Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Ingegneria Elettronica, Telecomunicazioni e Tecnologie dell'Informazione

Ciclo XXIX

Settore Concorsuale di afferenza: 09/F2
Settore Scientifico disciplinare: ING-INF/03

COOPERATIVE INTERFERENCE DETECTION, LOCALIZATION, AND MITIGATION IN GNSS

Presentata da: Marco Bartolucci

Coordinatore dottorato:
Prof. Alessandro Vanelli Coralli

Relatore:
Prof. Giovanni Emanuele Corazza

Esame finale anno 2017

Men substitute words for reality and then argue about the words.

Edwin Howard Armstrong

Contents

List of Figures

List of Tables

Acronyms

ADC	analog-to-digital converter
ADS-B	automatic dependent surveillance - broadcast
AGC	automatic gain control
AWGN	additive white Gaussian noise
CAF	complex ambiguity function
CDF	cumulative density function
CFAR	constant false alarm rate
CN0	carrier-to-noise spectral density ratio
CRB	Cramér-Rao bound
DME	distance measuring equipment
DRSS	differential received signal strength
DSWE	distributed-sensing waveform estimation
ED	energy detector
EGNOS	European geostationary navigation overlay service
EKF	extended Kalman filter
ENU	east-north-up
FC	fusion center
FDOA	frequency difference of arrival
FIM	Fisher information matrix

FS	Fourier series
FSAE	Formula SAE
GBAS	ground bases augmentation system
GNSS	Global Navigation Satellite Systems
I-JJDL	innovation-driven joint jammer detection and localization
IF	intermediate frequency
INS	inertial navigation system
JJDL	joint jammer detection and localization
KF	Kalman filter
KLE	Karhunen-Loève expansion
KST	Kolmogorov-Smirnov test
LTE	Long Term Evolution
MAP	maximum a-posteriori
MMSE	minimum mean square error
MSE	mean square error
O-JJDL	observability-driven joint jammer detection and localization
PDF	probability density function
PPD	personal privacy devices
PSD	power spectral density
PVT	position velocity and time
RFI	radio-frequency interference
RMS	root mean square
RMSE	root mean square error

SA	service area
SAE	Society of Automotive Engineers
SDR	software-defined radio
SNR	signal-to-noise ratio
SoO	signals of opportunity
TACAN	tactical air navigation
TDOA	time difference of arrival
UWB	Ultrawide Band

Original contributions

- Development of a novel cooperative interference detection and localization algorithm which features a constant false alarm rate and provides an simple design methodology. Thanks to its flexibility, the algorithm can be easily modified in order to use different localization techniques. [P1], [P2], [P3] [P4];
- Development of an optimized GNSS/INS integration algorithm which allows to obtain sub-meter positioning accuracy, thanks to the optimization of the stochastic model of the extended Kalman filter [P5];
- Development of a novel technique for the time synchronization of low-cost software-defined radios, which reduces the synchronization error to less than one sampling interval. This technique enables the implementation of new navigation applications on low-cost hardware [P6];
- Development of a novel cooperative GNSS jammer waveform estimation algorithm, suitable for interference cancellation. The algorithm is robust against channel model mismatch thanks to the cooperation of multiple nodes [P7], [P8];
- Contribution in the development of a novel time-frequency GNSS interference rejection method, based on the S-transform. The technique allows the mitigation of non-stationary interferers and it is suitable for real-time implementations [P9];
- Contribution to the analysis of the performance of Internet of Things application layer protocols over satellite links, characterized by high packet error rates and consistent delays [P10].

List of Publications

[P1] M. Bartolucci, R. Casile, G. Pojani, and G. E. Corazza, "Joint jammer detection and localization for dependable GNSS," in *Proc. ION'2015 Pacific PNT Meeting*, Honolulu, Hawaii, Apr. 2015, pp. 498–506.

[P2] M. Bartolucci, G. Pojani, Y. Abdoush, and G. E. Corazza, "Joint jammer detection and localization in GNSS," *IEEE Trans. Aerosp. Electron. Syst.*, [to be submitted].

[P3] M. Bartolucci, G. Pojani, J. A. del Peral-Rosado, J. A. García-Molina, M. Crisci, and G. E. Corazza, "TDOA/FDOA-based joint jammer detection and localization," *IEEE Trans. Aerosp. Electron. Syst.*, [in preparation].

[P4] G. Pojani, M. Bartolucci, J.A. García-Molina, and G. E. Corazza, "Snapshot localization of a single jammer using TDOA and FDOA measurements," in *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, [submitted].

[P5] G. Pojani, M. Bartolucci, M. Conti, and G. E. Corazza, "Optimal EKF for quasi-tightly coupled GNSS/INS integration," in *Proc. 9th Baska GNSS Conference*, Baska, Croatia, May 2015, pp. 101–122.

[P6] M. Bartolucci, J. A. del Peral-Rosado, R. Estatuet-Castillo, J. A. García-Molina, M. Crisci, and G. E. Corazza, "Synchronisation of low-cost open source SDRs for navigation applications," in *Proc. 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC'16)*, Noordwijk, The Netherlands, Dec. 2016, pp. 1–7.

[P7] M. Bartolucci, R. Casile, G. E. Corazza, A. Durante, G. Gabelli, and A. Guidotti, "Cooperative/distributed localization and characterization of GNSS jamming interference," in *Localization and GNSS (ICL-GNSS), 2013 International Conference on*, June 2013, pp. 1–6.

[P8] M. Bartolucci, R. Casile, G. Gabelli, A. Guidotti, and G. E. Corazza, "Distributed-sensing waveform estimation for interference cancellation," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, September 2014, pp. 3461 – 3468.

[P9] Y. Abdoush, G. Pojani, M. Bartolucci, and G. E. Corazza, "Time-frequency interference rejection based on the S-transform for GNSS applications," in *Proc. IEEE International Conference on Communications (ICC)*, [accepted].

[P10] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of things application layer protocol analysis over error and delay prone links," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Sept 2014, pp. 398–404.

Introduction

This thesis is the outcome of the three-year work done during the Ph.D. course in "Electronics, Telecommunications and Information Technologies Engineering". The main research topic is focused on cooperative interference countermeasures in Global Navigation Satellite Systems (GNSS).

Due to the low cost of GNSS receivers and their consequent diffusion, a wide range of location-aware applications are arising. Some of these applications are critical and have strict requirements in terms of availability, integrity and reliability. Examples of critical applications are precision landing and en-route navigation in air transportations; automated highways and mileage-based toll in road transportations; search and rescue in safety of life applications. A failure in fulfilling one or more requirements of a critical application may have dramatic consequences and cause serious damage. One of the most challenging threats for critical GNSS application, is represented by interference. In particular, jamming devices, operating inside GNSS bands, are easily and cheaply purchasable on the Internet. These devices transmit disturbing signals with the aim of preventing the correct operations of GNSS receivers. In order to satisfy the requirements of critical applications, it is necessary to promptly detect, localize and remove such interfering sources. Moreover, it is important to characterize the interfering signals in order to develop interference avoidance and mitigation techniques that ensure robustness of GNSS receivers to interference.

This work is organized as follows. The Introduction discusses the problem of interference in GNSS and gives the motivation for this work.

Chapter 1 tackles the problems of interference detection and localization. These problems are solved by a novel family of algorithms in a joint and cooperative fashion.

Chapter 2 treats the interference mitigation problem. The chapter illustrates two

interference mitigation techniques, one based on the cooperative interfering waveform estimation and the other based on multisensor integration.

Chapter 3 describes a novel synchronization technique suitable for low-cost software-defined radios. The synchronization technique allows for an easy and accessible implementation of navigation-related applications.

Chapter 4 contains the conclusions.

Appendix A provides the useful theoretical background.

GNSS applications

The worldwide availability and accuracy of GNSS for location and timing, has made GNSS the preferred solution for a wide and growing range of applications in disparate fields: i.e., transport, law enforcement, highways management, health services, financial services, information services, cartography, safety monitoring, scientific and environmental studies, search and rescue, telecommunications, and asset tracking. In the road transport field GNSS is used for in-car navigation, commercial fleet management, taxi services, public transport monitoring, emergency vehicle location, distance based charging systems. In aviation, GNSS is currently used in commercial aircrafts for en-route navigation and approaches to enabled airports (e.g., using European geostationary navigation overlay service (EGNOS)); automatic dependent surveillance - broadcast (ADS-B) is used in the areas where there is no radar coverage; this involves aircrafts calculating their position using GNSS and inertial navigation systems and broadcasting it to other aircrafts. Scientific applications include surveying, environmental and atmosphere monitoring, animal behavior studies, meteorology and climate research. GNSS timing is of paramount importance for telecommunications applications. Synchronous technologies require a time source with appropriate accuracy, stability and reliability and GNSS is the preferred synchronization solution.

A few of the aforementioned applications are critical, a failure affecting any of these applications may lead to hazardous situations, risk of death, or extensive damage or losses. Indeed, critical applications have typically strict requirements in terms of accuracy, availability, and integrity. On the other hand, the intrinsic vulnerability of GNSS signals to interference, makes the fulfilment of these requirements a challenging task [1]. For this reason, the issue of interference has been extensively investigated and studied by the scientific community. The next section provides a classification of the most common interfering sources.

Interfering sources

The first and intuitive classification is based on intention. Interfering sources can be organized in unintentional and intentional: the former is the accidental or undesired transmission of radio-frequency signals over one or more GNSS bands; the latter is the deliberate transmission of disturbing or malicious signals to disrupt or mislead GNSS receivers.

Unintentional interference

Unintentional interference can be further classified into natural or man-made interference. The former is due to natural phenomena, such as solar radiation bursts. As an example, in 2006 a solar flare caused the unavailability of GPS signals over a large area of the globe, as reported in [2]. Man-made unintentional interference typically arises from malfunction of electronic devices. As an example tv broadcasters are potential interference source because higher order harmonics fall in GNSS bands. In addition, falls in this category the intrinsic interference caused by the coexistence of GNSS and traditional aircraft navigation systems, such as distance measuring equipment (DME) and tactical air navigation (TACAN), as they transmit in the proximity of the L5/E5A band [3]. Other unintentional man-made sources are due to weather and radar systems.

Intentional interference

Intentional interference can be further split in two categories, jamming and spoofing. Jamming is the transmission of a disturbing signal in one or more GNSS bands with the objective to overwhelm nearby receivers [2], [4]. Depending on the power of jamming signals at the receivers, the interferer may cause a deterioration of the position solution accuracy, or undermine its availability.

Spoofing is the transmission of counterfeit GNSS signals in order to deceive receivers [5]. In this dissertation we focus on jammers as they are more likely to affect civil GNSS user [2].

Civil GNSS jammers

Civil GNSS jammers can generate a variety of different waveforms. According to [4] the waveforms can be classified based on their complexity:

- Class I, continuous wave;

- Class II, chirp with one saw-tooth function;
- Class III, chirp with multiple saw-tooth functions;
- Classe IV, chirp with frequency-hopping.

Class I jammers are the most simple ones. They transmit a tone, i.e., a continuous wave signal. This kind of waveform is very narrowband and easy to mitigate. More harmful jammers are the ones of Class II, III, and IV. They transmit wideband swept-frequency continuous wave signal (i.e., chirp signals). An example of Class II jamming signal is shown in Figure 1 [2]. The radio-frequency carrier is modulated by a single saw-tooth function, resulting in a bandwidth of about 12 MHz around the L1/E1 center frequency. The waveform repeats with a chirp period of 8 μs. According to [6], most of the jammers are modulated by a single saw-toot function. Even Class III jammers are ideally modulated by a single saw-tooth function, multiple saw-tooths arise due to the nonlinearities in jammers front-ends.

Civil jammers are usually available for purchase on the Internet as personal privacy devices (PPD). These kind of devices promise to disable GNSS receivers in order to avoid tracking [7], [4]. PPD feature peak transmit powers from a few milliwatts to tens of Watts. They are usually installed in vehicles and are intended to make GNSS signals unavailable within a few tens of meters around the jammer. However evidence show that these devices cause problems well beyond the nominal operating range [6], [7]. A few examples of in-car jammers are shown in Figure 2 [4].

Cheap in-car jammers have been involved in a few incidents in the past. In the late 2009, the ground bases augmentation system (GBAS) of Newark Liberty International Airport (New Jersey, USA) was suffering from brief daily outages due to the emissions of an in-car jammer used in a vehicle traveling on the nearby highway. [8]. A measurement campaign conducted in London in 2012, showed the presence of jamming events repeating every working day at the same time [2]. In April 2012, the police in Kent (UK) had arrested a gang of car thieves responsible for the theft of more than 150 high-valued vehicles. The thieves where using GNSS jammers to disable in-car tracking systems. In November 2013 a Melbourne newspaper reported the that a hundred of taxi drivers where using jammers in order to fool their employer company into giving them jobs, even if they were not in the area. The devices were discovered as they were interfering with the receivers of nearby police cars and ambulances.

These incidents suggest that the threat represented by jammers should not be

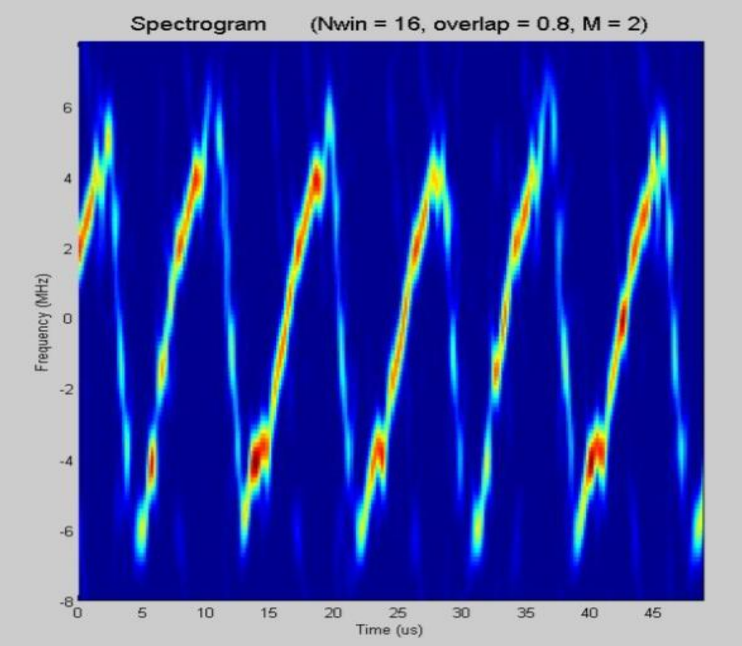

Figure 1: Spectrogram of a typical Class II jamming signal

Figure 2: Examples of in-car PPD

underestimated. With a growing number of new applications relying on GNSS, we expect an increase of jammer events frequency. Therefore, it is important to develop suitable jammer detection, localization and mitigation techniques.

The following section describes the adopted jamming signal model that will be used in this dissertation.

Jamming signal model

Jammer interfering signals are generated through phase/frequency modulation of a sine wave with a generic periodic signal $z(t)$, with period T_j. For civil jammers $z(t)$ is typically a periodic saw-tooth waveform, expressed as:

$$z(t) = f_i + (f_f - f_i)\left(\frac{1}{2} + \frac{t - T_j/2}{T_j} - \left\lfloor \frac{1}{2} + \frac{t - T_j/2}{T_j} \right\rfloor\right) \tag{1}$$

where f_i and f_f are the initial and final frequencies, respectively, and T_j is the chirp period. A jamming signal can be represented according to [9].

$$s(t) = A(t)\exp\left\{j\left[2\pi f_c t + \varphi(t)\right]\right\} \tag{2}$$

where $A(t)$, f_c, and $\phi(t)$ represent amplitude, carrier frequency and phase ofd the signal, respectively.

$$\varphi(t) = \begin{cases} 2\pi \int_0^t z(\xi)\,\mathrm{d}\xi & \text{for FM signals} \quad (3a) \\ z(t) & \text{for PM signals} \quad (3b) \end{cases}$$

Since the modulating signal is periodic with period T_j, it can be written as

$$z(t) = \sum_{k=-\infty}^{+\infty} z_0\left(t - kT_j\right) \tag{4}$$

with $z_0(t)$ null outside $[0, T_j]$. If the signal amplitude is slowly varying, it is safe to assume it constant within each repetition period:

$$A(t) = A_k \in \mathbb{R},\ t \in [kT_j, (k+1)T_j[\tag{5}$$

and the jamming signal can be expressed as

$$s(t) = \sum_{k=-\infty}^{+\infty} A_k S_0(t - kT_j) e^{j\phi_k} \tag{6}$$

with

$$S_0(t) = \mathbf{1}_{[0,T_j[}(t) \begin{cases} \exp\left\{j2\pi\left[f_c t + \int_0^t z_0(\xi)\,\mathrm{d}\xi\right]\right\} & \text{FM signals} \quad (7a) \\ \exp\left\{j\left[2\pi f_c t + z_0(t)\right]\right\} & \text{PM signals} \quad (7b) \end{cases}$$

$$\phi_k = \begin{cases} 2\pi\left[f_c k T_j + k\int_0^{T_j} z_0(\xi)\,\mathrm{d}\xi\right] & \text{for FM signals} \quad (8a) \\ 2\pi f_c k t_j & \text{for PM signals} \quad (8b) \end{cases}$$

where $\mathbf{1}_{[a,b[}(t)$ is the indicator function defined as:

$$\mathbf{1}_{[a,b[}(t) = \begin{cases} 1 & \text{if } t \in [a,b[\quad (9a) \\ 0 & \text{if } t \notin [a,b[\quad (9b) \end{cases}$$

Chapter 1

Joint Interference Detection and Localization

Several countermeasures to interference have been studied since the early days of GNSS. The first necessity is to detect the presence of interference, which as a minimum allows to reduce the trust on the position velocity and time (PVT) solution. But this may not be sufficient in many application scenarios, particularly those in which integrity must be preserved by civil/military servants. In these conditions, localization of the interference source is also essential, in order to identify and resolve the problem at its roots. For This purpose, cooperation between different nodes can be conceived. Before going into the details of our solution to the interference detection and localization problem. we shall review the main state-of-the-art techniques in this topic.

1.0.1 Interference detection

In the literature, interference detectors are mostly implemented into GNSS signal processing chains. At the expense of additional hardware, other architectures could integrate sensors (e.g., inertial navigation systems) to cross-check the position solution or use multiple antennas to distinguish the angles of arrival. Hereinafter, we focus on cost-effective detection methods, which are performed either by processing the whole pre-correlation signal or by monitoring the post-correlation and channel-specific measurements available.

Detectors based on pre-correlation observables (i.e., computed before despreading), share an inherent advantage: they do not require any a-priori knowledge of the

jamming waveform. They search the incoming signal for interference by examining either the quantized amplitude levels at the automatic gain control (AGC) output or the raw data captured at baseband, or at intermediate frequency (IF). As proposed in [10], the adaptive digitization loop is a valuable tool for assessing the presence of jammers within the processing chain of a standard receiver. Nevertheless, this stage is not specifically designed to cope with interference: its dynamic range falls short of flexibility, because either the quantization accuracy or the additional resolution could be insufficient. Alternatively, other pre-correlation detectors inspect the raw data of the analog-to-digital converter (ADC), in order to test the noise-like properties of the interference-free GNSS signal. Indeed, temporal correlations between samples, distortions of the power spectral density (PSD), and inconsistent statistics are evidence of possible radio-frequency interference (RFI). In this category fall the conventional detection and excision methods, which analyze the energy computed in either the time or the frequency domains, as described in [11] [12] [13] [14] [15]. Similarly, recent and promising techniques proposed in [16] [17] [18] analyze the intensity of a time-frequency representation of the signal. However, regardless of the domain, energy detectors are limited by the noise level uncertainty. Therefore, their decision threshold depends on the real-time estimation of the time-varying noise variance, other methods provide non-parametric RFI detection and may be referred to as blind. For instance, the spectrum-sensing technique in [19] takes a decision depending on the eigenvalues of the covariance sample matrix. Another example is the chi-square goodness-of-fit test exploited in [20], which assumes that a zero-mean white Gaussian process can model the received GNSS signal, as it is dominated by noise, in the absence of interference and multipath fading.

As opposed to pre-correlation observables, jamming attacks can be revealed also by combining several post-correlation measurements derived for each channel, although the despreading operation generally reduces detection performance and responsiveness. In [21], the estimated correlator output power is the only candidate test statistics showing consistent performance with continuous, pulsed, and broadband interference, but it requires the calculation of the receiver-specific expected noise floor. A more sophisticated approach is presented in [22], which detects RFI at the tracking loop by examining the autocorrelation peak shape through a multi-correlator receiver. In [23], the presence of jammers can be determined based on the positioning error estimated through the time difference between pseudo-range and carrier phase. This estimation process features low sensitivity to the jammer char-

acteristics, however it inevitably introduces a significant delay. A well-established technique is presented in [24] based on monitoring the channel carrier-to-noise spectral density ratio (CN0), which is estimated from the relative signal-to-noise ratio (SNR) measured for each visible satellite. The purpose is taking advantage of the features of off-the-shelf commercial GNSS receivers. Even though commercial receivers usually provide CN0 values, yet their internal estimation process is unknown. Therefore, the channel thresholds are designed by resorting to an information-demanding statistical characterization of the CN0 curve against satellite elevation angles. Furthermore, separate CN0 estimates cannot provide reliable detection, since they do not distinguish between drops in GNSS signal strength and rise in interference power, as asserted in [25]. They are meaningful in well-surveyed static scenarios only. For example in [26], in-car jammers are detected by measuring SNR levels by means of stations placed along roadside. As far as dynamic scenarios are concerned, CN0 values of multiple satellites should rather be collectively taken into account with a unique decision metric. A sum of squared CN0 variations is proposed in [27], under the assumption that jammers cause correlated changes in all the channels.

All of the above methods foresee a receiver working in isolation. However, in particular for mission critical applications, it is conceivable to deploy a network of cooperating nodes, which can provide decisive performance improvement.

1.0.2 Interference localization

Once the detector flags a jamming attempt, the position of the interference source is determined, possibly leading to the removal of the device responsible of the event. Here, we consider the techniques employing a network of sensor nodes and a fusion center (FC), equipped with more computation power, which collects data from the nodes and performs signal processing.

Traditional localization techniques are mainly based on ranging: the distance between the transmitter and each receiver is estimated from either the power, delay, or bearing of the interference, with little or no a-priori knowledge about the waveform of interest. These measurements are compared in terms of dilution of precision in [28]. On the one hand, the effectiveness of the differential received signal strength (DRSS) scheme, depends on the accuracy of the underlying propagation model, the level of noise, and the geometry of node deployment. On the other hand, time difference of arrival (TDOA) calculated through cross-correlation guarantees superior and reliable performance, even in the presence of multipath fading, at the expense of a

heavier communication and computational burden. For the same reasons mentioned regarding detection, we neglect angle of arrival, which requires multiple antennas for beamforming. The authors of [29] use both power and time information for detecting and localizing a jammer. However, they obtain the DRSS from the amplitude quantized levels of the AGC. As a result, due to the aforementioned non-linearity of the AGC voltage, the power/distance relation has some intrinsic boundaries, beyond which the position solution results inaccurate or impossible. They also employ hyperbolic localization thorough TDOA, similarly to [30, 31], under the assumption that all nodes running independently are strictly synchronized among themselves. Such a scheme indeed relies on the fine alignment of all the data streams in both time and frequency. In principle, this requirement could be fulfilled through the GNSS time solution in the interference-free sections of each signal record. However, since RFI is likely to be excessive, every node usually resort to a coarse alignment by estimating the current time from its internal clock. Another requirement underlying the TDOA scheme regards the communication bandwidth necessary for transferring the datasets of each node to the FC, which performs the cross-correlation. In [32], a technique is presented to relax the demand for bandwidth, by partially distributing the computational load among the nodes. This method accurately tracks a single jammer by means of an extended Kalman filter (EKF), which makes use of TDOA measurements obtained as the time when the interfering tone passes through a chosen frequency. Nevertheless, this technique implies that the jammer waveform is known. On the contrary, the approach presented here, combines the use of an EKF with the adoption of the DRSS scheme. This solution enables the localization of the interference source with minimum computational burden and data exchange, regardless of the jammer characteristics.

1.1 Joint jammer detection and localization algorithm

This chapter describes a novel technique, called joint jammer detection and localization (JJDL), for simultaneously detecting and localizing a single jammer through the cooperation of low-complexity nodes. The original concept for the JJDL algorithm was to use the notion of system observability for jammer detection. in the observability-driven joint jammer detection and localization (O-JJDL) algorithm [P1], the nodes are equipped with an energy detector (ED) and they take a partial decision on the absence/ presence of a jammer. Moreover, they estimate the

normalised distance from the potential jammer and send it to a FC. The FC, based on the degree of observability of an EKF, takes an aggregate decision on the presence or absence of a jammer. In case a jammer is detected, a second EKF estimates the jammer's position. However, this initial concept of JJDL algorithm presented a few issues and it is reported here merely for historical reasons. Those issues have been solved in the improved version of the algorithm, called innovation-driven joint jammer detection and localization (I-JJDL) [P2].

In the I-JJDL algorithm, each node collects simple observables from the incoming raw signal samples, these observables are a function of the distance from the interference source and/or relative velocity of the jammer. The observables are then sent to a FC, where an EKF tracks the source position (and velocity), even in dynamic scenarios. Concurrently, the observation of the filter innovation provides a non-parametric test statistics for jamming detection, which is both independent from the waveform characteristics and robust to the noise level uncertainty. The decision threshold can be conveniently set in order to obtain a constant false alarm rate (CFAR). Moreover, as proven by simulation results, this metric has also a high detection capability, and yet it is fast in revealing jammer presence.

1.2 System model

We consider an area of interest that must be kept free from interference sources, identified as service area (SA). The SA, of size $l \times l$, is populated with M sensor nodes, which have known positions and unitary antenna gains. They monitor a selected GNSS band, store signal samples, and are able to communicate with the FC. The communication link between the FC and the nodes is not object of this dissertation and, therefore, it is assumed as ideal. For the sake of simplicity the considered scenario is two-dimensional. The extension to three dimensions is straightforward. A jammer starts transmitting an unknown waveform, with carrier frequency f_c at time instant t_J from an unknown location within the SA. Figure 1.1 shows a generic scenario: the i-th sensor node, located in (X_i, Y_i) is depicted as a white dot, the jammer located in (X, Y), is depicted as a black dot, and d_i is the distance between them. The goal of the JJDL method is to reveal the presence of a potential jammer and to estimate its location. For this purpose, we model the detection problem as a binary decision problem with the following hypotheses:

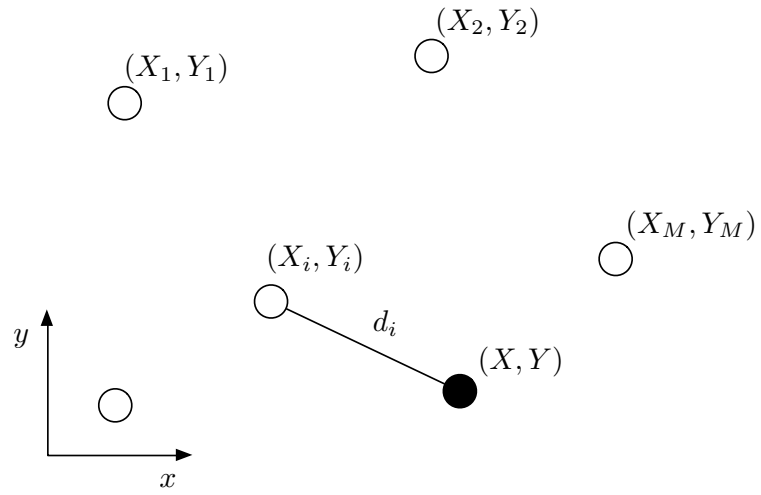

Figure 1.1: Scenario

$$\begin{cases} H_0, & \text{Jammer absent} \quad (1.1a) \\ H_1, & \text{Jammer present.} \quad (1.1b) \end{cases}$$

The hypothesis H_0 describes an interference-free scenario, while the hypothesis H_1 represents the presence of a jammer. In absence of interference, nodes will receive the GNSS signal plus the additive white Gaussian noise (AWGN) term $n_i(t)$. In the presence of interference, nodes will also receive the jamming waveform $s_{r.i}(t)$. Therefore, the signal received by the i-th node can be written as:

$$r_i(t) = \begin{cases} n_i(t), & \text{if } H_0 \text{ holds} \quad (1.2a) \\ s_{r,i}(t) + n_i(t), & \text{if } H_1 \text{ holds} \quad (1.2b) \end{cases}$$

whose discrete-time formulation is given by

$$\mathbf{r}_i = \begin{cases} \mathbf{n}_i, & \text{if } H_0 \text{ holds} \quad (1.3a) \\ \mathbf{s}_{r,i} + \mathbf{n}_i, & \text{if } H_1 \text{ holds,} \quad (1.3b) \end{cases}$$

In the following, we present first the original concept of the algorithm (O-JJDL) and then the improved version (I-JJDL)

1.3 Observability-driven JJDL algorithm

The O-JJDL algorithm consists of three main phases:

(i) Energy detection;

(ii) Aggregate decision;

(iii) Jammer localization.

In the first phase, each node measures the received power and takes a partial decision on the absence or presence of an interfering signal. Whenever the detection is asserted, the single node estimates a measure of its distance d_i from the jammer. During the second phase, an EKF takes an aggregate decision on the same hypothesis. If the aggregate decision is asserted, the position of jammer is recursively estimated with higher accuracy by a second EKF.

1.3.1 Energy detection

Each node is equipped with an ED and is able to decide if an interfering signal is present ($\hat{H}_1$) or absent ($\hat{H}_0$), with a given constant false alarm probability P_{fa}, as explained in appendix A.2. According to this partial decision, the radiometer converts the received power into an observable y_i, defined as:

$$y_i = \begin{cases} \dfrac{c}{4\pi f_c} \sqrt[\alpha]{\dfrac{P_t}{T(\mathbf{r}_i) - P_N}} & \text{if } \hat{H}_1 \quad (1.4a) \\ Y & \text{if } \hat{H}_0 \quad (1.4b) \end{cases}$$

where c is the light velocity in vacuum, f_c is the jammer carrier frequency, P_t is the jammer transmit power, P_N is the receiver noise power, Y is an arbitrary large number, and $T(\mathbf{r}_i)$ is the test statistics of the ED, i.e. the average squared magnitude of N samples of the received signal:

$$T(\mathbf{r}_i) = \frac{1}{N} \sum_{n=1}^{N} |r_i[n]|^2 \quad (1.5)$$

In the following, the path-loss exponent is set to $\alpha = 2$ for simplicity but it can be fitted to the desired propagation environment. Since the jamming transmit power is unknown, we are unable to determine the true distance between nodes and the jammer. Instead, we suppose a transmit power of 1W. This assumption leads to inevitable errors in the interferer's position estimation, which do not affect detection performance. This issue of positioning accuracy is later resumed and addressed in the localization stage of the algorithm. A diagram summarizing the operation of each individual node is shown in Figure 1.2. Although it is easy to detect a high power interfering signals by means of energy detectors, this becomes quite challenging when the received signal power is comparable with noise power. In this case, indeed, the decision threshold results to be very low and false alarms are very likely to occur. A combined solution capable of solving this issue is presented in the next subsection.

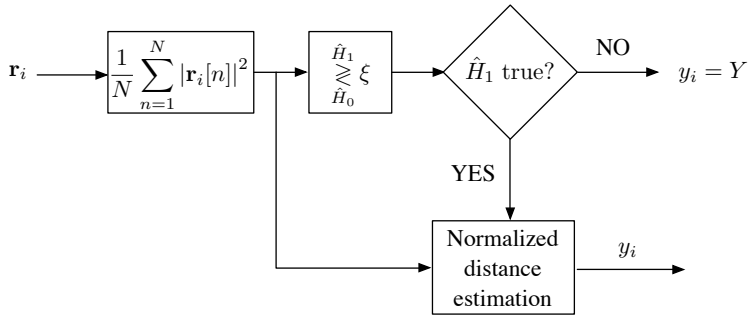

Figure 1.2: O-JJDL node block diagram

1.3.2 Aggregate decision

The detection Kalman filter's unknown state vector to be estimated is composed by the jammer's coordinates, $\mathbf{x}_k = (X_k, Y_k)^T$. Here the index k indicates a generic EKF iteration. The equations describing the optimal EKF (see §A.1) are

$$\begin{cases} \hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k & \text{(1.6a)} \\ \mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q}_k & \text{(1.6b)} \end{cases}$$

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} & \text{(1.7a)} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \right) & \text{(1.7b)} \\ \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- & \text{(1.7c)} \end{cases}$$

where $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k$ are the a-priori and a-posteriori state estimates, respectively; $\mathbf{A}_k$ is the state transition matrix describing the evolution of the system in absence of state noise and control inputs; $\mathbf{B}_k$ is the control-input matrix; $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are the a-priori and a-posteriori estimation error covariance matrices, respectively; $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the process noise and measurement noise covariance matrices, respectively; $\mathbf{H}_k$ is the observation matrix, and $\mathbf{K}_k$ is the optimal EKF gain. We suppose a stationary transition model with an identity matrix $\mathbf{A}_k = \mathbf{I}_2$ for each time instant k, which is equivalent of assuming a fixed interferer location. Furthermore, no controlled input signals are considered in the system, $\mathbf{B} = 0$. As far as observations are concerned, the nonlinear relation connecting measurements and the state vector is given by:

$$h_{i,k}^D(\mathbf{x}_k) = \sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2} \tag{1.8}$$

The observation model $\mathbf{H}_k^D$ of size $2 \times M$ is described by the Jacobian matrix of $h_{i,k}^D$:

$$\mathbf{H}_k^D = \begin{pmatrix} -\frac{X_1-X_k}{\sqrt{(X_1-X_k)^2+(Y_1-Y_k)^2}} & -\frac{Y_1-Y_k}{\sqrt{(X_1-X_k)^2+(Y_1-Y_k)^2}} \\ \vdots & \vdots \\ -\frac{X_M-X_k}{\sqrt{(X_M-X_k)^2+(Y_M-Y_k)^2}} & -\frac{Y_M-Y_k}{\sqrt{(X_M-X_k)^2+(Y_M-Y_k)^2}} \end{pmatrix} \tag{1.9}$$

The fundamental idea of this algorithm comes from the intuition that the observables y_i match this model if and only if an interfering signal is detected. On the contrary, if no interference is recognized, the EKF is not optimal and thus does not provide a minimum mean square error (MMSE) solution to the state estimation problem. More information on EKF observability can be found in §A.1.2.

Let us now consider the eigenvalues of the estimation error covariance matrix $\mathbf{P}_k$: its largest eigenvalue corresponds to the least observable linear combination of state components (X_k, Y_k). Therefore, the maximum eigenvalue of $\mathbf{P}_k$ measures the degree of observability of the EKF. Based on this remark, we can define a discriminating function as the maximum eigenvalue of $\mathbf{P}_k$:

$$D(\mathbf{P}_k) = \max_{i=1,2}\{\lambda_i\} \tag{1.10}$$

Whenever the discriminating function takes low values, the system is characterized by high observability, indicating the presence of a jammer. As a consequence, we can define a threshold ξ_e for an aggregate decision test on the aforementioned hypothesis based on measurements collected from all the nodes:

$$D(\mathbf{P}) \underset{\hat{H}_1}{\overset{\hat{H}_0}{\gtrless}} \xi_e \tag{1.11}$$

where $\hat{H}_0$ is chosen if the discriminating function is above the threshold and $\hat{H}_1$ is chosen otherwise. An example of discriminating function is shown in Figure 1.3. The figure shows the behaviour of the discriminating function over time for different values of average SNR at the nodes: the jammer starts transmitting at $t_J = 500$ and the discriminating function clearly falls in correspondence of this instant.

1.3.3 Jammer localization

Once an interference source is detected, a second EKF is initialized to estimate its position. This localization filter differs from the detection EKF in the observation model only. At this stage, we want to obtain a precise estimate of the jammer location through a more accurate distance evaluation. This is done by estimating

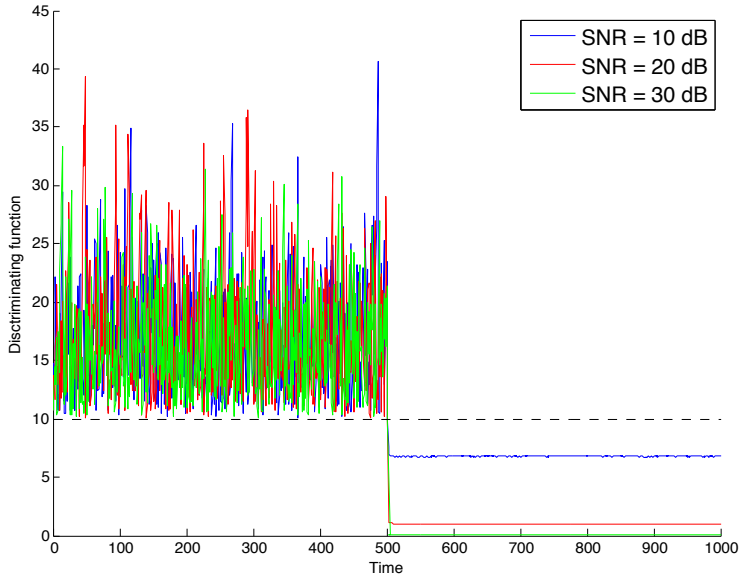

Figure 1.3: Example of O-JJDL discriminating function

also the interfering signal transmit power P_t in eq. (1.4a). Thus the unknown state vector has an additional component and becomes $\mathbf{x} = (X_k, Y_k, P_t)$ In this case, the nonlinear observation function is:

$$h_{i,k}^{L} = \frac{\sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}}{\sqrt{P_t}} \tag{1.12}$$

The partial derivatives of this function and the observation model are given by:

$$\frac{\partial h_{i,k}^{L}}{\partial X_k} = -\frac{X_i - X_k}{\sqrt{[(X_i - X_k)^2 + (Y_i - Y_k)^2]P_t}} \tag{1.13}$$

$$\frac{\partial h_{i,k}^{L}}{\partial Y_k} = -\frac{Y_i - Y_k}{\sqrt{[(X_i - X_k)^2 + (Y_i - Y_k)^2]P_t}} \tag{1.14}$$

$$\frac{\partial h_{i,k}^{L}}{\partial P_t} = -\frac{\sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}}{2\sqrt{P_t^3}} \tag{1.15}$$

$$\mathbf{H}_k^l = \begin{pmatrix} \frac{\partial h_{1,k}^{L}}{\partial X_k} & \frac{\partial h_{1,k}^{L}}{\partial Y_k} & \frac{\partial h_{1,k}^{L}}{\partial P_t} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_{M,k}^{L}}{\partial X_k} & \frac{\partial h_{M,k}^{L}}{\partial Y_k} & \frac{\partial h_{M,k}^{L}}{\partial P_t} \end{pmatrix} \tag{1.16}$$

A diagram of the complete algorithm with both detection and localization filters is represented in Figure 1.4. Although the initial concept of the O-JJDL algorithm

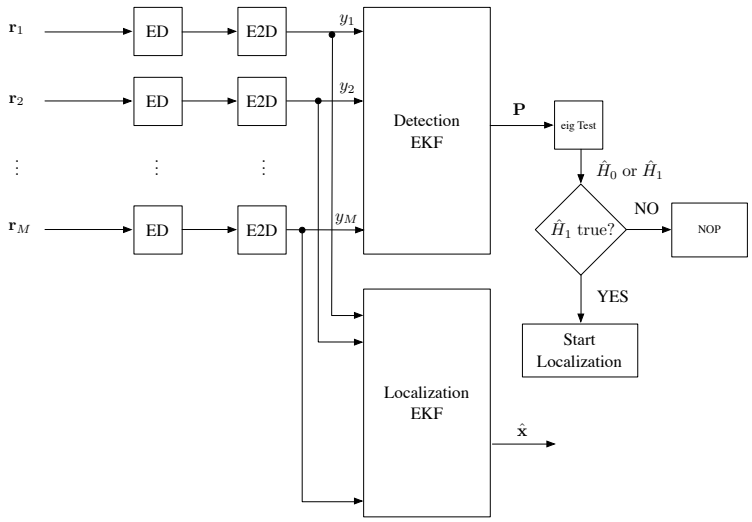

Figure 1.4: O-JJDL block diagram

comes from an original and interesting idea, the algorithm presents a few drawbacks that make its use difficult:

1. Dependence of the discriminating function on the scenario;
2. Dependence of the discriminating function on $\mathbf{R}_k$ and $\mathbf{Q}_k$;
3. Necessity two EKF;
4. No a-priori information on detection performance.

The I-JJDL algorithm is an evolution of the original algorithm, overcomes these drawbacks and it is more suitable for real-life applications.

1.4 Innovation-driven JJDL algorithm

The I-JJDL algorithm consists of three main phases:

(i) Observables estimation;

(ii) Jammer position tracking;

(iii) Jammer detection.

In step (i), the observables are estimated from the received signal; in step (ii) the FC estimates the current position (and velocity) of the potential jammer with an EKF; in step (iii) the FC takes a decision on the presence or absence of a jammer, based

on the FC behaviour. Here two versions of the I-JJDL algorithm are presented: the first version employs power-based observables, while the second one makes use of TDOA and frequency difference of arrival (FDOA) measurements [P3]. Step (iii) is identical in both algorithms, while steps (i-ii) depend on the kind of observables.

It is wort mentioning that the algorithm can be adapted to cope with different kinds of observables, as long as they are a function of the relative distance/velocity of the jammer.

1.5 Observables estimation

1.5.1 Power-based I-JJDL algorithm

In the power-based I-JJDL algorithm, the nodes estimate the received power from the incoming signal. In order to do so, the following assumptions are made. The jamming signal is assumed to have narrower bandwidth than that of the receivers equipping the sensor nodes, and a constant power. Clearly, the relationship between received power, transmit power, and distance is in general a complex function of propagation characteristics. These are in turn dependent on the selected environment. In order to show a solution which is both simple and agnostic, we assume a classic exponential path loss model with parameter α, which can be fit to the environment. With the aforementioned assumptions, the power received by the i-th node can be expressed as:

$$P_{r,i} = \begin{cases} P_{N,i}, & \text{if } H_0 \text{ holds} \quad (1.17a) \\ P_t \left(\dfrac{c}{4\pi f_c d_i} \right)^{\alpha} + P_{N,i}, & \text{if } H_1 \text{ holds} \quad (1.17b) \end{cases}$$

where $P_{N,i}$ is the noise power, P_t is the jammer transmit power, and c is light velocity in vacuum. The received signal is then discretized as in eq. (1.3a-1.3b).

Using an observation window with N samples, nodes estimate the received power through the following estimator [P1]:

$$P_{r,i,k} \approx T(\mathbf{r}_{i,k}) = \frac{1}{N} \sum_{n=1}^{N} \left| r_{i,k}\left[n\right] \right|^2 \quad (1.18)$$

which averages the square magnitude of N samples of the incoming signal. This operation is performed for every epoch k of the EKF execution. The equation relating the estimated received power to the distance between the jammer and the i-th node

depends on the transmit power P_t, as expressed in:

$$d_{i,k} = \frac{c}{4\pi f_c} \sqrt[\alpha]{\frac{P_t}{T(\mathbf{r}_{i,k}) - P_N}} \tag{1.19}$$

However, the problem of retrieving the distance from the received power is undetermined when the transmit power is unknown. For this reason, we introduce a new observable called distance per unit of transmit power that is measured in meter per Watt:

$$y_{i,k} = \frac{c}{4\pi f_c} \sqrt[\alpha]{\frac{1}{T(\mathbf{r}_{i,k}) - P_N}} \tag{1.20}$$

The nodes compute these observables, which are then sent to the FC and collected in the vector $\mathbf{y}_k \in \mathbb{R}^M$. The ambiguity related to P_t will be solved by the EKF.

1.5.2 TDOA/FDOA-based I-JJDL algorithm

This version of the algorithm exploits TDOA and FDOA measurements instead of received power. Although the processing of these kind of observables is more cumbersome and strict synchronization of sensing nodes is required, TDOA and FDOA provide a more robust and accurate alternative to power-based measurements. Moreover, as suggested in [P6] and §3, synchronization of low-cost software-defined radio (SDR) is today possible and affordable.

TDOA is defined as the difference in time between the reception of a signal by different nodes. FDOA is the differential Doppler between different nodes. In TDOA /TDOA localization, one of the M sensing nodes is designated as reference node ($i = r$). Observables will be estimated with respect to the reference node only, as:

$$TDOA_i = TOA_r - TOA_i \tag{1.21}$$

$$FDOA_I = FOA_r - FOA_i \tag{1.22}$$

where TDOA is the difference of time of arrivals and FDOA is the difference of frequency of arrivals. In the presence of M nodes, the number of linearly independent TDOA or FDOA observables is $M - 1$. Each node records the raw samples of the received signal and sends them to the FC which is in charge of estimating the $M - 1$ TDOA/FDOA observables. The observable estimation phase of the I-JJDL, unlike the power-based case, is performed by the FC. TDOA and FDOA measurements are

obtained from the signals thanks to the processing of the complex ambiguity function (CAF), defined as [33]:

$$\chi_i(\tau, k) = \sum_{n=0}^{N-1} s_r(n)\, s_i^*(\tau + n)\, e^{-j2\pi \frac{kn}{N}} \tag{1.23}$$

where s_r and s_i are the digital signals received by node r and i, respectively; τ is the lag, k is the digital Doppler frequency, and N is the number of signal samples used for the computation of the CAF. The expression of the CAF can be rewritten in a more computationally efficient way as:

$$\chi_i(\tau, k) = \mathrm{DFT}\left\{s_r(n)\, s_i^*(\tau + n)\right\} \tag{1.24}$$

The TDOA and FDOA measurements are obtained by finding the maximum point of the CAF's magnitude in the lag-doppler domain.

$$\left(\widehat{TDOA}_i, \widehat{FDOA}_i\right) = \underset{\tau,k}{\mathrm{argmax}} \left\{|\chi_i(\tau, k)|\right\} \tag{1.25}$$

Once the observables are obtained, they are used in the next step of the algorithm for tracking the position and velocity of a potential jammer.

1.6 Jammer position tracking

The task of this phase is to estimate the position of the potential jammer by running the EKF with the observables collected in the previous step. As introduced in [34], the optimal EKF can be described by equations (1.26a)–(1.26b) and (1.27a)–(1.27c), where $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k$ are the a-priori and a-posteriori state estimates, respectively; $\mathbf{A}_k$ is the state transition matrix describing the evolution of the system in absence of state noise and control inputs; $\mathbf{B}_k$ is the control-input matrix; $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are the a-priori and a-posteriori estimation error covariance matrices, respectively; $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the process noise and measurement noise covariance matrices, respectively; $\mathbf{H}_k$ is the observation matrix, and $\mathbf{K}_k$ is the optimal EKF gain (§A.1).

$$\begin{cases} \hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} + \mathbf{B}_k \mathbf{u}_k & \text{(1.26a)} \\ \mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q}_k & \text{(1.26b)} \end{cases}$$

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} & \text{(1.27a)} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-\right) & \text{(1.27b)} \\ \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- & \text{(1.27c)} \end{cases}$$

1.6.1 Power-based algorithm

In the power-based case, the state vector to be estimated is $\mathbf{x}_k = \left(X_k, Y_k, \sqrt[\alpha]{P_{t,k}}\right)$. The use of $\sqrt[\alpha]{P_{t,k}}$ in place of $P_{t,k}$ simplifies the EKF equations and improves the filter convergence properties. The equation describing the dependence of the state vector on the i-th observable is the following nonlinear function:

$$h_{i.k}\left(\mathbf{x}_k\right) = \frac{\sqrt{\left(X_i - X_k\right)^2 + \left(Y_i - Y_k\right)^2}}{\sqrt[\alpha]{P_{t,k}}}, \tag{1.28}$$

Therefore, the whole observation model is described by the function $\mathbf{h}_k : \mathbb{R}^3 \to \mathbb{R}^M$ defined as $\mathbf{h}_k = \left(h_{1,k}, h_{2,k}, \ldots, h_{M,k}\right)^T$. The observation matrix is obtained from (1.28) by linearization (eq. (1.29)–(1.32)).

$$\mathbf{H}_k = J\,\mathbf{h}_k = \begin{pmatrix} \frac{\partial h_{1,k}}{\partial X_k} & \frac{\partial h_{1,k}}{\partial Y_k} & \frac{\partial h_{1,k}}{\partial \sqrt{P_{t,k}}} \\ \frac{\partial h_{2,k}}{\partial X_k} & \frac{\partial h_{2,k}}{\partial Y_k} & \frac{\partial h_{2,k}}{\partial \sqrt{P_{t,k}}} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_{M,k}}{\partial X_k} & \frac{\partial h_{M,k}}{\partial Y_k} & \frac{\partial h_{M,k}}{\partial \sqrt{P_{t,k}}} \end{pmatrix} \tag{1.29}$$

$$\frac{\partial h_{i,k}}{\partial X_k} = -\frac{X_i - X_k}{\sqrt{\left[\left(X_i - X_k\right)^2 + \left(Y_i - Y_k\right)^2\right]}\sqrt[\alpha]{P_{t,k}}} \tag{1.30}$$

$$\frac{\partial h_{i,k}}{\partial Y_k} = -\frac{Y_i - Y_k}{\sqrt{\left[\left(X_i - X_k\right)^2 + \left(Y_i - Y_k\right)^2\right]}\sqrt[\alpha]{P_{t,k}}} \tag{1.31}$$

$$\frac{\partial h_{i,k}}{\partial \sqrt{P_{t,k}}} = -\frac{\sqrt{\left(X_i - X_k\right)^2 + \left(Y_i - Y_k\right)^2}}{\sqrt[\alpha]{P_{t,k}^2}} \tag{1.32}$$

The determination of the optimal $\mathbf{Q}_k$ and $\mathbf{R}_k$ is beyond the scope of this dissertation, but the interested reader can refer to [34], [35], and [36]. Pragmatically, it is reasonable to assume the components of the process noise and the measurement noise from different nodes as uncorrelated. This translates into diagonal noise covariance matrices and, we can set these matrices as:

$$\begin{cases} \mathbf{Q}_k = \sigma_Q \mathbf{I}_3 & \forall k \quad (1.33a) \\ \mathbf{R}_k = \sigma_R \mathbf{I}_M & \forall k \quad (1.33b) \end{cases}$$

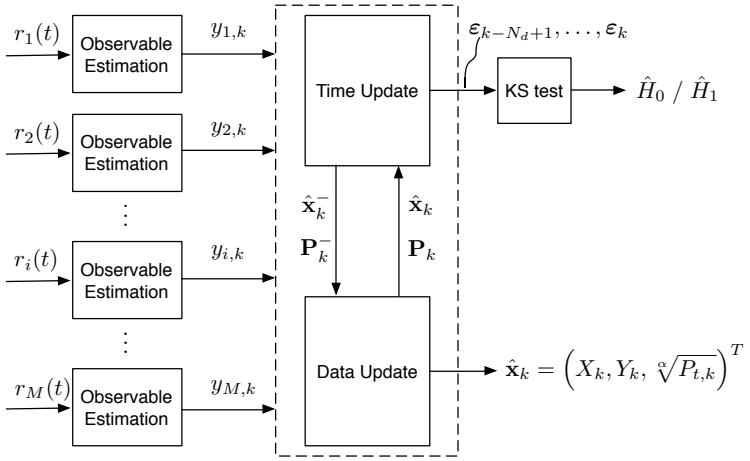

Figure 1.5: Power-based algorithm block diagram

As far as the a-priori model is concerned, the jammer can be assumed to be quasi static. This assumption can be shown to hold in most application scenarios, given the coherence time involved by typical dynamic settings. Since the jamming power is constant, the state transition matrix $\mathbf{A}_k$ is an identity matrix. The absence of controlled inputs translates into a null $\mathbf{B}_k$ matrix.

$$\begin{cases} \mathbf{A}_k = \mathbf{I}_3, & \forall k \quad (1.34a) \\ \mathbf{B}_k = \mathbf{0}, & \forall k \quad (1.34b) \\ \hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} & (1.34c) \end{cases}$$

It is worth noticing that the EKF estimates a state vector regardless of the actual presence of a jammer. Indeed, this forms the basis for the detection phase.The algorithm is summarized by the block diagram in Figure 1.5.

1.6.2 TDOA/FDOA-based algorithm

In the TDOA/FDOA version of the TDOA algorithm, the unknown system state to be estimated, includes the position and velocity of the jammer along the X and Y axes.

$$\mathbf{x}_k = (X_k, Y_k, V_{x,k}, V_{y,k}) \tag{1.35}$$

For TDOA measurements, equation 1.28 relating observables to the system state, becomes

$$h_{i,k}^{TDOA}(\mathbf{x}_k,) = \frac{\sqrt{(X_r - X_k)^2 + (Y_r - Y_k)^2}}{c} - \frac{\sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}}{c} \tag{1.36}$$

For FDOA, instead eq. 1.28 becomes

$$h_{i,k}^{FDOA}(\mathbf{x}_k) = \\ = \frac{f_c}{c}\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_r, Y_r)\rangle}{\|(X_k, Y_k) - (X_r, Y_r)\|} - \frac{f_c}{c}\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_i, Y_i)\rangle}{\|(X_k, Y_k) - (X_i, Y_i)\|} \tag{1.37}$$

where $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ indicate the Euclidean norm and the dot product, respectively. The observation model, in this case, is described by the function $\mathbf{h}_k : \mathbb{R}^4 \to \mathbb{R}^{2M-2}$ defined as $\mathbf{h}_k = \left(h_{1,k}^{TDOA}, \ldots, h_{M-1,k}^{TDOA}, h_{1,k}^{FDOA} \ldots, h_{M-1,k}^{FDOA}\right)^T$, and the observation matrix of equations (1.38)-(1.44) is the Jacobian matrix of the observation model.

$$\mathbf{H}_k = J\,\mathbf{h}_k = \begin{pmatrix} \frac{\partial h_{1,k}^{TDOA}}{\partial X_k} & \frac{\partial h_{1,k}^{TDOA}}{\partial Y_k} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_{M-1,k}^{TDOA}}{\partial X_k} & \frac{\partial h_{M-1,k}^{TDOA}}{\partial Y_k} & 0 & 0 \\ \frac{\partial h_{1,k}^{FDOA}}{\partial X_k} & \frac{\partial h_{1,k}^{FDOA}}{\partial Y_k} & \frac{\partial h_{1,k}^{FDOA}}{\partial V_{x,k}} & \frac{\partial h_{1,k}^{FDOA}}{\partial V_{y,k}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_{M-1,k}^{FDOA}}{\partial X_k} & \frac{\partial h_{M-1,k}^{FDOA}}{\partial Y_k} & \frac{\partial h_{M-1,k}^{FDOA}}{\partial V_{x,k}} & \frac{\partial h_{M-1,k}^{FDOA}}{\partial V_{y,k}} \end{pmatrix} \tag{1.38}$$

$$\frac{\partial h_{i,k}^{TDOA}}{\partial X_k} = \frac{1}{c}\frac{X_k - X_r}{\sqrt{(X_r - X_k)^2 + (Y_r - Y_k)^2}} - \frac{1}{c}\frac{X_k - X_i}{\sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}} \tag{1.39}$$

$$\frac{\partial h_{i,k}^{TDOA}}{\partial Y_k} = \frac{1}{c}\frac{Y_k - Y_r}{\sqrt{(X_r - X_k)^2 + (Y_r - Y_k)^2}} - \frac{1}{c}\frac{Y_k - Y_i}{\sqrt{(X_i - X_k)^2 + (Y_i - Y_k)^2}} \tag{1.40}$$

$$\frac{\partial h_{i,k}^{FDOA}}{\partial X_k} = -\frac{f_c}{c}\left(Y_k - Y_r\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_r, Y_r)\rangle}{\sqrt{\|(X_k, Y_k) - (X_r, Y_r)\|^3}} + \frac{f_c}{c}\left(Y_k - Y_i\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_i, Y_i)\rangle}{\sqrt{\|(X_k, Y_k) - (X_i, Y_i)\|^3}} \tag{1.41}$$

$$\frac{\partial h_{i,k}^{FDOA}}{\partial Y_k} = +\frac{f_c}{c}\left(X_k - X_r\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_r, Y_r)\rangle}{\sqrt{\|(X_k, Y_k) - (X_r, Y_r)\|^3}} - \frac{f_c}{c}\left(X_k - X_i\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_i, Y_i)\rangle}{\sqrt{\|(X_k, Y_k) - (X_i, Y_i)\|^3}} \tag{1.42}$$

$$\frac{\partial h_{i,k}^{FDOA}}{\partial V_{x,k}} = +\frac{f_c}{c}\left(X_k - X_r\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_r, Y_r)\rangle}{\|(X_k, Y_k) - (X_r, Y_r)\|} - \frac{f_c}{c}\left(X_k - X_i\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_i, Y_i)\rangle}{\|(X_k, Y_k) - (X_i, Y_i)\|} \tag{1.43}$$

$$\frac{\partial h_{i,k}^{FDOA}}{\partial V_{y,k}} = +\frac{f_c}{c}\left(Y_k - Y_r\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_r, Y_r)\rangle}{\|(X_k, Y_k) - (X_r, Y_r)\|} - \frac{f_c}{c}\left(Y_k - Y_i\right)\frac{\langle (V_{x,k}, V_{y,k}), (X_k, Y_k) - (X_i, Y_i)\rangle}{\|(X_k, Y_k) - (X_i, Y_i)\|} \tag{1.44}$$

The a-priori model, in this case, assumes the jammer is moving with a constant velocity.

$$\mathbf{a}_k = \begin{cases} X_{k+1} = X_k + V_{x,k}dt & \forall k \quad (1.45a) \\ Y_{k+1} = Y_k + V_{y,k}dt & \forall k \quad (1.45b) \\ V_{x,k+1} = V_{x,k} & \forall k \quad (1.45c) \\ V_{y,k+1} = V_{y,k} & \forall k \quad (1.45d) \end{cases}$$

where dt is the time interval between two consecutive EKF iterations, in seconds. The state transition matrix $\mathbf{A}_k$ is the Jacobian matrix of $\mathbf{a}_k$:

$$\mathbf{A}_k = J\,\mathbf{a}_k = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.46}$$

Controlled inputs are absent and $\mathbf{B}_k = 0\ \forall k$. The same considerations for the measurement and noise covariance matrices in the power-based case, hold also in

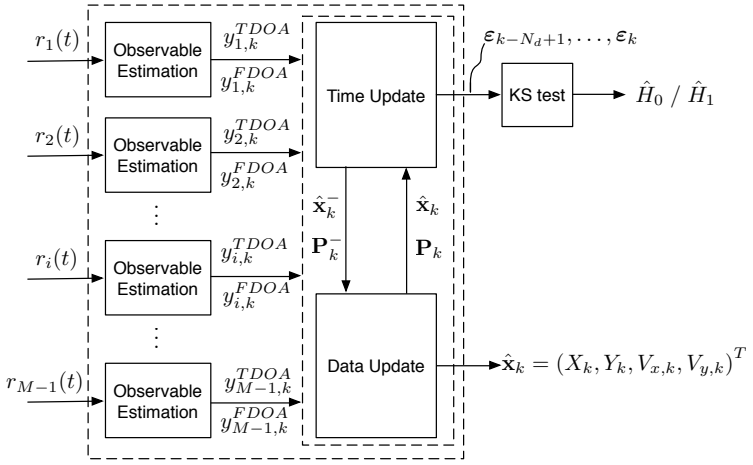

Figure 1.6: TDOA/FDOA-based algorithm block diagram

this context. Allowing different measurement error variances for TDOA and FDOA the covariance matrices are:

$$\begin{cases} \mathbf{Q}_k = \sigma_Q \mathbf{I}_4 & \forall k \quad (1.47a) \\ \mathbf{R}_k = \text{diag}\left(\underbrace{\sigma_{TDOA}^2, \ldots, \sigma_{TDOA}^2}_{M-1}, \underbrace{\sigma_{FDOA}^2, \ldots, \sigma_{FDOA}^2}_{M-1} \right) & \forall k \quad (1.47b) \end{cases}$$

The optimal values for σ_{TDOA} and σ_{FDOA} can be set according to §A.3.

The complete algorithm is summarized by the block diagram in Figure 1.6.

1.7 Jammer detection

At this step, the task is to take a decision about the presence or absence of interference, based on the output error of the EKF. More specifically, we monitor the innovation of the filter, which is the difference between the actual observables and the measurements predicted using the a-priori state estimate and the observation model. The innovation vector $\boldsymbol{\varepsilon}_k \in \mathbb{R}^M$ is defined as:

$$\boldsymbol{\varepsilon}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \tag{1.48}$$

Let $F_0(x)$ be the cumulative density function (CDF) of the squared magnitude of the innovation, in the absence of jamming attempts. This function can be accurately estimated by initially running the algorithm in a controlled interference-free scenario.

$$F_0(x) = \text{Prob}\left\{ \|\boldsymbol{\varepsilon}_k\|^2 \leq x \right\} \tag{1.49}$$

Let $F_{N_D}(x)$ be the empirical CDF of the squared magnitude of the innovation, estimated using the last N_D samples $(\boldsymbol{\varepsilon}_{k-N_d+1}, \ldots, \boldsymbol{\varepsilon}_k)$. If $i = k - N_D + 1$, then the empirical CDF is expressed as:

$$F_{N_D}(x) = \frac{1}{N_D} \sum_{n=i}^{i+N_d-1} \mathbf{1}\left(\|\boldsymbol{\varepsilon}_n\|^2 \leq x\right) \tag{1.50}$$

where $\mathbf{1}(\cdot)$ is the indicator function, defined as

$$\mathbf{1}(Z_i \leq x) = \begin{cases} 1 & \text{if } Z_i \leq x \quad (1.51a) \\ 0 & \text{otherwise} \quad (1.51b) \end{cases}$$

Now it is possible to introduce the decision statistics which yield the main properties and benefits of the proposed algorithm. Once the empirical CDF is available, it is compared with that computed in the absence of jammers by means of the Kolmogorov-Smirnov test (KST):

$$D_{N_D}(d) = \sup |F_{N_D}(x) - F_0(x)| \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} d \tag{1.52}$$

The KST compares the supremum of the difference between the empirical distribution and the distribution of the innovation in the absence of jammer, with a threshold d. The algorithm decides in favour of $\hat{H}_1$ whenever the test statistics D_{N_D} exceeds the threshold; it decides in favour of $\hat{H}_0$ if the test statistics is below the threshold. The desirable property of the KST is that its test statistics do not depend on F_0. In fact, it depends only on the number N_D of samples used in the empirical CDF estimation and on the test threshold d. Once these two parameters are set, the false-alarm probability is constant. This probability is defined as that of the test statistics to exceed the threshold, given that H_0 is true:

$$P_{fa} = \text{Prob}\{D_{N_D} > d \mid H_0\} \tag{1.53}$$

Although no closed form of the false-alarm probability exists, Marsaglia et al. [37] proposed an algorithm to compute this probability. This algorithm becomes more and more accurate by increasing the product $d\sqrt{N_D}$ and reaches accuracies comparable to the machine precision. Figure 1.7 shows the theoretical false alarm probability as a function of the number of samples for estimating the empirical CDF and the KST threshold.

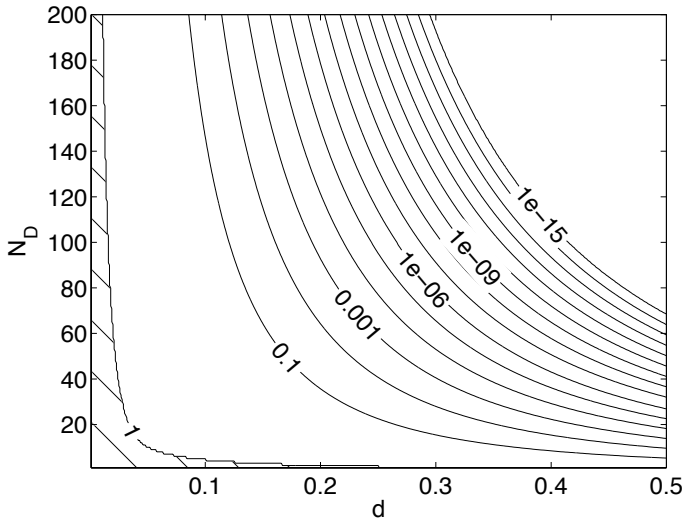

Figure 1.7: Theoretical probability of false alarm from eq. (1.53)

1.8 Numerical results

In our reference scenario, used for performance evaluation, we consider a square SA of size $l \times l$. By selecting the bottom-left corner of this region as the origin of the axes, we arrange the nodes in a grid with coordinates given by the Cartesian product of the following linearly-spaced sets:

$$\left\{\underbrace{0,\ldots,l}_{\sqrt{M}}\right\} \times \left\{\underbrace{0,\ldots,l}_{\sqrt{M}}\right\},\ \sqrt{M} \in \mathbb{N}_{>0}, \tag{1.54}$$

which results in the node deployment of Figure 1.8. Note that the algorithm performance does not depend critically on regularity, but on achieving sufficiently small dilution of precisions. The experiment starts in the absence of interference. At the time instant t_J, a jammer starts to transmit a chirp signal from a random and unknown position within the SA and keeps transmitting until the experiment ends at instant t_N. In the TDOA/FDOA-based algorithm, the jammer moves with constant velocity, towards a random direction. The performance of the algorithm is evaluated with Monte Carlo simulations. Simulation parameters are collected in Tables 1.1-1.2. Jammer detection capabilities are evaluated in terms of false-alarm rate ($\hat{P}_{fa}$), missed-detection rate ($\hat{P}_{md}$), and detection delay ($\hat{\delta}$). For the localization, the chosen metrics are the root mean square position (and velocity) error ($\widehat{RMSE}$ and $\widehat{RMSE_V}$)). False-alarm (missed-detection) rate is measured as the ratio be-

Table 1.1: Simulation parameters (power-based algorithm)

Symbol	**Description**	**Value**
T_c	Chirp period	$50\,\mu$s
B_c	Chirp bandwidth	5 MHz
α	Path loss exponent	2
P_t	Jammer transmit power	-20 dBW
f_c	Jammer carrier frequency	1575.42 MHz
B_r	Node receiver bandwidth	10 MHz
R_s	Node receiver sampling rate	20 Msps
P_N	Node receiver noise power	-127 dBW
σ_Q	EKF process noise standard deviation	10^-6
σ_R	EKF measurement noise standard deviation	1
N	Number of samples used for power estimation	100
t_N	Experiment duration (EKF iterations)	20000
t_j	Jammer starting instant (EKF iterations)	10000

Table 1.2: Simulation parameters (TDOA/FDOA-based algorithm)

Symbol	Description	Value
T_c	Chirp period	50 μs
B_c	Chirp bandwidth	5 MHz
α	Path loss exponent	2
P_t	Jammer transmit power	-20 dBW
f_c	Jammer carrier frequency	1575.42 MHz
B_r	Node receiver bandwidth	10 MHz
R_s	Node receiver sampling rate	20 Msps
P_N	Node receiver noise power	-127 dBW
σ_Q	EKF process noise standard deviation	10^-6
σ_R	EKF measurement noise standard deviation	1
N	CAF observation interval	1 ms
V	Jammer velocity	50 km/h
t_N	Experiment duration (EKF iterations)	20000
t_j	Jammer starting instant (EKF iterations)	10000

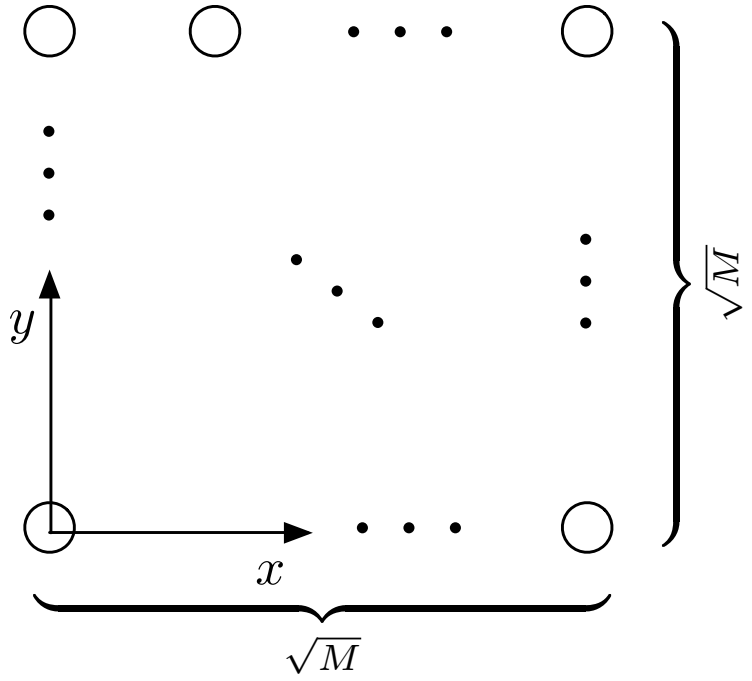

Figure 1.8: Node positions in the service area

tween the number of false-alarm (missed-detection) events over the total number of experiments, as indicated in (1.55-1.56)

$$\hat{P}_{fa} = \frac{\#\text{false alarm events}}{\#\text{experiments}} \approx \text{Prob}\left\{\hat{H}_1 \mid H_0\right\} \tag{1.55}$$

$$\hat{P}_{md} = \frac{\#\text{missed detection events}}{\#\text{experiments}} \approx \text{Prob}\left\{\hat{H}_0 \mid H_1\right\} \tag{1.56}$$

The detection delay is the time difference between the instant when interference ($\hat{t}_J$) is detected and the actual instant when the jammer starts transmitting (t_J), measured in EKF epochs, as follows:

$$\hat{\delta} = \hat{t}_J - t_J \tag{1.57}$$

Both $\widehat{RMSE}$ and $\widehat{RMSE}_V$ are measured only when the jammer is transmitting, as displayed according to the following expression:

$$\widehat{RMSE} = \sqrt{\sum_{k=t_J}^{t_N} \frac{\|X - \hat{X}_k, Y - \hat{Y}_k\|^2}{t_N - t_J}} \tag{1.58}$$

$$\widehat{RMSE}_V = \sqrt{\sum_{k=t_J}^{t_N} \frac{\|V_x - \hat{X}_{x,k}, V_y - \hat{Y}_{y,k}\|^2}{t_N - t_J}} \tag{1.59}$$

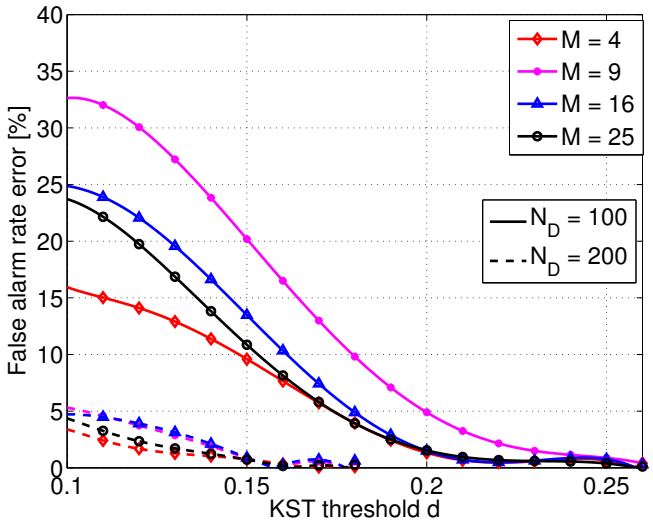

Figure 1.9: Error between theoretical false-alarm probability and measured false-alarm rate

Detection results are valid for both versions of the algorithm. On the contrary localization results are shown separately for the power-based and TDOA/FDOA-based algorithms.

1.8.1 Detection performance

The first significant result worth analyzing is the behavior of the false-alarm rate compared to the corresponding theoretical probability. Figure 1.9 shows the error between the theoretical false-alarm probability and the measured rate ($|\hat{P}_{fa} - P_{fa}|/P_{fa}$) as a function of the KST threshold d, by varying the number of nodes M and the number N_D of innovation samples used for the CDF estimation. From the figure, we may notice that for increasing N_D and d, the error decreases, as indeed suggested in [37]. The maximum error of 32% is obtained for $d = 0.1$ and $N_D = 100$ and it is located in a region where the false-alarm probability is too high (of the order of 0.1) to be of practical interest (see Figure 1.7). In the region of practical interest ($P_{fa} < 10^{-3}$), the error falls rapidly below 1%. Moreover, the fact that the error does not depend significantly on the number of sensor nodes M is remarkable. In fact, as highlighted before, false-alarm probability essentially depends on N_D and d. This property is very important, as it possibly allows for designing a jammer-detecting network of nodes with CFAR characteristics, by selecting the appropriate values of

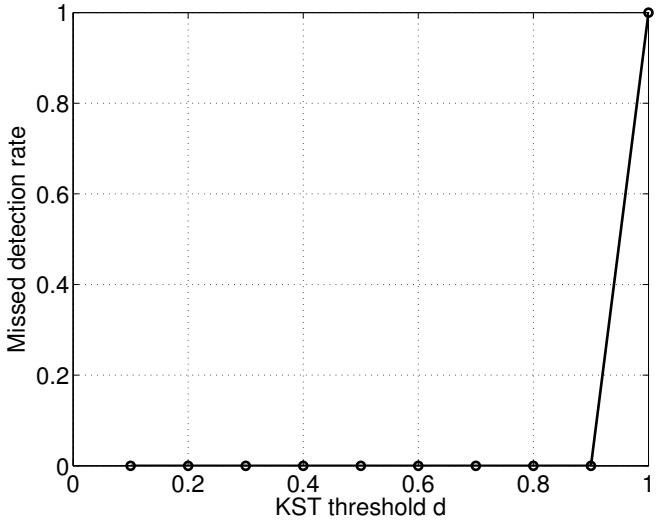

Figure 1.10: Missed detection rate

N_D and d.

Regarding missed detection probability, due to the intrinsic characteristics of the decision metric, it presents a step-wise behaviour for growing values of the threshold, so that it is null for all practical means, as shown in Fig. 1.10. About the accuracy of Figures 1.9-1.10, the curves have been obtain by repeating the reference experiment until 1000 false-alarms/missed-detection events occurred with the maximum number of iterations set to 10000. The final result is the average of these multiple realizations

The choice of N_D and d has an obvious impact on the detection delay. Figure 1.11 shows this dependence. However, since practical applications can often tolerate delays up to seconds, N_D and d are not critical design parameter, as far as delay is concerned. Moreover, the delay is fairly linear with the KST threshold, and the slope of the lines is exactly N_D. As a consequence, the delay can be described by (1.60).

$$\delta = \delta_0 + N_d d \tag{1.60}$$

1.8.2 Localization performance

The localization performance of the power-based algorithm is compared with the square root of the Cramér-Rao lower bound (SR-CRB) (§A.3). Figure 1.12 shows the RMSE as a function of the number of nodes. The localization RMSE is limited to $0.2\,m$ with $M = 4$ and approaches the SR-CRB by increasing the number of

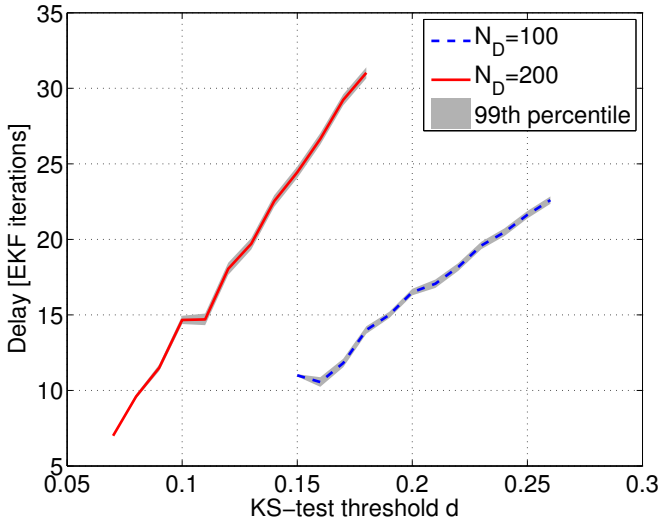

Figure 1.11: Detection delay

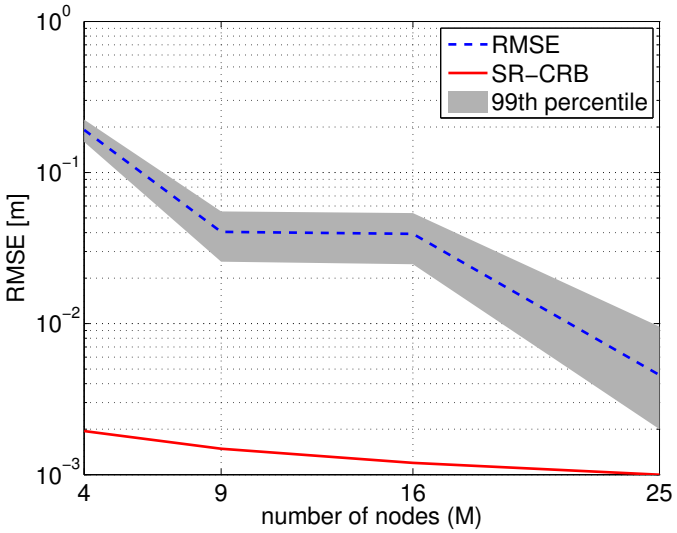

Figure 1.12: Localization root mean square error (power-based algorithm)

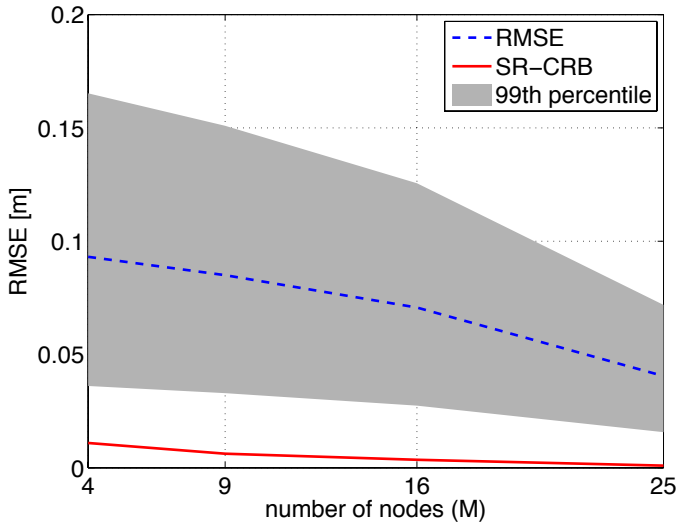

Figure 1.13: Localization root mean square error (TDOA/FDOA-based algorithm)

monitoring nodes.

One may argue that the algorithm has not been tested in a dynamic scenario with a moving jammer, which is a quite usual situation. Actually, the velocity of the jammer does not compromise the performance of the algorithm. Indeed, in this case, the only problem that may rise is the underestimation of the received power caused by the Doppler shift, which could shift part of the signal spectrum outside of the receiver band. However, in order for this to happen, the frequency shift has to be dramatic, in the order of hundreds of kHz, which corresponds to to unrealistic speeds.

In the TDOA/FDOA-based version of the algorithm, we compare $\widehat{RMSE}$ and $\widehat{RMSE}_V$ with the lower bounds in Figures 1.13-1.14. We obtain similar results to the power-based case, but we are able to estimate the jammer velocity as well.

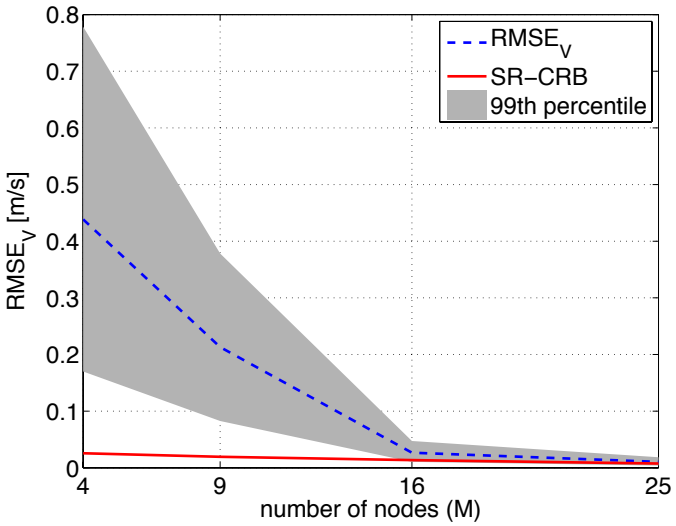

Figure 1.14: Velocity estimation root mean square error (TDOA/FDOA-based algorithm)

1.9 TDOA/FDOA-based I-JJDL on low-cost SDR

As a proof of concept, this section describes the implementation of the TDOA/FDOA-based I-JJDL algorithm using low-cost SDR. The experiment has been carried out in the RF System and Payload laboratory, ESTEC (Figure 1.15) . A diagram of the test setup is shown in Figure 1.16. The jammer is emulated using a HackRF (see §XX), which transmits a chirp signal with a band of 20MHz, a chirp period of 50 us, and a transmit power of 1 mW. The interfering signal is fed to a RF power divider and then to a SPIRENT VR5 HD Spatial Channel Emulator. The SPIRENT VR5 emulates the scenario in Figure 1.17.

Four HackRF have been used to emulate the four sensor nodes. Each HackRF is synchronized in frequency using a high accuracy clock. The time synchronization is provided thanks to two mass market GNSS receivers. The sampling frequency of the sensor node is 10 MHz, the receiver bandwidth is 5 MHz, and the CAF integration interval is 1 ms. The experiment has been repeated 1000 times and, for each realization each node records 200 ms. The jammer starts transmitting at a random instant within this 200ms interval. The signals have been processed using the TDOA/FDOA-based algorithm described in the present chapter.

Figure 1.18 shows the probability density function (PDF) of the localization error. The localization error is limited to 25 m, and the mode is 4 m. This error is the effect

Figure 1.15: Laboratory setup

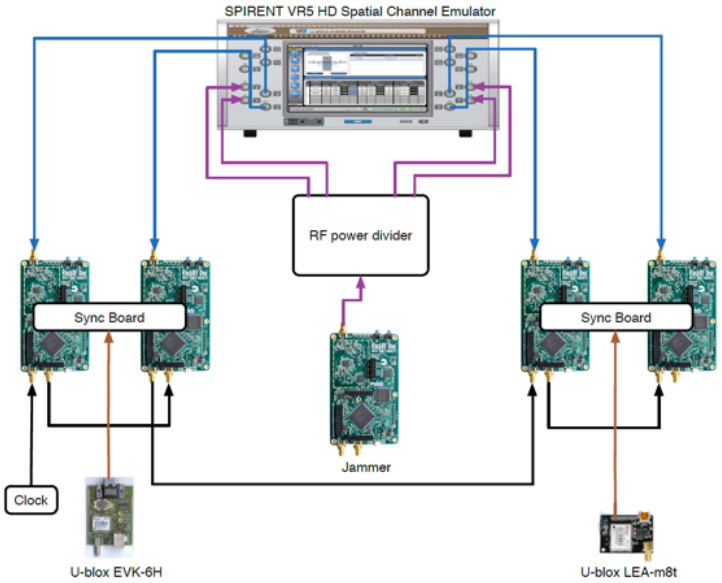

Figure 1.16: Diagram of the test setup

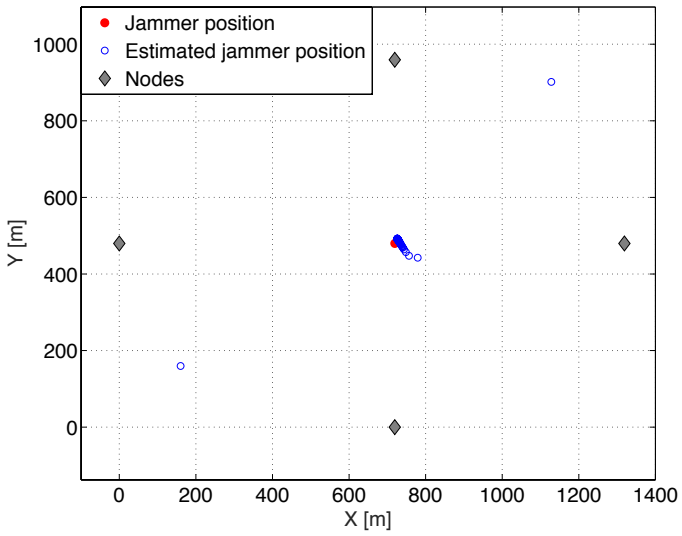

Figure 1.17: Emulated test scenario

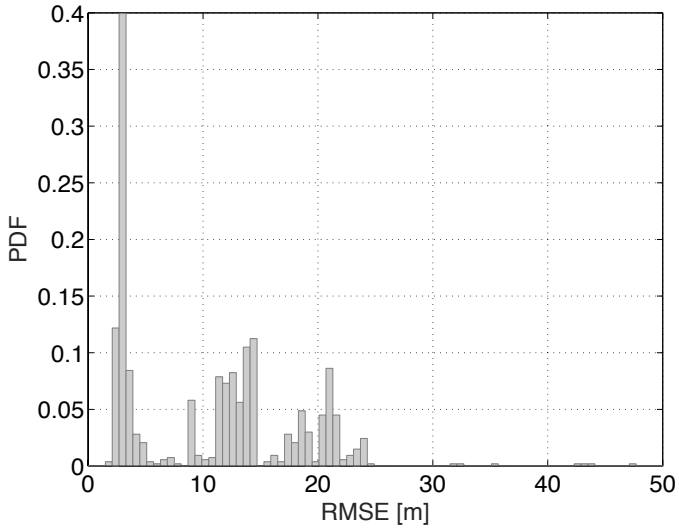

Figure 1.18: Experimental root mean square localization error

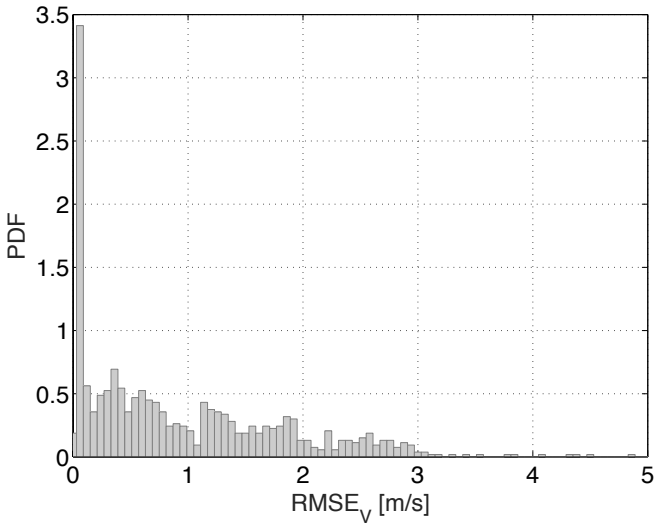

Figure 1.19: Experimental root mean square velocity estimation error

of the residual synchronization error (100 ns) on the TDOA estimation. Although the residual synchronization error introduces a bias in the EKF observables (30 m), the algorithm is still able to provide a sufficient accuracy for the jammer localization application.

Figure 1.19 shows the probability density function (PDF) of the velocity estimation error. This error is limited to 3 m/s and the mode is below 1 m/s. The velocity estimation shows a better accuracy because all the sensor nodes share the same clock.

Chapter 2

Interference Mitigation

In this chapter, we consider two possible approaches to the interference mitigation problem. The first approach is described by the distributed-sensing waveform estimation (DSWE) algorithm which allows for interference cancellation in time domain, thanks to cooperative multi-node interfering waveform estimation [P8].

The second approach does not involve interference cancellation, but achieves interference mitigation thanks to integration of GNSS and inertial navigation system (INS). Notwithstanding this method has been devised in order to improve the tracking accuracy in racing vehicles, it can be successfully applied in order to improve reliability, accuracy and availability of the position solution in interfered scenarios [P5].

2.1 Distributed-sensing waveform estimation

Several techniques have been proposed for wide-band interference characterization and mitigation. Traditional time-domain interference mitigation technique, such as pulse blanking are ineffective against chirp jammers [38]. The most promising techniques are the transformed-domain ones. Borio et al. proposed time-frequency analysis for interference excision [16]; Wavelet transform has been used for the same purpose in [39]; The first attempts at using Karhunen-Loève transform for interference mitigation appeared in [40] and [41]. However, none of the aforementioned techniques exploits the cooperation of multiple users for interference characterization and mitigation.

Here, we present the DSWE algorithm based on Karhunen-Loève expansion (KLE). Karhunen-Loève Expansion (§A.4) (sometimes denoted as Karhunen-Loève

Transform or Principal Component Analysis) allows to represent a signal, in the proper vector space, such that noise and redundancy are removed. As an intuitive example, it is possible to consider an object that is moving on a straight line in a three-dimensional Euclidean space. Suppose three cameras, one for each orthogonal Cartesian direction, are observing the experiment. It is intuitive stating that the best representation of the object's trajectory would be the one recorded by the camera which is orthogonal to the straight line on which the object is moving: even if the object's motion takes place in a three-dimensional space, we are interested just in what happens in a single direction, all the information is contained in a single direction. Even if there is no camera in the privileged direction, by combining in the appropriate way the contributions of the three cameras, it is always possible to represent the trajectory in a single direction. In a similar way, we apply this idea to waveform estimation. The proposed algorithm, applied to interfering signals, provides accurate waveform estimates suitable for interference cancellation. Moreover, the estimation algorithm is robust against channel model mismatch and offers superior performance, even in the presence of very low SNR.

After waveform estimation, each node is able to cancel the effect of interference, by coherently subtracting the disturbing signal, after frequency and phase estimation (obtainable using traditional non data aided estimation). Notwithstanding the algorithm has been tested on chirp signals, it is meant to be applied to any wide-sense stationary signal, that is basically most of modulated signals.

2.1.1 System model

We consider a network of M waveform estimating nodes placed on a grid, as shown in 2.1. We refer to the region occupied by nodes as service area.

Assume that a jammer is transmitting an interfering signal within the service area. The network of nodes, by exploiting the structure of the jamming signal, must provide an estimate of the interfering signal waveform, suitable for interference cancellation.

2.1.2 Algorithm description

The DSWE algorithm exploits the structure of interfering signal in order to characterize them. The waveform estimation algorithm is composed by three main phases:

(i) Period estimation

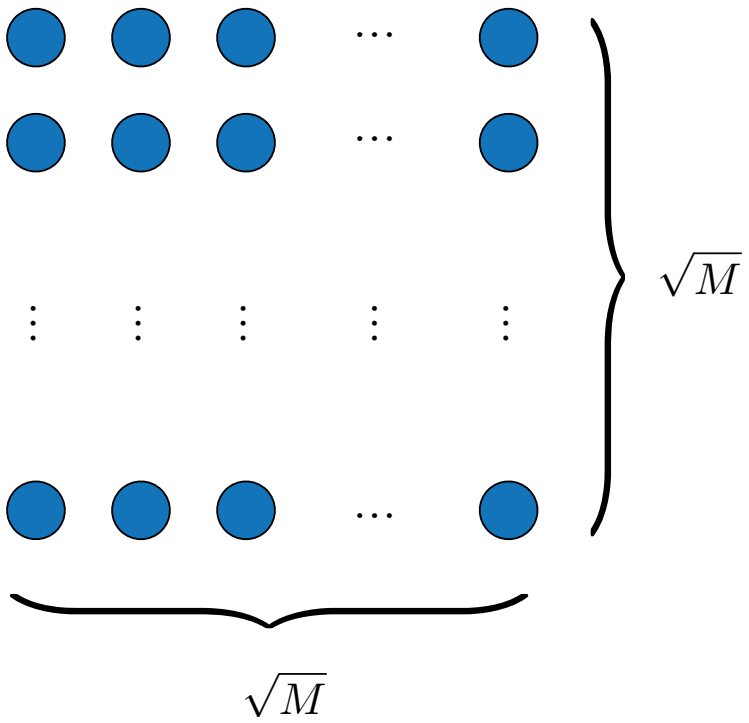

Figure 2.1: Grid of waveform estimating nodes

(ii) Averaging

(iii) Interfering signal reconstruction

The first two steps are performed by nodes, while the third one is performed by a FC, which collects the data from nodes. In the following it is assumed that each node records the received interfering signals in an observation interval T_O, with a sampling frequency f_s.

Period estimation

The objective of the first step is to provide an estimate of the duration of the interfering signal period: this is obtained thanks to the autocorrelation properties of the signal. The autocorrelation function of the interfering signal is periodic and its repetition period is equal to the period of $z(t)$. Figure 2.2 shows a block diagram of the period estimation algorithm. Firstly, the autocorrelation function of the

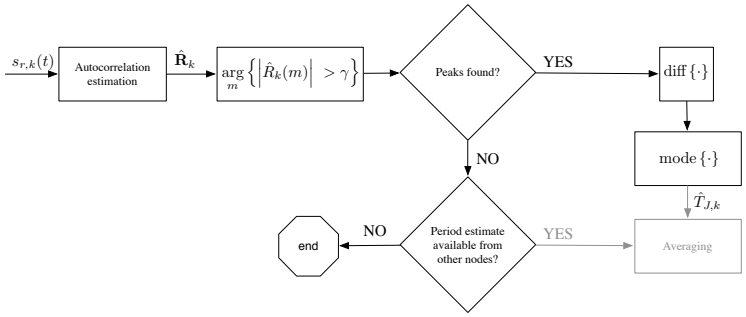

Figure 2.2: Block diagram of the period estimation phase

interfering signal is estimated:

$$\hat{R}_k(m) = \begin{cases} \dfrac{1}{\sqrt{\hat{R}_k(0)}} \displaystyle\sum_{n=0}^{N-m-1} s^*_{r,k}(n+m) s_{r,k}(n) & m \geq 0 \quad (2.1a) \\ \hat{R}_k(-m) & m < 0 \quad (2.1b) \end{cases}$$

After the autocorrelation is estimated, the correlation peaks are found by comparing the modulus of $\hat{R}_k$ with a threshold γ, then the distance between any two consecutive peaks is computed and the most probable distance is selected as the estimated period $\hat{T}_{j,k}$, where k indicates the k-th node. Once the firts node has estimated the period, the value of $\hat{T}_j$ is broadcast to all the nodes.

Averaging

In the second step, the average waveform in a period is computed:

$$\bar{s}_{r,k} = \frac{1}{\lfloor N_s/\hat{T}_j \rfloor} \sum_{n=1}^{\lfloor N_s/\hat{T}_j \rfloor} s_{r,k}\left(t - n\hat{T}_j\right) e^{-j\hat{\phi}_n} \quad (2.2)$$

where N_s is the number of samples per observation interval $N_s = T_O f_s$. In order to average out noise and boost the SNR, it is important to sum coherently the samples of the recorded waveform. To this aim, an open loop phase estimator is used, allowing the correction of the waveform's phase ϕ_n. Moreover, the expected value of the waveform is removed, in order to simplify the equations in the following steps, without loss of generality. The operations performed by each node in phase (ii) are depicted in Figure 2.3.

2.1.3 Interference signal reconstruction

The averaged samples of the interfering signal waveform $\bar{s}_{r,k},\ k = 1, \ldots, M$ are sent to the FC, which aims to reconstruct the transmitted modulating signal waveform

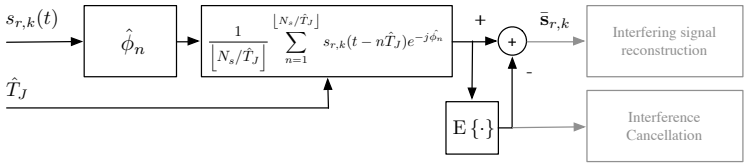

Figure 2.3: Averaging phase block diagram

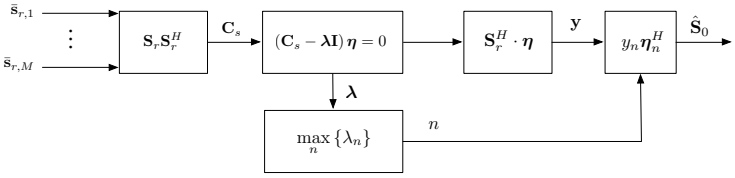

Figure 2.4: Fusion center block diagram

$z(t)$, using KLE (§A.4). First, the averaged samples of the signal waveform recorded by nodes, are collected in the matrix $\mathbf{S}_r \in \mathcal{M}_{M\times N_s(\mathbb{R})}$, then the signal autocovariance matrix $\mathbf{C}_s \in \mathcal{M}_{N_s}(\mathbb{R})$ is estimated by:

$$\mathbf{C}_s = \mathbf{S}_r\mathbf{S}_r^H \tag{2.3}$$

where $(\cdot)^H$ is the Hermitian operator. The eigenvalues $\lambda = (\lambda_1, \ldots, \lambda_{N_s})^T \in \mathbb{C}^{N_s}$ and the corresponding eigenvectors $\boldsymbol{\eta} \in \mathcal{M}_{N_s}(\mathbb{C})$, are computed by diagonalization of the matrix $\mathbf{C}_s$. Then the KLE coefficients are obtained by projecting the signal space onto the eigenvector space:

$$\mathbf{y} = \mathbf{S}_r^H \cdot \boldsymbol{\eta} \tag{2.4}$$

and the eigenvalue with maximum magnitude is determined

$$\bar{\lambda}_{max} = \max_{n=1,\ldots,N_s}\{\lambda_n\} \tag{2.5}$$

The interfering signal modulating waveform samples $\hat{\mathbf{S}}_0$, are reconstructed using the eigenvector corresponding to the eigenvalue with maximum amplitude only:

$$\hat{\mathbf{S}}_0 = y_n\eta_n^H \tag{2.6}$$

thus, reducing the number of dimensions of the signal space to 1. By tracking the received signal frequency and phase, it is possible to subtract the interfering signal waveform from the received signal in time domain. The interference reconstruction phase is summarized in Figure 2.4

Table 2.1: Simulation parameters (Distributed-sensing waveform estimation algorithm

Symbol	Description	Value
P_t	Jammer transmit power	20 dBm
α	Path-loss exponent	3
f_d	Maximum Doppler frequency	600 Hz
N_0	AWGN power spectral density	-196 dBW/Hz
B	Receiver bandwidth	5 MHz
f_s	Sampling frequency	16 MHz
f_c	Jamming signal carrier frequency	1575.42 MHz
ϵ	Maximum tolerated MSE	0.5

2.1.4 Numerical results

In order to evaluate the algorithm performance, we used Monte-Carlo simulations. Before commenting the simulations results, we define a waveform estimation success probability P_S as the probability of a correct period and waveform estimation, that is the probability that at least one node is able to estimate the period and the mean square error (MSE) between the transmitted interfering waveform and the estimated one is below the maximum tolerated MSE, ϵ:

$$P_S = \text{Prob}\left\{\left|\hat{T}_{j,k} - T_j\right| < \frac{1}{f_s} \wedge \text{MSE}[\hat{\mathbf{S}}_0, \mathbf{S}_0] < \epsilon\right\} \tag{2.7}$$

MSE, in the following, is computed on normalized signals. In order to evaluate the success probability, we used a network of M nodes, as shown in Figure 2.1, placed on a 1 km $\times$ 1 km square region. The jamming source position is randomly chosen with uniform distribution within the service area and an estimate of the interfering signal waveform is obtained thanks to the algorithm described in the previous section. The results presented in the remainder of this section are obtained by averaging ten thousands repetition of the same experiment, in order to have statistically valid results. Simulation parameters are summarized in Table 2.1. A generic node k receives the interfering signal transmitted by the jammer through a time-varying flat-fading channel, with impulse response h. The signal received by the $k-$th node

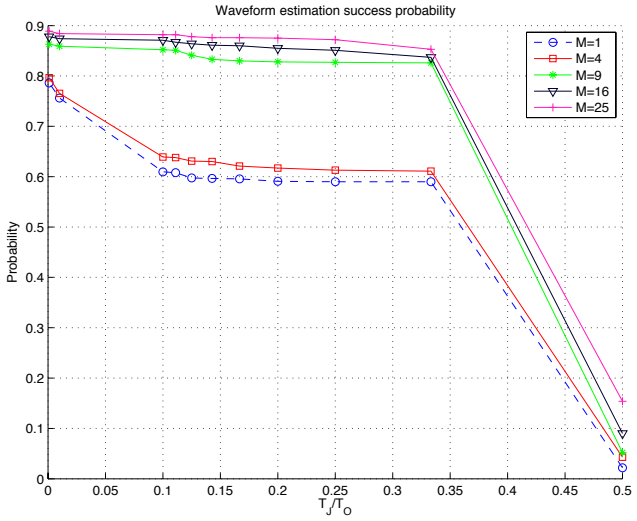

Figure 2.5: Waveform estimation success probability (AWGN)

can be written as

$$s_{r,k}(t) = s(t) * h(t,\tau) \tag{2.8}$$

where $*$ denotes the convolution operator. Assuming two-dimensional isotropic scattering and vertical monopole antennas at the nodes, the PSD fo the received fading signal can be modelled as in [42]:

$$S(f) = \begin{cases} \dfrac{1}{\pi f_d \sqrt{1-\left(\frac{f}{fd}\right)^2}} & |f| < f_d \quad (2.9a) \\ 0 & \text{elsewhere} \quad (2.9b) \end{cases}$$

In order to understand the impact of a dynamic channel on performance, we tested the algorithm both in static and dynamic channel conditions: the reference algorithm performance is obtained in AWGN. With this simple model, the received signal is an attenuated and delayed version of the transmitted one:

$$s_{r,k}(t) = s(t-\tau_k)\sqrt{\frac{K}{d_k^{\alpha}}} \tag{2.10}$$

where d_k is the distance between the k-th node and the jamming source, α is the path-loss exponent and K is the path-loss at the reference distance. The performance in AWGN, are shown in Figures 2.5-2.6 Figure 2.5 shows the waveform estimation success probability dependence on the ratio between period and observation interval duration T_j/T_O , for different network dimensions M. The success

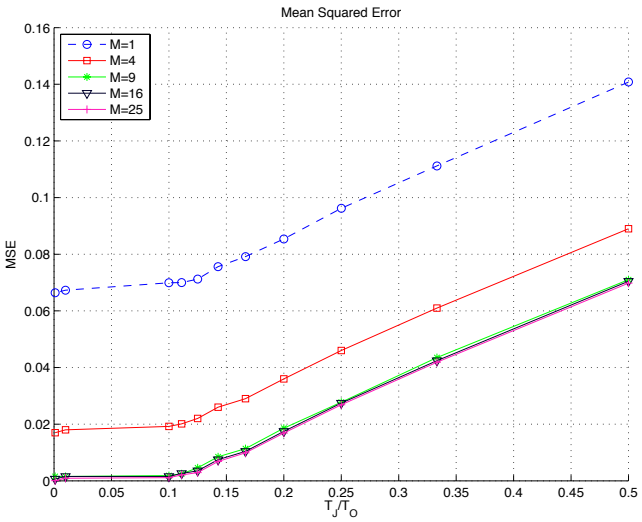

Figure 2.6: mean squared error between transmitted and estimated waveforms (AWGN)

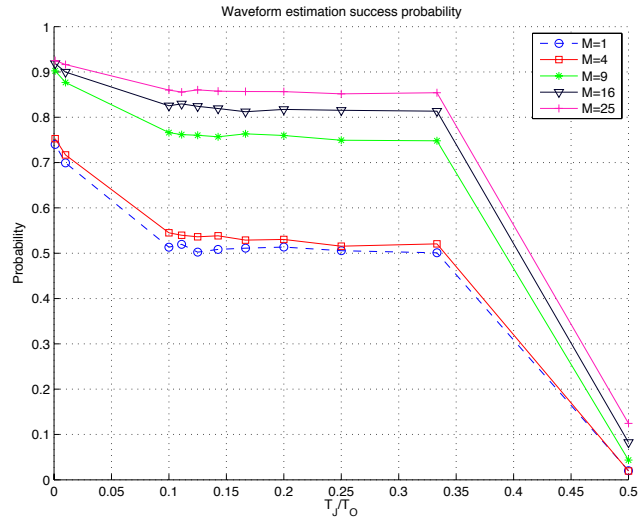

Figure 2.7: Waveform estimation success probability (dynamic channel)

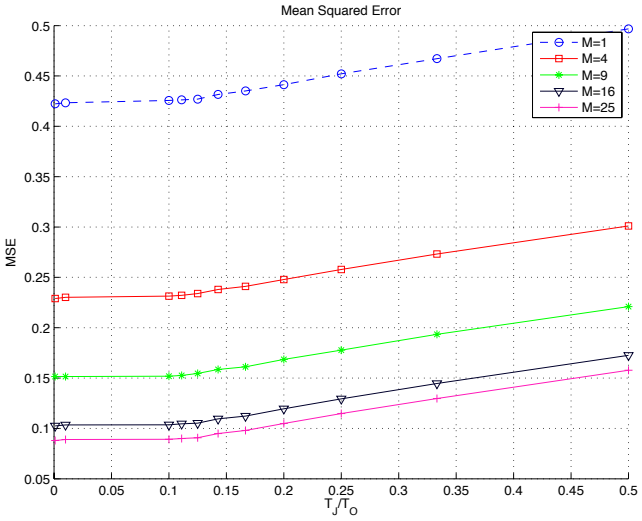

Figure 2.8: mean squared error between transmitted and estimated waveforms (dynamic channel)

probability decreases when increasing T_j/T_O, and we have $P_S > 0.8$ for $M > 9$ and $T_j/T_O < 3$: the observation of just 3 chirp periods is sufficient to obtain very good performance. For $M < 9$, we have a lower success probability and we need to record hundreds of chirp periods to have success probabilities better than 70%. As far as MSE is concerned, we have outstanding performance, as shown in Figure 2.6: MSE is always below 0.15. Its behavior is decreasing for decreasing values of T_j/T_O and for increasing size of the network. A saturation effect is present: increasing the network size beyond $M = 9$ does not provide enough benefits to justify the higher network complexity.

Then the algorithm has been tested using the dynamic channel and simulation results are collected in Figures 2.7-2.8 . The results show a similar behavior for both channels, with performance degradation in terms of both success probability and MSE, in the dynamic channel scenario. Thus, we can draw the same conclusions as in the AWGN case. Figure 2.7 shows the waveform estimation success probability in the dynamic channel case: for low values of T_j/T_O ($\leq 1/100$) and for $M \geq 16$, the success probability is analogous to the AWGN case but, for increasing value of the ratio between period duration and observation interval, we have a stronger performance degradation. For $T_j/T_O < 1/100$, the success probability in the dynamic channel case is 5-8% worse with respect to the AWGN scenario. Talking about MSE and looking at Figure 2.8, we notice a stronger performance degradation: while in AWGN, mean

squared error can be lower than 0.01 for $M = 25$ and $T_j/T_O < 1/10$, we can not reach values below 0.08 in the dynamic channel case. Moreover, the curves associated to different values of M are more distant from each other, meaning that increasing the network size gives more consistent benefits in the dynamic channel scenario. The DSWE algorithm improves MSE from values greater than 0.4 to values lower than 0.08. This means that, using our algorithm we can obtain a very low residual after interference cancellation. Notwithstanding the moderate performance degradation with respect to the MSE case, the algorithm shows outstanding performance even in presence of mobility.

2.2 Optimal EKF for Quasi-Tightly Coupled GNSS/INS Integration

As many emerging automotive applications (e.g. autonomous navigation, dynamic control, driving assistance, route guidance, etc.) rely on localization services, enhanced reliability of vehicles positioning is becoming a crucial constraint. However, requirements in terms of both accuracy and availability turn out to be challenging in typical vehicular scenarios. The position solution is indeed potentially subject to harsh electromagnetic environments (e.g. urban canyons, interference), where ranging signals are hindered or disturbed, and to large dynamics whenever high speeds occur. With an increasing number of sensors being made available in modern vehicles, large amounts of data can be used to aid in the localization process together with satellite systems. Because of their complementary features, GGNSS are often employed together with INS in order to improve the localization reliability [43]. As well known, the satellites constellation provides unbiased and repeatable position estimates, which are however available at slow rates and with limited precision. On the other hand, inertial sensors (such as accelerometers, magnetometers, and gyros) continuously allow for position updates at high rates, but their inner drift grow progressively into potentially unbounded errors. With the aim of balancing out these drawbacks, the received GNSS information is exploited to periodically calibrate the INS, thus preventing the overall accuracy from degrading with time. As a result, this fusion effectively increases the average performance of a navigation system over long periods as well as its robustness to temporary service blockage or outage.

In this section, we present a navigation technique that enables the tracking of vehicles moving at high speeds with limited complexity. In particular, we consider the

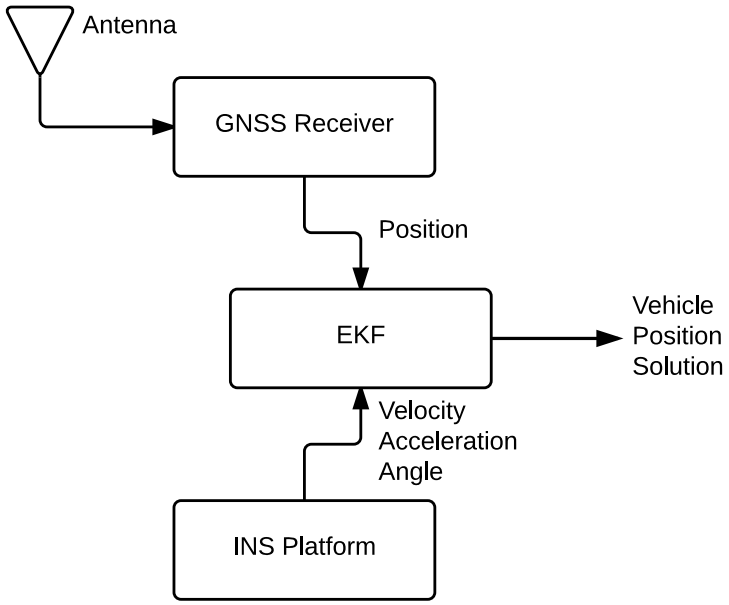

Figure 2.9: Quasi-tight GNSS/INS integration

development of a precise tracking system for Formula Society of Automotive Engineers (SAE) prototypes as our exemplary application. For this purpose, we resort ot a particular quasi-tight GNSS/INS integration [44] that blends the final positioning estimate computed by the receiver and several raw measurements collected by an inertial platform in a closed-loop form. Unlike more complex tight and ultra-tight couplings [45], this method works with no modification of the GNSS engine, since it combines given coordinates and sensing data in a single solution, as shown in Fig. 2.9. At the same time, a configuration of this kind can potentially achieve a higher accuracy than a simpler loose level of integration, which is instead characterized by a weighted average of two independent solutions. In brief, the system we target is capable of performing well with moderate complexity in order to be within the reach of a low-cost implementation. At this regard, we carry out a specific quasi-tight coupling by adopting suitable deterministic description of the model that estimates the vehicle position in real-time. The mathematical formulation we propose represents a novel realization of this integration.

An EKF is used to merge GNSS and INS information, as velocities and accelerations of the tracked vehicle are described through time-variant relationships that are nonlinear. Moreover, we address the optimization of the stochastic properties of the mathematical model underlying the estimation process. The noise covariance

matrices are indeed tuning parameters of the Kalman filter and their identification increases the quality of the state estimation. Our goal is therefore to achieve a quasi-optimal performance of the EKF. For this task, we choose the Bayesian approach proposed in [46] by P. Matisko and V. Havlena, since it can provide more information about the entering noise parameters than other procedures reported in the literature (e.g. Autocovariance Least Squares [47]). This technique numerically approximates the probability distributions of noise matrices from output measurements along an important sampling method. Even though the formulation of the principle underlying this approach results straightforward, its implementation was once impractical due to its demand for high computational power. Nowadays, the processing requirements of this Bayesian method can be handled even by entry-level devices. In [46], its computation complexity and time consumption are compared to those given for Autocovariance Least Squares, highlighting the obtained advantages in terms of memory saving. Hence, a real-time execution of such a recursive algorithm is then expected to be already feasible into current GNSS/INS like the proposed quasi-tight integration.

2.2.1 Deterministic model

The deterministic model we use to describe the motion of the vehicle under test is based on the following assumptions. First of all, the dynamic physical quantities identifying the current state (i.e. position, velocity, and acceleration) are related to a so-called main two-dimensional reference system XY with the Y axis pointing to the North of a geodetic system east-north-up (ENU), as illustrated in Figure 2.10. Hence, the vehicle is supposed to move on a flat surface, as the Z axis corresponding to the Up direction is not taken into account. Such an hypothesis proves to be realistic as we consider the scenario of a typical Formula SAE racetrack, which usually does not feature either significant altitude differences or large dimensions. The curvature of the earth in each point may be consequently neglected. Further, the orientation of the motion with respect to the main axes is taken into account by defining a two-dimensional *body* reference system $x_{body}y_{body}$, which is in-built with the vehicle and the inertial platform. This has a tangential y_{body} axis along the trajectory and a perpendicular x_{body} axis outgoing from the right side. The instantaneous rotation angle between Y and y_{body} is denoted by θ and its measure is available thanks to the magnetometer on board. Before starting the simulation, when the vehicle is still stationary, we assume to align the two reference systems through the initial value θ_0.

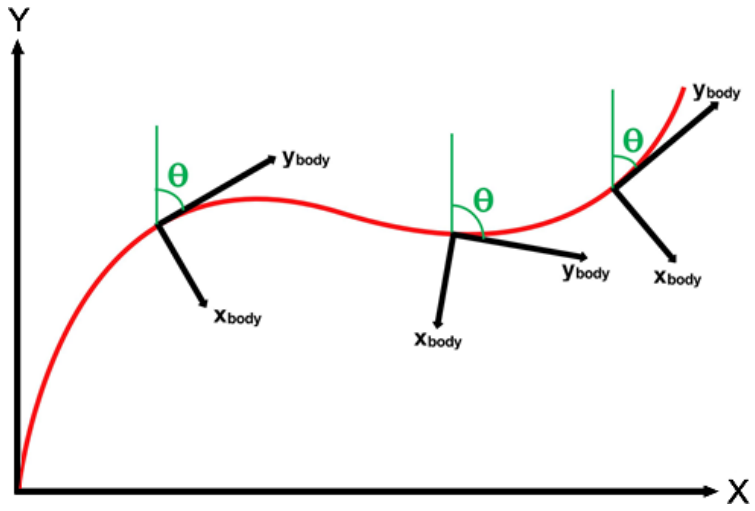

Figure 2.10: Main and body reference systems

Without loss of generality, we also set the first position coordinates into the origin of the main reference system and we finally sketch the vehicle size as a point-like particle. With the goal of testing the situations that may occur in classic Formula SAE (FSAE) racetracks, we design some basic types of motions for our particle-like vehicle:

1. straight lines with uniform acceleration;
2. quick turns at constant velocity;
3. circular hairpins at constant velocity;
4. slaloms at constant velocity.

These trajectories can be generated separately or rather be assembled together to arrange a whole circuit.

The vehicle dynamics are modelled according to the kinematic equations of a point-like particle motion over time. They are described through several time-varying interdependent variables: the coordinates (X, Y) and the ground speed fv_{GNSS} read from the GNSS receiver with respect to the main reference system and the velocity magnitude v_{INS} and the acceleration components $(a_{x_{body}}, a_{y_{body}})$ respectively measured by a gyroscope and an accelerometer in the body reference system.

Since the task of the Kalman filter is a position reckoning as accurate as possible, it suitably combines redundant sequential observations to average out the estimate

errors. For this reason, in addition to the position coordinates of interest, the *actual* state vector $\mathbf{x}$ includes also four variables related to the axial velocities and accelerations, which aid the estimation process, since they contain useful positional information that are extracted through the deterministic model. Hence, according to a discrete-time formulation, at each k-th iteration the state has six components that are all related to the main reference system XY:

$$\mathbf{x}_k = (X_k, Y_k, V_{x,k}, V_{y,k}, a_{x,k}, a_{y,k})^T \tag{2.11}$$

The a-priori EKF model considers the vehicle as a point moving with constant acceleration:

$$\mathbf{a}_k = \begin{cases} X_{k-1} + v_{x,k-1}dt + \dfrac{1}{2}a_{x,k-1}dt^2 & \forall k \quad (2.12a) \\ Y_{k-1} + v_{y,k-1}dt + \dfrac{1}{2}a_{y,k-1}dt^2 & \forall k \quad (2.12b) \\ v_{x,k-1} + a_{x,k-1}dt & \forall k \quad (2.12c) \\ v_{y,k-1} + a_{y,k-1}dt & \forall k \quad (2.12d) \\ a_{x,k-1} & \forall k \quad (2.12e) \\ a_{y,k-1} & \forall k \quad (2.12f) \end{cases}$$

Where dt is the time interval between two consecutive EKF iterations. Hence, the state transition matrix is

$$\mathbf{A}_k = J\,\mathbf{a}_k = \begin{pmatrix} 1 & 0 & dt & 0 & \frac{1}{2}dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & \frac{1}{2}dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.13}$$

Controlled inputs are absent and $\mathbf{B}_k = 0\ \forall k$. In this case, the equation relating the system state to the measurements is:

$$\mathbf{h}_k = \begin{pmatrix} X_{GNSS,k} \\ Y_{GNSS,k} \\ V_{GNSS,k} \\ V_{INS,k} \\ a_{x,INS,k} \\ a_{y,INS,k} \end{pmatrix} = \begin{pmatrix} X_k \\ Y_k \\ \sqrt{V_{x,k}^2 + V_{y,k}^2} \\ \sqrt{V_{x,k}^2 + V_{y,k}^2} \\ a_{x,k} \\ a_{y,k} \end{pmatrix} \tag{2.14}$$

Therefore, the observation matrix is

$$\mathbf{H}_k = J\,\mathbf{h}_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{v_{x,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & \frac{v_{y,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & 0 & 0 \\ 0 & 0 & \frac{v_{x,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & \frac{v_{y,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.15}$$

Usually the GNSS solution is available with a lower rate with respect to INS measurements. Therefore, GNSS measurements may quickly become obsolete. In order to solve this problem, we use the following observation matrix whenever the GNSS measurements are not up to date:

$$\mathbf{H}_k = J\,\mathbf{h}_k = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{v_{x,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & \frac{v_{y,k}^2}{\sqrt{v_{x,k}^2+v_{y,k}^2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

In other words, whenever the up-to-date GNSS position is available, it is then used to calibrate the inertial platform. Otherwise the sensing measurements would lead to wrong estimates after a while, due to their growing biases.

2.2.2 Stochastic model

The stochastic model represents the uncertainties that characterize the dynamic system. As usual in the reference literature, we consider both the state noise and the observation noise to be mutually independent to each other and to have multivariate Gaussian $\mathcal{N}$ distributions with zero mean:

$$p(\mathbf{w}_k) = \mathcal{N}(\mathbf{0}; \mathbf{Q_k}), \forall k \tag{2.17}$$

$$p(\mathbf{v}_k) = \mathcal{N}(\mathbf{0}; \mathbf{R_k}), \forall k \tag{2.18}$$

where $\mathbf{Q} \in \mathcal{M}_{6\times6}(\mathbb{R})$ and $\mathbf{R} \in \mathcal{M}_{6\times6}(\mathbb{R})$ denote the state and observation noise covariance matrices respectively. While the matrix $\mathbf{Q}_k$ is intended to be recursively optimized by the proposed algorithm, the matrix $\mathbf{R}_k$ is instead assumed to be known

at each time step. Coherently with the definition of $\mathbf{H}_k$ for the observation model, this noise covariance matrix has two alternative formulations. The first definition is valid whenever the GNSS information are up-to-date:

$$\mathbf{R_k} = \begin{bmatrix} \sigma_X^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_Y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{v_{GNSS}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_{INS}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{a_{x_{body}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{a_{y_{body}}}^2 \end{bmatrix} \tag{2.19}$$

On the contrary, the following expression is built so that the Kalman filter relies more on the other measurements coming from the inertial sensors:

$$\mathbf{R_k} = \begin{bmatrix} \Sigma & 0 & 0 & 0 & 0 & 0 \\ 0 & \Sigma & 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_{INS}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{a_{x_{body}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{a_{y_{body}}}^2 \end{bmatrix} \tag{2.20}$$

where $\Sigma \in \mathbb{R}^+$ indicates an arbitrary large value.

2.2.3 Statistical model optimization

The goal of the following discussion is to ensure that the extended Kalman filter performance is close to the optimum by getting an insight into the optimal noise covariance matrices. For this purpose, we recursively compute at every iteration the matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ that maximize the probability of observing the current and past output measurements. The advantage of a recursive algorithm is that it performs an *adaptive* filtering, which is necessary if the true noise covariance matrices are time-varying. In brief, this technique returns the noise covariance matrices that are characterized by the maximal probability density function (pdf) [46]. Let us define a *likelihood function* as the probability distribution of the actual output observed vector $\mathbf{y}_k$ conditioned by the previous measured data $\mathbf{Y}^{k-1}$ and by the current matrices $\mathbf{Q_k}$ and $\mathbf{R_k}$. Since the entering noise is assumed white and Gaussian as usual, this

function results a multivariate Gaussian N distribution with the mean vector $\bar{\mathbf{y}}_k$ and the covariance matrix $\mathbf{P}_{\mathbf{k}}^{\mathbf{y}}$ that are predicted for the current output $\mathbf{y}_k$:

$$\mathbf{Y}^k = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_k\} \tag{2.21}$$

$$p(\mathbf{y}_k|\mathbf{Y}^{k-1}, \mathbf{Q_k}, \mathbf{R_k}) = \mathcal{N}(\bar{\mathbf{y}}_k; \mathbf{P}_{\mathbf{k}}^{\mathbf{y}}) = \mathcal{N}(\hat{\mathbf{y}}_k; \mathbf{P}_{\mathbf{k}}^{\mathbf{y}}) \tag{2.22}$$

The posterior conditional pdf of $\mathbf{Q_k}$ and $\mathbf{R_k}$ conditioned on the observed data $\mathbf{Y}^k$ (up to the current k-th time step) is updated by recursively multiplying the likelihood function with the prior conditioned pdf, as follows:

$$p(\mathbf{Q_k}, \mathbf{R_k}|\mathbf{Y}^k) = \frac{p(\mathbf{y}_k|\mathbf{Y}^{k-1}, \mathbf{Q_k}, \mathbf{R_k})}{p(\mathbf{y}_k|\mathbf{Y}^{k-1})} \cdot p(\mathbf{Q_k}, \mathbf{R_k}|\mathbf{Y}^{k-1}) \tag{2.23}$$

In other words, the most probable pair of noise covariances matrices is found by numerically approximating the Bayes formula at every iteration. This approach is employed in a recursive algorithm that is in order of optimizing the statistical model.

2.2.4 Recursive algorithm

Chosen a suitable parameters vector $\mathbf{\Omega}$, we define a set S of state noise covariance matrices according to the prior knowledge about the stochastic model. The initial parameters have usually a logarithmic scale to assure a higher density for smaller covariances. For the sake of simplicity, the matrix $\mathbf{Q}$ is parametrized as a multiplier of the unit matrix:

$$\mathbf{\Omega}_k = (\Omega_k^1, \Omega_k^2, ..., \Omega_k^N), N \in \mathbb{N} \tag{2.24}$$

$$S_k = S(\mathbf{\Omega}_k) = \{\mathbf{Q}(\Omega_k^i) = \Omega_k^i \cdot \mathbf{I}_6, \forall i\}, \Omega^i \in \mathbb{R}^+ \tag{2.25}$$

As far as the observation noise covariance is concerned, the matrix $\mathbf{R}$ is instead fixed according to Eq (2.19) and (2.20), because we assume to know the exact measurements statistics.

Given a grid of noise covariances matrices, the algorithm executes multiple EKF at every iteration, one for each pair $(\mathbf{Q}(\Omega_k^i), \mathbf{R}_k)$, parallelizing the state estimation. In the meanwhile, the conditional posterior probability distribution $p(\mathbf{Q}(\Omega_k^i), \mathbf{R_k}|\mathbf{Y}^k)$ is also recursively computed using Eq. (2.23) for all the covariances covered by the grid at the current time step. After all the desired matrices have been processed, the optimal state noise covariance matrix to be used by the filter is selected according to the maximum a-posteriori (MAP) criterion:

$$\hat{\mathbf{Q}}_k = \underset{\Omega_k^i}{\operatorname{argmax}}\, p(\mathbf{Q}(\Omega_k^i), \mathbf{R_k}|\mathbf{Y}^k) \tag{2.26}$$

Before the next iteration, the algorithm generates new points for the set S_{k+1} from the last posterior pdf values by resorting to the importance sampling method, which belongs to the Monte Carlo family. Then, a prior PDF is assigned to these points and the whole estimation and optimization process is repeated again.

At first, the algorithm enters an initialization phase to train the conditional posterior probability distribution through the Bayesian approach, without updating the grid of parameters and thus keeping the first set S_0. The length of this starting period depends on the dynamic system order and the available measured data. Afterwards, the iterative generation of new sets of state noise covariances is also enabled in order to improve the search for the maximum by the MAP criterion. Since parameters with high posterior PDF are desired, those with low probability are suitably replaced at each time step. The optimization of the matrix $\mathbf{Q}$ therefore aims at identifying and selecting the current optimal (i.e. best possible) estimate among all the possible state vectors $\hat{\mathbf{x}}$ that are computed through parallel EKFs.

We implemented an algorithm that is an improved version of the recursive estimation method described in [46]. Indeed, we devised a mechanism to prevent the probabilities of all the possible noise covariances from being nullified when the deterministic model undergoes a fast change in the measured dynamics, as it may happen with high speeds along critical trajectories. If the conditional posterior pdf results to be zero for each point of the parametrization grid, the initialization phase is run again and a new probability distribution shape is then re-formed. The iterative operation of the algorithm we used is summarized in the diagram of Fig. 2.11.

2.2.5 Numerical results

The overall performance of the presented navigation system is evaluated by simulating the tracking of a fast-moving vehicle on different trajectories as well as on a whole circuit. In this regard, the chosen figure of merit is root mean square error (RMSE) of the position estimates that are computed by our quasi-tight GNSS/INS implementation based on EKF. In order to verify the benefit of recursively adapting the state noise covariance matrix, the RMSE is calculated both with and without the proposed optimization process and parallel estimation. For all the tests, the simulations are repeated 100 times sweeping through the following initial logarithmic parametrization:

$$\begin{aligned}\boldsymbol{\Omega}_0 = (&0.0010, 0.0017, 0.0028, 0.0046, 0.0077,\\ &0.0129, 0.0215, 0.3590, 0.0599, 0.1000)\end{aligned} \tag{2.27}$$

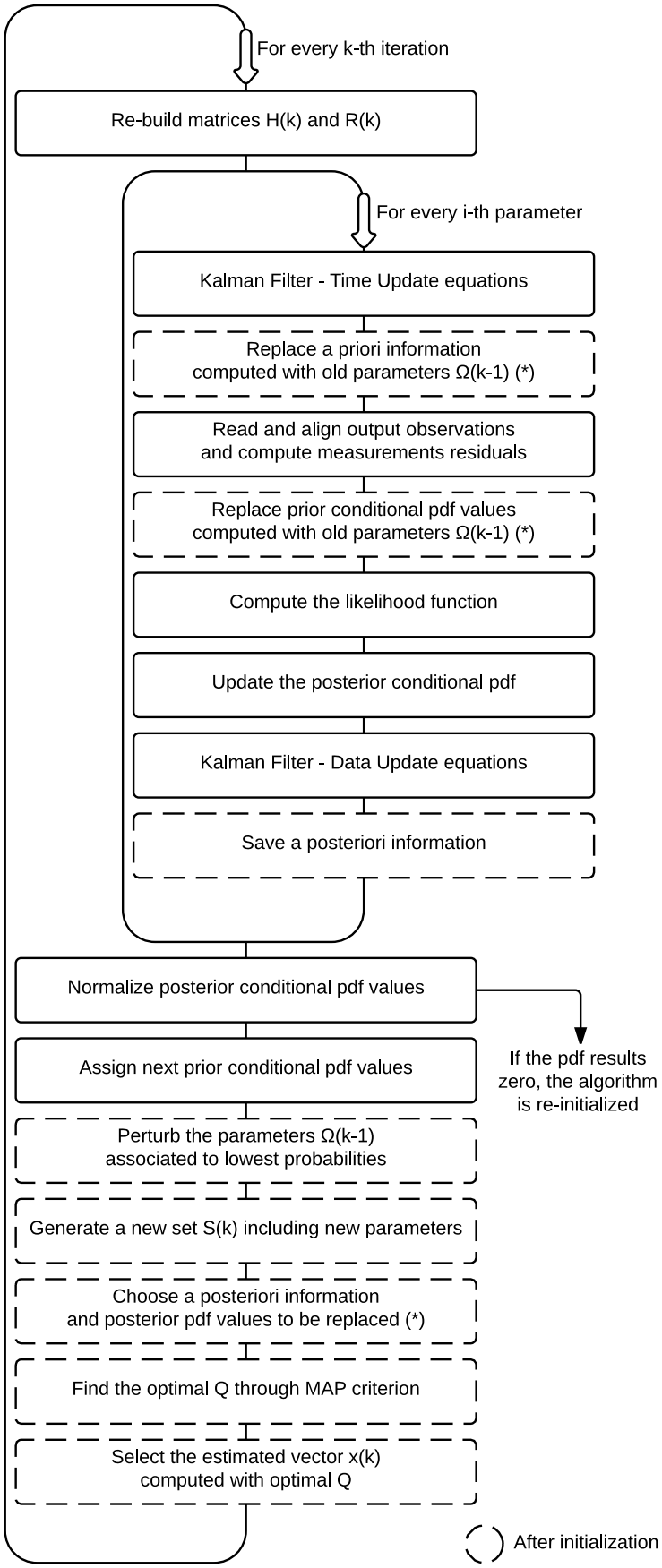

Figure 2.11: Stochastic model optimization diagram

	μ	σ
e_X, e_Y	0	10 m
$e_{v_{GNSS}}, e_{v_{INS}}$	0	1 m/s
$e_{a_{x_{body}}}, e_{a_{y_{body}}}$	0	0.03 m/s^2
e_θ	0	0.044 rad

Table 2.2: First order statistics of measurement errors

The simulation configuration is briefly characterized as follows. The specific parameters that shape the uncertainty affecting the output observations are listed in Table 2.2, where $\mathbf{e}_\alpha$ is the error vector on the measurement α. Then, we consider an update frequency factor of 10 between the measured data coming from the GNSS and the INS. As far as the estimation process is concerned, we assume to identify the exact transition model and observation model for the Kalman filter. This means the deterministic information are known.

Four different types of trajectories are generated, each one has a simulation time of 10 seconds, as shown in Fig. 2.12, 2.13, 2.14, and 2.15. The aim of these tests is to separately analyze the algorithm efficiency while tracking the basic motions of a particle-like vehicle. The GNSS/INS estimation process is run with two configurations: a recursively optimized matrix $\hat{\mathbf{Q}}_{opt}$ based on parallel filtering and a single filter with fixed minimal noise covariance matrix that is defined as:

$$\mathbf{Q}_{min} = \mathbf{Q}(\Omega^1) = \mathbf{Q}(0.001) \tag{2.28}$$

In other words, we compare the positional accuracy of the optimizing technique with those achieved with minimum state noise, which hence is associated to a well-identified deterministic model. The constants characterizing the kinematic equations and the RMSE values along the X and Y axes for both the configurations are reported in Table 2.3. The obtained results demonstrate that adapting the noise covariance enhances the average performance in terms of tracking accuracy, even if there is just a little uncertainty left on the state, which already represents an advantageous situation. In the simplest case of a linear motion, the RMSE is reduced by

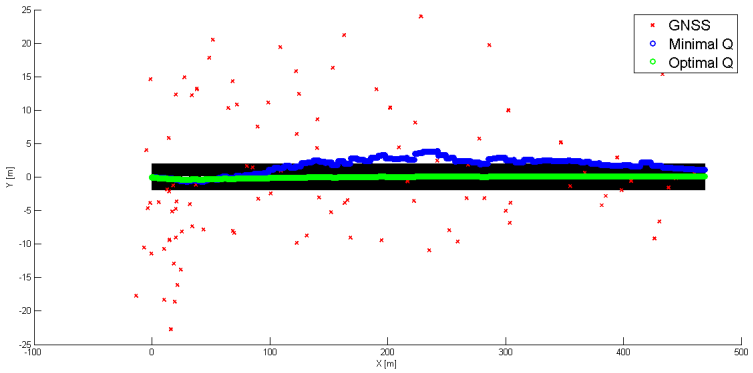

Figure 2.12: Example of straight line (1)

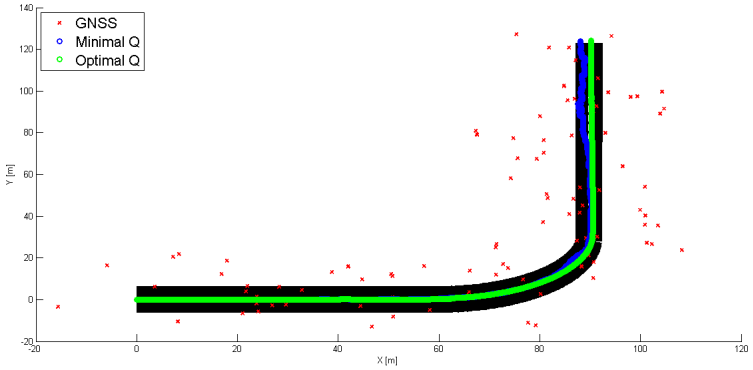

Figure 2.13: Example of quick turn (2)

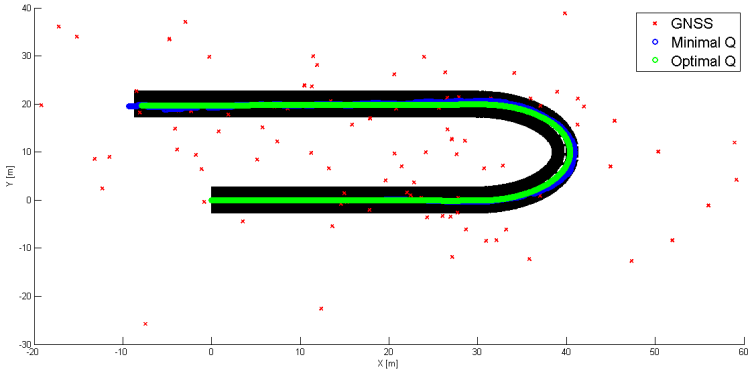

Figure 2.14: Example of circular hairpin (3)

	$RMSE(\hat{\mathbf{Q}}_{opt})$		$RMSE(\mathbf{Q}_{min})$	
	X	Y	X	Y
1) $a_X = 9.4\,m/s^2$	0.46 m	0.2 m	1.92 m	0.3 m
2) $\|v\| = 20\,m/s$	0.98 m	0.58 m	9.98 m	11.64 m
3) $\|v\| = 10\,m/s$	0.64 m	0.35 m	7.07 m	6.1 m
4) $\|v\| = 10\,m/s$	1.28 m	1.88 m	15.47 m	12.58 m

Table 2.3: Single racetracks results

a factor of about 4, but the improvement significantly grows as trajectory become more intricate. Whenever the transition and observation models do not fit the reality, optimizing the filter statistic inputs is expected to be necessary. As far as the state is concerned, different degrees of uncertainty are investigated by examining the vehicle moving on a circuit.

A final test is carried out on a circuit clearly inspired inspiration from a typical scenario of a FSAE competition. The route is designed to assemble all the analyzed segments with either uniform accelerations or constant velocities, as depicted in Fig. 2.16. In addition to the two previous configurations, we also consider the case of

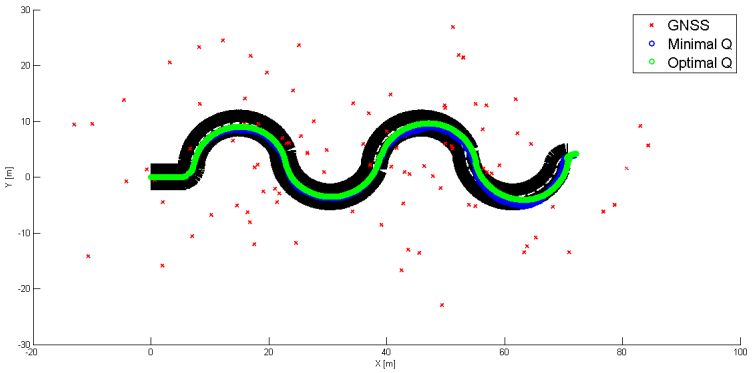

Figure 2.15: Example of slalom (4)

$RMSE(\hat{\mathbf{Q}}_{opt})$		$RMSE(\mathbf{Q}_{min})$		$RMSE(\mathbf{Q}_{rand})$	
X	Y	X	Y	X	Y
1.17 m	2.39 m	9.29 m	5.94 m	71.5 m	39.63 m

Table 2.4: Complete circuit results

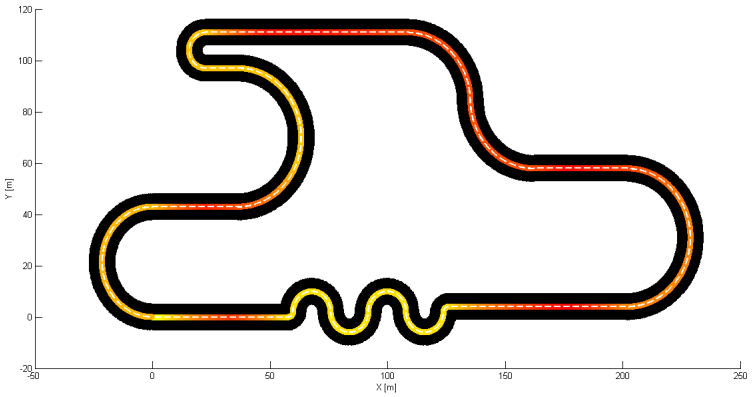

Figure 2.16: Circuit design with colored vehicle speed

a matrix $\mathbf{Q}_{rand}$ that is randomly chosen in the set based on the parameters of Eq. (2.27). At each simulation, a different state noise covariance is thus entering the Kalman filter. This event could happen when the deterministic model is poorly characterized over time. The performance achieved after one lap of 44.06 seconds are summarized in Table 2.4. As expected, the best accuracy of about 1-2 meters is enabled by selecting $\mathbf{Q}_{opt}$ (Fig. 2.17), whereas $\mathbf{Q}_{min}$ ensures a decent precision on average, and a random $\mathbf{Q}_{rand}$ can cause the EKF to diverge from the actual positions. Here the improvements reaches again a factor of 4. These results show the great potentiality of implementing the presented optimization method for a quasi-tightly coupled GNSS/INS system as well as other automotive applications.

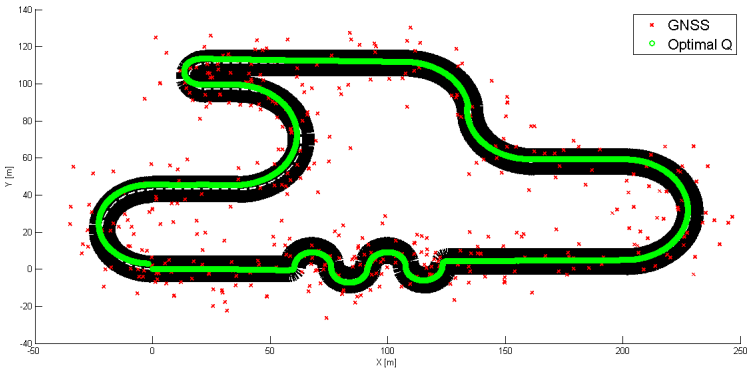

Figure 2.17: Optimized tracking on the circuit under test

Chapter 3

Synchronisation of low-cost open SDR for navigation applications

In the early 2000s, SDR started to gain popularity in the scientific community but, due to their high cost, they have been accessible only to researchers and not to ordinary users. This trend has changed in the last few years with the introduction of low-cost SDR on the market, which has made this technology available to everyone. The cost of these devices is at least one order of magnitude lower than a comparable professional SDR. Moreover most low-cost SDR are open-hardware and open-software, hence fully customisable. Professional SDR offer time and frequency synchronisation of multiple SDR, enabling a wide range of navigation applications. All of these applications require accurate time and frequency synchronisation, but low-cost SDR generally do not support time synchronisation.

Time and frequency synchronisation of multiple SDR enables or improves a wide range of navigation applications. It is possible to classify these navigation applications in two main categories: receiver- and network-oriented applications. Receiver-oriented applications are the ones that allow a single receiver to compute its own position, while network-oriented applications are the ones involving more than one receiver that are not usually colocated.

Receiver-oriented applications may take advantage of the synchronisation of multiple SDR to build multi-band and/or multi-system receivers. An important example of this possible use case is represented by hybrid navigation applications, i.e., receivers able to determine their own position by exploiting multiple sources or systems. This kind of receivers perform ranging measurements from different signals of

opportunity (SoO) [48], such as WiFi, Bluetooth, Ultrawide Band (UWB), 3G, 4G Long Term Evolution (LTE), and prospectively 5G, as well as GNSS, in order to solve the positioning problem. Hence, these receivers must be able to tune their radios to different bands and demodulate different signals simultaneously. This can be done by using multiple SDR, but synchronisation is required among them for time-based or frequency-based ranging. Another possible application falling into this category is represented by multi-band GNSS receivers, where different GNSS signals in different frequency bands are simultaneously processed, for better accuracy and robustness in harsh environments [49].

An example of network-oriented application is cooperative or peer-to-peer (P2P) positioning [50], in which multiple GNSS users cooperate to achieve a position solution in difficult environments. In this application, users need to communicate with each other either to exchange navigation data or to perform terrestrial ranging between users. The communication between users or the terrestrial ranging can be implemented using SDR and synchronisation is mandatory for time-based and frequency-based terrestrial ranging. Another important network-oriented application is the detection and localisation of GNSS jammers or spoofers: in this case multiple sensor nodes monitor the GNSS bands and send snapshots of the signal to a server in the cloud, which detects and localise potential jammers using TDOA and FDOA [51]. Sensor nodes can be implemented using SDR and strict synchronisation is required for TDOA/FDOA-based ranging.

Although these navigation applications are feasible with current professional SDR, they cannot be exploited with very low-cost equipment. In this chapter, we propose and validate an algorithm that enables sample-level synchronisation of multiple low-cost SDR by using an off-the-shelf GNSS receiver.

3.1 Open-source and open-hardware SDR

Different definitions of software-defined radio are possible and we adopt the one given in [52]: a *software-defined radio* is a *radio* in which some or all of the physical layer functions are *software defined*. In the last decade the cost of SDR has dropped, enabling a widespread diffusion of this technology. Most of the low-cost SDR are open-source and open-hardware: schematics and PCB layouts are public; the code of the firmware and software is available free of charge and can be modified according to the users' need. Example of popular low-cost SDR are *RTL-SDR* [53], a digital

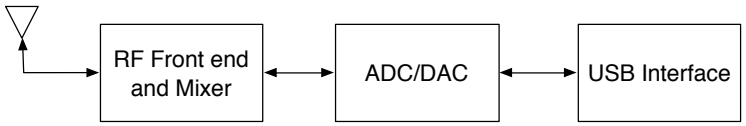

Figure 3.1: Generic SDR architecture

Figure 3.2: Selected SDB architecture

TV tuner that may be used as SDR (receive only); *HackRF One* [54], featuring a wide frequency range and a wide bandwidth (half-duplex transmit and receive), *BladeRF* [55], featuring USB 3.0 (full-duplex transmit and receive).

The aforementioned SDR share a common architecture, shown in Figure 3.1: in receive mode, the signal coming from the antenna is filtered, amplified, and downconverted to baseband by the RF front end and mixer, then the signal is discretised and quantised by an Analogue to Digital Converter (ADC) and the samples are transmitted to a computer (host) in charge of signal processing, via an USB interface.
In transmit mode, signal samples are fed to a Digital to Analogue Converter (DAC) through the USB interface, then the signal is upconverted, amplified by the RF front end, and transmitted.

3.1.1 Selected development board

Our selected development board (SDB) is HackRF One [54], a low-cost open-source and open-hardware SDR, capable of transmitting or receiving signals from 1 MHz to 6 GHz. The ADC/DAC operates at up to 20 Msps (8 bit I/Q samples). Baseband filter and transmit/receive gain are configurable by software, and pin headers on the PCB allow future expansions. The architecture of the SDB is shown in Figure 3.2 and it is composed by three stages: RF, Intermediate Frequency (IF), and baseband (BB). In receive mode, the RF signal is converted to IF by an *RFMD RFFC5071/2*, a wideband synthesiser with integrated 6 GHz mixer, and a *Maxim Integrated MAX2837* wireless broadband transceiver, is responsible for the conversion from IF to BB and the ADC/DAC (codec) functions. A *Xilinx XC2C64A* Complex Programmable Logic Device (CPLD) acts as glue logic between the codec and the *NxP LPC43XX* micro controller (MCU) which provides the USB interface to the

user. The same components are responsible for the opposite functions in transmit mode. The SDB provides *clock input* and *clock output* ports for frequency synchronisation of multiple SDR. Time synchronisation is not supported by the current firmware but, thanks to the hardware and software openness, it is possible to use a time pulse for time synchronisation purposes. The next section describes in detail the synchronisation algorithm and the necessary hardware connections.

3.2 Synchronisation Algorithm

In the following we tackle the synchronisation of SDBs in receive mode. A similar approach may be applied to achieve the same result in transmit mode. The algorithm can be also applied with minor modification to other open source SDRs with similar architectures. The idea is to use an expansion pin header on the PCB to add new signals for time synchronisation: the start of reception should be triggered by a time pulse; the pulse may be generated by one of the SDBs or by external hardware. Figure 3.3 shows the hardware connections between two SDBs: the signal SYNC_IN (pin 16 of the expansion header P28, top and bottom SDBs) is the input for the synchronisation pulse. SYNC_CMD (pin 15 of the expansion header P28, top SDB) is the pulse command signal. We consider two configurations: in the first one (A), the pulse command is generated by one of the SDBs; in the second configuration (B) the pulse command is the 1PPS signal of a GNSS receiver, i.e., a time pulse with a one-second period, synchronised with GPS time. While configuration (A) does not require additional hardware, it requires the receivers to be colocated. In configuration (B), instead, SDBs may be located far away from each other. Moreover, in this case, the recordings are synchronised with GPS time. The signals SYNC_IN and SYNC_CMD are connected to the CPLD, as shown in Figure 3.4, along with other signals involved in the receive mode. During the initial setup, the center frequency, filters bandwidth, gains, and ADC sampling rate are configured. Then, the ADC starts streaming the samples of the received signal to the CPLD (signal ADC_DATA[7..0]). The samples are available on both rising (I) and falling edge (Q) of CODEC_CLK. The CPLD simply makes the data (HOST_DATA[7..0]) available to the MCU on the rising edge of HOST_CLK, whose frequency is doubled with respect to CODEC_CLK. The MCU communicates to the CPLD the wish to start the recording, setting HOST_DISABLE to '0'. At this stage the MCU is ignoring the data samples and it will continue until the CPLD sets HOST_CAPTURE to

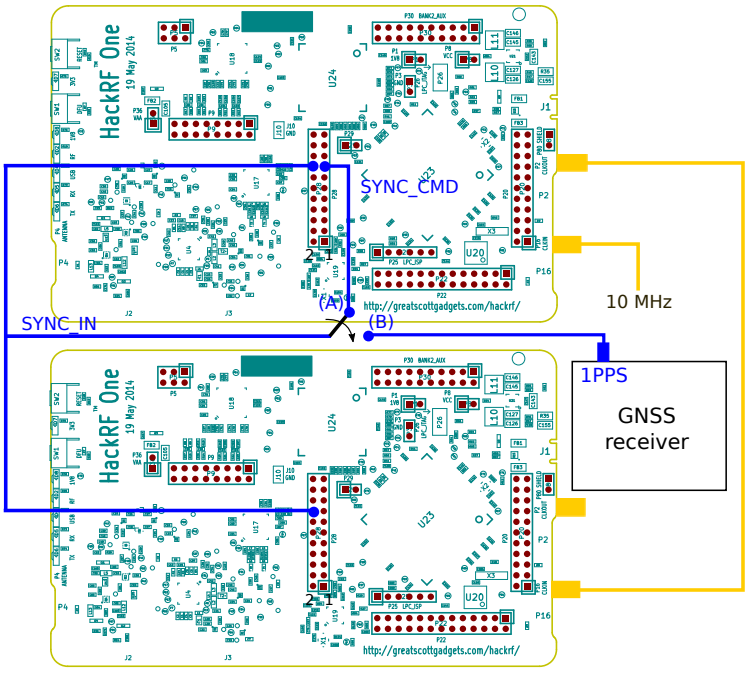

Figure 3.3: Synchronisation of two SDBs: hardware connections

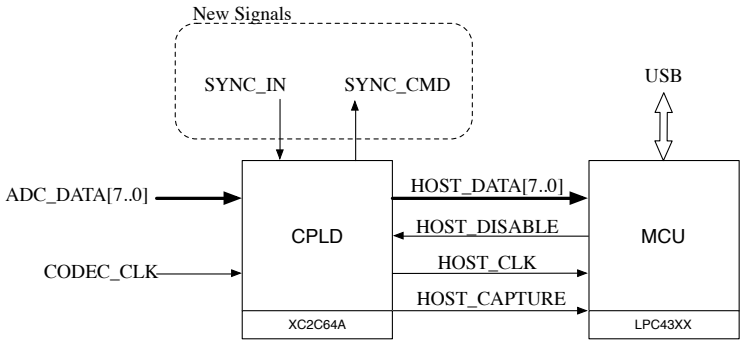

Figure 3.4: CPLD and MCU signals

'1'. The CPLD controls HOST_CAPTURE in order to ensure the correct alignment of interleaved I/Q samples. Once the I/Q data alignment is correct, the MCU may finally start streaming the samples via USB. When the recording phase ends, the MCU pulls HOST_DISABLE high, to signal the CPLD the end of the recording. A snippet of the original VHDL controlling the signal HOST_CAPTURE is available in Listing 3.1.

Listing 3.1: Original VHDL (simplified)

```
-- MCU ignores data samples when HOST_CAPTURE = '0'. The following code guarantees
     that the MCU receives I/Q interleaved samples with I samples in the odd
    positions and Q samples in the even ones.
process(HOST_CLK)
begin
  if rising_edge(HOST_CLK) then
    if CODEC_CLOCK = '0' then
      HOST_CAPTURE <= not HOST_DISABLE
    end if;
  end if;
end process;
```

This architecture leads to the conclusion that signal HOST_CAPTURE is a good candidate for time synchronisation. In principle, it is sufficient to hold HOST_CAPTURE low when SYNC_IN is low or HOST_DISABLE is high. This should be enough to ensure that the synchronisation error is below one sampling period. However, the time pulse must stay high during the whole receiving phase, otherwise the stream of samples to the user is interrupted. Therefore, this method does not work with the 1PPS signal, which is typically a low duty-cycle square wave. This approach may be improved by using a latched version of the signal SYNC_IN, as shown in Listing 3.2. In this case, it is not required that the pulse command stays high during the whole receiving phase and this approach works with 1PPS signals.

Listing 3.2: Modified VHDL (simplified)

```
-- The MCU ignores data when HOST_DISABLE='1' or sync_in_latched='0'. The
    following code allows multiple SDBs to start receiving samples synchronously.
process(HOST_CLK)
begin
  if rising_edge(HOST_CLK) then
    if CODEC_CLOCK = '0' then
      HOST_CAPTURE <= not HOST_DISABLE and sync_in_latched;
    end if;
  end if;
end process;
```

```
-- When the MCU pulls HOST_DISABLE low, HOST_SYNC_CMD becomes high (the
    synchronisation pulse is sent to the other SDBs). sync_in_latched is high when
      HOST_DISABLE is low and there is a rising edge of SYNC_IN; it is low when
    HOST_DISABLE is high.
process(HOST_DISABLE, SYNC_IN)
begin
  SYNC_CMD <= not HOST_DISABLE;
  if HOST_DISABLE = '0' then
    if rising_edge(SYNC_IN) then
      sync_in_latched <= SYNC_IN;
    end if;
  else
    sync_in_latched = '0';
  end if;
end process;
```

In order to validate the synchronisation algorithm, we are using two different approaches in the next section, based on GNSS and LTE signal processing.

3.3 Experimental Results

This section discusses the laboratory setup, the GNSS and LTE validation approaches, and the experimental results on the synchronisation offset. A statistical model is then proposed to characterise the resulting synchronisation offset.

3.3.1 Experimental setup

The experimental testbed is located in the RF Systems and Payload laboratory of the European Space Agency (ESTEC, The Netherlands). A diagram of the test setup is shown in Figure 3.5, where the signal source is either a GNSS antenna (GNSS) or a LTE network emulator (LTE). A high-end active GNSS antenna is located at the roof of the building in open-sky conditions. The Spirent E2010S network emulator generates the LTE signal from one base station (BS) at a system bandwidth of 1.4 MHz in AWGN conditions, with a signal-to-noise ratio (SNR) around 30 dB.

The input signal goes through a RF power divider and then to the two SDBs. A reference signal of 10 MHz generated by an active hydrogen maser is used to synchronise, in frequency, the clocks of the LTE network emulator and the two SDBs. The time synchronisation is achieved with square pulse of 1 Hz obtained from either a GNSS receiver (1PPS) or an Arbitrary Function Generator (AFG). Hardware connections between the synchronisation pulse and the two SDBs, described in Section 3.2, are implemented in a prototyping PCB, as shown in Figure 3.8. The

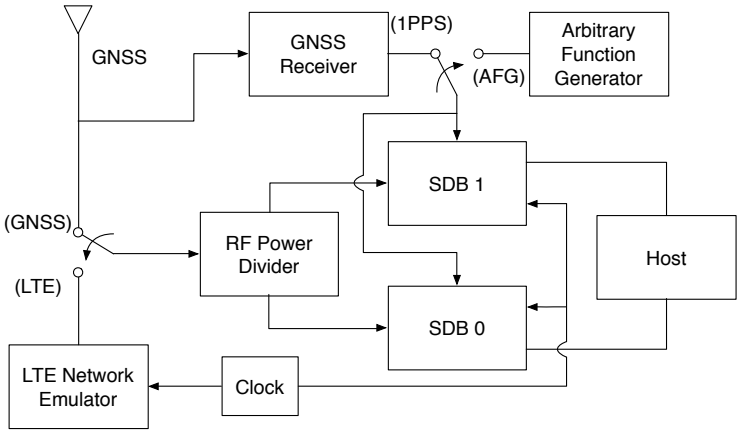

Figure 3.5: Experimental setup of the laboratory

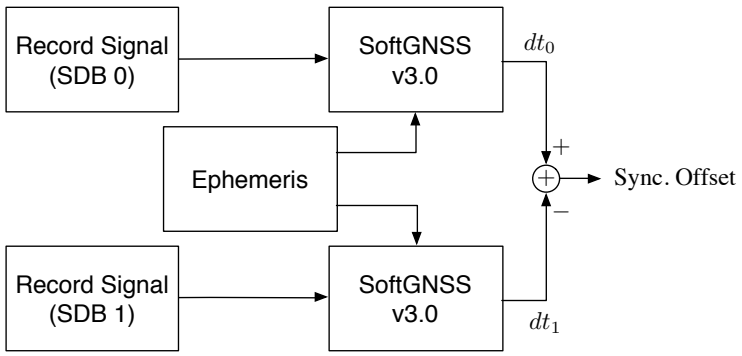

Figure 3.6: Diagram of the GNSS-based validation approach

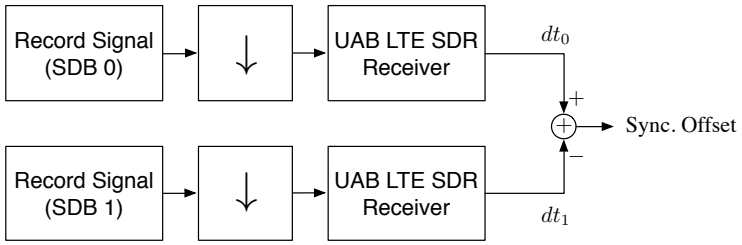

Figure 3.7: Diagram of the LTE-based validation approach

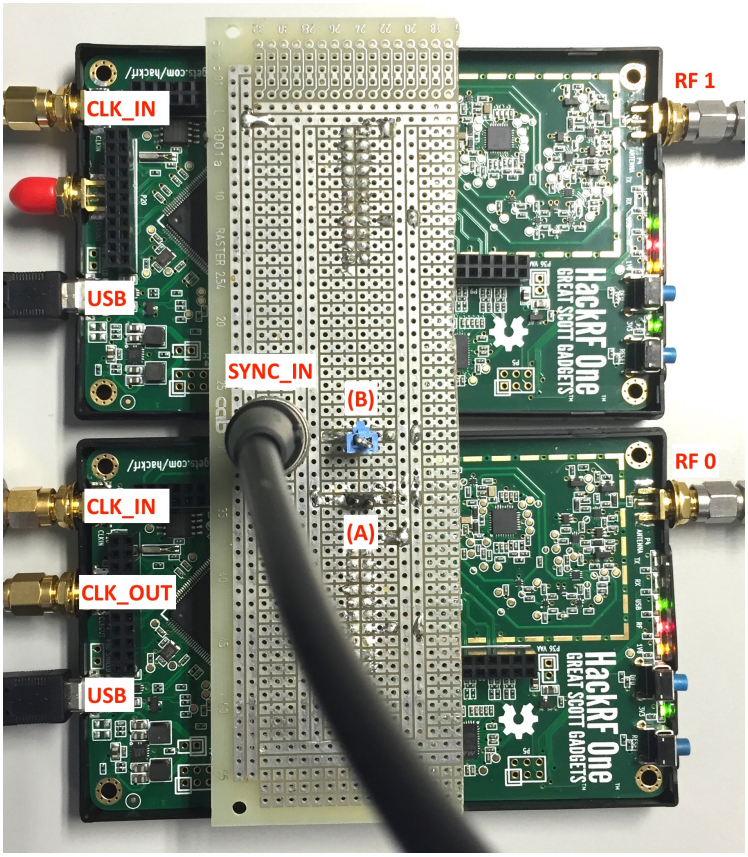

Figure 3.8: SDB connection board

signals received by the SDBs are recorded at a sampling rate F_s by a laptop PC (host) and processed using either the GNSS- or LTE-based approaches. In order to obtain statistically stable results, experiments are repeated 1000 times. SDBs are reset after each recording, to ensure that the experiments are performed in the same conditions. This has required the modification of the MCU firmware and the SDB Linux tools: a new function has been added to allow the host to reset SDBs using the USB interface. A few recordings experienced losses of samples at high sampling rates, because the slow hard drive installed in the laptop could not cope with the high data rate sample stream. To avoid this problem signals are recorded in a tmpfs [56] virtual memory disk, a disk residing in RAM memory. Another possible issue that may occur is due to the drift of the host clock with respect to GNSS time: if the host starts a recording when the 1PPS synchronisation pulse is high, the CPLD will not be able to detect a rising edge and the signal will not be recorded. To avoid this problem is possible to use a GNSS receiver to synchronise the clock of the host, using *GPSD* in combination with *NTP* [57].

3.3.2 GNSS validation approach

The GNSS validation approach is based on the estimation of the position, velocity and time (PVT) solution using the *SoftGNSS v3.0* software GPS receiver [58]. This software was originally meant to work at IF with real samples, but the recordings captured by the SDBs are at BB and the samples are complex. Therefore, we modified the acquisition and tracking stages of SoftGNSS, in order to be able to work with this kind of signals. A further modification to the acquisition and the tracking phases has been done in order to use the same satellites: only the satellites that are visible to both SDBs are taken into account for acquisition and tracking, in order to have comparable solution accuracies for both SDBs. Moreover, the software requires at least 36 seconds of recorded signals, in order to compute a PVT solution. This is due to the fact that the receiver needs to demodulate the navigation data, in order to extract the ephemeris and determine the orbit of the visible satellites: the receiver needs to demodulate 5 subframes, each composed by 300 bits (6 s, since the bit time is 20 ms). The additional 6 seconds are needed because the tracking phase may start in the middle of a subframe. Since processing 36-seconds recordings is cumbersome and time consuming, we propose an assisted GNSS (A-GNSS) method: instead of retrieving the ephemeris data from the signals, we download *Receiver Independent Exchange Format (RINEX)* files coming from the *International GNSS Service (IGS)*

station located in Delft. IGS provides open access, high quality GNSS data, products, and services in support of research. The data downloaded from the IGS station is used to assist the SoftGNSS receiver and allows to obtain a PVT solution in ten seconds. The knowledge of the time of week (TOW), which is the number of 1.5 seconds elapsed from the beginning of the GPS week, is still required to compute the pseudorange measurements and hence, to obtain the navigation solution. The TOW can be read at the beginning of each subframe, therefore requires to demodulate at least 6 seconds of GPS signal. In order to further reduce the processing time, we applied the so-called coarse time positioning method [59], which allows to obtain a navigation solution without the need of reading the TOW. In this method, the time of week is treated as an additional unknown by the least squares algorithm, therefore the unknown navigation solution vector is $(X, Y, Z, c\,dt, t_W)^T$, where (X, Y, Z) is the position of the receiver, dt is the clock offset between the GPS time and the time of the receiver, c is the light velocity, and t_W is the TOW. The observation model relating pseudoranges to the navigation solution vector is given by

$$P_i = \sqrt{(X_i - X)^2 + (Y_i - Y)^2 + (Z_i - Z)^2} + c\,dt + c\frac{f_{d,i}}{f_{L1}}t_W + e_i, \qquad (3.1)$$

where (X_i, Y_i, Z_i) is the position of i-th satellite, $f_{d,i}$ is the Doppler frequency, f_{L1} is L1 band center frequency, i.e., 1575.42 MHz, and e_i is a measurement error term. Using this model, the least squares algorithm requires at least five visible satellites in order to find a navigation solution. If the total number of visible satellite is N, the geometry matrix, i.e., the Jacobian matrix of the observation model, $\mathbf{A} \in \mathcal{M}_{N\times 5}(\mathbb{R})$ is given by

$$\mathbf{A} = \begin{pmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 & 1 & cf_{d,1}/f_{L1} \\ \Delta x_2 & \Delta y_2 & \Delta z_2 & 1 & cf_{d,2}/f_{L1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_N & \Delta y_N & \Delta z_N & 1 & cf_{d,N}/f_{L1} \end{pmatrix}, \qquad (3.2)$$

where the first four columns are unchanged with respect to the traditional least squares navigation algorithm [58], and the last column is added for the TOW estimation. The synchronisation offset between the two SDBs is computed after the PVT solution, as the difference between the clock offsets of the two SDRs, i.e., $\theta = dt_0 - dt_1$. The GNSS approach is summarised in Figure 3.6. Since GNSS signals have been specifically designed for precise timing, the use of GNSS signals can be a natural choice to validate the proposed synchronisation method. However, a number of possible problems may arise using this approach. The most important drawback

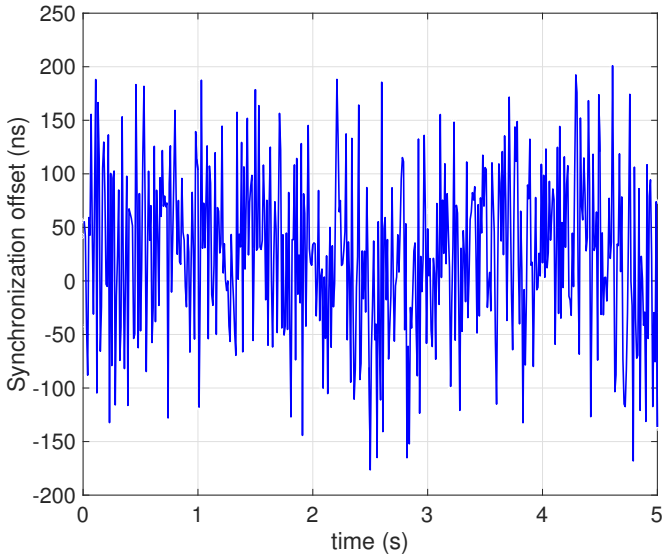

Figure 3.9: Synchronisation offset using the GNSS approach

of this validation method is that the accuracy of the clock offset estimator is not constant and depends on the number of visible satellites, on the carrier-to-noise-density ratio (C/N_0) of the received signals, and on the geometry of the satellite constellation. The high variability of the aforementioned accuracy and the fact that, occasionally it is not easy to have the same five satellites visible to both SDBs, make the use of this validation method impractical and the analysis of results troublesome. An example of synchronisation offset between the two SDBs obtained with the GNSS approach is shown in Figure 3.9. The synchronisation offset is estimated every 1 ms in a 5 seconds recording taken at a sampling frequency $F_s = 10$ MHz. The synchronisation offset, in this case, should be limited to $\pm 1/F_s = \pm 100$ ns but, in spite of the fact that the majority of the estimates fall within this interval, many outliers are present. This phenomenon is due to the low accuracy of the clock offset estimation caused by low C/N_0. Therefore, the GNSS approach is in practice cumbersome with the original least squares algorithm because it requires 36 seconds of data and it is not accurate enough when using the coarse time method with the described modification of the SoftGNSS receiver. In the following, we present a simpler yet effective LTE-based validation approach.

3.3.3 LTE validation approach

Terrestrial signals can be used as signals of opportunity to validate or calibrate the synchronisation procedure, in case there is a lack of visible GNSS satellites. This opportunistic approach can be based on cellular systems (e.g. 2G, 3G or 4G), broadcast television (e.g. DVB-T), Wi-Fi, Bluetooth or any other radio signal. The selection of the opportunistic system depends on the signal availability, and the signal bandwidth to achieve a certain time-delay estimation (TDE) accuracy. An accessible time reference within the signal can be useful to validate the synchronisation over time. Thus, this thesis considers the 4G LTE system due to its wide adoption in urban environments, high signal bandwidth (up to 20 MHz), and periodical time reference, i.e. system frame number (SFN) every 10 ms. In contrast to the GNSS approach described in the previous section, the LTE validation procedure is based on the estimation of the time-delay from the signal transmitted by a single base station with unknown location.

As it is shown in Figure 3.5 and 3.7, the LTE signal is first split and fed to the two SDBs that record a snapshot of 200 ms. The signal is then resampled to 2 MHz. The UAB LTE SDR software receiver, which is described in [60] and [61], is used to acquire and track the 1.4-MHz LTE signal captured by each SDB. The noise bandwidth of the delay-locked loop and frequency-locked loop is set to 30 Hz and 50 Hz, respectively, and the resolution of the TDE is defined to 0.25 ns. The SFN is calculated by using the LTE Cell Scanner software developed by [62]. Only ten LTE radio frames are tracked, and the last time-delay estimate of each processed capture is used to calculate the synchronisation offset between the two SDBs. The synchronisation is validated between multiple captures by using the time between recordings and the SFN.

3.3.4 Synchronisation offset

The PDF of the synchronisation offset is computed based on the LTE approach, by considering the 1PPS and AFG pulses. As it is shown in Figure 3.10, the resulting synchronisation offset is within $\pm 1/F_s$, i.e., ± 200, ± 100 and ± 50 ns for F_s equal to 5, 10 and 20 MHz, respectively. This test confirms the achievable accuracy of the proposed synchronisation procedure, which is bounded by the sampling period of the SDB. In addition, the same results are obtained with both synchronisation pulses, demonstrating the flexibility and reproducibility of the proposed algorithm. These results are validated by measuring the error of the TDE. As it is shown in

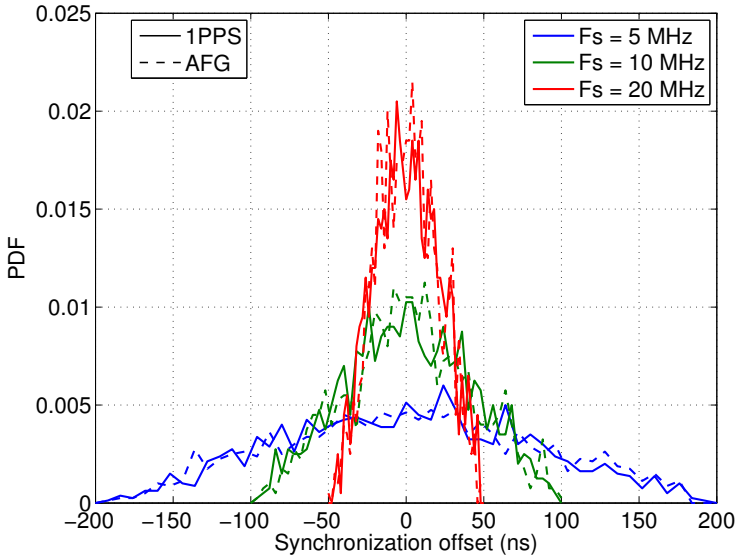

Figure 3.10: PDF of the synchronisation offset using the LTE approach

Figure 3.11, the PDF of this error is confined within ±5 ns. Thus, the estimation error of the test practically has no effect on the results. The mean μ_θ and standard deviation σ_θ of the synchronisation offset and the mean μ_τ and standard deviation σ_τ of the TDE error can be seen in Table 3.0(a). The main difference between the use of a synchronisation pulse generated by a GNSS receiver and a signal generator is the clock drift. Since the GNSS receiver is synchronised to the accurate atomic clocks of the satellites, the resulting square pulse is very stable. In contrast, the synchronisation pulse of the AFG has a clock drift due to its local oscillator. This effect can be observed in Figure 3.12, where the TDE of a single SDB is plotted over time. The TDE obtained with the AFG for different signal captures has a noticeable drift, which is wrapped to 10 ms (i.e., length of a radio frame) in the case of LTE, while there is no TDE drift with 1PPS.

3.3.5 Statistical model of the synchronisation offset

Due to the familiarity of the previous test results, a statistical model is here defined to characterise the synchronisation offset of the proposed algorithm. For this purpose, the synchronisation offset is normalised by $1/F_s$ for each test, resulting in the bar plot of Figure 3.13. These results lead to the conclusion that a truncated normal distribution [63] can fit the synchronisation offset to the algorithm. Thus, the

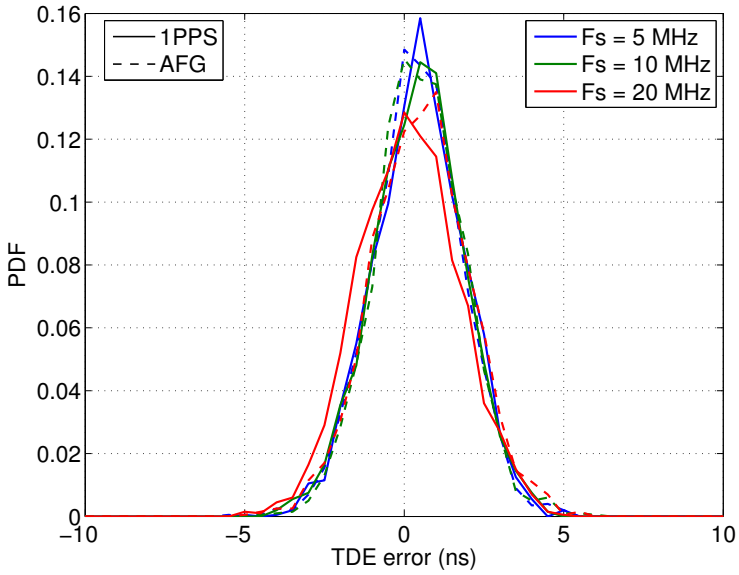

Figure 3.11: PDF of TDE error with the LTE approach

(a)

Sampling frequency	Signal source	Sync. pulse	Sync. offset μ_θ (ns)	Sync. offset σ_θ (ns)	TDE error μ_τ (ns)	TDE error σ_τ (ns)
5 MHz	LTE	AFG	1.45	82.56	0.40	1.34
		1PPS	-0.55	80.10	0.44	1.34
10 MHz	LTE	AFG	2.12	40.19	0.47	1.32
		1PPS	0.91	40.69	0.42	1.36
20 MHz	LTE	AFG	0.60	19.73	0.49	1.46
		1PPS	0.81	20.62	0.13	1.49

(b)

Signal source	Normalised Sync. offset μ_θ (samples)	Normalised Sync. offset σ_θ (samples)
LTE	0.01	0.40

Table 3.1: Mean μ_θ and standard deviation σ_θ of the synchronisation offset and TDE error for the LTE approach.

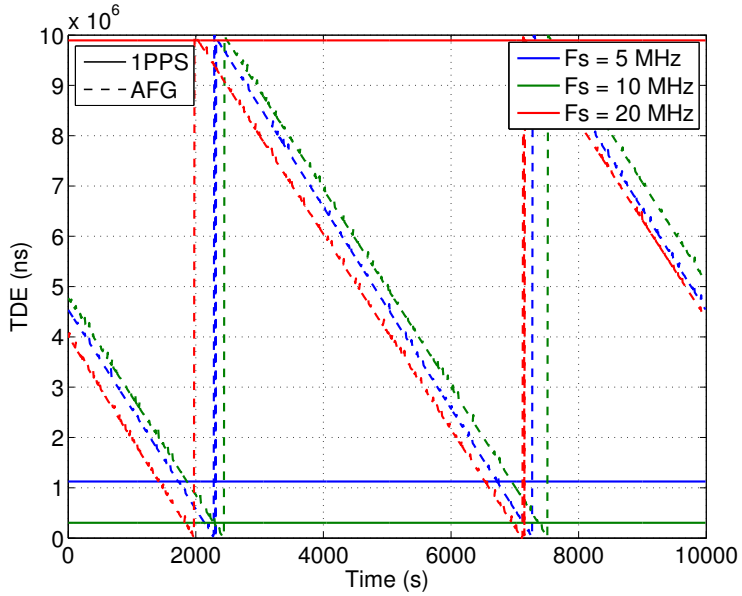

Figure 3.12: Synchronisation offset over time with the LTE approach

resulting synchronisation offset can be modelled as

$$\theta \sim \begin{cases} \mathcal{N}(\mu_\theta, \sigma_\theta), & \text{if } \theta \in (-1, 1), \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

where the PDF of the truncated Gaussian distribution is

$$f(\theta, \mu_\theta, \sigma_\theta) = \frac{\dfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\dfrac{(\theta-\mu_\theta)^2}{2\sigma^2}\right\}}{\dfrac{1}{2}\text{erf}\left\{\dfrac{1-\mu_\theta}{\sigma_\theta\sqrt{2}}\right\} - \dfrac{1}{2}\text{erf}\left\{\dfrac{-1-\mu_\theta}{\sigma_\theta\sqrt{2}}\right\}}, \tag{3.4}$$

and $\text{erf}\{\cdot\}$ is the error function. The truncated normal distribution is shown in Figure 3.13 by using the fitting parameters of Table 3.0(b), obtained with maximum likelihood estimators [63]. The observed resemblance suggests the validity of the model and confirms the performance of the proposed synchronisation method, whose accuracy is confined within a sampling period.

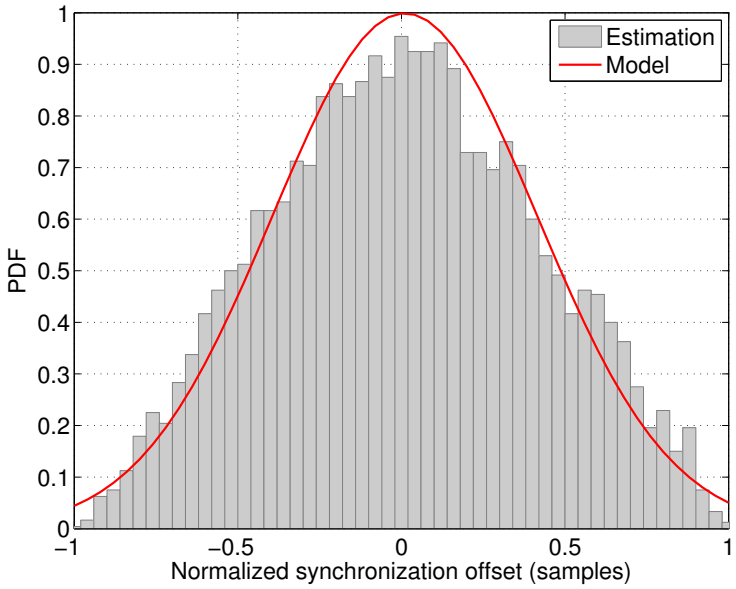

Figure 3.13: Statistical model of the synchronisation offset

Chapter 4

Conclusions

This thesis dealt with the problem of interference detection, localization, and mitigation in GNSS. The problem has been tackled in a cooperative fashion. The JJDL algorithm has been proposed for the detection and localization of jamming interference sources. The algorithm guarantees a constant false alarm rate with an easy design flow. The algorithm is customizable in order to cope with different localization techniques, and here two different flavours have been presented.

The DSWE algorithm has been proposed as a cooperative jamming waveform estimation tool, suitable for interference cancellation. The problem of interference mitigation has been tackled also from the point of view of GNSS and INS integration, thanks to an optimized implementation of the quasi-tight integration paradigm.

Finally, the development of a new strategy for synchronization of low-cost software-defined radios has allowed the succesful implementation of the TDOA/FDOA-based JJDL algorithm on low-cost hardware.

Appendix A

Kalman filters

A.1 Kalman filters

The Kalman filter (KF) is a linear recursive MMSE estimator of a hidden system state (i.e. not directly measurable) as a function of the previous estimate and of multiple sequential observations [34]. According to a discrete formulation of the problem, given k time samples, the actual state vector $\mathbf{x}_k$ is described as a linear dynamic system having the form:

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

where $\mathbf{A}_k$ is the matrix representing the transition model (i.e. free evolution), $\mathbf{B}_k$ is the control model (i.e. forced response), $\mathbf{u}_k$ is the input control vector, and $\mathbf{w}_k$ is the state noise vector. The actual state vector $\mathbf{x}_k$ is also linearly related to the current output measurements vector $\mathbf{y}_k$ through:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{A.2}$$

where $\mathbf{H}_k$ is the observation model and $\mathbf{v}_k$ is the measurement noise vector. The problem consists in computing the estimate $\hat{\mathbf{x}}$ of the actual state $\mathbf{x}_k$ from the observations $\mathbf{y}_0, \ldots, \mathbf{y}_k$, so as to minimize the expected value $\mathrm{E}\{\cdot\}$ of the squared norm of the unknown estimation error vector $\mathbf{e}_k$, defined as:

$$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \tag{A.3}$$

The KF is a computational efficient solution that provides optimal estimation under the following hypothesis:

- the vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are white-noise sequences and thus also mutually independent and independent of the initial state vector $\mathbf{x}_0$:

$$\mathrm{E}\left\{\mathbf{v}_k\mathbf{w}_k^T\right\} = \mathrm{E}\left\{\mathbf{v}_k\mathbf{x}_0^T\right\} = \mathrm{E}\left\{\mathbf{w}_k\mathbf{x}_0^T\right\} = 0,\ \forall k, j \in \mathbb{N} \setminus \{0\} \tag{A.4}$$

- the real dynamic system is accurately identified, both in its deterministic inputs consisting of the model matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{H}_k$ and in its stochastic inputs represented by the vectors $\mathbf{w}_k$ and $\mathbf{v}_k$.

Hence, if a reliable description of the system and of its initial conditions is available, the KF recursively provides maximum likelihood estimates of the state $\hat{\mathbf{x}}_k$ and of its error covariance matrix $\mathbf{P}_k$ that satisfy the following equalities:

$$\mathrm{E}\{\mathbf{e}_k\} = 0 \tag{A.5}$$

$$\mathbf{P}_k = \mathrm{E}\{\mathbf{e}_k\mathbf{e}_k^T\} \tag{A.6}$$

$$\hat{\mathbf{x}}_k = \arg\min_{\hat{\mathbf{x}}_k} \mathrm{E}\{\|\mathbf{e}_k\|^2\} = \arg\min_{\hat{\mathbf{x}}_k} \mathrm{Tr}(\mathbf{P}_k) \tag{A.7}$$

Moreover, let us suppose the entering noise vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ to have a multivariate Gaussian distribution $\mathcal{N}$ with zero mean and known covariance matrices, respectively denoted as $\mathbf{Q}_k$ and $\mathbf{R}_k$:

$$\mathbf{w}_k \approx\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{A.8}$$

$$\mathbf{v}_k \approx\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{A.9}$$

In particular, the matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ are chosen to reflect the degree of confidence that is placed in the model eq. (A.1) and (A.2), respectively. In case the KF ensures estimation optimality and is affected by AWGN, its operation may be expressed as the recursion of a system of simple equations defined in two steps: time update and data update. In the first phase, the a priori estimates $\mathbf{x}_k^-$ and $\mathbf{P}_k^-$ are predicted by projecting the previous estimates as follows:

$$\begin{cases} \hat{\mathbf{x}}_k^- = \mathbf{A}_k\hat{\mathbf{x}}_{k-1} + \mathbf{B}_k\mathbf{u}_k & \text{(A.10a)} \\ \mathbf{P}_k^- = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^T + \mathbf{Q}_k & \text{(A.10b)} \end{cases}$$

The error $\boldsymbol{\varepsilon}_k$ associated to the predicted output measurements $\hat{\mathbf{y}}_k$ is called innovation (or residual). Note that, under the conditions underlying the previous assumptions, the residuals are expected to form a white-noise Gaussian sequence. In other words, the error statistics propagate through the linear system dynamics. Consequently,

the whiteness property of the residual sequence may be considered as a measure of filter performance.

$$\boldsymbol{\varepsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^- \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_k) \tag{A.11}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \tag{A.12}$$

In the second phase, the state estimate is refined a posteriori by a weighted average that combines the a priori state predictions with the current residual $\boldsymbol{\varepsilon}_k$ through the Kalman optimal gain $\mathbf{K}_k$:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{A.13}$$

$$\begin{cases} \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \boldsymbol{\varepsilon}_k & \text{(A.14a)} \\ \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- & \text{(A.14b)} \end{cases}$$

In the next subsection, the KF theory is applied to nonlinear systems.

A.1.1 Extended Kalman filter

Given a generic discrete-time nonlinear dynamic system, such as:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k) \tag{A.15}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \tag{A.16}$$

where f_k and h_k are used to denote differentiable nonlinear functions. The EKF is an efficient method for nonlinear estimation. At each step, it essentially provides first-order approximations (Taylor series expansion) to the actual model terms around the mean value $\bar{\mathbf{x}}_k$ predicted for the current state $\mathbf{x}_k$, as shown below:

$$\mathbf{x}_k \approx \bar{\mathbf{x}}_k + \mathbf{A}_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{w}_{k-1} \tag{A.17}$$

$$\mathbf{y}_k \approx \bar{\mathbf{y}}_k + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k + \mathbf{v}_k \tag{A.18}$$

$$\bar{\mathbf{x}}_k = \mathrm{E}\left\{f_k(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_k)\right\} = f_k(\hat{\mathbf{x}_{k-1}}, \mathbf{u}_{k-1}, \mathbf{0}) \triangleq \hat{\mathbf{x}}_k^- \tag{A.19}$$

$$\bar{\mathbf{y}}_k = \mathrm{E}\{h_k(\bar{\mathbf{x}}_k, \mathbf{v}_k)\} = h_k(\bar{\mathbf{x}}_k, \mathbf{0}) \triangleq \hat{\mathbf{y}}_k \tag{A.20}$$

where $\mathbf{A}_k$ and $\mathbf{H}_k$ are computed as the Jacobian matrices of ! and h respectively; they describe the linear deviations from the reference mean values $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{y}}_k$. Such a linear approximation however impairs the accuracy of the system model and can lead to a suboptimal performance and even divergence of the filter. In order to control the divergence, the noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ can be tuned so that the residuals feature the desired statistical distribution, which is white and Gaussian, according to the initial assumptions.

A.1.2 Observability

As a recursive least-squares estimator, the KF relies on the fact that the errors can be averaged out by suitably combining more observations than the number strictly necessary to determine the desired unknowns. For a discrete-time system, observability means that the redundant observation of the output measurements over a time span $\{t_1, \ldots, t_k\}$ provides sufficient information to estimate the initial state $\mathbf{x}_0$ [64]. This, along with the knowledge of the system model, uniquely specifies the state vector at each time instant. In particular, a system is defined observable when the rank of its so-called observability matrix is maximum. This test returns a binary response that nevertheless does not provide any insight into the current degree of observability of the system. On the contrary, the eigenvalues λ_i of the error covariance matrix $\mathbf{P}_k$ are proved to be equal to the variances of the different linear combinations of the state components. Indeed, the least observable linear combination of state components is indicated by the largest eigenvalue of !!, while the most observable one is evidenced by the smallest eigenvalue. According to eq. (A.7), it is plain to see that the system gets more observable as the KF optimally minimizes the mean square error:

$$\mathrm{Tr}(\mathbf{P}_k) = \sum_i \lambda_i \tag{A.21}$$

Finally, it is convenient to adopt the following normalization in order to obtain adimensional eigenvalues:

$$\mathbf{P}_k^{'} = (\mathbf{P}_0)^{-1} \mathbf{P}_k (\mathbf{P}_0)^{-1} \tag{A.22}$$

A.2 Energy detection

Given a certain bandwidth, the problem of the detecting an unknown signal can be effectively solved by means of a precise radiometer [65]. The energy detection indeed applies to any deterministic signal, as far as its statistical characteristics are known, and is regardless of its waveform structure. By assuming a discrete AWGN channel, the energy measured over a certain sensing time returns a decision test static T consisting of the sum of N squares of Gaussian variables:

$$T(\mathbf{r}_i) = \frac{1}{N} \sum_{n=1}^{N} |r_i[n]|^2 \tag{A.23}$$

where $\mathbf{r}_i$ refers a generic received sample vector. A suitable sampling span is chosen in order to have a sufficient number of terms:

$$N \geq T_0 B \tag{A.24}$$

where T_0 is the observation interval and B denotes the band of interest. Under the simplifying assumption of having no uncertainty on the entering noise, the test statistic is approximated by a normal distribution $\mathcal{N}$ whether the signal is hypothesized present (H_1) or absent (H_0):

$$T(\mathbf{r}_i) \sim \begin{cases} \mathcal{N}\left(\sigma^2, \dfrac{2\sigma^4}{N}\right) & \text{if } H_0 \quad \text{(A.25a)} \\ \mathcal{N}\left(P+\sigma^2, \dfrac{2(P+\sigma^2)^2}{N}\right) & \text{if } H_1 \quad \text{(A.25b)} \end{cases}$$

where P is the received signal power and σ^2 is the noise variance that identify the SNR, $SNR = P/\sigma^2$. Hence, the energy detection is characterized by the following theoretical probabilities of false alarm P_{fa} and missed detection P_{md}:

$$P_{fa} = \text{Prob}\left\{T(\mathbf{r}_i) > \xi | H_0\right\} = Q\left(\frac{\xi - \sigma^2}{\sqrt{\frac{2}{N}}\sigma^2}\right) \tag{A.26}$$

$$P_{md} = \text{Prob}\left\{T(\mathbf{r}_i) < \xi | H_1\right\} = 1 - Q\left(\frac{\xi - (P+\sigma^2)}{\sqrt{\frac{2}{N}}(P+\sigma^2)}\right) \tag{A.27}$$

in which $Q(\cdot)$ represents the complementary cumulative density function of the standard normal distribution and ξ is the radiometer threshold. As a consequence of these approximations, any input signal results to be detectable at arbitrarily low SNR by increasing the number N of samples. In other words, the so-called SNR wall is neglected, as the energy detection is supposed robust to channel modelling uncertainties. By fixing a target P_{fa} it is possible to obtain a constant false alarm energy detector in which the test threshold may be computed by inverting eq. (A.26):

$$\xi = \sqrt{\frac{2}{N}}\sigma^2 Q^{-1}(P_{fa}) + \sigma^2 \tag{A.28}$$

where $Q^{-1}(\cdot)$ is the inverse of $Q(\cdot)$.

A.3 Estimation bounds

The Cramér-Rao bound (CRB) provides the estimation performance lower bound. When an EKF is used, the CRB depends on the accuracy of the deterministic and stochastic models.

A.3.1 Accuracy of TDOA and FDOA measurements

The accuracy of TDOA and FDOA estimation using CAF is given in [33]:

$$\sigma_{TDOA} = \frac{1}{\beta\sqrt{BTS_i}} \tag{A.29}$$

$$\sigma_{FDOA} = \frac{1}{T_e\sqrt{BTS_i}} \tag{A.30}$$

where B is the emitter signal bandwidth, T is the observation interval, S_i is the input SNR. β and T_e are the root mean square (RMS) bandwidth and duration of the emitter signal, respectively:

$$\beta = 2\pi\sqrt{\frac{\int_{-\infty}^{+\infty} f^2 W_s(f)\,\mathrm{d}f}{\int_{-\infty}^{+\infty} W_s(f)\,\mathrm{d}f}} \tag{A.31}$$

$$T_e = 2\pi\sqrt{\frac{\int_{-\infty}^{+\infty} t^2\,|a(t)|\,\mathrm{d}t}{\int_{-\infty}^{+\infty} |a(t)|\,\mathrm{d}t}} \tag{A.32}$$

where $W_s(f)$ denotes the emitter signal power spectral density and $|a(t)|$ indicates the emitter signal complex envelope. Using σ_{TDOA} and σ_{FDOA} it is possible to build the optimal EKF measurement error covariance matrix $\mathbf{R}_k$. The process noise covariance matrix $\mathbf{Q}_k$ can be optimized as described in §2.2.3. Once those two matrices are known, it is possible to compute the CRB.

A.3.2 Cramér Rao bound for EKF

The posterior CRB for EKF is given in [66]. In the specific case of TDOA/FDOA-based localization. The Fisher information matrix (FIM) can be computed iteratively as

$$\mathbf{J}_k = \mathbf{Q}_k^{-1}\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k - \left(\mathbf{A}_k^T\mathbf{Q}_k^{-1}\right)^T\left(\mathbf{J}_{k-1} + \mathbf{A}_k\mathbf{Q}_k^{-1}\mathbf{A}^T\right)^{-1}\mathbf{A}_k^T\mathbf{Q}_k^{-1} \tag{A.33}$$

The initial FIM $\mathbf{J}_0$ can be set according to the available a-priori information. The CRB is then

$$\mathbf{P}_k \geq CRB = \mathbf{J}_k^-1 \tag{A.34}$$

where $\mathbf{A} \geq \mathbf{B}$ indicates that $\mathbf{A} - \mathbf{B}$ is positive definite. In order to compare position errors and velocity estimation errors with the CRB we define the SR-CRB as

$$\sigma_{SR-CRB} = \sqrt{\mathrm{Tr}\,\mathbf{J}_k^{-1}} \tag{A.35}$$

A.4 Karhunen-Loéve expansion

KLE is a promising signal processing method because provides some advantages with respect to Fourier series (FS) Both KLE and FS provide a spectral decomposition of the signal using a set of basis functions, KLE is more flexible because its basis function can be of any form, the basis functions can be tailored to a specific signal in order to have a better spectral decomposition. FS basis functions are limited to sine and cosine [67] [40]. KLE merges deterministic and stochastic analyses of the signal; although the set of basis function is deterministic, it also provides stochastic information about the signal. KLE weighs the basis functions with respect to their probable power contribution, allowing one to efficiently distinguish the signal from the noise. On the other hand, FS uses only one parameter per basis function that represents its exact power. Moreover, KLE is able to detect much weaker signals than FS or Fourier Transform, because of the ability to separate stochastic and deterministic informations. The main disadvantage of KLE with respect to FS is the complexity. it is well known that FS?s complexity is of $O(n \cdot \log(n))$, while KLE complexity is much higher, $O(n^2)$. This difference is due to the basis functions: FS uses predefined basis functions, while KLE looks for the best representation for each individual signal.

A.4.1 From Fourier series to Karhunen-Loéve expansion

FS and KLE have many similarities, but also many differences. As well known any periodic signal can be represented as a Fourier series:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos\left(\omega_n t\right) + b_n \sin\left(\omega_n t\right) \right] \tag{A.36}$$

where the coefficients are

$$a_n = \frac{2}{T} \int_T x(t) \cos(\omega_n)\, \mathrm{d}t \tag{A.37}$$

$$b_n = \frac{2}{T} \int_T x(t) \sin(\omega_n)\, \mathrm{d}t \tag{A.38}$$

These coefficients are the projection of the basis functions (sine and cosine) on the signal. In an similar way, it is possible to represent a stochastic process $X(t)$ over the finite time interval $0 \leq t \leq T$ using KLE:

$$X(t) = \sum_{n=1}^{\infty} Z_n \Phi_n(t) \tag{A.39}$$

Now the coefficients Z_n are random variables, and the deterministic functions Φ_n are eigenfunctions. These can assume any form as long as they are orthonormal. Similarly to the coefficients of Fourier series, KLE coefficients are obtained projecting the signal onto the eigenfunctions:

$$Z_n = \int_0^T X(t)\Phi_n(t)\,\mathrm{d}t \tag{A.40}$$

Since Z_n are random variables, their variance is more important than the coefficients themselves. The variance is in fact the eigenvalue λ_n associated to the eigenfunction $\phi_n(t)$. Each eigenvalue represents the expected power of the corresponding eigenfunction. KLE computes at first the covariance of the processed signal and then allows to find eigenvalues and eigenfunctions:

$$\int_0^T \mathrm{E}\left\{X(t_1)X(t_2)\Phi_n(t_1)\,\mathrm{d}t_1\right\} = \lambda_n\Phi_n(t_2) \tag{A.41}$$

Assuming $\mathrm{E}\left\{X(t)\right\} = 0$, $\mathrm{E}\left\{X(t_1)X(t_2)\right\}$ is the autocovariance of the process $X(t)$ at the instants t_1 and t_2. This is the only known quantity of eq. (A.41); eigenvalues and eigenfunctions are unknown. If we represents $X(t)$ in the discrete-time, the same equation becomes:

$$\sum_{k=1}^{N} \mathrm{E}\left\{X_k X_l\right\}\Phi_{n,k}\Delta t = \lambda_n\Phi_{n,l} \tag{A.42}$$

which is a set of N linear equations in N unknowns; the autocovariance is now a Toeplitz matrix $\mathrm{E}\left\{X_k X_l\right\} \in \mathcal{M}_N(\mathbb{R})$ and Δt denotes the sampling interval. Equation (A.42) can be solved using linear algebra and always admits an unique solution.

References

[1] J. A. Volpe, "Vulnerability assessment of the transportation infrastructure relying on the global positioning system: Final report," Aug. 2001.

[2] K. Sheridan, T. Whitworth, G. Gabelli, R. Casile, A. Guidotti, G. E. Corazza, C. Hoelper, and G. Fremont, "Detection, evaluation and characterisation of threats to road applications (DETECTOR): Applications and threats analysis," June 2012, Deliverable D2.1.

[3] G. X. Gao, "DME/TACAN interference and its mitigation in L5/E5 bands," in *Proc. of the 20th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION-GNSS 2007)*, Forth Worth, TX, Sept. 2007.

[4] Thomas Kraus, Roland Bauernfeind, and Bernd Eissfeller, "Survey of in-car jammers-analysis and modeling of the rf signals and if samples (suitable for active signal cancelation)," in *Proc. of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, OR, Feb. 2011, pp. 430–435.

[5] Todd E Humphreys, Brent M Ledvina, Mark L Psiaki, Brady W O'Hanlon, and Paul M Kintner Jr, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Proc. of the ION GNSS international technical meeting of the satellite division*, 2008, vol. 55, p. 56.

[6] Ryan H Mitch, Ryan C Dougherty, Mark L Psiaki, Steven P Powell, Brady W O'Hanlon, Jahshan A Bhatti, and Todd E Humphreys, "Signal characteristics of civil gps jammers," in *Proc. 24th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS)*, Portland, OR, Sept. 2011, pp. 1907–1919.

[7] J. C. Grabowski, "Field observations of personal privacy devices," in *Proc. of the 2012 International Technical Meeting of The Institute of Navigation*, Newport Beach, CA, Jan. 2012, pp. 689–741.

[8] C Tedeschi, "The newark liberty international airport (ewr) gbas experience," in *12th Int'l. GBAS Working Group Meeting (I-GWG-12), Atlantic City, NJ*, 2011.

[9] G. Gabelli, R. Casile, A. Guidotti, and G. E. Corazza, "GNSS jamming interference: Characterization and cancellation," in *26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2013)*, San Diego, California, Jan. 2013, pp. 828 – 834.

[10] F. Bastide, D. Akos, C. Macabiau, and B. Roturier, "Automatic gain control (AGC) as an interference assessment tool," in *Proc. 16th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS)*, Portland, OR, Sept. 2003, pp. 2042–2053.

[11] A.T. Balaei, "Statistical inference technique in pre-correlation interference detection in GPS receivers," in *Proc. 19th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS)*, Fort Worth, TX, Sept. 2006, pp. 2232–2240.

[12] A. Tani and R. Fantacci, "Performance evaluation of a precorrelation interference detection algorithm for the GNSS based on nonparametrical spectral estimation," *IEEE Syst. J.*, vol. 2, no. 1, pp. 20–26, Mar. 2008.

[13] L. Marti and F. van Graas, "Bias detection and its confidence assessment in global positioning system signals," in *Proc. IEEE Aerospace Conf.*, Mar. 2004, vol. 3, pp. 1608–1617.

[14] P. T. Capozza, B. J. Holland, T. M. Hopkinson, and R. L. Landrau, "A single-chip narrow-band frequency-domain excisor for a global positioning system (GPS) receiver," *IEEE J. of Solid-State Circuits*, vol. 35, no. 3, pp. 401–411, Mar. 2000.

[15] G. Dimos, T. Upadhyay, and T. Jenkins, "Low-cost solution to narrowband GPS interference problem," in *IEEE Proc. Nat. Aerospace and Electronics Conf. (NAECON)*, Dayton, OH, May 1995, vol. 1, pp. 145–153.

[16] D. Borio, L. Camoriano, S. Savasta, and L. L. Presti, "Time-frequency excision for GNSS applications," *IEEE Syst. J.*, vol. 2, no. 1, pp. 27–37, Mar. 2008.

[17] S. Savasta, L. Lo Presti, and M. Rao, "Interference mitigation in GNSS receivers by a time-frequency approach," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 49, no. 1, pp. 415–438, Jan. 2013.

[18] K. Sun, M. Zhang, and D. Yang, "A new interference detection method based on hybrid time-frequency distribution for GNSS receivers," *IEEE Trans. Veh. Technology*, 2016.

[19] F. D. Nunes and F. M. G. Sousa, "Jamming detection in GNSS signals using the sample covariance matrix," in *IEEE Proc. 6th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, Dec. 2012, pp. 1–8.

[20] B. Motella, M. Pini, and L. L. Presti, "GNSS interference detector based on chi-square goodness-of-fit test," in *IEEE Proc. 6th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, Dec. 2012, pp. 1–6.

[21] A. Ndili and P. Enge, "GPS receiver autonomous interference detection," in *IEEE Position Location and Navigation Symp.*, Palm Springs, CA, Apr. 1998, pp. 123–130.

[22] Frederic Bastide, Eric Chatre, and Christophe Macabiau, "GPS interference detection and identification using multicorrelator receivers," in *Proc. 14th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GPS)*, Salt Lake City, UT, Sept. 2001, pp. 872–881.

[23] Youngsun Yun, Changdon Kee, Jason Rife, Ming Luo, Sam Pullen, and Per Enge, "Detecting RFI through integrity monitoring at a DGPS reference station," in *Proc. 61st Annu. Meeting of The Inst. of Navigation*, Cambridge, MA, June 2005, pp. 795–804.

[24] R. Calcagno, S. Fazio, S. Savasta, and F. Dovis, "An interference detection algorithm for COTS GNSS receivers," in *IEEE Proc. 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, Dec. 2010, pp. 1–8.

[25] E. Axell, F. M. Eklöf, M. Alexandersson, P. Johansson, and D. M. Akos, "Jamming detection in GNSS receivers: Performance evaluation of field trials," in *Proc. 26th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS+)*, Nashville, TN, Sept. 2013, pp. 2542–2551.

[26] Y. Ying, T. Whitworth, and K. Sheridan, "GNSS interference detection with software defined radio," in *IEEE First AESS European Conf. on Satellite Telecommun. (ESTEL)*, Rome, Italy, Oct. 2012, pp. 1–6.

[27] D. Borio and C. Gioia, "Real-time jamming detection using the sum-of-squares paradigm," in *IEEE Proc. Int. Conf. on Localization and GNSS (ICL-GNSS)*, Gothenburg, Sweden, June 2015, pp. 1–6.

[28] R. J. R. Thompson, A. T. Balaei, and A. G. Dempster, "Dilution of precision for GNSS interference localisation systems," in *European Navigation Conf. (ENC GNSS)*, Naples, Italy, May 2009, pp. 1–11.

[29] Jonas Lindstrom, Dennis M. Akos, Oscar Isoz, and Marcus Junered, "GNSS interference detection and localization using a network of low cost front-end modules," in *Proc. 20th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS)*, Fort Worth, TX, Sept. 2007, pp. 1165–1172.

[30] O. Isoz and D. Akos, "Development of a deployable low cost interference detection and localization system for the GNSS L1/E1 band," in *IEEE Proc. 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, Dec. 2010, pp. 1–4.

[31] J. P. Poncelet and D. M. Akos, "A low-cost monitoring station for detection & localization of interference in GPS L1 band," in *IEEE Proc. 6th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, Dec. 2012, pp. 1–6.

[32] R. H. Mitch, M. L. Psiaki, B. W. OHanlon, S. P. Powell, and J. A. Bhatti, "Civilian GPS jammer signal tracking and geolocation," in *Proc. 25th Int. Tech. Meeting of The Satellite Division of the Inst. of Navigation (ION GNSS)*, Nashville, TN, Sept. 2012, pp. 2901–2920.

[33] S. Stein, "Algorithms for ambiguity function processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 588–599, Jun 1981.

[34] Brian D. O. Anderson and John B. Moore, *Optimal filtering*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[35] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. 15, no. 2, pp. 175–184, Apr 1970.

[36] Bernt M Åkesson, John Bagterp Jørgensen, Niels Kjølstad Poulsen, and Sten Bay Jørgensen, "A generalized autocovariance least-squares method for Kalman filter tuning," *Journal of Process control*, vol. 18, no. 7, pp. 769–779, 2008.

[37] G. Marsaglia, W. W. Tsang, and J. Wang, "Evaluating Kolmogorov's distribution," *Journal of Statistical Software*, vol. 8, no. 18, pp. 1–4, 2003.

[38] Thomas Pany, *Navigation signal processing for GNSS software receivers*, Artech House, Norwood, 2010.

[39] F. Dovis and L. Musumeci, "Use of wavelet transforms for interference mitigation," in *IEEE Proc. Int. Conf. on Localization and GNSS (ICL-GNSS)*, Tampere, Finland, June 2011, pp. 116–121.

[40] A Szumski, "Finding the interference–the karhunen–loève transform as an instrument to detect weak rf signals," *InsideGNSS*, vol. 3, pp. 56–64, 2011.

[41] L. Musumeci and F. Dovis, "A comparison of transformed-domain techniques for pulsed interference removal on gnss signals," in *2012 International Conference on Localization and GNSS*, June 2012, pp. 1–6.

[42] Wm. C. Jakes, *Microwave mobile communications*, Wiley, New Yourk, 1974.

[43] Mohinder S Grewal, Lawrence R Weill, and Angus P Andrews, *Global positioning systems, inertial navigation, and integration*, John Wiley & Sons, 2007.

[44] Bruno M Scherzinger, "Quasi tightly coupled GNSS-INS integration process," Sept. 2 2014, US Patent 8,825,396.

[45] C. Palestini, L. Deambrogio, F. Bastia, and G. E. Corazza, "An insider view on tracking loops: A novel ultra-tight GNSS/INS hybridization approach," in

IEEE/ION Position, Location and Navigation Symposium, May 2010, pp. 1093–1099.

[46] Peter Matisko and Vladimír Havlena, "Noise covariance estimation for kalman filter tuning using bayesian approach and monte carlo," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 11, pp. 957–973, 2013.

[47] Brian J Odelson, Murali R Rajamani, and James B Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, no. 2, pp. 303–308, 2006.

[48] Davide Dardari, Pau Closas, and Petar M Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[49] D. Chen, W. Pan, P. Jiang, J. Jin, T. Mo, and J. Zhou, "Reconfigurable dual-channel multiband RF receiver for GPS/Galileo/BD-2 systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 11, pp. 3491–3501, Nov 2012.

[50] L. Deambrogio, C. Palestini, F. Bastia, G. Gabelli, G. E. Corazza, and J. Samson, "Impact of high-end receivers in a peer-to-peer cooperative localization system," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, Oct 2010, pp. 1–7.

[51] Jose Antonio Garcia-Molina and Massimo Crisci, "Cloud-based localization of GNSS jammers," in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, September 2015, pp. 3289 – 3295.

[52] SDR Forum, "SDRF cognitive radio definitions," `http://www.sdrforum.org/pages/documentLibrary/documents/SDRF-06-R-0011-V1_0_0.pdf`, 2007, [Online; accessed 09-June-2016].

[53] "RTL-SDR.com, RTL-SDR (RTL2832U) and software defined radio news and projects. also featuring Airspy, HackRF, FCD, SDRplay and more.," `http://www.rtl-sdr.com`, [Online; accessed 09-June-2016].

[54] "HackRF One, an open source SDR platform," `https://greatscottgadgets.com/hackrf/`, [Online; accessed 09-June-2016].

[55] "bladeRF - the USB 3.0 Superspeed Software Defined Radio," http://nuand.com, [Online; accessed 09-June-2016].

[56] "tmpfs documentation," https://www.kernel.org/doc/Documentation/filesystems/tmpfs.txt, [Online; accessed 09-June-2016].

[57] "GPSD time service howto," http://www.catb.org/gpsd/gpsd-time-service-howto.html, [Online; accessed 09-June-2016].

[58] Kai Borre, Dennis M Akos, Nicolaj Bertelsen, Peter Rinder, and Søren Holdt Jensen, *A software-defined GPS and Galileo receiver: a single-frequency approach*, Springer Science & Business Media, 2007.

[59] Frank Stephen Tromp Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*, Artech House, 2009.

[60] J. A. del Peral-Rosado, J. M. Parro-Jiménez, J. A. López-Salcedo, G. Seco-Granados, P. Crosta, F. Zanier, and M. Crisci, "Comparative results analysis on positioning with real LTE signals and low-cost hardware platforms," in *2014 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Dec 2014, pp. 1–8.

[61] J. A. del Peral-Rosado, J. A. López-Salcedo, G. Seco-Granados, P. Crosta, F. Zanier, and M. Crisci, "Downlink synchronization of LTE base stations for opportunistic ToA positioning," in *2015 International Conference on Location and GNSS (ICL-GNSS)*, June 2015, pp. 1–6.

[62] "Evrytania LLC, LTE cell scanner," https://github.com/Evrytania/LTE-Cell-Scanner, [Online; accessed 09-June-2016].

[63] A. C. Cohen, "Estimating the mean and variance of normal populations from singly truncated and doubly truncated samples," *Ann. Math. Statist.*, vol. 21, no. 4, pp. 557–569, 12 1950.

[64] F. M. Ham and R. G. Brown, "Observability, eigenvalues, and kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 2, pp. 269–273, March 1983.

[65] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, April 1967.

[66] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior cramer-rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[67] Ian Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.

Acknowledgments

Questa tesi è il risultato di tre anni di lavoro. Vorrei ringraziare le molte persone che mi sono state vicine durante questo percorso e che mi hanno aiutato a crescere.

Ringrazio Giovanni che ha creduto in me e mi ha dato la possibilitá di intraprendere questo percorso. Da lui ho appreso tanto.

Thanks to Massimo Crisci and José Garcia who hosted me in ESTEC. It has been a wonderful opportunity to grow. Thanks to the people I worked with while I was there.

Ringrazio i colleghi del laboratorio con cui ho condiviso questi tre anni.

Grazie a tutti gli amici di Bologna e del paesello per i tanti bei momenti passati insieme e per il supporto nei momenti piú difficili.

Infine grazie ai miei genitori che mi hanno sempre supportato e sostenuto.