Integrity Monitoring Using ARAIM Algorithm in Urban Environment

Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(XXX cycle)

Integrity Monitoring Using ARAIM Algorithm in Urban Environment

Hieu Trung Tran

* * * * * *

Supervisors
Prof. Letizia Lo Presti
Prof. Fabio Dovis

Doctoral Examination Committee:
Dr. Beatrice Motella, Referee, ISMB (Instituto Superiore Mario Boella)
Prof. Duc Van Nguyen, Referee, Hanoi University of Science and Technology
Prof. Gustavo Belforte, Polytechnic of Turin
Prof. Jaume Sans Subirana, Polytechnic University of Calalonia
Prof. Tung Hai Ta, Hanoi University of Science and Technology

Politecnico di Torino
4th September, 2018

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Hieu Trung Tran

Turin, 4th September, 2018

Summary

Aviation is one of the earliest application of the Global Navigation Satellite System (GNSS). Since the early days of the Global Positioning System (GPS), satellite navigation has been an essential part of the aviation industry. Being a particular mean of transport which usually involves a large number of human lives, civil aviation always requires a high level of reliability from the navigation system. Such requirement brings about the concept of integrity, which concerns about the consistency and reliability of a navigation system, is defined as the capability of the system to provide timely warning when it should not be used for navigation. The concept of integrity allows the standardization of guidance systems' performance, with the utmost purpose of keeping safety for every flight.

The concept of integrity has gained interests in other GNSS applications as well, especially in those that also require high reliability from the navigation solution, such as Intelligent Transport System (ITS), railways. This leads to the necessity to adapt the integrity monitoring techniques, in particular the Receiver Autonomous Integrity Monitoring (RAIM) algorithms, to use in working conditions other than the typical airport areas, such as urban environment. As a matter of fact, adaptation of RAIM algorithms to urban environment requires a throughout analysis of the environmental difference of the working condition as well as the requirement of the intended applications.

This thesis focuses on developing a Kalman filter-based Advanced RAIM (ARAIM) algorithm for urban environment, which is an adaptation of the conventional ARAIM algorithm for civil aviation. ARAIM algorithm is considered the next generation of RAIM, aiming at providing higher integrity performance for more stringent phase of flight. The first step is to survey the necessary changes to adapt ARAIM algorithm to urban scenario. Experimental study highlight the prerequisite of finding a

noise model to represents the signal noise level in urban area. After a suitable noise model was found after a comparative study, the KF-based ARAIM algorithm was developed. This method evaluates the separation of state correction using different subsets of measurement to detect abnormalities as well as potential faulty satellites for exclusion. The proposed method was also validated using simulation and real data. Performance analysis results show that the proposed algorithm can effectively follows the changes of signal quality which is expected to occur frequently when moving in urban environment, confirming its suitability for integrity monitoring in urban environment.

Acknowledgements

The research work in this thesis has been carried out from 2014 to 2017 at the Politecnico di Torino, Turin, Italy. The program also includes a valuable 3-month period in the Ećole nationale de l'aviation civile (ENAC), Toulouse, France.

First of all, I would like to express my sincere gratitude to my supervisors, Prof. Letizia Lo Presti and Prof. Fabio Dovis for the guidance and support over my years in Polito as well as for offering me the research opportunities. I would like to extend my gratitude also to the professors of Politecnico, Prof. Gustavo Belforte for the great encouragement and kindest understanding. I would like to thank Prof. Olivier Julien, Dr. Carl Milner and the staff at the SIGNAV lab of ENAC for the unique experience during such short period in Toulouse.

I would like to thank all the staff of the NavSAS group, my friends and colleagues in Politecnico for the great 3 years, especially for Mattia and Giulio for the collaboration, Falco for the expertise in Kalman filter and valuable input on my works, and Gabriella for all the support. Thank you all to the NavSAS group for all the great memories and wonderful moments. I would like also to thank my friends and colleagues in NAVIS Centre, Hanoi, Vietnam for the kind support despite the distance.

A special thank to all my flatmates throughout the years, who have been my special second family, far away from home.

This doctoral work has been financially supported by the AREAS+ project and the BELS project, which are all gratefully acknowledged.

Last but not least, I would like to thank my parents, and Tu for the unconditional love and support, and for always being with me.

Contents

List of Tables

List of Figures

Abbreviations

ABAS Aircraft-based Augmentation System

ADC Analog-to-Digital Converter

AGC Automatic Gain Control

AL Alarm Limit

APV Approach with Vertical guidance

ARAIM Advanced Receiver Autonomous Integrity Monitoring

ATC Air Traffic Control

BOC Binary offset carrier

CNS/ATM Communications, Navigation and Surveillance / Air Traffic Management

CW continuous wave

DGPS Differential GPS

DLL Delay Locked Loop

DOF degree of freedom

EGNOS European Geostationary Navigation Overlay Service

EM electromagnetic

ENU East-North-Up

EU European Union

FAA Federal Aviation Administration

FAF Final Approach Fix

FDE Fault Detection and Exclusion

FFT Fast Fourier Transform

FLL Frequency Locked Loop

GAGAN GPS-aided GEO Augmented Navigation

GBAS Ground-based Augmentation System

GIVE Grid Iono Vertical Error

GNSS Global Navigation Satellite System

GPS Global Positioning System

HA Horizontal Accuracy

HAL Horizontal Alarm Limit

HMI Hazardous Misleading Information

HPL Horizontal Protection Level

IAF Initial Approach Fix

IAP Instrument Approach Procedure

ICAO International Civil Aviation Organization

IF Intermediate Frequency

ITS Intelligent Transport System

ITU International Telecommunication Union

KF Kalman Filter

LADGPS Local Area Differential GPS

LMMC Land Mobile Multipath Channel

LOS Line of Sight

LPV Localizer Performance with Vertical guidance

LS Least square

MAPT Missed Approach Point

MHSS RAIM Multiple Hypothesis Solution Separation RAIM

MSAS Multi-functional Satellite Augmentation System

NB narrowband

NLOS Non-Line of Sight

NPA Non Precision Approach

PA Precision Approach

PE Positioning Error

PHMI Probability of Hazardous Misleading Information

PL Protection Level

PLL Phase Locked Loop

PRN Pseudo Random Noise

PVT Position, Velocity, Time

RAIM Receiver Autonomous Integrity Monitoring

RF Radio Frequency

RNP Required Navigation Performance

RNSS Regional Navigation Satellite Systems

SBAS Satellite-based Augmentation System

SDR Software-Defined Radio

SFT Sparse Fourier Transform

SIS Signal In Space

SOL Safety of Life

SS Solution Separation

TEC Total Electron Content

TTA Time to Alert

UDRE User Differential Range Error

URE User Range Error

VA Vertical Accuracy

VAL Vertical Alarm Limit

VPL Vertical Protection Level

WAAS Wide-Area Augmentation System

WADGPS Wide Area Differential GPS

WB wideband

WLS Weighted Least Square

WLSR Weighted Least Square Residual

Introduction

Navigation has been an essential technology of human life since the early ages of history. Knowing one's location as well as the direction to the destination, especially across open ocean. Throughout the history, many different techniques to aid human in calculating one's location and deriving direction have been developed, the latest of which is the GNSS. Combining the study from multiple fields of technologies, GNSS has developed integrated itself into many other applications and services. Starting with just one system, the Global Positioning System (GPS) developed by the United States since the late 1970s, nowadays the world has seen many other GNSSes come to view: the GLONASS, developed by Russia starting from early 1980s, and the Galileo created by the European Union (EU) from 1999. More satellites have been gradually launched for the developing systems, thus increasing the operational capability as well as global coverage. On the other hand, it is also worth mentioning other Regional Navigation Satellite Systems (RNSS), which, as the name suggests, cover only certain areas, such as the Beidou from China, QZSS from Japan, IRNSS from India. All these progressive appearance of navigation satellite systems comes hand in hand with the development of many other technologies that are also fundamental to today's daily life: telecommunication, electronics, e-commerce, etc.

GNSS technology is closely linked with most form of transportation, one of which is civil aviation. Being a particular form of transportation which involves a large number of human lives, air safety has been an important issue throughout the aviation history. As a result, the application of GNSS into civil aviation led to new standard and requirement for GNSS performance to ensure reliability throughout the flight. One of the most important concepts defined for reliability assurance is GNSS Integrity. The integrity of a GNSS system is the capability of the system to

provide timely warnings to users when the system should not be used for navigation. Integrity concerns directly with the consistency and the reliability in term of performance of the navigation system. In fact, integrity is one of four parameters in evaluating the performance of a navigation system as defined by the Required Navigation Performance (RNP) Navigation Specification - the other parameters are: accuracy, availability and continuity.

Integrity monitoring service can be provided from different augmentation systems. The GBAS utilizes monitoring stations on the ground in airport area, then broadcasts the integrity assessment information to aircrafts in approach. The Satellite-based Augmentation System (SBAS), on the other hand, consists of a larger network of ground stations over the area of a continent, and a constellation of geostationary satellites to provide integrity assessment service over a large area. The Aircraft-based Augmentation System (ABAS), however, relies only on the navigation data from GNSS receivers and other equipments on-board for the local integrity monitoring operation. As a result, ABAS algorithms, and particularly Receiver Autonomous Integrity Monitoring (RAIM) allows operational integrity monitoring everywhere, while GBAS and SBAS have only limited service coverage. It is important to note that, RAIM was initially developed to provide integrity monitoring service for Non Precision Approach (NPA), which is a landing procedure that only provides lateral guidance and has low integrity requirement. Recent works have led to the next generation of RAIM called ARAIM, which can provide better integrity performance, aiming at more stringent phases of flight.

Although the integrity concept was created for civil aviation, it has gain interests in other fields as well, especially navigation-based applications that also require high level of reliability, such as Intelligent Transport System (ITS), road tolling applications, and railways. This leads to a multitude of works in adapting previously aviation-made integrity monitoring algorithms, in particular, RAIM algorithms due to its flexibility and autonomous nature, to a different working environment, which is urban area. However, the environmental differences between airports and urban areas render it difficult for the adaptation. Notably, urban environment sees a higher level and more unpredictable multipath effect, due to the presences of buildings and other physical obstacles. Besides, the integrity requirements for urban-based applications (which are usually not Safety of Life (SOL))

should not be as strict as in aviation (which can be considered SOL application). In fact, an ideal adaptation of RAIM and ARAIM algorithms to urban environment has to take into account all the differences, not only in term of environment characteristics, but also the requirement of the intended applications.

Based on the analysis above, this thesis is dedicated to develop an adaptation of ARAIM algorithm for integrity assessment in urban environment. The first step of the work involves an in-depth study of the original ARAIM algorithm for civil aviation. This combines with consideration with the differences between civil aviation and urban environment application leads to a list of changes necessary for the urban-based ARAIM. As a matter of fact, the underneath noise model of the algorithm is the most important point of change, due to the much higher level of noise encountered in urban context. The second step of the work surveys several alternatives for the noise model. To assist the testing phase, a MATLAB-based signal generator has been developed, which can specifically simulate the multipath effect present in urban areas. The chosen noise model is based on the carrier-to-noise ratio (C/N_0), allows it to dynamically represents the signal quality in a way that the original ARAIM's model cannot. In the final step, the ARAIM algorithm is adapted onto a KF model. The KF model was chosen due to its flexibility and capability to integrate with other external sensors, thus allowing potential for future extensions.

Thesis outline

The thesis is organized as follows.

- Chapter 1 presents the overview of the GNSS receiver architecture and brief descriptions about the operation of underlying components. The Chapter also explain about the principles for estimating Position, Velocity, Time (PVT).
- Chapter 2 describes the concept of GNSS other related concepts such as integrity monitoring methods and integrity performance requirement. In particular, since the concept of integrity was designed for civil aviation, a section is devoted to terminology related to performance assessment in civil aviation.

- Chapter 3 discuss about the issues when adapting ARAIM algorithm from aviation to urban environment. Experimental study in this Chapter solidifies the necessity of a suitable noise model for urban use. A comparative study leads to the selection of a model based on C/N_0 ratio.
- Chapter 4 is dedicated to the development of the proposed KF-based ARAIM algorithm, which is later validated using simulation and real data.

List of publication

- Hieu Trung Tran, Tung Ta Hai, and Letizia Lo Presti. "Adaptation of ARAIM Algorithm for Urban Environment Applications". In: Proceeding of International Symposium on GNSS 2015. Kyoto, Japan, Nov. 2015.
- Hieu Trung Tran and Gustavo Belforte. "An ARAIM Adaptation for Kalman Filter". In: Workshop on Maritime Navigation and Communication (ComNavi 2017). Dec. 2017, pp. 27–37. ISBN:978-604-67-1036-3.
- Lucilla Alfonsi et al. "Analysis of the regional ionosphere at low latitudes in support of the Biomass ESA mission". In: IEEE Transactions on Geoscience and Remote Sensing (2018), pp. 1-13. DOI: 10.1109/TGRS.2018.2838321
- Hieu Trung Tran and Letizia Lo Presti. "Kalman Filter-Based ARAIM Algorithm for Integrity Monitoring in Urban Environment". In: ICT Express (2018). ISSN: 2405-9595. DOI: 10.1016/j.icte.2018.05.002

Chapter 1

Overview of the GNSS receivers

As mentioned in the previous Chapter, the technology of navigation is about finding one's location. The GNSS receiver is the device on the users' end that help answering the question about location, and actually showing it as the final result. This Chapter is dedicated to the architecture, operation of GNSS receivers, the PVT calculation process, as well as the error sources that may hinder the operation of the receivers. These can be considered the technical foundation for the discussion related to integrity monitoring in subsequent Chapters.

1.1 Receiver architecture

1.1.1 Basic receiver architecture and operation

In general, a GNSS receiver is an electronic device used to receive and process the navigation signal from GNSS satellites to calculate the Position, Velocity, Time of the receiver. Although there are multitude of processing algorithms to process the navigation signal, the general architecture of a GNSS receiver is depicted in Figure 1.1.

The first block of the process is the frontend. The main functions of the frontend are to receive the navigation signal from the antenna, convert it from the navigation band (such as GPS L1 band at 1.57542 GHz [118]) to a lower Intermediate Frequency (IF) and finally digitize the downconverted signal. A frontend for GNSS receivers usually consists of [129]:

- A passive bandpass prefilter to filter out-of-band interferences.
- Local oscillators to aid the downconverting process to IF
- The Analog-to-Digital Converter (ADC) to digitizes the signal.
- The Automatic Gain Control (AGC)

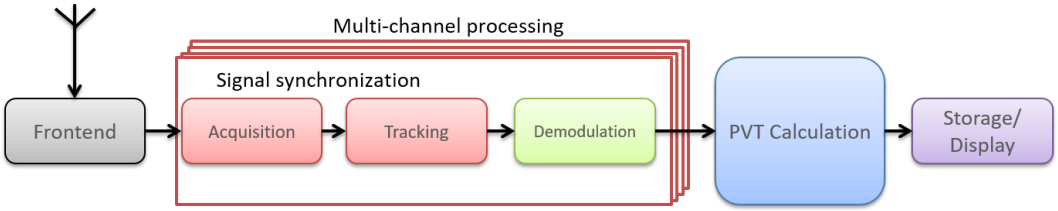


Figure 1.1: Generic architecture of a GNSS receiver

Out of the frontend, the digitized signal is a combination of the SIS of all visible satellites. For instance, the digitized signal of the i^{th} can be written in mathematical form as:

$$r_i(nT_s) = \sqrt{2P_i}d_i(nT_s - \tau_i)c_i(nT_s - \tau_i)\cos(2\pi(f_{IF} + f_{D,i})nT_s + \phi_i) + n_W(nT_s) \quad (1.1)$$

The terms are defined as follows:

- T_s is the sampling interval
- P_i is the received power of the signal
- d_i is the navigation data bits
- c_i is the Pseudo Random Noise (PRN) spreading code
- f_{IF} is the intermediate frequency
- $f_{D,i}$ is the Doppler frequency
- ϕ_i is the initial carrier phase
- τ_i is the code delay

- n_W is a discrete-time random process, which is the result of sampling the noise at the frontend. n_W can be assumed to be Gaussian with zero mean, variance $\sigma_W^2 = N_0 B_{IF}$ where N_0 is the power of received noise, B_{IF} is the bandwidth of the frontend

Acquisition stage

After the fronend is the signal synchronization block, in which the received SIS of each satellite is synchronized with the local replica signal. In this block, the signal is processed in multi-channel manner, each channel processes the signal of one satellite. The first step of the process is the acquisition stage, which roughly estimate several parameters of the digitized signal: the spreading code c_i, the Doppler frequency $f_{D,i}$ and code delay τ_i. For each PRN spreading code, the receiver will generate a testing (discrete) signal with different values for $f_{D,i}$ and τ_i, then compare this testing signal with the received one by calculating the cross-correlation function. The search space is discrete to ensure low search time, as depict in Figure 1.2, with each crosspoint represents a testing signal. The search bin size is determined as [129]:

$$\Delta f_{D,bin} = \frac{1}{2T_{coh}} \tag{1.2}$$

$$\Delta \tau_{bin} = \frac{F_s}{2R_c} \tag{1.3}$$

where T_{coh} is the length (in time) of the signals being used for cross-correlation calculation; R_c is the spreading code rate, for GPS $R_c = 1.023MHz$; F_s is the sampling frequency of the frontend. In other words, the acquisition stage can be considered a process of tuning to find which satellites are broadcasting. Estimating $f_{D,i}$ and τ_i for the broadcasting satellites is equivalent to finding a point in the search space that is closest to the actual point corresponding to the received signal.

The most straight-forward method for acquisition stage is to run a serial search in the search space. This method, although time consuming, is simple enough to be implemented in hardward receiver. On the other hand, there are also parallel search methods using Fast Fourier Transform (FFT) [129], which is more favorable

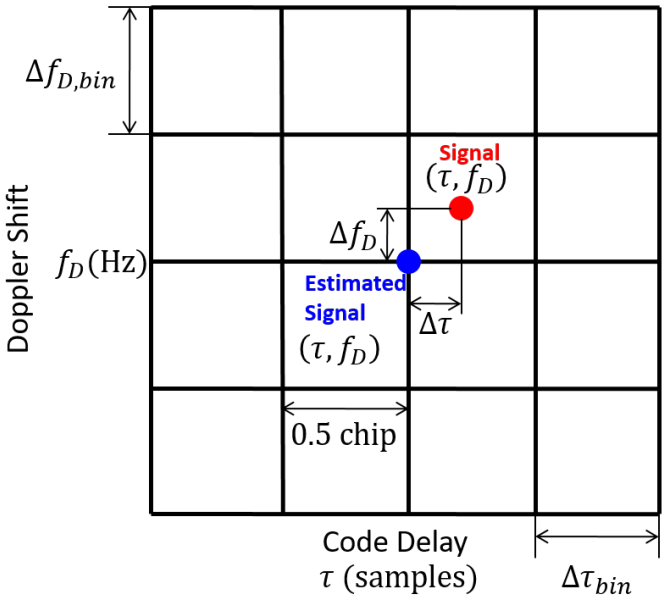


Figure 1.2: Seach space for the acquisition stage

for software-based receivers. Recent methods also propose the use of Sparse Fourier Transform (SFT) [114, 107] or FFT decomposition [1] to improve execution time of the acquisition stage.

Tracking stage

The tracking stage finely estimates the $f_{D,i}$ nd τ_i starting from the rough estimation from the acquisition stage, as well as closely follows the changes of those parameters along the received signal. There are two parallel process in the tracking stage:

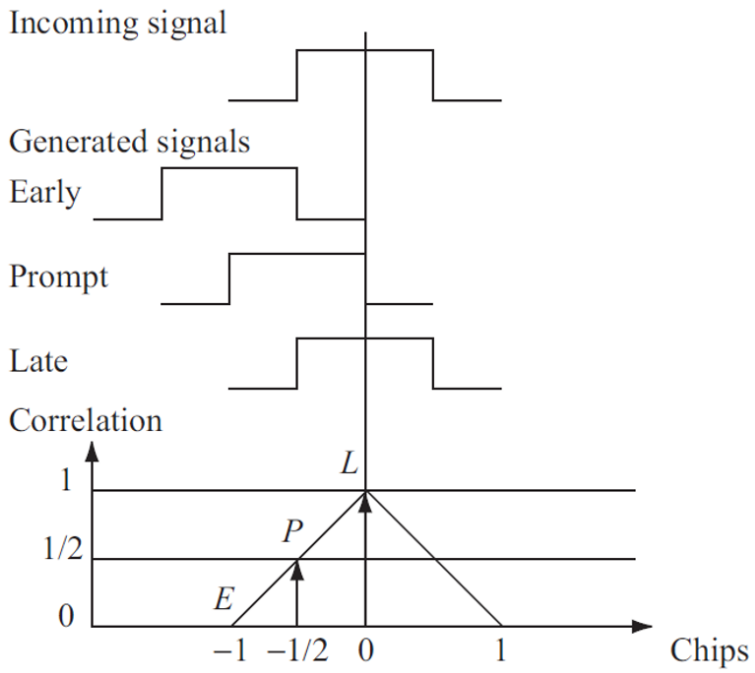
- Code tracking process is where the code phase is dynamically estimated. In principle, this process tries to replicate the spreading code in the received signal by creating 3 separate versions of the spreading code with 0.5 chip delay each and cross correlate each version to the received signal. This exploits the self-correlation properties of the spreading code, especially in the case of GPS: the self-correlation value will decrease to 0 when the code is shifted more than 1 chip-length. Figure 1.3 [22] demonstrates the use of 3 versions of the code to determine the code delay to track GPS L1 signal. A Delay

Locked Loop (DLL) will do the cross correlation calculation, calculate the difference between the 3 versions of the code using a discrimination function, then adjust the local spreading code generator accordingly.

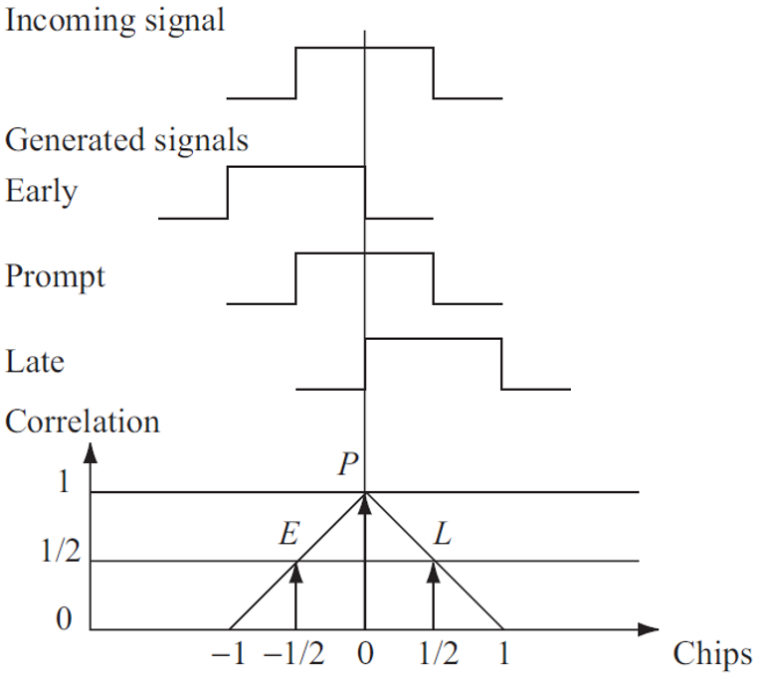
As a side note, while the spreading code property mentioned above is very straightforward for GPS L1 signal, the cases of Galileo signals on the other hand are more complex. Galileo signals utilize the Binary offset carrier (BOC) modulation, adding a sub-carrier of frequency equals or higher than the chip rate. BOC modulation allows interoperability of many GNSS systems on the same band without interference as well as more robust against multipath, but at the same time makes the tracking process more complex due to a more complicated self-correlation function and discrimination curve than the case of GPS L1 (see Figure 1.4 [23]). The detailed tracking methods for such signals are out of the scope of this thesis, interested readers can refer to [23, 38, 63] for more details.

- Carrier tracking process, on the other hand, aims at replicate the carrier of the received signal by adjusting the local carrier frequency and phase. Carrier tracking is also done by locked loops, calculating the cross correlation between the carriers of the received signal and the locally generated signal, then adjust accordingly. The carrier tracking process is usually executed in two steps: a Frequency Locked Loop (FLL) will estimate the carrier frequency (in what is called the *pull-in* phase) then a Phase Locked Loop (PLL) will finely estimate the phase of the carrier.

A generic tracking scheme is presented in Figure 1.5. The input signal (from the frontend) is multiplied with the local spreading code and local carrier generated by the receiver for cross correlation calculation. The result is then passed through the discriminator to evaluate the errors of $f_{D,i}$ nd τ_i as generated by the receiver. The evaluated errors are used as feedback for local generators to adjust for the next round. On the other hand, the discriminator output is also used to decide the lock status of the PLL and DLL. When the locked loops are locked, the output of the tracking process will be of square wave-like form, representing the navigation data element in (1.1), which will be fed to the demodulation block to obtain the navigation messages. In summary, the goal of the tracking process is to wipe off

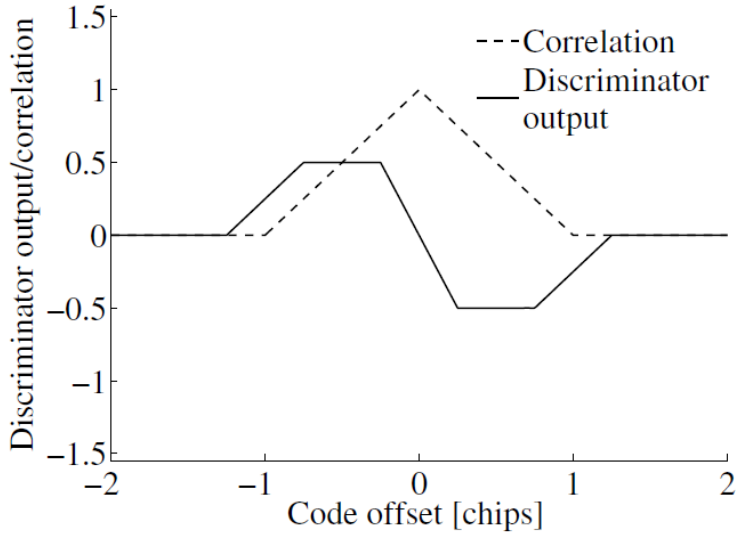


(a) Local code is too early and should be delayed

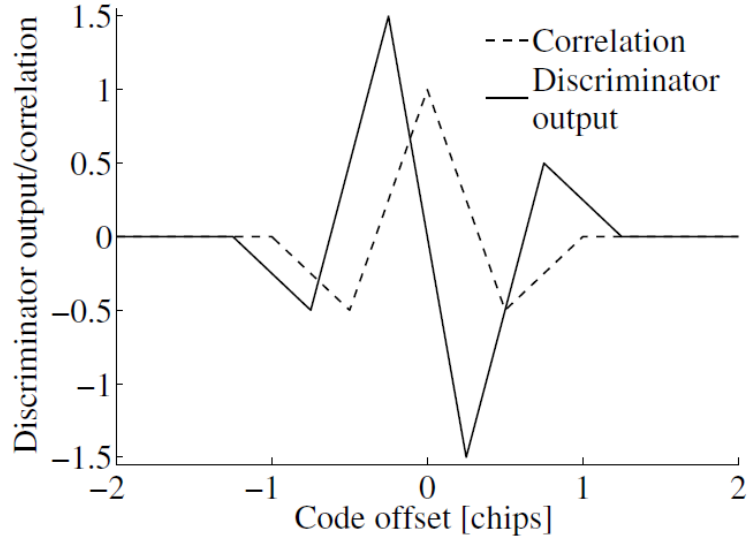


(b) Local code and received code are synchronized

Figure 1.3: Local code versions correlation with the received signal



(a) GPS L1 C/A



(b) Galileo BOC(1,1)

Figure 1.4: Correlation function and early-minus-late discrimination curve of different signals

the spreading code and carrier, to recover the navigation data underneath.

PVT calculation

The next important block in the GNSS receiver is the PVT calculation block. This purpose of this block is to calculate the position of receiver, as well as the

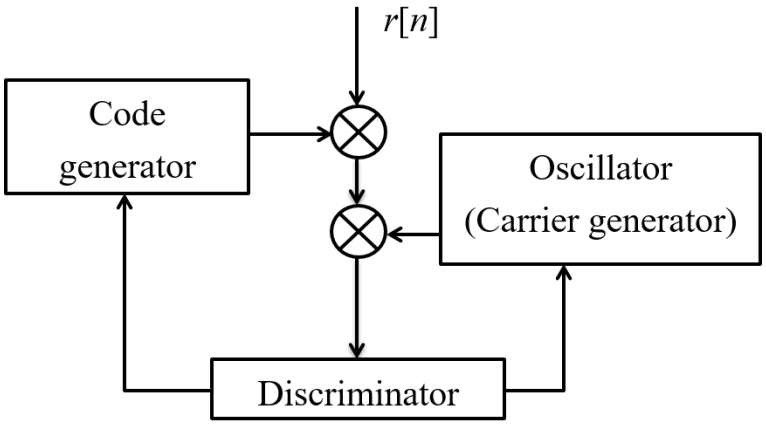


Figure 1.5: Generic scheme for the tracking stage

velocity and time. In general, the process of PVT calculation involves the following steps:

- The navigation data obtained from the tracking block is demodulated into the navigation message, which contains the ephemeris, almanac as well as correction data required to calculate the position of the satellites.
- With the code delay estimated also from the tracking block, the pseudoranges between the satellites and the receivers are calculated using the transmission time method mentioned before.
- Using the calculated satellites' positions, pseudoranges and Doppler frequencies (also from the tracking stage), the receiver's position, velocity and time can be calculated.

Detailed positioning algorithms will be described in Section 1.2.

1.1.2 Software receiver approach

Nowadays, most of the GNSS receivers in the market are hardware receiver, where the signal synchronization process is executed using application-specific integrated circuit (ASIC). While allowing very high computation performance, hardware receivers are not easy to be updated or replaced. This is not ideal, especially when the GNSS environment is changing rapidly with many new satellite systems

come into view. Therefore, the idea of Software-Defined Radio (SDR) approach was adopted to GNSS receivers [24, 98], and has become a favorable approach especially for research purpose. In software receivers, the signal synchronization and subsequent stages are implemented in software on computers or embedded systems instead of hardware as before. This allows great flexibility to the solution [90, 102], easy to reconfigure or update without arduously altering the hardware. Due to the difference in nature between the two platforms, it is necessary to develop processing algorithms suitable for software receivers to ensure manageable performance or even real-time capability [73, 24]. In fact, software receiver has been an important topic recently in the field of GNSS [105, 42, 40].

1.2 PVT schemes

The idea of the positioning process is based on a technique called trilateration. In principle, it is possible for a receive to calculate the position knowing the geometrical distance from its location to another three points, in this case three satellites. In GNSS, the geometrical range can be estimated by measuring the signal transmission time from the satellites to the receiver. Since the transmission velocity of electromagnetic (EM) wave (in vacuum) is known to be equal the speed of light $c = 299792458$ m/s, the distance can be computed by multiplying the transmission time and the speed of light. However, it is important to note that, the transmission medium is, in fact, not vacuum. The EM wave containing the signal is attenuated when passing through the atmosphere, plus the thermal noise on both the satellites and the receivers, potentially clock errors, leading to the computed distance is not exactly the true geometrical range from the satellite to the receiver. The range including other errors is called *pseudorange*.

With the pseudoranges obtained, the position of the receiver is the solution to the following system of simultaneous equations:

$$\begin{cases} \rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} \end{cases} \tag{1.4}$$

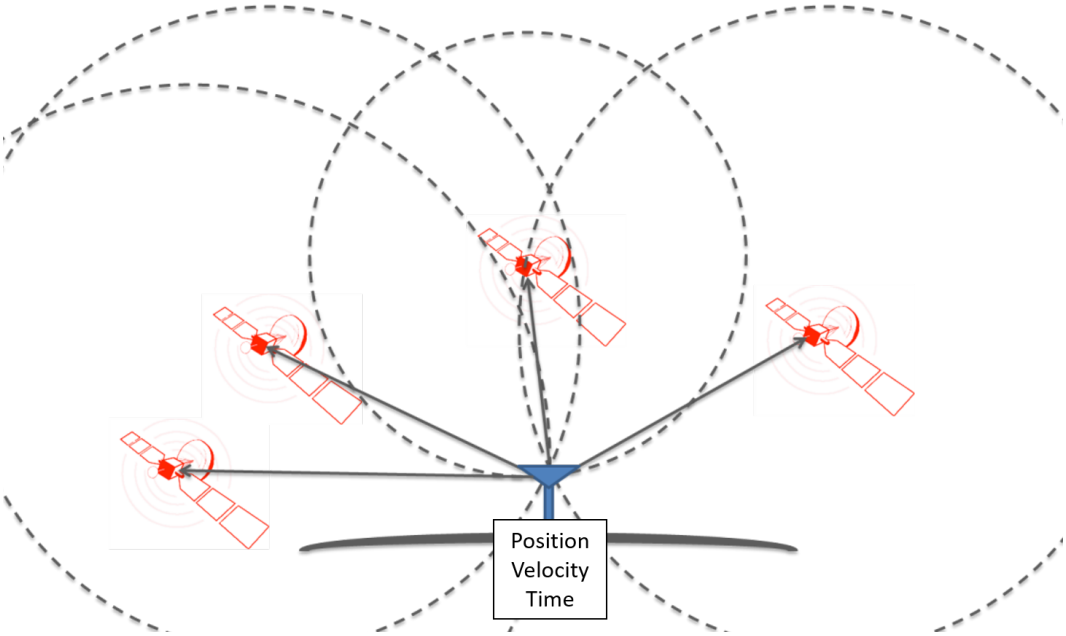


Figure 1.6: Trilateration technique using GNSS satellites

where x_u, y_u, z_u are the coordinates of the receiver, which is also the unknown of the system; (x_i, y_i, z_i) with $i = 1,2,3$ are the coordinates of the satellites, which can be computed using the navigation data broadcasted from the satellites, ρ_i are the pseudoranges measured by the receiver. In fact, (1.4) can be interpret as finding the intersection of 3 spheres in 3D space, the centers of which are the satellites. Since there are 3 unknowns corresponding to the 3 coordinates of the receiver, it is necessary to have at least 3 equations to solve.

In reality, in order to measure the transmission time between the satellites and the receiver, the clocks on both sides have to be perfectly synchronized. In fact, this is usually not the case, since the satellites' clocks are typically very high quality, while the receiver side clock is not. Therefore, it is necessary to introduce another equation into (1.4) for the clock difference, due to which (1.4) becomes:

$$\begin{cases} \rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + c\delta t_u \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + c\delta t_u \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + c\delta t_u \\ \rho_4 &= \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + c\delta t_u \end{cases} \tag{1.5}$$

As thus, it is required at least 4 satellites (and 4 equations) to calculate the position of a receiver.

Since (1.5) is a non-linear system of equations, linearizing this using Taylor expansion will make it easier to solve [127]. As a result, the linearized form of a ranging equation can be expressed as:

$$\begin{aligned}\Delta\rho_i &= \frac{x_i - \hat{x}_u}{r_i}\Delta x_u + \frac{y_i - \hat{y}_u}{r_i}\Delta y_u + \frac{z_i - \hat{z}_u}{r_i}\Delta z_u - c\Delta t_u \\ &= h_{x,i}\Delta x_u + h_{y,i}\Delta y_u + h_{z,i}\Delta z_u \end{aligned} \tag{1.6}$$

where $\hat{\boldsymbol{u}} = (\hat{x}_u, \hat{y}_u, \hat{z}_u, c\delta t_u)^T$ is the (known) linearization point; $\hat{\boldsymbol{h}}_i = (h_{x,i}, h_{y,i}, h_{z,i})$ is a unitary vector pointing from the linearization point to the i^{th} satellite; $\boldsymbol{\Delta u} = (\Delta x_u, \Delta y_u, \Delta z_u, -c\Delta t_u)^T$ is the difference between the linearization point and the true point; r_i is the geometrical between the linearization point and the satellite. From this, the linearized system of equations can be written in matrix form as follows, in the case of N_{sat} satellites:

$$\boldsymbol{\Delta\rho} = \bar{\boldsymbol{H}}\boldsymbol{\Delta u} \tag{1.7}$$

where $\boldsymbol{\Delta\rho} \in \mathbb{R}^{N_{sat}}$ is the vector contains the difference between the measured pseudorange and the distance from the satellites to the linearization point; $\bar{\boldsymbol{H}} \in \mathbb{R}^{N_{sat}\times 4}$ is called the linearization matrix, and is given by:

$$\bar{\boldsymbol{H}} = \begin{bmatrix} h_{x,1} & h_{y,1} & h_{z,1} & 1 \\ h_{x,2} & h_{y,2} & h_{z,2} & 1 \\ h_{x,3} & h_{y,3} & h_{z,3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{h}_{x,N_{sat}} & h_{y,N_{sat}} & h_{z,N_{sat}} & 1 \end{bmatrix} \tag{1.8}$$

In fact, it is also possible to define $\bar{\boldsymbol{H}}$ in East-North-Up (ENU) form, centering at the linearization point. In this method, the row elements of $\bar{\boldsymbol{H}}$ can be defined

as [135]:

$$\begin{aligned} h_{x,i} &= -\cos(\theta_i)\sin(\alpha_i) \\ h_{y,i} &= -\cos(\theta_i)\cos(\alpha_i) \\ h_{z,i} &= -\sin(\theta_i) \end{aligned} \tag{1.9}$$

This form of $\bar{\boldsymbol{H}}$ is useful when subsequent evaluation is considered in horizontal/vertical plane around the receiver's position. On the other hand, since the measured pseudoranges are affected by many different errors encountered by the signal when transmitting to the receiver (see Section 1.3), it is necessary to include also the pseudorange errors in (1.7). Let $\boldsymbol{\epsilon} \in \mathbb{R}^{N_{sat}}$ be the vector containing the pseudorange errors, the most generalized form of (1.7) can be written as:

$$\boldsymbol{\Delta\rho} = \bar{\boldsymbol{H}}\boldsymbol{\Delta u} + \boldsymbol{\epsilon} \tag{1.10}$$

In the end, with the linearization, the problem of calculating the solution becomes finding the correction $\boldsymbol{\Delta u}$ from the linearization point to the true position, which can be done by solving (1.9). There are several method for solving this: LS method, KF, and more recently H-infinity (H_∞). The LS method is considered a classic approach [9], calculate iteratively the value of $\boldsymbol{\Delta u}$ until the linearization point converge to the potential true position. The KF method applies the Kalman filter model [30] into solving the navigation equation, can produce the position in only one iteration. The H_∞, on the other hand, can be considered a special form of KF but with different way to deal with the process noise [44]. H_∞ is out of the scope of this thesis, therefore interested readers can find more about H_∞ in [44, 57].

1.2.1 Least square

The LS solution is the simplest method to estimate the position of the receiver from the pseudoranges information. It is a memory-less approach in which the receiver position vector $\boldsymbol{u}_k$ is estimated at epoch t_k using only the $\boldsymbol{\rho}_k$ pseudorange vector derived at that same epoch. Recall relations binding $\boldsymbol{u}_k$ and $\boldsymbol{\rho}_k$ is linearized leading to a set of N_{sat} linear equations that can be expressed in matrix form at epoch t_k:

$$\boldsymbol{\Delta\rho}_k = \bar{\mathbf{H}}_k \boldsymbol{\Delta u}_k + \boldsymbol{\epsilon}_k \tag{1.11}$$

$\boldsymbol{\Delta u}_k$ is then derived with the classical weighted least square formula [35]:

$$\boldsymbol{\Delta}\hat{\boldsymbol{u}}_k = (\bar{\mathbf{H}}_k^T \boldsymbol{\Sigma}_k^{-1} \bar{\mathbf{H}}_k)^{-1} \bar{\mathbf{H}}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Delta\rho}_k \tag{1.12}$$

in which $\boldsymbol{\Sigma}_k$ is the covariance matrix of $\boldsymbol{\epsilon}_k$.

Obviously this evaluation can be performed only if at least four satellites are in sight so that $N_{sat} \geq 4$. It should be remarked that, since the least square solution is a memory-less approach depending only on measurements at time t_k, it does not perform any smoothing on the obtained PVT results which are in general quite noisy.

1.2.2 Kalman filter

KF is widely applied in many engineering fields, it was developed in the 1960s to recursively estimate the state vector of dynamic systems. In GNSS receivers, KF is utilized in different ways, to integrate inertial sensors with the GNSS receiver at different levels [41], or it is embedded in the tracking part of the receivers with vector tracking loop architecture [119]. Commonly, the KF uses a state vector containing the receiver position coordinates and clock bias along with corresponding derivatives [30, 41]. In this work, the receiver is designed to work with GPS L1 and Galileo E1b signal, therefore the state vector consists of 9 states as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{u}_k \\ -c\Delta t_{GPS,k} \\ -c\Delta t_{Gal,k} \\ \dot{\boldsymbol{u}}_k \\ -c\dot{\Delta} t \end{bmatrix} \tag{1.13}$$

where $\boldsymbol{u}_k \in \mathbb{R}^3$ is the position coordinate vector ($\mathbb{R}$ represents the set of real numbers); Δt_{GPS} and Δt_{Gal} is the difference between the receiver's clock and system time of GPS and Galileo, respectively; c is the speed of light; $\dot{\Delta} t$ is the clock drift.

With N_{sat} being the total number of satellites, the notation used for the filter

description at epoch t_k is:

- $\mathbf{x}_k \in \mathbb{R}^9$ is the process state vector
- $\mathbf{\Phi}_k \in \mathbb{R}^{9\times 9}$ is the state transition matrix relating $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$
- $\mathbf{w}_k \in \mathbb{R}^9$ is the process noise vector, assumed to be a white sequence with known covariance matrix $\mathbf{Q}_k$
- $\mathbf{z}_k \in \mathbb{R}^{2N_{sat}}$ is the measurement vector containing the pseudorange and Doppler measurement for each satellite.
- $\mathbf{H}_k \in \mathbb{R}^{2N_{sat}\times 9}$ is the matrix giving the linear connection between the measurement and the state vector
- $\mathbf{v}_k \in \mathbb{R}^{2N_{sat}}$ is the measurement error vector, which is assumed white with known covariance $\mathbf{R}_k$ and zero crosscorrelation with $\mathbf{w}_k$.
- $\mathbf{R}_k$ is the covariance matrix of $\mathbf{v}_k$, it is assumed to be diagonal and it is derived according to the model in [70].

The linear connection matrix $\mathbf{H}_k$ is given by [31]:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{h}_{u,GPS} & \mathbf{1}_{N_{sat}\times 1} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{0}_{N_{sat}\times 3} & \mathbf{0}_{N_{sat}\times 1} \\ \mathbf{h}_{u,Gal} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{1}_{N_{sat}\times 1} & \mathbf{0}_{N_{sat}\times 3} & \mathbf{0}_{N_{sat}\times 1} \\ \mathbf{0}_{N_{sat}\times 3} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{h}_{u,GPS} & \mathbf{1}_{N_{sat}\times 1} \\ \mathbf{0}_{N_{sat}\times 3} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{0}_{N_{sat}\times 1} & \mathbf{h}_{u,Gal} & \mathbf{1}_{N_{sat}\times 1} \end{bmatrix} \tag{1.14}$$

where each row of $\mathbf{h}_u \in \mathbb{R}^{N_{sat}\times 3}$ is the unit vector pointing from the linearization point to the location of each satellite [30], in which $\mathbf{h}_{u,GPS}$ and $\mathbf{h}_{u,Gal}$ contain the vectors corresponding to satellites of GPS and Galileo constellation, respectively.

In this thesis, the alternative form of Kalman filter[30] is used. The state covariance matrix $\mathbf{P}_k$ is calculated from the apriori covariance matrix $\mathbf{P}_k^-$ as:

$$\boldsymbol{P}_k^{-1} = \left(\boldsymbol{P}_k^-\right)^{-1} + \boldsymbol{H}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{H}_k \tag{1.15}$$

The apriori covariance matrix $\mathbf{P}_k^-$ can be propagated from the previous epoch:

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{1.16}$$

$\mathbf{P}_k$ can be initialized by either assuming to be diagonal with large value [78], or estimated using Least-square results of several initializing steps.

The Kalman Gain $\boldsymbol{K}_k$ is given by:

$$\boldsymbol{K}_k = \boldsymbol{P}_k \boldsymbol{H}_k^T \boldsymbol{R}_k^{-1} \tag{1.17}$$

The state estimate $\hat{\mathbf{x}}_k$ can be calculated from the predicted state vector $\mathbf{x}_k^-$ as:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}^-) \tag{1.18}$$

$\hat{\mathbf{x}}_k^-$ is propagated from the previous epoch by $\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k\hat{\mathbf{x}}_{k-1}^-$

Considering the definition of the state vector $\boldsymbol{x}_k$ in (1.13), the state transition matrix $\mathbf{\Phi}_k$ is defined as:

$$\mathbf{\Phi}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.19}$$

in which Δt is the time interval between epochs. In this work the position is calculated every second, thus $\Delta t = 1$ s.

The measurement error covariance matrix $\boldsymbol{R}_k \in \mathbb{R}^{2N_{sat}\times 2N_{sat}}$ can be defined as:

$$\boldsymbol{R}_k = \begin{bmatrix} \mathbf{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \dot{\mathbf{\Sigma}}_k \end{bmatrix} \tag{1.20}$$

in which $\mathbf{\Sigma} \in \mathbb{R}^{N_{sat}\times N_{sat}}$ is a diagonal matrix, whose diagonal elements are the pseudorange error variances, while $\dot{\mathbf{\Sigma}}$ is a diagonal matrix containing the variances

related to derivatives of pseudoranges. Different solutions have been suggested in literature to assign numerical values to these variances. The interested reader can refer to [39] for more insight on some of them. Moreover, the choices for the pseudorange error variances for urban scenario will be discussed further in Chapter 3.

1.3 Error sources

When the navigation signal is transmitted from the satellite to the receiver, it is subjected to multiple environmental effect from the propagation medium. This leads to additional errors when the receiver estimates the pseudorange measurements for the PVT process. Therefore it is necessary to understand the nature of each error source and the potential error inflicted. In general, the pseudorange error is contributed most significantly by the atmospheric error when the signal propagates through the layers, especially the troposphere and the ionosphere, and multipath error when the signal reflects over physical obstacles along the trajectory. Detailed description for each error source follows.

1.3.1 Atmospheric error

When the electromagnetic signals from GNSS satellites are propagated to the user's receiver, the signals are delayed under the effect of the atmospheric layers. From the physical point of view, this effect is caused by the refractive index of the atmosphere layers being different from that of vacuum. The propagation delay can be modeled in mathematical form as [86]:

$$d_{atm} = \left[\int_s n_{gr} ds - \int_s ds\right] + \left[\int_s ds - s_0\right] \tag{1.21}$$

where d_{atm} denote the propagation delay; s_0 is the propagation path in vacuum, which is the straight line between the satellite and the receiver; s is the actual, bended trajectory; n_{gr} is the group refractive index of the atmosphere. Note that n_{gr} varies depending on the height, hence the inclusion of the integral. The first term on the right side of (1.21) is the *optical delay*, caused by the refractive index $n_{gr} > 1$ thus decreasing the velocity. The second term of (1.21) is the delay due

to the ray being bended thanks to the phase refractive index n_{ph}, called *geometric delay*.

The contribution to the propagation delay mainly comes from the ionospheric and tropospheric layers. However, the nature of the refractive indexes in these layers are different, hence causing different type of delays.

Tropospheric error

In the tropospheric layer, $n_{gr} = n_{ph}$ and only varies with height. As a result, the group velocity v_{gr} and phase velocity v_{ph} are equals, and are defined as:

$$v_{gr} = \frac{c}{n_{gr}} \tag{1.22}$$

$$v_{ph} = \frac{c}{n_{ph}} \tag{1.23}$$

$$\tag{1.24}$$

where c is the speed of light. This results in a positive delay in the final ranging estimation. On the other hand, note that the geometrical delay in (1.21) is, in fact, negligible for satellites with elevation angles higher than 5 degree. As a result, the delay in tropospheric error can be written as:

$$d_{tropo} = \int_s (n_{gr} - 1)\, ds = 10^{-6} \int_s N ds \tag{1.25}$$

where $N = 10^{-6}(n-1)$ is called refractivity. In tropospheric layer, N is divided into two component: *hydrostatic* component caused by dry gases, *and non-hydrostatic* component caused by water vapor. These two components are also called *dry* and *wet* component, respectively. In fact, the hydrostatic component contributes about 90% of the total delay [77]. The effect of this component depends predictably on the local temperature and atmospheric pressure and does not change much, therefore it is possible to develop models for this component. In the field of navigation, the tropospheric model proposed by [33] is widely used in receivers and augmentation systems [89]. In general, tropospheric delay ranges about 2-8m, depending on the elevation angle of the satellites, general location of the user

Ionospheric error

The distinct characteristic of the ionosphere is the presence of free electrons and ionized particles, the density of which can vary during the day. The ionized density may decrease due to energy loss caused by the sun during daytime, thus recombine into molecules. The increased number of molecules and the absences of the sun during nighttime cause the molecules to breakup and increase the ionized density again [35].

In the ionospheric error, the phase refractive index n_{ph} depends also on the frequency. Therefore unlike in the tropospheric layer, it is possible to derive the relation between n_{gr} and n_{ph} as:

$$n_{gr} = \frac{1}{n_{ph}} \tag{1.26}$$

At the frequency of GNSS signals, the refractive indexes can be derived as [35]:

$$n_{gr} = 1 + \frac{40.3}{f^2} N_e \tag{1.27}$$

$$n_{ph} = 1 - \frac{40.3}{f^2} N_e \tag{1.28}$$

where N_e is the electron density (in e^-/m^3); f is the signal frequency. It can be observed that, while $n_{gr} > 1$ as in (1.27), n_{ph} is in fact less than 1. This means that the affected signal will have a group delay with $v_{gr} < c$ while having a carrier phase advance due to $v_{ph} > c$. In other words, the received signal will have a positive code measurement delay and a negative carrier phase measurement delay. With the refractive indexes defined, the actual delay can be written as:

$$d_{iono,gr}(f) = \frac{40.3}{f^2} \text{TEC} \tag{1.29}$$

$$d_{iono,ph}(f) = -\frac{40.3}{f^2} \text{TEC} \tag{1.30}$$

where TEC is the Total Electron Content, is given by the integral of N_e along the propagation path:

$$TEC = \int_s N_e ds \tag{1.31}$$

The TEC value depends largely on the geographical location of the receiver, the time during the day and the solar activity. Particularly, equatorial area such as Brazil or South East Asia (SEA) witness a high level of ionospheric unpredictability [2]. The resulting ionospheric scintillation besides delay on pseudorange measurement, can even cause lost of track in tracking loops [52]. On the other hand, as mentioned above, the GNSS signals refraction depends also on the frequency, therefore it is possible to use double frequency measurement to remove the delay caused by the ionosphere [35]. For single frequency, however, there are also prediction models to mostly remove the delay effect caused by the ionosphere, such as the Klobuchar model [68] which is widely used in GPS receivers [118]. Alternatively, the NeQuick-G model is a real-time adaptation of the ITU-R NeQuick ionospheric electron density model [113, 37, 49], which has been adopted by the Galileo system for single-frequency ionospheric delay correction [4, 34]. In most cases, the ionospheric delay ranges from 3m to 30m, depending on the time of the day (which is directly related to TEC), elevation angle of the satellite and general location of the receiver.

1.3.2 Multipath

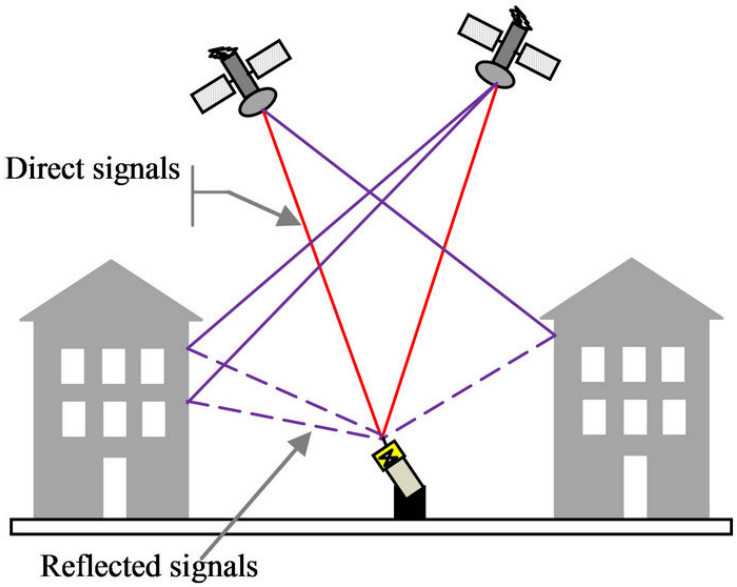


Figure 1.7: Principle of multipath interference [69]

Multipath error is a kind of delay caused by the arriving signal reflected on surfaces near the receiver's antenna. This is kind of error is prominent in environments with a lot of obstacle, such as urban areas where modern buildings with glass and metal are actually strong reflectors. Due to reflection, the path of the received signal is longer than the actual path of Line of Sight (LOS) signal, therefore inflicts significant delay in pseudorange measurement resulting in positioning error. In fact, there are two type of multipath: the Non-Line of Sight (NLOS) multipath, where only the reflection arrives at the antenna, and the multipath interference, in which both the actual signal and the reflected one arrive at the antenna.

NLOS multipath effect is caused by the blocking of the actual LOS signal by some obstacles. This effect induces a delay in the pseudorange measurement equals to the path delay, typically about tens of meters [48]. This delay is always positive due to the reflected signal travels further than the LOS one. The power of the NLOS is usually weaker than the actual signal, no more than 42 dBHz [136].

On the other hand, the multipath interference is more complex, as both the LOS and NLOS signals are mixed together (see Figure 1.7 [69]). Multipath interference causes the distortion of the code correlation peak in the tracking block of the receiver, resulting in incorrect estimation of the code phase. Depending on the tracking block design, the induced code phase error can be up to half a code chip [47, 25], corresponding to about 150 m of pseudorange error for GPS signal.

As mentioned above, multipath error potentially causes significant error to the positioning process, especially in urban environment [136]. Therefore there has been multitude of works focused on multipath detection and mitigation techniques. Notably, since the multipath effect alters the quality of the received signal, several works propose detection and mitigation method based on C/N_0 ratio [48, 59, 26]. On the other hand, there are also other approaches based on dual-polarization NLOS detecion [58] and vector tracking [51, 50].

Regarding the aspect of multipath modeling, the works in [116, 76] introduce a configurable urban channel model, called the Land Mobile Multipath Channel (LMMC) Model. This model, implemented in MATLAB, simulates the multipath effect caused by a virtual urban environment on satellite telecommunication signals, in which GNSS is a special case. The model allows the customization of such virtual urban scenario, including the width of the road and the relative position of

the receiver on that road, the size of buildings, trees and street poles on either (or both) sides of the road. With the elevation and azimuth angles of the satellite varying, the effect caused by the virtual environment also differs, based on underlying mathematical models developed from extensive data collection campaign in urban areas [116]. The model has been standardized in International Telecommunication Union (ITU) document [55].

1.3.3 Interference

Since the operation of GNSS receivers utilize the Radio Frequency (RF) signal from navigation satellites, it is susceptible to RF interference, whether it is unintentional or intentional (jamming). In literature, RF interference is usually classified as wideband (WB) or narrowband (NB) [130]. Such classification reflects the bandwidth size of the interference in relative with the bandwidth of the actual GNSS signal. A special case of NB is when the interference has a very narrow bandwidth, reducing it to a single tone, called continuous wave (CW) interference.

Studies have shown that NB interference poses the most dangerous effects on the performance of GNSS receivers [14, 11, 130], especially CW interference as reported in some real cases [12, 64]. In general, NB can impair different stages of the GNSS receivers, especially the acquisition and tracking stages, leading to incorrect tracking output or losing track completely [130]. Facing this, in literatures, besides the works on the effects of interferences [14, 12, 91], there are also many proposed techniques for detection[83, 54] and mitigation [94, 11, 112, 21, 111].

1.4 Conclusion

This Chapter has presented the overview of the GNSS receiver, including the architecture and operation, the potential error sources, and most important, the algorithms for PVT calculation. The output of the PVT evaluation process as well as its intermediate parameters will be necessary for the integrity assessment approach, as will be discuss in subsequent Chapters.

It is also important to note that, the PVT calculation process is, in fact, an approximation process, since the pseudorange input is subjected to external errors

and uncertainty, rendering the system of geometric equation 1.4 does not have an unique solution. This is one of many reasons for the necessity of integrity monitoring concept, which will be discussed in the next Chapter.

Chapter 2

Integrity monitoring

This Chapter presents about the concept of integrity, its origin and approaches in the field of civil aviation, where the concept was born. Integrity concerns about the reliability of the positioning results, which can be obtained from the PVT estimation process introduced in the previous Chapter. Such reliability is particularly important in the field of civil aviation.

This Chapter consists of three main sections. The first part introduces a general introduction to integrity monitoring and related concepts. Since integrity is a civil aviation concept, the second part of the Chapter dedicates to civil aviation terminology, as well as the navigation requirement associated with integrity monitoring. The last part of the Chapter presents existing approaches and technology for integrity assessment in civil aviation.

2.1 Integrity concepts

In the field of GNSS, integrity is defined as the capability of a positioning system to provide timely warning to users when the system becomes unreliable and should not be used for navigation purposes [133]. In other words, the purpose of the integrity assessment process concerns with whether the positioning system is reliable or not, or the level of trust an user can put into said positioning system. The concept of integrity was born since the early days of the GPS system for civil aviation. It was defined in the requirement specifications of the International Civil Aviation Organization (ICAO) to use the GNSS in the Communications, Navigation

and Surveillance / Air Traffic Management (CNS/ATM) system. The concept of integrity is necessary due to several reasons:

- The position calculation process is, in fact, an approximation process. Since there are always errors from various sources, it is impossible to calculate the exact, true position, only an approximation at certain level of accuracy. Integrity assessment methods provide a mean to monitor the positioning error to ensure the reliability of the results.
- There is a possibility of unexpected system-side faults, such as satellite clock error, ephemeris error, which can cause excessive positioning errors. These kind of error can affect a limited number of satellites, or most unlikely, a whole satellite constellation. While traditional integrity monitoring methods allow users to detect satellites error, recent algorithms also take into account constellation-wide faults.
- Applications such as aviation involve the safety of human lives. Therefore integrity monitoring is of crucial importance to ensure the reliability and safety of the flight, especially in certain phases of flight such as landing or taking off, where the reliability of the positioning systems is a major tool for pilots to make decisions.

To realize the integrity assessment process, some concepts have been defined:

- **Protection Level (PL)**: the statistical bound surrounding the calculated position, *ensuring* to cover the true position at a certain level of confidence. PL is defined in horizontal and vertical direction, namely HPL and Vertical Protection Level (VPL), respectively. The probability of Positioning Error (PE) exceeding the PL is less than or equal the required *integrity risk*. The PL can be calculated using the measurements from the user's receiver following the Aircraft-based Augmentation System (ABAS) approach, or from external augmentation systems such as Satellite-based Augmentation System (SBAS) or Ground-based Augmentation System (GBAS).
- **Alarm Limit (AL)**: the bound around the true position, which describes the region that is *required* to contain the calculated horizontal position. The event

that the actual (absolute) PE exceeds AL is called a situation of Hazardous Misleading Information (HMI), and the probability of such event is called the Probability of Hazardous Misleading Information (PHMI). If the PE is detected to have exceeded the AL, the system should be flagged as *unavailable* for use. Similar to PL, AL is also defined in horizontal and vertical direction, called Horizontal Alarm Limit (HAL) and Vertical Alarm Limit (VAL), respectively. AL values are usually defined based on the requirement of the intended application, subsequently dictates the availability of the positioning system for such application.

- **Integrity risk**: the maximum allowed probability of integrity failure, which is an undetected failure that leads to a HMI event. In other words, the required integrity risk is the upper limit for the PHMI, and the probability that PE exceeds PL without detection.

- **Time to Alert (TTA)**: the maximum allowable period from the occurrence of the fault to the alert being given by the navigation device. This value is also application-dependent

The concept of integrity sometimes can be mistaken for accuracy, as both concepts involve the statistical distribution of the PE, especially in the field of civil aviation. In fact, integrity and accuracy are two out of the four parameters used to evaluate the performance of a GNSS system defined by the Required Navigation Performance (RNP) [**icao1995special**] - the other parameters are availability and continuity. It is important to note that, accuracy not only measures the conformity of the calculated position of the receiver to the true position, but also associates the level of certainty of such assessment. The concepts of integrity and accuracy have several distinctive differences:

- The most important difference is the inclusion of alarm. While accuracy mainly concerns with the correctness of the positioning solution and does not require an alarm, integrity assesses the consistency and reliability of such solution and thus raises alarms when the bad performance of the system may lead to risky situation.

- Mathematically, the two concepts utilize different percentile for respective requirement. In aviation, the accuracy specifications require the accuracy level measured at the 95% percentile, while the integrity requirement has a percentile above 99% (leaving the integrity risk ranges between 10^{-7} to 10^{-4} depending on the operation).

2.2 Civil aviation concepts

Since the integrity concepts, most existing integrity monitoring methods and their inherent performance specification are defined for the use in civil aviation. Therefore, it is necessary to introduce briefly about the terminology and the underlying concepts in order to understand correctly about the subsequently introduced integrity monitoring algorithms. Most of the definitions in this section are taken from the Pilot/Controller Glossary issued by the Federal Aviation Administration (FAA) [104].

2.2.1 Phases of flight

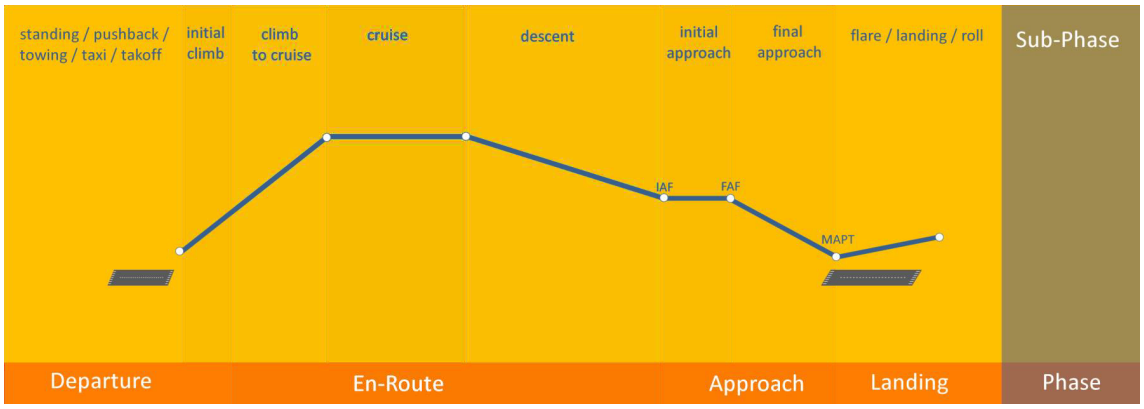


Figure 2.1: Typical phases of flight [87]

In civil aviation, a flight consists of multiple phases when an aircraft travels from its departure to the destination. A comprehensive graph of the phases of flight can be found in Figure 2.1. In particular, the main phases include:

- **Departure**: the transition from take-off (end of runway, preceded by other ground-based operations such as standing, taxi,...) to the first cruise altitude.

- **En-route**: the second main phase, when the aircraft travels from the departure to the destination airport.
- **Approach**: the phase when the aircraft attempts to land, containing two subphases. The initial approach is from the Initial Approach Fix (IAF) to Final Approach Fix (FAF), and the final approach is from FAF to the beginning of the landing flare - the transition of the aircraft just before touchdown. On the other hand, if a landing cannot be made safely (due to onboard instrument errors, hindering weather,...), a missed approach must be performed at the Missed Approach Point (MAPT), where the pilot aborts the landing operation and climbs upward, following by other instruction by the Air Traffic Control (ATC).

The operation of a pilot during approach phase is defined in the Instrument Approach Procedure (IAP), which is a series of predetermined maneuvers for the orderly transfer of an aircraft from the beginning of the IAF to a landing or to a point from which a landing may be made visually.

There are three category of IAP:

- Non Precision Approach (NPA): approach procedures using systems that only provide lateral guidance and no vertical guidance. In other words, the pilot only knows the heading of the aircraft, does not receive altitude and glidepath guidance.
- Approach with Vertical guidance (APV): procedures using systems that provide both lateral and vertical guidance. APV using GNSS vertical guidance has 2 performance levels: APV-I and APV-II.
- Precision Approach (PA): standard IAP in which an electronic glideslope/glidepath is provided. PA systems provide both lateral and vertical guidance at 3 performance levels: CAT-I, CAT-II and CAT-III.

On the other hand, Localizer Performance with Vertical guidance (LPV) is another form of approaches that specifically relies on GPS positioning and integrity monitoring service provided by a SBAS in the US called Wide-Area Augmentation System (WAAS) (see Section 2.3.2). While LPV is considered APV by definition, it offers performance level equivalent to PA CAT-I [32].

2.2.2 Navigation system requirements

The requirements for navigation systems in civil aviation are defined by ICAO with 4 aforementioned parameters: accuracy (95%), integrity, availability and continuity. While the concepts of accuracy and integrity have been discussed above, the other 2 concepts are also related to integrity. The availability of a navigation system defined as the ability of that system to provide the required function and performance at the initiation of the intended operation. It is characterized by the percentage of time the system is to be used for navigation during which reliable navigation information is presented to the users [7]. From the practical point of view, a system is declared available whenever it is able to provide a navigation output with accuracy and integrity at the level required for a specific operation. On the other hand, the continuity of a system is the ability of the total system to perform its function without unscheduled interruption during the intended operation [7].

The integrity requirements for aviation operations are defined in Table 2.1 [46]. Note that while the integrity risk for NPA is defined on per hour basis, that for other approaches are defined per approach, which is equivalent to 150 seconds. It is easy to observe that the integrity risk requirement for NPA is significantly less constrained with respect to other vertical guidance approaches. On the other hand, since NPA provides lateral guidance only, VAL is not defined for it. On the other hand, the required level of integrity performance for the approach are quite different, especially for the VAL. While the integrity risk level is almost identical for APV and PA approaches, the VAL for APV-II is more stringent than APV-I, while CAT-I and LPV 200 have almost identical requirements.

The requirements for accuracy, including Horizontal Accuracy (HA) and Vertical Accuracy (VA), continuity and availability are detailed in Table 2.2 [46]. Similar to the integrity requirements, NPA does not have a VA requirement, plus the requirements are notably more relaxed than other approaches (except for availability). LPV 200 and CAT-I have almost the same requirements, and in this case, is more strict than APV approaches.

Table 2.1: Integrity requirement for aviation approaches [46]

Approach	Integrity risk	Time-to-alert	HAL	VAL
NPA	1×10^{-7}/h	10 s	556 m	N/A
APV-I	2×10^{-7}/approach	10 s	40 m	50 m
APV-II	2×10^{-7}/approach	6 s	40 m	20 m
PA CAT-I	1×10^{-7}/approach	6 s	40 m	35 m to 10 m
LPV 200	2×10^{-7}/approach	6.2 s	40 m	35 m

Table 2.2: Integrity requirement for aviation approaches [46]

Approach	HA (95%)	VA (95%)	Continuity risk	Availability risk
NPA	220 m	N/A	10^{-8}/h to 10^{-4}/h	10^{-5} to 10^{-2}
APV-I	16 m	20 m	8×10^{-6}/15 s	10^{-5} to 10^{-2}
APV-II	16 m	8 m	8×10^{-6}/15 s	10^{-5} to 10^{-2}
PA CAT-I	16 m	6 m to 4 m	8×10^{-6}/15 s	10^{-5} to 10^{-2}
LPV 200	16 m	4 m	8×10^{-6}/15 s	10^{-5} to 10^{-2}

2.3 Integrity monitoring methods

Integrity monitoring methods are algorithms designed to protect users, and ensure that integrity is maintained throughout the operation. There are three main approaches: SBAS, GBAS and ABAS.

2.3.1 Ground-based Augmentation System

GBAS, defined by ICAO, is an augmentation system in which the aircraft receives augmentation information directly from a ground-based transmitter. The augmentation data provided by GBAS includes differential correction to improve positioning solution, and data related to integrity monitoring service. GBAS is often installed in airports to assist aircrafts when taking off or landing, and is capable to provide its service to an area up to 23 nautical miles (approx. 42 km) radius around the host airport. GBAS broadcast the augmentation data in the VHF frequency band (108 - 118 MHz)

For the correction service, GBAS adopts a positioning techniques known as

Local Area Differential GPS (LADGPS) [100]. In principle, GBAS (or any DGPS system in general) utilizes a monitoring and control station on the ground (called *reference station* in DGPS terminology) to monitor the navigation signal quality. Since the exact location of the station is known, it can precisely measure the ranging errors and transmit them to users. With the provided correction from the ground monitor station, the receiver on board (called *rover* in DGPS terminology) can mitigate potential positioning error, thus improve overall accuracy. Following this approach, GBAS can correct the errors that are common to both the ground station and the on board receiver, which includes: ionospheric error, tropospheric error, satellite clock error and ephemeris error. Other errors that are local to each receiver cannot be mitigated by GBAS: multipath effect, interference and receiver noise.

For the integrity monitoring service, GBAS also monitors the quality of the navigation signal with the ground station, and provide a real-time indication that the signal integrity is ensure or not. GBAS also calculates the PL on the ground and sends to the user, under the assumptions that:

- The receiver on board is fault-free.
- The pseudoranges of the receiver on board after correction using GBAS data are affected by noise only, i.e. other errors have already been mitigated.
- One of the reference receiver may be faulted.
- Both ground station and on board receiver use the same satellite constellation.

Currently, GBAS can offer service for APV level I and II [45], PA CAT I [8]. GBAS has also been proposed to cover PA CAT II/III as well [45]. On the other hand, a map of existing airports around the world currently supporting or testing GBAS is depicted in Figure 2.2.

2.3.2 Satellite-based Augmentation System

SBAS is defined as a wide coverage augmentation system, in which the user receives augmentation information from a satellite-based transmitter. SBAS operation is based on the technique of the Wide Area Differential GPS (WADGPS) [67]. While the LADGPS and GBAS system can only provide correction that are

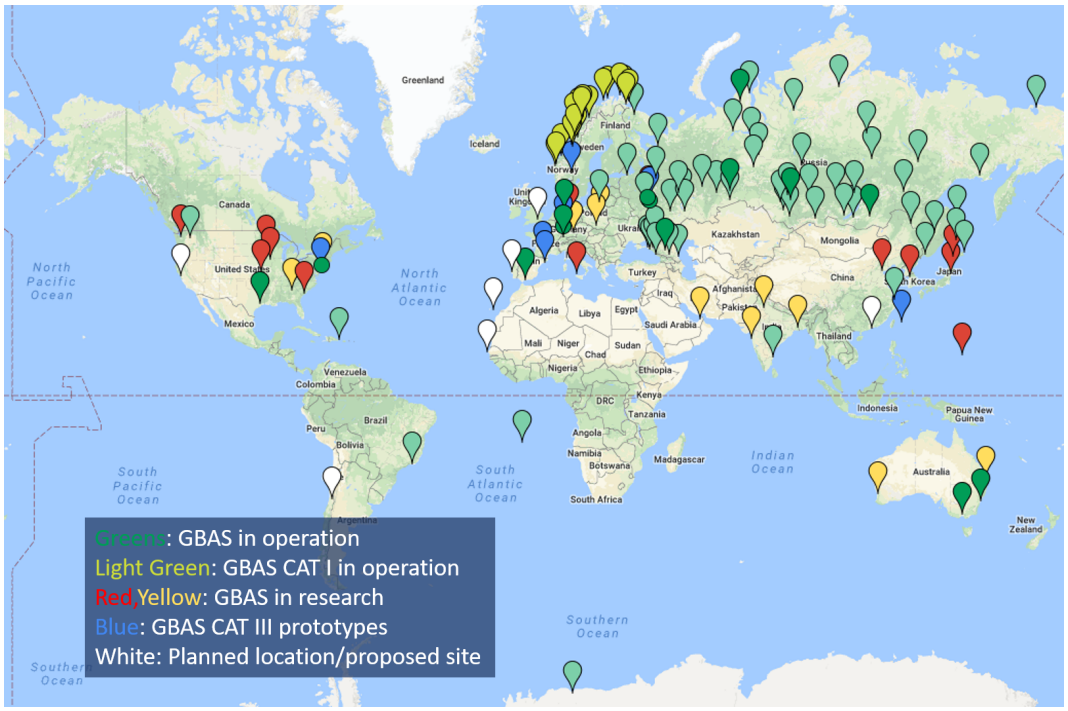


Figure 2.2: GBAS facilities world wide

common to both the ground station and airborne receiver and limited to an area near the airport, WADGPS can generates and broadcast correction for users over a large coverage area, even over a continent [66]. WADGPS corrections are generated by a network of monitoring stations operated in the coverage area. In fact, SBAS and GBAS have several distinctive differences:

- The SBAS broadcast frequency band can be identical to the GPS signals, while GBAS uses VHF band.
- SBAS utilizes geostationary satellites covering certain area of service, allow the system to provide service over a larger area then GBAS. In fact, the satellites of SBAS can also provide ranging measurement similar to GNSS satellites, thus improves geometry and redundancy for the positioning solution [7].
- SBAS generates correction data using a network of monitoring station in the area of service, instead of a few stations installed in an airport as in the case of GBAS.

- SBAS provides vector corrections, i.e. individual correction for each type of error (clock, ionosphere, ephemeris) while GBAS wraps all the error into scalar pseudorange corrections.

For the integrity monitoring service, SBAS provides the health status of all the satellites in the monitored constellation [7]. The satellite signals can be declared "Not monitored", if not visible from the ground stations, or "Do not use" in case of integrity failure. On the other hand, SBAS also provides a set of parameters for reliability assessment and PL calculation on the user's side [88]. The parameters include:

- User Differential Range Error (UDRE): estimated bound of the pseudorange residual error after applying the ephemeris/clock correction for each satellite.
- Grid Iono Vertical Error (GIVE): estimated bound of the pseudorange residual error after applying the ionospheric correction for each satellite.

At each epoch, the PL are computed by the user's receiver using the parameters provided by SBAS, airborne parameters, user's geometry following the SBAS user algorithm [88]. If the calculated PL exceeds the corresponding AL, the SBAS service is declared unavailable for the intended opeartion.

Currently, there are several operational SBAS systems in the world (WAAS, EGNOS, MSAS and GAGAN), others are under implementation or research phase. WAAS was developed by the US Department of Transportation and the FAA. The development began in 1994 and the system is activated for general aviation in 2003. WAAS now covers largely the North America area, including the US (up to Alaska), Canada and Mexico. WAAS supports en-route, terminal and approach operations down to PA CAT I [46].

On the other hand, the Multi-functional Satellite Augmentation System (MSAS) provides SBAS service in Japan area with 2 geostationary satellites and a small network of monitoring stations and control centres. The system is operational since 2007, and provide guidance service for operations from en-route to NPA. In India, the GPS-aided GEO Augmented Navigation (GAGAN) system was developed to improve the accuracy of GNSS receivers as well as modernize the CNS/ATM system. GPS-aided GEO Augmented Navigation (GAGAN) has 15 reference stations, 3

control centers, 3 satellites (1 of which is geostationary). GAGAN has been certified for precision approach service APV-I in 2015.

In Europe, the European Geostationary Navigation Overlay Service (EGNOS) is the latest operational system, developed by the European Space Agency (ESA) and EUROCONTROL. The system covers the whole European area using 3 GEO satellites and a network of 40 ground stations. European Geostationary Navigation Overlay Service (EGNOS) provides 3 basic services:

- Open Service: available to any EGNOS-compatible GNSS receivers, operational since 1/10/2009.
- EGNOS Data Access Service (EDAS): provide services for professional users also via ground transmission systems. The data provided includes EGNOS augmentation message (as also transmitted by EGNOS satllites) and GPS raw data collected by EGNOS monitoring stations.
- SOL service: mainly for safety-critical application, such as civil aviation. This service is provided with enhanced and guaranteed performance, and includes also integrity monitoring features. The service is available since March 2012.

2.3.3 Aircraft-based Augmentation System

While GBAS and SBAS provide integrity monitoring at the system level, the ABAS approach aims at providing integrity monitoring at sensors/receiver level. Distinctively, ABAS utilizes a set of integrity assessment algorithms that can operate autonomously, rely solely on the measurement data from the on board receiver. Therefore, ABAS allows integrity monitoring service to be available even outside of SBAS/GBAS coverage, or can be considered a redundancy in presence of other augmentation systems. ABAS also allow detection of highly localized error or receiver side error, that might not be detected by ground stations of external augmentation system.

ABAS integrity monitoring service relies on a set of algorithms called Receiver Autonomous Integrity Monitoring (RAIM) [10]. In principle, the integrity assessment process using ABAS algorithms usually consists of two main steps. The first step is called Fault Detection and Exclusion (FDE), in which ABAS exploits the

redundancy in the observed measurements and performs consistency test to check for faults in the input data, then the algorithm may attempt to exclude the faulty input if a fault is detected. If the input is deemed consistent, ABAS proceeds to the next step, which is to calculate the PL. In short, the goals of the integrity assessment process are to protect users from excessive positioning errors by detecting and mitigating (if possible) the faults, and alert users in the worst case. Traditionally, ABAS algorithms operate under several assumptions [10]:

- Only one constellation is used (although it's possible to extend the algorithm to multi-constellation),
- There is at most one satellite fault at a time,
- The PL depends mostly on the satellite geometry.
- The measured pseudoranges may be affected by errors from multiple sources (atmospheric error, satellite clock error, multipath, etc). The error components are independent, and the resulting pseudorange error is zero-mean and Normal-distributed

There are two main approaches for ABAS methods: residual-based approach [10], which operates in pseudorange domain, and solution separation approach [29], which runs in position domain. While residual-based ABAS is simple and fast, solution separation ABAS can have better performance due to its customization [61]

Range domain

In literature three main FDE algorithms can be categorized in the range domain [65, 28, 10]: range comparison, least-squares residuals and parity method. All the methods are able to determine the presence of a satellite failure by using the redundancy of the measurements of the over-determined system of linearized equations (1.11). FDE algorithms check for the consistency using the so-called Global Test [71] (also known as range-residual test) that requires at least five satellites in view. The process of a FDE algorithm comes from the statistical detection theory [99] and consists in verifying if a quantity (identified as *test statistic*) exceeds a given

threshold. The threshold is chosen so that, in fault-free conditions, the *test statistic* has a very low probability to pass it, thus ensuring very low false positive alarms. Under faulty conditions, however, the values achieved by the *test statistic* is no longer distributed as in faultless conditions and therefore more likely exceeds the threshold thus allowing for the recognition of the fault.

Using the set of linearized measurement equations (1.11), the Global Test can be defined at each epoch k as:

$$\text{Global Test} = \begin{cases} H_0 : E\{\boldsymbol{\epsilon}_k\} = \mathbf{0}, Var\{\boldsymbol{\epsilon}_k\} = \boldsymbol{\Sigma}_k \text{ if } \tau_{global} < T_{global} \\ H_1 : E\{\boldsymbol{\epsilon}_k\} \neq \mathbf{0}, Var\{\boldsymbol{\epsilon}_k\} \neq \boldsymbol{\Sigma}_k \text{ if } \tau_{global} > T_{global} \end{cases} \tag{2.1}$$

where H_0 is the fault-free hypothesis; H_1 is the fault-present hypothesis; $\boldsymbol{\Sigma}_k$ is the covariance matrix of the pseudorange errors $\boldsymbol{\epsilon}_k$, which are assumed to be normally distributed; τ_{global} and T_{global} are the *test statistic* and detection threshold, respectively. In the residual method, τ_{global} is given by:

$$\tau_{global} = \hat{\boldsymbol{r}}_k^T \boldsymbol{\Sigma}_k^{-1} \hat{\boldsymbol{r}}_k \tag{2.2}$$

where $\hat{\mathbf{r}}_k = \bar{\boldsymbol{H}}_k \boldsymbol{\Delta}\hat{\boldsymbol{u}}_k - \boldsymbol{\Delta}\boldsymbol{\rho}_k$ is the residual between predicted and measured pseudoranges [35]. For Weighted Least Square (WLS) solution, $\hat{\mathbf{r}}$ is given by:

$$\hat{\mathbf{r}}_k = \bar{\boldsymbol{H}}_k \left(\bar{\boldsymbol{H}}_k^T \boldsymbol{\Sigma}_k^{-1} \bar{\boldsymbol{H}}_k \right)^{-1} \bar{\boldsymbol{H}}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Delta}\boldsymbol{\rho}_k - \boldsymbol{\Delta}\boldsymbol{\rho}_k = -\boldsymbol{R}_k \boldsymbol{\Delta}\boldsymbol{\rho}_k \tag{2.3}$$

In faultless conditions the errors in $\boldsymbol{\epsilon}_k$ are normally distributed, thus τ_{global} (which is a sum-square of the errors) follows a central Chi-square distribution with $N_{sat} - p$ degree of freedom (DOF), where p is the number of parameters to be estimated, in the case of (1.11) is 4.

The threshold T_{global}, against which the *test statistic* must be compared, is obtained arbitrarily fixing the probability of false alarm P_{fa} that is perceived acceptable for the faulty free condition. It results that

$$P_{fa} = \int_{T_{global}}^{\infty} f_{\chi^2}(x; N_{sat} - p) dx \tag{2.4}$$

where $f_{\chi^2}(x; K)$ is the central Chi-square probability density function (pdf) with K DOF.

T_{global} can then be derived as [10]:

$$T_{global} = Q_{\chi^2}(1 - P_{fa}, N_{sat} - p) \tag{2.5}$$

where $Q_{\chi^2}(P, K)$ is the quantile function of the probability P of central Chi-square distribution with K DOF, while P_{fa} is the probability of false alarm.

Note that τ_{global} and T_{global} depend on the number N_{sat} of available satellites, therefore in general they are function of the epoch k. This dependence has been however omitted in the notation for sake of simplicity, also because no change usually occurs on short time horizon of few minutes.

In the parity method [117], the measurement vector $\boldsymbol{\Delta\rho}_k$ is transformed into the state estimation space (with p dimension) and an orthogonal space called the Parity space (with $N_{sat} - p$ dimension). The parity vector $\boldsymbol{p}_k$ is obtained using QR factorization [65]. From the mathematical point of view, while it is possible to use both $\hat{\boldsymbol{r}}_k$ and $\boldsymbol{p}_k$ to calculate τ_{global}, $\hat{\boldsymbol{r}}_k$ in fact contains p constraint corresponding to the unknowns of (1.11), which may obscure some inconsistency of interest. As a result, using $\boldsymbol{p}_k$ is considered easier to develop fault detection method.

Position domain

The second approach of RAIM consistency test is performed in the position domain, called Solution Separation (SS) approach [29, 138]. The motivation of the approach is that, if the whole set of pseudorange measurements is consistent, then the position calculated using the whole set (called the *all-in-view* position) and the those computed using subsets of measurements should be close together, as exampled on Figure 2.3. The subsets of satellites is called *fault mode*, corresponding to a fault hypothesis (i.e. a potential faulty was excluded from that subset), while the position calculated using such subset is called *fault-tolerant* position. Intuitively, excessive separation of fault-tolerant positions indicate potential satellite faults as well as candidates for exclusion attempt.

Following this, the approach adopts multiple seperate binary tests to perform

on each subset (called SS tests). The test statistics for each test is simply the separation between the all-in-view position $\boldsymbol{u}_k^{(0)}$ and the subset position $\boldsymbol{u}_k^{(q)}$ calculated by the q^{th} subset. The detection threshold is derived from the covariance of the separation and the desired probability of false alarm P_{fa} [138].

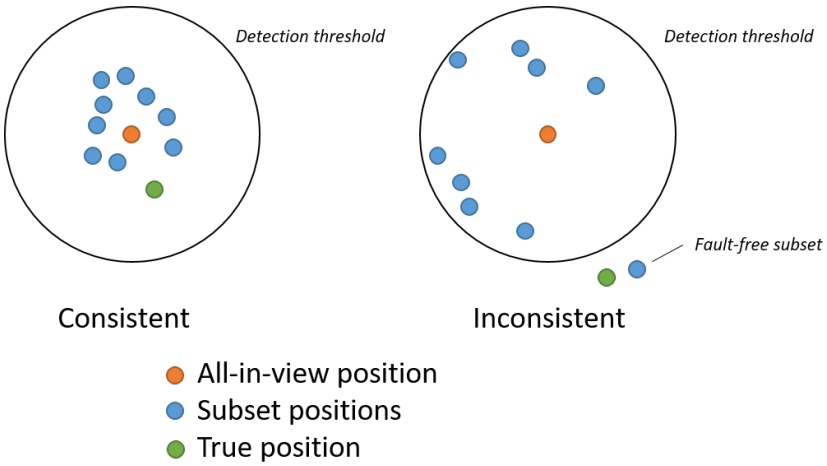


Figure 2.3: Solution separation consistency test

Recent methods

Comparatively, residual-based RAIM is simple to implement and fast, since there is only one Global Test required for consistency test (futher Local Test may be needed for exclusion), the SS RAIM in position domain can have better performance due to its customization [61]. In fact, the simplicity of residual-based RAIM sees it becomes the baseline algorithm for numerous works, such as the Weighted Least Square Residual (WLSR) RAIM with adaptive P_{fa} [3] to ensure availability at all times. The work in [137] combines both approach into a hybrid FDE method, in which SS tests are adopted for fault detection, and residual-based tests are used for fault exclusion.

Recent works have proposed many optimization in different aspects to improve the performance of traditional RAIM algorithms. The works in [27, 5] allow detection of more than one failure, given enough measurements, thus effectively improve the single-fault assumption of traditional RAIM. Moreover, a variant of residual-based RAIM called Novel Integrity-Optimized RAIM (NIORAIM) [53] proposes

the use of a suboptimal weighting in the LS process to improve the integrity performance and availability of the system. The weighting for the SS approach is also optimized by the Optimally Weighted Average Solution (OWAS) [75]. The Multiple Hypothesis Solution Separation RAIM (MHSS RAIM) [17, 19, 74] proposes the allocation of the integrity risk optimally between different types of system fault. From another approach, the work in [96] suggests the computation of PHMI directly from the input measurement.

On the other hand, ABAS algorithms for KF-based positioning method have also been recently proposed. Most methods follow residual-based approach, using the innovation vector [132, 140] or the ranging residual vector [62, 15] for the consistency test, as well as corresponding methods to calculate the PL.

In the field of civil aviation, recent works in navigation integrity [134, 103, 20, 19, 74] have led to the next generation of ABAS algorithm, called Advanced Receiver Autonomous Integrity Monitoring (ARAIM) [134, 18], based on the SS approach (for LS). ARAIM improves the traditional ABAS in various ways, such as multi-constellation capability, generalized satellite fault hypothesis instead of single-fault assumption. ARAIM also adopts a noise model developed from extensive data collection campaign tailored for aviation use [84], allowing optimal weighing for weighted LS computation in such environment. ARAIM aims at providing better availability, lower PL [125, 92, 85], making it suitable for more stringent phases of flight which are usually protected by SBAS and GBAS [18]. The overall scheme of the ARAIM algorithm is depicted in Figure 2.4. Interested readers can find detailed description and derivation of ARAIM in [134, 18].

2.4 Conclusion

Integrity monitoring measures the level of trust an user can put into the navigation solution evaluated from a receiver. This issue is of particular importance in the field of civil aviation, where such reliability relates to the lives of the passengers. Integrity requirements are defined for the approach phase, where the knowledge of vertical position of the plane is very important to the decision of the pilot. This Chapter has introduced the concept of integrity, its necessity and requirement in the context of civil aviation, as well as the main approaches for integrity monitoring,

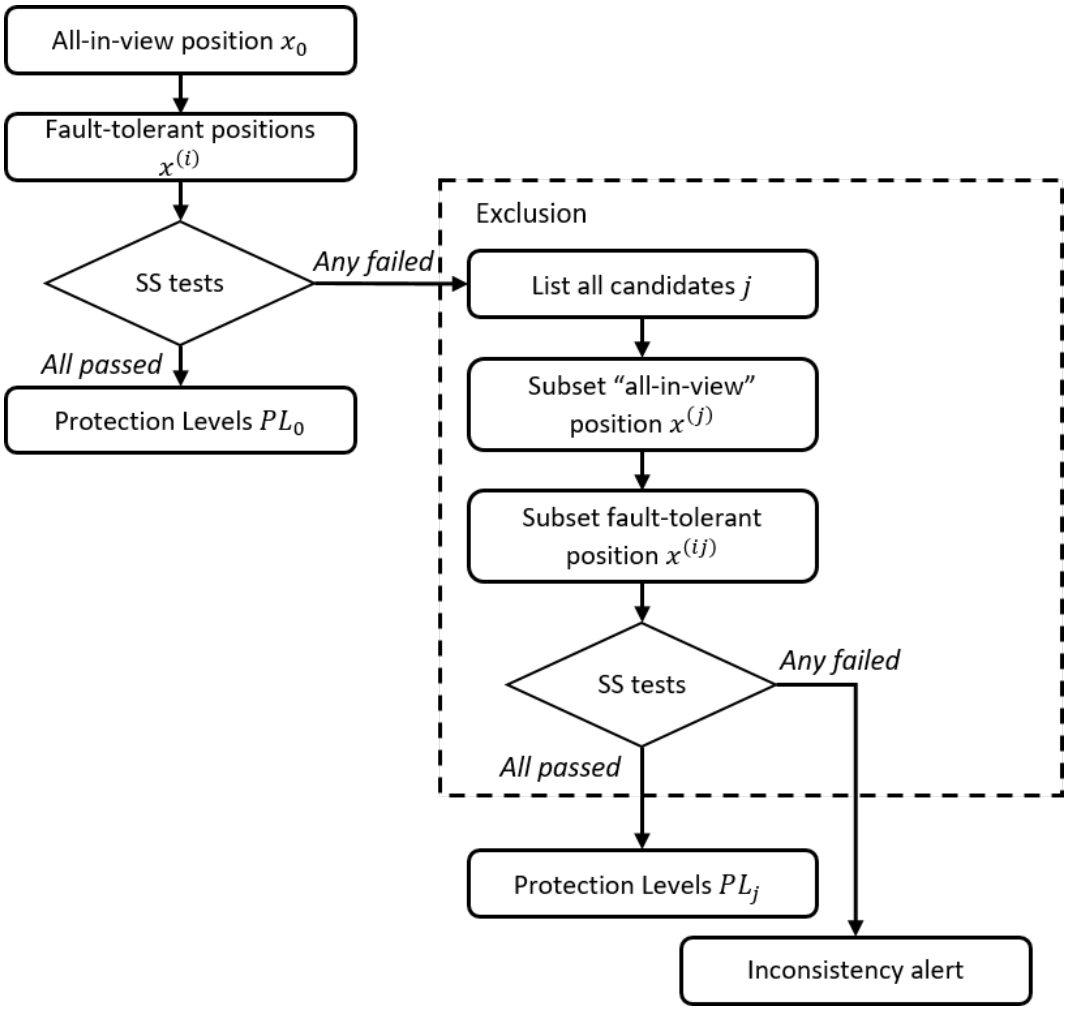


Figure 2.4: Scheme of the ARAIM algorithm

in particular the ABAS approach which can be implemented on receiver side.

Amidst the presented methods and approaches for integrity monitoring, this thesis aims at designing a new method for integrity assessment in urban scenario. The new method is based on the most recent ARAIM algorithm, which follows the SS approach, but will be adapted for KF instead of the original LS. Furthermore, the new method will operate in a different environment (urban scenario), which will require necessary changes to fit the requirement of the new environment. More detailed discussion and explanation in term of motivation, survey as well as the development of the proposed algorithm will be presented in subsequent Chapters.

Chapter 3

Adapting ARAIM algorithm for urban environment

With increasing interest in the concept of integrity in applications other than the original civil aviation use, it is necessary to study the required change to the integrity assessment algorithms to suit the characteristics of the new environment. It is crucial to note that, the environment in urban scenario is vastly different from the airports, where the RAIM/ARAIM algorithms were originally designed for. This Chapter discusses the motivation for adapting the ARAIM algorithm to urban environment, then investigates the necessary changes to the ARAIM algorithm, most importantly the noise model for modeling the pseudorange errors. Various options for the noise model are also studied and compared using simulated data, in order to choose a suitable model. A part of the content in this Chapter has been published in [124].

3.1 Recent works on integrity monitoring in urban environment

Although initially developed for the field of aviation, the concept of integrity has gain vast interest in other applications as well, especially those that require high level of reliability for the navigation data, such as railways [79, 97, 56], or ITS [16], where reliability is very important to ensure the operational of the applications

[36, 121]. An example is the Liability Critical Applications [13] in road traffic management, where the computed PVT are the used as input for subsequent legal decisions or economic transactions, such as in a automatic road tolling collection system. Consequently, any positioning error exceeding a certain threshold can lead to a misjudged legal decision or miscalculated transaction, which undoubtedly also lead to legal issues. This raises the problem of keeping the positioning error below a certain level, or rather, monitor and bound the error with a very high level of confidence and alert users in case of breach. This problem is, in fact, close to what the concept of integrity was developed for [36].

This leads to the interests of adapting the integrity monitoring concept into urban scenario. Among the three approaches mentioned in Section 2.3, GBAS can be considered the most limited approach. As mentioned earlier, GBAS requires the installation of ground stations in the area of operation, which can be costly. On the other hand, GBAS reference stations have to be located in areas that is not affected by multipath or other interferences, which is the opposite characteristics of urban areas. Therefore GBAS is not the ideal choice for integrity monitoring in urban environment.

On the other hand, [110, 122] propose to use SBAS to calculate the PL for urban applications. However, currently the SBAS coverage around the world is quite limited, so it is not possible to apply the SBAS approach to the areas without coverage. Also, it is important to note that, GBAS and SBAS aims at providing integrity monitoring service on a system perspective, while users in urban areas can experience other errors that SBAS can not detect, such as multipath, interferences or receiver side errors.

This analysis leaves the remaining option to be ABAS, which is also the focus of a multitude of recent works. The works of [3, 16] attempted to adapt traditional RAIM algorithm for urban use, but the assumption of the traditional RAIM approach is not suitable for the urban scenario [72, 43]. Recall from Section 2.3.3 that the RAIM algorithm assumes single-fault scenario, and the PL depends mostly on the geometry of the satellites. In fact, in urban environment the navigation signals are objected heavily by multipath and other interferences, which affect all satellites indiscriminately. This leads to the possibility of excessive error on more than one satellites. Multipath in urban area is also a prominent cause for positioning error

[48] and thus should be considered in calculating the PL. [139] proposed an adaptive, residual-based RAIM for urban scenario, but assumed all the pseudorange errors to have similar error variance, which is not quite realistic in such context. The study in [124], on the other hand, suggested several necessary points to change to adapt ARAIM algorithm to urban context, such as the noise model and integrity risk allocation.

3.2 List of changes

3.2.1 Limitation of aviation-based integrity for urban environment

It is important to note that, the original integrity concept and its extensive array of integrity assessment techniques are systematically developed for aeronautical usage, tailored for its characteristic environment and requirement. As a result, adapting integrity monitoring methods, particularly ARAIM algorithm, from aviation to urban context is not a trivial task. There are several recent studies [82, 36, 124] mentioning the limitation of aviation-based integrity monitoring methods (ARAIM algorithm in particular) for urban use, notably two main points: strict requirement, and the noise model for the pseudorange error variances.

The first point is the strict requirement of the algorithm. As introduced in the previous Chapter, the ARAIM was designed to satisfy the stringent requirement of the LPV-200 procedure (see Section 2.2.2 for more details). While a similar set of standard for vehicular applications is not available yet, such requirement in aviation context is still considered too strict for urban use [82, 106]. In this case, the requirement includes the integrity and continuity risk, the AL (especially HAL, considering urban applications usually need to know the location of vehicles on a planar map), which need to reconsider to suit the urban use. On the other hand, in order to meet the aforementioned strict requirement of LPV-200 procedure, the ARAIM algorithm also incorporate the use of nominal bias [18], taking into account the small errors that remain constant in a short period of time [81]. This may not be necessary for urban applications.

The second point is the noise condition between the two environment and associated noise model. The most prominent environmental difference between aviation context and urban environment is the multipath condition. In aviation, the large part of the multipath comes from reflection on the ground and on the body of the aircraft itself [84]. However, the multipath condition in urban areas is much more complex, due to reflection on buildings, trees, other infrastructures, as pointed out in Section 1.3.2. This results in larger amplitude of error from multipath, as well as more unpredictable [136, 48, 36]. Therefore, the existing error model currently adopted in ARAIM algorithm, which estimates the error based on satellites' elevation angles, is not sufficient to model such kind of multipath error.

The brief analysis above leads to the following changes in the ARAIM algorithm to adapt it to urban environment:

- The values of constants associated with the requirement
- The nominal biases
- The noise model

More in-depth analysis and solutions for each point above will be presented in subsequent Sections of this Chapter.

3.2.2 Values of constants

ARAIM algorithm [18] utilizes a set of constants to describe the integrity, continuity budget and allocation, as well as several thresholds for the SS tests. The Probability of Hazardous Misleading Information P_{HMI} values represent the integrity budget, or the risk of integrity violation during operation. The false alarm P_{FA} values represent the continuity budget, or the risk of discontinuity due to false alarm. The values for integrity and continuity budget in [18] are from [6], standardized for aviation, and allocated to horizontal and vertical directions. The constraints for these parameters are:

$$P_{HMI,VERT} = P_{HMI} - P_{HMI,HOR} > 0 \tag{3.1}$$

$$P_{FA,VERT} = P_{FA} - P_{FA,HOR} > 0 \tag{3.2}$$

where $P_{HMI,HOR}$ and $P_{HMI,VERT}$ are the horizontal and vertical allocation of P_{HMI}, respectively. Likewise, P_{FA} is allocated to $P_{FA,HOR}$ and $P_{FA,VERT}$ for horizontal and vertical directions.

From the mathematical point of view, too tight integrity and continuity budget will increase the resulting PL, thus reduce the availability of the positioning system and vice versa. In fact, the values for those constants are usually based on the requirement of the intended application. For example, in the case of LPV 200 procedure (see Section 2.2), P_{HMI} is 10^{-7}, allocating much of the budget to the vertical component, thus $P_{HMI,VERT}$ is 9.8×10^{-8} while $P_{HMI,HOR}$ is just 2×10^{-9}. These values are very tight, considering the intended use of ARAIM algorithm for integrity monitoring when the plane is about to land, which undoubtedly require very high reliability on the vertical positioning. On the other hand, [43] suggests some reference values for Road tolling and Emergency support of urban road, which are less tight than the aviation counterpart. The values are reported on Table 3.1, which contains also the aviation constants for comparison purpose.

Table 3.1: Integrity requirement for different applications

Service	Horizontal Alarm Limit (m)	Time to Alert (s)	P_{HMI}
Road tolling	$10 - 50$	$6 - 10$	10^{-4}
Emergency support	$5 - 10$	$1 - 2$	10^{-5}
LPV-200 (aviation)	40	6.2	10^{-7}

In term of allocation, for road traffic applications, the accuracy on the horizontal direction is more important than on the vertical direction. Therefore the budget for the horizontal component should be higher. This applies for both the integrity risk and the continuity risk. An example of risk budget allocation is proposed in Table 3.2. Note that the integrity risk budget in this case refers to the Road tolling application in Table 3.1.

3.2.3 Nominal biases

Recall that the ARAIM algorithm aims at satisfying the strict requirement of LPV-200 approach [18], which has smaller limits than NPA. This leads to the consideration of small range errors (due to multiple sources) that remain constant

Table 3.2: Interity and continuity budget allocation

Constant	Value
P_{HMI}	10^{-4}
$P_{HMI,HOR}$	9.8×10^{-5}
$P_{HMI,VERT}$	2×10^{-6}
P_{FA}	4×10^{-6}
$P_{FA,HOR}$	3.9×10^{-6}
$P_{FA,VERT}$	10^{-7}

in a period of time. These errors are modeled as nominal biases [18, 103], i.e. biases when the receiver operates in a fault-free condition [80]. ARAIM utilizes a model for nominal biases, which is developed based on smoothed dual frequency (L1-L5) multi-constellation (GPS/Galileo) signal. The sources of errors for nominal biases are [81]:

Nominal signal deformation: perturbation affecting the signal waveform transmitted by the satellite as observed by the receiver (affecting the correlation between the received signal and the local signal). These deformations affect signals in each band independently and of course will affect the signal combination as well.

User antenna group delay variation: delay according to the angle of arrival with respect to the center of measurement. Analysis in [81] shows that the antenna group delay depends on the user antenna, the azimuth and elevation angle of the incoming signal, the electromagnetic fields radiated by various part of the aircraft itself.

Satellite antenna group delay variation: bias depending on the nadir angle, which is not fully reflected in the User Range Error (URE) value broadcasted from the satellite.

The working condition in urban context is different from the condition listed above when considering the nominal biases:

- The occasional urban canyons and trees can cause cycle slips or losing track ofsatellites, therefore carrier phase positioning may not always be available.
- The antennas for mass market receivers on urban vehicles are different from antennas used for receivers on aircraft, therefore models for nominal biases based on experiments in aviation scenario would not be applicable.
- The main reason for including the nominal biases in ARAIM is the high requirement of the intended application in aviation. The requirement for urban applications is less strict than aviation, observing the proposed integrity budget mentioned above. Therefore the effect of nominal biases can be negligible.

Thus, it is advisable to remove the nominal biases when adapting ARAIM to urban environment.

3.2.4 Noise models

Noise model is the statistical representation of the pseudorange error $\boldsymbol{\epsilon_k}$ in (1.11). For ARAIM algorithm, the noise model represents its assumption about the input pseudorange measurements quality which are affected by errors from various sources. When adapting ARAIM to urban use, the most announced difference between urban environment and airport environment is arguably the multipath condition. While in aviation, the multipath effect is mainly caused by reflecting on the ground (while landing) or on the body of the plane itself [84], multipath in urban environment is caused by tall buildings, trees and other obstacles. Therefore it is necessary to study the impact of the noise model on the integrity monitoring performance of ARAIM, and from there choose a correct model for the intended urban environment.

Current noise model in ARAIM

The noise model of ARAIM is used to calculate the pseudorange error covariance matrices for integrity and continuity monitoring. The diagonal covariance matrices are defined by [18]:

$$C_{int}(i,i) = \sigma^2_{URA,i} + \sigma^2_{tropo,i} + \sigma^2_{user,i} \tag{3.3}$$

$$C_{acc}(i,i) = \sigma^2_{URE,i} + \sigma^2_{tropo,i} + \sigma^2_{user,i} \tag{3.4}$$

where C_{int} is the covariance matrix for integrity-related calculation; C_{acc} is the matrix for accuracy and continuity process; $\sigma_{URA,i}$ and $\sigma_{URE,i}$ are the standard deviation of the clock and ephemeris error of the ith satellite used for integrity and accuracy, respectively; $\sigma_{tropo,i}$ is the standard deviation of the tropospheric delay, $\sigma_{user,i}$ represents the user contribution to the error budget.

For GPS, $\sigma_{user,i}$ is given by[88]:

$$\sigma^{GPS}_{user,i} = \sqrt{\frac{f^4_{L1} + f^4_{L5}}{\left(f^2_{L1} - f^2_{L5}\right)^2}} \sqrt{\sigma^2_{MP} + \sigma^2_{noise}} \tag{3.5}$$

$$\sigma_{MP,i} = 0.13[\text{m}] + 0.53[\text{m}] \exp\left(-\frac{\theta_i}{10[\text{deg}]}\right) \tag{3.6}$$

$$\sigma_{noise,i} = 0.15[\text{m}] + 0.43[\text{m}] \exp\left(-\frac{\theta_i}{6.9\,\text{deg}}\right) \tag{3.7}$$

where f_{L1} is the carrier frequency of GPS L1 signal (1575.42 MHz) and f_{L5} is the carrier frequency of GPS L5 signal (1176.45 MHz); θ_i is the elevation angle in degrees; $\sigma_{MP,i}$ represents the error due to airframe multipath; $\sigma_{noise,i}$ is the error due to wideband noise and interference. It is important to note that the models follow aviation standard [88], developed using carrier-smoothed data from multitude of flights and data collection campaign [84], thus reflecting the working environment of aviation application. The final model represents the error variance as a function of satellite's elevation angle.

For Galileo, $\sigma^{Galileo}_{user,i}$ is given in tabular form, and also depends on the elevation angle of the satellites (see Table 3.3 [18], where θ_i is the elevation angle).

Effect of noise level on the Protection Level

In order to see how sensitive and critical it is to have a good model for integrity assessment in urban environment, an experiment was conducted (on a real dataset)

Table 3.3: Galileo elevation dependent SIS user error [18]

θ_i	$\sigma^{Galileo}_{user,i}$	θ_i	$\sigma^{Galileo}_{user,i}$
5°	0.4529 m	50°	0.2359 m
10°	0.3553 m	55°	0.2339 m
15°	0.3063 m	60°	0.2302 m
20°	0.2638 m	65°	0.2295 m
25°	0.2593 m	70°	0.2278 m
30°	0.2555 m	75°	0.2297 m
35°	0.2504 m	80°	0.2310 m
40°	0.2438 m	85°	0.2274 m
45°	0.2396 m	90°	0.2277 m

to see the effect of different levels of noise on the computed PL. Because the exact noise model is unknown, the experiment started with the ARAIM noise model and used a multiplier, denoted α, on the user noise component to simulate different noise levels. In particular, (3.5) becomes:

$$\sigma^{GPS}_{user,i} = \alpha\sqrt{\sigma^2_{MP} + \sigma^2_{noise}} \tag{3.8}$$

Since $\sqrt{\frac{f^4_{L1}+f^4_{L5}}{\left(f^2_{L1}-f^2_{L5}\right)^2}} \approx 2.5$, the value of α was chosen from 2.5 to 9 in order to simulate the high noise level in urban scenario. This multiplier was applied for all GPS satellites in view. For each value of α, the noise model was recalculated and fed to the ARAIM implementation. The output of the process is the PL. The whole process is implemented in MATLAB.

The data input was a real dataset recorded in NAVIS Centre, Hanoi, Vietnam March 24, 2013, when all 4 available Galileo satellites (at that time) flew over Hanoi. With the clear sky condition, the data set captured about 13-16 satellites in view in total (for GPS and Galileo satellites). For the Galileo satellites, the user contribution $\sigma^{Galileo}_{user,i}$ is defined in tabular form, also depending on the elevation angles of the satellites [18]. The multiplier α was also applied to Galileo satellites by multiplying directly to the predefined values.

The result of the experiment is depicted in Figure 3.1, which shows the HPL

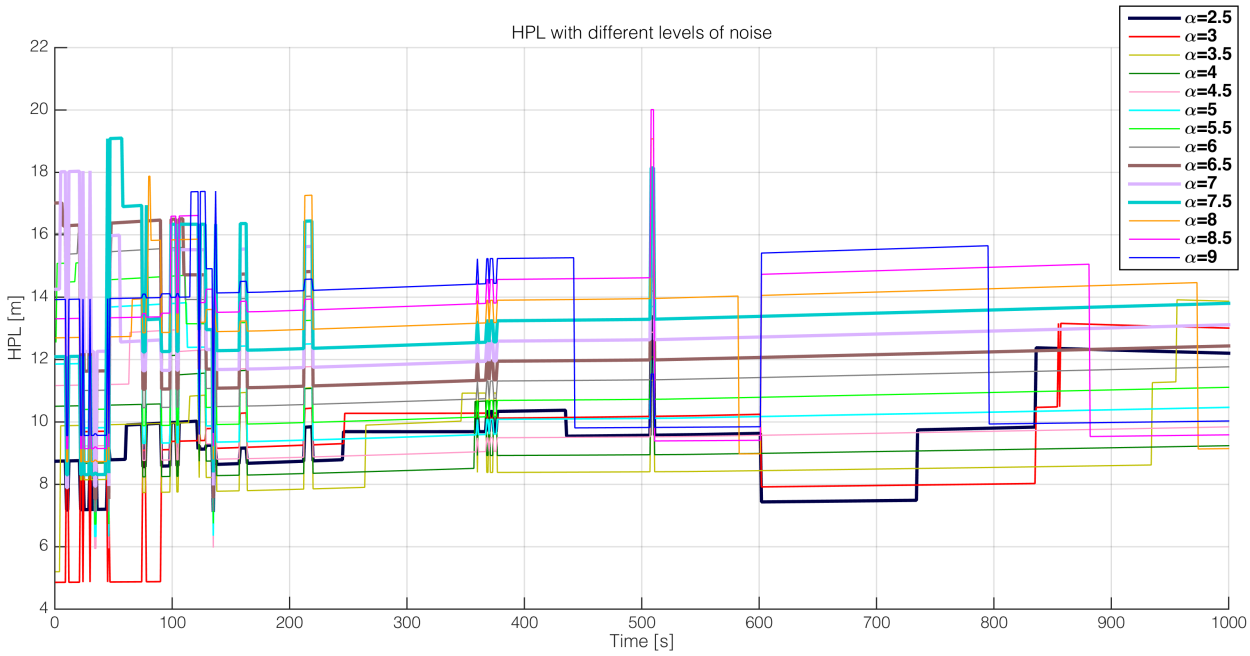
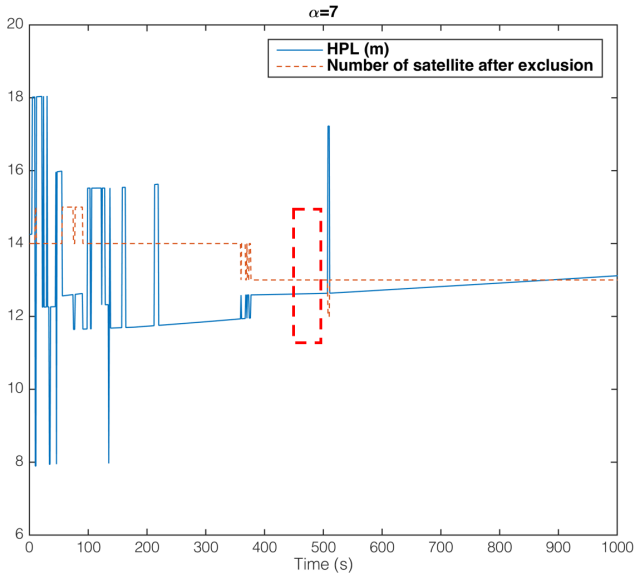


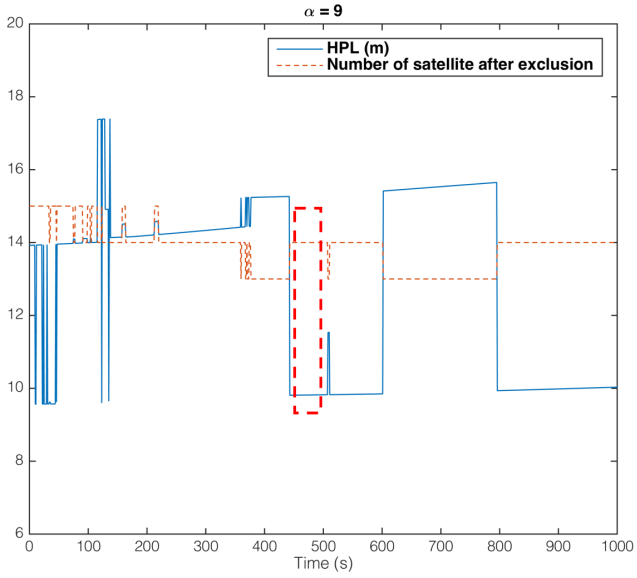
Figure 3.1: HPL with different levels of noise, using multiplier α

with different values of α. This output shows that altering the noise level can affect the HPL, thus altering the availability of the algorithm. It is also important to note that, higher noise level (higher α) does not necessarily lead to higher HPL. This observation is depicted in Figure 3.2, which compares the results (HPL, number of satellites) of two cases of $\alpha = 7$ and $\alpha = 9$. There are several periods in which higher α had more satellites (as highlighted on the figures), resulting in better geometry. Figure 3.3 shows the skyplot of such highlighted period along with the C/N_0 value for each satellites. Galileo satellites are marked with number higher than 40, i.e. number 41, 42, 49, 50 corresponds to Galileo PRN 11, PRN 12, PRN 19, PRN 50, respectively. No satellite was excluded in case of $\alpha = 9$, while satellite Galileo PRN 11 (satellite 41) was excluded in case of $\alpha = 7$ as marked on Figure 3.2a. While having the lowest C/N_0 value, the low elevation angle of this satellite improves the overall geometry in case of $\alpha = 9$, resulting in lower HPL and better availability. In fact, the noise level also affects the SS detection threshold, increases the tolerance towards satellites with lower signal quality in the exclusion process.

On the other hand, the robustness of the algorithm is also worth mentioning.



(a) $\alpha = 7$



(b) $\alpha = 9$

Figure 3.2: HPL and number of satellites in two cases of α

While the noise level increased nearly 4 times from the designed level (α from 2.5 to 9), HPL value only increased in order of meters. Nevertheless, with the observation above, if the real working condition of the urban environment application (or any

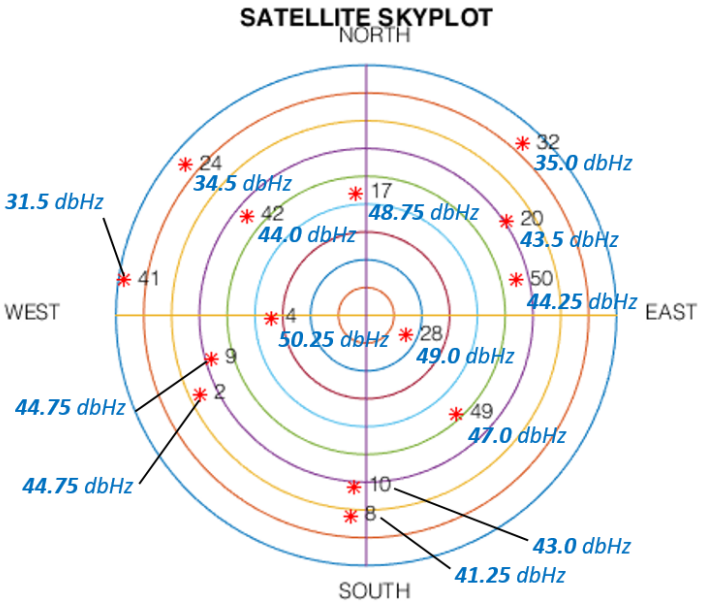
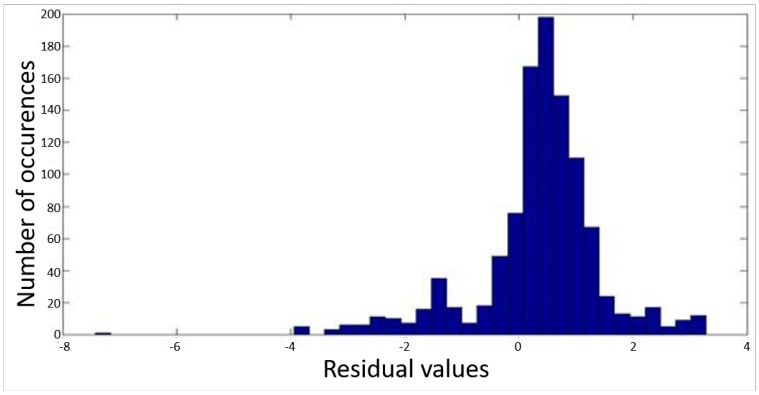


Figure 3.3: Skyplot of the highlighted period

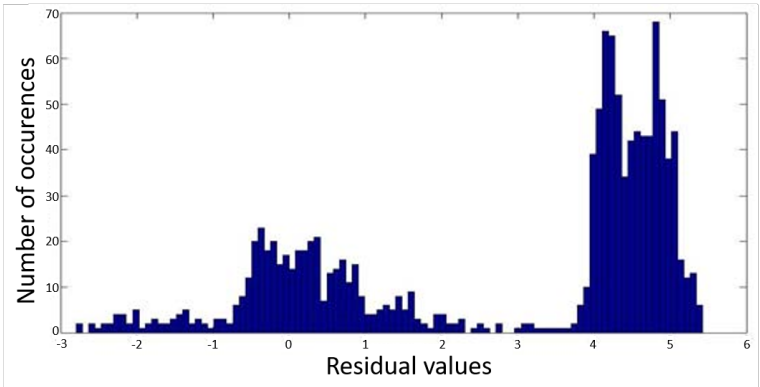
other intended environment) is not as good as the aviation case – mass market receiver, low quality antennas, hostile environment, then using the aviation noise model may result in assessing the integrity situation incorrectly and potential loss of availability.

3.3 Noise model alternatives for urban environment

Considering the importance of finding a good model that can truly represents the working condition of the urban environment, this section is dedicated to investigate three alternatives for ARAIM's model, namely the Mixed Gaussian model [128, 93], a model for urban scenario proposed by Salos Andres for WLSR RAIM [3], and a model proposed by Kuusniemi which bases on C/N_0 ratio [70].



(a) Nominal condition



(b) Constrained condition

Figure 3.4: Example of pseudorange residual distribution for two different satellites [128]

3.3.1 Mixed Gaussian model

The idea of the Mixed Gaussian model [128, 93] is that, while in nominal condition, the pseudorange error distribution is Gaussian in line-of-sight reception state, but in a constrainted environment the distribution is unknown. Figure 3.4 [128] shows two different types of error (pseudorange residual) distribution according to the received satellite. While the error distribution in clear sky condition can be approximated by Gaussian distribution (Figure 3.4a), in a constrained condition it is not trivial to use a single Gaussian model (Figure 3.4b) Therefore, [128] propose approximating the pseudorange error distribution in a constrained environment by a linear mixture of Gaussian distributions. The unknown density function is expressed by:

$$f_X(x) = \sum_{k=1}^{N_G} p_k f_{X,k}(x) \tag{3.9}$$

where N_G is total number of mixture Gaussian component; p_k is the weight of the kth component; $f_{X,k}(x)$ is the density function of the k$^t h$ component, with mean μ_k and variance σ_k^2. In practice, using pseudorange residuals from consecutive epochs, it is possible to evaluate μ_k and σ_k^2 following the Expectation-maximization (EM) clustering algorithm [108].

It is necessary to derive the expression for the variance of the mixed function. From the definition of the mixed function, the ith (raw) moment can be written as:

$$\mu^{(i)} = E_{f_X}\left\{X^i\right\} = \sum_{k=1}^{N_G} p_k E_{f_{X,k}}\left\{X^i\right\} = \sum_{k=1}^{N_G} p_k \mu_k^{(i)} \tag{3.10}$$

where $\mu^{(i)}$ is the ith moment of f_X and $\mu_k^{(i)}$ is the ith moment of $f_{X,k}$.

Thus, the variance of the mixed distribution can be written as:

$$var\left\{X\right\} = \sigma_X^2 = \mu^{(2)} - \left(\mu^{(1)}\right)^2 = \sum_{k=1}^{N_G} p_k \mu_k^{(2)} - \left(\sum_{k=1}^{N_G} p_k \mu_k^{(1)}\right)^2 \tag{3.11}$$

On the other hand, for any kth component, it is possible to write:

$$\mu_k^{(2)} = \sigma_k^2 + \left(\mu_k^{(1)}\right)^2 \tag{3.12}$$

Substitute (3.12) into (3.11), yields:

$$\begin{aligned} var\left\{X\right\} &= \sum_{k=1}^{N_G} p_k \mu_k^{(2)} - \left(\sum_{k=1}^{N_G} p_k \mu_k^{(1)}\right)^2 \\ &= \sum_{k=1}^{N_G} p_k \left(\sigma_k^2 + \left(\mu_k^{(1)}\right)^2\right) - \left(\sum_{k=1}^{N_G} p_k \mu_k^{(1)}\right)^2 \\ &= \sum_{k=1}^{N_G} p_k \sigma_k^2 + \sum_{k=1}^{N_G} p_k \left(\mu_k^{(1)}\right)^2 - \left(\sum_{k=1}^{N_G} p_k \mu_k^{(1)}\right)^2 \end{aligned} \tag{3.13}$$

In other words, the variance of the mixed function is the weighted sum of the

components' variances, plus a non-negative term representing the dispersion of the mixed distribution.

In practice, the process to evaluate the mixed variance can be implemented as follows:

- For each satellite, pseudorange residuals from consecutive epochs are recorded and used to evaluate μ_k and σ_k using EM method.
- The error variance for each satellite can be calculated using expression (3.13).
- The estimated error variances are put into the pseudorange error diagonal covariance matrices of ARAIM [18].

On the other hand, it is possible to apply a sliding window when calculating the means and variances to keep the values current, as well as saving computational time. It is worth mentioning that, choosing a good value for N_G can be tricky depending on the situation, and can be done via experiments and observation.

3.3.2 Andres's model

The second alternative for ARAIM noise model is developed by Salos Andres in [3] for a WLSR RAIM algorithm for urban application. The noise model in this work follows a similar approach to ARAIM's model, consisting of numerous component corresponding to different sources of error:

- Tropospheric error
- Ionospheric error
- Multipath error (LOS)

For tropospheric error, the same model proposed for ARAIM is adopted. On the other hand, for ionospheric error, [3] derived the model from the standard model for civil aviation GPS L1/CA receivers. The final model can be written as:

$$\sigma_{iono} = F \times \tau_v = \begin{cases} F \times 9 & 0 \leq \phi < 20 \\ F \times 4.5 & 20 \leq \phi < 55 \\ F \times 6 & 55 \leq \phi \end{cases}$$
$$F = 1 + 16 \left(0.53 - \frac{\theta}{180\,\text{deg}}\right)^3 \tag{3.14}$$

where ϕ is the geomagnetic latitude of the receiver; θ is the elevation angle of the satellite in degrees; τ_v is the minimum standard deviation of vertical errors, depending on ϕ [88]; F is the mapping function which scales the ionospheric delay estimated for signal arriving at the user's zenith to other elevation angles. It is also important to note that ionospheric error is not considered in the error model of ARAIM, due to the assumption that the receivers should operate in dual-frequency mode, thus mitigating the ionospheric error. However, this may not be the case for receivers used for urban environment applications.

For multipath error, [3] analysed simulated pseudoranges which had been altered by the effects the LMMC Model [116, 76] (see Section 1.3.2). The scheme of the simulation is as follow:

- The urban channel model simulates a vehicle travelling on a 20 m-wide road, with buildings, trees and lampposts on both sides of the road. The height, density of obstacles on both sides are configurable. Using this model, it is possible to calculate a collection of received signal rays from a satellite with varying elevation and azimuth angles.
- The resulting signals are processed by a tracking module to compute the pseudorange error and additional PLL and DLL outputs. Since the correct range from the receiver and the simulated satellite is known at any time instant, the error calculation is straightforward.
- A lock detector based on PLL error and C/N_0 decides whether the receiver is tracking the signal or not.

The result of the simulation is depicted on Figure 3.5 [3]. Besides the narrow-band version of GPS L1 C/A, the multipath errors do not depend on the varying

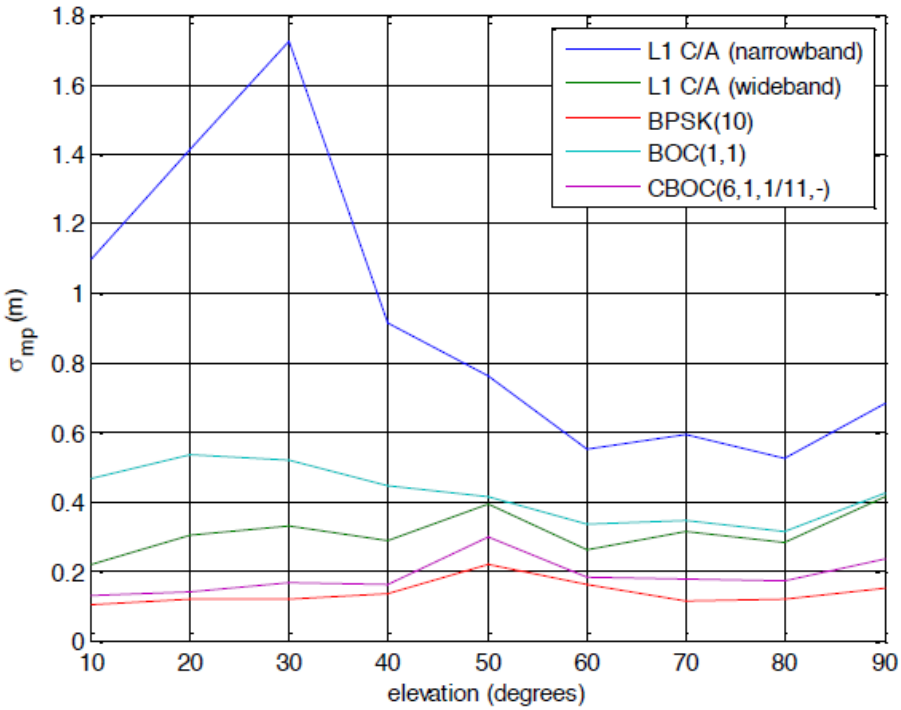


Figure 3.5: Multipath nominal error model

elevation angle, unlike the model developed for aviation application. This model does not cover the NLOS multipath with large delays, because such effect would render the receiver unable to track the signal, thus excluding the samples from the model evaluation.

This model had been validated on simulated data in [3].

3.3.3 Kuusniemi's model

The third option for the noise model in urban scenario is developed by Heidi Kuusniemi in [70]. The model is the result of analyzing pseudorange and pseudorange rate errors in degraded environment (indoor and urban areas), can be consider a solution to improve the solution estimation for weighted positioning approaches, as well as to enhance the performance of reliability assessment process.

In nominal condition (clear visibility, low multipath), the amount and impact of pseudorange errors are usually coherent with the satellites' elevation angles. Higher elevation angles give better signal quality, and vice versa. This observation

has been reflected in the current noise model of ARAIM, which is based on real world observation [84]. However, for constraint environment, such as indoor or urban areas, this may not be the case [131]. In fact, [70] points out the better correlation between pseudorange errors and received signal quality, represented by the C/N_0 ratios. As a result, a noise model for pseudorange and pseudorange rate error was proposed and validated using real data collected in extensive period of time. The noise models for pseudorange error and pseudorange rate error are given as [70]:

$$\sigma_i^2 = a + b \times 10^{\frac{-C/N_{0,i}}{10}} \tag{3.15}$$

$$\dot{\sigma}_i^2 = \dot{a} + \dot{b} \times 10^{\frac{-C/N_{0,i}}{10}} \tag{3.16}$$

where σ_i^2 and $\dot{\sigma}_i^2$ are the variance for pseudorange error and pseudorange rate error, respectively; $C/N_{0,i}$ is the carrier to noise ratio of the signal from the ith satellite, which is time dependent; a, b, $\dot{a}$ and $\dot{b}$ are the coefficients of the model whose values depend on the assumed environment. Some examples of these coefficients are reported in Table 3.4.

Table 3.4: Example values for measurement error variance model

Parameters	Lightly degraded	Heavily degraded
a (m^2)	10	500
b (m^2Hz)	150^2	10^6
$\dot{a}$ (m^2/s^2)	0.01	0.001
$\dot{b}$ (m^2Hz/s^2)	25	40

3.4 Comparison of noise models

In this section, previously introduced noise models will be studied and compared using simulated data. As discussed earlier, a good noise model for ARAIM should be able to represent as close as possible the quality of the measurement, as well as its changes over time. Therefore, the noise models will be superimposed on actual

pseudorange errors obtained from simulated data to verify this.

3.4.1 Signal simulator

The experiments in this work utilize a simulated data set generated by a MATLAB-based generator. The adopted signal generator is an updated version of the work in [120], with additional multi-constellation support (GPS and Galileo) and a multipath generator for urban scenario. The multipath generator uses the aforementioned LMMC Model [116, 76] (see Section 1.3.2). In term of implementation, the multipath generator can be considered as a filter, using the output of the LMMCM as impulse response.

The signal generating process is depicted in Figure 3.6. At first, signals of individual satellites are generated (for both GPS and Galileo), before passing through the multipath generator to be mixed with multipath. A power control block then adjust the power of each signal per predefined C/N_0 settings. Next, all the signals are mixed together, along with white noise before being quantized and saved to output file.

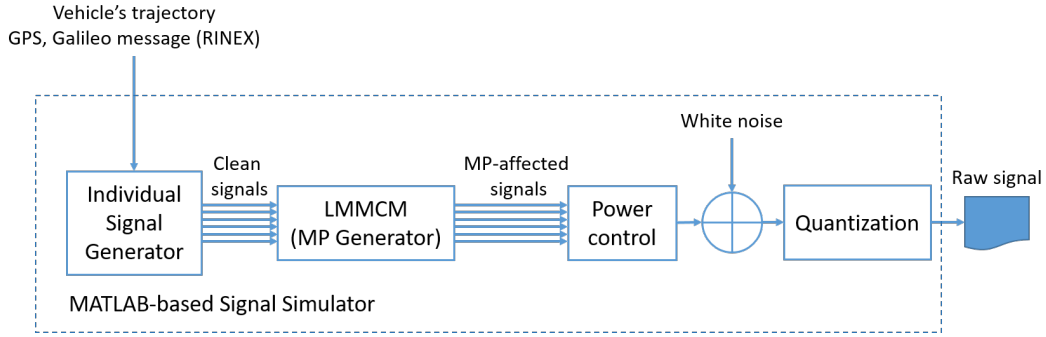


Figure 3.6: Signal simulation scheme

3.4.2 Simulation scheme

The simulation scenario represents a moving vehicle in urban area for 160 seconds with velocity of about 40 km/h. The route is depicted in Figure 3.7, with a slope of 8% (about 4.6°) from 90 s to 140 s. This slope can be considered typical in urban roads in Vietnam. A short segment of the route is affected by multipath

(on all satellites), from 106 s to 154 s. The multipath model was setup in urban surrounding mode, with a road width of 19 m. Both sides of the road were populated with buildings (maximum height 40 m), trees and light posts. To be realistic, in this simulation, the route followed by the car is chosen on the map of Hanoi, Vietnam, and the parameters selected for the multipath model are consistent with the real environment. There are 8 GPS and 6 Galileo satellites visible. The skyplot at the beginning of the simulation is reported in Figure 3.8.

The simulation data was generated with sampling frequency of 16.367 MHz, intermediate frequency of 4.123 MHz, using 5 bits of quantization. The C/N_0 ratio values for all satellites ranges from 37 dB-Hz to 42 dB-Hz, depending on the elevation angles.

3.4.3 Comparison results

The considering noise models are implemented in a MATLAB-based software receiver. The actual pseudorange errors are obtainable easily from the data simulation process. The following noise models are implemented:

Original ARAIM error model as described in [18]

Mixed Gaussian model as described in Section 3.3.1. A sliding windows was also used for this approach, also with the length of 20 epochs. N_G was chosen equal to 2, according to the observation of pseudorange residual distribution of several satellites in the mentioned data set, as shown on Figure 3.9.

WLSR error model proposed by Andres, as described in Section 3.3.2. Note that $\tau_v = 4.8$ since the data set simulates the position in Hanoi, Vietnam.

Kuusniemi's error model as in [70] and introduced in Section 3.3.3. The parameters use the *Lightly degraded* setting in Table 3.4.

The results for two satellites (GPS PRN 6 and Galileo PRN 17) are report in Figure 3.10, where the actual pseudorange error are shown along with the standard deviation of the considering noise models. Each model shows similar trend in both cases. The ARAIM's noise model gives the lowest value, due to the fact that the model was developed for a different working environment, as stated earlier, not to

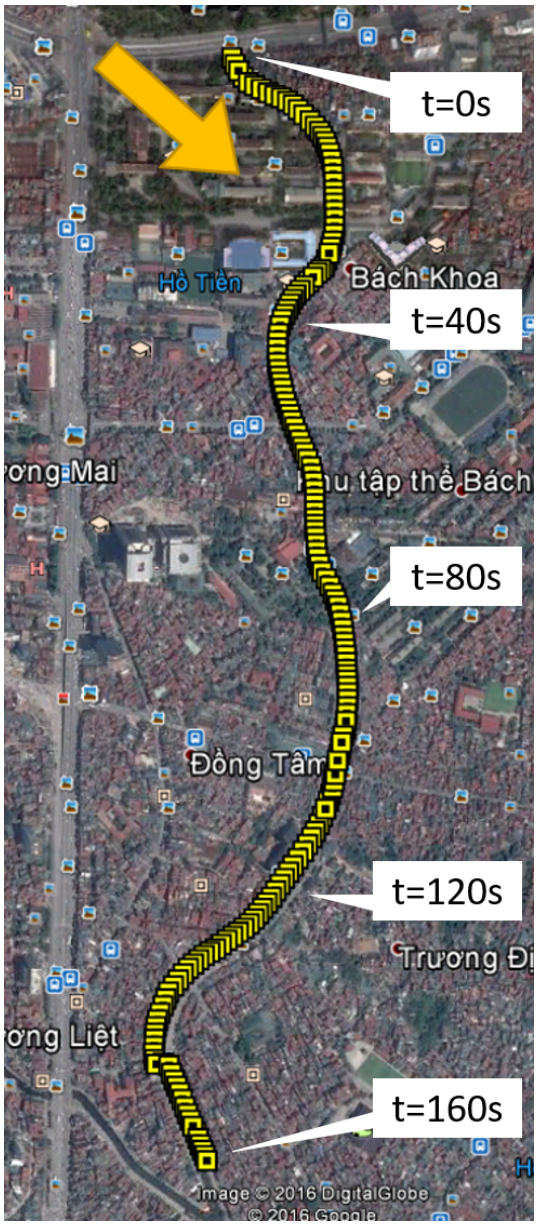


Figure 3.7: Simulation route

mention better professional receivers on planes. On the other hand, the WLSR's model gives the highest value for the standard deviation, considering this is the closest adaptation of the ARAIM's model for urban use. It is worth mentioning that, both models are based on elevation angles of the satellites, which do not change much during the simulation period (less than 3 minutes), therefore the values of those model are almost constant throughout the experiment, even during

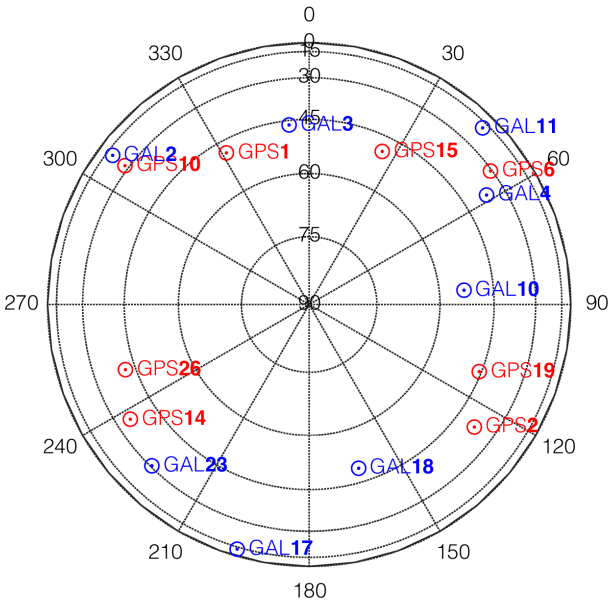


Figure 3.8: Simulation skyplot

the short duration of multipath presence.

The Mixed Gaussian yields a standard deviation value in between the ARAIM and the WLSR noise models, and it does change slightly under the presence of multipath. This is because the model is calculated based on the actual pseudorange residuals, which are undoubtedly affected by multipath or any changes in signal quality. However, the response of this model is not as drastic and immediate as the case of Kuusniemi's model, which adapts instantly with the presence of multipath as well as when it is turned off. The level of Kuusniemi's model is generally lower than the Mixed Gaussian, however.

The noise models are also compared statistically in Table 3.5, which reports the percentage of the standard deviations of the models covering the actual pseudorange error of the two mentioned satellites, as well as the cumulative percentage on all available satellites. In fact, at each epoch, the value of the noise model can be calculated, and compared with the actual pseudorange error (which is known due to the simulated process). The percentage values reported in Table 3.5 are evaluated

Table 3.5: Coverage percentage of the noise models' standard deviations. All values are in percentage unit.

Noise model	GPS PRN 6		Galileo PRN 17		All satellites	
	σ_i	$3\sigma_i$	σ_i	$3\sigma_i$	σ_i	$3\sigma_i$
Kuusniemi model	60.65	95.32	57.98	95.11	51.35	92.14
Mixed Gaussian model	65.93	97.52	71.87	95.90	56.49	93.41
WLSR's model	85.96	100	97.07	100	86.39	100
ARAIM's model	18.71	52.05	26.90	65.59	73.39	99.45

by counting the number of epochs where the calculated standard deviation (and also the 3× std. dev.) is higher than the actual error, dividing by the total number of epochs. Note that, since all models assume Gaussian distribution, the coverage rule can be the criteria to choose a suitable model, from a statistical point of view. In detail, 66% of the actual error should be contained within one standard deviation, and about 99% should be within 3 standard deviation.

The statistical report is coherent with the plots on Figure 3.10 in term of coverage. The ARAIM's model has the lowest coverage, while the WLSR's model with the excessive standard deviation covers about 90% of the errors. On the other hand, the Kuusniemi model and the Mixed Gaussian model share a similar level of coverage (the Mixed Gaussian is slightly higher), and both are sufficient from the statistical perspective.

From the analysis of the experiment above, the Kuusniemi model and the Mixed Gaussian model are the best candidates for noise models in urban scenario, statistically. The Mixed Gaussian model attempts to represent the errors using a theoretical approach, while the Kuusniemi model is developed based more on experiments and observation. However, from the implementation point of view, the Mixed Gaussian is more complex to deploy, requiring the pseudorange residuals from previous epochs. Also, how to choose a good value for the sliding windows and the number of distributions is not an easy problem, as mentioned earlier. On the other hand, the Kuusniemi model's capability to follow closely the change in signal quality is its strong point, considering the environment in urban area can change rapidly when the receiver is in motion. Therefore, the Kuusniemi model is

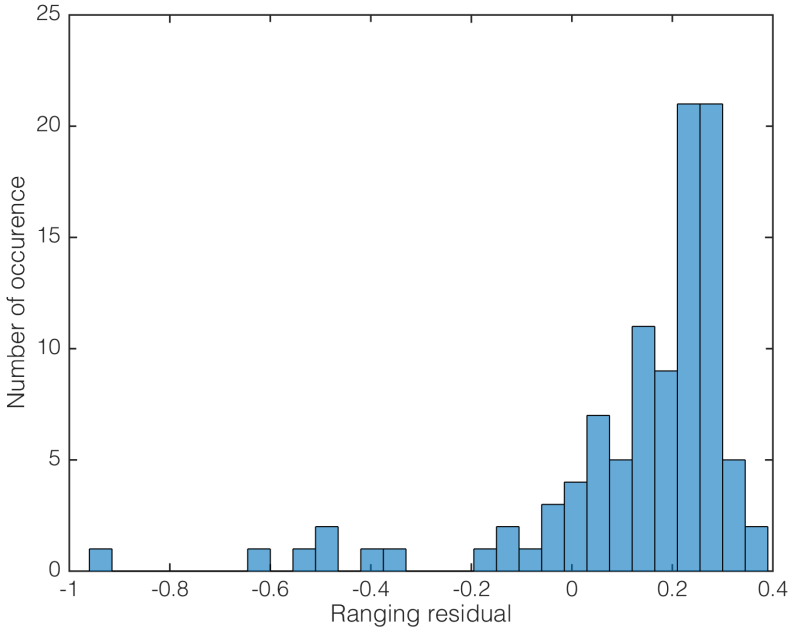
chosen in this work for modeling pseudorange errors in urban environment.

3.5 Conclusion

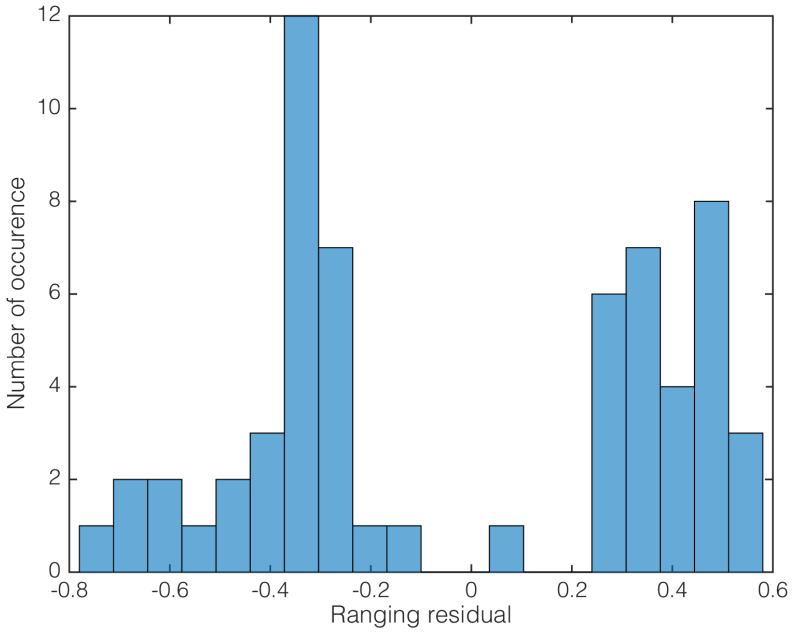
This chapter has listed and investigated three main points to be considered when adapting the integrity assessment algorithms, ARAIM in particular, to urban environment. The changes are the algorithm's constants, which should be considered under the intended application's requirement, the nominal biases, which can be neglected, and the noise model. In fact, the noise model is the most important change in this case, as a reliable representation of the working environment is crucial to the integrity assessment process, as shown in the experiment about the effect of noise levels on calculated PL. It is shown that, incorrect representation of the pseudorange error can lead to potential loss of availability. On the other hand, the experiment also showed the robustness of the ARAIM's current noise model as a notable result.

To cope with the noise model question, three alternatives are presented, implemented and compared. Simulation results and statistical results show that the Kuusniemi model is the best choice for modeling the pseudorange error in urban environment, notably thanks to the capability to follow the change of signal quality during the simulation period.

On the other hand, a MATLAB-based signal generator has been developed to provide simulation data for experiments in this thesis. The generator utilizes the standardized LMMC Model to simulate the multipath effects which are always present in urban environment.

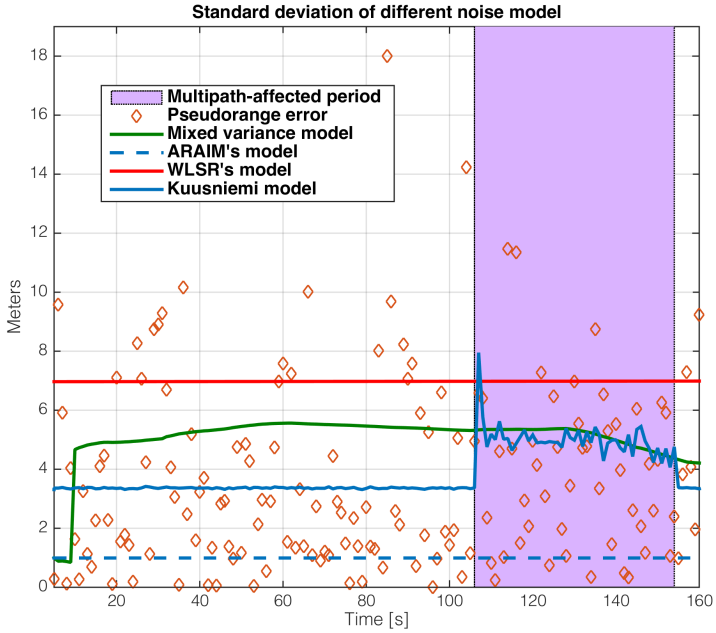


(a) Clean period

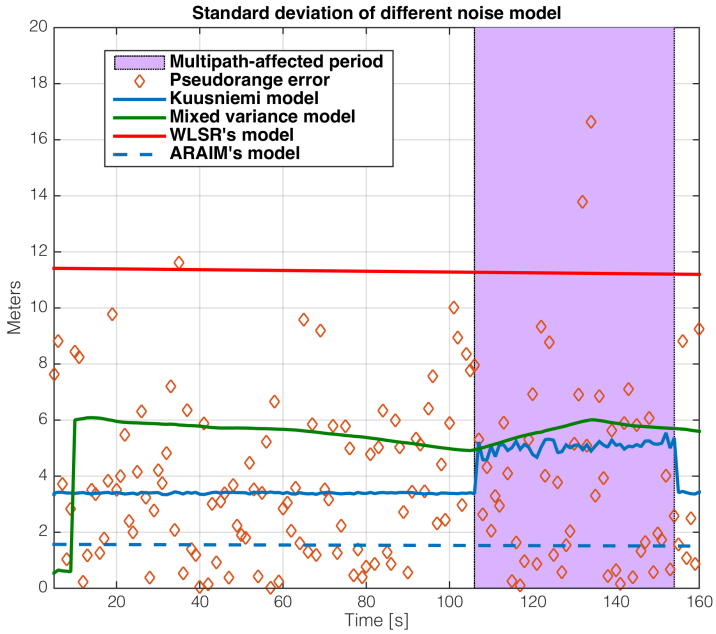


(b) Multipath-affected period

Figure 3.9: Pseudorange residual distribution of a satellite during clean and multipath-affected periods



(a) GPS PRN 6



(b) Galileo PRN 17

Figure 3.10: Standard deviation of different noise models on two satellites

Chapter 4

Kalman filter-based ARAIM for urban environment

In this Chapter, a new algorithm is proposed for integrity monitoring in urban environment, based on the analysis presented in previous Chapters. The proposed algorithm, called KF-based ARAIM can be considered an adaptation of the original LS-based ARAIM (which was developed for aviation) for KF, aiming at providing integrity assessment in urban scenario. The algorithm applies the solution separation approach of ARAIM to the KF scheme presented in Chapter 1, also adopting the C/N0-based noise model introduced in Chapter 3. Part of the content in this Chapter has been published in [123, 126].

4.1 Concept

The overall scheme of the KF-based ARAIM algorithm is presented on Figure 4.1. Instead of predicting forward using the state transition matrix, the algorithm calculates the subset Kalman gains $\boldsymbol{K}_k^{(q)}$, state corrections $\boldsymbol{x}_k^{(q)}$ and the error covariance matrices $\boldsymbol{P}_k^{(q)}$, using subsets of satellites. Here, q denotes a fault hypothesis (sometimes called *fault mode* in literatures[18, 134]) corresponding to a subset of satellites. The initial all-in-view state vector $\boldsymbol{x}_k^{(0)}$ and subset state vectors $\boldsymbol{x}_k^{(q)}$ are used as inputs for the SS tests, which measure the consistency of the measurement based on the deviation between the subset states and the all-in-view state. If any of the tests fails, exclusion will be attempted on potential faulty satellites. Otherwise,

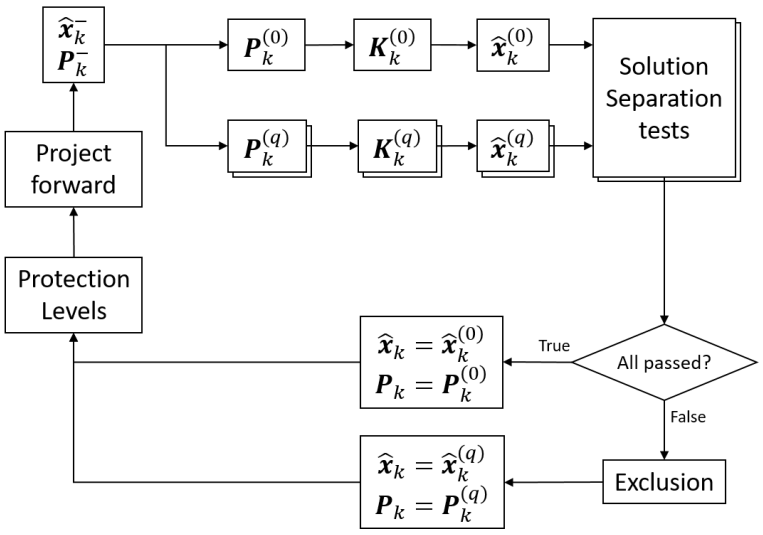


Figure 4.1: Scheme of the KF-based ARAIM algorithm

the algorithm will proceed to calculate the Protection Levels, before projecting the state and error covariance matrix ahead. The whole process is then repeated for the next epoch.

4.2 Subset derivation

The SS approach of the algorithm involves running the Kalman filter repeatedly on multiple subsets of satellites. The subset of satellites of fault hypothesis q is denoted $S^{(q)}$, while $S^{(0)}$ denotes the set containing all the satellites in view. For each fault hypothesis q, a diagonal matrix $\boldsymbol{W}^{(q)}$ is defined as:

$$\boldsymbol{W}^{(q)}(i,i) = \boldsymbol{W}^{(q)}(i+N_{sat}, i+N_{sat}) = \begin{cases} 1 & \text{if } i \text{ is in } S^{(q)} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

With this, the subset observation matrix $\boldsymbol{H}_k^{(q)}$, measurement vector $\boldsymbol{z}_k^{(q)}$ and measurement error covariance $\boldsymbol{R}_k^{(q)}$ are given by:

$$\begin{aligned}\boldsymbol{H}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{H}_k \\ \boldsymbol{z}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{z}_k \\ \boldsymbol{R}_k^{(q)} &= \boldsymbol{W}^{(q)}\boldsymbol{R}_k\boldsymbol{W}^{(q)} = \boldsymbol{W}^{(q)}\boldsymbol{W}^{(q)}\boldsymbol{R}_k = \boldsymbol{W}^{(q)}\boldsymbol{R}_k\end{aligned} \tag{4.2}$$

Noting that, both $\boldsymbol{R}_k$ and $\boldsymbol{W}^{(q)}$ are diagonal matrices, therefore their multiplication are commutative. Let $\boldsymbol{R}_k^{(q)+}$ be the Moore-Penrose pseudoinverse of $\boldsymbol{R}_k^{(q)}$. In fact, since the pseudoinverse of $\boldsymbol{W}^{(q)}$ is itself and $\boldsymbol{R}_k^{-1}$ is also diagonal, $\boldsymbol{R}_k^{(q)+}$ is given by:

$$\boldsymbol{R}_k^{(q)+} = \boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1} \tag{4.3}$$

Replacing (4.2) and (4.3) into (1.15), the subset state covariance matrix $\boldsymbol{P}_k^{(q)}$ can be obtained from:

$$\begin{aligned}\left(\boldsymbol{P}_k^{(q)}\right)^{-1} &= \left(\boldsymbol{P}_k^{-}\right)^{-1} + \boldsymbol{H}_k^{(q)T}\boldsymbol{R}_k^{(q)-1}\boldsymbol{H}_k^{(q)} \\ &= \left(\boldsymbol{P}_k^{-}\right)^{-1} + \boldsymbol{H}_k^{T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1}\boldsymbol{H}_k^{T}\end{aligned} \tag{4.4}$$

Similarly, replacing (4.2) and (4.3) into (1.17), the subset Kalman Gain $\boldsymbol{K}_k^{(q)}$ can be written as:

$$\boldsymbol{K}_k^{(q)} = \boldsymbol{P}_k^{(q)}\boldsymbol{H}_k^{(q)T}\boldsymbol{R}_k^{(q)+} = \boldsymbol{P}_k^{(q)}\boldsymbol{H}_k^{T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1} \tag{4.5}$$

The subset state update $\boldsymbol{x}_k^{(q)}$ can be expressed as:

$$\begin{aligned}\hat{\boldsymbol{x}}_k^{(q)} &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k^{(q)}\left(\boldsymbol{z}_k^{(q)} - \boldsymbol{H}_k^{(q)}\hat{\boldsymbol{x}}_k^{-}\right) \\ &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{P}_k^{(q)}\boldsymbol{H}_k^{T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1}\left(\boldsymbol{W}^{(q)}\boldsymbol{z}_k - \boldsymbol{W}^{(q)}\boldsymbol{H}_k\hat{\boldsymbol{x}}_k^{-}\right) \\ &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{P}_k^{(q)}\boldsymbol{H}_k^{T}\boldsymbol{W}^{(q)}\boldsymbol{R}_k^{-1}\boldsymbol{W}^{(q)}\left(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^{-}\right) \\ &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k^{(q)}\left(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^{-}\right)\end{aligned} \tag{4.6}$$

4.3 Algorithm description

The proposed algorithm uses the apriori estimate $\hat{\boldsymbol{x}}_k^-$ and the apriori error covariance $\boldsymbol{P}_k^-$ as inputs. Their initial estimates can be obtained processing several initialization epochs using least-square positioning method. Once initialized, the algorithm consists of six steps, detailed as follows (see Figure 4.2).

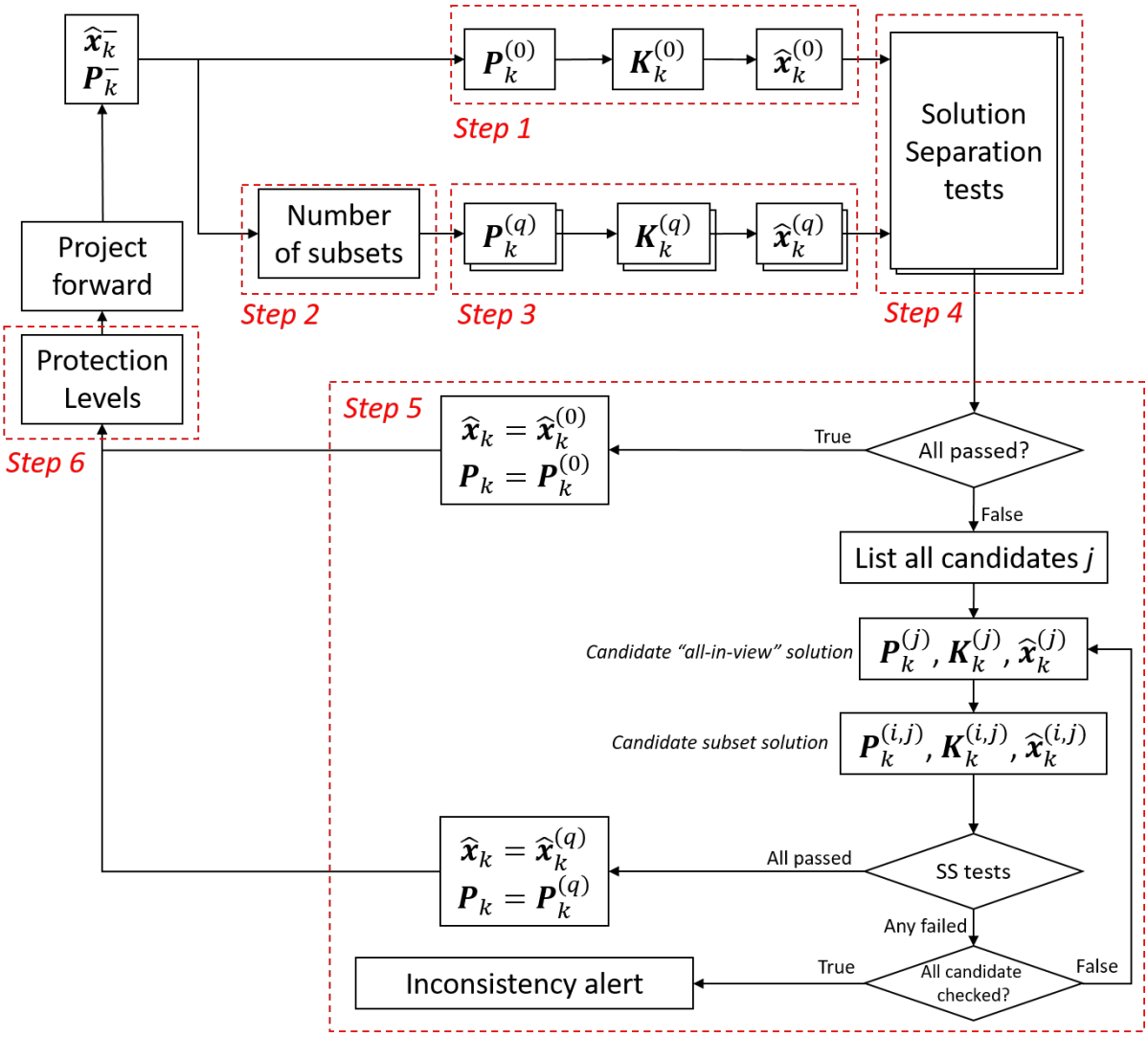


Figure 4.2: Detailed scheme of the KF-based ARAIM algorithm

Step 1. All-in-view state update

In this step, the state vector and error covariance are updated following the

aforementioned KF loop, using all available measurements. Note that the observation matrix $\boldsymbol{H}_k^{(0)}$ has to be defined in ENU coordinate, using the apriori estimate $\hat{\boldsymbol{x}}_k^-$ as the origin. The measurement error covariance matrix $\boldsymbol{R}_k^{(0)}$ is evaluated for every epoch based on the C/N_0 values of all satellites, using relations (3.15) and (3.16) [70].

The output state vector and error covariance matrix are denoted $\hat{\boldsymbol{x}}_k^{(0)}$ and $\boldsymbol{P}_k^{(0)}$, respectively.

Step 2. Number of fault hypotheses

This step follows the formula of conventional ARAIM, presented in [18]. Output of this step is the maximum number of concurrent satellite faults $N_{sat,max}$ and the total number of fault modes N_{fault}. In other word, N_{fault} represents the total number of possible satellite combinations by excluding from 1 to $N_{sat,max}$ satellites:

$$N_{fault} = \sum_{g=1}^{N_{sat,max}} \binom{N_{sat}}{N_{sat}-g} \tag{4.7}$$

Step 3. Subset state update

Subset state can be obtained by doing the correction step of KF using a subset of measurements, excluding potential faulty satellites. For each q from 1 to N_{fault}, the algorithm determines:

- The subset state $\hat{\boldsymbol{x}}_k^{(q)}$,
- The difference $\Delta\hat{\boldsymbol{x}}_k^{(q)}$ between $\hat{\boldsymbol{x}}_k^{(q)}$ and $\hat{\boldsymbol{x}}_k^{(0)}$
- The covariance of $\Delta\hat{\boldsymbol{x}}_k^{(q)}$, denoted $\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)}$

The posteriori state covariance $\boldsymbol{P}_k^{(q)}$ is obtained from (4.4), rewritten here for completeness:

$$\left(\boldsymbol{P}_k^{(q)}\right)^{-1} = \left(\boldsymbol{P}_k^-\right)^{-1} + \boldsymbol{H}_k^T \boldsymbol{W}^{(q)} \boldsymbol{R}_k^{-1} \boldsymbol{H}_k^T \tag{4.8}$$

The subset Kalman gain is computed as (4.5):

$$\boldsymbol{K}_k^{(q)} = \boldsymbol{P}_k^{(q)} \boldsymbol{H}_k^T \boldsymbol{W}^{(q)} \boldsymbol{R}_k^{-1} \tag{4.9}$$

The subset state estimation update can be obtained as (4.6):

$$\hat{\boldsymbol{x}}_k^{(q)} = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k^{(q)}(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-) \tag{4.10}$$

The difference between subset state and all in view state updates is thus:

$$\begin{aligned}\Delta\hat{\boldsymbol{x}}_k^{(q)} &= \hat{\boldsymbol{x}}_k^{(q)} - \hat{\boldsymbol{x}}_k \\ &= (\boldsymbol{K}_k^{(q)} - \boldsymbol{K}_k)(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\end{aligned} \tag{4.11}$$

The covariance matrix $\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)}$ of $\Delta\hat{\boldsymbol{x}}_k^{(q)}$ is calculated as

$$\begin{aligned}&\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)} = \boldsymbol{P}_k^{(0)} + \boldsymbol{P}_k^{(q)} \\ &- 2\left((\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k^{(q)}\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^{(q)T}\right)\end{aligned} \tag{4.12}$$

The detailed derivation for $\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)}$ can be found in Appendix A.

The alternative state covariance matrix $\boldsymbol{P}_{k,alt}^{(q)}$, which will be used to calculate the PL, is given by:

$$\begin{aligned}\boldsymbol{P}_{k,alt}^{(q)} = &\left(\boldsymbol{I} - \boldsymbol{K}_k^{(q)}\boldsymbol{H}_k\right)\boldsymbol{P}_k^-\left(\boldsymbol{I} - \boldsymbol{K}_k^{(q)}\boldsymbol{H}_k\right)^T \\ &+ \boldsymbol{K}_k^{(q)}\boldsymbol{R}_k\boldsymbol{K}_k^{(q)T}\end{aligned} \tag{4.13}$$

The derivation of $\boldsymbol{P}_{k,alt}^{(q)}$ is provided in Appendix B.

Step 4. Solution separation tests

Let s = 1,2,3 corresponds to east, north, up component, respectively. For each q, the SS test is executed on all 3 components of coordinates: north, east and up. For each component, the test threshold is calculated as:

$$T_{k,s}^{(q)} = K_{fa,s}\sigma_{\Delta x,s} \tag{4.14}$$

Where:

$$\sigma_{\Delta x,s} = \left(\sqrt{\boldsymbol{P}^{(q)}_{\Delta \hat{\boldsymbol{x}}_k}} \right)_{s,s} \tag{4.15}$$

The coefficients $K_{fa,s}$ follow the same formula as for ARAIM [18]:

$$K_{fa,1} = K_{fa,2} = Q^{-1}\left(\frac{P_{FA\_HOR}}{4N_{fault}} \right) \tag{4.16}$$

$$K_{fa,3} = Q^{-1}\left(\frac{P_{FA\_VERT}}{2N_{fault}} \right) \tag{4.17}$$

Where $Q^{-1}(p)$ is the $(1-p)$-quantile of a Normal distribution; P_{FA_HOR} and P_{FA_VERT} are the probability of false alarm allocated along the horizontal and vertical direction, respectively, and are usually chosen based on the intended application.

The SS test is considered passed if for all q and s we have:

$$\tau^{(q)}_{k,s} = \frac{\left| \Delta \hat{\boldsymbol{x}}^{(q)}_{k,s} \right|}{T^{(q)}_{k,s}} \leq 1 \tag{4.18}$$

Step 5. Checking for exclusion

If all the SS tests are passed, the whole satellite set is considered clean, and the all-in-view state vector and associated matrices will be used for the projection step. In contrast, if any of the test failed, the fault hypothesis with failed test becomes a candidate for exclusion.

The failed fault hypotheses are sorted in decreasing order of $\tau^{(j)}_{k,s}$, with j denotes any failed fault hypothesis, or exclusion candidate. For each j, the satellites set $S^{(j)}$ are treated as all-in-view set, then the algorithm (from Step 1 to Step 4) is executed again, using all subset solution $\hat{\boldsymbol{x}}^{(j,q)}_k$ within the candidate set $S^{(j)}$. If all the SS test of the considering candidate are passed, i.e. $\tau^{(j,q)}_{k,s} < 1 \; \forall q \neq j$, then $S^{(j)}$ is considered a consistent satellite set, the remaining satellites (those in $S^{(0)} - S^{(j)}$) will be excluded. The state vector and associated matrices of $S^{(j)}$ will then be used for subsequent calculation step.

When a good satellite set is found (with all SS tests passed), the algorithm will proceed to Protection Level calculation. On the other hand, if none of the candidate is deemed consistent, the algorithm will give out an inconsistent alert.

Step 6. Protection Level

Define:

$$\sigma_s^{(q)} = \sqrt{\left[\boldsymbol{P}_{k,alt}^{(q)}\right]_{s,s}} \; \forall q = 0..N_{fault} \tag{4.19}$$

Let $P_{HMI,VERT}$ and $P_{HMI,HOR}$ denote the integrity risk allocated to the vertical and horizontal, respectively. Similar to P_{FA_HOR} and P_{FA_VERT}, the allocated integrity risks are chosen based on the intended application. In case of no exclusion, the VPL is the solution to the equation:

$$\begin{aligned} P_{HMI,VERT} = 2Q\left(\frac{VPL}{\sigma_3^{(0)}}\right) \\ + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL - T_{k,3}^{(q)}}{\sigma_3^{(q)}}\right) \end{aligned} \tag{4.20}$$

Where $Q(\cdot)$ is the complement of the Normal cumulative distribution function. Similarly, the HPL is calculated on 2 directions of the horizontal plane, namely HPL_1 and HPL_2, as follows (with $s = 1{,}2$):

$$\begin{aligned} \frac{1}{2}P_{HMI,HOR} = 2Q\left(\frac{HPL_s}{\sigma_s^{(0)}}\right) \\ + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_s - T_{k,s}^{(q)}}{\sigma_s^{(q)}}\right) \end{aligned} \tag{4.21}$$

The final HPL is then computed as:

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{4.22}$$

On the other hand, if exclusion was attempted and subset candidate j was excluded from the satellite set, then the PL equations are defined to account for the event. In this case, the VPL is the solution to the equation:

$$P_{HMI,VERT} = 2Q\left(\frac{VPL}{\sigma_3^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL}{\sigma_3^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL - T_3^{(j,q)}}{\sigma_3^{(j,q)}}\right) \tag{4.23}$$

where $N_{fault,j}$ is the number of fault hypotheses in the remaining set of satellites after excluding exclusion candidate j.

The HPL is calculated in similar manner as in the no detection case:

$$\frac{1}{2}P_{HMI,HOR} = 2Q\left(\frac{HPL_1}{\sigma_1^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_1}{\sigma_1^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(j,q)}}{\sigma_1^{(j,q)}}\right) \tag{4.24}$$

$$\frac{1}{2}P_{HMI,HOR} = 2Q\left(\frac{HPL_2}{\sigma_2^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_2}{\sigma_2^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_2^{(j,q)}}{\sigma_2^{(j,q)}}\right) \tag{4.25}$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{4.26}$$

The detailed derivation to obtain the PL equations is provided in Appendix B. The method to solve these equation is provided in Appendix C.

It is important to note that, the proposed algorithm and its underlying KF model adopt the Kuusniemi's noise model, from the analysis in the previous chapter. With this model adopted, the presence of multipath (or the change of signal quality in general) can affect the PL. In the presence of multipath, the C/N_0 ratios of affected satellites drop, which increase $\boldsymbol{R}_k$. This leads to the increase of $\sigma_s^{(q)}$ and $T_{k,s}(q)$ for all q, due to $T_{k,s}(q)$ is calculated from $\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)}$ (as in (4.14) and (4.15)). As a result, the PL increases. Intuitively, the drop of signal quality leads to a higher PL, as the receiver should expect higher error in such situation. Section 4.4.1 will further confirm this analysis in an experiment.

4.4 Performance analysis

The proposed KF-based ARAIM algorithm has been implemented in MATLAB and tested against simulation data as well as real data collection. The simulated data is generated from a MATLAB signal generator introduced in Section 3.4.1, with the configuration described in Section 3.4.2. The simulated data allows the algorithm to be tested in a controlled environment, where the errors are known (such as pseudorange measurement errors, positioning errors), thus validating the proposed algorithm. Also, the simulated data consists of clean and multipath-affected segment, thus showing distinctive behaviour of the algorithm in different signal condition.

On the other hand, the real data set is collected in Turin, Italy [1]. Experiment on real data can show how the algorithm behaves and responses in a realistic scenario, where the car is moving through different environments.

The test results on both sets of data are presented below.

4.4.1 Results on simulation data

Subset positions

A pseudorange bias of 200m was injected to two satellites (PRN 2 of GPS and PRN 17 of Galileo) in a short period, from 37 s to 44 s. The subset positions in both cases, with and without pseudorange fault, are calculated and compared at a single epoch (40s) to study the effect of such fault on measurement consistency. The $C/N0$ ratios for the satellites are 39.7 dB-Hz and 40.1 dB-Hz, respectively. In figure 4.3 and 4.4 are reported all the possible estimated positions considering all the possible subsets of satellites (for both the cases of bias and bias free measurements). On the figures, the plus marker represents the true position (which is known thanks to the simulation process), the diamond marker represents the all-in-view solution, while the dots are the subset positions, the star marker is the final output position (after exclusion once the faults are detected), and the large circle depicts the HPL. It is

[1] The real data collections used to test and validate the proposed algorithm were kindly provided by Gianluca Marucco, Micaela Troglia Gamba (Istituto Superiore Mario Boella (ISMB), Italy), and Hong Lam Nguyen (Politecnico di Torino).

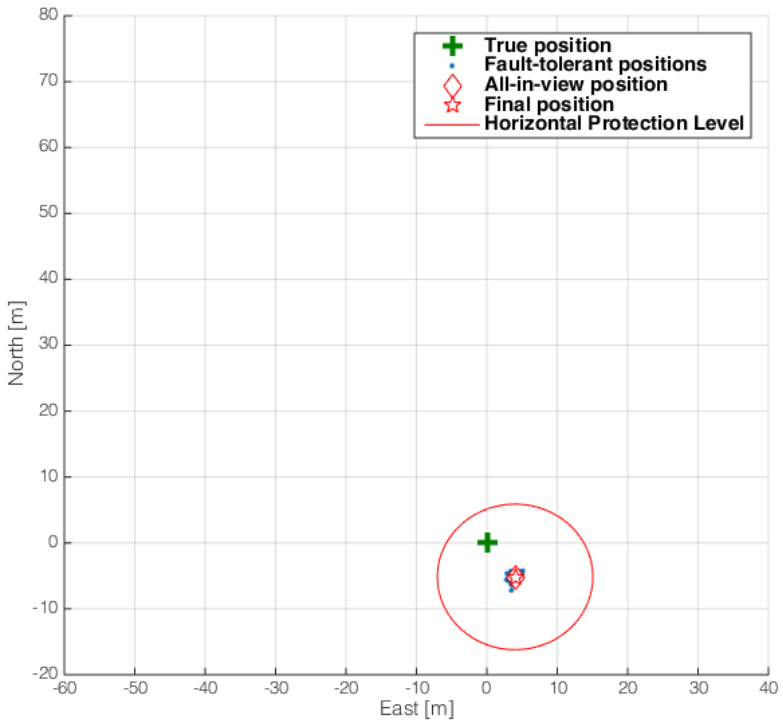


Figure 4.3: Subset positions in case of no fault

observed that for the clean data, in Figure 4.3, the subset positions are relatively consistent to the all-in-view one, deviated around 5 meters around the true position. In this case, the final position is the all-in-view position, since no fault is detected and exclusion is not necessary. On the other hand, as shown in Figure 4.4, the faulty measurement shows lower consistency, where the subset positions deviate more than 50 meters from the all-in-view solution and far away from the true position, obviously. If the exclusion mechanism is not applied, the output (which is the all-in-view position) can deviate about 50 m away from the true position. The final position after FDE is less than 10 m away from the true position. It's worth mentioning that the HPL on both plots also cover the true position, satisfying the definition mentioned at the beginning of the paper.

On the other hand, it can be observed that the HPL in case of fault is slightly higher than in the clean case. This is because the two faulty satellites have been successfully detected and excluded, thus mitigating the potential error but also

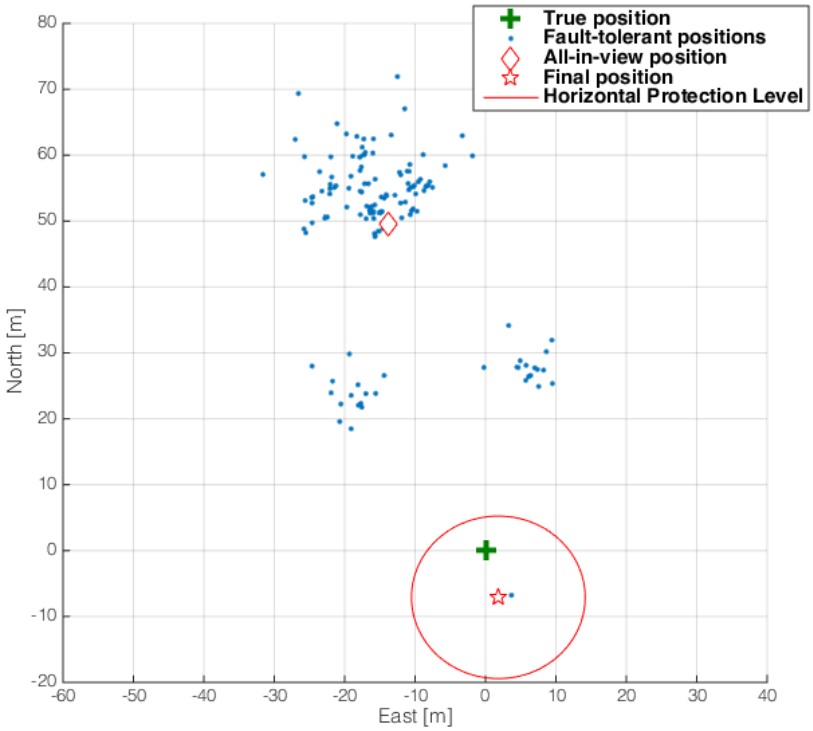


Figure 4.4: Subset positions in case of two faults injected

inevitably degrading the overall satellite geometry.

Fault detection and exclusion

Similar to the analysis of the subset positions, a random ranging fault was injected to 2 satellites: PRN 26 of GPS and PRN 2 of Galileo for a short period. The ranging fault is normal distributed with mean of 200 m and standard deviation of 10 m. The fault persisted for 8 s, from 37 s to 44 s.

Table 4.1: Values for Probabilities of False Alarm

Parameter	Value
P_{FA}	6.65×10^{-5}
P_{FA_HOR}	6.317×10^{-5}
P_{FA_VERT}	3.325×10^{-6}

The SS tests were executed on the whole simulated data. For the coefficients defined in (4.16) and (4.17), the values for Probabilities of False Alarm are provided in Table 4.1, which is the same order of magnitude to the value used in [109]. On the other hand, the measurement noise covariance matrix follows the model in [70], using the *Lightly degraded* environment parameters as suggested in previous experiment.

The proposed method successfully detected and excluded the faulty satellite throughout the fault occurrence. The positioning results in both cases - without and with fault exclusion - are reported in Figure 4.5. Note that the faulty positioning results are indicated, and deviate about 30 m from the true positions. With fault exclusion, the route remained smooth and correct.

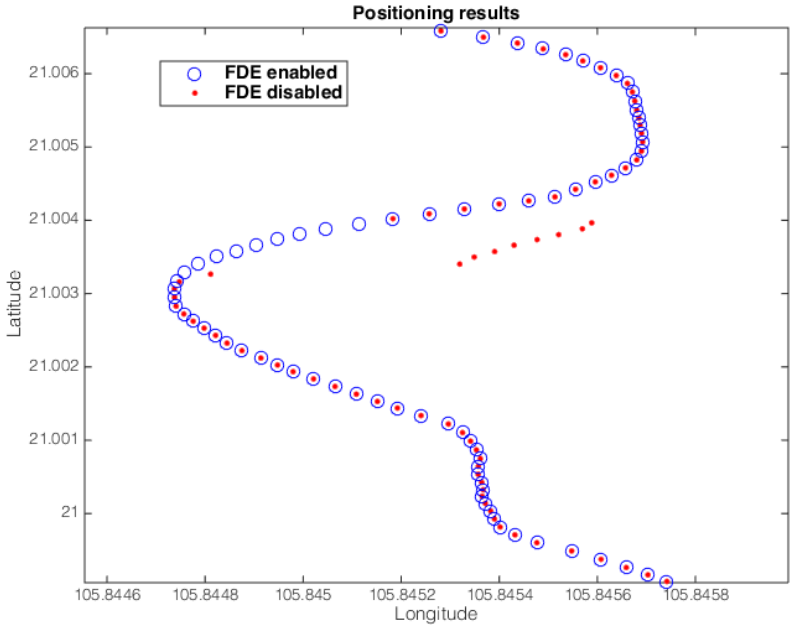


Figure 4.5: Positioning results with fault exclusion

Protection Levels

After all the Solution Separation tests are passed, the algorithm should proceed with the Protection Levels calculation. In this simulation, the integrity budget is allocated as in Table 3.2. Recall that the risk budget is allocated much higher in

horizontal direction rather than vertical direction, as navigation in urban traffic usually relies more on horizontal direction. Figure 4.6 and 4.7 show the HPL and the Horizontal Positioning Error (HPE) for both LS-based ARAIM and the proposed KF-based ARAIM, respectively. Two segments are highlighted on both figures. The first features the manually injected faults on two satellites as described in Section 4.4.1, which persisted from 37 s to 44 s. The second highlighted period corresponds to the multipath-affected period as introduced in Section 3.4.2.

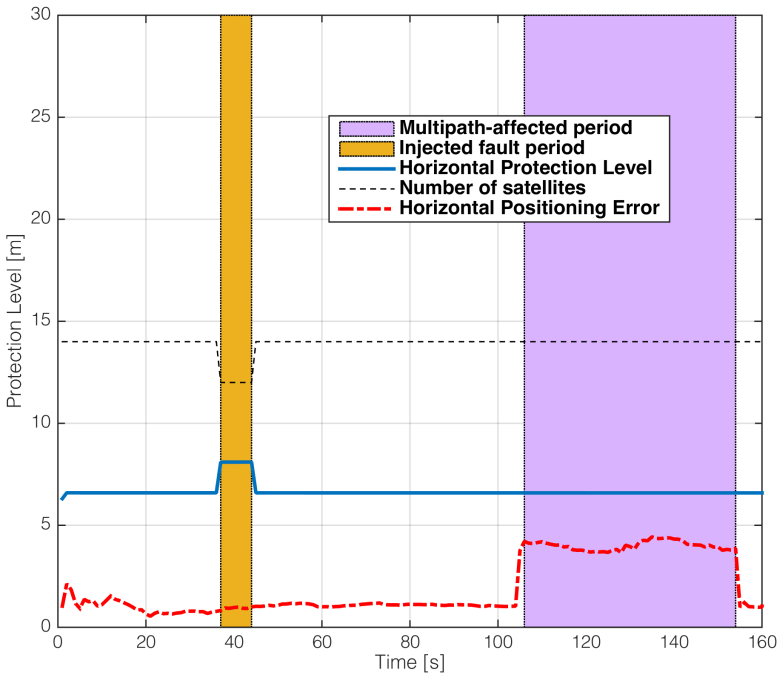


Figure 4.6: Protection Levels and Positioning Error for LS-based ARAIM

Observing figure 4.6, the HPL of the LS-based ARAIM is about 6 m for most of the simulation duration. During the manual fault period (the first highlighted period), the algorithm successfully detected and excluded the faulty satellites, hence the lower number of satellites and increases the HPL. However, the HPL remains the same during the multipath period. This is due to the noise model of the algorithm relies solely on the elevation angles of the satellites, which remain mostly the same throughout the simulation period.

In contrast, the KF-ARAIM method's HPL in Figure 4.7 shows some drastic

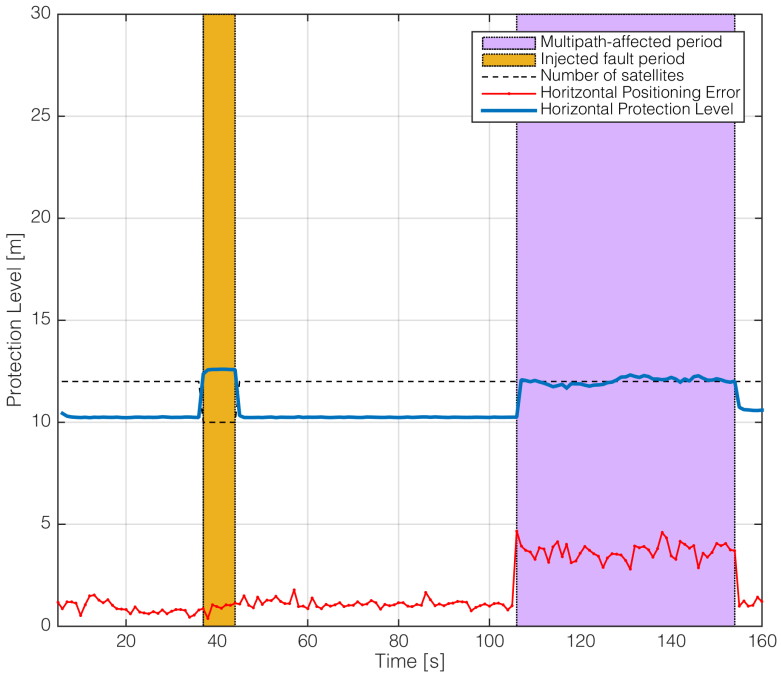


Figure 4.7: Protection Levels and Positioning Error for KF-based ARAIM

changes. During the clean period, the HPL is around 10 m. This is higher than the HPL evaluated by the LS-based ARAIM, which is mainly contributed by the different noise models. As shown in Section 3.4.3, the Kuusniemi's model (which is integrated in the proposed algorithm) yields a higher value (more than double) for the standard deviation than the original ARAIM error model. While this might look like an advantage for the original ARAIM algorithm, it is important to note that the clean segment of the data is simply for comparison and testing purpose, and does not realistically reflect a real-life signal in urban scenario - hence the necessity of the multipath-affected signal. In the manual fault period (first highlighted period), similar to the LS-based ARAIM, the algorithm successfully detected and excluded the faulty satellites, thus reducing the number of satellites. This reduces the quality of the overall satellite geometry, thus increased the HPL. During the multipath period, the HPL increases up to about 14 m due to the decrease of C/N_0 becaused of the multipath presence. This observation confirms the analysis in Section 4.3 about the effect of multipath on evaluated PL using the proposed method.

4.4.2 Real data collection

Figure 4.8: Real data collection route

Besides the simulation data, the performance analysis also utilizes real data sets collected in the streets of Turin, Italy on a car. The GNSS signals are collected using an ANTCOM antenna, mounted on the roof of the car, connected to the NSL STEREO Frontend [95] and saved to file using a grabber running on a laptop. The reference track was acquired simultaneously by a NovAtel receiver with an Inertial Measurement Unit (IMU) [115], using the same signals splitting from the mounted antenna.

The route of the data collection is depicted in Figure 4.8 as the red line. The collection is about 800 seconds long. The beginning of the route is a test run in a large courtyard in the main campus of Politecnico di Torino, with low building nearby. For the remaining duration, the car moved along normal streets, Corso Francesco Ferucci, with trees and apartment building on both side of the road. Some photos showing the environment are presented also in Figure 4.8. The data set consists of 5 GPS satellites and 2 Galileo satellites throughout the period. The

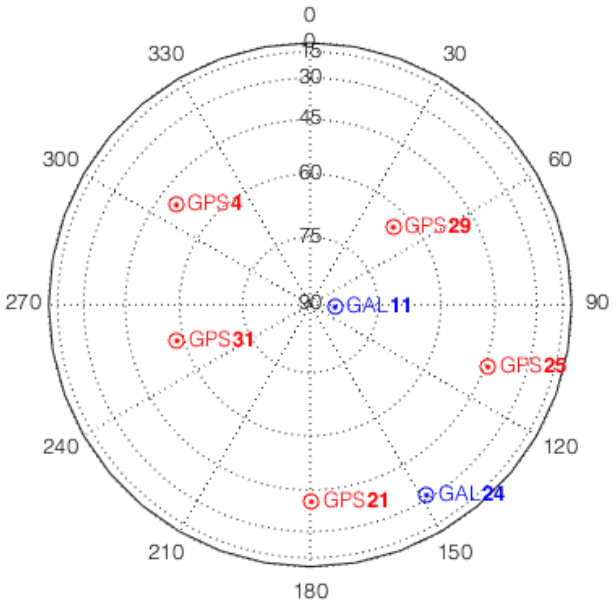


Figure 4.9: Skyplot of the real data set

skyplot of the data set is shown in Figure 4.9.

4.4.3 Results on real data

Figure 4.10: Positioning results of the real data collection

The real data set was processed using the proposed KF-based ARAIM algorithm, and the output positioning results are compared with the reference track obtained from the Novatel receiver introduced in Section 4.4.2. The positioning results are plotted in Figure 4.10 as yellow squares. Several segments of interest

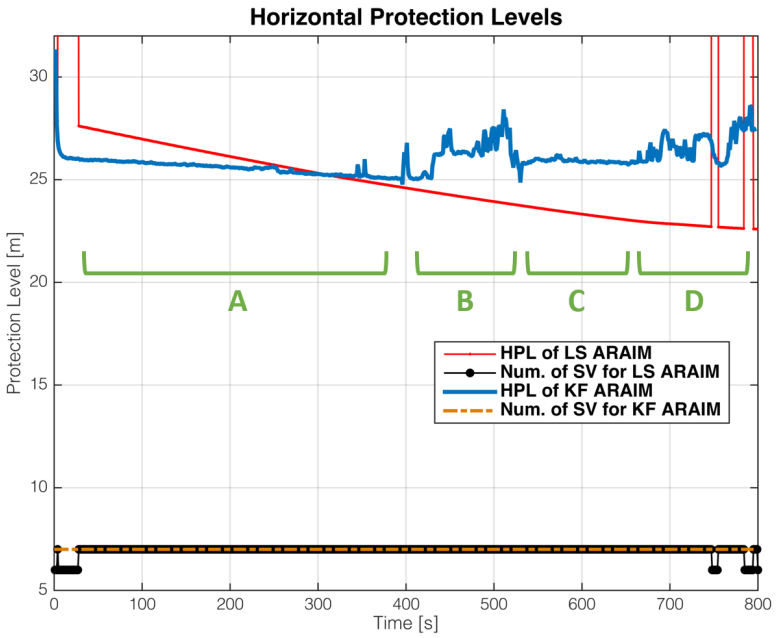


Figure 4.11: HPL of the real data collection evaluated by KF-based ARAIM and LS-based ARAIM

are also highlighted. The route starts with a test run in a large courtyard with low building nearby - marked as segment A. Segment B sees the car run in a 6-lane street with trees and apartment buildings on both sides of the street. The car stops at segment C for about 100 seconds, due to traffic light. For the remaining segment D, the car continues to move along the 6-lane street. The HPL for the whole period is depicted in Figure 4.11, also highlighting the corresponding zones of interest.

The positioning results for segment A are shown in Figure 4.12, where the calculated positions are the yellow squares, the red line is the reference track. It can be observed that the estimated positions are very close to the reference track (less than 2 m). The HPL for this period, as reported in Figure 4.11, is about 25-26m and very smooth thanks to the open sky and low buildings. Segment B's positioning results are depicted in Figure 4.13, which shows similar quality to Segment A as the results are also very close to the reference track. There are some perturbation probably due to the multipath from high buildings on both sides of the street. The HPL of this segment is also higher and noisier, up to 28 m, confirming the degradation of the signal quality in this part of the route.

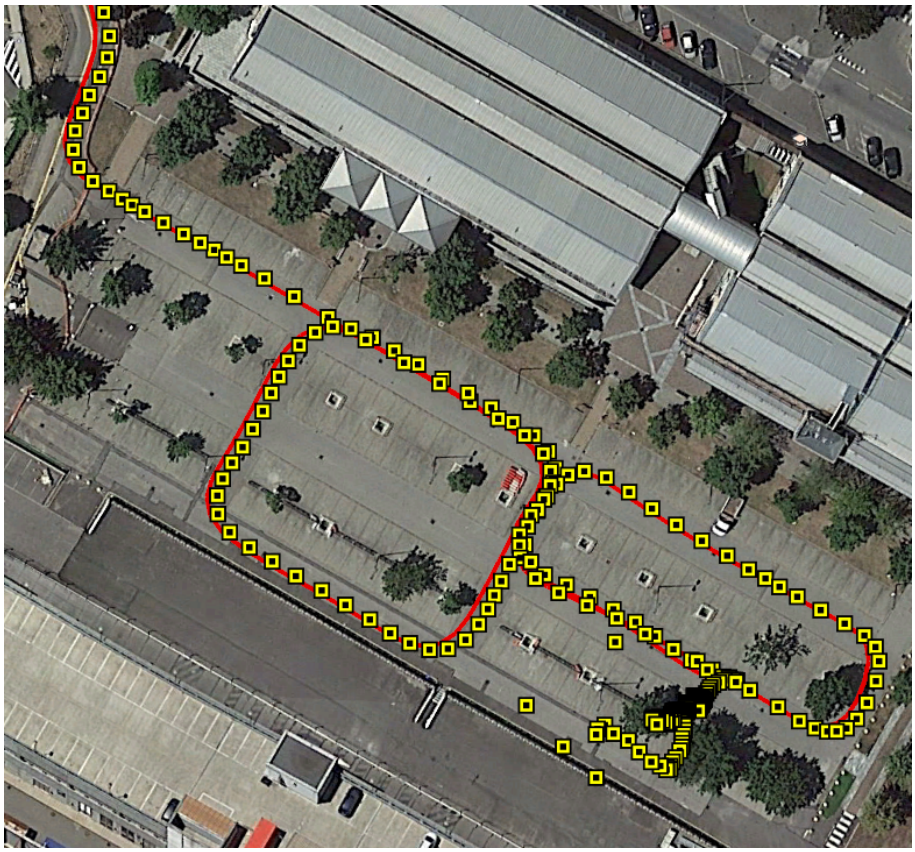


Figure 4.12: Positioning results of segment A

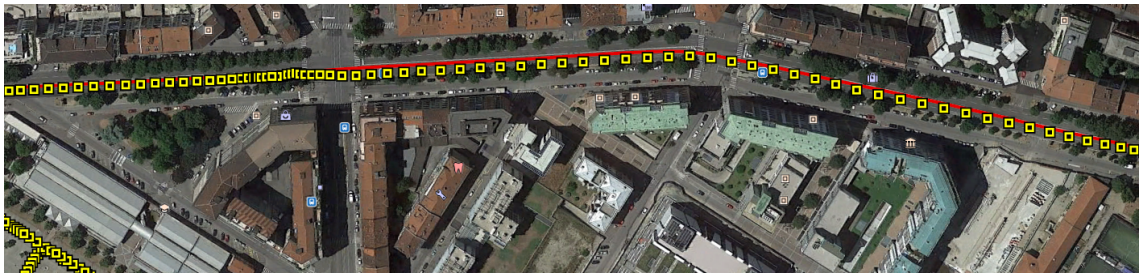


Figure 4.13: Positioning results of segment B

At the stop at segment C, the HPL is lower (around 26 m) since the car stops next to a park, thus less multipath is visible as in segment B. Following up, the positioning results of segment D are shown in Figure 4.14, which consists of a large crossing (marked E) and a roundabout (marked F) besides the normal street similar to segment B. It can be observed in Figure 4.15, which zoomed in on Segment D, that the HPL lowers when the car passes through these points, while remains higher

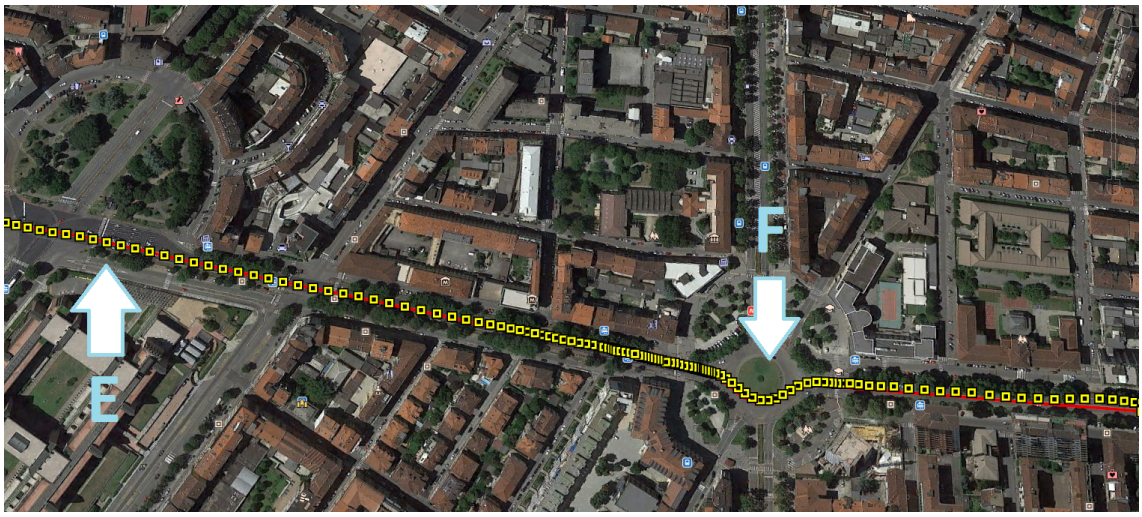


Figure 4.14: Positioning results of segment D

and noisy in other parts, similar to segment B.

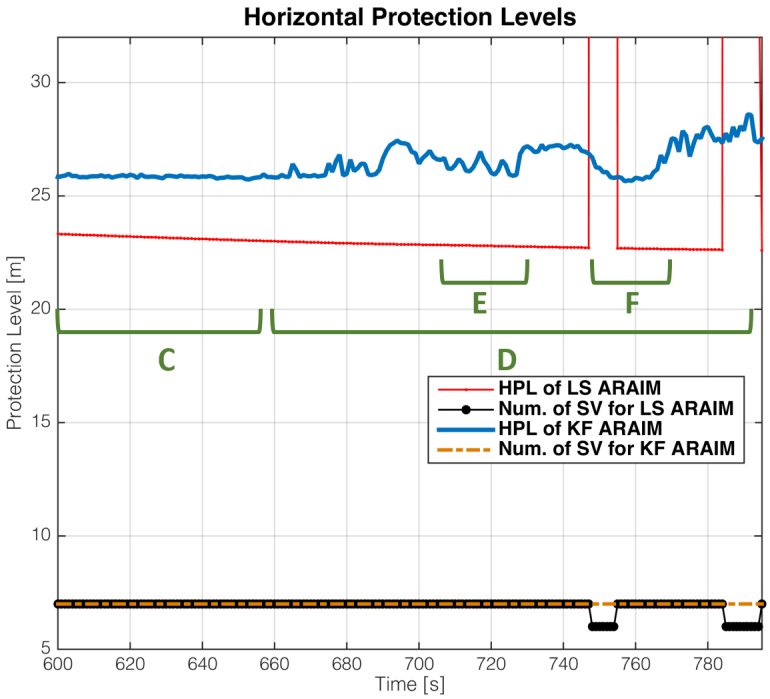


Figure 4.15: HPL of the real data collection for segment D (zoomed in)

For comparison, the same processing was done using the conventional LS-based

ARAIM and the HPL is reported also in Figure 4.11. Unlike KF-ARAIM, the HPL evaluated by LS-ARAIM is very smooth and ranges from 26 to 28 m for the most part (which is similar to KF-ARAIM). However, there are some exclusion occurred at the beginning and near the end of the period, which excessively increased the HPL, due to degraded geometry in a situation where the number of satellites is already low. Note that for KF-ARAIM, no satellite was excluded during the whole operation.

Overall, the HPL of over 25 m should be sufficient to protect the user from integrity risk. More importantly, the HPL calculated by the proposed algorithm can follow the change in the measurement quality as the car moves through different surrounding. The number of satellites is also maintained throughout the duration, which is also important due to the low visibility of the urban environment. Therefore, it can be concluded that the proposed algorithm is suitable for integrity monitoring in urban scenario.

4.4.4 Computation time

The computational complexity of the proposed algorithm depends heavily on the number of visible satellites N_{sat}, considering the SS approach needs to evaluate all the subset solutions. Assuming the maximum number of concurrent faulty satellites (calculated in Step 2 of the algorithm) to be 2, the total number of subsets (including the all-in-view set) can be expressed as:

$$\begin{pmatrix} N_{sat} \\ 1 \end{pmatrix} + \begin{pmatrix} N_{sat} \\ 2 \end{pmatrix} + 1 = N_{sat} + \frac{N_{sat}\left(N_{sat}-1\right)}{2} + 1 \tag{4.27}$$

Under the presence of faulty satellites, the algorithm would attempt the exclusion process, which essentially repeats the SS tests on reduced (candidate) sets of satellites.

The computation time of the proposed KF-based ARAIM algorithm is measured on a MATLAB implementation of the algorithm, deployed on a laptop PC with Intel® Core™i7-4710HQ CPU @ 2.50GHZ and 16GB DDR3 RAM. It is also compared with the LS-based counterpart. The results are depicted in Figure 4.16

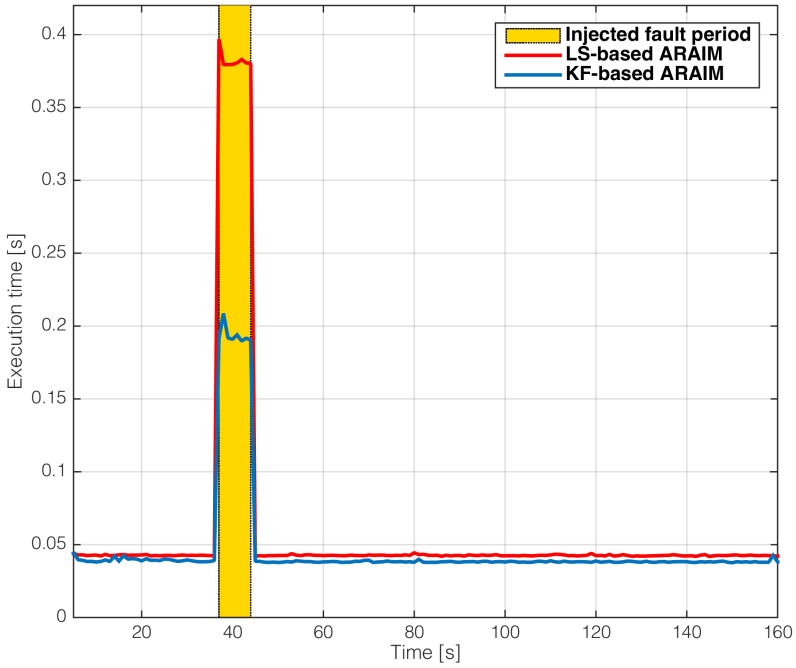


Figure 4.16: Computation time of both algorithm

for the simulated data set, and in Figure 4.17 for the real data set. The implementation for both algorithm has been optimized in term of design, memory allocation and programming technique, to ensure fair comparison.

It can be observed on Figure 4.16 that the execution time of the proposed KF-based ARAIM is almost identical to the LS counterpart, and noticeably lower during the manual fault period. This is due to the different noise model between the two algorithm. The noise level of the Kuusniemi's model allows higher tolerance to solution separation in the KF-based ARAIM, resulting in less number of candidate for the exclusion step than the case of the LS-based algorithm. As a result, the LS-based ARAIM takes more time to go through all the candidate before choosing the correct faulty satellites to exclude.

The same observation can be made on Figure 4.17 for the real data case. While the execution time for both algorithm is nearly identical for most of the experiment period, there are several periods of false exclusions for the LS-based algorithm, as already shown in Figure 4.11. For these period, the execution time is noticeably

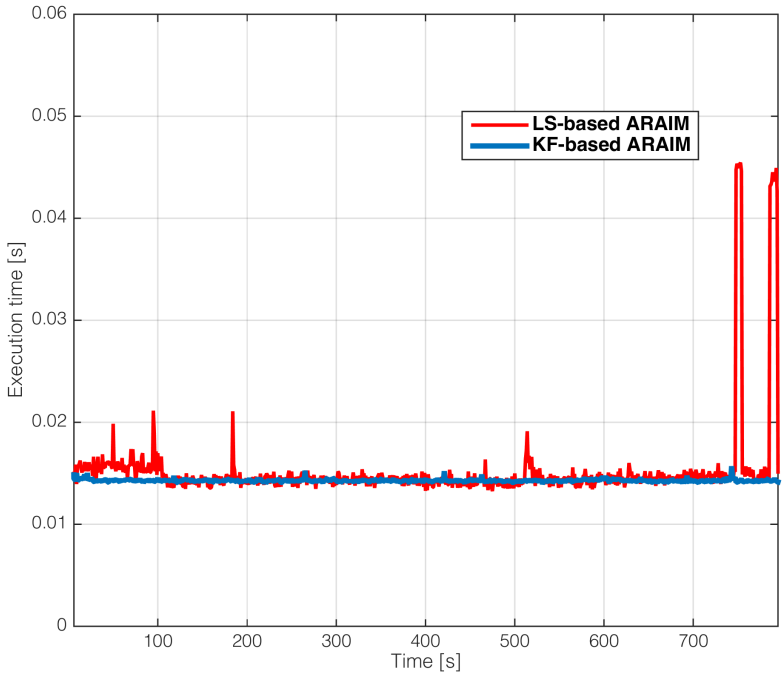


Figure 4.17: Computation time of both algorithm

higher, since the algorithm has to process also the exclusion candidates. The proposed KF-based ARAIM, however, does not (falsely) detect faulty satellites during these period, thus maintain the same execution time thoroughly.

It is important to note that, the execution time reported here is measured on a MATLAB implementation, thus does not reflect actual runtime of the algorithms in real-time implementations. This experiment only compares the computation time relatively between the proposed algorithm and the closely-related LS-based ARAIM algorithm. Further real-time comparison will be in the scope of future work.

4.5 Conclusion

In this Chapter, a KF-based ARAIM algorithm was proposed, following the SS approach as an integrity monitoring method for urban environment. The method is an adaptation of ARAIM algorithm from LS to KF, also adopting the noise model based on C/N_0 [70]. The method was implemented in MATLAB and tested on

simulation data, which simulates the multipath effect that's often encountered in urban scenario. The results confirms the validity of the method, good consistency under the effect of multipath. The resulting Protection Level responses well to the change of the signal quality, and thus can protect the users from environmental-based degradation. Further validation was done using real data collection, obtained in the urban area of Turin, Italy. The results also show that the Protection Level can follow the change of measurement quality as the vehicle goes through different area, while maintaining integrity and availability.

On the other hand, the proposed algorithm was also compared with conventional LS-based ARAIM in both simulation and real data validation. Comparative results show that the proposed algorithm is a more suitable method for urban scenario in term of integrity and availability, especially in the real data validation. While the LS-based ARAIM attempted unnecessary exclusion in several points (leading to excessive PL and potential loss of availability), the porposed KF-based ARAIM maintained the number of satellites thus ensuring the availability during the whole period.

Conclusion and future work

Integrity is the concept about the level of trust an user can put into the navigation result of a positioning system. It has been an important factor in evaluating the performance of navigation systems in the field of civil aviation. Beyond that, the necessity of highly reliable navigation performance in other types of application calls for the adaptation of the integrity concept and the related integrity monitoring techniques to contexts other than civil aviation. This thesis has explored the possibility of adapting the ARAIM algorithm - a new generation of integrity monitoring method originally developed to satisfy the stringent requirement of the LPV-200 procedure - into urban scenario. In fact, the adaptation process has to take into account all the notable difference between the condition of airports area (where the ARAIM algorithm usually operates) and the aiming urban scenario.

Chapter 3 focuses on outlining the necessary changes, as well as finding the solution for the adaptation problem. From an in-depth study of the original ARAIM algorithm, it is necessary to change the integrity and continuity risk requirements, as well as removing the negligible nominal bias. Both of these changes are largely due to the potential lower requirement of urban application with respect to the initial aviation procedure. The third point to change is also the most important, which is the necessity for a reliable noise model. Considering the unpredictable nature of multipath in urban environment, it is important for the noise model to be able to follow and adapt to the sudden change of the signal and measurement quality. Comparative study on several noise models using simulated signal has led to the choice of the C/N_0-based noise model [70], which is very responsive to the change of signal quality. It is also worth mentioning that, a MATLAB-based signal generator has been developed for the experiments in this thesis, which can specifically simulates signal affected by multipath effect prominently encountered

in urban areas. Using the standardized multipath model LMMC [76], the signal generator can be considered a reliable testing tool for urban-related experiments.

From the results of Chapter 3, Chapter 4 proposes the KF-based ARAIM algorithm, which adapts the original LS-based ARAIM onto a KF model, along with the C/N_0-based noise model. Experiments using simulated data, which resembles the multipath effects usually present in urban areas, validates the proposed method. The method shows good consistency, availability and responsiveness in the presence of multipath. The computed PL also reflects the changes of signal quality throughout the simulation period, while maintaining integrity and availability. The same observation can be made when experimenting with a real data set collected in the streets of Turin, Italy. The HPL fully reflects the changes of environment and measurement quality when the vehicle pass through different areas.

The proposed algorithm is also compared with the conventional ARAIM with its original aviation settings in both simulation and real data validation. The comparisons clearly show the better suitability of the proposed method for integrity monitoring in urban context, especially in the real data experiment. While the conventional LS-based ARAIM unnecessary exclusion in several points (leading to excessive PL and undesired loss of availability), the proposed algorithm maintained the number of satellites and ensuring the availability during the whole period.

For future work, it is possible to further improve the proposed KF-based ARAIM in several ways:

- A different method to find a consistent set of satellites can be considered for the exclusion step. Currently, the method employs an exhaustive search, going through all found candidates and stops when a consistent set is found, prioritizing exclusion lowest number of satellites as possible. It is possible to follow other approaches, such as to search in a heuristic way using the SS test results, since such results are usually higher when the correct satellites are excluded. This will reduce the number of candidates to process, and thus improve the computation time.

- The noise model can be modified in an adaptive way, instead of fixing the parameters values as the current setup. This will allows even better adaptation to the changing environment as the vehicle moves around.

- The algorithm can be extended to accommodate external sensors for the positioning procedure, such as inertial sensors. In this case, the PL equations and the SS tests should be extended to take into account the potential faults of the sensors as well, with corresponding integrity and continuity risks to be allocated accordingly.

On the other hand, real time implementation and testing of the algorithm can be considered in parallel with the aforementioned improvement. An option would be deploying the algorithm into GNSS software receivers running on PC or specialized hardware. Under such circumstances, further designing maybe necessary to port the algorithm to the desired hardware system.

Appendix A

Covariance of fault correction from all-in-view position

For discrete Kalman filter, it has been defined:

$$\boldsymbol{e_k} = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(0)} \tag{A.1}$$

where $\boldsymbol{x}_k$ is the true state vector, $\hat{\boldsymbol{x}}_k^{(0)}$ is the all-in-view estimated state vector $\boldsymbol{e_k}$ is assumed to have zero mean and covariance $\boldsymbol{P}_k^{(0)}$.

From this, we can write for each fault hypothesis q:

$$\begin{aligned} \Delta\hat{\boldsymbol{x}}_k^{(q)} &= \hat{\boldsymbol{x}}_k^{(q)} - \hat{\boldsymbol{x}}_k^{(0)} \\ &= (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(0)}) - (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(q)}) = \boldsymbol{e}_k - \boldsymbol{e}_k^{(q)} \end{aligned} \tag{A.2}$$

Thus, the covariance matrix of $\Delta\hat{\boldsymbol{x}}_k^{(q)}$ can be defined as:

$$\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)} = \boldsymbol{P}_k^{(q)} + \boldsymbol{P}_k^{(0)} - 2cov(\boldsymbol{e}_k, \boldsymbol{e}_k^{(q)}) \tag{A.3}$$

While $\boldsymbol{P}_k^{(0)}$ and $\boldsymbol{P}_k^{(q)}$ can be calculated using (1.15) and (4.4), respectively, the covariance between $\boldsymbol{e}_k$ and $\boldsymbol{e}_k^{(q)}$ can be expressed as:

$$
\begin{aligned}
cov(\boldsymbol{e}_k, \boldsymbol{e}_k^{(q)}) &= E\left[\boldsymbol{e}_k \boldsymbol{e}_k^{(q)T}\right] \\
&= E\left\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(q)}\right)^T\right\} \\
&= E\left\{\left[\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-\right) - \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{x}_k^-\right)\right]\right. \\
&\quad \left.\left[\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-\right) - \boldsymbol{K}^{(q)}\left(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-\right)\right]^T\right\}
\end{aligned} \tag{A.4}
$$

Expanding (A.4), note that $\boldsymbol{e}_k$ and $\boldsymbol{e}_k^{(q)}$ has zero mean, (A.4) becomes:

$$
\begin{aligned}
cov(\boldsymbol{e}_k, \boldsymbol{e}_k^{(q)}) &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k^{(q)} \boldsymbol{H}_k)^T \\
&\quad + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^{(q)T}
\end{aligned} \tag{A.5}
$$

Replacing (A.5) into (A.3), $\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)}$ can be written as:

$$
\begin{aligned}
&\boldsymbol{P}_{\Delta\hat{\boldsymbol{x}}_k}^{(q)} = \boldsymbol{P}_k^{(q)} + \boldsymbol{P}_k^{(0)} \\
&- 2\left((\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k^{(q)} \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^{(q)T}\right)
\end{aligned} \tag{A.6}
$$

Appendix B

Protection Level equation

The derivation for the Protection Level (PL) equation follows a similar approach to one described in [134] and [60], considering the KF model. In [60], the integrity risk is defined to cover both fault detection and fault exclusion.

For fault detection, the event of integrity risk occurrence can be defined when there is a large error (larger than the PL) but no fault was detected. For each fault hypothesis q, the integrity risk for the s-coordinate is defined as a joint probability:

$$P\left(|\boldsymbol{e}_{k,s}| > PL, \left|\Delta\boldsymbol{x}_{k,s}^{(q)}\right| < T_{k,s}^{(q)}\right) \tag{B.1}$$

where $\boldsymbol{e}_{k,s} = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^{(0)}$ is the true error of $\hat{\boldsymbol{x}}_k^{(0)}$.

For simplicity, the derivation in this section will omit the notation of k, since all the Kalman filter-related vectors and matrices are all correspond to discrete time t_k. That said, (B.1) can be rewritten as:

$$P\left(|\boldsymbol{e}_s| > PL, \left|\Delta\boldsymbol{x}_s^{(q)}\right| < T_s^{(q)}\right) \tag{B.2}$$

Defining $\boldsymbol{e}^- = \boldsymbol{x} - \hat{\boldsymbol{x}}^-$ as the error of the apriori state estimation $\hat{\boldsymbol{x}}^-$, the difference between the updated and true state vector can be expressed as:

$$\begin{aligned}\hat{\boldsymbol{x}}^{(q)} - \boldsymbol{x} &= \hat{\boldsymbol{x}}^{-} - \boldsymbol{x} + \boldsymbol{K}^{(q)}(\boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}^{-}) \\ &= \hat{\boldsymbol{x}}^{-} - \boldsymbol{x} + \boldsymbol{K}^{(q)}(\boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} - \boldsymbol{H}\hat{\boldsymbol{x}}^{-}) \\ &= (\boldsymbol{K}^{(q)}\boldsymbol{H} - \boldsymbol{I})\boldsymbol{e}^{-} + \boldsymbol{K}^{(q)}\boldsymbol{v} \\ &= \boldsymbol{S}^{(q)}\boldsymbol{e}^{-} + \boldsymbol{K}^{(q)}\boldsymbol{v} \end{aligned} \tag{B.3}$$

where

$$\begin{aligned}\boldsymbol{S}^{(q)} &= \boldsymbol{K}^{(q)}\boldsymbol{H} - \boldsymbol{I} \\ \boldsymbol{S}^{(0)} &= \boldsymbol{K}\boldsymbol{H} - \boldsymbol{I} \end{aligned} \tag{B.4}$$

In case of fault, the satellite faults can be modeled as a fault vector $\boldsymbol{b}_{fault} \in \mathbb{R}^{2N_{sat}}$ which only affect certain satellites, can be incorporated into the measurement error vector as:

$$\boldsymbol{v}_{fault} = \boldsymbol{v} + \boldsymbol{b}_{fault} \tag{B.5}$$

where $\boldsymbol{v}$ is the nominal measurement error vector, $\boldsymbol{v}_{fault}$ is the measurement error vector in case of fault. Assume that $\boldsymbol{b}_{fault}$ only affect satellites in fault hypothesis q, so that:

$$\boldsymbol{W}^{(q)}\boldsymbol{b}_{fault} = \boldsymbol{0} \tag{B.6}$$

$$\boldsymbol{W}^{(h)}\boldsymbol{b}_{fault} \neq \boldsymbol{0} \ \forall h \neq q \tag{B.7}$$

$$\tag{B.8}$$

With that assumption, the fault affects the all-in-view state as follows:

$$\begin{aligned}\hat{\boldsymbol{x}}^{(0)} - \boldsymbol{x} &= \boldsymbol{S}^{(q)}\boldsymbol{e}^{-} + \boldsymbol{K}^{(0)}\boldsymbol{v}_{fault} \\ &= \boldsymbol{S}^{(q)}\boldsymbol{e}^{-} + \boldsymbol{K}^{(0)}\boldsymbol{v} + \boldsymbol{K}^{(0)}\boldsymbol{b}_{fault}\end{aligned} \tag{B.9}$$

$$\begin{aligned}\hat{\boldsymbol{x}}^{(q)} - \boldsymbol{x} &= \boldsymbol{S}^{(q)}\boldsymbol{e}^{-} + \boldsymbol{K}^{(q)}\boldsymbol{v}_{fault} \\ &= \boldsymbol{S}^{(q)}\boldsymbol{e}^{-} + \boldsymbol{K}^{(q)}\boldsymbol{v}\end{aligned} \tag{B.10}$$

From (B.1) we have:

$$\begin{aligned}&P\left(|\boldsymbol{e}_{k,s}| > PL, \left|\Delta\boldsymbol{x}_{k,s}^{(q)}\right| < T_{k,s}^{(q)}\right) \\ &= P\left(\left|\hat{\boldsymbol{x}}_s^{(0)} - \boldsymbol{x}_s\right| > PL, \left|\hat{\boldsymbol{x}}_s^{(q)} - \hat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)}\right) \\ &= P\left(\left|\hat{\boldsymbol{x}}_s^{(0)} - \hat{\boldsymbol{x}}_s^{(q)} + \hat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL, \left|\hat{\boldsymbol{x}}_s^{(q)} - \hat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)}\right) \\ &\leq P\left(\left|\hat{\boldsymbol{x}}_s^{(0)} - \hat{\boldsymbol{x}}_s^{(q)}\right| + \left|\hat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL, \left|\hat{\boldsymbol{x}}_s^{(q)} - \hat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)}\right) \\ &\leq P\left(\left|\hat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL - T_s^{(q)}, \left|\hat{\boldsymbol{x}}_s^{(q)} - \hat{\boldsymbol{x}}_s^{(0)}\right| < T_s^{(q)}\right) \\ &\leq P\left(\left|\hat{\boldsymbol{x}}_s^{(q)} - \boldsymbol{x}_s\right| > PL - T_s^{(q)}\right) \\ &= Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right)\end{aligned} \tag{B.11}$$

where $Q(\cdot)$ is the complement of the Gaussian cumulative distribution function, with zero mean and unit variance, and $\sigma_s^{(q)}$ is defined as:

$$\begin{aligned}\sigma_s^{(q)2} &= \left[\boldsymbol{S}^{(q)}\boldsymbol{P}^{-}\boldsymbol{S}^{(q)T} + \boldsymbol{K}^{(q)}\boldsymbol{R}\boldsymbol{K}^{(q)T}\right]_{s,s} \\ &= \left[\boldsymbol{P}_{alt}^{(q)}\right]_{s,s}\end{aligned} \tag{B.12}$$

Let $p_{fault}^{(q)}$ be the fault probability for fault hypothesis q, the boundary for the integrity risk in case of detection can be written as:

$$
\begin{aligned}
P_{HMI,D} &= \sum_{q=0}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \\
&= p_{fault}^{(0)} Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \\
&\leq Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right)
\end{aligned} \tag{B.13}
$$

The first term of (B.13) represents the event of no detection under fault-free condition, while the second terms represents the missed detection when subset q is faulty.

In case of exclusion, the event of integrity risk occurrence can be defined when there is a large error (larger than the PL), a fault is detected and an exclusion candidate j is actually excluded. Note that for a subset j to be excluded, all fault hypothesis q on the remaining satellite set must pass the solution separation test. For each exclusion candidate j and each fault hypothesis q, the integrity risk for exclusion along the s-coordinate is defined as:

$$
P\left(\left|\boldsymbol{e}_s^{(j)}\right| > PL, \left|\Delta\boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}, \left|\Delta\boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)}\right) \tag{B.14}
$$

where $\Delta\boldsymbol{x}_s^{(j,q)} = \boldsymbol{x}_s^{(j,q)} - \boldsymbol{x}_s^{(j)}$ is the solution separation for fault hypothesis q after excluding candidate j, i.e. considering the remaining satellites as all in view; $\boldsymbol{e}_s^{(j)}$ is the positioning error after having removed subset j; $T_s^{(j,q)}$ is the test threshold for the corresponding solution separation. Notice that subset j is detected to be a fault due to $\left|\Delta\boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}$, then is removed because $\left|\Delta\boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)}$. Following similar derivation as for detection case, (B.14) can be written as:

$$
\begin{aligned}
&P\left(\left|\boldsymbol{e}_s^{(j)}\right| > PL, \left|\Delta\boldsymbol{x}_s^{(j)}\right| > T_s^{(j)}, \left|\Delta\boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)}\right) \\
&\leq P\left(\left|\boldsymbol{e}_s^{(j)}\right| > PL, \left|\Delta\boldsymbol{x}_s^{(j,q)}\right| < T_s^{(j,q)}\right) \\
&= P\left(\left|\hat{\boldsymbol{x}}_s^{(j)} - \boldsymbol{x}_s\right| > PL, \left|\hat{\boldsymbol{x}}_s^{(j,q)} - \boldsymbol{x}_s^{(j)}\right| < T_s^{(j,q)}\right) \\
&\leq P\left(\left|\hat{\boldsymbol{x}}_s^{(j,q)} - \boldsymbol{x}_s\right| > PL - T_s^{(j,q)}\right) \\
&= Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)
\end{aligned} \tag{B.15}
$$

Considering all exclusion hypothesis j and all fault hypothesis q, the integrity risk in case of exclusion can be expressed as:

$$
\begin{aligned}
P_{HMI,E} &= \sum_{q=0}^{N_{fault}} \sum_{j=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \\
&= p_{fault}^{(0)} \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) \\
&+ \sum_{q=1}^{N_{fault}} \sum_{j=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \\
&= p_{fault}^{(0)} \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) \\
&+ \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right) \\
&+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \\
&\leq \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right) \\
&+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right)
\end{aligned} \tag{B.16}
$$

The first term in the last line of (B.16) represents wrong exclusion of subset j under fault free condition. The second term represents the case where $j = q, q > 0$

which is the correct exclusion case. Note that with $j = q$, subset (j, q) simplifies to (q) since they are the same subset. The third term represents the wrong exclusion case with $j \neq q$, i.e. excluding subset j while the fault is in fact in subset q.

From (B.13) and (B.16), the PL equation for both detection and exclusion can be formally expressed as:

$$\begin{aligned} P_{HMI} &= P_{HMI,D} + P_{HMI,E} \\ &= Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \\ &+ \sum_{j=1}^{N_{fault}} Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right) \\ &+ \sum_{q=1}^{N_{fault}} \sum_{\substack{j=1 \\ j\neq q}}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \end{aligned} \tag{B.17}$$

(B.17) is the formal equation to evaluate the PL, which takes into account the integrity risks from both detection and exclusion. However, this form should be considered as a predictive form to calculate PL before either the events of detection and exclusion occurs. In implementation, only certain terms of the equation should be used in certain cases, to avoid suboptimal allocation of the integrity risk and thus ensure sufficient availability. Therefore, in case of simple detection without exclusion implementation, the practical equation should eliminate the exclusion terms (since there are no exclusion candidate to be considered):

$$P_{HMI,0} = Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \tag{B.18}$$

In case of fault detection and exclusion of a subset j, the missed detection terms (the first two terms) of (B.17) should be eliminated since at least a fault was detected on subset j, and the exclusion terms should only consider the excluded subset j:

$$P_{HMI,j} = Q\left(\frac{PL}{\sigma_s^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{PL}{\sigma_s^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(j,q)}}{\sigma_s^{(j,q)}}\right) \tag{B.19}$$

Equation (B.17) and the other forms of (B.18), (B.19) are the generalized equations for PL along the direction of s coordinate. From this, the specific equation for VPL and HPL can be defined for each case accordingly. In case of no detection, the equation for VPL can be written as:

$$P_{HMI,VERT,0} = Q\left(\frac{VPL}{\sigma_3^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{VPL - T_3^{(q)}}{\sigma_3^{(q)}}\right) \tag{B.20}$$

The equations for HPL can be defined as:

$$\frac{1}{2} P_{HMI,HOR,0} = Q\left(\frac{HPL_1}{\sigma_1^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(q)}}{\sigma_1^{(q)}}\right) \tag{B.21}$$

$$\frac{1}{2} P_{HMI,HOR,0} = Q\left(\frac{HPL_2}{\sigma_2^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_2^{(q)}}{\sigma_2^{(q)}}\right) \tag{B.22}$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{B.23}$$

In case of exclusion, the equation for VPL can be defined as:

$$P_{HMI,VERT,j} = Q\left(\frac{VPL}{\sigma_3^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL}{\sigma_3^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{VPL - T_3^{(j,q)}}{\sigma_3^{(j,q)}}\right) \tag{B.24}$$

Similarly, the equations for HPL can be defined as:

$$\frac{1}{2}P_{HMI,HOR,j} = Q\left(\frac{HPL_1}{\sigma_1^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_1}{\sigma_1^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_1 - T_1^{(j,q)}}{\sigma_1^{(j,q)}}\right) \tag{B.25}$$

$$\frac{1}{2}P_{HMI,HOR,j} = Q\left(\frac{HPL_2}{\sigma_2^{(j)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_2}{\sigma_2^{(q)}}\right) + \sum_{q=1}^{N_{fault,j}} p_{fault}^{(q)} Q\left(\frac{HPL_2 - T_2^{(j,q)}}{\sigma_2^{(j,q)}}\right) \tag{B.26}$$

$$HPL = \sqrt{HPL_1^2 + HPL_2^2} \tag{B.27}$$

Appendix C

Solution method for the PL equation

Let $P_{ex}(PL)$ represents the right side of (B.18):

$$P_{ex}(PL) = Q\left(\frac{PL}{\sigma_s^{(0)}}\right) + \sum_{q=1}^{N_{fault}} p_{fault}^{(q)} Q\left(\frac{PL - T_s^{(q)}}{\sigma_s^{(q)}}\right) \tag{C.1}$$

Let $P_{HMI,s}$ be the integrity risk P_{HMI} allocated to the s-coordinate. Thus, the PL equation becomes

$$P_{ex}(PL) = P_{HMI,s} \tag{C.2}$$

Denote PL_{sol} as the solution of (C.2). While it is not trivial to solve (C.2) analytically, in this work, the equation is solved by linear approximation. The starting point is to find initial PL such that:

$$\begin{aligned} PL_{up,init} &> PL_{true} \\ PL_{low,init} &< PL_{true} \end{aligned} \tag{C.3}$$

For Multi-Hypothesis Solution Separation (MHSS) RAIM [101], the PL is calculated as:

$$PL_{sol} = \max_{q} \{PL_q\} \tag{C.4}$$

where PL_q is the PL imposed by fault hypotheses q (each corresponds to a fault hypothesis q). In fact, while PL_q can be computed by optimally allocating $P_{HMI,s}$ to each fault hypothesis [17], denoted $P_{HMI,s}^{(q)}$. Interested reader can refer to [101, 17] for this approach. In this work, (C.4) was exploited to calculate the initial $PL_{low,init}$ and $PL_{up,init}$ using different allocation of $P_{HMI,s}$. Note that $Q(\cdot)$ is a monotonically decreasing function, thus overly allocated $P_{HMI,s}^{(q)}$ will yield PL_q lower than PL_{true}, such as putting whole integrity budget for each fault hypothesis:

$$PL_{low,init} = \max \left\{ \begin{array}{c} Q^{-1}\left(\frac{P_{HMI,s}}{2}\sigma_{k,s}^{(0)}\right), \\ \max_q Q^{-1}\left(\frac{P_{HMI,s}}{p_{fault}^{(q)}}\sigma_{k,s}^{(q)} + T_{k,s}^{(q)}\right) \end{array} \right\} \tag{C.5}$$

In contrast, suboptimal $P_{HMI,s}^{(q)}$ will produce PL_q larger than PL_{true}, such as evenly allocate the integrity budget to all fault hypothesis plus all-in-view:

$$PL_{up,init} = \max \left\{ \begin{array}{c} Q^{-1}\left(\frac{P_{HMI,s}}{2\left(N_{fault}+1\right)}\sigma_{k,s}^{(0)}\right), \\ \max_q Q^{-1}\left(\frac{P_{HMI,s}}{p_{fault}^{(q)}\left(N_{fault}+1\right)}\sigma_{k,s}^{(q)} + T_{k,s}^{(q)}\right) \end{array} \right\} \tag{C.6}$$

By using linear interpolation on $P_{ex}(PL)$ between $PL_{up,init}$ and $PL_{low,init}$, the result is an approximation of PL_{sol}. Since $P_{ex}(PL)$ is convex and monotonically decreasing, this approximation can be considered a upper bound for PL_{sol}:

$$\begin{aligned} PL_{approx,up} = {} & PL_{low,init} + \left(P_{HMI,s} - P_{ex}\left(PL_{low,init}\right)\right) \\ & \times \frac{PL_{up,init} - PL_{low,init}}{P_{ex}\left(PL_{up,init}\right) - P_{ex}\left(PL_{low,init}\right)} \end{aligned} \tag{C.7}$$

Similarly, linear interpolation can be done on $\log\left(P_{ex}(PL)\right)$ to obtain a lower bound for PL_{sol}, since $\log\left(P_{ex}(PL)\right)$ is concave:

$$\begin{aligned} PL_{approx,low} = & PL_{low,init} \\ & + \left(\log\left(P_{HMI,s}\right) - \log\left(P_{ex}\left(PL_{low,init}\right)\right)\right) \\ & \times \frac{PL_{up,init} - PL_{low,init}}{\log\left(P_{ex}\left(PL_{up,init}\right)\right) - \log\left(P_{ex}\left(PL_{low,init}\right)\right)} \end{aligned} \tag{C.8}$$

Then, PL_{sol} can be approximate as the median of $PL_{approx,low}$ and $PL_{approx,up}$, or one more iteration of linear interpolation can be execute between $PL_{approx,low}$ and $PL_{approx,up}$ should more accurate approximation be desired.

Bibliography

[1] Shaik Fayaz Ahamed, G Sasibhushana Rao, and L Ganesh. "Fast Acquisition of GPS Signal Using FFT Decomposition". In: *Procedia Computer Science* 87 (2016), pp. 190–197.

[2] L. Alfonsi et al. "Analysis of the regional ionosphere at low latitudes in support of the Biomass ESA mission". In: *IEEE Transactions on Geoscience and Remote sensing* (2018), pp. 1–13. DOI: 10.1109/TGRS.2018.2838321.

[3] Salos Andrés and Carlos Daniel. "Integrity monitoring applied to the reception of GNSS signals in urban environments". PhD thesis. University of Toulouse, July 2012.

[4] Antonio Angrisano et al. "Assessment of NeQuick ionospheric model for Galileo single-frequency users". In: *Acta Geophysica* 61.6 (2013), pp. 1457–1476.

[5] J. Angus. "RAIM with Multiple Failures". In: *Navigation* 53.4 (2006), pp. 249–257.

[6] *Annex 10, Amendment 84. Aeronautical Telecommunications - Radio Navigation Aids*. Standard. ICAO, Nov. 2009.

[7] *Annex 10, Amendment 85. Convention on International Civil Aviation, Aeronautical Telecommunications - Radio Navigation Aids*. Standard. ICAO, July 2006.

[8] *Annex 6, Amendment 31. Air Operators and Approved Maintenance Organizations*. Standard. ICAO, July 2007.

[9] P Axelrad and RG Brown. "GPS navigation algorithms". In: *Global Positioning System: Theory and applications*. 1 (1996), pp. 409–433.

[10] B. W. Parkinson and J. J. Spilker. *Chapter 5: Receiver Autonomous Integrity Monitoring Global Positioning System: Theory and Applications*. American Institute of Aeronautics and Astronautics, USA, 1996.

[11] Asghar Tabatabaei Balaei, Beatrice Motella, and Andrew Dempster. "A preventative approach to mitigating CW interference in GPS receivers". In: *GPS solutions* 12.3 (2008), pp. 199–209.

[12] Asghar Tabatabaei Balaei, Beatrice Motella, and Andrew G Dempster. "GPS interference detected in Sydney-Australia". In: *Proceedings of the IGNSS Conference*. 2007, pp. 74–76.

[13] TW Beech, MA Martinez-Olague, and J Cosmen-Schortmann. "Integrity: A key enabler for liability critical applications". In: *Proceedings of the US ION 61st Annual Meeting, 27–29 June, Cambridge, MA*. 2005, pp. 1–10.

[14] John W Betz. "Effect of narrowband interference on GPS code tracking accuracy". In: *2000: Navigating into the New Millennium* (2000), pp. 16–27.

[15] S. Bhattacharyya and D. Gebre-Egziabher. "Kalman filter based RAIM for GNSS receivers". In: *IEEE Transactions on Aerospace and Electronic Systems* 51.3 (July 2015), pp. 2444–2459. ISSN: 0018-9251. DOI: 10.1109/TAES.2015.130585.

[16] T. Binjammaz, A. Al-Bayatti, and A. Al-Hargan. "GPS integrity monitoring for an intelligent transport system". In: *2013 10th Workshop on Positioning, Navigation and Communication (WPNC)*. Mar. 2013, pp. 1–6. DOI: 10.1109/WPNC.2013.6533268.

[17] J. Blanch, T. Walter, and P. Enge. "RAIM with Optimal Integrity and Continuity Allocations Under Multiple Failures". In: *IEEE Transactions on Aerospace and Electronic Systems* 46.3 (July 2010), pp. 1235–1247. ISSN: 0018-9251. DOI: 10.1109/TAES.2010.5545186.

[18] Juan Blanch et al. "Advanced RAIM user Algorithm Description: Integrity Support Message". In: *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*. Sept. 17, 2012, pp. 2828–2849.

[19] Juan Blanch et al. “An Optimized Multiple Hypothesis RAIM Algorithm for Vertical Guidance”. In: *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*. Sept. 2007.

[20] Juan Blanch et al. “Critical Elements for a Multi-Constellation Advanced RAIM”. In: *Navigation* 60.1 (2013), pp. 53–69. ISSN: 2161-4296. DOI: 10.1002/navi.29. URL: http://dx.doi.org/10.1002/navi.29.

[21] D. Borio, L. Camoriano, and L. Lo Presti. “Two-Pole and Multi-Pole Notch Filters: A Computationally Effective Solution for GNSS Interference Detection and Mitigation”. In: *IEEE Systems Journal* 2.1 (Mar. 2008), pp. 38–47. ISSN: 1932-8184. DOI: 10.1109/JSYST.2007.914780.

[22] Kai Borre et al. “Carrier and code tracking”. In: *A software-defined GPS and GALILEO receiver: a single-frequency approach* (2007), pp. 87–108.

[23] Kai Borre et al. “Galileo signal”. In: *A software-defined GPS and GALILEO receiver: a single-frequency approach* (2007), pp. 31–52.

[24] K Borre et al. *A software-defined GPS and Galileo receiver: a single frequency approach*. 2007.

[25] Michael S Braasch. “Multipath effects”. In: *Global Positioning System: Theory and Applications*. 1 (1996), pp. 547–568.

[26] Joe Bradbury. “Prediction of urban GNSS availability and signal degradation using virtual reality city models”. In: *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*. Institute of Navigation Publications. Sept. 2007, pp. 2696–2706.

[27] R Grover Brown. “Solution of the Two-Failure GPS RAIM Problem Under Worst-Case Bias Conditions: Parity Space Approach”. In: *Navigation* 44.4 (1997), pp. 425–431.

[28] R. Grover Brown. “A Baseline GPS RAIM Scheme and note on the equivalence of three RAIM methods”. In: *NAVIGATION* 39.3 (1992). [CrossRef], pp. 301–316. DOI: 10.1002/j.2161-4296.1992.tb02278.x. URL: http://dx.doi.org/10.1002/j.2161-4296.1992.tb02278.x.

[29] R. GROVER BROWN and PAUL W. McBURNEY. "Self-Contained GPS Integrity Check Using Maximum Solution Separation". In: *NAVIGATION* 35.1 (1988). [CrossRef], pp. 41–53. ISSN: 2161-4296. DOI: 10.1002/j.2161-4296.1988.tb00939.x. URL: http://dx.doi.org/10.1002/j.2161-4296.1988.tb00939.x.

[30] Robert Grover Brown and Patrick YC Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Third. Vol. 1. John Wiley and Sons, 1997.

[31] J Blake Bullock et al. "Integration of GPS with other sensors and network assistance". In: *Understanding GPS: Principles and Applications, Mobile Communication Series* (2006), pp. 459–558.

[32] Hank Cabler and Bruce DeCleene. "LPV: New, improved WAAS instrument approach". In: *ION GPS 2002: 15 th International Technical Meeting of the Satellite Division of The Institute of Navigation*. 2002.

[33] J Paul Collins. "Assessment and development of a tropospheric delay model for aircraft users of the global positioning system". In: (1999).

[34] European Commission. "European GNSS (Galileo) Open Service - Ionospheric Correction Algorithm for Galileo Single Frequency Users". In: (1.2 2016).

[35] Rob Conley. "Performance of stand-alone GPS". In: *Understanding GPS: Principle and Applications* (2006), pp. 301–378.

[36] J. Cosmen-Schortmann et al. "Integrity in urban and road environments and its use in liability critical applications". In: *2008 IEEE/ION Position, Location and Navigation Symposium*. May 2008, pp. 972–983. DOI: 10.1109/PLANS.2008.4570071.

[37] G Di Giovanni and SM Radicella. "An analytical model of the electron density profile in the ionosphere". In: *Advances in Space Research* 10.11 (1990), pp. 27–30.

[38] Fabio Dovis, Paolo Mulassano, and Letizia Lo Presti. "A novel algorithm for the code tracking of BOC (n, n) modulated signals". In: *Proceedings of the Institute of Navigation GPS Conference*. Long Beach, CA, Sept. 2005, pp. 152–155.

[39] G. Falco, M. Nicola, and E. Falletti. "Constellation-aware method for computing the covariance matrix of GNSS measurements". In: *2016 European Navigation Conference (ENC)*. May 2016, pp. 1–8. DOI: 10.1109/EURONAV.2016.7530568.

[40] E. Falletti et al. "N-FUELS and SOPRANO: Educational tools for simulation, analysis and processing of satellite navigation signals". In: cited By 0. 2013, pp. 303–308. DOI: 10.1109/FIE.2013.6684836. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84893283733&partnerID=40&md5=5a55a0bd99b3803d5a1a18a3c1fc162c.

[41] Emanuela Falletti, Marco Rao, and Simone Savasta. "The Kalman Filter and Its Applications in GNSS and INS". In: *Handbook of Position Location*. John Wiley and Sons, Inc., 2011, pp. 709–751. ISBN: 9781118104750. DOI: 10.1002/9781118104750.ch22. URL: http://dx.doi.org/10.1002/9781118104750.ch22.

[42] Micaela Troglia Gamba, Mario Nicola, and Emanuela Falletti. "Performance assessment of an ARM-based dual-constellation GNSS software receiver". In: *Localization and GNSS (ICL-GNSS), 2015 International Conference on*. IEEE. 2015, pp. 1–6.

[43] Gianluca Gargiulo et al. "GNSS integrity and protection level computation for vehicular applications". In: *Proceedings of the 16th Ka and Broadband Communications Navigation and Earth Observation Conference*. Vol. 2022. Milan, Italy, Oct. 2010.

[44] Audrey Giremus, Eric Grivel, and Francis Castanie. "Is H-infinity filtering relevant for correlated noises in GPS navigation?" In: *Digital Signal Processing, 2009 16th International Conference on*. IEEE. 2009, pp. 1–6.

[45] *GNSS Based Precision Approach Local Area Augmentation System (LAAS).* Signal-In-Space Interface Control Document RTCA/DO-246:2017. The Radio Technical Commission for Aeronautics, July 2017.

[46] *GPS Wide Area Augmentation System (WAAS) Performance Standard, 1st edition.* Standard. US Department of Transportation and FAA, Dec. 2008.

[47] Paul D Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech house, 2013.

[48] Paul D Groves et al. "A portfolio approach to NLOS and multipath mitigation in dense urban areas". In: *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013).* The Institute of Navigation. Sept. 2013, pp. 3231–3247.

[49] G Hochegger et al. "A family of ionospheric models for different uses". In: *Physics and Chemistry of the Earth, Part C: Solar, Terrestrial & Planetary Science* 25.4 (2000), pp. 307–310.

[50] LT Hsu. "Integration of vector tracking loop and multipath mitigation technique and its assessment". In: *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013).* Institute of Navigation Publications. Sept. 2013, pp. 3263–3278.

[51] Li-Ta Hsu, Paul D Groves, and Shau-Shiun Jan. "Assessment of the multipath mitigation effect of vector tracking in an urban environment". In: *Proceedings of the ION 2013 Pacific PNT Meeting.* Institute of Navigation Publications. Apr. 2013, pp. 498–509.

[52] Todd E Humphreys, Mark L Psiaki, and Paul M Kintner. "GPS carrier tracking loop performance in the presence of ionospheric scintillations". In: *Radionavigation Laboratory Conference Proceedings.* 2005.

[53] Patrick Y Hwang and R Grover Brown. "RAIM-FDE Revisited: A New Breakthrough In Availability Performance With nioRAIM (Novel Integrity-Optimized RAIM)". In: *Navigation* 53.1 (2006), pp. 41–51.

[54] Oscar Isoz et al. “Assessment of GPS L1/Galileo E1 interference monitoring system for the airport environment”. In: *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*. Vol. 3. Inst. of Navigation. 2011, pp. 1920–1930.

[55] ITU. *Propagation data required for the design of Earth-space land mobile telecommunication systems*. 2009.

[56] T. Iwamoto, T. Takewa, and W. Tsujita. “Integrity monitoring of train positioning with GNSS”. In: *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. Apr. 2016, pp. 185–189. DOI: 10.1109/PLANS.2016.7479700.

[57] Chen Jiang, Shu-Bi Zhang, and Qiu-Zhao Zhang. “A New Adaptive H-Infinity Filtering Algorithm for the GPS/INS Integrated Navigation”. In: *Sensors* 16.12 (2016). ISSN: 1424-8220. DOI: 10.3390/s16122127. URL: http://www.mdpi.com/1424-8220/16/12/2127.

[58] Ziyi Jiang and Paul D Groves. “NLOS GPS signal detection using a dual-polarisation antenna”. In: *GPS solutions* 18.1 (2014), pp. 15–26.

[59] Ziyi Jiang et al. “Multi-constellation GNSS multipath mitigation using consistency checking”. In: *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*. The Institute of Navigation. 2011, pp. 3889–3902.

[60] M. Joerger and B. Pervan. “Fault detection and exclusion using solution separation and chi-squared ARAIM”. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.2 (Apr. 2016), pp. 726–742. ISSN: 0018-9251. DOI: 10.1109/TAES.2015.140589.

[61] Mathieu Joerger, Fang-Cheng Chan, and Boris Pervan. “Solution Separation Versus Residual-Based RAIM”. In: *Navigation* 61.4 (2014). NAVI-2014-027, pp. 273–291. ISSN: 2161-4296. DOI: 10.1002/navi.71. URL: http://dx.doi.org/10.1002/navi.71.

[62] Mathieu Joerger and Boris Pervan. “Kalman filter-based Integrity Monitoring Against Sensor Faults”. In: *Journal of Guidance, Control, and Dynamics* 36.2 (2013), pp. 349–361.

[63] Olivier Julien et al. "A new unambiguous BOC(n,n) signal tracking technique". In: *GNSS 2004, European Navigation Conference*. Rotterdam, Netherlands, May 2004. URL: https://hal-enac.archives-ouvertes.fr/hal-01021725.

[64] Marcus Junered et al. "A Modular GPS Remote Sensing Receiver for Small Platforms". In: *International Technical Meeting of the Satellite Division of the Institute of Navigation: 26/09/2006-29/09/2006*. Institute of Navigation, The. 2006, pp. 634–642.

[65] Elliott D Kaplan and Christopher J Hegarty. *Understanding GPS: principles and applications*. Artech House: Norwood, MA, USA, 2005.

[66] Changdon Kee. "Wide area differential GPS". In: *Global Positioning System: Theory and applications*. 2 (1996), pp. 81–115.

[67] Changdon Kee, Bradford W Parkinson, and Penina Axelrad. "Wide area differential GPS". In: *Navigation* 38.2 (1991), pp. 123–145.

[68] John A Klobuchar. "Ionospheric time-delay algorithm for single-frequency GPS users". In: *IEEE Transactions on aerospace and electronic systems* 3 (1987), pp. 325–331.

[69] G Sateesh Kumar, G Sasi Bhushana Rao, and MNVSS Kumar. "GPS signal short-term propagation characteristics modeling in urban areas for precise navigation applications". In: *Positioning* 4.02 (2013), p. 192.

[70] Heidi Kuusniemi. *User-Level Reliability and Quality Monitoring in Satellite-Based Personal Navigation*. Finland: Tampere University of Technology, 2005.

[71] Heidi Kuusniemi and Timo Jokitalo. "Indoor and Weak Signal Navigation (chapter 12). In *GNSS: Applications and Methods*, 1st ed.; S. Gleason and D. Gebre-Egziabher (Eds.)" In: Artech House: Norwood, MA, USA, 2009.

[72] Olivier Le Marchand et al. "Performance Evaluation of Fault Detection Algorithms as Applied to Automotive Localisation". In: *European Navigation Conference - GNSS 2008*. Toulouse, France, Apr. 2008. URL: https://hal.archives-ouvertes.fr/hal-00445170.

[73] Brent M Ledvina et al. “A 12-channel real-time GPS L1 software receiver”. In: *Proc. of the Institute of Navigation National Technical Meeting*. 2003, pp. 22–24.

[74] Young C Lee and Michael P McLaughlin. “Feasibility analysis of RAIM to provide LPV-200 approaches with future GPS”. In: *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*. Sept. 2007, pp. 2898–2910.

[75] Young Chang Lee. “A new improved RAIM method based on the optimally weighted average solution (OWAS) under the assumption of a single fault”. In: *Proceedings of the institute of navigation national technical meeting (NTM), Monterey, CA*. 2006, pp. 18–20.

[76] Andreas Lehner and Alexander Steingass. “A novel channel model for land mobile satellite navigation”. In: *Institute of Navigation Conference ION GNSS*. 2005, pp. 13–16.

[77] Alfred Leick, Lev Rapoport, and Dmitry Tatarnikov. *GPS satellite surveying*. John Wiley & Sons, 2015.

[78] M. Linderoth et al. “Initialization of the Kalman filter without assumptions on the initial state”. In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 4992–4997. DOI: 10.1109/ICRA.2011.5979684.

[79] J. Liu, B. g. Cai, and J. Wang. “Particle swarm optimization for integrity monitoring in BDS/DR based railway train positioning”. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. July 2014, pp. 792–797. DOI: 10.1109/CEC.2014.6900654.

[80] Christophe Macabiau et al. “Impact of ARAIM Nominal Bias Bounding Techniques on Final ARAIM User Performance”. In: *Proceedings of the 2014 International Technical Meeting of The Institute of Navigation (ION ITM 2014*. San Diego, CA, Jan. 2014, pp. 68–77.

[81] Christophe Macabiau et al. "Nominal Bias Analysis for ARAIM User". In: *ION International Technical Meeting 2015*. Institute of Navigation. Dana Point, United States, Jan. 2015. URL: https://hal-enac.archives-ouvertes.fr/hal-01145811.

[82] D. Margaria, E. Falletti, and T. Acarman. "The need for GNSS position integrity and authentication in ITS: Conceptual and practical limitations in urban contexts". In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. June 2014, pp. 1384–1389. DOI: 10.1109/IVS.2014.6856485.

[83] Lukas Marti and Frank van Graas. "Interference detection by means of the software defined radio". In: *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*. Long Beach, CA, Sept. 2004, pp. 99–109.

[84] Gary A. McGraw. "Development of the LAAS accuracy models". In: *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*. Sept. 2000, pp. 1212–1223.

[85] H. Mei, X. Zhan, and X. Zhang. "GNSS vulnerability assessment method based on ARAIM user algorithm". In: *2017 Forum on Cooperative Positioning and Service (CPGPS*. May 2017, pp. 111–115. DOI: 10.1109/CPGPS.2017.8075107.

[86] VB Mendes. "Modeling the neutral-atmospheric propagation delay in radiometric space techniques". In: *UNB geodesy and geomatics engineering technical report* 199 (1999).

[87] Carl Milner. *GNSS for Civil Aviation*. PowerPoint Presentation slides. May 2015.

[88] *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*. Standard RTCA/DO-229D. The Radio Technical Commission for Aeronautics, Dec. 2006.

[89] *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment"*. Standard RTCA/DO-229C. The Radio Technical Commission for Aeronautics, Nov. 2001.

[90] Andrea Molino et al. "N-Gene GNSS software receiver for acquisition and tracking algorithms validation". In: *Signal Processing Conference, 2009 17th European*. IEEE. 2009, pp. 2171–2175.

[91] B. Motella, M. Pini, and F. Dovis. "Investigation on the Effect of Strong Out-of-Band Signals on Global Navigation Satellite Systems Receivers". In: *GPS Solutions* 12.2 (2008), pp. 77–86.

[92] Ahmed El-Mowafy and Balwinder Singh Arora. "The Current ARAIM Availability According to LPV-200 Using GPS and BeiDou in Western Australia". In: *Proceeding of the IGNSS2013 Symposium - The International Global Navigation Systems Society (IGNSS)*. Surfers Paradise, Queensland, Australia: July 2013, pp. 1–16.

[93] Donnay F. Nahimana, Juliette Marais, and Emmanuel Duflos. "A Jump Markov System for Modelling a Realistic Error Model Depending on Satellite Reception State in Urban Environment". In: *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*. Sept. 2007, pp. 1692–1698.

[94] Tu Thi-Thanh Nguyen et al. "An Adaptive Bandwidth Notch Filter for GNSS Narrowband Interference Mitigation". In: *REV Journal on Electronics and Communications* 4.3-4 (2015). DOI: 10.21553/rev-jec.76. URL: http://www.rev-jec.org/index.php/rev-jec/article/view/76.

[95] *NSL: Software Defined Radio GNSS Solutions*. Available at http://www.nsl.eu.com/primo.html. Last visited November 21, 2017.

[96] P. B. Ober. "Integrity according to Bayes". In: *Proceeding of IEEE 2000. Position Location and Navigation Symposium*. Mar. 2000, pp. 325–332. ISBN: 0-7803-5872-4. DOI: 10.1109/PLANS.2000.838321.

[97] V. Palma et al. "Performance evaluation in terms of accuracy positioning of local augmentation and integrity monitoring network for railway sector". In: *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*. May 2014, pp. 394–398. DOI: 10.1109/MetroAeroSpace.2014.6865956.

[98] Thomas Pany. *Navigation signal processing for GNSS software receivers*. Artech House, 2010. ISBN: 9781608070282.

[99] B.W. Parkinson and J.J. Spilker. *Progress in Astronautics and Aeronautics: Global Positioning System: Theory and Applications*. Vol. 2. Washington DC, USA: American Institute of Aeronautics & Astronautics, 1996, pp. 143–164. ISBN: 9781600864209. URL: https://books.google.com/books?id=t0eGFpSwN0wC.

[100] Bradford W Parkinson and Per K Enge. "Differential GPS". In: *Global Positioning System: Theory and applications*. 2 (1996), pp. 3–50.

[101] Boris S. Pervan, Samuel P. Pullen, and Jock R. Christie. "A Multiple Hypothesis Approach to Satellite Navigation Integrity". In: *Navigation* 45.1 (1998), pp. 61–71. ISSN: 2161-4296. DOI: 10.1002/j.2161-4296.1998.tb02372.x. URL: http://dx.doi.org/10.1002/j.2161-4296.1998.tb02372.x.

[102] Mark G Petovello et al. "Architecture and benefits of an advanced GNSS software receiver". In: *Journal of Global Positioning Systems* 7.2 (Dec. 2008), pp. 156–168. DOI: 10.5081/jgps.7.2.156.

[103] *Phase II of the GNSS evolutionary architecture study*. Technical report. FAA GNSS Evolutionary Architecture Study Panel Panel, Feb. 2010.

[104] *Pilot/Controller Glossary*. Standard. FAA, May 2016.

[105] Letizia Lo Presti et al. "Software defined radio technology for GNSS receivers". In: *Metrology for Aerospace (MetroAeroSpace), 2014 IEEE*. IEEE. 2014, pp. 314–319.

[106] Sam Pullen, Todd Walter, and Per Enge. "SBAS and GBAS Integrity for Non-Aviation Users: Moving Away from Specific Risk". In: *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*. San Diego, CA, Jan. 2011, pp. 533–545.

[107] M Venu Gopala Rao and D Venkata Ratnam. "Faster acquisition technique for software-defined GPS receivers". In: *Defence Science Journal* 65.1 (2015), pp. 5–11.

[108] S. J. Roberts et al. "Bayesian approaches to Gaussian mixture modeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (Nov. 1998), pp. 1133–1142. ISSN: 0162-8828. DOI: 10.1109/34.730550.

[109] D. Salós et al. "Receiver Autonomous Integrity Monitoring of GNSS Signals for Electronic Toll Collection". In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (Feb. 2014), pp. 94–103. ISSN: 1524-9050. DOI: 10.1109/TITS.2013.2273829.

[110] J. Santa et al. "Monitoring the Position Integrity in Road Transport Localization Based Services". In: *IEEE Vehicular Technology Conference*. Sept. 2006, pp. 1–5. DOI: 10.1109/VTCF.2006.575.

[111] S. Savasta, L. Lo Presti, and M. Rao. "Interference Mitigation in GNSS Receivers by a Time-Frequency Approach". In: *IEEE Transactions on Aerospace and Electronic Systems* 49.1 (Jan. 2013), pp. 415–438. ISSN: 0018-9251. DOI: 10.1109/TAES.2013.6404112.

[112] G. Seco-Granados, J. A. Fernandez-Rubio, and C. Fernandez-Prades. "ML estimator and hybrid beamformer for multipath and interference mitigation in GNSS receivers". In: *IEEE Transactions on Signal Processing* 53.3 (Mar. 2005), pp. 1194–1208. ISSN: 1053-587X. DOI: 10.1109/TSP.2004.842193.

[113] P Series. "Propagation data and prediction methods required for the design of Earth-space telecommunication systems". In: *Recommendation ITU-R* (2015), pp. 618–12.

[114] S. Soliman, F. Newagy, and I. Hafez. "Complexity reduction for GPS acquisition algorithms using Sparse Fourier Transform". In: *Proceeding of the 2017 34th National Radio Science Conference (NRSC)*. Mar. 2017, pp. 371–379. ISBN: 978-1-5090-4611-9. DOI: 10.1109/NRSC.2017.7893498.

[115] *SPAN-CPT Single Enclosure GNSS/INS Receiver*. Available at https://www.novatel.com/products/span-gnss-inertial-systems/span-combined-systems/span-cpt/. Last visited November 21, 2017.

[116] Alexander Steingass and Andreas Lehner. "Measuring the navigation multipath channel-a statistical analysis". In: *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*. The Institute of Navigation. Long Beach, CA, Sept. 2004, pp. 1157–1164.

[117] Mark A Sturza. "Navigation system integrity monitoring using redundant measurements". In: *Navigation* 35.4 (1988), pp. 483–501.

[118] *System Engineering & Integration*. Interface specification IS-GPS-200H. Global Positioning Systems Directorate, Mar. 2014.

[119] X. Tang et al. "Practical implementation and performance assessment of an Extended Kalman Filter-based signal tracking loop". In: *2013 International Conference on Localization and GNSS (ICL-GNSS)*. June 2013, pp. 1–6. DOI: 10.1109/ICL-GNSS.2013.6577275.

[120] N.D. Thuan, T.H. Tung, and L.P. Letizia. "A software based multi-IF output simulator". In: *Proceedings of the International Symposium of GNSS (IS-GNSS)*. Nov. 16, 2015.

[121] R. Toledo-Moreo, D. Betaille, and F. Peyret. "Lane-Level Integrity Provision for Navigation and Map Matching With GNSS, Dead Reckoning, and Enhanced Maps". In: *IEEE Transactions on Intelligent Transportation Systems* 11.1 (Mar. 2010), pp. 100–112. ISSN: 1524-9050. DOI: 10.1109/TITS.2009.2031625.

[122] Rafael Toledo-Moreo et al. "A study of integrity indicators in outdoor navigation systems for modern road vehicle applications". In: *2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles, IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems. Nice, France*. 2008.

[123] Hieu Trung Tran and Gustavo Belforte. "An ARAIM Adaptation for Kalman Filter". In: *Proceeding of the Workshop on Maritime Navigation and Communication (ComNavi 2017)*. Available at http://porto.polito.it/2695825/1/Belforte_129_C_73r.pdf. Last visited January 10, 2017. Dec. 2017, pp. 27–37. ISBN: 978-604-67-1036-3.

[124] Hieu Trung Tran, Tung Ta Hai, and Letizia Lo Presti. "Adaptation of ARAIM Algorithm for Urban Environment Applications". In: *Proceeding of International Symposium on GNSS 2015*. Kyoto, Japan, Nov. 2015.

[125] Hieu Trung Tran and Letizia Lo Presti. "Demonstration of Multi-GNSS Advanced RAIM Algorithm Using GPS and Galileo Signals". In: *ICSANE 2013 (International Conference on Space, Aeronautical and Navigational Electronics* 113.335 (2013), pp. 191–196.

[126] Hieu Trung Tran and Letizia Lo Presti. "Kalman Filter-Based ARAIM Algorithm for Integrity Monitoring in Urban Environment". In: *ICT Express* (June 2018). ISSN: 2405-9595. DOI: 10.1016/j.icte.2018.05.002. URL: http://www.sciencedirect.com/science/article/pii/S2405959518300158.

[127] James Bao-Yen Tsui. "Basic GPS Concepts". In: *Fundamentals of Global Positioning System Receivers: A Software Approach* (2005), pp. 7–31.

[128] N. Viandier et al. "GNSS Performance Enhancement in Urban Environment Based on Pseudo-range Error Model". In: *2008 IEEE/ION Position, Location and Navigation Symposium*. May 2008, pp. 377–382. DOI: 10.1109/PLANS.2008.4570093.

[129] Phillip W Ward, John W Betz, Christopher J Hegarty, et al. "Satellite signal acquisition, tracking, and data demodulation". In: *Understanding GPS principles and applications* 5 (2006), pp. 174–175.

[130] PW Ward. "Interference, multipath, and scintillation". In: *Understanding GPS: Principles and Applications* 5 (2006), pp. 243–299.

[131] Andreas Wieser. "Robust and fuzzy techniques for parameter estimation and quality assessment in GPS". PhD thesis. Graz University of Technology, Graz, Austria, 2001.

[132] Andreas Wieser, Mark G Petovello, and Gérard Lachapelle. "Failure scenarios to be considered with kinematic high precision relative GNSS positioning". In: *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2004)*. Long Beach, CA, Sept. 2004, pp. 1448–1459.

[133] Working Group C - ARAIM Technical Subgroup. *Federal Radionavigation Plan*. Technical documentation DOT-TSC-RSPA-84.8. DoD, US, DHS, US, and DoT, US, 2012.

[134] Working Group C - ARAIM Technical Subgroup. *Interim Report*. Technical report. EU/US Cooperation on Satellite Navigation, Dec. 2012.

[135] Working Group C - ARAIM Technical Subgroup. *Minimum Operational Performance Standards for Global Positioning System/Satellite-Based Augmentation System Airborne Equipment*. Standard RTCA/DO-229D. Radio Technical Commission for Aeronautics, Feb. 2013.

[136] Peng Xie and Mark G Petovello. "Measuring GNSS multipath distributions in urban canyon environments". In: *IEEE Transactions on Instrumentation and Measurement* 64.2 (2015), pp. 366–377.

[137] RYAN S. Y. YOUNG and GARY A. MCGRAW. "Fault Detection and Exclusion Using Normalized Solution Separation and Residual Monitoring Methods". In: *Navigation* 50.3 (2003), pp. 151–169. ISSN: 2161-4296. DOI: 10.1002/j.2161-4296.2003.tb00326.x. URL: http://dx.doi.org/10.1002/j.2161-4296.2003.tb00326.x.

[138] Ryan SY Young, Gary A McGraw, and Brian T Driscoll. "Investigation and comparison of horizontal protection level and horizontal uncertainty level in FDE algorithms". In: *Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996)*. Kansas City, MO, Sept. 1996, pp. 1607–1614.

[139] L. Zhang et al. "An adapted RAIM algorithm for urban canyon environment". In: *Proceeding of the 2017 Forum on Cooperative Positioning and Service (CPGPS*. May 2017, pp. 116–121. DOI: 10.1109/CPGPS.2017.8075108.

[140] Ralf Ziebold, Luis Lanca, and Michailas Romanovas. "On fault detection and exclusion in snapshot and recursive positioning algorithms for maritime applications". In: *European Transport Research Review* 9.1 (2016), p. 1. ISSN: 1866-8887. DOI: 10.1007/s12544-016-0217-5. URL: http://dx.doi.org/10.1007/s12544-016-0217-5.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.