the fault model, the solution separation strategy breaks down the integrity risk into each fault hypothesis and computes the protection level accordingly. Traditional ARAIM is also based on LS estimator and a combination of pseudorange measurement for integrity monitoring in aviation [1].

To provide integrity information for PPP with both pseudorange and carrier-phase measurement, Feng et al. proposed to perform the ratio test with a doubly non-central F-distribution instead

of a fix threshold during the ambiguity resolution process [10]. This method provides both ambiguity resolution and positioning protection for users, but only focus on single fault on a single satellite with simulated data. For further adaptation to multiple faults, Gunning et al. implemented both solution-separation-based and residual-based ARAIM algorithm for PPP in kinematic applications with real-world ionosphere-free combination data [11]. By sharing most modeled effects in PPP, the computation coast of running a bank of filters in solution-separation-based technique can be reduced and outperforms the residual based approach. Several fault modes were considered for integrity monitoring but their prior probabilities require more investigation. Two methods for FDE are presented in [12]. The original method requires the entire bank of filters to be completely reinitialized once a fault has been detected and excluded, which will lead to a jump in the protection levels. The second method with 'on-deck' filters is developed to avoid large jump and re-converging of the protection level. However, this will further increase the number of filters that need to be tracked and make the algorithm even more computationally expensive. Furthermore, only the float solution in PPP is considered in [11, 12].

A Kalman Filter-based integrity monitoring algorithm for PPP is implemented in [13] using the Multi-GNSS ADvanced Orbit and Clock Augmentation (MADOCA) message transmitted from Quasi-Zenith Satellite System (QZSS). The innovation-based FDE strategy can deal with at most two simultaneous faults. The proposed algorithm is validated with static and kinematic real-world data with simulated fault. The results indicate that the innovation-based FDE mechanism combined with a Kalman filter is reliable. A solution separation-based integrity monitoring approach for carrier-phase ambiguity resolution in PPP is proposed in [14]. This approach can accommodate observation faults and incorrect ambiguity fixing simultaneously through the modified MHSS scheme. However, this approach does not implement fault exclusion.

Previous research mainly focused on single fault instead of multiple faults in PPP integrity with MHSS ARIAM based approach due to the increasing complexity and computation load when multi-frequency and multi-constellation measurements are used [15]. This is because MHSS technique needs to go through a bank of filters corresponding to different fault hypothesis by assuming a specific combination of faulty measurements. One promising method to overcome this limitation is to use a robust filtering framework with PPP with Ambiguity Resolution (PPP-AR) [16]. Instead of excluding potential faulty measurements based on fault hypothesis, robust statistics-based filtering can down-weight the unhealthy measurements with adaptive re-weighting scheme to reduce their impact on positioning results. It has been shown that the robust filter can achieve similar performance to an ideal fault-free estimator regarding PPP float-ambiguity solution. Based on the existing robust filtering framework, it would be logical to further explore its performance on PPP fix-ambiguity solution and investigate the performance of robust filtering techniques for integrity monitoring in PPP.

1.3 Proposed method and internship objective

Based on previous discussion, the current challenges and limitations of PPP integrity monitoring can be sum up to the following points:

- The heavy computational load of MHSS based technique when multiple faults are considered.
- There is a lack of comparison between MHSS and robust filtering based integrity monitoring.

To address these challenges, we propose a robust filtering-based PPP for integrity monitoring. We will consider multiple fault scenarios and compare the results between with MHSS-based and robust estimator-based approach.

The objectives and expected deliverable of this internship are listed below:

- Review literature regarding integrity monitoring for high-precision navigation (i.e, PPP), integer ambiguity resolution and robust estimation.
- Become familiar with a GNSS Simulator for generation of measurements, noise and fault profiles injection and evaluation of different navigation filters.
- Implement and test a Multi Hypothesis-based Kalman Filter for integrity monitoring and compare it to the Robust Kalman Filtering solution under multi-fault scenarios.
- Understand how a research-oriented project works.
- Contribute to a research paper detailing the research output of the thesis.

1.4 Internship Environment

The Institute of Communications and Navigation at German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR) is located in Neustrelitz. It plays a crucial role in space programs such as sun observation, satellite orbit monitoring, and satellite communication and navigation. The department of Communication and Navigation for Nautical System consists of around 20 researchers and students from Spain, Italy, India, Germany, Mauritius, Brazil, Lietuva, and Taiwan. The international and relatively young team members bring a dynamic energy and make collaboration easy.

The following individuals supervised the preparation of this thesis:

- Andrea Bellés Ferreres, GNSS Research and Development Engineer of the Multi Sensor Systems group in the Institute of Communication and Navigation at DLR,
- Daniel Medina, Group Leader of the Multi Sensor Systems group in the Institute of Communication and Navigation at DLR.

1.5 Internship Plan

The tasks for this internship are described and shown in Figure 1.1 as follow:

- Topic approval: Define the objective and scope of the internship, including the problem we want to solve, targeted application, and expected results.
- Literature Review: Go through the existing literature to review the state-of-the-art in the research subject. Understand required theoretical knowledge, mathematical and software tools, and potential challenges.
- Methodology design: Based on the state-of-the-art, choose and design the appropriate method that can solve our problem and achieve targeted results.
- Implementation: Implement the designed methodology in software and verify the result with simulated data. If time allows, apply the verified method with real-world data.

- Analysis and writing: Analyze results from simulations or real-world data. Compare these results with existing methods, highlighting their strengths and weaknesses. Document research findings in a thesis that clearly explains the work conducted.
- Final submission: Submit the final thesis to the jury and perform oral defense on the work.

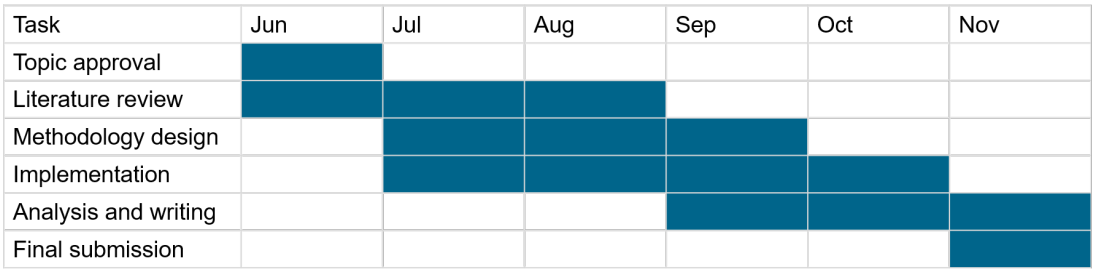


Figure 1.1: Plan for the task in this internship

1.6 Outline

Chapter 2 outlines the essential background for the estimation problem in Precise Point Positioning (PPP). It begins with the PPP observation model and the various ranging error models, followed by a description of the classical estimator used for solving PPP, i.e. Extended Kalman Filter (EKF). The concept of filter consistency and several approaches for evaluation are also presented. The chapter then introduces the mixed model of the estimation problem considering the integer nature of the carrier-phase ambiguity. Finally, the chapter provides the brief description of robust estimation.

Chapter 3 explores the concept of integrity monitoring with a focus on PPP. It begins by introducing the key parameters and definitions related to integrity monitoring. The chapter then introduce the classic RAIM, outlines the techniques examined in this thesis and presents flow charts to illustrate the fundamental differences between them. Finally, it describes the calculation of Protection Levels using the slope-based method inspired by RAIM.

Chapter 4 outlines the MATLAB implementation used to evaluate the performance of the integrity monitoring algorithms considered in this thesis. It begins with a high-level overview of the software structure and describes the inputs and outputs of each main module. The chapter then examines each implemented integrity monitoring algorithm in detail, using pseudo-code to clearly illustrate the workflow. Finally, it presents several tables summarizing the chosen parameter values and the overall simulation configuration in different scenarios.

Chapter 5 presents and discuss the results for the implemented integrity monitoring algorithms in different scenarios. Each scenario is investigated by running 20 Monte Carlo simulations. Finally, Chapter 6 provides the conclusion of this thesis and recommends future work directions.

Chapter 2

Estimation for Precise Positioning

There are several techniques that can achieve decimeter- to centimeter-level accuracy with GNSS. Classical differential GNSS techniques, such as RTK, allow users to achieve centimeter-level positioning accuracy when a reference station at a known location is available. This technique assumes that the baseline distance between the user and the reference station is within an acceptable range so that both experience nearly identical measurement noise. However, this assumption also limits the regional coverage of RTK. The Precise Point Positioning (PPP) technique has become increasingly popular as measurement-correction products from worldwide reference ground stations—such as those provided by the International GNSS Service (IGS)—have become publicly available. This technique enables users to achieve decimeter-level positioning accuracy worldwide, but typically requires a relatively long convergence time. To combine the strengths and compensate for the limitations of both methods, PPP-RTK uses a regional network of base stations to deliver precise measurement corrections, improving positioning performance. This approach offers wider service coverage than RTK and shorter convergence times than PPP. Carrier-phase Ambiguity resolution (AR) can also be performed using classical PPP. This approach, known as PPP-AR, fully exploits the integer nature of the carrier-phase ambiguity to further improve positioning accuracy from the decimeter to the centimeter level.

In this chapter, we begin with a description of the observation and state-dynamic models used in classical PPP. The PPP estimation problem is then presented, followed by the estimator and the techniques used to solve it. Finally, the methods and available tools for achieving ambiguity resolution in PPP are discussed.

2.1 Observation model

To achieve high accuracy positioning, PPP uses both code and carrier-phase observation. The code and carrier-phase observations between satellite s and receiver on frequency f can be formulated as follow:

$$\begin{aligned} \rho_f^s &= \|\mathbf{p}^s - \mathbf{p}\| + c(dt - dt^s) + \gamma_f \cdot I_1^s + m^s \cdot T + b_f - b_f^s + \varepsilon_f^s \\ \Phi_f^s &= \|\mathbf{p}^s - \mathbf{p}\| + c(dt - dt^s) - \gamma_f \cdot I_1^s + m^s \cdot T + \lambda_f a_f^s + B_f - B_f^s + \epsilon_f^s \end{aligned} \tag{2.1}$$

where $\mathbf{p}^s$ and $\mathbf{p}$ are the three-dimensional position of satellite s and receiver in Earth Center Earth Fixed (ECEF) frame in meters, c is the speed of light in m/s, dt and dt^s are the receiver and satellite clock bias in meters, $\gamma_f = \lambda_f^2/\lambda_1^2$ is a frequency-dependent factor for the slant ionospheric delay I_f^s in meters on frequency f, m^s is the mapping coefficient for tropospheric wet delay T in meters on satellite s, b_f and b_f^s are the receiver and satellite code biases in

meters, B_f and B_f^s are the receiver and satellite carrier-phase biases in meters, λ_f is the signal wavelength in meters on frequency f, a_f^s is the carrier-phase integer ambiguity in cycles, ε_f^s and ϵ_f^s are unmodelled code and carrier-phase white noises in meters. It is important to carefully model the error terms in PPP observation model to remove as much error as possible. Further description of the model used in the prediction of pseudorange and carrier-phase are provided in the following subsections.

2.1.1 Precise satellite orbit and clock

Precise orbit and clock information is necessary for PPP. The user can receive these information from a global network of hundreds-of survey-grade multi-GNSS receivers, for example: International GNSS Services (IGS). These receivers continuously monitor the GNSS constellation with strongly accurate models to produce highly precise orbit and clock solution products. Some product claims to provide GNSS Signal-In-Space Ranging Error (SISRE) better than 5 cm for different constellation such as Global Positioning System (GPS), Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS), and Galileo [12]. The data of precise satellite orbit and clock is usually in the Receiver INdependent EXchange (RINEX) .sp3 standard format and can be downloaded from several publicly available websites. One challenge of using the precise orbit data is to match satellite position with the time of transmission of the signals. This can be achieved by high order interpolation technique, such Lagrange interpolation, to ensure the level of accuracy of satellite position.

On the other hand, precise satellite clock products are usually provided at a much higher rate than precise orbit data. This is because the on-orbit atomic clocks at sub-nanosecond level are much more unpredictable than orbital parameters. Thanks to the relatively simple dynamics of clocks and high data rate, the precise clock product can typically be used with simple linear interpolation. Note that the precise clock products refer to a specific ionosphere-free dual frequency combination. Therefore, a differential code bias must be considered when the measurements different from the reference signal pair are used.

The satellite orbit parameters used in this thesis are simulated from an existing multi GNSS simulator tool in DLR. The simulator takes a predefined receiver trajectory and a RINEX file as input and outputs satellite position. For the simplicity of the simulation, the reference satellite clock is simply simulated as a random walk process.

2.1.2 Ionospheric delay

The ionosphere is an upper atmosphere that contains electrically charged particles. The ultraviolet radiation from the Sun breaks gas molecules into free electrons and ions. The ionosphere causes dispersive impact on L-band signal that leads to frequency-dependent delays or advances on GNSS code and carrier-phase measurement. The delays or advances of the signal depend on the number of free electrons along the signal's propagation path. The number of electrons in a tube with 1 m^2 cross section is known as the total electron content (TEC). The TEC is measured in TEC units (TECU) that defined as 10^{16} electrons $/m^2$. One TEC unit can cause a delay or advance of 16.2 cm on the GPS L1 frequency. In the case of very high ionospheric activity, the peak value of TEC in the vertical direction can reach more than 200 TECU [17].

When multiple frequency measurements are available, the Ionosphere-Free combination can be formed to reduce up to 99.9% of the ionospheric delay in code, or advance in carrier-phase. However, this technique of removing the ionosphere delay comes with some costs. The combined measurement will have significantly increased standard deviation compared with the original

uncombined measurement. Furthermore, the resulting wavelength of this combination becomes much smaller than uncombined measurement, for example: only approximately 6 mm for GPS L1 and L2 combination. This will destroy the integerness of the resulting combined ambiguity and make ambiguity resolution much more difficult [17]. The alternative is to apply externally estimated ionospheric delay on single frequency measurements to avoid the draw back of combining measurements. In this thesis, the ionospheric delay is simply simulated as a random walk process and added or subtracted to simulated code and carrier-phase measurement.

2.1.3 Tropospheric delay

The troposphere is a neutral atmosphere that is non-dispersive for L-band signal. Its effect on GNSS signal are frequency independent and have the same delay for code and carrier-phase measurement. The total tropospheric delay can be separated into hydrostatic part and a nonhydorstatic part, i.e. the dry and wet component. The former only depends on the pressure and accounts for the majority of the total delay, while the latter depends on water vapor pressure profile and is only responsible for the minor part of the delay. A common method to model the tropospheric delay is to relate the state of the atmosphere at an arbitrary altitude to the user's altitude and calculate the delay in the zenith direction. The zenith tropospheric delay is then multiplied with a mapping function to map the total delay to obtain the slant delay along the signal propagation path. The mapping functions depends on the elevation angle of the satellite and are different for the dry and wet component. An additional term can be introduced to estimated the total tropospheric slant delay in PPP as follows:

$$T_{\text{slant,total}} = m_d(E)D_{Z,D} + m_w(E)D_{Z,W} + m_w(E)\hat{T} \tag{2.2}$$

where m_d and m_w are the mapping functions for dry and wet delay, $D_{Z,D}$ and $D_{Z,W}$ are the modeled dry and wet zenith delay, E is the elevation angle of satellite, and $\hat{T}$ is the additional term to estimate tropospheric zenith delay.

In this thesis, the Vienna Mapping Function is used to compute the delay mapping functions with the input of satellite elevation, user's latitude and altitude, and the day of the year. For the simplicity of the simulation, only the term including $\hat{T}$ in Equation 2.2 is considered which is simulated as a random walk process.

2.1.4 Other sources of errors

In real-world PPP implementation, it is necessary to consider other important correction terms to achieve center meter level positioning. These corrections are presented below even though they are not modeled in this thesis to simplify the simulation.

2.1.4.1 Antenna phase center offset and variance

The geometry range is modeled as the distance between satellite and receiver in code-based Single Point Positioning. In reality, however, the ranging signals are transmitted from the phase center of the transmission antenna on satellites. The satellite orbit estimates from IGS refer to the center of mass of the satellite. Thus a body frame displacement between the satellite's center of mass and the antenna phase center need to be considered with a body frame attitude model. The attitude is modeled with the fact that the satellite always points its solar panels toward the sun to maximize received sunlight. The satellite's phase center offsets in the body frame are available in IGS published product. It must be projected on the positive line-of-sight vector from

the receiver to the satellite in ECEF frame. These offsets can be larger than one meter in the bore-sight direction which makes it an important correction in real-world implementation. The antenna phase center variation is a bias that depends on look angle and signal frequency. It is also provided in the IGS product as a function of elevation and azimuth from satellite's antenna frame with the value at sub-center meter level.

Similar effect also exists on the receiver when the antenna phase center is not aligned with the antenna's body frame reference point. The phase center offset of the antennas in the IGS network are also provided in the body frame and need to be projected on the negative line-of-sight vector from the receiver to satellite in ECEF frame. Typical values of the antenna phase offset and variation for a survey-grade antenna is at millimeter level and can reach to 10-20 centimeters for an inexpensive patch antenna. Therefore, the choice of an antenna is critical for precise application especially when ambiguity fixing is involved.

2.1.4.2 Relativity effect

The relativity effect in PPP can be split into two parts. The first part accounts for the deviation of the satellite clock due to non-circular orbit around the Earth. This satellite clock offset can be formulated with satellite's position and velocity vector in ECEF frame $\mathbf{p}^s$ and $\dot{\mathbf{p}}^s$ as follow:

$$\delta t_{clk}^{rel} = -\frac{2}{c^2}(\mathbf{p}^s \cdot \dot{\mathbf{p}}^s) \tag{2.3}$$

Typical value of this effect can reach meter-level.

The second part is the delays affecting the signal path due to the gravitational potential of the Earth. This is known as the Shaprio effect and the corresponding delay correction can be obtained from:

$$\delta t_{stc}^{rel} = \frac{2\mu}{c^3} \ln\left(\frac{||\mathbf{p}^s|| + ||\mathbf{p}|| + ||\mathbf{p}^s - \mathbf{p}||}{||\mathbf{p}^s|| + ||\mathbf{p}|| + ||\mathbf{p}^s + \mathbf{p}||}\right) \tag{2.4}$$

where μ is the Earth's gravitational constant. Typical value of Shaprio effect can reach 60 ps or 2 cm for GNSS satellites.

2.1.4.3 Phase wind up effect

Since the GNSS satellites are transmitting right-hand circularly polarized signals, a relative rotation in the boresight direction between the transmitting and receiving antenna can cause a change in the measured carrier phase. A full rotation corresponds to one cycle phase change for all frequency. This effect is known as the phase wind up and is closely related to the relative attitude of transmitter and receiver antenna. The phase wind up effect due to the rotation of the receiver antenna is identical for all received signal. Therefore can be absorb by an common epoch-wise parameter for all carrier-phase observation such as receiver clock. On the other hand, the phase wind up effect from the transmitting antenna is different between each satellite and need to be considered separately in undifferenced observations.

2.1.4.4 Solid earth tides effect

The gravitational attraction of the Sun and Moon causes a daily deformation of the Earth's crust on the order of tens of centimeters. The resulting horizontal and vertical displacement at a given point on the Earth's surface can be modeled as a function of the time of day, latitude, longitude, and the position of the Sun and the Moon in ECEF frame. Typical values of the displacement

caused by solid earth tides is about 0.3 meters in vertical direction and about 5 centimeter in the horizontal direction. This effect is highly predictable and must be considered if centimeter level positioning is required.

2.2 Estimation problem

The classical way to solve the PPP problem involves the estimation of several unknown state parameters recursively over time given a set of observations (or measurements) at each timestamp defined by the observation model described in 2.1. The set of unknown states are estimated using an Extended Kalman Filter (EKF), for which the evolution and stochastic model is defined in this section.

The measurement vector used in this thesis includes dual frequencies code and carrier-phase measurement from GPS (L1+L2 frequency) and Galileo (L1+L5 frequency) system. Based on the observation model in Equation (2.1), the measurement vector can be defined as follow:

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\rho}_{L1}^{s_G} \\ \boldsymbol{\rho}_{L1}^{s_E} \\ \boldsymbol{\rho}_{L2}^{s_G} \\ \boldsymbol{\rho}_{L5}^{s_E} \\ \boldsymbol{\Phi}_{L1}^{s_G} \\ \boldsymbol{\Phi}_{L1}^{s_E} \\ \boldsymbol{\Phi}_{L2}^{s_G} \\ \boldsymbol{\Phi}_{L5}^{s_E} \end{bmatrix}_{4m\times 1} \tag{2.5}$$

where s_G refers to the satellites in GPS, and s_E refers to the satellites in Galileo, and $m = m_G + m_E$ is the total number of available satellites which includes m_G satellites from GPS and m_E satellites from Galileo.

The unknown state vector includes the three-dimensional receiver ECEF position and velocity, receiver clock, tropospheric delay, ionospheric delay, and dual-frequency ambiguities, which can be expressed as follow:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_{3\times 1} \\ \dot{\mathbf{p}}_{3\times 1} \\ dt^G \\ dt^E \\ T \\ I_1^1 \\ \vdots \\ I_1^m \\ \mathbf{a}_{2m\times 1} \end{bmatrix}_{n\times 1} \tag{2.6}$$

where dt^G is the receiver clock bias for GPS, and dt^E is the receiver clock bias for Galileo, and n is the number of unknown states to be estimated. The state transition matrix for position and velocity is a simple constant velocity model, and for the remaining states is simply an identity

matrix:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta t\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{n-6} \end{bmatrix} \tag{2.7}$$

where $\mathbf{0}$ is the all-zeros matrix with the correct shape. The process noise is a simple diagonal matrix with each elements corresponding to the variance of the random walk model assumed to each state:

$$\mathbf{Q} = \Delta t \begin{bmatrix} \sigma_{pos}^2\mathbf{I}_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{vel}^2\mathbf{I}_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{dt}^2\mathbf{I}_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_I^2\mathbf{I}_m & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{amb}^2\mathbf{I}_{2m} \end{bmatrix} \tag{2.8}$$

The Jacobian matrix of the measurement model, also known as design matrix or sensitivity matrix, is the partial derivative of the measurement model with respect to each state in the state vector. Each row corresponds to one measurement and each column corresponds one state. The design matrices for each type of measurement are presented as follow:

$$\mathbf{H}_{\rho_{L1}^{s_G}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_G)}-\mathbf{p})^T}{||\mathbf{p}^{(s_G)}-\mathbf{p}||} & \mathbf{0}_{m_G\times 3} & \mathbf{1}_{m_G\times 1} & \mathbf{0}_{m_G\times 1} & m^{(s_G)}\mathbf{1}_{m_G\times 1} & \gamma_{L1}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times m_E} & \mathbf{0}_{m_G\times 2m} \end{bmatrix} \tag{2.9}$$

$$\mathbf{H}_{\rho_{L1}^{s_E}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_E)}-\mathbf{p})^T}{||\mathbf{p}^{(s_E)}-\mathbf{p}||} & \mathbf{0}_{m_E\times 3} & \mathbf{0}_{m_E\times 1} & \mathbf{1}_{m_E\times 1} & m^{(s_E)}\mathbf{1}_{m_E\times 1} & \mathbf{0}_{m_E\times m_G} & \gamma_{L1}\mathbf{I}_{m_E} & \mathbf{0}_{m_E\times 2m} \end{bmatrix} \tag{2.10}$$

$$\mathbf{H}_{\rho_{L2}^{s_G}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_G)}-\mathbf{p})^T}{||\mathbf{p}^{(s_G)}-\mathbf{p}||} & \mathbf{0}_{m_G\times 3} & \mathbf{1}_{m_G\times 1} & \mathbf{0}_{m_G\times 1} & m^{(s_G)}\mathbf{1}_{m_G\times 1} & \gamma_{L2}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times m_E} & \mathbf{0}_{m_G\times 2m} \end{bmatrix} \tag{2.11}$$

$$\mathbf{H}_{\rho_{L5}^{s_E}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_E)}-\mathbf{p})^T}{||\mathbf{p}^{(s_E)}-\mathbf{p}||} & \mathbf{0}_{m_E\times 3} & \mathbf{0}_{m_E\times 1} & \mathbf{1}_{m_E\times 1} & m^{(s_E)}\mathbf{1}_{m_E\times 1} & \mathbf{0}_{m_E\times m_G} & \gamma_{L5}\mathbf{I}_{m_E} & \mathbf{0}_{m_E\times 2m} \end{bmatrix} \tag{2.12}$$

$$\mathbf{H}_{\Phi_{L1}^{s_G}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_G)}-\mathbf{p})^T}{||\mathbf{p}^{(s_G)}-\mathbf{p}||} & \mathbf{0}_{m_G\times 3} & \mathbf{1}_{m_G\times 1} & \mathbf{0}_{m_G\times 1} & m^{(s_G)}\mathbf{1}_{m_G\times 1} & \\ & -\gamma_{L1}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times m_E} & \lambda_{L1}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times(2m-m_G)} & \end{bmatrix} \tag{2.13}$$

$$\mathbf{H}_{\Phi_{L1}^{s_E}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_E)}-\mathbf{p})^T}{||\mathbf{p}^{(s_E)}-\mathbf{p}||} & \mathbf{0}_{m_E\times 3} & \mathbf{0}_{m_E\times 1} & \mathbf{1}_{m_E\times 1} & m^{(s_E)}\mathbf{1}_{m_E\times 1} & \\ & \mathbf{0}_{m_E\times m_G} & -\gamma_{L1}\mathbf{I}_{m_E} & \mathbf{0}_{m_E\times m_G} & \lambda_{L1}\mathbf{I}_{m_E} & \mathbf{0}_{m_E\times m} \end{bmatrix} \tag{2.14}$$

$$\mathbf{H}_{\Phi_{L2}^{s_G}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_G)}-\mathbf{p})^T}{||\mathbf{p}^{(s_G)}-\mathbf{p}||} & \mathbf{0}_{m_G\times 3} & \mathbf{1}_{m_G\times 1} & \mathbf{0}_{m_G\times 1} & m^{(s_G)}\mathbf{1}_{m_G\times 1} & \\ & -\gamma_{L2}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times m_E} & \mathbf{0}_{m_G\times m} & \lambda_{L1}\mathbf{I}_{m_G} & \mathbf{0}_{m_G\times m_E} \end{bmatrix} \tag{2.15}$$

$$\mathbf{H}_{\Phi_{L5}^{s_E}} = \begin{bmatrix} -\frac{(\mathbf{p}^{(s_E)}-\mathbf{p})^T}{||\mathbf{p}^{(s_E)}-\mathbf{p}||} & \mathbf{0}_{m_E\times 3} & \mathbf{0}_{m_E\times 1} & \mathbf{1}_{m_E\times 1} & m^{(s_E)}\mathbf{1}_{m_E\times 1} \\ & \mathbf{0}_{m_E\times m_G} & -\gamma_{L5}\mathbf{I}_{m_E} & \mathbf{0}_{m_E\times(m+m_G)} & \lambda_{L5}\mathbf{I}_{m_E} \end{bmatrix} \tag{2.16}$$

where $\mathbf{1}_{m_G\times 1}$ is a all-ones vector with the shape of $m_G \times 1$, and $\mathbf{0}_{m_G\times 3}$ is a all-zero matrix with the shape of $m_G \times 3$. The whole design matrix can be built by vertically stacking the design matrix for each type of measurements with the same order in the measurement vector in

Equation (2.5) as follow:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\rho_{L1}^{sG}} \\ \mathbf{H}_{\rho_{L1}^{sE}} \\ \mathbf{H}_{\rho_{L2}^{sG}} \\ \mathbf{H}_{\rho_{L5}^{sE}} \\ \mathbf{H}_{\Phi_{L1}^{sG}} \\ \mathbf{H}_{\Phi_{L1}^{sE}} \\ \mathbf{H}_{\Phi_{L2}^{sG}} \\ \mathbf{H}_{\Phi_{L5}^{sE}} \end{bmatrix} \tag{2.17}$$

The measurement noise matrix is a diagonal matrix with each element corresponds to the measurement vector in Equation (2.5). The weighting for each measurement depends on the satellite elevation as follow:

$$\mathbf{R} = \begin{bmatrix} \sigma_\rho^2 \mathbf{I}_{2m} & \mathbf{0} \\ \mathbf{0} & \sigma_\Phi^2 \mathbf{I}_{2m} \end{bmatrix} \tag{2.18}$$

$$\sigma_\rho = a + \frac{b}{\sin(elev)} \tag{2.19}$$

$$\sigma_\Phi = 10^{-2} \cdot \sigma_\rho \tag{2.20}$$

where a and b are the weighting parameters for pseudorange measurement, and $elev$ is the satellite elevation in radian.

At this point, we have all the matrices and vectors in our hand except the initial state vector and its covariance matrix to perform the prediction and update in the Kalman Filter which is introduced in section 2.2.1. The initialization of the state vector and its covariance matrix can be decided by prior knowledge of the states or by using a least square estimation with the initial measurements.

2.2.1 Fundamentals of Extended Kalman Filter

Extended Kalman Filter (EKF) is a well-known recursive estimator to solve nonlinear estimation problem such as PPP [11, 12, 13, 16]. In practice, the discrete EKF algorithm is separated into two steps: state prediction step and measurement update step, as shown in Figure2.1. In the state prediction step, the filter predicts the states at the current epoch from the estimated states at the precious epoch. The prediction is based on the assumption of system's state process model $f(\cdot)$ in Equation (2.46). The mathematical expression in state prediction step can be expressed as follow:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{2.21}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.22}$$

where $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are the predicted state vector and covariance matrix at epoch k using the measurements up to epoch $k-1$, $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$ are the updated state vector and covariance matrix at epoch $k-1$ using the measurements up to epoch $k-1$, $\mathbf{F}_{k-1}$ is the state transition matrix at epoch $k-1$, $\mathbf{Q}_{k-1}$ is the process noise covariance at epoch $k-1$. After the prediction step, the predicted state and its covariance (a priori state and covariance) are used

as input with new measurements $\mathbf{y}_k$ in the measurement update step . It updates the state and covariance with the observation model $h(\cdot)$ in Equation (2.1) as follow:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{2.23}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\big(\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})\big) \tag{2.24}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{2.25}$$

where $\mathbf{K}_k$ is the Kalman gain, $\mathbf{H}_k$ is the Jacobian matrix of the measurement model. Note that the term $h(\hat{\mathbf{x}}_{k|k-1})$ is using the non-linear measurement model to calculate predicted measurement and subtracted from the observed measurement. The terms $\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})$ and $\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k$ are know as innovation vector and its covariance which can be used for fault detection in integrity monitoring as described in subsection 3.2.2.1. After the measurement update, the updated states and covariance (a posteriori) are provided to the user and used as the input to the state prediction step in next epoch. The iterative process continues until the last epoch in post-processing or algorithm termination in real-time processing. In certain situation, such as measurement unavailable or invalid, the measurement update step can be skipped and the filter will only perform state prediction.

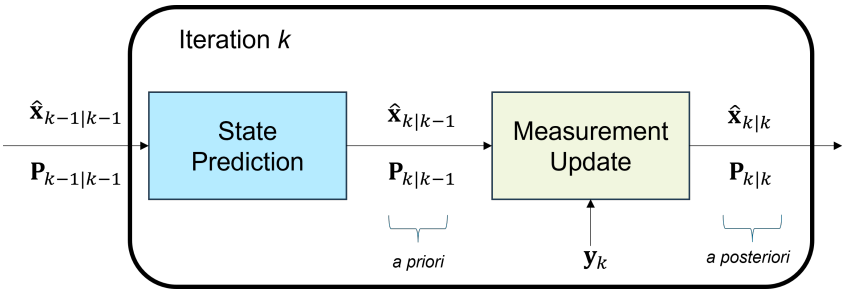


Figure 2.1: EKF algorithm is divided into state prediction and measurement update steps

2.2.1.1 Filter consistency

In the problem of estimating the state of a dynamic system, the Kalman Filter provides not only the current estimate of the state $\hat{\mathbf{x}}_{k|k}$ but also its associated covariance matrix $\mathbf{P}_{k|k}$. Sometimes the filter has divergence issue that it outputs unacceptably large state estimation errors. This can be caused by modeling errors, numerical errors, or programming errors [18]. To avoid the divergence issue, it is important to evaluate if the filter can provide acceptable estimation error. Under the Linear-Gaussian assumption, the conditional probability density function of the state $\mathbf{x}_k$ at the k epoch is

$$p(\mathbf{x}_k|\mathbf{y}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \tag{2.26}$$

The system modeling includes state dynamic model, measurement model, and the statistical properties of the random variables entering these models. Equation (2.26) means that when all the system models are accurate, the filter should provide the state estimation covariance that is consistent to its state estimation. We can rewrite Equation (2.26) by the two moment conditions as follow:

$$E[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] \triangleq E[\tilde{\mathbf{x}}_{k|k}] = 0 \tag{2.27}$$

$$E\left[[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}][\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}]^T\right] \triangleq E[\tilde{\mathbf{x}}_{k|k}\tilde{\mathbf{x}}_{k|k}^T] = \mathbf{P}_{k|k} \tag{2.28}$$

where $\tilde{\mathbf{x}}_{k|k}$ is the state estimation error at epoch k. Equation (2.27) is the unbiasedness requirement of the filter estimations and Equation (2.28) is the covariance matching requirement. Based on these two requirements, the consistency criteria of a filter can be defined as follows:

1. The state estimation error should be acceptable as zero mean and have magnitude comparable with the filter-calculated state covariance.

2. The innovation of the filter should also satisfy the same requirement.

3. The innovation should be acceptable as white.

The first criteria can only be tested when the true states are available, either in a simulation or with a very precise ground truth. On the other hand, the last two criteria can be tested in real-world application with real data because the innovation and its covariance are calculated within the filter. Several statistics to test the filter consistency are described in the following subsections.

Normalized Estimation Error Square

To check the consistency of the filter with a hypothesis test, the Normalized Estimation Error Square (NEES) is defined as follow:

$$\epsilon_k = \tilde{\mathbf{x}}_{k|k}^T \mathbf{P}_{k|k}^{-1} \tilde{\mathbf{x}}_{k|k} \tag{2.29}$$

In the hypothesis H_0 that the filter is consistent and Linear-Gaussian the quantity ϵ_k is chi-square distributed with n_x degrees of freedom with n_x equals to the dimension of $\mathbf{x}$. Knowing that the mean of a chi-square distribution equals to its degrees of freedom, we have:

$$E[\epsilon_k] = n_x \tag{2.30}$$

To make the test more statistically representative, it can be based on multiple runs in a Monte Carlo simulations to calculate N independent samples $\epsilon_k^i, i = 1, ..., N$ of the random variable ϵ_k. The average NEES can be calculated as

$$\bar{\epsilon_k} = \frac{1}{N} \sum_{i=1}^{N} \epsilon_k^i \tag{2.31}$$

Under hypothesis H_0, $N\bar{\epsilon}_k$ will follow a chi-square distribution with Nn_x degrees of freedom. The hypothesis that the state estimation errors are consistent with the filter-calculated covariance is accepted if

$$\bar{\epsilon}_k \in [r_1, r_2] \tag{2.32}$$

where the acceptance interval, also known as confidence interval, $[r_1, r_2]$ is selected such that the probability of H_0 being accepted is $1 - \alpha$ where α is a predefined probability of false alarm:

$$P\left(\bar{\epsilon}_k \in [r_1, r_2] | H_0\right) = 1 - \alpha \tag{2.33}$$

The acceptance interval can be calculated from the inverse cumulative distribution function F^{-1} of the chi-squared distribution and normalized by N as follow:

$$r_1 = \frac{1}{N} F^{-1}\left(\frac{\alpha}{2}, Nn_x\right), \; r_2 = \frac{1}{N} F^{-1}\left(1 - \frac{\alpha}{2}, Nn_x\right) \tag{2.34}$$

The larger the number of runs N in the Monte Carlo simulation, the narrower the interval will be, showing the variability reduction in such repeated simulations. If the normalized error statistic is over the upper bound of the acceptance interval, it indicates that the estimate is significantly biased, or the estimation errors are too large compare to the filter-calculated covariance, or the filter-calculated covariance is too small. In this case, the filter is considered to be optimistic. On

the other hand, if the statistic is smaller than the lower bound, it suggests that the covariance is too large. The filter can be considered to be pessimistic.

Normalized Innovation Square

Another useful approach to test if the filter is consistent is to calculate the Normalized Innovation Square (NIS) as follow:

$$\nu_k = \nu_k^T \mathbf{S}_k^{-1} \nu_k \tag{2.35}$$

where ν_k and $\mathbf{S}_k$ are the innovation vector and innovation covariance that are calculated in the update step in the filter as shown in Equation (2.23) and (2.24):

$$\nu_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}) \tag{2.36}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{2.37}$$

Under the hypothesis that the filter is consistent, ν_k follows a chi-square distribution with n_y degrees of freedom, with n_y equals to the dimension of the measurement. Similar to NEES, a Monte Carlo simulation of N independent runs can produce N independent samples of $\nu_k^i, i = 1, ..., N$ and the average NIS can be calculated as:

$$\bar{\nu_k} = \frac{1}{N} \sum_{i=1}^{N} \nu_k^i \tag{2.38}$$

A similar hypothesis test as shown in Equation (2.32) can be performed on the average NIS. The only difference is that the acceptance interval should be determined based on the fact that $N\bar{\nu_k}$ follows a chi-square distribution with Nn_y degrees of freedom. The filter is accepted to be consistent if

$$\bar{\nu_k} \in [r_1, r_2] \tag{2.39}$$

$$r_1 = \frac{1}{N} F^{-1}\left(\frac{\alpha}{2}, Nn_y\right), \; r_2 = \frac{1}{N} F^{-1}\left(1 - \frac{\alpha}{2}, Nn_y\right) \tag{2.40}$$

Innovation whiteness test

The third criteria mentioned earlier can be tested with the whiteness test. It is based on the fact that each element of the innovation vector should be uncorrelated in time if the the innovation can be considered as white. The test statistic of the whiteness test is the time-average autocorrelation which can be calculated as follow:

$$R_{\nu[i]}(j) = \frac{1}{K} \sum_{k=1}^{K} \frac{\nu_k[i]}{\sqrt{\mathbf{S}_k[i,i]}} \frac{\nu_{k+j}[i]}{\sqrt{\mathbf{S}_{k+j}[i,i]}} \tag{2.41}$$

where j is the time lag parameter of the correlation function, K is the number of samples across the time (i.e. epochs), $\nu_k[i]$ refers to the i^{th} innovation element at epoch k, and $\mathbf{S}_k[i,i]$ is the i^{th} diagonal element of the innovation covariance matrix at epoch k. Each term in the summation on the right hand side of Equation (2.41) can be considered as per-measurement normalized innovation. These normalized innovation should follow a standard Gaussian distribution with zero mean and unit variance in a consistent filter. Therefore, based on the central limit theorem, when K is large enough, a normal approximation for the density of $R_{\nu[i]}(j)$ for $j \neq 0$ is convenient and reasonable. In other words, if the innovations are zero-mean and white, then $R_{\nu[i]}(j)$ is distributed $\mathcal{N}(0, \frac{1}{N})$. Similar to the test for NEES and NIS discussed earlier, an acceptance interval is involved in the whiteness test but obtained from an unity-variance normal distribution

instead. Consider a 95% acceptance rate, the hypothesis of the innovation are zero-mean and white is accepted if

$$P\left(R_{\nu[i]}(j) \in \left[-\frac{2}{\sqrt{K}}, \frac{2}{\sqrt{K}}\right]\right) \approx 0.95 \tag{2.42}$$

By performing the consistency test, we can identify potential issue in the filter implementation such as modeling error. The process noise can be tuned to model disturbance and make the filter fulfill the consistent criteria mentioned before.

To solve the unknowns in Equation (2.6), PPP users need to acquire correction information from a global network including: (a) precise satellite orbits, (b) precise satellite clocks, (c) satellite code bias, and (d) satellite carrier-phase bias to achieve high accuracy positioning. Assuming all the corrections are available to users, the observation vector including corrected code and carrier-phase can be expressed as:

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\Phi} \end{bmatrix}, \mathbf{y} \in \mathbb{R}^N \tag{2.43}$$

where N is the number of measurements. To explicitly show the characteristics of carrier-phase ambiguity, a mixed model for the observation vector is introduced:

$$\mathbf{y} \sim \mathcal{N}(h(\mathbf{a}, \mathbf{b}), \mathbf{R}) \tag{2.44}$$

where $\mathbf{a} \in \mathbb{Z}^M$ is the vector of unknown carrier-phase ambiguities, and $\mathbf{b} \in \mathbb{R}^P$ is the vector of other unknowns such as receiver position, receiver clock bias, atmospheric delays, $h(\cdot)$ is the observation model as shown in Equation (2.1), and $\mathbf{R} \in \mathbb{R}^{N \times N}$ is the covariance matrix of the observations.

Based on Equation (2.44), the unknown state vector $\mathbf{x}$ can be defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{b}^T & \mathbf{a}^T \end{bmatrix}^T \tag{2.45}$$

It is practical to describe the evaluation of the unknown state vector overtime with a discrete state-space model using a konwn process model $f(\cdot)$ with the process noise $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} \mathbf{b}_k^T & \mathbf{a}_k^T \end{bmatrix}^T \\ \mathbf{x}_k &= f\left(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}\right) \end{aligned} \tag{2.46}$$

The estimation of state vector $\mathbf{x}_k$ in a non-linear observation model can be formulated as a minimization problem based on the assumed process model and observation model as follow:

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} \left(\|\mathbf{x}_k - f(\hat{\mathbf{x}}_{k-1})\|^2_{\mathbf{P}_k} + \|\mathbf{y}_k - h(\mathbf{x}_k)\|^2_{\mathbf{R}_k} \right) \tag{2.47}$$

The solution of Equation (2.47) can be obtained by a Kalman Filter or robust estimation.

2.2.2 Mixed model estimation problem

GNSS receivers can measure the phase of carrier signals with millimeter accuracy although, unfortunately, they can only track the fractional part of the carrier phase offset, consequently there is an unknown integer number of cycles (the ambiguities) to be estimated. Thus, achieving high precision is subject to the well-known problem of estimating the integer value of the carrier phase ambiguities, commonly referred to as Integer Ambiguity Resolution (IAR). This necessarily leads

to a model that contemplates the joint estimation of both real and integer unknown parameters as presented in Equation (2.47), which in this thesis is referred to as a mixed model. This estimation problem is typically formulated as a Least Squares (LS) problem, which is minimized applying a three-step method decomposition as shown in Figure 2.2. The first step is the estimation of the float solution, which is typically solved as a LS problem without applying the integer constraint on the ambiguities. Since the unknown parameters composing the state vector presented in (2.6) need to be estimated recursively, this LS problem is replaced by an EKF to obtain the float solution. The second step is the estimation of the integer ambiguities based on the float solution by applying the selected IAR method. Nowadays the most popular estimation methods are the Least-squares AMBiguity Decorrelation Adjustment (LAMBDA) [19] for positioning problems. The last step is to obtain the fixed solution, which uses the estimated integer ambiguities to improve the precision of the navigation.

- Step 1: Float Solution
  In this step, the integer nature of the ambiguity is ignored by assuming $\mathbf{a} \in \mathbb{R}^M$ and $\mathbf{b} \in \mathbb{R}^P$. An estimator such as EKF or Robust estimator is used to solve the unknown states and output their corresponding estimation covariance.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{a}} \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} & \mathbf{P}^T_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \\ \mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} & \mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} \end{bmatrix} \tag{2.48}$$

  where $\mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} \in \mathbb{R}^{P\times P}$, $\mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} \in \mathbb{R}^{M\times M}$, and $\mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \in \mathbb{R}^{M\times P}$ represents the covariance matrix between $\hat{\mathbf{a}} \in \mathbb{R}^{\mathbb{M}}$ and $\hat{\mathbf{b}} \in \mathbb{R}^{\mathbb{P}}$. All these matrices can be extracted from the float solution covariance matrix $\mathbf{P}$.

- Step 2: Integer Ambiguity Resolution
  The next step after obtaining float solution is to take the float ambiguity solution and its covariance matrix as input to a mapping function $\mathcal{J}$. This mapping function is used to map the ambiguity states from real-value domain to integer-value domain with the same dimension:

$$\check{\mathbf{a}} = \mathcal{J}_{\mathbf{a}}(\hat{\mathbf{a}}) \tag{2.49}$$

  To achieve this, a decorrelation process of the ambiguities is usually required which is briefly discussed in section 2.2.2.1. Then, a search for the best set of integer candidates is carried out, which is most-widely solved using Integer Least Square (ILS). Finally, a validation process is performed to ensure the reliability of the selected fix candidates by using a ratio test or performing a success rate estimation, which determines whether to accept or reject the fix candidates.

- Step 3: Fixed Solution
  The final step is to use to fixed ambiguities to improve the float solution. The update process to achieve fixed solution is as follow:

$$\check{\mathbf{b}} = \hat{\mathbf{b}} - \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{a}}}\mathbf{P}^{-1}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}\left(\hat{\mathbf{a}} - \check{\mathbf{a}}\right) \tag{2.50}$$

$$\mathbf{P}_{\check{\mathbf{b}}\check{\mathbf{b}}} = \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} - \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{a}}}\mathbf{P}^{-1}_{\hat{\mathbf{a}}\hat{\mathbf{a}}}\mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \tag{2.51}$$

  where $\check{\mathbf{b}}$ is the fixed solution of non-ambiguity parameters, $\mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{a}}} = \mathbf{P}^T_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \in \mathbb{R}^{P\times M}$ is the covariance matrix between non-ambiguity parameters and ambiguity parameters, $\mathbf{P}_{\check{\mathbf{b}}\check{\mathbf{b}}}$ is the covariance of the fixed solution of non-ambiguity parameters.

2.2.2.1 Ambiguity decorrelation

The float ambiguity vector in Equation (2.49) refers to the original ambiguities parameterization which are generally highly correlated and could have low precision. A reparametrization of the ambiguities known as "Z-transformation" can be applied to reduce their correlation and improve the precision of ambiguity resolution. The decorrelated ambiguities $\hat{\mathbf{z}} \in \mathbb{R}^M$ can be obtained as follow:

$$\hat{\mathbf{z}} = \mathbf{Z}^T \hat{\mathbf{a}} \tag{2.52}$$

where $\mathbf{Z} \in \mathbb{Z}^{M \times M}$ is the transform matrix, and the decorrelated float solution is given by

$$\begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{z}} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} & \mathbf{P}^T_{\hat{\mathbf{z}}\hat{\mathbf{b}}} \\ \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{b}}} & \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}} \end{bmatrix} \tag{2.53}$$

where the transformed covariance matrices are

$$\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}} = \mathbf{Z}^T \mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{a}}} \mathbf{Z}, \;\; \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{b}}} = \mathbf{Z}^T \mathbf{P}_{\hat{\mathbf{a}}\hat{\mathbf{b}}} \tag{2.54}$$

and similar to Equation (2.49), the ambiguity fixed solution after decorrelation is obtained as

$$\check{\mathbf{z}} = \mathcal{J}_{\mathbf{z}}(\hat{\mathbf{z}}) \tag{2.55}$$

where $\mathcal{J}_{\mathbf{z}}$ is the mapping function. Once the decorrelated ambiguity is fixed, the remaining non-ambiguity parameters can be updated as follow

$$\check{\mathbf{b}} = \hat{\mathbf{b}} - \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{z}}} \mathbf{P}^{-1}_{\hat{\mathbf{z}}\hat{\mathbf{z}}} (\hat{\mathbf{z}} - \check{\mathbf{z}}) \tag{2.56}$$

$$\mathbf{P}_{\check{\mathbf{b}}\check{\mathbf{b}}} = \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{b}}} - \mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{z}}} \mathbf{P}^{-1}_{\hat{\mathbf{z}}\hat{\mathbf{z}}} \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{b}}} \tag{2.57}$$

where $\mathbf{P}_{\hat{\mathbf{b}}\hat{\mathbf{z}}} = \mathbf{P}^T_{\hat{\mathbf{z}}\hat{\mathbf{b}}}$ and $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}$ are the transformed covariance matrices in Equation (2.54). Note that the back-transformation is not required for updating $\hat{\mathbf{b}}$ because we can use the decorrelated ambiguities and its covariance matrix directly [19].

2.2.2.2 Carrier-phase combination for ambiguity resolution

Fixing the ambiguity directly from single carrier-phase measurement can be challenging. As shown in Equation (2.1), the unknowns in the carrier-phase observation model, such as receiver clock bias, can be correlated and mixed with the ambiguity. It is impossible to separate the noise from different terms and isolate the ambiguity by dealing with a single carrier-phase measurement. One popular approach to this problem is to construct the dual frequency wide-lane (WL) and narrow-lane (NL) carrier-phase combination in cycles as follow:

$$\Phi^s_{WL} = \Phi^s_A - \Phi^s_B \tag{2.58}$$

$$\Phi^s_{NL} = \Phi^s_A - \Phi^s_B \tag{2.59}$$

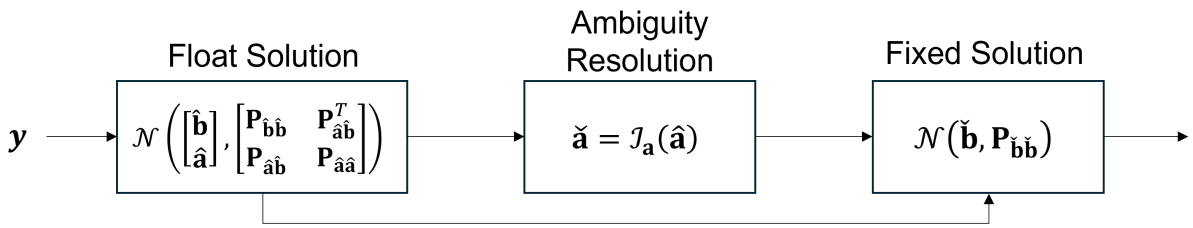


Figure 2.2: Ambiguity resolution flow chart

where Φ_A^s represents the carrier-phase measurement of frequency A. The WL combination has a much larger wavelength than the uncombined single carrier-phase measurement which can make the WL ambiguity easier to fix. However, the large WL wavelength also limits the possible positioning accuracy. Therefore, one can use the NL combination, which has a shorter wavelength than the original measurement, to further improve the positioning accuracy.

Nevertheless, it has been shown that this cascade way of fixing the ambiguity with WL-NL combination might limit the amount of fixed ambiguity and its success rate [20]. It is recommended to fix the ambiguity by using the raw ambiguity in a way similar to real-time kinematic (RTK) technique to achieve higher number of fixed ambiguity, faster convergence speed and better positioning accuracy in PPP-AR. The between satellite single-difference ambiguity can be formed to remove receiver-dependent bias and recover the ambiguity's integer nature. Given the float state estimation vector and covariance at epoch k, the single-difference ambiguity can be formed as [20]:

$$\hat{\mathbf{x}}_{k,SD} = \mathbf{C}_k \hat{\mathbf{x}}_k \tag{2.60}$$

$$\mathbf{P}_{\hat{\mathbf{x}}_{k,SD}\hat{\mathbf{x}}_{k,SD}} = \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^T \tag{2.61}$$

where $\mathbf{C}_k$ is the matrix that transforms the undifferenced ambiguities to between-satellite single-difference ambiguities and keeps the remaining states unchanged. The reference ambiguity, or pivot ambiguity, that other ambiguities are subtracted with can be selected by choosing the one with the lowest estimation uncertainty. Once the state vector with single-difference ambiguity and its covariance matrix is obtained, they can be considered as the new float solution in first step of LAMBDA and proceed through the remaining steps to achieve fixed solution.

2.3 Robust estimation

EKF can provide optimal estimation based on the assumption of Gaussian distributed measurement and process noise. However, this assumption is not easy to fulfill in real-world environment due to measurement outliers or model mismatch, which can dramatically reduce the performance of the filter. Therefore, Robust estimation is introduced to overcome this limitation and improve the robustness of the iterative filter. The robust estimation is widely used in the robotics field and starts to gain attention in the GNSS applications [16, 21]. Instead of using the measurement noise covariance $\mathbf{R}_k$, a re-weighted covariance matrix $\bar{\mathbf{R}}_k$ is used to solve the estimation problem. Equation (2.47) can be re-written as follow:

$$\hat{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} \left(\|\mathbf{x}_k - f(\hat{\mathbf{x}}_{k-1})\|^2_{\mathbf{P}_k} + \|\mathbf{y}_k - h(\mathbf{x}_k)\|^2_{\bar{\mathbf{R}}_k} \right) \tag{2.62}$$

The re-weighted covariance matrix is defined as:

$$\bar{\mathbf{R}}_k = \mathbf{R}_k^{1/2} \mathbf{W}_y^{-1} \mathbf{R}_k^{T/2} \tag{2.63}$$

where Choloskey decomposition is used on $\mathbf{R}_k = \mathbf{R}_k^{1/2}\mathbf{R}_k^{T/2}$, and the weighting matrix is calculated as follow:

$$\mathbf{W}_y = \text{diag}\left[w\left(\mathbf{R}_k^{-1/2} \left(\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}) \right) \right) \right] \tag{2.64}$$

where diag[·] represents the diagonal elements in the matrix, and w is a weighting function derived from the corresponding score function ψ. The weighting function can lower the weight of measurements with large innovations, such as outliers, and thus reduce their effect in the

estimation problem. Based on the required robustness level, several weighting function can be chosen to have different down weighting effect on large residual measurements. One of the most widely used weighting function is the Huber Loss function whose weighting function and score function are defined as follow:

$$\psi_{\text{Hub}}(x) = \begin{cases} x, & \text{if } |x| \leq c_{\text{Hub}}, \\ c_{\text{Hub}}\ \text{sign}(x), & \text{if } |x| > c_{\text{Hub}}, \end{cases} \qquad w_{\text{Hub}}(x) = \begin{cases} 1, & \text{if } |x| \leq c_{\text{Hub}}, \\ \dfrac{c_{\text{Hub}}}{|x|}, & \text{if } |x| > c_{\text{Hub}}. \end{cases} \tag{2.65}$$

where c_{Hub} is a control parameter to control the robustness and efficiency. A comprehensive description of different weighting function and scoring function can be found in [6].

One potential issue of using Equation (2.63) is the inverse of the weighting matrix which could lead to numerical singularity problem when any of its element is close to zero. This could happen when there is a measurement with very large residual. One way to address this issue is to use the Information Filter which is equivalent to the Kalman Filter operating with the inverse form of the covariance matrix [21]. The update step in the Information Filter can be expressed as follow:

$$\mathbf{z}_{k|k} = \mathbf{z}_{k|k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-T/2} \mathbf{W}_y \mathbf{R}_k^{-1/2} \mathbf{H}_k \big(\mathbf{y}_k - h(\mathbf{x}_{k|k-1}) + \mathbf{H}_k \mathbf{x}_{k|k-1}\big) \tag{2.66}$$

$$\mathbf{Z}_{k|k} = \mathbf{Z}_{k|k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-T/2} \mathbf{W}_y \mathbf{R}_t^{-1/2} \mathbf{H}_t \tag{2.67}$$

where $\mathbf{z}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1} \mathbf{x}_{k|k-1}$ is the information vector derived from the state prediction and covariance of a standard Kalman Filter, and $\mathbf{Z}_{k|k-1} = \mathbf{P}_{k|k-1}^{-1}$ is the information matrix. The inverse operation of $\mathbf{W}_y$ can therefore be avoided using Information Filter update and maintain the robustness of the filter against large residual measurements. The update of the information vector and information matrix is performed iteratively within the same epoch until the termination criteria is reached. The termination criteria can be a limited number of iteration or when the change in to states is lower than a predefined threshold.

The relative efficiency can be used to measure the "near optimality" of the robust filter with respect to the optimal EKF under ideal condition, i.e., observation and state dynamic model are perfectly known [6]. In the position estimation problem, the Root Mean Square Error (RMSE) of the position estimation can be used to calculate the relative efficiency of the robust filter as follow:

$$\text{Eff}\left(\hat{\mathbf{x}}_{\text{Robust}}\right) = \frac{\text{RMSE}(\hat{\mathbf{x}}_{\text{optimal EKF}})}{\text{RMSE}(\hat{\mathbf{x}}_{\text{Robust}})} \tag{2.68}$$

with $0 \leq \text{Eff}\left(\hat{\mathbf{x}}_{\text{Robust}}\right) \leq 1$. The relative efficiency is usually obtained by N Monte Carlo simulations which allows the calculation of the RMSE of position estimation using:

$$\text{RMSE}(\hat{\mathbf{x}}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(e_{E,i}^2 + e_{N,i}^2 + e_{U,i}^2\right)} \tag{2.69}$$

where $e_{E,i}, e_{N,i}, e_{U,i}$, refers to the position error in East-North-Up (ENU) direction respectively.

Chapter 3

Integrity Monitoring for Precise Positioning

In safety critical application with GNSS, such as aviation and automotive, integrity monitoring allows the user to use the system safely and make the decision to continue or stop normal operation when necessary. In this chapter, the background knowledge required to understand the concept of integrity monitoring is introduced, followed by a discussion of the techniques for integrity monitoring, with the focus on precise positioning. Finally, different approaches for protection level computation are presented.

3.1 Vocabulary

Several important parameters and definition related to integrity monitoring are presented below:

- Integrity: The level of trust that can be put on the information provided by the navigation system. It includes the system's ability to detect faults and provide timely warning to users.
- Continuity: The system's ability to maintain specified performance without unscheduled interruption during the intended operation period.
- Alert Limit (AL): The error tolerance of a given parameter measurement that should not be exceeded without providing an alert. It is an external requirement of a navigation system.
- Time To Alert (TTA): The maximum time allowed between the navigation system going out of tolerance and issuing an alert. It is an external requirement of a navigation system.
- Integrity risk: The probability that the position error exceeds the alert limit and the system fails to provide timely warning to the users. It is quantified by the probability of Hazardously Misleading Information (HMI), P_{HMI}.
- Continuity risk: The probability that the system fails to maintain specified performance without interruption during intended operation period.
- Protection Level (PL): A statistical error bound calculated by the receiver to ensure that the probability of the absolute position error exceeding this value is less than or equal to the target integrity risk. PL is a function of the geometry between satellite and user, and the expected error characteristics.

- Integrity event: Occurs whenever the Position Error (PE) exceeds the corresponding PL.
- Integrity Failure: Occurs when an integrity event lasts longer than the TTA without an alarm being triggered during that period. The probability of integrity failure is the integrity risk.
- Stanford diagram: A useful tool to visualize and explain the concept of integrity events, including Misleading Information (MI) and HMI events. A generic Stanford diagram is shown in Figure 3.1a. Note that the integrity events are located in the area below the diagonal dashed-line.
- Probability of false alarm: The probability that the fault detection algorithm declares a fault when no fault actually exists. It is related to the continuity risk as the false alarm unnecessarily disrupt the system's continuity even when it is fault-free. The probability of false alarm can be expressed mathematically as:

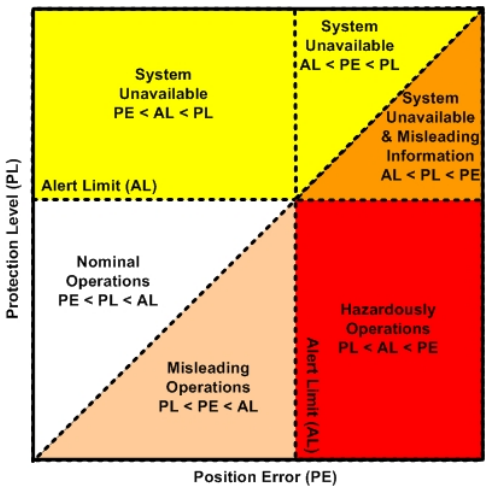
$$P_{fa} = P\{\text{test statistic} > \text{test threshold} \,|\, H_0\} \tag{3.1}$$

  where H_0 is the hypothesis that no fault exist. The area of P_{fa} is shown with a simple Gaussian hypothesis testing in Figure 3.1b. The value of P_{fa} is required to decide the threshold in a fault detection algorithm.
- Probability of missed detection: The probability that the algorithm fails to detect an actual fault. It is related to the integrity risk as the system let a fault go undetected and potentially provide user incorrect solution. The probability of missed detection can be expressed as:

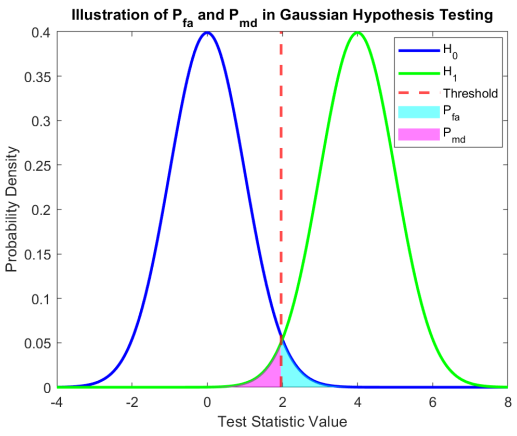
$$P_{md} = P\{\text{test statistic} < \text{test threshold} \,|\, H_1\} \tag{3.2}$$

  where H_1 is the hypothesis that fault exists. The area of P_{md} is shown with a simple Gaussian hypothesis testing in Figure 3.1b.

With the parameters and definitions mentioned above, a set of algorithms and processes can be designed to ensure that the user receives an alert within specified time if the navigation system should not be used for intended operation, either due to fault or excessive position error. This is the idea of integrity monitoring for a navigation system. Several integrity monitoring techniques for precise positioning are presented in the following section.



(a) Stanford diagram



(b) P_{fa} and P_{md}

Figure 3.1: Illustration of Stanford diagram and P_{fa}, P_{md}

3.2 Integrity Monitoring techniques

The well known Receiver Autonomous Integrity Monitoring (RAIM) is the classic technique to perform integrity monitoring based on GNSS code measurements. It considers fault-free and single-fault hypothesis. Under the fault-free hypothesis the pseudorange measurement are modeled by adding a error vector ϵ with zero-mean independent Gaussian distribution and a known covariance $\mathbf{\Sigma}$. Under the single-fault hypothesis, one measurement is modeled to be biased from this single fault. The least square residual can be used as test statistic to detection the single fault as follow:

$$q = \frac{\|r\|^2}{\sigma^2} \tag{3.3}$$

where r is the pseudorange measurement residual calculated by subtracting the predicted measurement from the observed measurement, and σ is the standard deviation of the pseudorange measurement errors.

The test statistic should follow a central chi-square distribution in the fault-free hypothesis, and follow a non-central chi-square distribution with in the single-fault hypothesis.

$$q \sim \chi^2(\mathrm{k}, 0), \text{ if } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{3.4}$$

$$q \sim \chi^2(\mathrm{k}, \lambda), \text{ if } \epsilon \sim \mathcal{N}(\mathbf{b}, \mathbf{\Sigma}) \tag{3.5}$$

where k is the degrees of freedom of the chi-square distribution which equals to the number of redundant measurements, λ is the non-centrality parameter of the chi-square distribution, ϵ is the measurement error vector, and $\mathbf{b}$ is the fault bias vector.

Using the central chi-square distribution and a desired probability of false alarm, a test threshold T can be defined with the chi-square inverse Cumulative Distribution Function (CDF) F^{-1} as follow:

$$T = F^{-1}_{\chi^2}(1 - P_{fa}, 0) \tag{3.6}$$

With a given probability of missed detection P_{md} and the obtained threshold T, the minimum detectable fault bias represented by λ_{bias} can calculated iteratively such that:

$$P_{md} = \int_0^T f_{\chi^2_{k,\lambda_{\text{bias}}}}(x)\, dx \tag{3.7}$$

where $f_{\chi^2_{k,\lambda_{\text{bias}}}}(x)$ is the probability density function of the non-central chi-square distribution.

Figure 3.2 shows the relationship between the probability of false alarm, test threshold, and probability of missed detection. We know that the probability density function of non-central chi-square distribution will shift to the right as the non-centrality parameter λ increases. With a fixed test threshold, we can change λ to find its proper value that leads to the desired probability of missed detection.

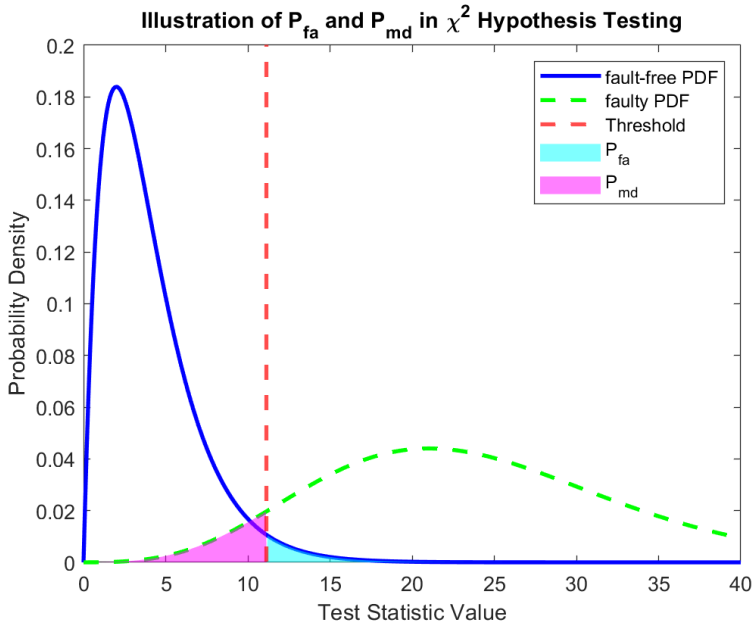


Figure 3.2: Relationship between P_{fa}, test threshold, and P_{md} using χ^2 test

After λ_{bias} is obtained, the slope parameter is introduced as a measure of the coupling between the effect of measurement bias in the observable parameter, i.e. the test statistic's $\sqrt{\lambda}$, and what RAIM tries to protect, the position error projected by the fault bias [22]. Their relation can be expressed as:

$$\text{position error} = \text{slope}_i \sqrt{\lambda} \tag{3.8}$$

where the slope is expressed as:

$$\text{slope}_i = \sqrt{\frac{(\mathbf{H}^+_{N,i})^2 + (\mathbf{H}^+_{E,i})^2}{\mathbf{S}_{i,i}}} \tag{3.9}$$

where the subscript i refers satellite i, $\mathbf{H}^+ = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T$ is the pseudo inverse of the design matrix $\mathbf{H}$ in the ENU frame, and $\mathbf{S}_{i,i}$ is the i^{th} diagonal element in the projection matrix $\mathbf{S} = \mathbf{I} - \mathbf{H}\mathbf{H}^+$. Each satellite has its own slope, and the satellite with the highest slope is the most difficult to detect. This is because, for a given position error, the satellite with the highest slope produces the smallest value of $\sqrt{\lambda}$, which in turn results in the highest probability of missed detection.

The Horizontal Protection Level (HPL) is then calculated as the projection in the position domain of the faulty measurement bias in the satellite that has the largest slope as follow:

$$\text{HPL} = \sigma\, \text{slope}_{max} \sqrt{\lambda_{bias}} \tag{3.10}$$

Ideally, the calculated HPL should bound the horizontal position error with an adequate margin; otherwise, an integrity event occurs.

By looking at the classical RAIM technique, we can identify the process of integrity monitoring into the following steps:

- Solve the estimation problem with an estimator, for example: Least Square in RAIM.
- Fault detection with statistical test and potentially perform fault exclusion.

- Estimation error bounding with the computation of protection level.

Based on the concept of RAIM, the integrity monitoring techniques for iterative estimator, i.e. EKF and robust estimator mentioned in previous chapter, are presented in the following sections.

3.2.1 EKF with Innovation based Fault Detection and Exclusion (EKF-FDE)

A simple Fault Detection and Exclusion process is combined with the EKF with single fault assumption. The flow chart of the EKF-FDE scheme is shown in Figure3.3. The EKF is initialized with prior knowledge of the state and its covariance.

At the beginning of each epoch k iteration, the state prediction and measurement update considering all-in-view solution are performed. After the measurement update, a fault detection based on chi-square test for all-in-view solution is applied. The test statistic q_k is selected by using the NIS of the filter which is calculated with Equation (2.35) as follow:

$$q_k = \nu_k^T \mathbf{S}_k^{-1} \nu_k \tag{3.11}$$

The test threshold is calculated from the inverse CDF of a chi-square distribution $F_{\chi^2}^{-1}$ as follow:

$$T_{\chi^2} = F_{\chi^2}^{-1}(1 - P_{fa}, \text{dof}) \tag{3.12}$$

with the degrees of freedom (dof) equals the dimension of the innovation vector and a desired probability of false alarm P_{fa}. The test is considered pass when the test statistic is below the threshold $q_k < T_{\chi^2}$ which indicates that the filter is consistent from the innovation point of view.

If the all-in-view chi-square test passes, it means that there is no fault detected and the all-in-view solution is outputted for the current epoch. If the all-in-view chi-square test fails, it means that a fault is detected and a fault exclusion is required. The fault exclusion is performed by removing all the measurements from the satellite which has the largest measurement innovation value from the all-in-view subset. For example, if satellite 1 has the largest innovation in its L1 code measurement, then all the measurements (both code and phase on all frequencies) from satellite 1 are excluded.

A new single-fault subset is formed after excluding the detected faulty satellite from the all-in-view subset. The chi-square test is applied again on the single-fault subset by calculating the test statistic with Equation (3.11) using the measurements excluding the detected faulty satellite. Note that the test threshold for the single-fault subset needs to be recalculated by using the dimension of the new innovation vector. If the single-fault chi-square test passed, it means that the single fault exclusion is successful. The filter performs measurement update again by using the measurements in the single-fault subset and outputs the solution. If the single-fault chi-square test fails, it means that the single fault exclusion is unsuccessful. In this case, the filter only outputs the predicted states and the solution of this epoch is declared unavailable.

This FDE scheme is simple and straight forward but with the limitation of being able to deal with single fault only. To deal with more than one fault, a more complex fault detection and exclusion approach is required as discussed in the next subsection.

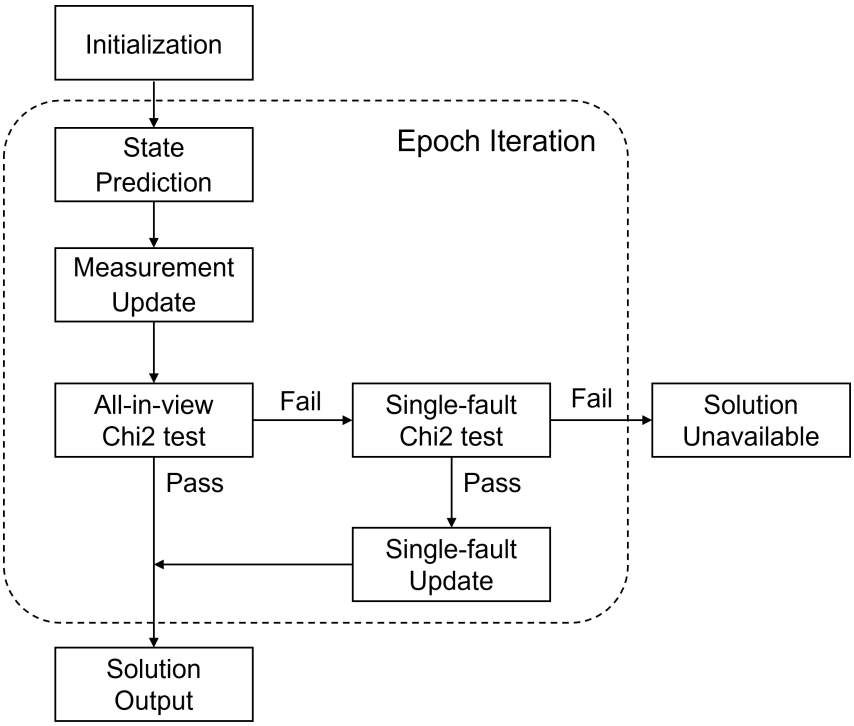


Figure 3.3: Flow chart for EKF with a single Fault Detection and Exclusion

3.2.2 Multiple Hypothesis Solution Separation (MHSS)

The MHSS is a popular technique which is capable of detecting and excluding multiple faults. The flow chart of the MHSS scheme is shown in Figure 3.5.

The first step in performing MHSS is to determine the fault models or subsets in the multiple hypothesis. This includes the probability of fault and the maximum number of simultaneous faults need to be monitored. The former is usually determined based on prior knowledge of the type of faults while the latter can be decided with the following equation [1, 15]:

$$N_{\text{max-event}} = \min[P(N_{\text{number of event}} > N_{\text{max-event}}) < P_{\text{threshold}}] \tag{3.13}$$

where $P(N_{\text{number of event}} > N_{\text{max-event}}) = P_{\text{not-monitored}}$ is the probability of not monitored faults. Equation (3.13) means that we want to find the $N_{\text{max-event}}$ such that the probability of not monitored faults is lower than a predefined probability threshold which is the system's requirement of integrity risk.

Once $N_{\text{max-event}}$ is decided, we can define the subsets (or hypothesis) based on all the possible combinations of faults among all the satellites and construct the bank of filters accordingly. Given n satellites in a single epoch, the total number of subsets is calculated with the following equation:

$$N_{total} = 1 + N_s = 1 + \sum_{i=1}^{N_{\text{max-event}}} C_i^n \tag{3.14}$$

where the 1 on the right-hand side of Equation (3.14) represents the all-in-view solution which is defined as subset 0 (null hypothesis) that all satellites in-view are fault-free, and $C_i^n = \frac{n!}{(n-i)!\,i!}$ returns the binomial coefficient for the number of combinations of n items taken i at a time.

An example of the subsets with $n = 5$ and $N_{\text{max-event}} = 2$ is given in Figure 3.4 where the green dots are the satellites assumed to be fault-free while the red dots are the ones assumed to be

faulty. By using Equation (3.14), we can get the total number of subsets as $1+\frac{5!}{(5-1)!1!}+\frac{5!}{(5-2)!2!}=16$. The number of subsets will increase exponentially with more available satellites and more simultaneous faults to be considered in a single epoch.

Tot. SV	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}
sv1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
sv2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
sv3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
sv4	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
sv5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Figure 3.4: All the multiple hypothesis with $n = 5$ and $N_{\text{max-event}} = 2$, green dots represents fault-free satellites, red dots represents faulty satellites.

Once the subsets are determined, a bank of EKFs are constructed accordingly. These bank of EKFs are initialized with the same setup for their initial states and covariance. Note that elements in the state vector need to match the considered satellites in each subset. Take the subset 0 and subset 1 in Figure 3.4 for example, the state vector in subset 0 will includes the ionospheric delay and ambiguity terms from sv1 to sv5 while subset 1 will only includes the ones from sv2 to sv5.

After subsets initialization, the state prediction and measurement update for each subset are performed independently in each epoch iteration by using the measurements from the considered satellites. At this point, the state estimate and covariance matrix of PPP float solution for each subset are obtained.

The fault detection process is applied next to decide which subset provides the solution that can be considered as fault-tolerant (fault-free) and outputted to user. Firstly, a fault detection test is applied to subset 0 (all-in-view subset). If subset 0 passes the test, the all-in-view solution is considered as fault-tolerant and provided to the user. Otherwise, a fault is detected in the all-in-view subset and the fault detection tests for the remaining subsets (subset 1 to subset N_s) are performed.

If one of the remaining subset can pass the test, it suggests that this subset successfully excludes the faulty satellite and is considered as the fault tolerant solution. One the other hand, if all the remaining subsets fail the test, it suggests that non of the subsets can exclude the faulty satellite correctly. In this case, only the predicted states and covariance is outputted and the solution at this epoch is declared unavailable. This indicates that the fault presents in the all-in-view subset is not monitored in the fault models, for example: more than $N_{\text{max-event}}$ satellites are faulty.

There are two approaches available to perform the fault detection test in the MHSS scheme: the chi-square test and the solution separation test. The chi-square test for MHSS is slightly different from the ones mentioned earlier and therefore will be further discussed in the following subsection.

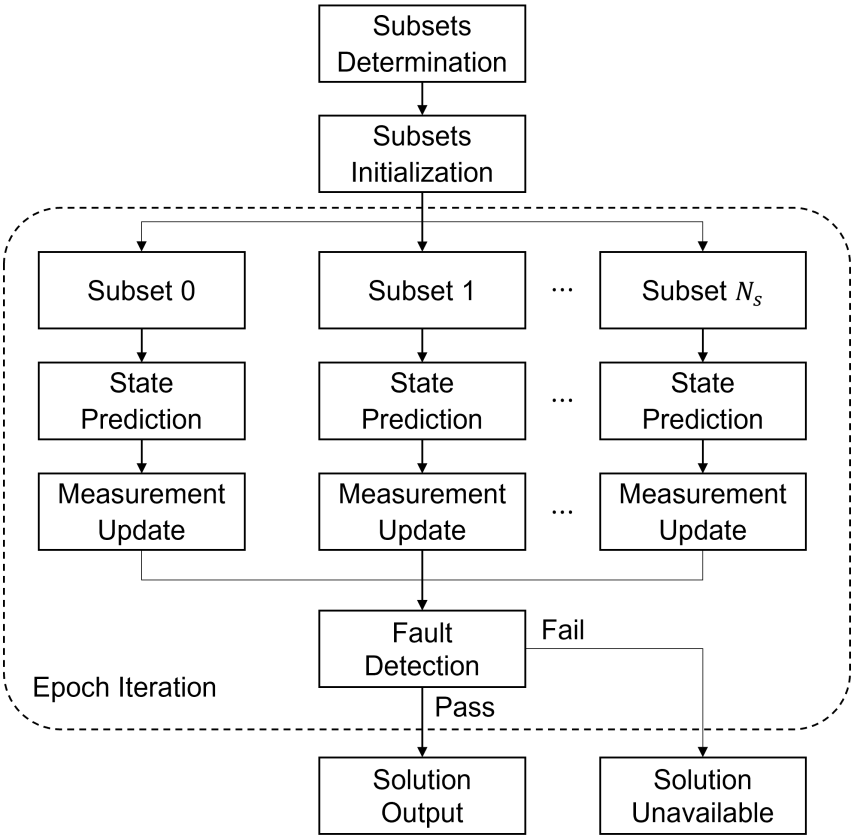


Figure 3.5: Flow chart for Multiple Hypothesis Solution Separation

3.2.2.1 Chi-square test

The chi-square test is based on the assumption that the measurement noise and process noise in the EKF are Gaussian distributed. The same test statistic in EKF-FDE, as shown in Equation (3.11), can also be used in the MHSS scheme. The difference between MHSS and EKF-FDE is that the former has different test statistic for each subset which is calculated as follow:

$$q_k^{(i)} = \left(\nu_k^{(i)}\right)^T \left(\mathbf{S}_k^{(i)}\right)^{-1} \nu_k^{(i)} \tag{3.15}$$

where the super script $q_k^{(i)}$, $\nu_k^{(i)}$, and $\mathbf{S}_k^{(i)}$ refers to the test statistic, innovation vector, and innovation covariance matrix of subset i respectively.

Under the null hypothesis, i.e. the all-in-view solution is fault-free, the test statistic should follow a central chi-square distribution [23]:

$$H_0 : q_k^{(0)} \sim \chi^2(\text{dof}, 0) \tag{3.16}$$

where the dof $= \dim\left(\nu_k^{(0)}\right)$ is the degree of freedom equals to the dimension of the innovation vector for all-in-view subset, and the second parameter 0 indicates that the chi-square distribution is central. Once the test statistics for the null hypothesis is obtained, it is compared with the corresponding threshold $T_{\chi^2}^{(0)}$ which is determined by the following equation similar to Equation (3.12) with the correct degree of freedom for the inverse chi-square CDF :

$$T_{\chi^2}^{(0)} = F_{\chi^2}^{-1}\left(1 - P_{fa}, \text{dof} = \dim\left(\nu_k^{(0)}\right)\right) \tag{3.17}$$

The null hypothesis is accepted if the test is passed, i.e. $q_k^{(0)} < T_{\chi^2}^{(0)}$. Otherwise, the test statistics of the remaining subsets ($q_k^{(i)}, i = 1, ..., N_s$) are considered and compared with the corresponding threshold ($T_{\chi^2}^{(i)}, i = 1, ..., N_s$) to find out which subset can pass the test, i.e. $q_k^{(i)} < T_{\chi^2}^{(i)}$. Note that the threshold of the remaining subsets (excluding the all-in-view subset) should be determined by the chi-square distribution with the correct degrees of freedom which equals to the dimension of the innovation vector of the subsets, i.e, dof $= \dim(\nu_k^{(i)})$.

The MHSS with the chi-square test can perform multiple fault detection and exclusion inherently. However, it comes with the drawback of implementation complexity and heavy computation load. In the next section, the robust estimation with EKF is introduced. It can be less computationally demanding than MHSS while being able to deal with multiple faults.

3.2.3 Robust Kalman Filter (RKF)

As discussed in section 2.3, the Robust Kalman Filter (RKF) uses a re-weighted measurement covariance matrix to mitigate the effect of measurements with large innovation. Unlike EKF-FDE and MHSS mentioned above, the RKF does not exclude any measurement, even the ones that are potentially faulty. The RKF flow chart is shown in Figure 3.6.

The initialization and state prediction are carried out in the same way as a traditional EKF. Within each epoch, the measurement update step is combined with a robust re-weighting process and performed iteratively using the information form as shown in Equations (2.67) and (2.66) until the termination criteria are reached. This robust estimation scheme can reduce the affect of multiple measurement faults on the estimation because it does not relay on any prior assumption of the error. However, this also means that it is difficult to have a formal statistical bound to guaranty that the level of trust can be put on the estimated solution because we do not know the distribution of the measurement noise or error. In other words, the classical chi-square test may not be valid with the robust estimator. Therefore, the RKF will always output a solution after the re-weighting process is converged. Although the chi-square test cannot be applied, we can still calculate the protection level and compare with the position estimation error in simulation to evaluate the integrity performance of the robust estimator.

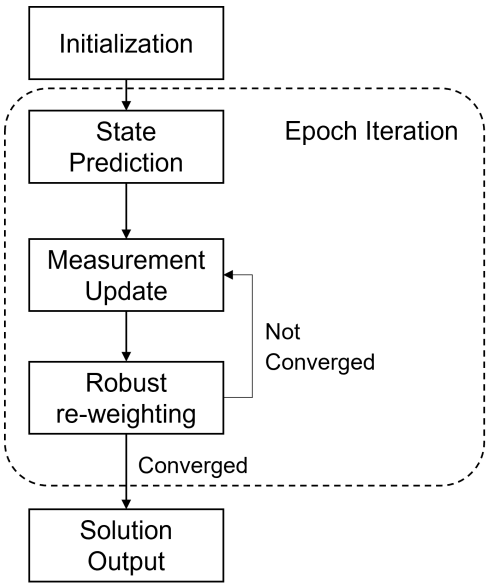


Figure 3.6: Flow chart for RKF

3.3 Protection Level calculation

Once the filter passes the fault detection test, the next step in integrity monitoring is to calculate the PL, which provides a statistical bound for position error. It is determined by the following aspects [11]:

- the threat model
- the nominal model
- the test statistic
- the integrity risk allocation
- the upper bounds used to simplify the computation

The PL is defined with these factors such that:

$$P\{\text{position error} > \text{PL} | \text{test passes}\} \leq \text{Integrity risk allocation} \tag{3.18}$$

In this thesis, the slope-based approach is adapted to obtain the PL using a EKF with both pseudorange code and carrier phase measurement. The slope in the horizontal direction is calculated as follow [24]:

$$\text{Hslope}_{k,i} = \sqrt{\frac{\mathbf{f}_i^T \mathbf{K}_k^T \tau_E^T \tau_E \mathbf{K}_k \mathbf{f}_i + \mathbf{f}_i^T \mathbf{K}_k^T \tau_N^T \tau_N \mathbf{K}_k \mathbf{f}_i}{\mathbf{f}_i^T \mathbf{S}_k^{-1} \mathbf{f}_i}} \tag{3.19}$$

where the subscript k refers to epoch k, $\mathbf{f}_i$ is a fault profile vector filled with zeros except the corresponding satellite i under evaluation, τ is a vector retrieving the interested state from a full state vector, and $\mathbf{S}$ is the innovation covariance. Each measurement has its own slope and the corresponding measurement noise σ_i. The measurement with the largest value of $(\text{Hslope} \cdot \sigma_i)$ is selected to compute the PL as follow:

$$\text{HPL} = \max(\text{Hslope}_i \cdot \sigma_i)\sqrt{\lambda_k} + k(P_{md})\sqrt{\tau_E^T \mathbf{P}_{k|k} \tau_E + \tau_N^T \mathbf{P}_{k|k} \tau_N} \tag{3.20}$$

where λ_k is obtained in the same way as the λ_{bias} in Equation (3.10), $k(P_{md})$ is a factor calculated from the normal inverse CDF and a desired probability of missed detection, and $\mathbf{P}_{k|k}$ is the state estimation covariance matrix. Equation (3.19) and (3.20) allow us to establish the relationship between P_{fa}, test threshold T, P_{md}, and λ in an iterative EKF and obtain the HPL for integrity monitoring.

Chapter 4

Implementation and Experiment Setups

To evaluate the integrity monitoring performance of the techniques presented in chapter 3, a PPP integrity monitoring software is implemented in MATLAB for simulation. The advantage of simulation is that we can have full control and prior knowledge to the filter configuration and the injection of faults. This allows us to have the ground truth for parameters that are difficult to access in real-world data such as the carrier-phase ambiguity and faults inside the filter. In this chapter, the detail of the software simulation structure is presented, followed by the types of filters considered in the simulation. Important parameters and stochastic models used in the simulation and filter configuration are also provided. Finally, the scenarios with different fault configurations evaluated in this thesis are presented.

4.1 Software Simulation Structure

There are three major scripts in the MATLAB software simulation in this thesis, including *RUN_Generate_Scenario*, *RUN_Gererate_Montecarlo*, and *RUN_Evaluate_Montecarlo*. The flow chart of the implementation is shown in Figure 4.1 with inputs and outputs in each script. More detailed description for each block are provided as follow:

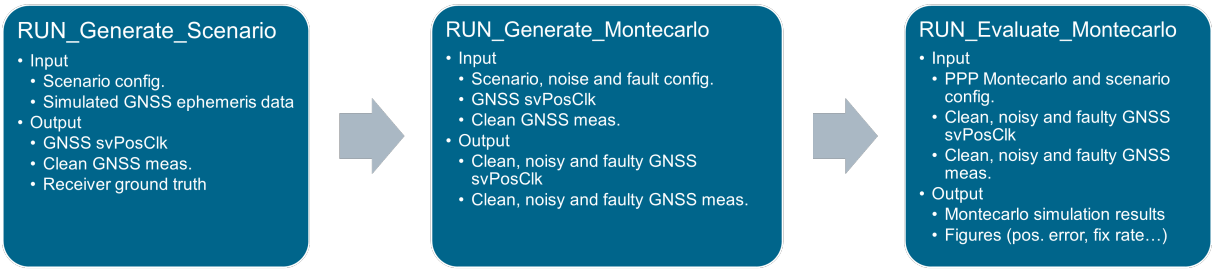


Figure 4.1: Flow chart for simulation implementation in MATLAB

- *RUN_Generate_Scenario*: The purpose of this script is to generate the considered scenario for PPP integrity monitoring. The inputs of this script are:

1. Scenario configuration file: It contains the required information to generate a simulated receiver states such as secnario name, number of epochs, receiver dynamics parameters, initial states and their covariance. It also contains the setups to generate the simulated GNSS ephemeris data including a RINEX file name and the considered GNSS constellation.
2. Simulated GNSS ephemeris data file: The GNSS-related configuration in the scenario configuration file is used in the DLR's Multi GNSS simulator Tool to perform orbits computation from the broadcasted ephemeris in brdc RINEX files for GNSS satellites.

The script processes these two input files and outputs the following data:

1. GNSS *svPosClk*: It contains the position, velocity, and clock parameters for the satellite in-view at each epoch generated from the DLR's Multi GNSS simulator Tool.
2. Receiver ground truth: It contains the ground truth for receiver dynamic states including position, velocity, and clock generated according to the receiver's dynamic configuration. It also contains satellite-related ground truth states such as ionospheric delay, tropospheric delay, tropospheric mapping function, and carrier-phase ambiguities.
3. Clean GNSS measurements: By combining the GNSS *svPosClk* and receiver ground truth states, clean GNSS pseudorange code and carrier-phase measurements are generated based on the observation model described in section 2.1. The 'clean' measurement suggests that neither measurement noise nor faults are injected to the measurements at this stage.

These outputs are saved as .mat files for further processing in the next script.

- *RUN_ Generate_ Montecarlo*: This script is used to generate the measurement data to be processed in PPP integrity monitoring Monte Carlo simulations. The inputs of this script are:

  1. Scenario, noise and fault configuration files: The scenario configuration file is the same one as the input in the *RUN_ Generate_ Scenario* script. It is used in this script to simply pass down the scenario configuration. The noise and fault configuration file contains the noise profile for measurement such as the noise ratio between code and phase, code measurement noise in zenith direction, and the weighting parameters for code measurement as shown in Equation (2.19). It also contains the number of Monte Carlo simulation intended to be run. The noise and fault for satellite position and satellite clock are also considered to simulate the uncertainty in ephemeris. Finally, the fault configurations intended to be injected into measurements are defined including the types of faults, the starting and end time of the faults, effected satellites, effected measurement frequency, effected observation type, and the magnitude and bias of the fault.
  2. GNSS *svPosClk*: The output from the *RUN_ Generate_ Scenario* script.
  3. Clean GNSS measurements: The output from the *RUN_ Generate_ Scenario* script.

  The script processes these input files and outputs the following data:

  1. Clean, noisy and faulty GNSS *svPosClk*: The ephemeris noise and fault are injected in the 'clean' GNSS *svPosClk* based on the configuration file. To separate the noisy data from the noisy and faulty data, two variables are created separately. One of them is the clean GNSS *svPosClk* injected with noise only, the other one is the clean GNSS *svPosClk* injected with noise and fault.
  2. Clean, noisy and faulty GNSS measurements: Similar to the ephemeris data, measurement noise and faults are injected into the 'clean' GNSS measurements, which are then separated into noise-only and noise-plus-fault datasets.

These output data are generated independently for each Monte Carlo simulations and saved as .mat files. They are then used in the final script to perform PPP integrity monitoring Monte Carlo simulations.

- *RUN_ Evaluate_ Montecarlo*: The final script contains the core part of the whole simulation. It includes measurement pre-processing, the computation of integrity parameters, PPP computation with different types of filters using different integrity monitoring techniques, estimation error analysis, integrity monitoring evaluation, and result visualization and saving. The inputs of this script are:

  1. PPP Monte Carlo and scenario configuration: The PPP Monte Carlo configuration file contains the name of considered scenario and test case. It also contains the simulation setups including the flag for ambiguity fixing and PL computation, the method to be used for ambiguity resolution, the $N_{\text{max-event}}$ in MHSS, and the number of re-weighting iteration in Robust filtering. Integrity related parameters such as P_{fa}, P_{md}, and alert limits are also defined in this file. The scenario configuration is again used to pass down the scenario configuration.
  2. Clean, noisy and faulty GNSS *svPosClk*: Outputs from *RUN_ Generate_ Montecarlo* script.
  3. Clean, noisy and faulty GNSS measurements: Outputs from *RUN_ Generate_ Montecarlo* script.

  The script then uses these inputs to run to Monte Carlo simulation for intended number of runs and outputs the following:

  1. Monte Carlo simulation results: Including the estimated states for float solution, estimated states for fixed position estimation (if enabled), horizontal and vertical PLs (if enabled), and state estimation errors.
  2. Figures: To visualize the the results of Monte Carlo simulation, including position estimation comparing with PLs in horizontal and vertical direction, the Stanford Diagram for integrity monitoring performance evaluation, ambiguity fixing results, etc.

Now we have the high level view of the structure of the simulation for PPP integrity monitoring. In the next section, the core part of the types of filters considered in *RUN_ Evaluate_ Montecarlo* script are discussed in greater detail.

4.2 Types of Filters

There are four types of filters considered in this thesis including EKF, EKF-FDE, MHSS, and RKF. In the MATALB implementation, they are defined as separate sub-classes under a super class *Filter*, as shown in the MATLAB class diagram in Figure 4.2. This structure allows all filters to share some common properties, such as the state vectors and measurement vectors, and methods, such as Kalman Gain computation and innovation computation. It also gives them the flexibility to define their own specific ones, such as the subset of filters in MHSS, fault detection and exclusion in EKF-FDE, and the re-weighting of observation covariance in RKF. Each type of filters are introduced in detail with their pseudo-code in the following subsections.

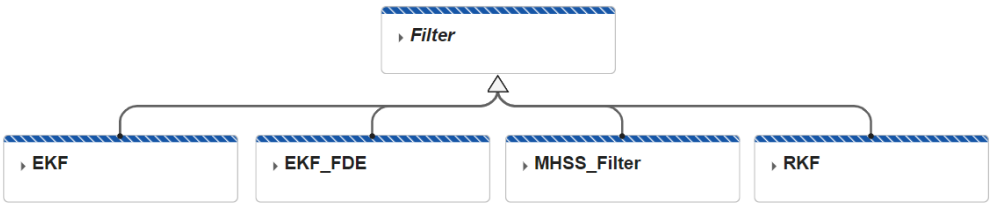


Figure 4.2: Implemented filters in MATLAB class diagram viewer

4.2.1 Ideal Filter and EKF

A classical EKF algorithm with the option to calculate PL and ambiguity resolution is implemented for PPP computation. The pseudo-code of the processing loop in EKF is shown in Algorithm 1. In each epoch, the filter first performs state prediction based on the state dynamic model followed by the measurement update based on the observation model. Two sets of measurements are considered separately in the measurement update in the 'ideal filter' and 'EKF' filter.

- Ideal filter: This filter only uses noisy measurement for measurement update. It is considered as the reference solution for the other filters because it does not consider any injected faults.
- EKF filter: This filter uses noisy and faulty (if injected) measurement for measurement update. It is used to evaluate how an EKF would perform when all the available measurements are used even if they are faulty.

After the measurement update, the test statistic and test threshold are calculated. A fault detection is performed by comparing the test statistic and the threshold. If the test is passed, the PL can be computed if enabled. Otherwise, no PL should be computed. Carrier-phase ambiguity resolution is performed at this point if PPP fixed solution is desired. Finally, the PPP solution is saved in the *Filter* object for further error analysis.

Algorithm 1: EKF processing loop

```
for each epoch do
    State prediction
    Measurement update
    Calculate test statistic and threshold
    if test statistic > threshold then
        Test fails
    else
        Test passes
    if test passes and PL computation enabled then
        Compute PL
    if ambiguity resolution enabled then
        Perform ambiguity resolution
    Store PPP solution
```

4.2.2 EKF-FDE

A single fault detection and exclusion process is added to the EKF algorithm described earlier. This would allow the filter to provide reasonable estimation when a single fault exists. The pseudo-code for the processing loop of EKF-FDE is shown in Algorithm 2. The difference between Algorithm 1 and Algorithm 2 lies after the fault detection test for all-in-view solution.

If the all-in-view solution passes the test, Algorithm 2 continues the same procedure as in Algorithm 1. Otherwise, single-fault exclusion is performed by removing the satellite with the largest innovation from all-in-view solution. The new set of satellites is name as single-fault set and its test statistic and test threshold are re-calculated. Another fault detection is performed by using these newly calculated test statistic and test threshold. If the test is passed, it suggests that the fault exclusion is successful and the single-fault set performs measurement update. Otherwise, the fault exclusion is unsuccessful. In this case, the filter only outputs prediction from the all-in-view set and declares solution unavailable. The remaining parts for PL computation, ambiguity resolution, and PPP solution saving are the same as in Algorithm 1.

Algorithm 2: EKF with single-fault detection and exclusion processing loop

```
for each epoch do
    State prediction
    Measurement update
    Calculate test statistic and threshold for all-in-view
    if test statistic > threshold then
        Single-fault exclusion
        Calculate test statistic and threshold for single-fault
        if test statistic > threshold then
            Test fails and solution unavailable
        else
            Test passes and measurement update for single-fault
    else
        Test passes and solution available
    if test passes and PL computation enabled then
        Compute PL
    if ambiguity resolution enabled then
        Perform ambiguity resolution
    Store PPP solution
```

4.2.3 MHSS

To deal with more than a single fault in each epoch, the MHSS filter has a much more complicated fault detection and exclusion process the the EKF with single FDE. The pseudo-code for MHSS is shown in Algorithm 3. After the number of subsets is determined, the algorithm goes through each subset and perform state prediction, measurement update, and calculate test statistic and test threshold independently. Note that each subset only use the measurements from the considered satellites during measurement update, and the calculation for test statistic and threshold.

Once the test statistics and thresholds for all the subsets are obtained, fault detection is firstly

performed on the all-in-view subset. If the all-in-view subset has a test statistic lower than its threshold, it is considered as the fault-tolerant solution. Otherwise, fault detection on the remaining subsets are required. If all of the remaining subsets have their test statistic larger than their threshold, then all the subsets fail the test and the solution at this epoch is declared unavailable. This suggests that there is a fault mode not covered in all the subsets and no subset can provide reasonable state estimation. Therefore, all subsets only output predicted states.

If there is one and only one subset passes the test (test statistic lower than threshold), this suggests that this subset successfully exclude the faulty satellite(s) and is considered as the fault tolerant solution. The remaining subsets (including the all-in-view) are considered to contain the faulty satellite(s) and only output the predicted states. If there are multiple subsets pass the test, then the subset that passes the test and has the minimum test statistic is considered as the fault tolerant solution. In this case, the remaining subsets (including the all-in-view) also only output the predicted states. This situation can be caused by a false alarm in the all-in-view subset. The decision of taking the subset with the minimum test statistic allows the MHSS to still provide reasonable state estimation without sacrificing the availability. The rest of the procedure for computing PL, ambiguity resolution, and saving PPP solution for fault tolerant solution are identical to the previous algorithms.

Algorithm 3: MHSS processing loop

```
for each epoch do
    for each subset do
        State prediction
        Measurement update
        Calculate test statistic and threshold
    // Fault detection and exclusion
    if all-in-view test statistic > threshold then
        if all remaining subsets have test statistic > threshold then
            all subsets fail the test, solution unavailable
            only prediction for all subsets
        else
            // Find subset(s) that pass the test
            if only one subset passes the test then
                fault-tolerant solution ← passed subset
                only prediction for remaining subsets
            else
                // multiple subsets pass the test
                fault-tolerant solution ← subset with minimum test statistic
                only prediction for remaining subsets
    else
        // All-in-view subset passes the test
        fault-tolerant solution ← all-in-view
    if test passes and PL computation enabled then
        Compute PL for fault-tolerant solution
    if ambiguity resolution enabled then
        Perform ambiguity resolution for fault-tolerant solution
    Store PPP solution
```

4.2.4 RKF

A robust re-weight process on the measurement covariance matrix is implemented with the EKF. This filter is called Robust Kalman Filter (RKF). The re-weighting process allows the filter to mitigate the effect of potential faults in the measurement without fault exclusion and the prior knowledge of fault modes. The pseudo-code of the RKF processing loop is shown in Algorithm 4. It starts with the same state prediction step as in EKF and followed by the measurement update combined with robust re-weighting.

In the measurement update step in RKF, the first step is to build the design matrix $\mathbf{H}$, observation covariance matrix $\mathbf{R}$, and the weight matrix $\mathbf{W}_y$. The $\mathbf{W}_y$ is initialized as an identity matrix. The $\mathbf{R}^{1/2}$ is obtained by Cholesky decomposition and used in the robust iteration loop which contains the re-weighting process described in section 2.3. Note that the information vector and matrix are used to avoid numerical singularity issue. The goal of the robust iteration loop is to find the $\mathbf{W}_y$ such that the Euclidean norm of the difference between the updated state at current epoch and previous epoch is lower than a selected threshold (for example: 10^{-3}). The robust iteration continues until this condition is met or it reaches the selected maximum number of iteration.

After the robust iteration loop, the updated state covariance matrix is calculated from the inverse of the information matrix. The test statistic and test threshold are calculated considering the re-weighted $\bar{\mathbf{R}}$ instead of the original $\mathbf{R}$. Finally, the PL computation and ambiguity resolution are performed when enable and the PPP solution is saved. Note that in the PL computation using Equation (3.19), $\bar{\mathbf{R}}$ should be used to calculate the Kalman Gain and innovation covariance matrix as shown in Equation (2.23) and (2.37).

Algorithm 4: RKF processing loop

```
for each epoch do
    State prediction
    // Measurement update with robust re-weighting
    Build H, R, and W_y
    Perform Cholesky decomposition on R to find R^{1/2}
    for each robust iteration do
        Compute innovation ν and re-weighted R̄ with (2.36) and (2.63)
        Compute information vector z and information matrix Z with (2.66) and (2.67)
        Compute state error Z^{-1}z
        Update the predicted state using the state error
        Compute W_y with the loss function (2.64)
        if ||updated state − state in previous iteration|| < threshold then
            break
    Calculate the state covariance matrix from information matrix
    Calculate test statistic and threshold
    if PL computation is enabled then
        Compute PL
    if ambiguity resolution is enabled then
        Perform ambiguity resolution
    Store PPP solution
```

4.3 Simulation and filter configuration

This section describes the parameter settings and stochastic models used in the simulation to generate the receiver ground truth states, satellite position, and measurements. The ones used within the filters described in section 4.2 are also presented.

- Scenario configuration: The error models, initial values, and initial covariance that are used to generate the receiver's ground truth states are shown in Table 4.1 and 4.2. The constant velocity model considers only first order integration of the process noise to the position state. These values are configured in the scenario configuration file described in section 4.1.

States	**Var.**	**Error model**
Receiver position	$\mathbf{p}$	Constant velocity $\sigma_{\text{pos}} = 0$ [m]
Receiver velocity (ENU)	$\dot{\mathbf{p}}$	Constant velocity $\sigma_{\text{vel,ENU}} = [10^{-1}, 10^{-1}, 10^{-3}]$ [m/s^2]
Receiver clock bias	dt	Random walk $\sigma_{dt} = 1$ [m]
Satellite clock bias	dt^s	Random walk $\sigma_{dt^s} = 1$ [m]
Ionosphere delay	I	Random walk $\sigma_I = 0.1$ [m]
Troposphere delay	T	Random walk $\sigma_T = 0.01$ [m]
Ambiguities	a	Constant $\sigma_{\text{amb}} = 0$ [cycle]

Table 4.1: Error model for receiver ground truth generation

States	**Var.**	**Initial value**	**Initial covariance**
Receiver position	$\mathbf{p}$	[53.3295056, 13.0717511, 10] [deg, deg, m]	$(10)^2$ [m^2]
Receiver velocity (ENU)	$\dot{\mathbf{p}}$	[0, 0, 0] [m/s]	$(2)^2 \cdot \text{diag}(1, 1, 10^{-3})$ [m^2]
Receiver clock bias	dt	$\sqrt{\text{initial covariance}} \cdot \text{randn}$	$(10^5)^2$ [m^2]
Satellite clock bias	dt^s	$\sqrt{\text{initial covariance}} \cdot \text{randn}$	$(10^4)^2$ [m^2]
Ionosphere delay	I	$\sqrt{\text{initial covariance}} \cdot \text{randn}$	$(5)^2$ [m^2]
Troposphere delay	T	$\sqrt{\text{initial covariance}} \cdot \text{randn}$	$(2)^2$ [m^2]
Ambiguities	a	$\text{randi}(-293, 293)$	$(1000)^2$ [cycle^2]

Table 4.2: Initial value and covariance for receiver ground truth generation

- Measurement noise configuration: The measurement noise for pseudorange code and carrier-phase are assumed to be centered Gaussian distributed and satellite elevation dependent, as shown in Table 4.3. These values are defined in the noise and fault configuration file and added to the 'clean' GNSS measurement. It is important to note that ephemeris noise is *not* added to the clean measurement.
- Filter configuration: The simulation allows us to have perfect prior knowledge of the state's dynamic and the measurement noise model. The filters are initialized with the values and covariance of the states as shown in Table 4.2. The process noise covariance matrix is built based on the state error models as shown in Table 4.1.

Extra care should be taken when constructing the observation covariance matrix. As mentioned earlier, the simulated measurement does not consider the ephemeris error. However, if the filters take it into account while predicting the measurement to simulate the uncertainty in the satellite orbit and clock information, this model mismatch can cause filter inconsistency, particularly when precise carrier-phase measurements are involved. To ensure the consistency of the filters, the observation covariance matrix can built based on the measurement noise in Table 4.3 plus the ephemeris error. With the assumption that the ephemeris errors are independent between satellites, the observation covariance matrix can be constructed based on Equation (2.18) by adding the ephemeris error on the diagonal elements as follow:

$$\mathbf{R} = \begin{bmatrix} \sigma_\rho^2 \mathbf{I}_{2m} & \mathbf{0} \\ \mathbf{0} & \sigma_\Phi^2 \mathbf{I}_{2m} \end{bmatrix} + \sigma_{svPos}^2 \mathbf{I}_{4m} + \sigma_{svClk}^2 \mathbf{I}_{4m} \tag{4.1}$$

For the simplicity of the simulation, the ephemeris errors are not considered in this thesis.

Error Source	**Var.**	**Error model**
Pseudorange	ε	$\sigma_\rho = a + \frac{b}{\sin(elev)}\ [m], a = 0, b = 0.3$
Carrier-phase	ϵ	$\sigma_\Phi = 10^{-2} \cdot \sigma_\rho\ [m]$

Table 4.3: Noise profile for noisy measurement generation

4.4 Considered Scenarios

Three scenarios are configured to investigate the performance of the PPP integrity monitoring algorithm under nominal, single fault, and double faults conditions. The common configuration of all these scenarios are summarized in Table 4.4. Note that the satellite geometry is assumed to be fixed. This assumption allows us to deal with the same set of satellites in all-in-view solution throughout the simulation. The configuration for each scenarios are described as follow:

Configuration	**Description**
Satellite geometry	fixed
Number of satellites	19
Number of epochs	3600
Sampling time	1 $[sec]$
Evaluated constellation	GPS (G), Galileo (E)
Considered frequency	L1 (G), L2 (G), E1 (E), E5a (E)
P_{fa}	0.001
P_{md}	10^{-5}
Horizontal alert limit	3 $[m]$

Table 4.4: Common configuration for all scenarios

- Nominal scenario: In this scenario, noisy measurements are used by all filters. No faults are injected to measurements.
- Single fault scenario: This scenario allows at most one faulty satellite per epoch. Three fault spans are considered within 3600 epochs. In each fault span, the faulty satellite is selected at random, and its measurements are affected by injected faults. The fault configuration for single fault scenario is summarized in Table 4.5. The first and second values in the fault parameters correspond to the fault's mean and standard deviation, respectively.

Configuration	**Description**
Fault type	step error
Starting epoch	$[500, 1500, 2500]$
Ending epoch	$[1000, 2000, 3000]$
Satellite ID	$[5, 30, 7]$
Satellite constellation	[G, G, G]
Affected frequency	[all, all, all]
Fault parameters	$[(100, 0), (100, 0), (100, 0)]$ $[m]$
Affected observation	[code, code, code]

Table 4.5: Fault configuration for single fault scenario

- Double faults scenario: This scenario permits up to two faulty satellites per epoch. The faulty satellites are randomly selected, and their measurements are affected by injected faults. The fault configuration for double faults scenario is summarized in Table 4.6.

Configuration	**Description**
Fault type	step error
Starting epoch	$[500, 500, 1500, 1500, 2500, 2500]$
Ending epoch	$[1000, 1000, 2000, 2000, 3000, 3000]$
Satellite ID	$[36, 20, 6, 11, 53, 5]$
Satellite constellation	[E, G, G, G, E, G]
Affected frequency	[all, all, all, all, all, all]
Fault parameters	$[(100, 0), (100, 0), (100, 0), (100, 0), (100, 0), (100, 0)]$ $[m]$
Affected observation	[code, code, code, code, code, code]

Table 4.6: Fault configuration for double faults scenario

Chapter 5

Results and Discussion

This chapter presents the simulation results of the implemented PPP integrity-monitoring algorithms. All results were generated using the configurations described in section 4.3. The chapter begins with the presentation of the receiver's ground-truth states, followed by an evaluation of filter consistency. The results for the scenarios outlined in section 4.4 are then presented in order of increasing fault complexity. Finally, time-profiling results for each filter across all scenarios are provided to give an overview of the required computational cost of the simulation.

5.1 Generated receiver ground truth states

The ground truth states of position trajectory and velocity in East-North-Up direction are shown in Figure 5.1 and 5.2. We can see that the receiver ground truth velocity follows a random walk pattern in East and North direction which matches the constant velocity dynamic model. The velocity in the Up direction is close to zero which corresponds to the configuration described in section 4.3.

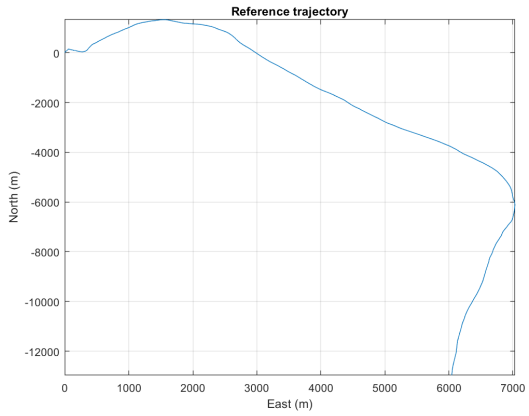


Figure 5.1: Reference ground truth trajectory

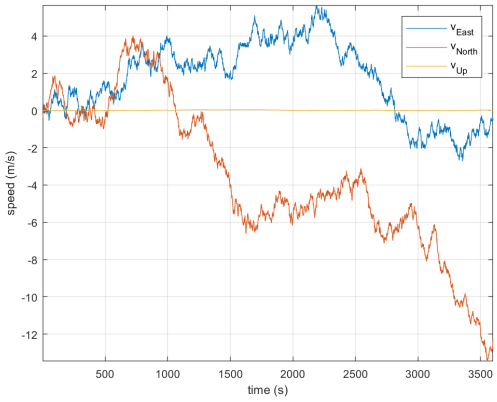


Figure 5.2: Ground truth of receiver velocity

The ground truth states of tropospheric and ionospheric delay are also modeled as random walk process. All measurements share the same tropospheric delay which is mapped to the measurement domain by a mapping function. On the other hand, each satellite has its own ionospheric delay and its effect on measurement is frequency dependent. The ground truth of

ambiguities are shown in Figure 5.3. Each ambiguity of the satellites stays as constant integer throughout the simulation.

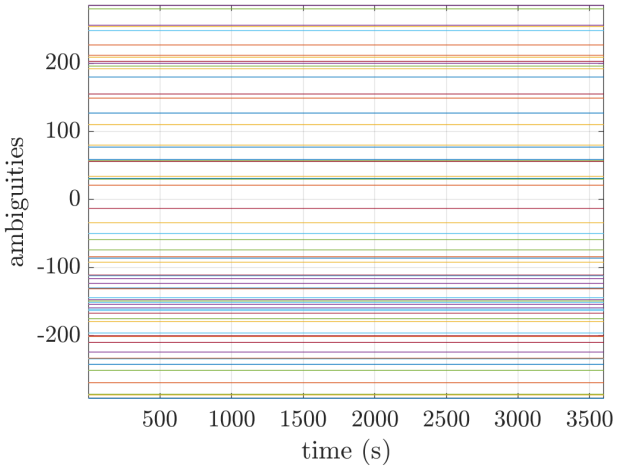
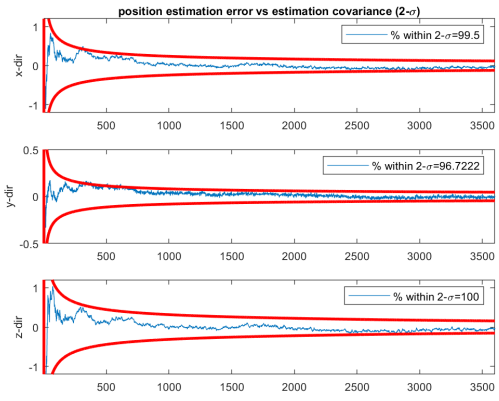


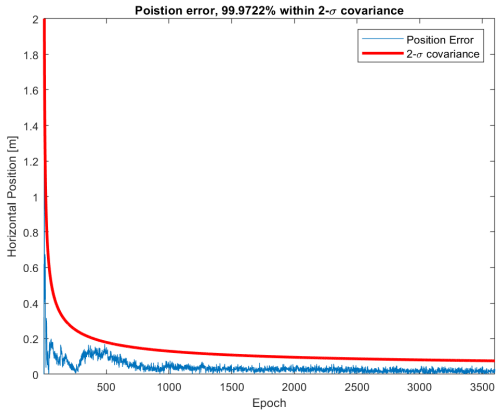
Figure 5.3: Ground truth of ambiguities

5.2 Filter consistency evaluation

The techniques described in section 2.2.1.1 are used to evaluate the filter's consistency under nominal scenario. These evaluations ensure that there is no significant model mismatch within the EKF and that both the measurement model and state dynamic model are properly selected. In addition, the chi-square test for fault detection relies on the assumption that the filter remains consistent when the measurements are affected only by noise. Therefore, ensuring the filter's consistency with the given configuration is essential. Figure 5.4a and 5.4b show the position estimation error versus the $2-\sigma$ estimation covariance in ECEF and in the horizontal direction for the ideal EKF under nominal scenario. We can see that the over 95% of the estimation errors in x, y, z, and horizontal directions are covered within the $2-\sigma$ covariance. This suggests that the filter can provide the state estimation covariance that is consistent to its state estimation.



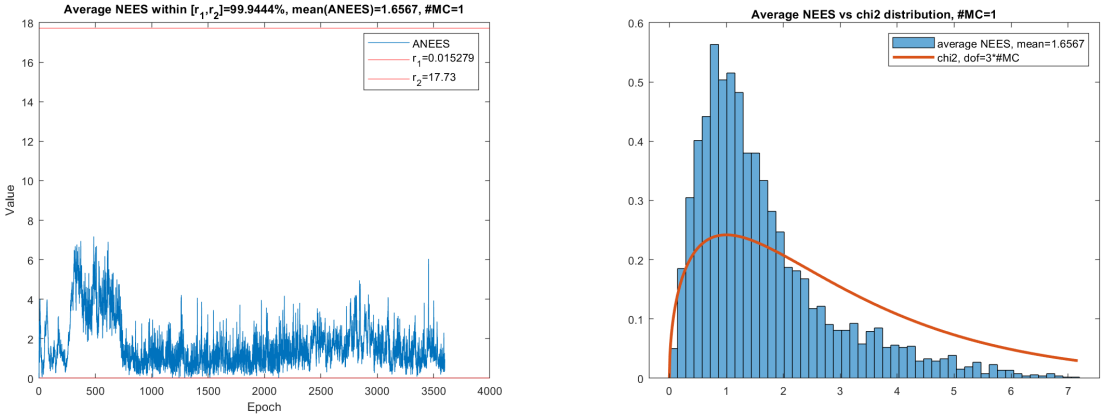
(a) x,y,z direction in ECEF



(b) Horizontal direction

Figure 5.4: Position estimation error vs $2-\sigma$ covariance

The NEES time series and its histogram comparing with a chi-square distribution for one Monte Carlo simulation run are shown in Figure 5.5a and 5.5b. Note that the NEES values are calculated considering only the position states, therefore the considered chi-square distribution should have degrees of freedom equals to 3. We can see that more than 99.9% of the NEES values falls within the confidence interval. In this case, the histogram of the NEES does not follow perfectly with the chi-square distribution with more values close to zero. This suggests that the estimation error is very small or the covariance matrix is too large. This is acceptable because the filter is pessimistic rather than optimistic.

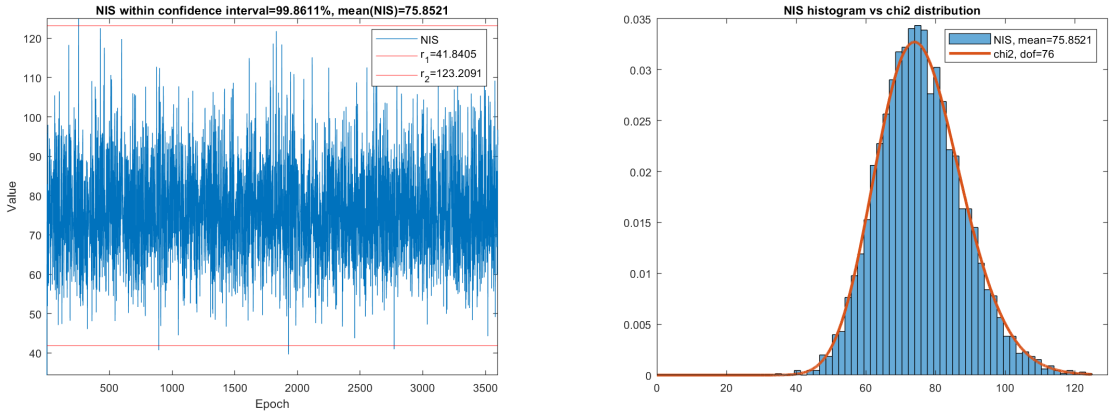


(a) NEES time series

(b) NEES histogram vs χ^2 distribution

Figure 5.5: Filter consistency investigation with NEES

Similar figures for NIS are presented in Figure 5.6a and 5.6b. Note that the considered chi-square distribution for NIS should have degrees of freedom equals to the length of the innovation vector, which is 76 in this case. We can see that there are more than 99.8% of the NIS values falls within the confidence interval. Furthermore, the distribution of NIS follows the chi-square distribution with the correct degrees of freedom. This suggests that the filter fulfills the second criteria for the innovation mentioned in 2.2.1.1 and can be accepted as consistent.



(a) NIS time series

(b) NIS histogram vs χ^2 distribution

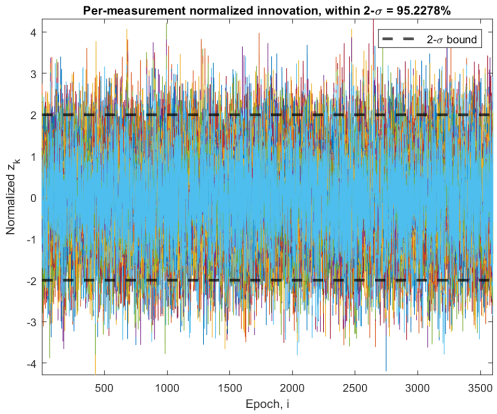
Figure 5.6: Filter consistency investigation with NIS

Figure 5.7a shows the per-measurement normalized innovation which is calculated by normalizing

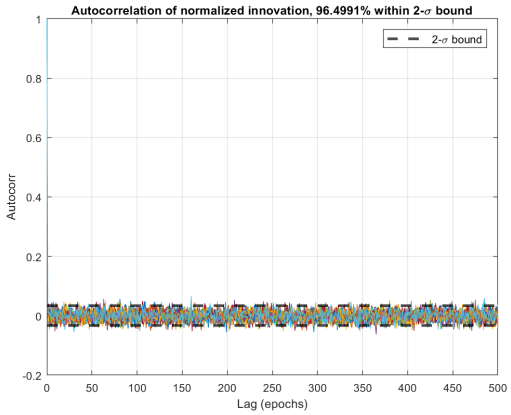
each innovation element with the corresponding diagonal element in the innovation covariance. We can see that around 95% of the per-measurement normalized innovation fall with the $2-\sigma$ bound. This conforms that the filter's innovation can be accepted as zero mean and have the magnitude comparable with the filter-calculated innovation covariance.

The time-average autocorrelation of the innovation is shown in Figure 5.7b where $2-\sigma$ bound is calculated using Equation (2.42) with $K = 3600$ epochs. Around 96% of the values fall within the $2-\sigma$ bound. This suggests that the elements in the innovation vector are uncorrelated in time, therefore the innovation can be accepted as white across time. To investigate the correlation of the innovation across measurements, the correlation coefficient between innovation elements can be calculated by considering all the innovation vectors in 3600 epochs. Figure 5.7c shows correlation coefficient matrix of the innovation vector. We can see that the diagonal elements are close to one because each innovation element is always directly correlated to itself. The off-diagonal element are close to zero which suggests that the innovation elements are indeed uncorrelated across measurements.

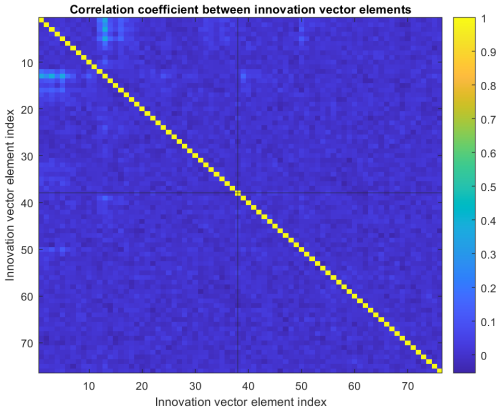
With the analysis of NEES, NIS, and innovation whitness test, we can ensure that the filter is consistent using the configuration described in section 4.3.



(a) Normalized innovation time series



(b) Autocorrelation of innovation



(c) Correlation coefficient between innovation elements

Figure 5.7: Filter consistency investigation with innovation

5.3 Scenario results

The results of 20 Monte Carlo simulations for nominal, single fault, and double fault scenarios are shown in the following sections. We focus on the horizontal Root Mean Square Error (RMSE) for each epoch which is calculated as follows:

$$\text{Horizontal RMSE} = \sqrt{\frac{1}{20}\sum_{j=1}^{20}\left(e_{E,j}^2 + e_{N,j}^2\right)} \tag{5.1}$$

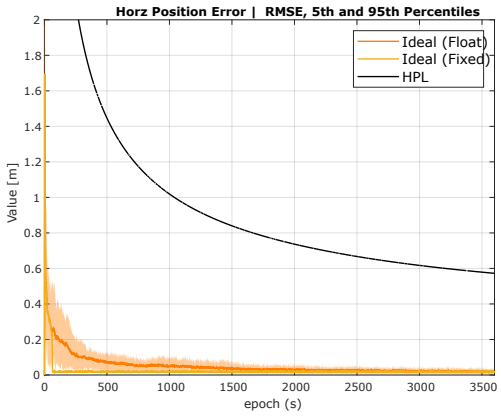
where $e_{E,j}$ and $e_{N,j}$ refer to the position error in the East and North direction at one epoch for the j^{th} Monte Carlo simulation. The 5% and 95% quantile of the horizontal RMSE are calculated at each epoch from the 20 Monte Carlo simulations and shown as the shaded area around the RMSE. The HPL at each epoch is calculated using Equation (3.20) and put in the same figure with RMSE for comparison. The CDF of the horizontal RMSE for float solution and fix solution are presented to have a clearer view of the position estimation accuracy. The confusion matrix for ideal EKF, EKF, EKF-FDE, and MHSS filters are used to evaluate the fault detection performance using chi-square test. The confusion matrix of RKF is not shown because the chi-square test is not valid for robust estimator, as discussed in section 3.2.3. Finally, in the single fault and double fault scenario, the detected faulty satellites in EKF-FDE and MHSS through out time are shown.

5.3.1 Nominal scenario

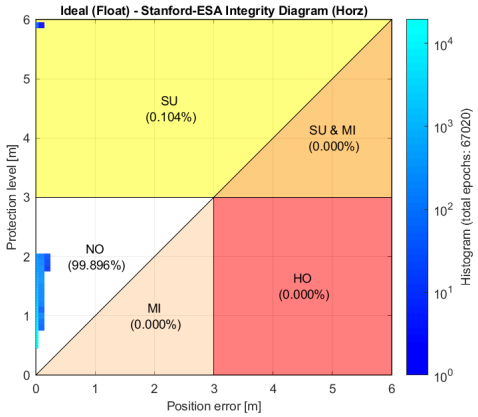
This section shows the results of 20 Monte Carlo simulations for all the filters in nominal scenario, i.e, measurements are only affected by noise. Figure 5.8 to 5.12 show the horizontal RMSE and Stanford diagram for each filter. We can see that the RMSE for all filters are below the calculated HPL which suggests that the HPL can properly protect the estimated position. In the nominal case, the ideal EKF is identical to EKF because they both use the same measurements and algorithm.

In the Stanford diagram for ideal EKF and EKF, we can that there are around 0.1% of the epochs are categorized as System Unavailable (SU). These epochs corresponds to the probability of false alarm in the simulation configuration ($P_{fa} = 0.001$) as shown in Table 4.4. They have small position errors, but still fail the chi-square test due to false alarm, thus no PLs are not calculated which are saved as nan values. To show these false alarms in the Stanford diagram, these nan values in PL are replaced with a very large value, for example 30 meters, to present them in the SU area. In the case of EKF-FDE and MHSS, lower percentage of epochs fall within SU because these two filters will try to exclude faults when the all-in-view subset fails the chi-square test. If the chi-square test still fails after fault exclusion, the solution at the epoch is marked as solution unavailable, shown as the red cross in Figure 5.10a and 5.11a. Because the solution of these epochs are declared unavailable, the the RMSE at these epochs are also not presented. Nevertheless, more than 99.9% of the all the epochs in 20 Monte Carlo simulation fall within the Normal Operation (NO) area. In the case of RKF, the PL is calculated at every epoch and compared with the estimated position error. Its Stanford diagram shows that all the epochs in the 20 Monte Carlo simulation fall within the NO area.

The horizontal RMSE and its CDF for the float solution of all filters are shown in Figure 5.13. We can see that all filters have similar performance to the ideal EKF under nominal scenario, with around 90% of the RMSE are under 10^{-1} m. The result of RKF is slightly worse than the ideal filter. Based on Equation (2.68), we can calculate the relative efficiency of the RKF with respect to the EKF which is around 95%.

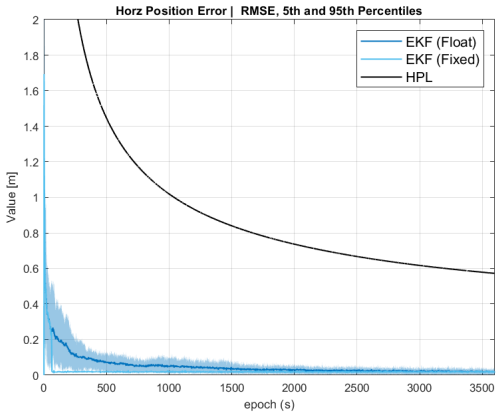


(a) Horizontal RMSE

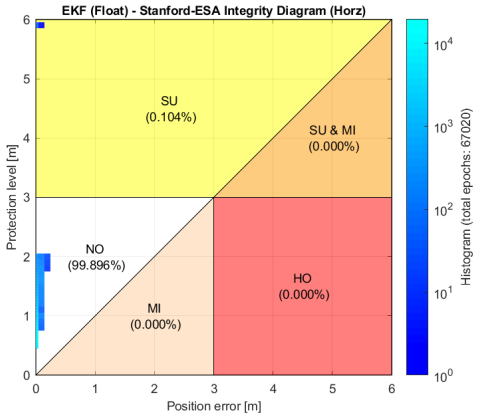


(b) Standford diagram

Figure 5.8: Ideal EKF in nominal scenario

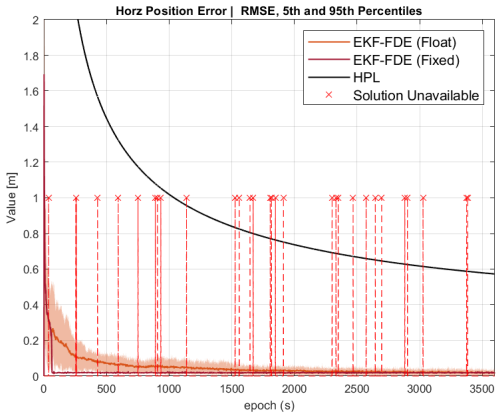


(a) Horizontal RMSE

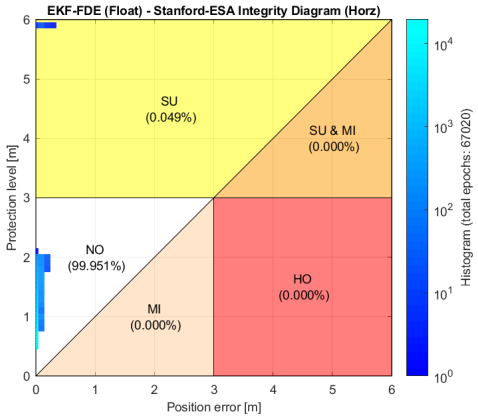


(b) Standford diagram

Figure 5.9: EKF in nominal scenario

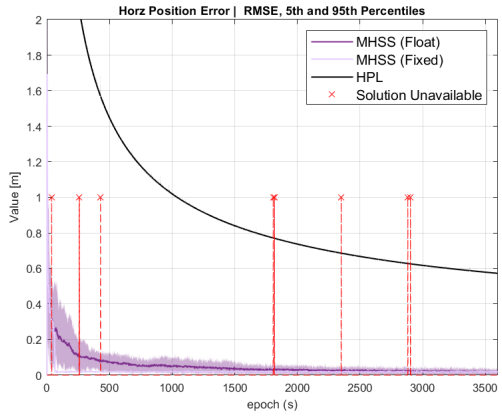


(a) Horizontal RMSE

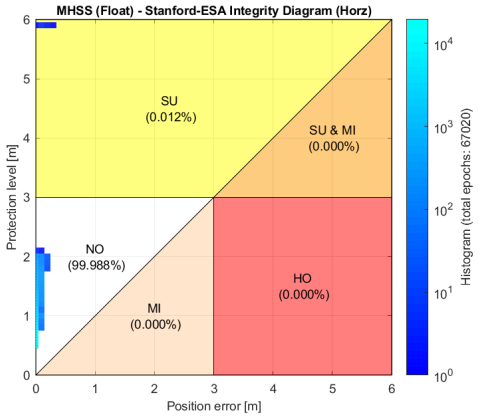


(b) Standford diagram

Figure 5.10: EKF-FDE in nominal scenario

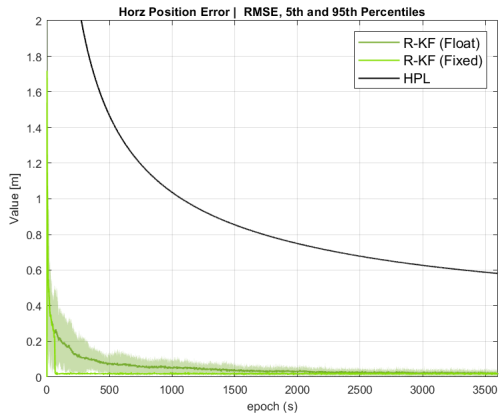


(a) Horizontal RMSE

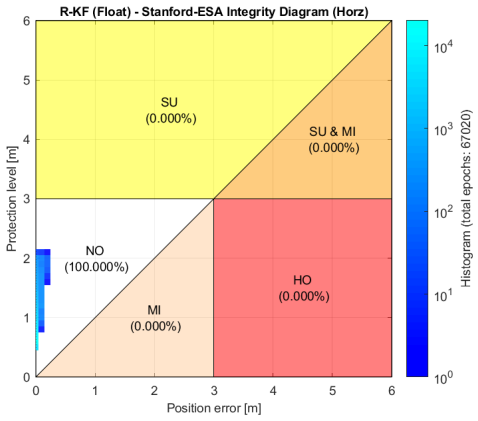


(b) Standford diagram

Figure 5.11: MHSS in nominal scenario



(a) Horizontal RMSE

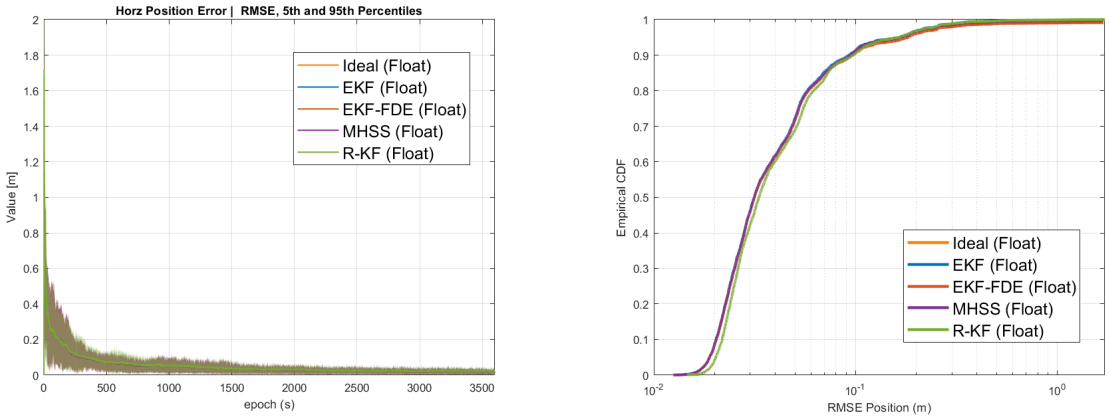


(b) Standford diagram

Figure 5.12: RKF in nominal scenario

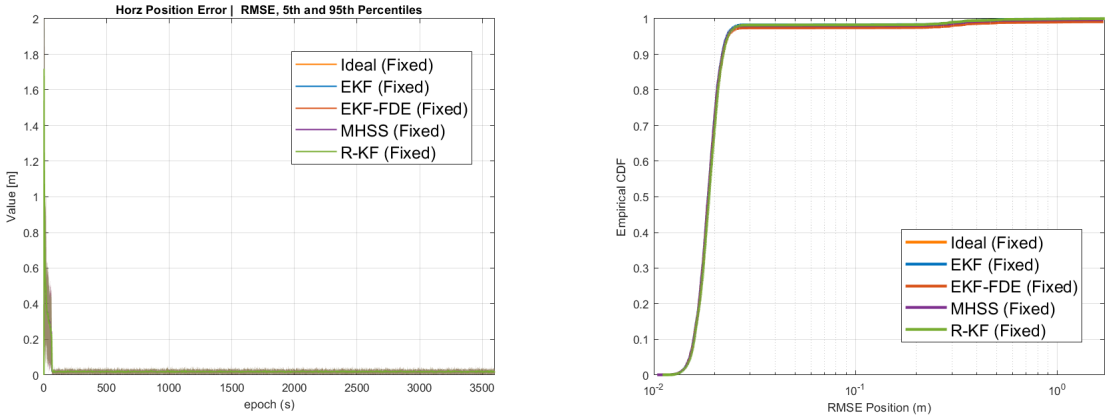
Similar results for the fix solution are shown in Figure 5.14. Figure 5.14a shows that with the Partial Ambiguity Resolution technique provided in LAMBDA 4.0 toolbox, all filters can achieve centimeter positioning accuracy with less convergence time than the float solution. The CDF in Figure 5.14b shows that more than 95% of the RMSE are below 0.03 m.

The results of ambiguity resolution are shown in Fig 5.15. The peaks for EKF-FDE and MHSS that drops to zero corresponds to those epochs which solutions are unavailable. The percentage of fixed ambiguities for ideal EKF, EKF, and RKF increase with time and reach 100% after around 2500 epoch in the simulation. This suggests that with a longer observation time, more ambiguities can be fixed to integer value. However, fixing the ambiguity to integer value is not equivalent to fixing to the correct integer value, i.e. the ground truth value of ambiguities. Figure 5.15b shows that only after around 2500 epochs the percentage of correct fixed ambiguities for ideal EKF, EKF, and RKF can reach 100%.



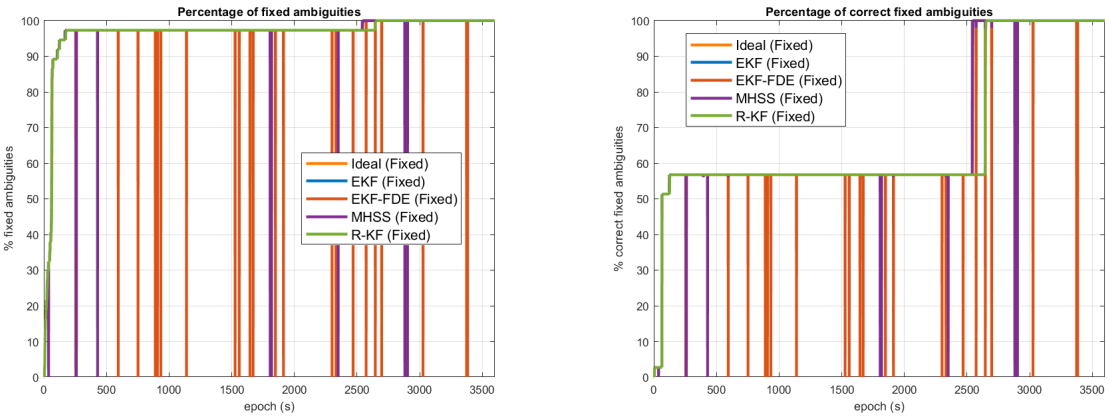
(a) Horizontal RMSE (b) CDF of horizontal RMSE

Figure 5.13: Comparison of float solution for all filters in nominal scenario



(a) Horizontal RMSE (b) CDF of horizontal RMSE

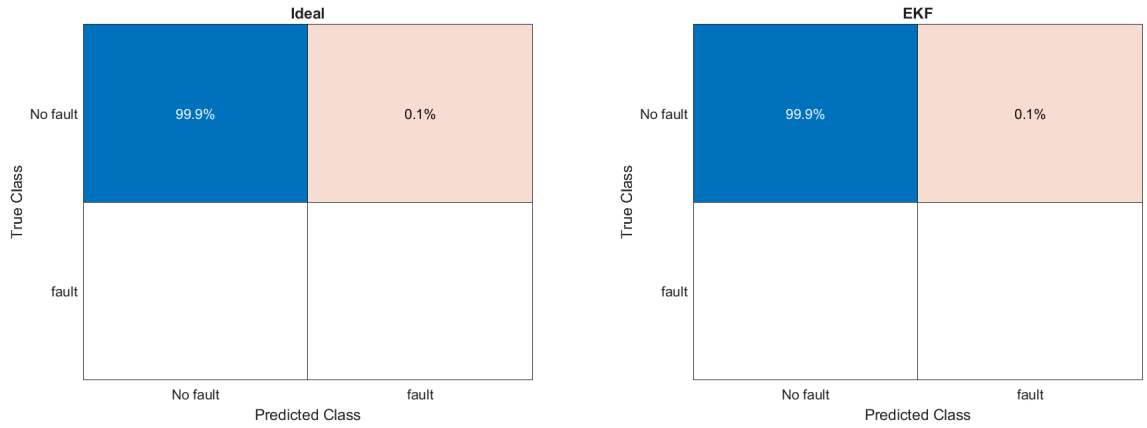
Figure 5.14: Comparison of fix solution for all filters in nominal scenario



(a) Percentage of fixed ambiguities (b) Percentage of correct fixed ambiguities

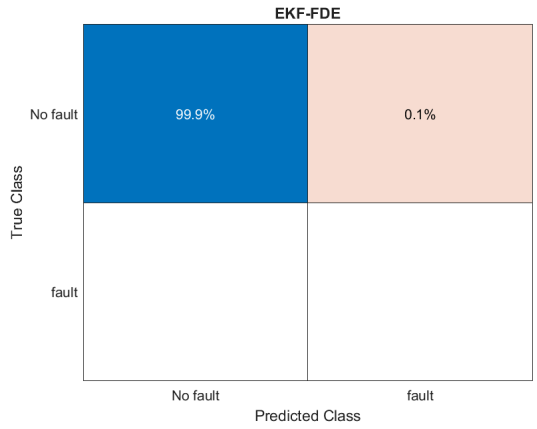
Figure 5.15: Comparison of ambiguity resolution for all filters in nominal scenario

The confusion matrices for ideal EKF, EKF, EKF-FDE, and MHSS considering the all-in-view subset are shown in Figure 5.16. We can see that the empirical false alarm rate is $\frac{0.1\%}{0.1\%+99.9\%} = 0.001$ which corresponds to the P_{fa} defined in the simulation configuration. This suggests that the fault detection for all-in-view subset with chi-square test is performing as expected.

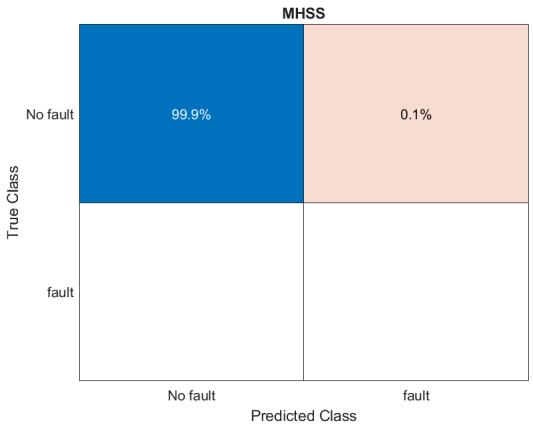


(a) Ideal EKF

(b) EKF



(c) EKF-FDE



(d) MHSS

Figure 5.16: Confusion matrix for fault detection in all-in-view solution in nominal scenario

5.3.2 Single fault scenario

This section presents the results of 20 Monte Carlo simulations for all filters in the single fault scenario. The fault configuration is shown in Table 4.5. The epochs that contain injected single fault are shown as gray shaded Fault Spans in Figure 5.17a to Figure 5.22a.

In the case of ideal EKF, the horizontal RMSE and Stanford diagram are similar to the nominal scenario. This is expected because the ideal EKF only considers noisy measurements.

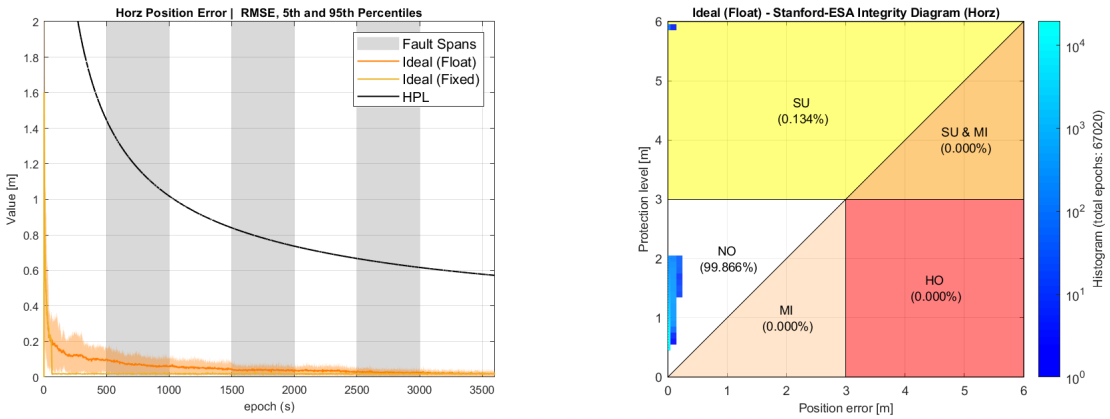
On the other hand, the horizontal RMSE of EKF varies violently during each fault span. The maximum RMSE error can exceed 8 meters at the end of the first fault span as shown in Figure 5.18a. Between each fault span when there is no faulty satellite exists, the RMSE does not drop back to the same level before the first fault span immediately, but slowly re-converge towards zero. This indicates that the effect of the previously injected fault can still affect the EKF position estimate when no fault exclusion or filter re-initialization is attempted. The Stanford

diagram for EKF in Figure 5.18b shows that over 90% of the epochs are in the SU area because the chi-square test starts to fail at the beginning of the first fault span and no fault exclusion is performed.

For the case of EKF-FDE, the injected single fault during each fault span can be properly detected as shown in Figure 5.21a. However, fault exclusion is not always successful and leads to the solution unavailable marks within the fault spans as shown in Figure 5.19b. The solution unavailable marks outside the fault spans suggest that these epochs have a false alarm in the all-in-view solution followed by an unsuccessful single fault exclusion. This can be considered as two false alarms in a row in each of these epochs, one in the all-in-view solution and the second one in the single-fault solution after fault exclusion. These epochs with solution unavailable marks refers to the data points fall within the SU area which accounts for around 0.095% of all the epochs in 20 Monte Carlo simulations.

The number of subsets in MHSS is calculated based on Equation 3.14. With a single fault assumption and 19 satellites, there are 20 subsets to be monitored in MHSS. The configured global P_{fa} is evenly distributed through all subsets, including the all-in-view subset, to perform fault detection. Figure 5.21b shows that all injected single faults can be correctly detected with only two false alarms. Unlike EKF-FDE, the MHSS can successfully exclude all detected faulty satellites and provides solutions for all epochs during fault spans, as shown in Figure 5.20a. This is because the P_{fa} for each subset is much lower than the global P_{fa}, therefore fewer false alarm events happened. However, the PLs for MHSS show visible jumps at the beginning and ending of each fault span. This is due to the change of the number of satellites of the fault tolerant subset inside and outside the fault span. The Stanford diagram of MHSS in Figure 5.20b shows that all epochs in the simulation fall within the NO area which confirms that the MHSS scheme can indeed correctly detect and exclude the single faulty satellite during fault spans.

In the case of RKF, Figure 5.22a shows that the horizontal RMSE slightly increases during the first fault span, but is always covered by the calculated HPL for the whole simulation. This suggests that the RKF is capable of mitigating the affect of the faulty satellite and provides reliable position estimate. The Stanford diagram in Figure 5.22b confirms the robustness of RKF against injected single-fault with all epochs in the simulation fall within the NO area.

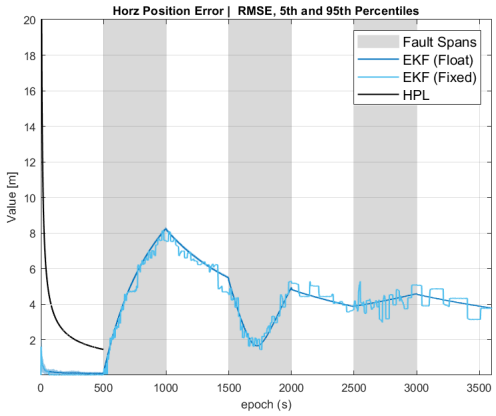


(a) Horizontal RMSE

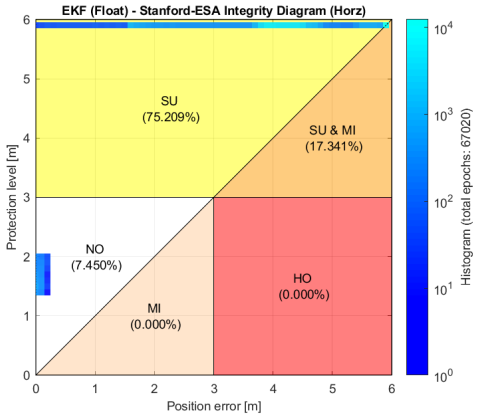
(b) Standford diagram

Figure 5.17: Ideal EKF in single fault scenario

Figure 5.23 shows the comparison of the float solutions between all filters. Note that the CDF for EKF-FDE does not reach to 100% in the y-axis because the solution unavailable epochs are

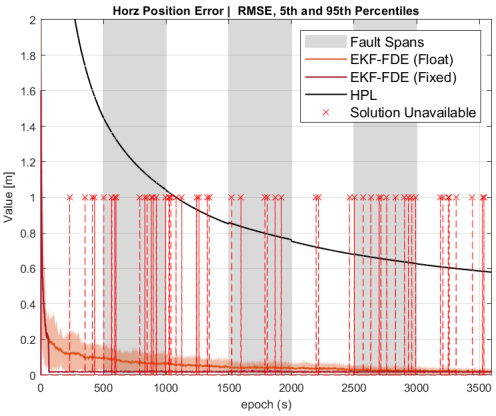


(a) Horizontal RMSE

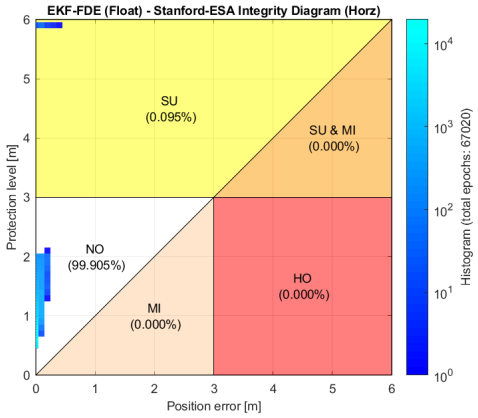


(b) Standford diagram

Figure 5.18: EKF in single fault scenario

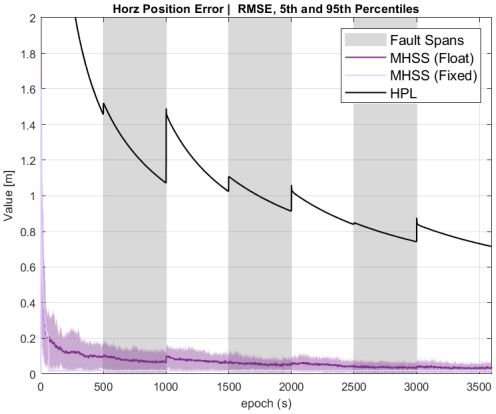


(a) Horizontal RMSE

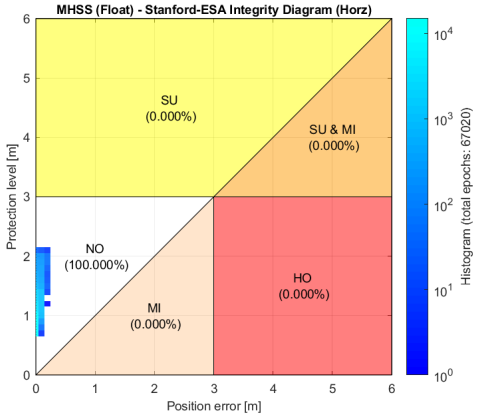


(b) Standford diagram

Figure 5.19: EKF-FDE in single fault scenario

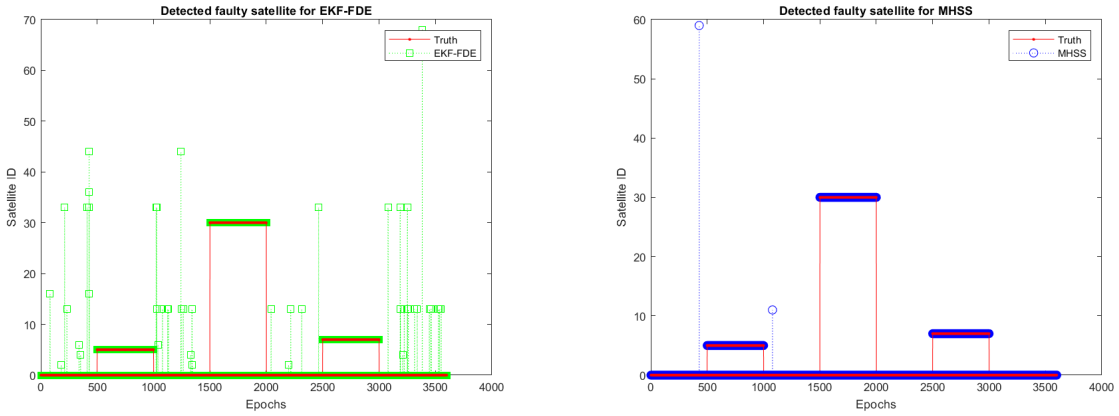


(a) Horizontal RMSE



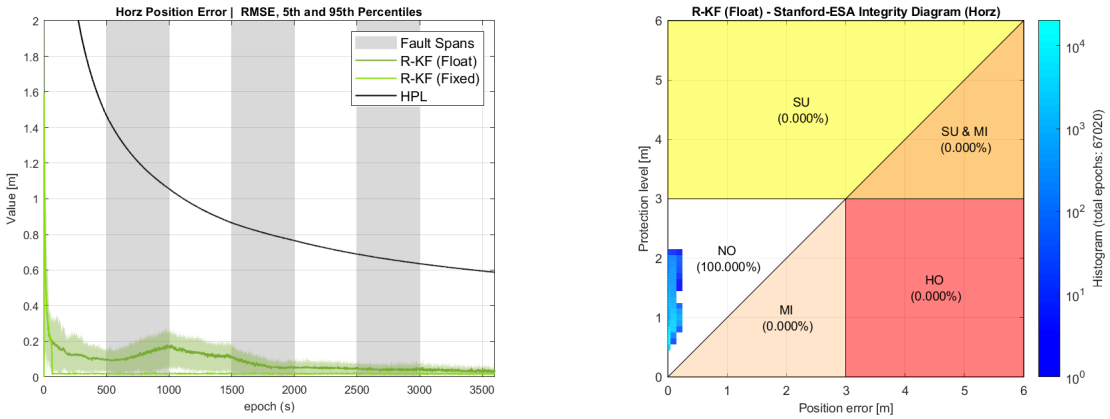
(b) Standford diagram

Figure 5.20: MHSS in single fault scenario



(a) EKF-FDE (b) MHSS

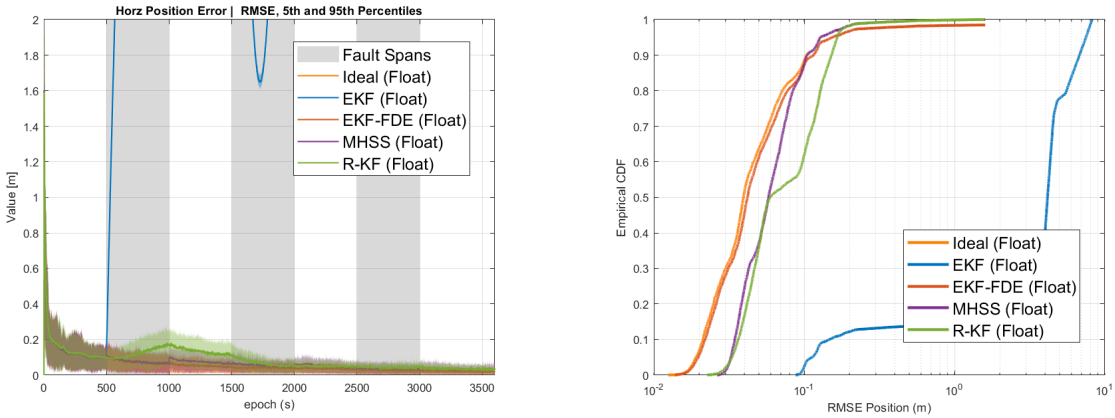
Figure 5.21: Detected faulty satellite in single fault scenario



(a) Horizontal RMSE (b) Standford diagram

Figure 5.22: RKF in single fault scenario

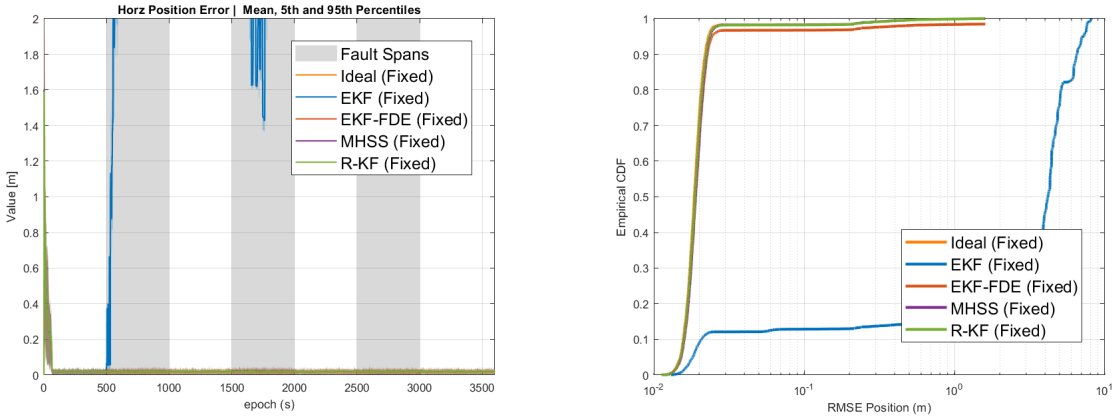
not included. The CDF shows that more than 95% of the epochs in EKF-FDE, MHSS, and RKF have RMSE lower than 0.2 m. This suggests that these three filters can still provide reasonable and reliable position estimate in the single fault scenario. The comparison of the fix solution is shown in Figure 5.24. We can see that with Partial Ambiguity Resolution (PAR) technique, the EKF-FDE, MHSS, and RKF filters can still reach centimeter accuracy in single-fault scenario. Figure 5.25 shows percentage of fixed ambiguity and correct fixed ambiguities in single fault scenario. The vertical lines for EKF-FDE corresponds to the solution unavailable epochs in Figure 5.19a. As shown in Figure 5.25b, the percentage of correct fixed ambiguities in EKF drops to lower than 10% after the first fault span. This could be related to the large RMSE in the float solution that makes correct ambiguity fixing difficult. Similar drops of the percentage of correct fixed ambiguities can also be seen in MHSS and RKF but with limited magnitude.



(a) Horizontal RMSE

(b) CDF of horizontal RMSE

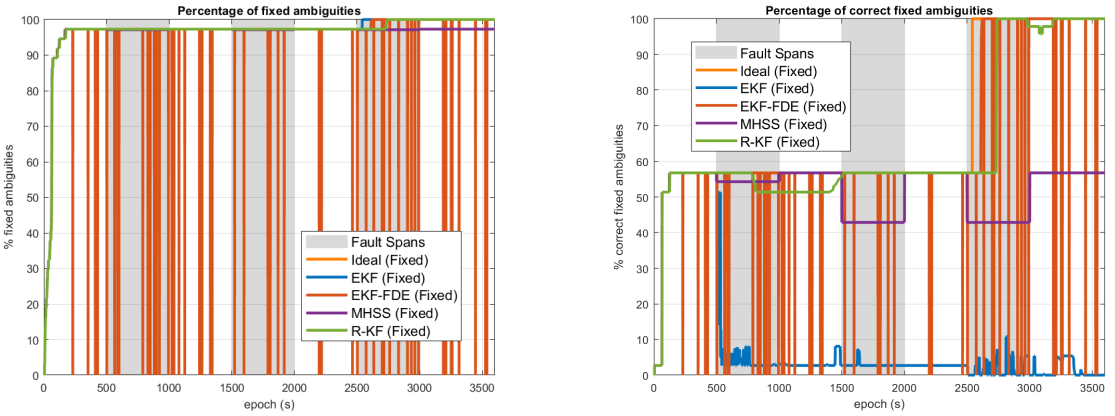
Figure 5.23: Comparison of float solution for all filters in single fault scenario



(a) Horizontal RMSE

(b) CDF of horizontal RMSE

Figure 5.24: Comparison of fix solution for all filters in single fault scenario

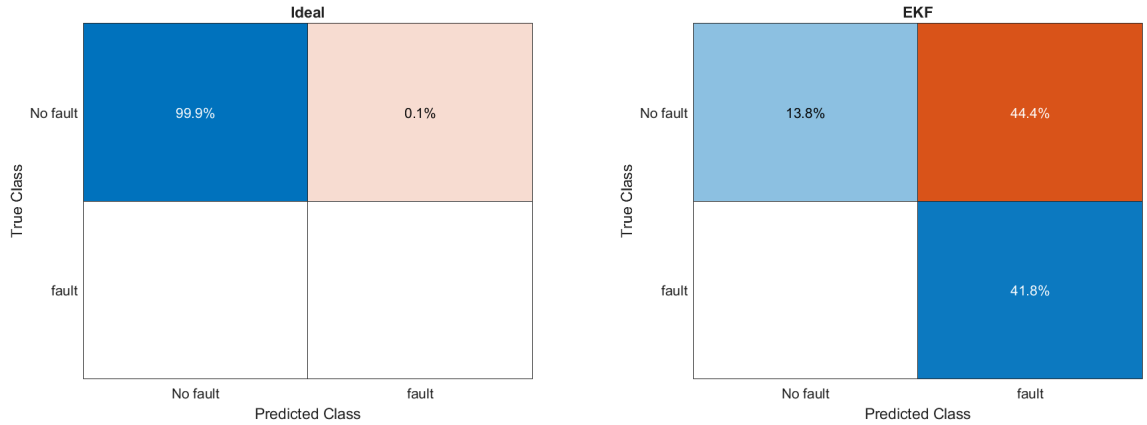


(a) Percentage of fixed ambiguities

(b) Percentage of correct fixed ambiguities

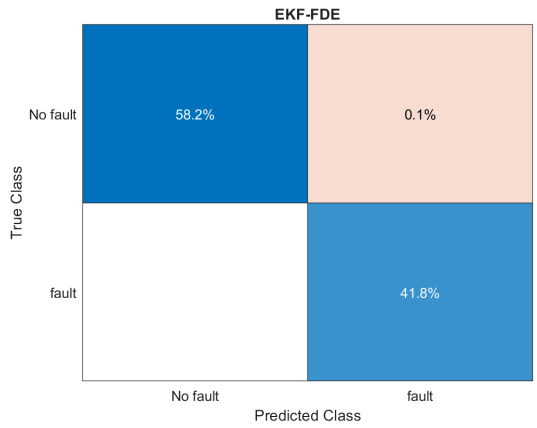
Figure 5.25: Comparison of ambiguity resolution for all filters in single fault scenario

The confusion matrices for fault detection in all-in-view solution for all filters except RKF in single fault scenario are shown in Figure 5.26. The confusion matrix for EKF shows 44.4% of false detection which corresponds to the epochs after each fault span and the filter is still re-converging from the affect of previous fault. The confusion matrices for EKF-FDE and MHSS have large values in the diagonal parts which suggests that these two filters can correctly identify if there is a fault in the all-in-view satellite.

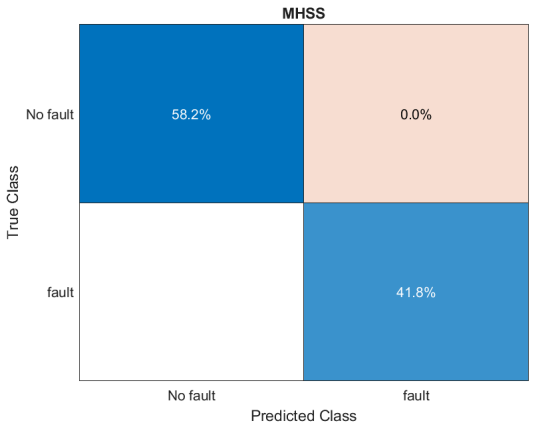


(a) Ideal EKF

(b) EKF



(c) EKF-FDE



(d) MHSS

Figure 5.26: Confusion matrix for fault detection in all-in-view solution in single fault scenario

5.3.3 Double fault scenario

This section presents the results of 20 Monte Carlo simulations for all filters in the double fault scenario. The fault configuration is shown in Table 4.6. Similar to the single fault scenario, the epochs that contain injected double faults refers to the Fault Spans in Figure 5.27a to 5.32a.

In Figure 5.27, the ideal EKF shows similar results to the nominal scenario and single fault scenario because only noisy measurements are used.

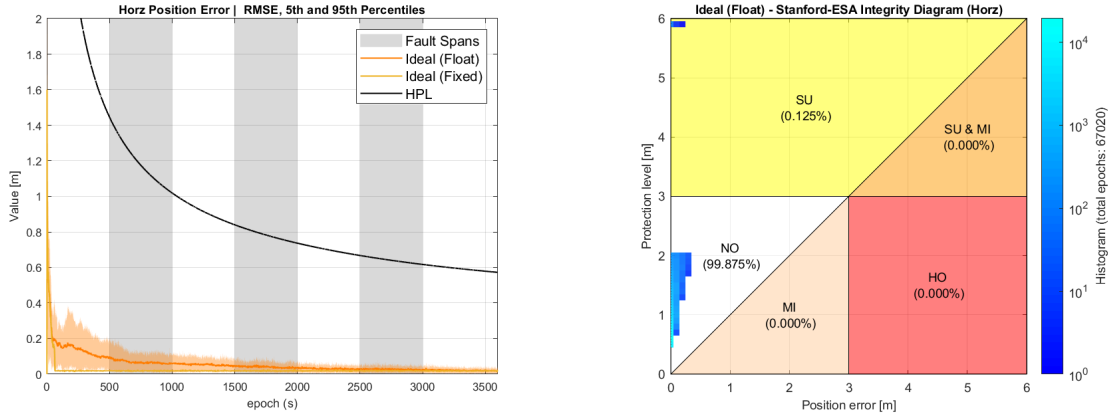
The EKF horizontal RMSE in Figure 5.28a shows a much larger error than the one in single fault scenario. This is expected because the EKF is affected by one more fault during the fault spans. The Stanford diagram in Figure 5.28b shows that over 90% of the epochs fall in the SU

area because the EKF fails the test starting from the beginning of the first fault span and no fault exclusion or re-initialization is performed.

In the case of EKF-FDE, it fails to exclude the injected double faults during fault spans and always declares solution unavailable, as shown in Figure 5.29a and 5.31a. This is because the EKF-FDE can only exclude a single fault and only performs prediction if single fault exclusion fails. The prediction keeps the state covariance at large values so that the filter can converge quickly when health measurements appear after the fault span. In the Stanford diagram in 5.29b, around 44% of the epochs fall within the SU area. These epochs correspond to the solution unavailable marks in Figure 5.29a. There are 0.030% of epochs fall within the Misleading Information area, which corresponds to the large jump of RMSE at the epoch right after the third fault span. This may suggest that the filter needs a few epochs to re-converge from the affect of the fault even if it is only performing prediction.

In the case of MHSS, there are in total 191 subsets to be monitored with double fault assumption and 19 satellites. The global P_{fa} is evenly distributed among all subsets to perform fault detection. Figure 5.31b shows that the MHSS can successfully detect the two faulty satellites with one false alarm in epoch 2217 and an unsuccessful fault exclusion in epoch 570. The unsuccessful fault exclusion corresponds to the solution unavailable mark in Figure 5.30a and the 0.001% of the epoch in SU in the Stanford diagram in Figure 5.30b. The computed HPLs also show visible jumps before and after each fault spans, but can always cover the estimated position error. There are over 99.9% of all epochs fall within the NO area in the Stanford diagram, which suggests that the MHSS can correctly detect and exclude the faulty satellites in the double fault scenario.

The horizontal RMSE for RKF in double fault scenario is shown in Figure 5.32a. It increases to around 0.4 m at the end of the first fault span and re-converges towards zero after the first fault span. The increase in RMSE is larger than the one in single fault scenario as shown in Figure 5.22a which is reasonable because the filter is affected by one more fault. The calculated HPL can always cover the horizontal RMSE. This can be confirmed with the Stanford diagram in Figure 5.32b which shows that 100% of all epochs fall within the NO area. This indicates that the RKF can provide reasonable and reliable position estimate under double faults.

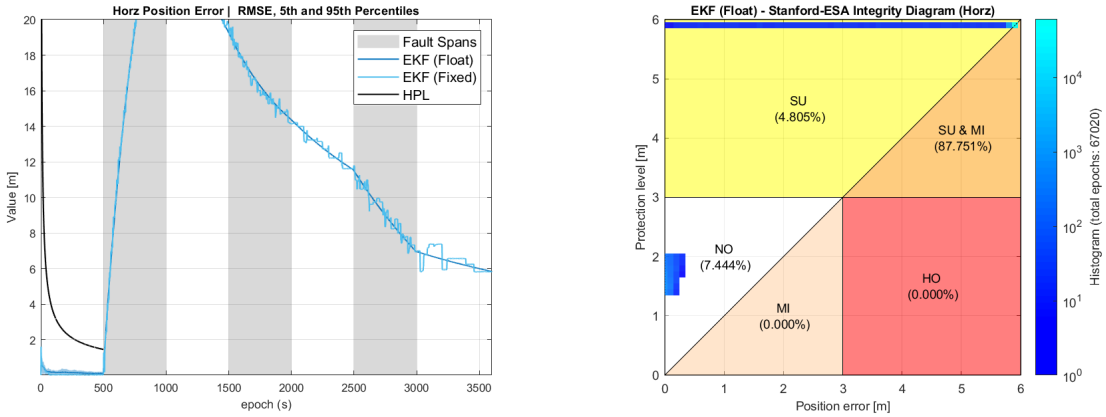


(a) Horizontal RMSE

(b) Standford diagram

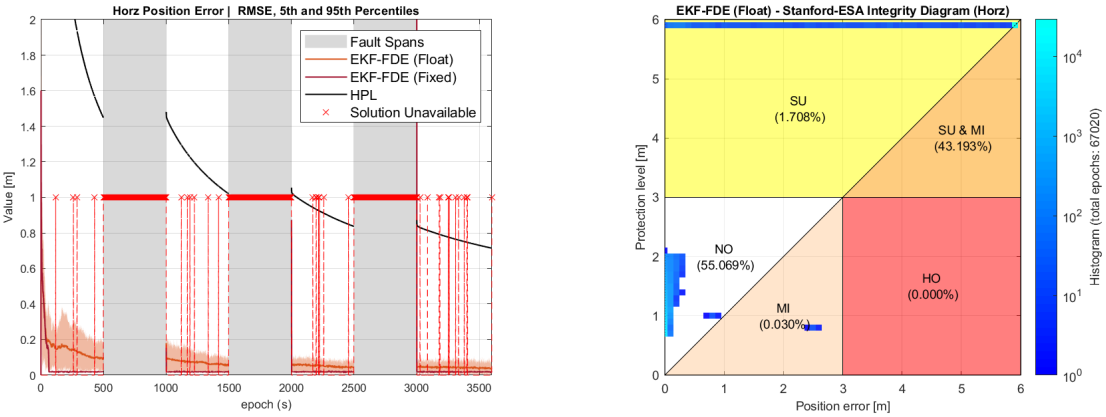
Figure 5.27: Ideal EKF in double fault scenario

The comparison of float solutions for all filters is shown in Figure 5.33. The CDF of the horizontal RMSE in Figure 5.33b shows that the MHSS performs the closest to the ideal EKF with over 95% of the RMSE below 0.2 m. The RKF performs slightly worse than MHSS but can still provide



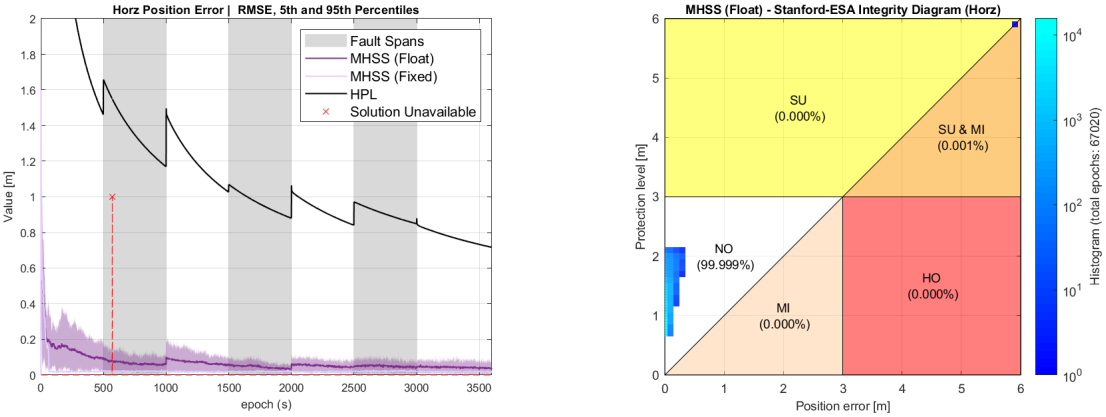
(a) Horizontal RMSE (b) Standford diagram

Figure 5.28: EKF in double fault scenario



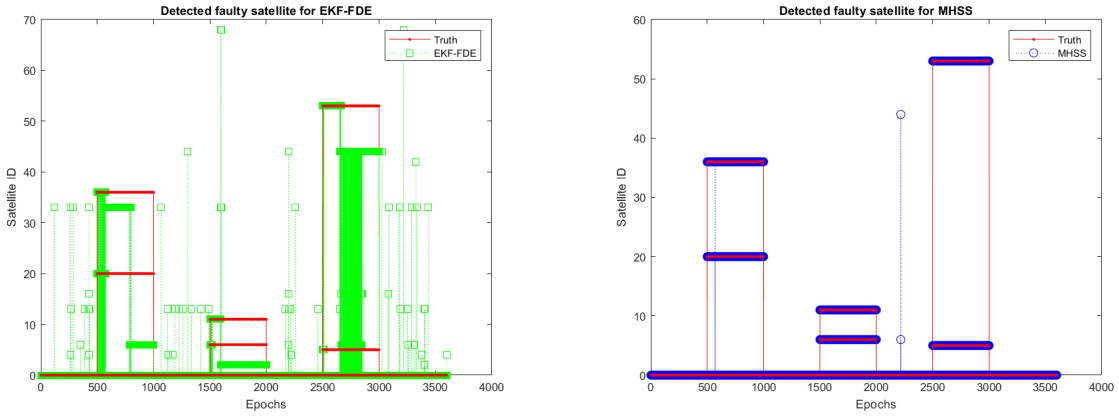
(a) Horizontal RMSE (b) Standford diagram

Figure 5.29: EKF-FDE in double fault scenario



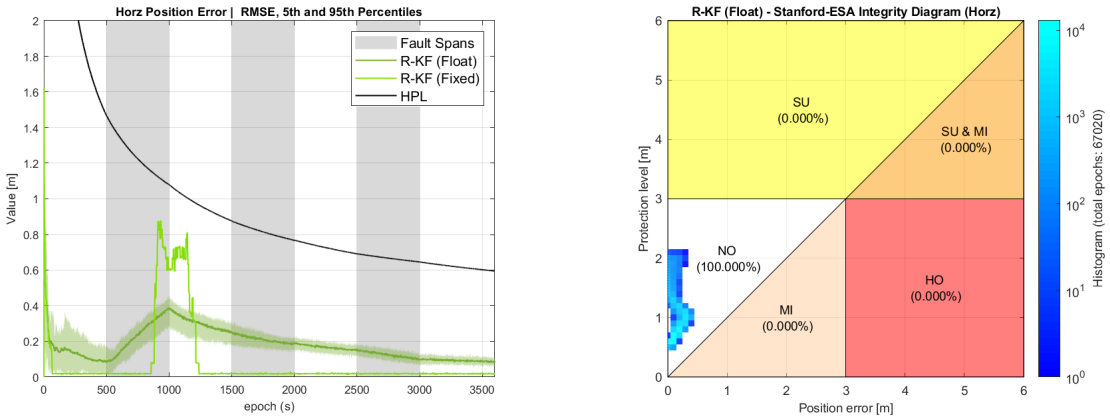
(a) Horizontal RMSE (b) Standford diagram

Figure 5.30: MHSS in double fault scenario



(a) EKF-FDE (b) MHSS

Figure 5.31: Detected faulty satellite in double fault scenario

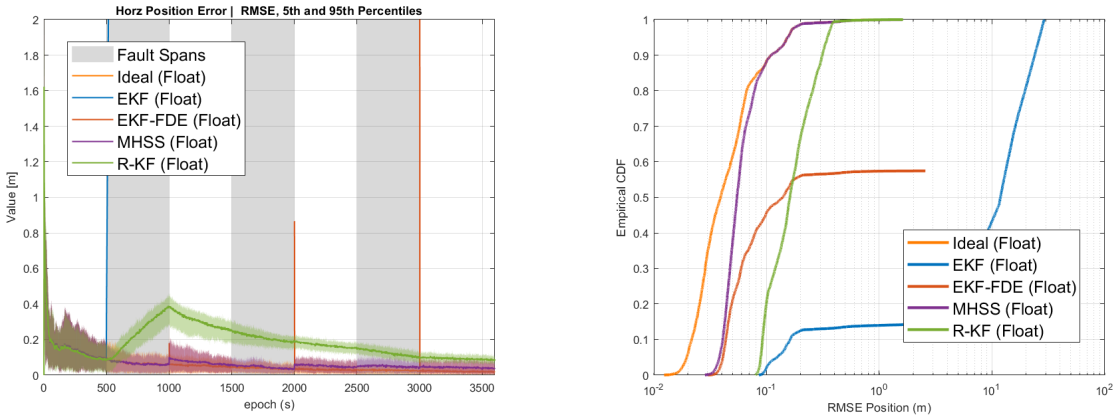


(a) Horizontal RMSE (b) Standford diagram

Figure 5.32: RKF in double fault scenario

decimeter accuracy with over 95% of the RMSE below 0.4 m. The CDF of RMSE for EKF-FDE stops at around 57% because the reaming epochs are declared solution unavailable. The EKF CDF shows the largest error to around 30 m due to the double faults in the measurement.

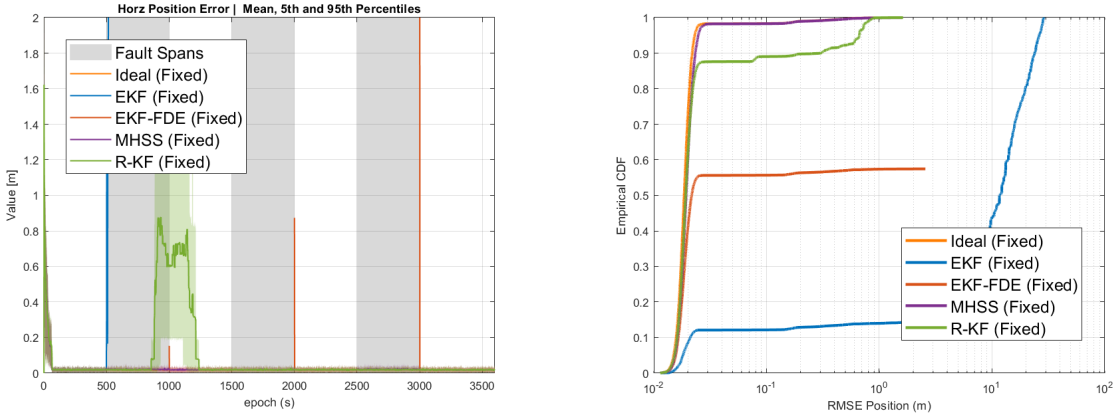
The comparison of fix solutions for all filters is shown in Figure 5.34. The horizontal RMSE of RKF shows a period of large jumps between around 900 to 1200 epochs. This period corresponds to the large RMSE in the float solution as shown in Figure 5.33a and the significant drop of the percentage of correct fixed ambiguities in Figure 5.35b. This suggests that the large RMSE in the float solution can cause incorrect ambiguity fixing and leads to a larger RMSE in the fix solution. The CDF of fix solution RMSE for EKF-FDE also stops at around 57% because the remaining epochs are declared solution unavailable. The MHSS shows similar performance to the ideal EKF for the horizontal RMSE in fix solution because it can correctly detect and exclude the faulty satellites. The confusion matrices for double fault scenario in Figure 5.36 look exactly the same as the ones in single fault scenario as shown in Figure 5.26 because they both focus on fault detection for all-in-view solution. One can refer to Figure 5.31 to have a clearer view of the fault detection and exclusion of faulty satellites for EKF-FDE and MHSS in double fault scenario.



(a) Horizontal RMSE

(b) CDF of horizontal RMSE

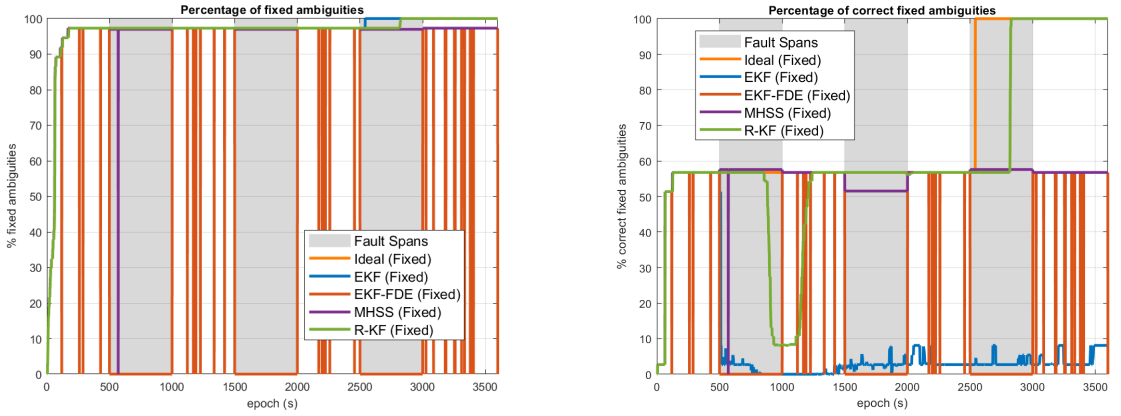
Figure 5.33: Comparison of float solution for all filters in double fault scenario



(a) Horizontal RMSE

(b) CDF of horizontal RMSE

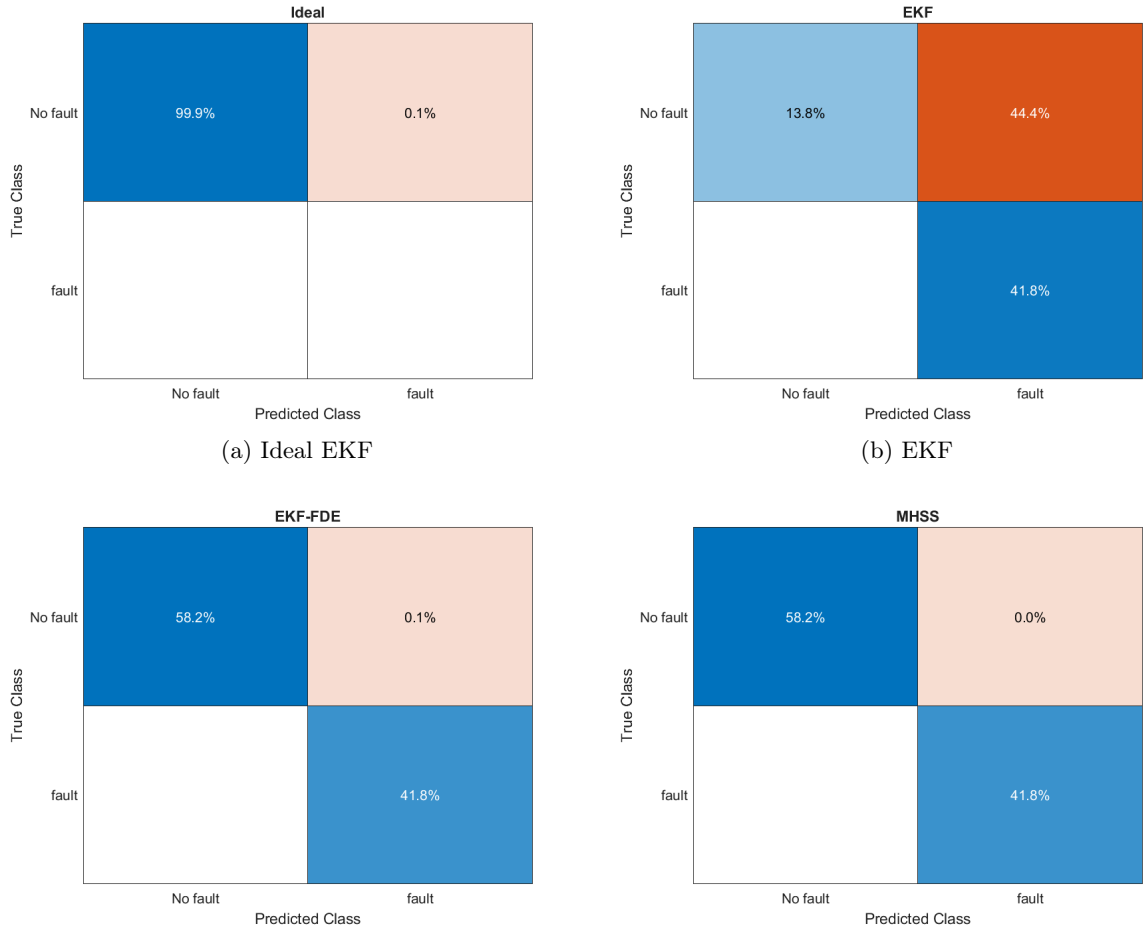
Figure 5.34: Comparison of fix solution for all filters in double fault scenario



(a) Percentage of fixed ambiguities

(b) Percentage of correct fixed ambiguities

Figure 5.35: Comparison of ambiguity resolution for all filters in double fault scenario



(a) Ideal EKF

(b) EKF

(c) EKF-FDE

(d) MHSS

Figure 5.36: Confusion matrix for fault detection in all-in-view solution in double fault scenario

5.4 Profiling results

The required processing time for different filters in one Monte Carlo simulation without enabling the calculation of PL and ambiguity resolution are recorded using the MATLAB profiling application. The profiling results are summarized in Table 5.1. Note the significant increase of processing time for MHSS from single-fault to double-fault scenario. This is due to the increase number of subsets that need to be considered in double-fault scenario which is also the major downside of the MHSS technique. The RKF requires much less processing time than MHSS in all scenarios while being able to mitigate the affect of faults in the measurements. This shows the potential of using robust estimator for integrity monitoring under multiple faults when the computation resources are limited.

The hardware information of the computer used to run the Monte Carlo simulations is shown in Table 5.2 for reference.

Filter type / Scenarios	Nominal [sec]	Single-fault [sec]	Double-fault [sec]
Ideal and EKF	144.989	64.390	88
EKF-FDE	38.865	37.583	34.173
MHSS	635.169	599.910	1728.679
RKF	171.474	151.919	90.897

Table 5.1: Profiling results for different filters in different scenarios

Specification	Details
Model	Dell Latitude 7330 Laptop
Operating System	Windows 11 Enterprise
Installed RAM	16 GB DDR4, 3200 MHz (integrated)
Processor	12th Gen Intel(R) Core(TM) i7-1265U, 1.80 GHz
System Type	x64-based processor

Table 5.2: Hardware configuration of the computer used for simulation

Chapter 6

Conclusion and future work

This thesis presents the current status and challenges in integrity monitoring for precise positioning in GNSS under multiple faults scenario. It investigates the PPP estimation problem with ambiguity resolution and introduces the classical iterative EKF to solve this mix model. Three approaches to evaluate the consistency of the Kalman Filter, including NEES, NIS, and innovation whitness test, are discussed to ensure correct selection of measurement noise and process noise in the EKF. A robust estimator is presented and integrated with the EKF to improve its robustness against outliers and faults. Several integrity monitoring techniques for PPP are presented and a slope-based Protection Level calculation method is introduced. These integrity monitoring techniques are implemented in MATLAB as *Filter* objects, including EKF (without fault exclusion), EKF-FDE (EKF with single Fault Detection and Exclusion), MHSS (Multiple Hypothesis Solution Separation), and RKF (Robust Kalman Filter). An ideal EKF which only uses noisy measurements is considered as the reference solution for all these filters.

To evaluate the performance of these integrity monitoring algorithms, an existing GNSS simulator is used to generate the receiver ground truth, GNSS data and measurements. Three scenarios are configured to investigate the integrity monitoring performance for these filters under nominal, single fault and double faults conditions. Each scenario is evaluated with 20 Monte Carlo simulations with one-hour-long simulated data. The Partial Ambiguity Resolution technique is applied for ambiguity resolution.

Under nominal scenario, all filters have similar performance to the ideal EKF which allows us to verify the implementation of each filter. All filters have their horizontal RMSE covered by the horizontal PL, and have over 99.8% of epochs fall within the Nominal Operation area in the Stanford diagram. The confusion matrices show that the empirical probability of false alarm is consistent with the simulation configuration. The relative efficiency of the RKF to the ideal EKF is around 95% in the nominal scenario.

The EKF fails to provide reasonable position estimation in the single fault and double fault scenario with less than 10% of the epochs fall within the Nominal Operation area in the Stanford diagram. This is because it does not perform any fault exclusion. The EKF-FDE can detect and exclude a single faulty satellite and provide reliable position estimation, with more than 99.9% of epochs fall within the Nominal Operation area in the Stanford diagram. However, it fails to exclude the two faulty satellites in the double fault scenario, with around 55% of the epochs fall within the Nominal Operation area in the Stanford diagram. This is due to its limitation of single fault detection and exclusion. The MHSS can successfully detect and exclude faulty satellites in both single and double fault scenarios and provide reliable position estimation. The Stanford diagrams show that 100% and more than 99.9% of the epochs fall within the Nominal Operation

area respectively. However, it comes with the downside of heavy computation load because of the execution of the bank of filters. The RKF does not detect and exclude any faulty satellites but it can mitigate their effects by downweighting those observations causing large residuals within a single filter and avoid the heavy computation load due to the bank of filters. It can provide reliable position estimation with 100% of the epochs fall within the Nominal Operation area in the Stanford diagram under both single and double fault scenarios. It can tolerate multiple simultaneous faults affecting the observations as well as model mismatches, e. g., due to poor dynamics. However, classical test statistics cannot be applied for this type of estimators since robust estimators do not make any assumption on the underlying distribution of the noises. Therefore, further research needs to be conducted to assess the reliability of M-estimators for integrity monitoring.

Future work can focus on the limitation and assumption in the current implementation. Firstly, the measurement generation can consider more error sources in the measurement model to make the measurement even more realistic, such as satellite ephemeris error, phase center offset, relativity effect, etc. Furthermore, the geometry of satellites can change over time and lead to the change of satellites in the all-in-view solution. This will make the simulation closer to reality but can make the determination of the bank of filters in MHSS very challenging. The fault detection is performed based on chi-square test in MHSS. Other test such as solution separation test can be explored. Finally, different types of faults such as ramp error or cycle slips can be injected into the measurement to investigate their effects on the performance of integrity monitoring. Another aspect to be explored is the protection of the fix solution by providing informative protection levels while considering the effect of wrong ambiguity fixes the estimation and integrity.

Bibliography

[1] J. Blanch, T. Walter, P. Enge, Y. Lee, B. Pervan, M. Rippl, and A. Spletter, "Advanced raim user algorithm description: Integrity support message processing, fault detection, exclusion, and protection level calculation," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, pp. 2828–2849, 2012.

[2] D. Imparato, A. El-Mowafy, and C. Rizos, "Integrity monitoring: From airborne to land applications," in *Multifunctional Operation and Application of GPS*, pp. 23–43, 2018.

[3] J. Blanch *et al.*, "Reducing computational load in solution separation for kalman filters and an application to ppp integrity," in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*, 2019.

[4] Y. Ge, Z. Wang, and Y. Zhu, "Reduced araim monitoring subset method based on satellites in different orbital planes," *GPS Solutions*, vol. 21, no. 4, pp. 1443–1456, 2017.

[5] S. Feng, W. Ochieng, T. Moore, *et al.*, "Carrier phase-based integrity monitoring for high-accuracy positioning," *GPS Solutions*, vol. 13, pp. 13–22, 2009.

[6] D. Medina, *Robust GNSS Carrier Phase-based Position and Attitude Estimation*. PhD thesis, 03 2022.

[7] A. Bellés, D. Medina, P. Chauchat, S. Labsir, and J. Vilà-Valls, "Robust m-type error-state kalman filters for attitude estimation," in *2023 31st European Signal Processing Conference (EUSIPCO)*, pp. 840–844, 2023.

[8] A. Belles Ferreres and D. Medina, "Robust ppp-ar using m-estimators for multi-fault scenarios," 2025.

[9] B. W. Parkinson and P. Axelrad, "Autonomous gps integrity monitoring using the pseudorange residual," *Navigation*, vol. 35, no. 2, pp. 255–274, 1988.

[10] S. Feng, W. Ochieng, J. Samson, M. Tossaint, M. Hernandez-Pajares, J. M. Juan, J. Sanz, À. Aragón-Àngel, P. Ramos-Bosch, and M. Jofre, "Integrity monitoring for carrier phase ambiguities," *The Journal of Navigation*, vol. 65, no. 1, pp. 41–58, 2012.

[11] K. Gunning, J. Blanch, T. Walter, L. De Groot, and L. Norman, "Design and evaluation of integrity algorithms for ppp in kinematic applications," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, pp. 1910–1939, 2018.

[12] K. Gunning, *Safety critical bounds for precise positioning for aviation and autonomy*. Stanford University, 2021.

[13] C.-W. Wang and S.-S. Jan, "Kalman filter-based integrity monitoring for madoca-ppp in terrestrial applications," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 436–445, IEEE, 2023.

[14] S. Wang, X. Zhan, Y. Zhai, and Z. Gao, "solution separation-based integrity monitoring for integer ambiguity resolution-enabled gnss positioning," in *Proceedings of the 2023 International Technical Meeting of The Institute of Navigation*, pp. 492–513, 2023.

[15] W. Zhang and J. Wang, "Integrity monitoring scheme for single-epoch gnss ppp-rtk positioning," *Satellite Navigation*, vol. 4, no. 1, p. 10, 2023.

[16] A. Bellés and D. Medina, "Robust ppp-ar using m-estimators for multi-fault scenarios," in *2025 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 251–257, IEEE, 2025.

[17] P. J. Teunissen, O. Montenbruck, *et al.*, *Springer handbook of global navigation satellite systems*, vol. 10. Springer, 2017.

[18] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[19] L. Massarweh, S. Verhagen, and P. J. G. Teunissen, "Lambda toolbox version 4.0 – documentation," tech. rep., Delft University of Technology, Delft, The Netherlands, 2024. Available online at `https://www.tudelft.nl/citg/over-faculteit/afdelingen/geoscience-remote-sensing/research/lambda/lambda`.

[20] Z. Wu, S. Bian, and B. Ji, "Analysis on number of ambiguities fixed in gnss precise point positioning ambiguity resolution," *GPS Solutions*, vol. 29, no. 2, p. 82, 2025.

[21] L. Chang and K. Li, "Unified form for the robust gaussian information filtering based on m-estimate," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 412–416, 2017.

[22] C. D. S. Andrés, *Integrity monitoring applied to the reception of GNSS signals in urban environments*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2012.

[23] Y. Gao, Y. Gao, B. Liu, and Y. Jiang, "Enhanced fault detection and exclusion based on kalman filter with colored measurement noise and application to rtk," *Gps Solutions*, vol. 25, no. 3, p. 82, 2021.

[24] A. B. F. Daniel Arias Medina, Xiangdong An *et al.*, "Integrity concept for carrier phase positioning with integer ambiguity resolution." Unpublished manuscript, 2024.