

POLITECNICO DI TORINO

Master's Degree in Communications and Computer
Networks Engineering



Master's Degree Thesis

Use of Particle Filters for Cooperative GNSS Positioning

Supervisors

Prof. Fabio DOVIS

Dr. Alex MINETTO

Candidate

Simone ZOCCA

ID: 265384

Academic Year 2019 / 2020

Abstract

In the context of vehicle-positioning applications, Global Navigation Satellite Systems (GNSSs) have remarkable role. However, these applications have very strict safety requirements, thus needing an improvement in the performance of such positioning systems. This necessity has led to the development of Cooperative Positioning methods, also thanks to the recent rise of Vehicle-To-Vehicle (V2V) communication. The aim of such methods is in fact to improve both the accuracy and precision of the stand-alone positioning system by exploiting the exchange of relative ranging information by a network of vehicles.

Moreover, in spite of the fact that the positioning problem is non-linear (i.e. trilateration), many solutions approach it by means of linearization (i.e. Least Mean Square, Extended Kalman Filter). These methods may also introduce errors by assuming that general probability distributions of the input measurements are Gaussian distributions.

However, the error of these relative distances derived from the measurement exchanged by the vehicles have shown to be, in general, not Gaussian distributed, leading to a reduction in performance due to the mismodelling of such probability distributions. This has prompted the study of other solutions (i.e. Particle Filter), able to handle the problem without linearization, and also the non-Gaussian distribution of the measurement errors. Most importantly, these distributions are also non-stationary, thus requiring a real-time estimation in order to allow the Particle Filter to provide the best possible solution. In particular, the errors can be affected by a bias, and it is the aim of the thesis to understand whether the relative motion of the vehicles can affect the bias introduced in the measurements they exchange. A further goal is to design an adaptive algorithm to select optimal likelihood functions, based on the relative position, motion and GNSS measurements of the vehicles, in order to improve the estimate provided by the Particle Filter. Finally, the performance of the previously mentioned solutions is evaluated, to assess if the use of the Particle Filter is justified even when the cooperative measurements can be approximated to be Gaussian distributed.

Eventually, an optimized integration of cooperative ranging measurements is performed in order to complement satellite-based measurements. This approach aims at compensating for the limited availability and high geometrical dilution of precision that are frequently experienced in urban environments. With this intent, an Agent Network (AN) is implemented, and scenarios in which such agents are static or moving are both studied, in order to gain a better understanding of the effect that their relative motion can cause on the distribution of the measured distances. The collaboration between agents is implemented with an exchange,

at each time instant, of both their estimated positions and their available GNSS measurements (i.e. Doppler and pseudoranges). Thanks to these, inter-agent distances are then computed by means of Weighted Least Square Double Difference method. The integration of this additional information is expected to provide an improvement in terms of accuracy of the positioning solution.

Table of Contents

1	GNSS Overview	1
1.1	GNSS Fundamentals	1
1.2	System Structure	2
1.2.1	Space Segment	2
1.2.2	Ground Segment	2
1.2.3	User Segment	3
1.3	GNSS Signal	3
1.3.1	Carrier	4
1.3.2	Ranging Codes	5
1.3.3	Navigation Data	6
1.4	GNSS Receiver	7
1.4.1	Front-End Stage	7
1.4.2	Acquisition Stage	8
1.4.3	Tracking Stage	9
1.5	PVT Computation	9
1.5.1	PT Estimation using Pseudoranges	10
1.5.2	Velocity Estimation using Doppler Shift	12
1.5.3	Quality of the PVT Solution	12
2	Cooperative DGNSS	15
2.1	Differential Methods	17
2.1.1	Synchronization of Pseudoranges	18
2.1.2	Double Difference of Pseudoranges	19
2.1.3	Weighted Least Square Double Difference	22
3	Bayesian Estimation for Hybrid PVT	25
3.1	Kalman Filter	25
3.1.1	Extended Kalman Filter Routine	26
3.2	Particle Filter	28
3.2.1	Particle Filter Algorithm	28

4	Error Modelling for Cooperative GNSS	31
4.1	Simulation Scenario	32
4.1.1	Ranging error for Static and Kinematic agent	33
4.1.2	Dependence on Motion	34
4.1.3	Dependence on Relative Motion	35
4.1.4	Dependence on Time	37
4.1.5	Bias Removal	39
4.2	Circular Dataset	42
4.2.1	Model Adaptation	42
4.2.2	Doppler-based Model	44
4.3	Analysis of Receiver Architecture for Bias Correction	45
4.4	Aiding Agent Position Synchronization	47
4.4.1	Performance of Position Synchronization methods	49
5	Automatic Adaptive Likelihood Switch	51
5.1	Baseline Length Error Distribution	51
5.2	Likelihood Distribution Families	53
5.2.1	GEV Distribution	53
5.2.2	Rayleigh Distribution	54
5.3	Mahalanobis Distance	55
5.3.1	Mahalanobis Distance Definition	56
5.4	Automatic Adaptive Likelihood Switch Algorithm	57
5.4.1	AALS Threshold Value	58
5.4.2	AALS Implementation	60
5.5	AALS Performance Evaluation	61
5.5.1	Distribution Families for Distant Peers	61
5.5.2	Distribution Families for Close Peers	62
5.5.3	Performance Comparison	65
6	Conclusions	67
	Bibliography	69

Chapter 1

GNSS Overview

1.1 GNSS Fundamentals

Global Navigation Satellite System (GNSS) can be described as a one-way radionavigation system, where satellites broadcast synchronized timing signals that are exploited by receivers in order to estimate their Position, Velocity and Time (PVT) based on passive ranging measurements. These satellites are typically Medium Earth Orbit (MEO), although some regional or augmentation system use Geosynchronous Earth Orbit (GEO).

In particular, Global Positioning System (GPS) is a GNSS owned and maintained by the Federal Aviation Administration (FAA), which operates on a global scale. Other GNSS systems include European Union's Galileo, Russia's GLONASS and China's BeiDou.

Any GNSS receiver starts from the identification of the visible satellites (i.e. acquisition of line-of-sight navigation signals), w.r.t. which an estimation of a satellite-user distance called pseudorange is obtained. The position of satellites is known at any time instant, or can be accurately computed thanks to the data carried in their navigation message. Pseudorange measurements are then used to obtain a Position and Time (PT) solution by means of a spherical trilateration method, while velocity can be computed using the variation rate of the pseudoranges, which is related to the Doppler shift measurements due to the relative velocity of satellite and receiver [1].

Although these GNSS have slightly different features and work according to different system parameters, they are based on the same basic concepts. As such, this Chapter will focus mainly on GPS, in order to give a basic understanding of the working principles of a satellite-based radio navigation positioning system.

1.2 System Structure

Any GNSS is made up of three main segments: the Space segment which includes the satellites, the Control segment composed of both control and monitor ground stations, and finally the User segments which includes both military and civilian users with a wide range of different receiver devices [1].

1.2.1 Space Segment

The GPS Space segment includes a constellation of at most 32 satellites, distributed across six circular orbits with at least four satellite each, as shown in Figure 1.1. The orbits have an inclination of 55° degrees w.r.t. the earth's equator and are separated by 60° right ascension, with satellites not equally spaced within each one. The orbit is approximately circular with a semi-major axis of 26,560 km (typically classified as MEO), while its period is of around 12 hours, so that each satellite passes over roughly the same location twice a day. Each satellite has a highly stable atomic clock on board. The configuration of the orbits is such that at least 4 satellites are always in line-of-sight from each point of the earth's surface, provided there are no obstacles. This number corresponds to the minimum number of satellites w.r.t. which a pseudorange needs to be measured in order to obtain a position solution [1] [2].

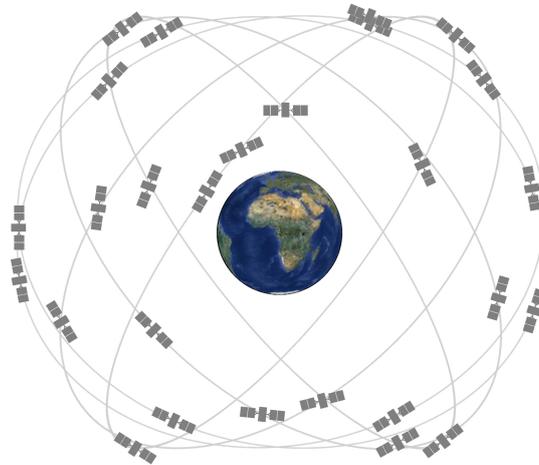


Figure 1.1: GPS constellation [3].

1.2.2 Ground Segment

The Ground segment is handled by the Master Control Station, whose purpose is to remotely operate monitor stations, as can be seen from Figure 1.2. These

dedicated monitor station track the flight trajectories of all satellites, then this information is used to regularly contact satellites from ground antennas, providing them navigation updates along with the navigation message the satellites will transmit [4]. These updates are necessary to make sure the atomic clocks on board are kept synchronized within a few nanoseconds with each other. Also, they are used to adjust the ephemeris of the internal orbit model of each satellite. These updates are applied by satellite maneuvers performed in order to correct its orbit. During these corrections, the satellite is marked in order to not be used by receivers for the pseudorange computation [1].

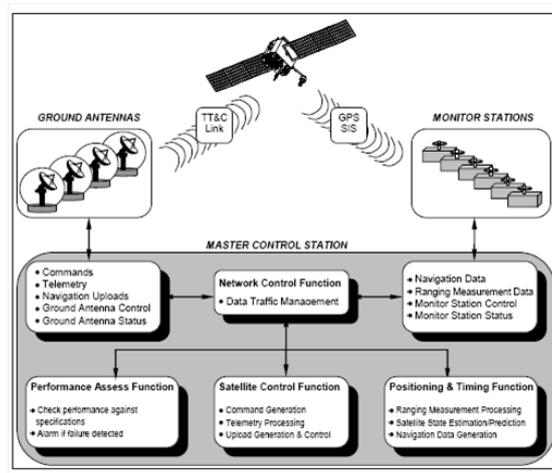


Figure 1.2: GPS Ground Segment [5].

1.2.3 User Segment

The User segment refers to the collection of all the military and civilian GNSS receivers. These devices consists of a receiving antenna, a receiver processor able to monitor multiple satellites simultaneously, and a clock. The performance of GPS navigation devices can vary greatly depending on the quality of its components [1] [2]. A more detailed description of the main stages of a GNSS receiver and their purpose is provided in Section 1.4.

1.3 GNSS Signal

Each satellite in the GPS constellation continuously transmits a signal which is made of three components: a carrier, a ranging code which uniquely identifies the satellite, and the navigation data. The structure of the overall transmitted signal

is shown in Figure 1.3.

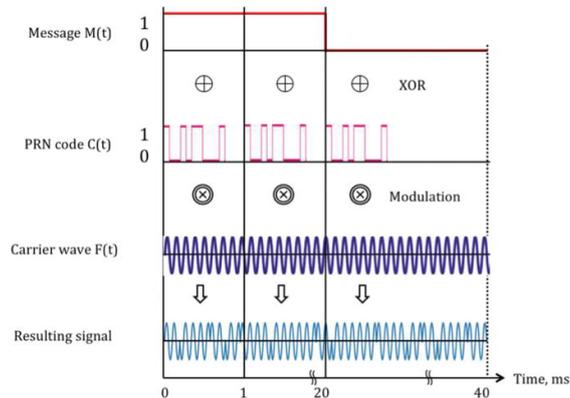


Figure 1.3: GPS signal structure [6].

1.3.1 Carrier

All GPS satellites transmit two spread spectrum carrier signals on the same frequency bands, referred to as L1 and L2. The first is used for Standard Positioning Service, so for civil usage, while the second is used for Precise Positioning Service, provided to the United States military and its allied organizations. The carrier frequency of the two bands, respectively $f_1 = 1575.42$ MHz and $f_2 = 1227.60$ MHz, are both integer multiples of a base frequency $f_0 = 1.023$ MHz ($f_1 = 1540f_0$ and $f_2 = 1200f_0$). It should be added that, since the satellites are moving, and possibly the receivers as well, the received signal is affected by a Doppler shift.

In recent years, the L2 band has also been grant access to civil application in order to calculate and compensate for propagation delays. When the signal transmitted by the satellites travels through the ionosphere, it is delayed by an amount that depends on the density of electrons along the path of the signal, but also on the frequency of the signal itself. By measuring the difference in delay between two signals transmitted at different carrier frequencies, the receivers are able to fully compensate for this delay, which would otherwise be a source of error [2]. A third civilian signal is also broadcasted on the L5 ($f_5 = 1150f_0$) band by 14 GPS satellites (as of September 2020). This signal is transmitted at higher power, and on a grater bandwidth in order to be more robust to jamming (intentional interference). Furthermore, its more modern signal design (CNAV) allow for different message types and Forward Error Correction (FEC), in the attempt to satisfy more strict requirements involving safety-of-life or other high-performance application. Modern GNSS receivers are able to simultaneously work with signals from different bands, leading to improved accuracy and robustness [7].

1.3.2 Ranging Codes

Satellite networks employ a Code Division Multiple Access (CDMA) spread-spectrum technique, where data is encoded by pseudo-random sequences, also known as PRN codes (as depicted in Figure 1.3), that are unique for each satellite. These codes are also mutually orthogonal, thus allowing the receiver to separate the signal coming from each satellite [1].

The C/A code, used for civilian applications, is a short and coarse-grained code, whose purpose is to allow a fast identification and acquisition of the satellite signal. It has a chip rate equal to $f_0 = 1.023$ Mchips/s, and belongs to the family of Gold sequences, used for their good correlation properties. Maximal Length Linear Shift Registers can be used to generate m-sequences, which are then summed to obtain Gold sequences. Different C/A codes are obtained by summing different shifted versions of the same m-sequences, obtained from shift registers of size $N = 10$, therefore creating sequences of length $2^N - 1 = 1023$ chips.

The P-code is instead a unique segment of an extremely long PRN sequence of around 10^{14} chips which lasts for 267 days. The sequence is split into 37 sections each lasting one week before repeating. The chip rate of the P-code is $10f_0 = 10.23$ Mchips/s.

The signal transmitted on the L1 band uses Binary Phase-Shift Keying (BPSK), and is further modulated by two PRN codes in quadrature, both the Coarse Acquisition (C/A) code and the Precision (P(Y)) code, as can be seen in Figure 1.4. The L2 signal is instead modulated by the P(Y) code only, since its intended use is for military applications only.

GPS receivers have full knowledge of the C/A codes, which can be locally generated and used to identify satellites by correlation between the local PRN code and the received signal [2].

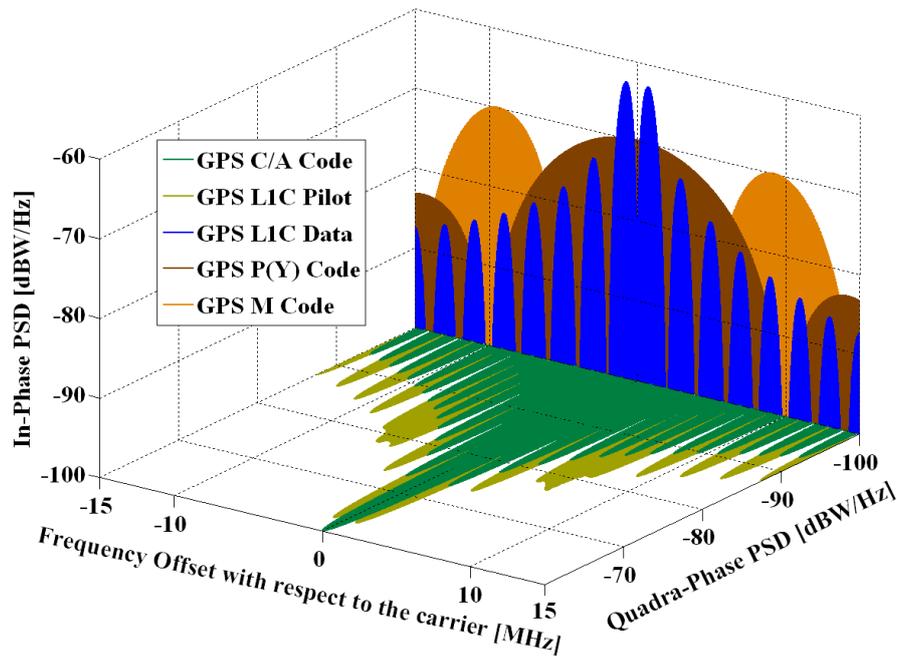


Figure 1.4: Spectra of GPS signal on L1 band [8].

1.3.3 Navigation Data

The navigation data is a binary-coded message that includes information about the status of the satellite, as well as its ephemeris and clock parameters, ionospheric correction model parameters, relativistic corrections, and an almanac containing low-precision ephemeris for all the other satellites of the constellation [1]. Data is transmitted at a rate of 50 bits/s, so that each bit lasts for 20 ms. The navigation message is structured into words which are contained into sub-frames which form a frame (or page) whose total duration is of 30 seconds, as shown in Figure 1.5. Along with data channels, the Galileo system also transmits pilot channels, shifted in phase by 90 degrees w.r.t. data channels so that they can be separated at receiver. These pilot channels use long codes, and contain no data.

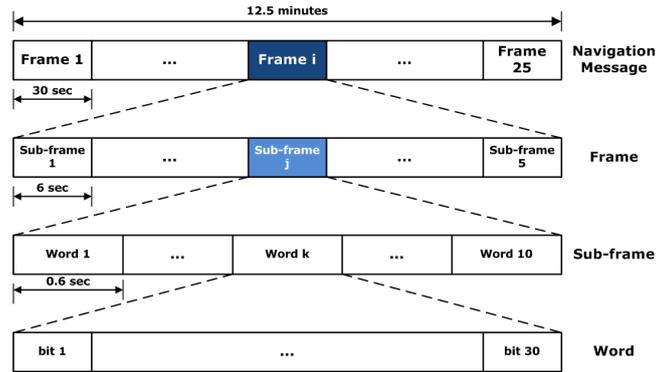


Figure 1.5: Overall structure of GPS navigation message [9].

1.4 GNSS Receiver

The purpose of a navigation receiver is to provide an estimate of Position, Velocity and Time, based on the information carried by the signals sent by the satellites. Its architecture can be divided in different stages, as depicted in Figure 1.6.

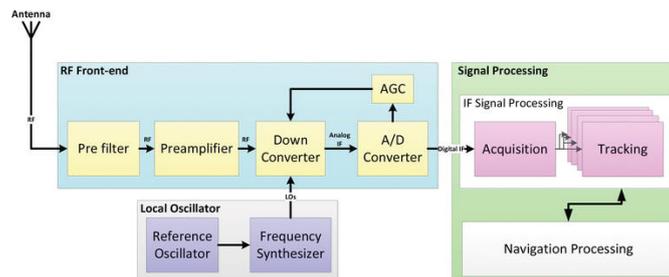


Figure 1.6: Generic architecture of a GPS receiver [10].

1.4.1 Front-End Stage

At first, the signal coming from the satellites is pre-amplified and filtered in order to limit the bandwidth, the receiver then needs to move the signal to Intermediate Frequency f_{IF} , in order to perform sampling and quantization for the following digital stages [1]. The signals can then be processed in Software Defined Radio (SDR), as it is done for the work presented in this thesis.

1.4.2 Acquisition Stage

In order to obtain measurements needed for the PVT computation, the set of visible satellites needs to be determined so that the receiver can split and distinguish the signal coming from each of them. The number of codes to be searched can be narrowed down to the set of satellites which should be visible, if rough information about the user position and a recent almanac are available. To acquire the signal coming from each satellite, the receiver generates a replica of the known C/A code, as described in Section 1.3.2. If the locally generated sequence corresponds to the one of a visible satellite, then in the cross-correlation between the local replica and the incoming signal, a peak should appear when the two sequences are aligned in time. In order to determine this shift, the correlation of all possible shifts between the two signals has to be computed [1]. Alternatively, the properties of the Discrete Fourier Transform can be exploited to obtain the circular cross-correlation. This approach requires the storage of L samples from the incoming signals, but provides the L values of the cross-correlation in one step. Since the Fast Fourier Transform (FFT) is an efficient implementation of the DFT, this approach can be computationally efficient for long sequences, w.r.t. the linear cross-correlation. Furthermore, the local replica is multiplied by a local carrier whose frequency varies over a search space centered around f_{IF} . Based on the values of the correlation between the two signal over this bi-dimensional search space of time (code delay) and frequency (Doppler shift), the receiver should decide whether the satellite is present, and in that case give a first estimate of delay and Doppler shift based on the position of the peak as can be seen in Figure 1.7.

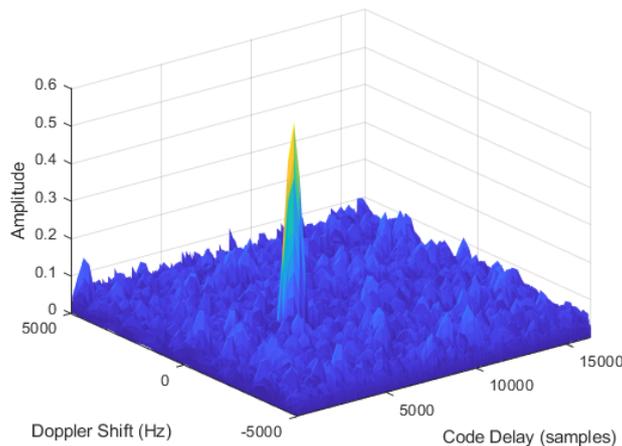


Figure 1.7: Example of correlation peak in the search space.

1.4.3 Tracking Stage

The tracking stage works as a feedback control loop. Given the rough delay and Doppler shift estimation, a code wipe off can be performed by subtracting the aligned local replica of the code from the incoming signal. The result should be a clean sine wave, which can be used to obtain a finer estimation of the frequency, and hence of the Doppler shift. In the same way, a carrier wipe off is obtained by subtracting from the incoming signal a local carrier with the same frequency. This is possible thanks to the Doppler shift estimation obtained through the acquisition stage. Once again, the results should be a clean code signal, that allows for a finer estimation of the delay.

At each iteration of these steps, a new estimation, possibly more accurate, of Doppler shift and delay is obtained, which can in turn improve the code and carrier wipe off in the next iteration. These two loops are called Phase Lock Loop and Delay Lock Loop, and are initiated thanks to the previous estimation of parameters performed in the acquisition stage. As the name suggests, these loops allow to track the changes in delay and Doppler shift continuously, as the satellite moves [1].

1.5 PVT Computation

Once the shift between the received signal and the local replica has been obtained, the distance between the satellite and the receiving device can be estimated. The PRN codes are transmitted by the satellites at precisely known time instants, and the information about their transmission time is contained in the navigation data. The time of reception of the sequence can instead be retrieved from the receiver clock. The difference between these two times is the Time of Flight (ToF) of the signal, which is multiplied by the speed of light c to obtain a measure of pseudorange between the satellite and the receiver. After the receiver has been able to detect the visible satellites, and has managed to obtain pseudorange measurements from at least four satellite, a PVT solution can be computed. The reason why at least four pseudoranges are needed, is that the user has to estimate 4 unknowns: its three coordinates (x_u, y_u, z_u) defined in a Cartesian reference systems, and the bias of its clock w.r.t. the GPS time δt_u [1].

1.5.1 PT Estimation using Pseudoranges

The pseudorange measurement obtained from user u w.r.t. satellite s can be written as

$$\rho_u^s = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2 + (z_s - z_u)^2} + b_{ut} \quad (1.1)$$

where the bias $b_{ut} = c \cdot \delta t_u$ is computed as the product between the clock bias δt_u and the speed of light. Therefore, ρ_u^s is the geometrical distance between the receiver and the satellite plus the bias b_{ut} . The position of the satellite $\mathbf{p}_s = (x_s, y_s, z_s)$ can be derived from the information present in the navigation data sent by the satellite itself, while the pseudorange value is measured by the receiver, thus leaving four unknown quantities. Since, as mentioned in Section 1.5, the pseudorange measurements is affected by errors as well, (1.1) can also be re-written as

$$\rho_u^s = r_u^s + b_{ut} + \epsilon \quad (1.2)$$

where ϵ account for all the remaining errors which are not corrected. The generic formula of the pseudorange given in (1.1) can be approximated by Taylor expansion around a known location and time $\hat{\mathbf{p}}_u = (\hat{x}_u, \hat{y}_u, \hat{z}_u, \hat{b}_{ut})$ as shown in Figure 1.8.

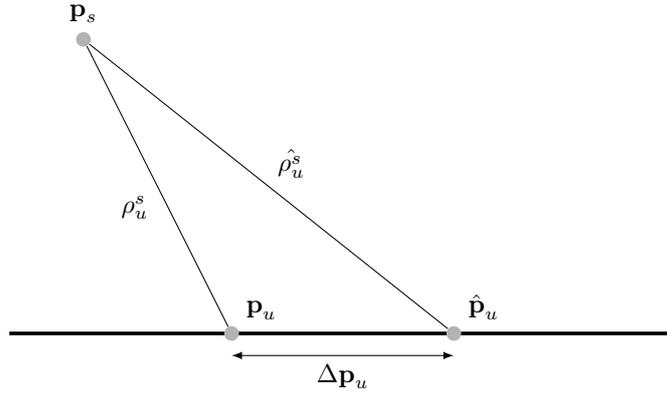


Figure 1.8: Linearization around known position $\hat{\mathbf{p}}_u$.

The difference between the approximation point and the receiver position is $\Delta \mathbf{p}_u = \hat{\mathbf{p}}_u - \mathbf{p}_u$. Then, a first order approximation yields

$$\Delta \rho_u^s = \hat{\rho}_u^s - \rho_u^s = a_{xs} \Delta x_u + a_{ys} \Delta y_u + a_{zs} \Delta z_u - \Delta b_{ut} \quad (1.3)$$

where the coefficients are defined as

$$a_{xs} = \frac{x_s - \hat{x}_u}{\hat{r}_u^s}, \quad a_{ys} = \frac{y_s - \hat{y}_u}{\hat{r}_u^s}, \quad a_{zs} = \frac{z_s - \hat{z}_u}{\hat{r}_u^s} \quad (1.4)$$

$$\hat{r}_u^s = \sqrt{(x_s - \hat{x}_u)^2 + (y_s - \hat{y}_u)^2 + (z_s - \hat{z}_u)^2}.$$

Clearly from (1.4), \hat{r}_u^s is the Euclidean distance between the satellite and the linearization point. In this way, the solution of the system of four equation (one for each pseudorange) can be obtained starting from an estimated point and then solving iteratively. The system of the equations in (1.3) can be written in matrix notation as

$$\Delta\rho = \mathbf{H} \cdot \Delta\mathbf{p} \quad (1.5)$$

where the matrix \mathbf{H} contains the coefficients defined in (1.4) for each of the N visible satellites

$$\mathbf{H} = \begin{bmatrix} a_x^{(1)} & a_y^{(1)} & a_z^{(1)} & 1 \\ a_x^{(2)} & a_y^{(2)} & a_z^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_x^{(N)} & a_y^{(N)} & a_z^{(N)} & 1. \end{bmatrix} \quad (1.6)$$

In case $N = 4$, then the solution can immediately be found by inverting (1.5) to obtain $\Delta\mathbf{p} = \mathbf{H}^{-1}\Delta\rho$. If instead $N > 4$, then a least square solution has to be used and is given by the value of $\Delta\mathbf{p}$ which minimizes the square of the residual

$$\mathbf{R}_{SE}(\Delta\mathbf{p}) = (\mathbf{H}\Delta\mathbf{p} - \Delta\rho)^2. \quad (1.7)$$

By differentiating w.r.t. $\Delta\mathbf{p}$, the gradient of \mathbf{R}_{SE} is obtained

$$\nabla\mathbf{R}_{SE} = 2(\Delta\mathbf{p})^T\mathbf{H}^T\mathbf{H} - 2(\Delta\rho)^T\mathbf{H}. \quad (1.8)$$

From (1.8), the transpose is taken and, and since the minimum value of \mathbf{R}_{SE} has to be found, the gradient is set to zero

$$2\mathbf{H}^T\mathbf{H}(\Delta\mathbf{p}) - 2\mathbf{H}^T(\Delta\rho) = 0. \quad (1.9)$$

Under the assumption that $(\mathbf{H}^T\mathbf{H})^{-1}$ is non-singular, then the final solution is found as

$$\Delta\mathbf{p} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta\rho. \quad (1.10)$$

This obtained solution is a Maximum Likelihood (ML) solution, which minimizes the quadratic error of the approximation point [1].

1.5.2 Velocity Estimation using Doppler Shift

In case the velocity of the receiver has to be estimated, pseudorange measurements do not provide any information about their dynamics, since it only measure their distance w.r.t. the satellites. Instead, Doppler shift measurements can be used, as they contain information about the user-to-satellite relative velocity. The frequency of the signal received by i from satellite s can be modeled as

$$f_u^s = f_c \left(1 - \frac{\mathbf{v}_u^s \cdot \mathbf{e}_u^s}{c} \right) \quad (1.11)$$

where f_c is the nominal carrier frequency as described in Section 1.3.1, while \mathbf{e}_u^s is the steering vector pointing from receiver u to satellite s , whose coefficients are computed in (1.4). \mathbf{v}_u^s is the user-to-satellite relative velocity.

It follows that the Doppler shift can be computed as

$$\Delta f_u^s = f_u^s - f_c = -f_c \frac{\mathbf{v}_u^s \cdot \mathbf{e}_u^s}{c}. \quad (1.12)$$

The relative velocity can be defined as

$$\mathbf{v}_u^s = \mathbf{v}_s - \mathbf{v}_u \quad (1.13)$$

which is simply the difference between the two velocities (first derivative of the position).

Due to the drift of the receiver clock, the received signal is not exactly centered around the nominal frequency and is instead obtained as $f_u^s = f_c(1 + \dot{b}_u)$. Then, (1.11) can be rearranged to obtain

$$c \frac{f_u^s - f_c}{f_c} + \mathbf{v}_s \cdot \mathbf{e}_u^s = \mathbf{v}_u \cdot \mathbf{e}_u^s - \frac{c f_u^s \dot{b}_u}{f_c}. \quad (1.14)$$

The satellite velocity \mathbf{v}_s can be retrieved from the ephemeris, so the right side of (1.14) contains the derivatives of the receiver position and clock drift which are unknown. Given the Doppler shift measurements obtained w.r.t. at least 4 satellites, then a system can be solved to obtain the receiver velocity.

1.5.3 Quality of the PVT Solution

The quality of the resulting solution depends on the number of available pseudorange measurements and on the geometry of the satellites in space. To achieve the most precise solution as possible, it would be preferable to have as many visible satellites as possible, distributed in space in order to be in different directions from the user (i.e. not in the same portion of the sky from the perspective the receiver). This comes from the fact that when performing a PVT estimation, a system of

equations has to be solved. If the equations were all linearly independent, then the best solution would be possible. But, since satellites in line-of-sight can only be above the line of the horizon, this condition of linear independence is not satisfied. Therefore, especially in urban scenarios in which large portions of the sky are hidden to the receiver, possibly due to the presence of buildings or other obstacles, the PVT computation is still possible, as long as four satellites are visible, but the quality of the solution degrades.

From (1.10), the elements of the error vector $\Delta\rho$ can be modelled as random variables with variance σ_{URE}^2 (User Equivalent Range Error). On the other hand, considering the position error $\Delta\mathbf{p}$, its covariance can be computed as

$$\text{cov}(\Delta\mathbf{p}) = (\mathbf{H}^T\mathbf{H})^{-1} \sigma_{URE}^2. \quad (1.15)$$

By defining

$$\mathbf{G} = (\mathbf{H}^T\mathbf{H})^{-1} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{bmatrix} \quad (1.16)$$

then the Geometrical Dilution Of Precision (GDOP) factor can be derived as

$$\text{GDOP} = \sqrt{g_{11} + g_{22} + g_{33} + g_{44}} \quad (1.17)$$

and the standard deviation of the positioning error is therefore obtained as

$$\sigma_p = \text{GDOP} \sigma_{URE}. \quad (1.18)$$

It should be noticed from (1.6) that matrix \mathbf{H} is obtained from the coefficients in (1.4), and thus only depends on the geometry of the problem. Therefore, in (1.18) the GDOP works as an amplification factor for the uncertainty on the user measurements that depends on the position of the satellites. This means that the integration of other auxiliary measurements, possibly coming ground anchor nodes, would compensate for the lack of visibility, by providing ranging information that is geometrically relevant.

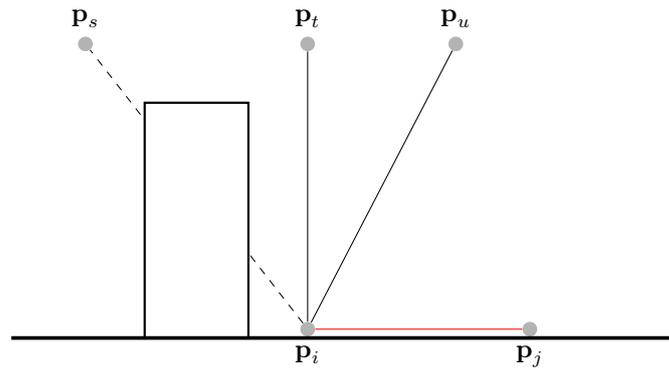


Figure 1.9: Example of limited visibility due to obstacles.

As can be seen from Figure 1.9, the presence of obstacles in urban environments can decrease the number of line-of-sight signals. The integration of auxiliary ranging measurements (red line), not only can compensate for the lack of visible satellites, but also improves the geometry of the system to be solved.

Equally important to the quality of a PVT solution, is the precision of the pseudorange and Doppler measurements obtained from each satellite as modelled by σ_{URE}^2 . Several sources of error affect the pseudorange measurements, like propagation delay errors in the ionosphere or troposphere, multipath error introduced by the environment surrounding the receiving device, or errors in the parameters of the satellite trajectory [1].

Chapter 2

Cooperative DGNSS

The positioning solution obtained through GNSSs is used by an increasing number of Intelligent Transportation System (ITS). However, these vehicular applications require very strict safety-related constraints, which are not always met by stand-alone GNSS [11]. Especially in some harsh environments, like urban canyons, the quality of the positioning solution degrades quickly due to limited visibility, multipath and possibly both intentional and unintentional interference as well.

To overcome this limitation, different enhancements of GNSS have been developed in order to improve the positioning solutions [12]. Many of these applications are focused on the use of sensors such as 3D laser scanners (also known as LIDAR), radar sensors or camera systems [13]. The high precision information provided by these sensors allows for vehicles to gain a comprehensive knowledge of their surroundings, including possible obstacles, and thus allowing them to navigate through it. Since the use of these sensors leads to a drastic increase in both costs and computational complexity, auxiliary methods based on the cooperation between different networked receivers, exchanging GNSS-only measurements, have been proposed [14] [15] [16].

In particular, Cooperative GNSS exploits the ability of multiple receiving devices (also called agents) to communicate with each other, so that pieces of information such as GNSS measurements can be exchanged among them. This interaction between agents can be done through civil networks infrastructure such as cellular, or private/public Wi-Fi networks. In case these are not available, direct communications between receivers is also a possibility. In either case, the transmission of cooperative messages has to be regulated by proper communication protocols that guarantee low latency (e.g. USP, RTSP), which are not discussed in this work.

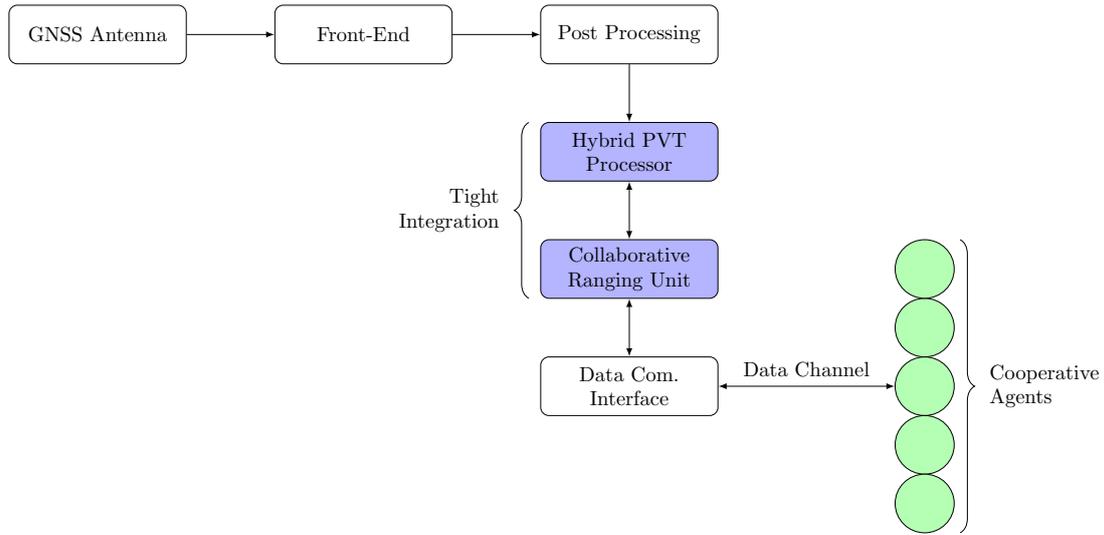


Figure 2.1: Simplified block scheme of a cooperative DGNS receiver.

Figure 2.1 shows a simplified scheme of a cooperative receiver. The Data Communication Interface handles the exchange of GNSS-only measurements among agents through a data channel. This information is then exploited by the Collaborative Ranging Unit (CRU) to compute inter-agent distances. The Hybrid PVT Processor is then responsible of computing a PVT solution based on both the GNSS measurements coming from the respective antenna and front-end, and the cooperative measurements computed by the CRU. The work presented in this thesis focuses on the integration of the additional cooperative measurements in the PVT computation in order to improve the quality of the positioning solution. In the context of cooperative positioning, vehicles are able to broadcast messages containing information regarding their most recent PVT solution, as well as measured pseudoranges and Doppler shifts, to other vehicles. At the receiver, this information can be used by the CRU to compute inter-agent distances (also called baseline lengths). The idea is to use the position of other agents as reference anchors (in addition to the satellites) for relative ranging. The goal of these methods is to exploit the additional information transmitted by aiding agents to improve both accuracy and precision of the navigation solution, assuming reliable information is transmitted among the cooperating receivers [17].

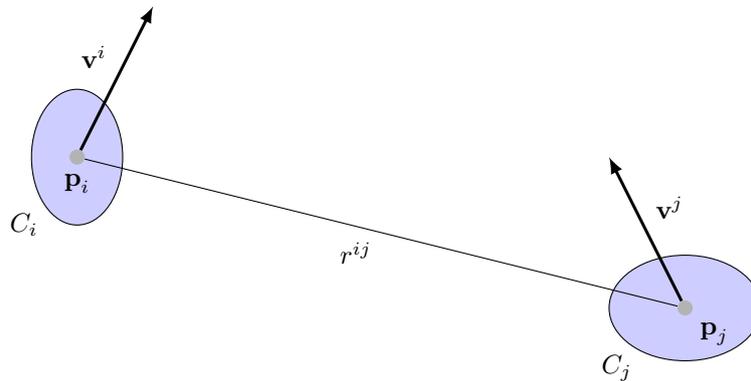


Figure 2.2: Simplified cooperative scenario with two agents.

Figure 2.2 shows a two-dimensional simplified cooperative scenario where two dynamic agents with velocity \mathbf{v} are cooperating. The positioning solution of both of them is affected by an error with covariance matrix C . Such covariance matrix represented as an ellipse describes the uncertainty on the positioning solution. Since in navigation applications it is important to measure not only the position, but its uncertainty as well, the estimation of the covariance matrix C also has to be performed. The aim of the aided agent, lets say j , is to estimate the true Euclidean distance between the position of the two agents

$$r^{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|. \quad (2.1)$$

A possible strategy would be to have the two agents simply exchange their computed position (i.e. Absolute Positions Distance), but since the position of agent j has yet to be estimated, (2.1) cannot be used to estimate the inter-agent distance directly. Instead, this quantity can be estimated thanks to physical ranging (using the sensors described in Section 2) or through the exchange of position-related data such as GNSS measurements. This additional ranging information can be used by agent j to improve its own positioning solution.

2.1 Differential Methods

The estimation of these inter-agent distances is performed through methods similar to those used in Differential GNSS, with the difference that in the cooperative case both devices are moving, and base and aided station are determined according to the quality of their a-priori information regarding state vector and its covariance matrix. In order to perform this estimation, either raw code or carrier measurements can be used, although the latter is not preferred since it is

prone to cycle slipping phenomena and requires the carrier phase ambiguity to be resolved [14]. Different techniques can be used to exploit these measurements to obtain baseline length estimates, namely: Inter-Agent Ranging, Raw Pseudorange Ranging, Single Differences and Double Differences (DD). Previous studies have shown that DD technique can provide the most accurate estimations, under the conditions of small uncorrelated error term contribution on the range measurements, as well as small multipath error [14].

2.1.1 Synchronization of Pseudoranges

Differential methods aim at estimating the distance between two GNSS receivers based on the difference between their measurements.

In principle, since the receivers all work independently on their PVT solution, the estimation of the pseudoranges is not performed at the same time instant, as shown in Figure 2.3.

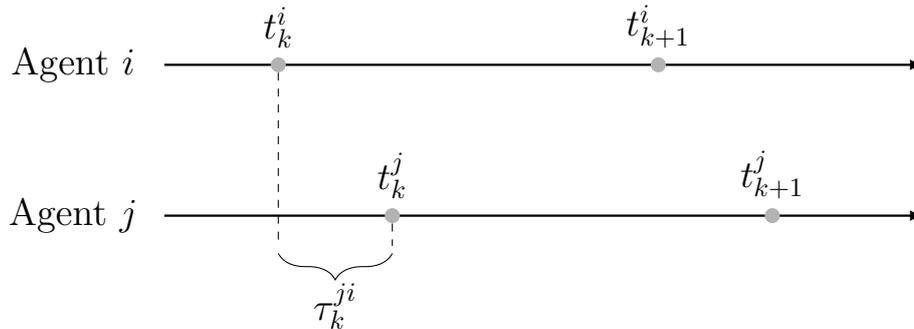


Figure 2.3: Time scale of two different cooperative agents.

The variable t_k^i refers to time instant at which agent i has collected its measurements and performed its k -th PVT solution. The computational delays introduced by these operation are therefore not considered in the definition. The inter-epoch misalignment between two agents is defined as

$$\tau_k^{ji} = t_k^j - t_k^i. \quad (2.2)$$

Since each measurement is time tagged to the GNSS time, the inter-epoch misalignment is the difference in time between the time tag of two different receivers. This means that each cooperative agent computed its distance w.r.t. to a given satellites at a different time instant, and thus when the satellite is in a different position. Even if the time misalignment of the two time tags is small, the resulting pseudorange measurement can be greatly different because satellites move extremely fast. Since the computation of the baseline length is based on the difference in

distance w.r.t. a common reference point, the estimation is coherent only if time-consistent measurements are combined. Therefore, the pseudoranges from two different agents need to be synchronized in time.

It is important to note that the asynchronous nature of the different PVT solutions is unrelated to the clock bias of the respective receiver, but is simply due to the fact that different devices evaluate independent solutions.

The aided agent j , which receives pseudoranges from an aiding agent i , can adjust them according to

$$\rho_i^s(t_k^j) = \rho_i^s(t_k^i) + \tau_k^{ij} \lambda_0 \Delta f_i^s(t_k^i) \quad (2.3)$$

where Δf_i^s is the Doppler shift between satellite s and agent i , and λ_0 is the wavelength corresponding to the carrier frequency of the transmitted signal.

The idea behind (2.3) is to correct the pseudorange measurement w.r.t. satellite s obtained by agent i , as if it was measured at the same time as agent j did. This correction is performed by exploiting the Doppler shift measurement, which implicitly contains the information on how fast the satellite and the agent were moving away or towards each other, thus allowing to predict the pseudorange at that time instant through a direct measurement performed previously.

It should be added that this correction is a linearization of the motion of both satellite and agent, and thus is only accurate if τ_k^{ji} is small enough.

2.1.2 Double Difference of Pseudoranges

Once the pseudoranges computed by different agents are synchronized, as described in Section 2.1.1, they can be used to compute the baseline length as will be showed hereafter. Considering a pseudorange measured by agent i from satellite s , it can be expressed similarly to (1.2) as

$$\rho_i^s = r_i^s + b_i + x^s + \epsilon_i^s \quad (2.4)$$

where r_i^s is the user-satellite true range, b_i is the error due to the receiver clock bias, x^s is the noise related to satellite s , while ϵ_i^s is noise term related to both satellite and agent. By subtracting the pseudoranges obtained from two agents w.r.t. to the same satellite a Single Difference is obtained as

$$S_{ij}^s = \rho_i^s - \rho_j^s = \Delta r_{ij}^s + (b_i - b_j) + (\epsilon_i^s - \epsilon_j^s). \quad (2.5)$$

Clearly, $\Delta r_{ij}^s = r_i^s - r_j^s$ is the difference in true range. From (2.5) it can be seen that the noise term related to the satellite x^s has been cancelled, as it is in common for both the agents involved in the computation of the SD.

Since the distance between the satellite and the agents is much larger than the one between the two agents, the vectors pointing from the two agents toward the

satellite can be considered parallel. For the same reason, the error in the estimation of the position can be ignored when computing vector \mathbf{e}^s , as it is negligible w.r.t to the satellite-agent distance. As a result, \mathbf{e}^s can be obtained thanks to the GNSS corrections and the satellite ephemeris [11]. In such case, the difference in true range can also be represented as

$$\Delta r_{ij}^s = \mathbf{e}^s \cdot \mathbf{d}^{ij} \quad (2.6)$$

where \mathbf{e}^s is the aforementioned vector pointing from the agents to the satellite, and \mathbf{d}^{ij} is the baseline vector, pointing instead from agent i to agent j , as depicted in Figure 2.4.

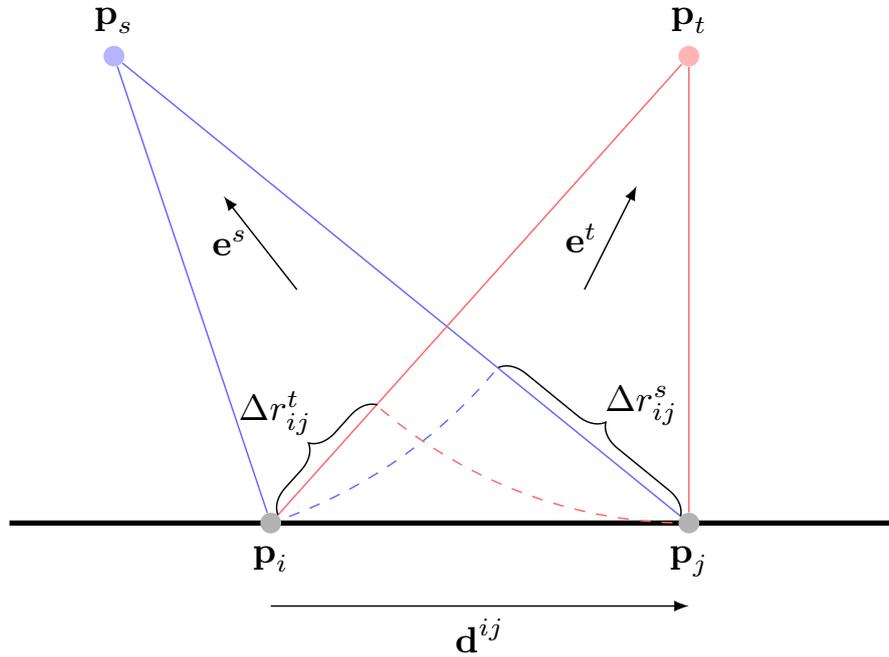


Figure 2.4: Double Difference of pseudoranges.

After computing another SD w.r.t. a second common satellite t , then a Double Difference can be obtained as

$$D_{ij}^{st} = S_{ij}^s - S_{ij}^t = (\mathbf{e}^s - \mathbf{e}^t) \cdot \mathbf{d}^{ij} + [(\epsilon_i^s - \epsilon_j^s) - (\epsilon_i^t - \epsilon_j^t)]. \quad (2.7)$$

When computing the DD, also the clock bias of both agents i and j can be removed, since the term is in common to both the SD being subtracted from each other. Furthermore, all the other common errors between the two agents are also removed. Both ionosphere and troposphere error cancellations hold effectively when agents are in close proximity, since the signal received by both agents has crossed

roughly the same section of atmosphere.

Previous results also show that considerable improvement is obtained from integrating auxiliary measurements when the multipath conditions are very similar for the two cooperating agents. For example, in case two receivers are in proximity, and they both received a signal reflected from the same obstacle, the bias introduced by the excess path from the reflection is cancelled, since it is common for both of them. Furthermore, when computing a ranging measurement w.r.t. to a reference point that is on ground level, the satellite geometry improves (the aiding agent essentially acts as an additional ground anchor node), potentially leading to an improvement in the quality of the PVT solution due to a reduction of the GDOP, as explained in Section 1.5.3.

Going back to the DD calculation, given a set of common satellites visible to both the agents, and taking one of those as a reference satellite (lets say s), (2.7) can be expressed in matrix notation as

$$\mathbf{D}_{ij} = \mathbf{H} \mathbf{d}^{ij} + \boldsymbol{\epsilon} \quad (2.8)$$

where \mathbf{D}_{ij} is the column vector of the DD w.r.t. the reference satellite and each of the other common satellites with indexes $\{1, \dots, n\}$. Similarly, the other column vectors are defined as

$$\begin{aligned} \mathbf{H} &= [(\mathbf{e}^1 - \mathbf{e}^s), \dots, (\mathbf{e}^n - \mathbf{e}^s)]^T \\ \boldsymbol{\epsilon} &= [((\epsilon_i^1 - \epsilon_j^1) - (\epsilon_i^s - \epsilon_j^s)), \dots, ((\epsilon_i^n - \epsilon_j^n) - (\epsilon_i^s - \epsilon_j^s))]^T \end{aligned} \quad (2.9)$$

under the assumption that the non-common noise terms ϵ are zero mean and with equal variance, than the baseline vector \mathbf{d}^{ij} can be obtained through linear LMS estimator according to

$$\mathbf{d}^{ij} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{D}_{ij}. \quad (2.10)$$

Eventually, the baseline length can be obtained as the norm of the baseline vector

$$d^{ij} = \|\mathbf{d}^{ij}\|. \quad (2.11)$$

It is important to highlight an crucial difference in the way different ranging measurements are obtained. GNSS pseudoranges are computed as a product between the estimated time of flight of the signal and the speed of light. On the other hand, the baseline lengths are computed in (2.11) as the Euclidean norm of a multivariate vector. Regardless of the differential technique used to obtain the baseline vector estimate, when its magnitude is small, the resulting distribution of the error on the baseline length is skewed towards positive values, since negative distances can not be obtained.

2.1.3 Weighted Least Square Double Difference

From (2.8), it can be seen that the only remaining error term after the DD computation is ϵ . In non-urban scenarios in which the multipath effect is small, the term ϵ is mostly due to code acquisition error, a quantity that is strictly related to the CNR of the received signal from each satellite.

A possible strategy to reduce the impact of ϵ would be to compute the DD using only pseudoranges from satellites with high CNR values, above a given threshold [18]. By selecting only pseudorange measurements performed from a satellite with strong received signal would reduce the errors, but the improvement would be counteracted by the reduction in the number of measurements used in the computation. As an alternative, instead of only using measurements from high CNR sources, measurements can be weighted based on the strength of the received signal. The WLS-DD method, proposed in [11], is a linear unbiased estimator that weights measurements based on the knowledge of their covariance matrix. The baseline vector is obtained as

$$\mathbf{d}^{ij} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{D}_{ij}. \quad (2.12)$$

It is assumed that the error terms ϵ are independent and zero-mean, but differently from (2.10) of non-equal variance, since the signal strength from different satellite is in general different. In (2.12), \mathbf{W} is the weight matrix obtained as the inverse of the covariance matrix of ϵ .

Exactly as the DD method detailed in Section 2.1.2, a satellite (once again let it be called with index s) with the highest possible CNR value for both the two agents i and j has to be chosen as a reference satellite. If this is ensured, then the value $(\epsilon_i^s - \epsilon_j^s)$ is small enough and the following simplification can be done

$$\begin{aligned} \epsilon^t &= (\epsilon_i^t - \epsilon_j^t) - (\epsilon_i^s - \epsilon_j^s) \\ \epsilon^t &\approx (\epsilon_i^t - \epsilon_j^t). \end{aligned} \quad (2.13)$$

Since the ϵ terms are assumed to be uncorrelated, than the weight matrix can be simplified to a diagonal matrix

$$\mathbf{W} = \begin{bmatrix} \frac{1}{(\sigma_0)^2} & 0 & \dots & 0 \\ 0 & \frac{1}{(\sigma_1)^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{(\sigma_n)^2} \end{bmatrix} \quad (2.14)$$

where the set of other satellite used for the computation is $\{0, 1, \dots, n\}$. Unfortunately, the standard deviation σ_a of a generic noise error term cannot be

measured directly, but it can be assumed to be inversely proportional to the CNR

$$(\sigma_a)^2 \propto \left(\frac{1}{(S_i^a)^2} + \frac{1}{(S_j^a)^2} \right) \quad (2.15)$$

where S_i^a is the CNR of the signal received by agent i from satellite a . Equation (2.15) can be substituted into (2.14) to obtain the weight matrix \mathbf{W} .

In order to avoid using pseudorange measurement with high noise, a minimum threshold of CNR is set, and only satellite with CNR values above the threshold are used in the computation. Since (2.12) is used to estimate a baseline vector which has three components, one for each spatial dimension, the signal from at least other three common satellites (plus the reference one) need to have a CNR value higher than the threshold. If this condition is not met, or if no satellite has a CNR value high enough to be used as the reference satellite, an estimation of the baseline vector is still possible, but its quality is not guaranteed.

Chapter 3

Bayesian Estimation for Hybrid PVT

In the previous Chapter, it was described how the performance of stand-alone GNSS was not sufficient for applications with strict safety requirements, and cooperative methods to improve the performance of the positioning solutions were introduced. In a scenario where additional information (i.e. inter-agent distances) is exploited, the receiver needs to be able to integrate the auxiliary measurement with the other GNSS observables. In the context of GNSS, the state to be estimated is

$$\begin{aligned}\boldsymbol{\theta} &= [x \ y \ z \ b \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{b}] \\ \mathbf{p} &= [x \ y \ z] \\ \mathbf{v} &= [\dot{x} \ \dot{y} \ \dot{z}]\end{aligned}\tag{3.1}$$

where \mathbf{p} and \mathbf{v} are respectively the vectors of position and velocity of a receiver in a given reference frame. Instead, b and \dot{b} are the bias and drift of the receiver clock w.r.t. to the GNSS time scale. The aim is to estimate the state vectors starting from a vector \mathbf{z} containing the observed measurements, including the cooperative ones. In particular, Bayesian Filters exploit the a-priori knowledge regarding the temporal evolution of the system.

3.1 Kalman Filter

The KF is an algorithm that, from a series of measurements containing noise, estimates unknown variables thanks to a joint probability distribution. Given a discrete time representation of a system, a state transition equation can

be expressed according to

$$\boldsymbol{\theta}_k = f_{k-1}(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}). \quad (3.2)$$

The state at given epoch k is a function, possibly non-linear, of the previous state and of \mathbf{v}_{k-1} , a multivariate random variable representing the noise affecting the states. The measurements can instead be expressed as

$$\mathbf{z}_k = h_k(\boldsymbol{\theta}_k, \mathbf{w}_k). \quad (3.3)$$

Once again, h_k is a possibly non linear function, and \mathbf{w} is another multivariate random variable representing the measurement noise. It should be added that vector \mathbf{z}_k containing the GNSS measurements can be extended to the cooperative case by simply appending a vector \mathbf{d}_k with the auxiliary measurements as

$$\bar{\mathbf{z}}_k = [\mathbf{z}_k \quad \mathbf{d}_k] \quad (3.4)$$

The Kalman Filter provides the optimal unbiased estimation of the state only if the following assumptions hold:

- Both \mathbf{v}_{k-1} and \mathbf{w}_k are realizations of a multivariate Gaussian random variable with known parameters
- f_{k-1} is a known linear function of $\boldsymbol{\theta}_k$ and \mathbf{v}_{k-1}
- h_k is a known linear function of $\boldsymbol{\theta}_k$ and \mathbf{w}_k

In the context of GNSS, these assumptions are not met, since trilateration is not linear by definition, and the motion of the receiver is often non-linear as well. Most importantly, the error affecting the cooperative ranging measurements is, in general, not Gaussian distributed.

To deal with such conditions, sub-optimal filters have been adopted. In particular, the Extended Kalman Filter is widely used in positioning applications. It works by linearizing the system by means of Taylor expansion and then solving through linear KF. On the other hand, the Unscented KF is instead able to approximate posterior probability as Gaussian-distributed.

3.1.1 Extended Kalman Filter Routine

The algorithm starts by predicting the state at epoch k based on the a posteriori state estimate $\boldsymbol{\theta}_{k-1|k-1}$ at the previous epoch, and on the system inputs \mathbf{u}_k as

$$\hat{\boldsymbol{\theta}}_{k|k-1} = \mathbf{F}_k \hat{\boldsymbol{\theta}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \quad (3.5)$$

where B_k is the input matrix, while the linearized state transition matrix is derived from (3.1) as

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \boldsymbol{\theta}} \right|_{\hat{\boldsymbol{\theta}}_{k-1|k-1}, \mathbf{u}_k}. \quad (3.6)$$

The system is therefore linearized around the current estimate. It should be added that the EKF can suffer from severe divergence of the estimation if the system is not modeled properly.

The covariance matrix is also predicted as

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^t + \mathbf{Q}_k \quad (3.7)$$

in which \mathbf{Q}_k is the covariance matrix of the multivariate random variable \mathbf{v}_k . Afterwards, the update procedure starts by computing the innovation vector according to

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\boldsymbol{\theta}}_{k|k-1}) \quad (3.8)$$

this quantity is obtained as the difference between the current measurement and the predicted state. Subsequently, the covariance prediction obtained in (3.7) is used to compute the Kalman Gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (3.9)$$

where \mathbf{R}_k is instead the covariance matrix of the noise \mathbf{w}_k . Finally, the a posteriori estimate of both the state vector and the covariance matrix are computed as

$$\hat{\boldsymbol{\theta}}_{k|k} = \hat{\boldsymbol{\theta}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k. \quad (3.10)$$

Because of how the innovation vector is defined, the second term in (3.10) acts as a correction term of the predicted state according to the input measurements. The covariance matrix is updated as well

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (3.11)$$

these correspond to the final estimates at epoch k , and will be subsequently used to perform the prediction procedure in the next iteration $k + 1$.

The EKF is therefore able to deal with non-linear systems, but still assumes that the process and observation noises are Gaussian distributed, which is not guaranteed to be the case in the case of cooperative inter-agent distance measurements, as already mentioned in Section 3.1.

3.2 Particle Filter

The Particle Filter is a Monte-Carlo approximation of an optimal sequential Bayesian estimation algorithm. The important feature of the PF is that it relaxes the constraints of the KF, in particular it is able to natively handle both non-linear system and non-Gaussian probability densities (or PDF). As such, in the context of cooperative positioning applications based on inter-agent distances, it can provide a better performance w.r.t. the EKF. The drawback of this method is that the estimation requires a higher computational effort, especially when the cardinality of the state vector to be estimated increases.

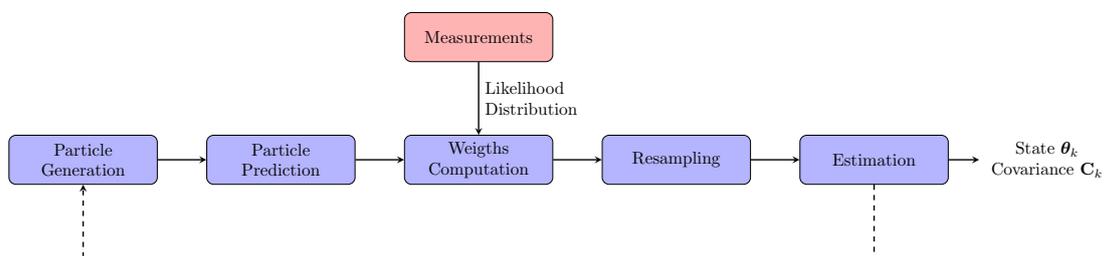


Figure 3.1: Block diagram of Particle Filter routine.

As shown in Figure 3.1, the algorithm starts by generating a cloud of particles around the current estimate, which are then propagated forward similarly to what is performed in (3.5). Each particle represents a possible state estimate, and is assigned a weight (probability) based on the value of the input measurements and a likelihood function. The final state estimate is then obtained as the weighted average of all the particles, and used to initialize the generation of particles at the next epoch.

3.2.1 Particle Filter Algorithm

At first, a set of N particles is initialized according to the a priori density

$$\hat{\boldsymbol{\theta}}_k^i \sim p(\boldsymbol{\theta}_k | \hat{\boldsymbol{\theta}}_{k-1}^i) \quad (3.12)$$

where $\boldsymbol{\theta}_k^i$ is the state vector of particle i at time k . The number of generated particles needed to guarantee the best performance of the PF filter increases with the number of dimensions of the state to be estimated. The number of particles has also a big influence on the computational time needed by the PF to obtain a solution.

Next, in the prediction step, each particle generated previously is propagated

forward according to the system model as it is done in the EKF routine. Afterwards, the weights computation is performed based on a predefined likelihood function $p(\mathbf{z}_k | \hat{\boldsymbol{\theta}}_k^i)$ as

$$w_k^i = \frac{\mathcal{L}(\bar{\mathbf{z}}_k | \hat{\boldsymbol{\theta}}_k^i)}{\sum_{i=1}^N \mathcal{L}(\bar{\mathbf{z}}_k | \hat{\boldsymbol{\theta}}_k^i)} = \frac{\prod_n p(z_{n,k} - z_{n,k}^i)}{\sum_{i=1}^N \prod_n p(z_{n,k} - z_{n,k}^i)}. \quad (3.13)$$

Function p refers to the PDF of the measurement error, and the likelihood is actually then obtained as the product of the PDFs for all the measurements. What is from now on called the likelihood distribution, only refers to the PDF of a single measurement. The likelihood function should be chosen so that it describes the way the error is expected to be distributed for the measurements. The PF is able to assign different likelihood functions separately for each input measurement.

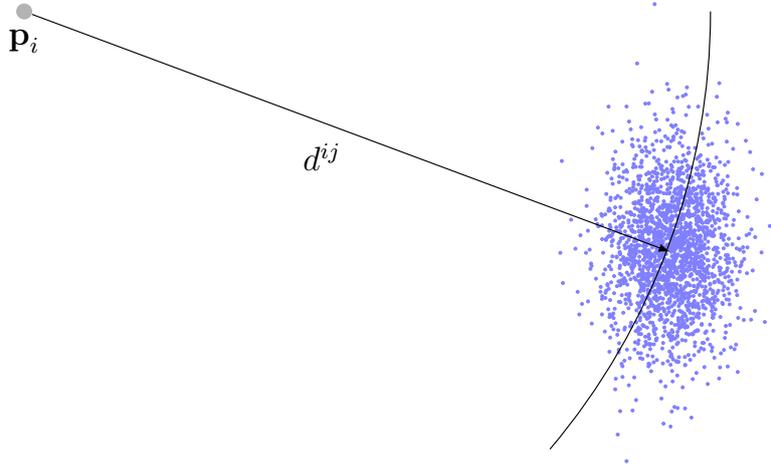


Figure 3.2: Projected particles are assigned weights based on the input measurements.

As shown in Figure 3.2, a weight is associated to each generated particle based on its distance w.r.t. the anchor point, which in case of cooperative measurements is the position of the aiding agent \mathbf{p}_i . If for example a Gaussian distribution is used as likelihood, then particles whose distance from the anchor point is the same as the estimated baseline length d^{ij} will be assigned the highest weights. Clearly, other particles are assigned decreasingly smaller weights according to the shape of the likelihood distribution. Each particle is assigned a weight for each of the input measurements, and their final weight is given by the product of all assigned weights.

Afterwards, the resampling step is performed, where particles are re-combined. This step is fundamental since it has to deal with two problems of the PF: degeneracy problem and sampling impoverishment.

The degeneracy problem occurs after multiple iterations of the PF algorithm, when the weight of a single particle progressively converges to 1, while all other particles have negligible weight. When this happens, first the PF collapses to a single point thus losing its peculiarity. Second, most of the computational effort of the algorithm is used to predict and assign weights to particles which have little to no contribution to the final solution. It has been shown that, if no re-sampling procedure is performed, this problem cannot be avoided.

Therefore, the purpose of the re-sampling step is to generate a new set of particles, starting from the ones already existing, where particles with higher weights are more likely to be drawn into this new set.

This method solves the degeneracy problem, but introduces sampling impoverishment. Since particles with high weight are more likely to be drawn into the new set, they are statistically chosen multiple times, thus leading to a decreased diversity in the particles of the re-sampled set. This issue can be solved by performing the re-sampling step only when the degeneracy effect has become significant. The level of degeneracy can be represented by the effective number of particles, given as

$$N_e = \frac{1}{\sum_i (w_k^i)^2}. \quad (3.14)$$

This value is compared with a threshold (e.g. $N_t = \frac{2N}{5}$), and the re-sampling step is performed if $N_e \leq N_t$ (degeneracy higher than the threshold).

In the end, the Bayesian estimation is given as the weighted average of the re-sampled particles as

$$\hat{\boldsymbol{\theta}}_k = \sum_{i=1}^N w_k^i \boldsymbol{\theta}_k^i \quad (3.15)$$

Chapter 4

Error Modelling for Cooperative GNSS

In this Chapter, the distribution of the error of the baseline length is analyzed for two different available datasets, involving both static and kinematic agents. In particular, when a dynamic agent is involved in the cooperation, the error on the estimated baseline length shows a bias whose value evolves with time.

The behaviour of the baseline length error is observed for different parts of the trajectories of the kinematic agents, and some metrics are introduced in an attempt to model the bias.

Since, it is observed that the distribution of the baseline length error is not stationary, the aim is to gain a better understanding of how the motion of the cooperating agents affects this distribution. With that knowledge, the PF algorithm can be modified to be able to adapt the likelihood used to weight particles based on such defined metrics.

At first, bias prediction model based on both absolute and relative motion between the agents is discussed. Then, based on the knowledge acquired during the modelling process about the link between the bias and the dynamics of the agents, a more in-depth analysis of the receiver architecture handling the cooperative measurements is performed, leading to the discovery of the origin of the observed bias.

Finally, two possible strategies to correct this bias are proposed and their performance in terms of accuracy of the navigation solution is evaluated and compared to the previous implementation of the receiver architecture.

4.1 Simulation Scenario

The scenario of cooperative vehicles is implemented through an Agent Network (AN). A vehicular aided agent T (also referred as Target agent) moves on a Bernoullian lemniscate trajectory while exchanging GNSS measurements with some static aiding agents, whose position is exactly on the track as can be seen in Figure 4.1. Agent T is also static for roughly the first 30 seconds of the simulation, and then uniformly accelerates until full speed is reached.

In this scenario, at each epoch of the simulation (measurements rate of 10 Hz), agents interact with each other by providing an estimate of their state vector, along with the set of GNSS measurements (i.e. Doppler and pseudoranges). Pseudorange measurements are synchronized and inter-agent distances are then computed by means of WLS-DD method as explained in Section 2.1.3.

In addition to the stand-alone GNSS measurements, the cooperative ranging measurements are integrated in the PVT solution using the PF algorithm in order to improve the estimate of the position. All simulation results presented in this chapter are obtained working with a simulated constellation of 8 visible satellites and the cooperation of only one aiding agent at a time. As such, this can be considered as a worst-case scenario for cooperation, as only one inter-agent distance measurement is available. Extending the study to multi-agent cooperation is therefore expected to provide further improvement to the quality of the positioning solution.

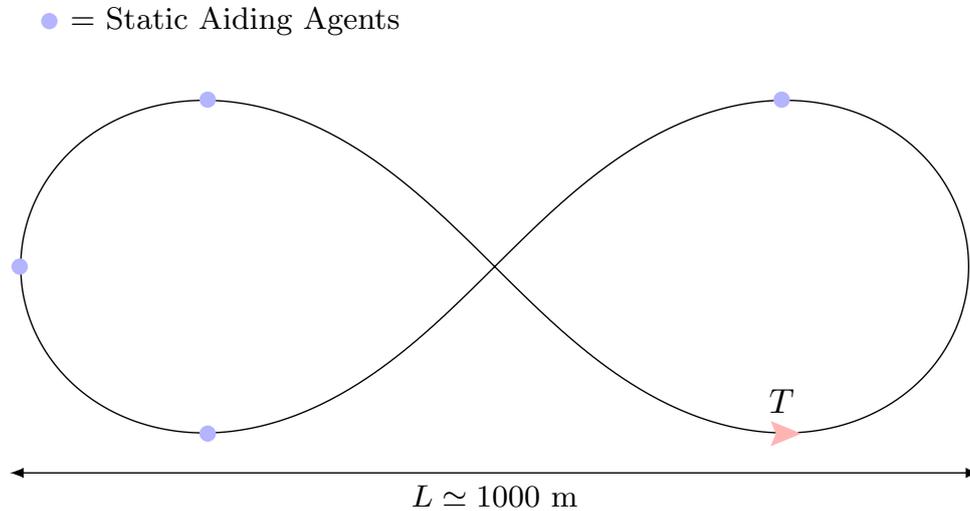


Figure 4.1: Position of the cooperative Agents on the track.

4.1.1 Ranging error for Static and Kinematic agent

At first, the ranging error obtained when two static agents exchange their GNSS measurements is considered. This error is the result of the difference between the output of the WLS-DD and the true distance between the agents. Since both the trajectory of the moving agent and the static position of the aiding agents are simulated, their true position is known at each time instant.

A qualitative evaluation of the behaviour that is shown in Figure 4.2, suggests that both mean and variance of the distribution of the ranging error are constant over time.

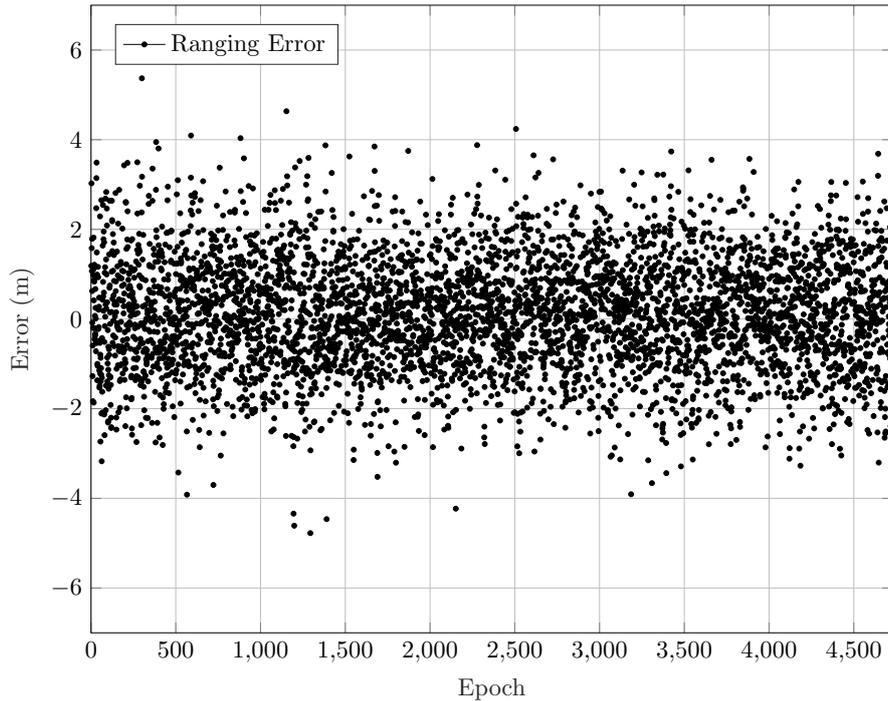


Figure 4.2: Ranging error over time when both the agents are static.

Considering the ranging error computed when one of the agents is the moving target, as shown in Figure 4.3, the distribution of the baseline error is not zero-mean anymore, as it presents a bias that evolves with time.

The difference w.r.t. the previously considered static case suggests that the bias might be introduced by the motion of one of the agents involved in the computation of the DD. This notion is further supported by the fact that the bias, excluding the linear increment over time, which will be discussed later, is periodic with the same period as the motion of the agent along the trajectory.

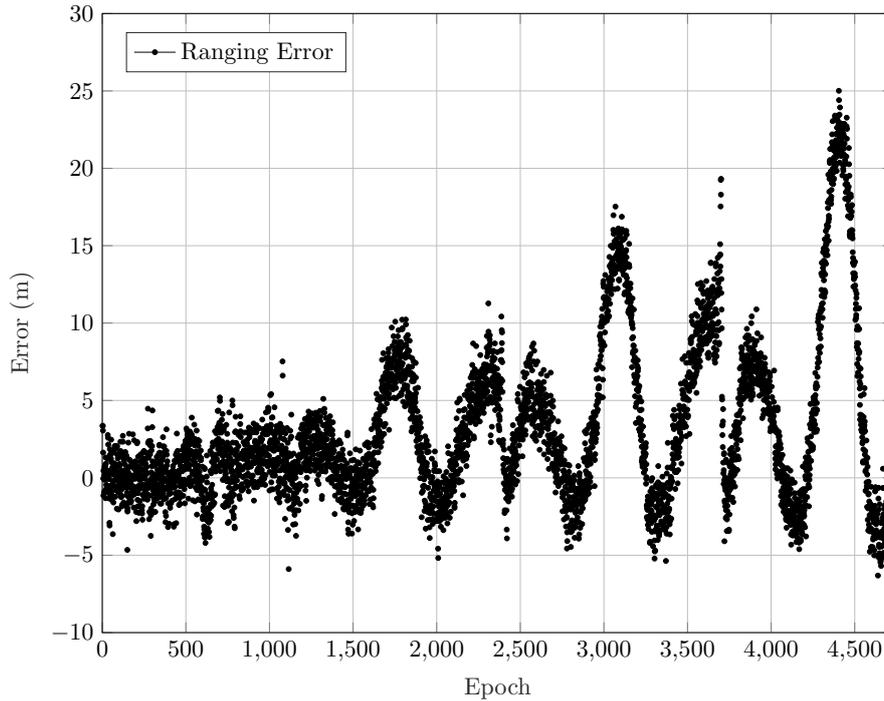


Figure 4.3: Ranging error over time when one moving and one static agent are involved.

4.1.2 Dependence on Motion

To address a possible correlation between the motion of the agent and the experienced bias, a first quantity is investigated as

$$|\alpha_k^{ij}| = \|\mathbf{v}_{k-1}^i \times \mathbf{v}_k^i\|. \quad (4.1)$$

Since the definition of the metric is derived from the observation of the behaviour of the bias over time, it is only tentatively defined, as well as all other metrics that will be introduced in this chapter.

In (4.1), α_{ij_k} is defined as the *ranging error bias* between agents i and j , at simulation epoch k . Hence, the bias at each iteration is defined as the norm of the cross product between two consecutive velocity vectors. As previously mentioned, since only one agent is in motion, only its velocity is considered in the computation of (4.1). Furthermore, it is reminded that the cross product of two vectors is defined as

$$\|a \times b\| = \|a\| \|b\| |\sin \theta| \quad (4.2)$$

where θ is the angle between vectors a and b in the plane that contains them.

Therefore, there are only two cases in which the norm of the cross product of two vectors is equal to zero: when the norm of either vector is zero, or when θ is zero. Going back to the quantity defined in (4.1), this value is zero when two consecutive velocity vectors are parallel (i.e. when the trajectory is straight), or when the speed (the norm of the velocity vector) is zero, which is when the agent is static. This last case is consistent with what was observed when both agents were static, and no bias was indeed observed. As shown in Figure 4.4, a first comparison between the model and the actual errors shows that, while the zero-points of the metric correspond to moments of zero bias experienced in the simulation (except for a slight delay which will be addressed later), the general behavior of the bias is not fully captured just yet.

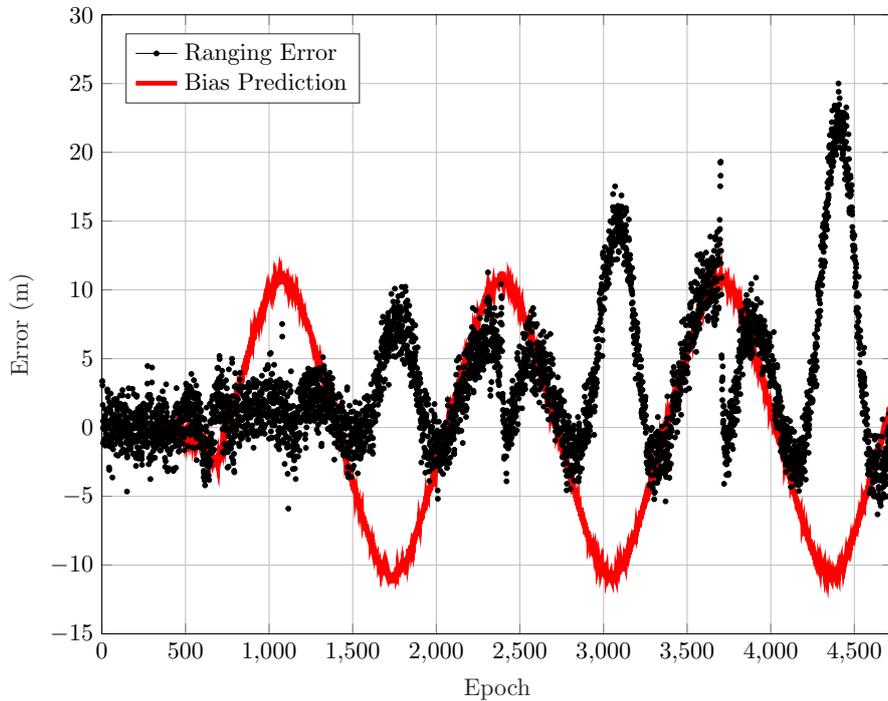


Figure 4.4: Ranging error over time compared with the cross product of consecutive velocity vectors as in (4.1).

4.1.3 Dependence on Relative Motion

A closer look at the bias behavior for the measurements exchanged between the target peer and different cooperative agents, showed slight differences between the cases, thus suggesting a possible dependency also on the relative motion or position, and not only on the trajectory of the moving agent, as considered so far.

For this reason, a second metric for the bias estimation is introduced as

$$|\beta_k^{ij}| = \|\mathbf{d}_{k-1}^{ji} \times \mathbf{d}_k^{ji}\| \quad (4.3)$$

where the vector \mathbf{d}_k^{ij} is the baseline vector, pointing from the static agent j to the moving one i . Given the definition provided in (4.3), once again two possible cases where the resulting bias is zero can be identified: when the norm of the baseline vector is zero (i.e. the baseline length), or when baseline vectors at two consecutive epochs are parallel, thus pointing in the same direction. This last case occurs when the agent is moving directly towards or from the static agent (i.e. when its angular velocity with respect to the static agent is zero). It can be observed in Figure 4.5 that as for the quantity $|\alpha_k^{ij}|$, also for the metric $|\beta_k^{ij}|$, points where the value is zero correspond to points in the trajectory in which the ranging error shows zero bias. Moreover, it is interesting to notice that zero-bias points that were previously not captured by (4.1), are instead modeled by (4.3).

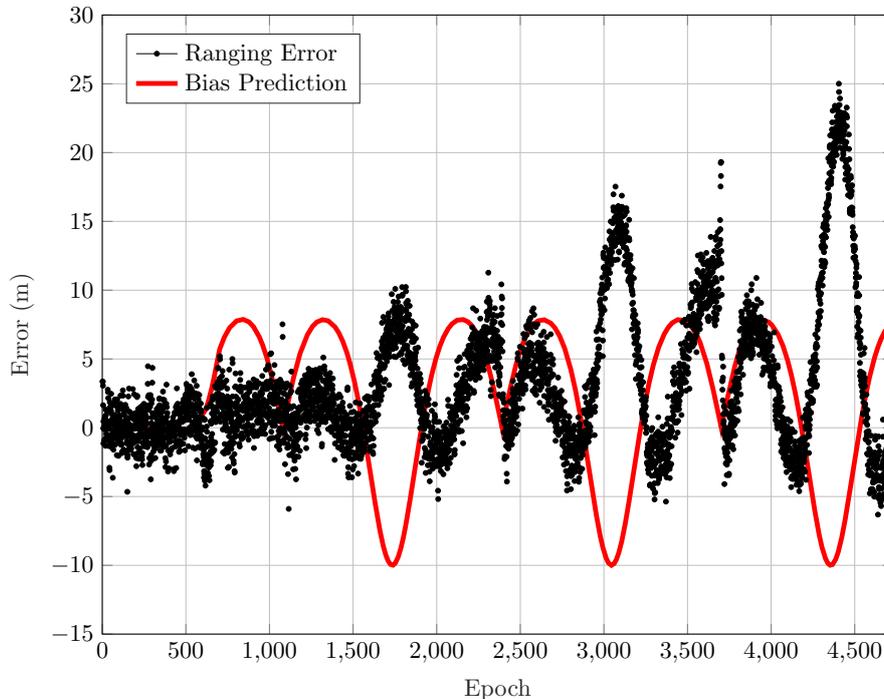


Figure 4.5: Ranging error over time compared with cross product of baseline vectors.

Since both the metrics defined in (4.1) and (4.3) are obtained by computing the norm of a vector, they are always non-negative. On the other hand, as it can be seen in Figure 4.3, the bias observed in the measurements can also be negative

for certain parts of the trajectory. As a consequence, the sign of these functions has to be considered as well. More specifically, it is introduced by taking into consideration the sign of the components of the vector of which the norm is taken. For each of the two metrics defined so far, their sign is computed as the sign of the sum of the components of the cross-product. An equivalent definition could be proposed by taking the sign of the product, but since the functions contain oscillations due to noise, the sum has shown to be a more stable options close to zero-crossing points.

4.1.4 Dependence on Time

As mentioned in 4.1.1, the bias also showed a linear dependency w.r.t. the simulation epoch k . By merging this behavior with the previously defined quantities, its absolute value is therefore computed as

$$|\phi_k^{ij}| = |\alpha_k^{ij}| |\beta_k^{ij}| a k \quad (4.4)$$

where a is a coefficient used to adjust the model to match the slope of the linear increment of the bias.

The now complete model still does not perfectly match the behavior of the bias, which is shifted in time. A careful analysis involving the computation of the cross-correlation between the bias prediction model and the ranging errors, shows that the bias curve is lagging the model by 43 epochs. This is tied to the specific dataset under study, and possibly due to some fault during the processing of the simulated signals during the creation of the dataset. As a consequence, the shift here introduced is not actually part of the model.

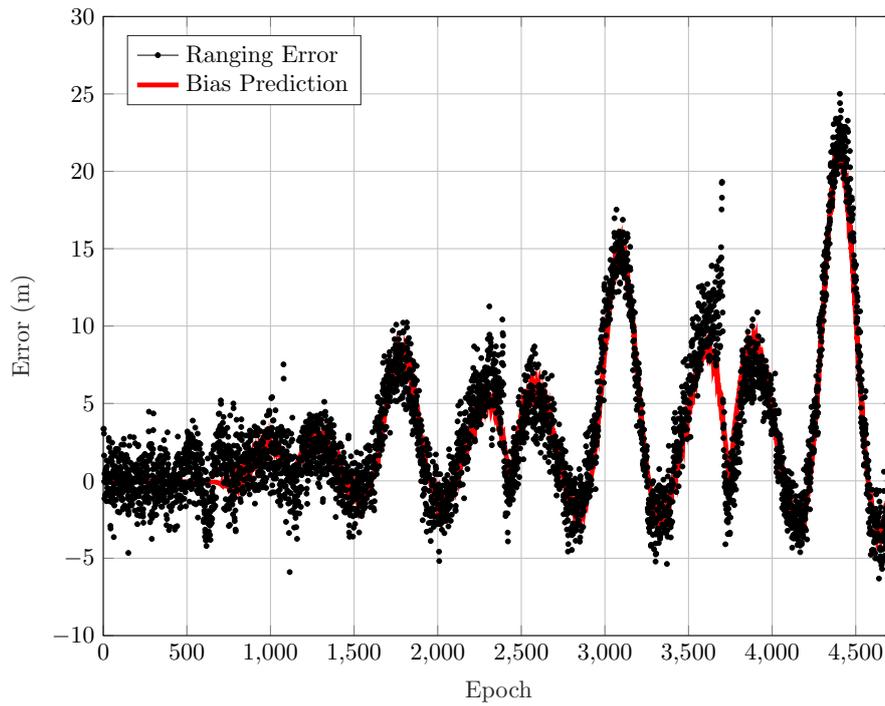


Figure 4.6: Model shifted in time to match the actual behavior.

The origin of this delay in time is not known, but its value is consistent for all the exchanged measurements between all possible pairs of agents. For the sake of an evaluation of the improvement of the positioning solution given by the removal of the bias, the prediction model is shifted in time to properly match the actual behavior, as shown in Figure 4.6.

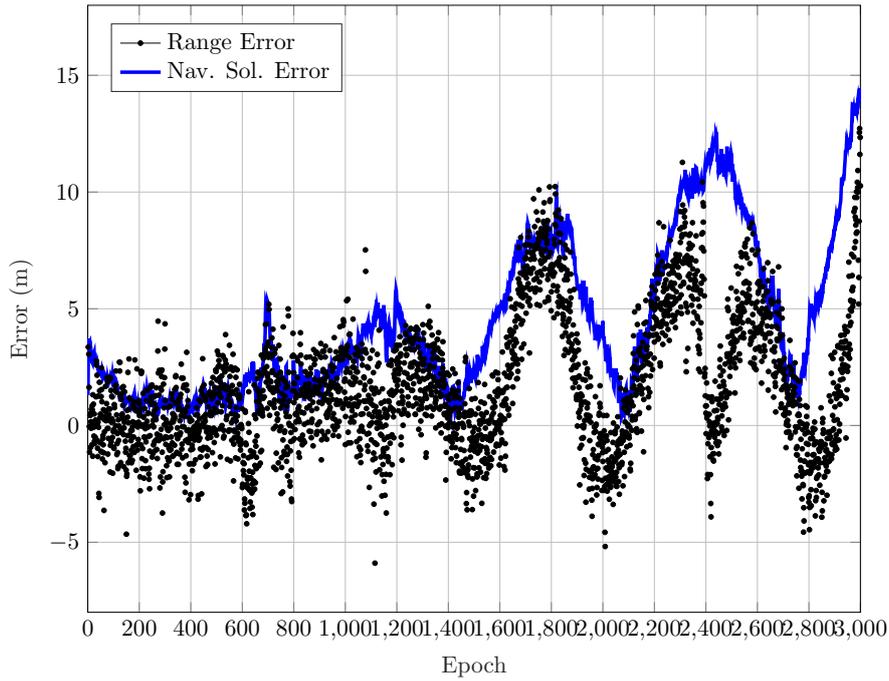


Figure 4.7: Navigation Solution error compared with ranging error over time.

A comparison between the error magnitude of the stand-alone positioning solution (i.e. without cooperation) and the ranging error, shows a similar behaviour and increment over time for both, as can be seen in Figure 4.7. If the positioning solution shows a systematic error, it can be supposed that it is due to either the signal generator, or the receiver. As a result, the bias observed in the distribution of the baseline length error, might not be due to how the cooperative measurements are combined to compute the baseline, but from a systematic error already present in the input measurements of the dataset.

4.1.5 Bias Removal

In order to assess the improvement obtained by the PF when dealing with unbiased measurement, the model previously defined is subtracted from the inter-agent ranging data. The resulting residual error and its distribution are shown in Figure 4.8a and 4.8b respectively. Despite a sub-optimal compensation of the bias, the use of the prediction model guarantees a residual error distribution considerably closer to some well-known probability distributions, hence enabling the measurements likelihood function used in the PF to better match the actual error distribution.

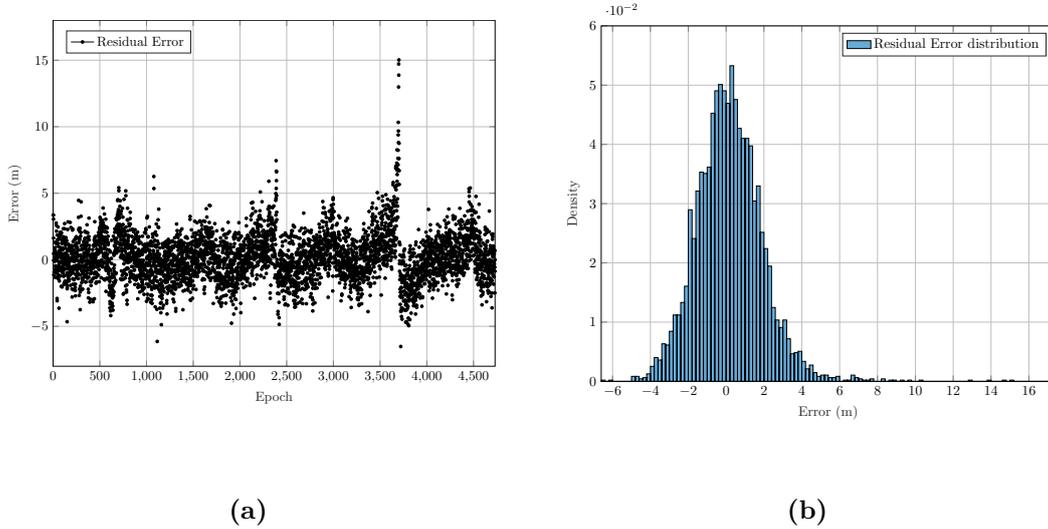


Figure 4.8: Residual ranging error and its distribution.

To quantify the improvement in terms of precision given by the implementation of the a posteriori correction of the bias discussed in the previous sections, the CDF of the navigation error is evaluated for the first lap of the simulation, which corresponds to the first 1900 epoch of the simulation. The value of the bias predicted by the model is subtracted from the ranging measurement at each epoch of the simulation, this is equivalent as having as errors on the measurements the values in Figure 4.8a instead of the ones in Figure 4.3. The values being subtracted from the measurements according to the model in (4.4) are known a-priori thanks to the simulated environment, and from those values their velocity and baseline vectors are computed. A real-time implementation of the model would instead require the aided agent to compute the bias prediction using estimation of these quantities, that are affected by error, thus leading to a degradation of the performance.

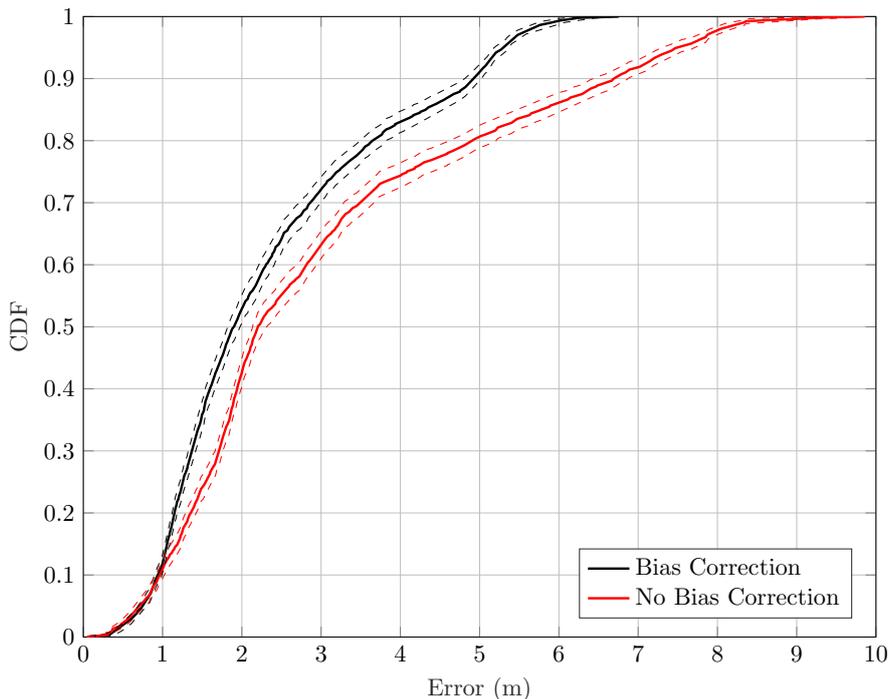


Figure 4.9: Particle filter performance improvement when bias correction is applied to inter-agent distances.

From the CDF of the positioning error shown in Figure 4.9, it is clear to see that, even a heuristic correction of the bias can greatly improve the accuracy of the positioning solution.

	50-th percentile	75-th percentile	95-th percentile
Bias Correction	13.7 %	21.3 %	29.3 %

Table 4.1: Percentage improvement when using a posteriori correction of the bias.

As can be seen from 4.1, the percentage improvement obtained with the correction is greater at the 95-percentile, which consist of the most critical cases. It should be remarked that, as mentioned before, the dataset displayed an increase in time of the bias of the ranging error that is likely due to come corruption of the data and is not expected to be found in a real scenario.

4.2 Circular Dataset

With the intent of gaining a better understanding of the role that the agent's motion might have on the stationarity of the experienced ranging error, a new scenario is studied. In this case, both the target agent T and the collaborative agents C_i are moving. The first follows a clock-wise circular trajectory, while the others are moving as a convoy in counter-clockwise direction on another circular track of slightly larger radius, as depicted in Figure 4.10. It should be added that for this scenario, both the bias increase in time, and the shift in time between the model and the actual behavior of the bias are not present, thus confirming that the phenomena observed in the Bernoullian scenario were likely due to systematic errors in the post-processing of the data.

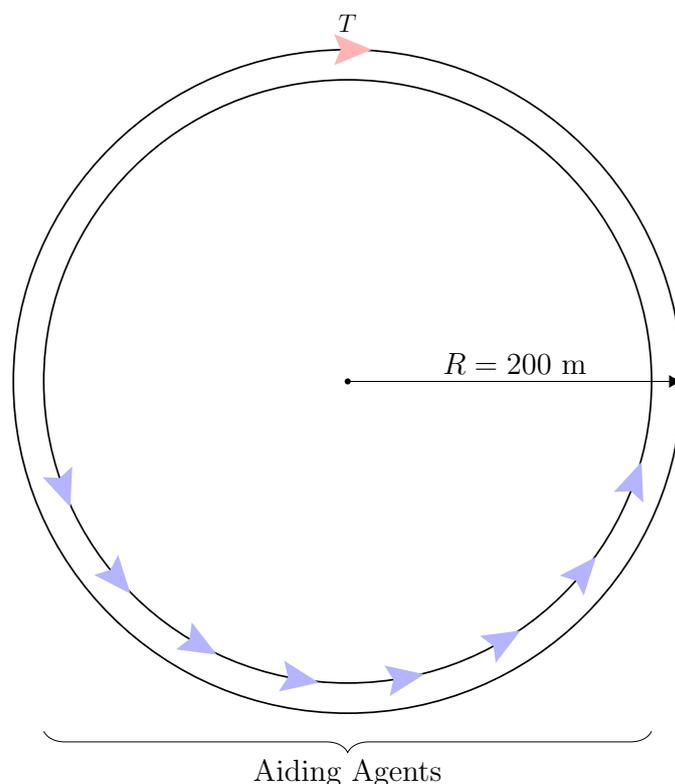


Figure 4.10: Trajectories of the cooperative agents.

4.2.1 Model Adaptation

The model obtained in Section 4.1.4 is applied to the new scenario, excluding for the shift and linear increment in time. As can be seen in Figure 4.11, the model is

still able to capture the periodicity of the trajectory, but is otherwise not consistent with the bias in the error. This is to be expected, since the old model was devised for the case in which only one agent is moving (and hence it only considers its motion). Furthermore, the metrics previously considered might have been misled by the presence of the systematic error in the positioning solution present in the other dataset.

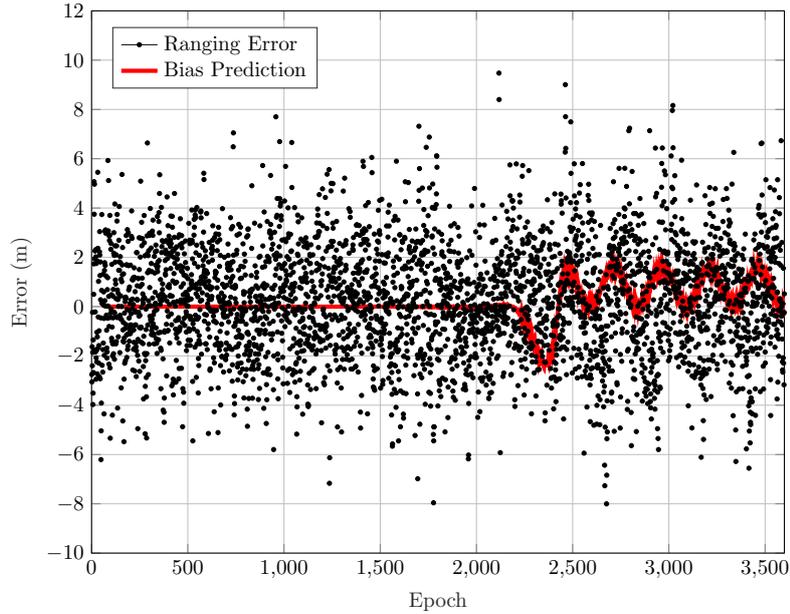


Figure 4.11: Ranging Error over Time compared with model obtained previously.

For these reasons, a new model is studied in order to consider a scenario where both agents are in motion. The newly adopted metric is defined as

$$|\gamma_k^{ij}| = \|\mathbf{v}_k^i \times \mathbf{v}_k^j\| \quad (4.5)$$

where \mathbf{v}_k^i and \mathbf{v}_k^j are the velocities of agents i and j respectively.

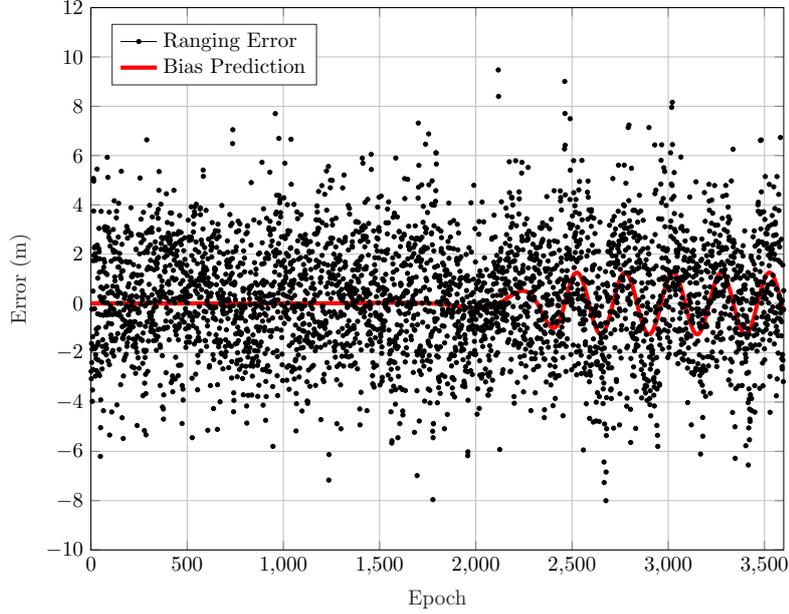


Figure 4.12: Ranging Error over Time compared with modified model.

As it can be seen in Figure 4.12, the metric in (4.5) seems to accurately capture the behaviour of the bias.

4.2.2 Doppler-based Model

Since the *ranging error bias* has been linked to the agent’s velocity, the possibility to express this value in terms of Doppler measurements is explored. A new metric is defined as the difference in measured Doppler between the two agents (with respect to the same satellite), multiplied by its derivative

$$|\delta_k^{ij}| = (\Delta f_i^a - \Delta f_j^a) \frac{\partial(\Delta f_i^a - \Delta f_j^a)}{\partial t}. \quad (4.6)$$

As can be seen in Figure 4.13, from a qualitative point of view, the model based on Doppler measurements correctly captures the behaviour of bias. Despite this, the possibility of defining this model using only one Doppler measurement is allowed by the extreme simplicity of the trajectory, but a general case would need the combination of different Doppler contribution, according to the geometry of the problem, to effectively represent the agent’s motion in 3-D space. On top of that, the noisy nature of Doppler measurements does not provide any advantage w.r.t. using the velocity. It should be remarked here that, although linked, Doppler

shift are measured directly from the incoming signal of each satellite, while the velocity is obtained as an output of the PVT computation.

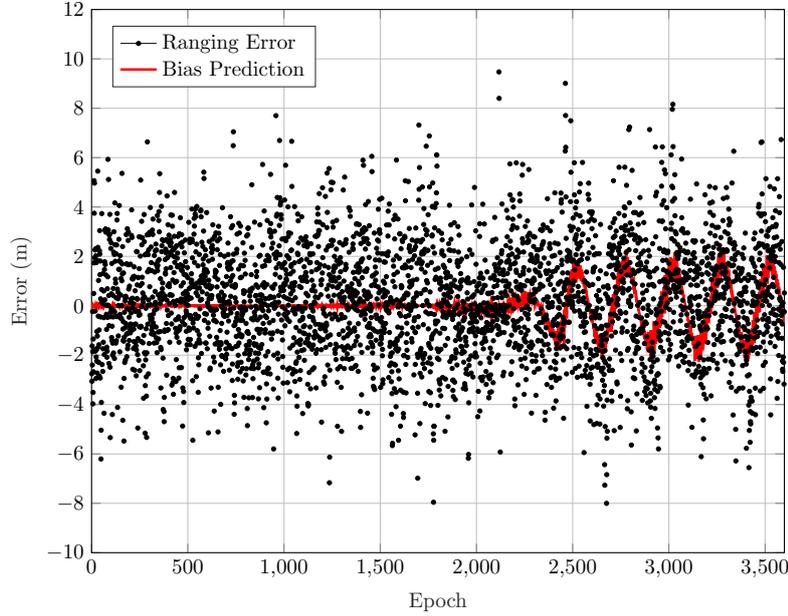


Figure 4.13: Ranging error over Time compared with prediction model based on Doppler measurements.

4.3 Analysis of Receiver Architecture for Bias Correction

The analysis performed so far, with the attempt of modeling the bias affecting the ranging error, has showed a clear link between the motion of the agents and the experienced bias, but since it is still unclear whether this bias is inherent of the nature of the problem or if it is introduced when computing the DD, a more in-depth look at the receiver software is required. In particular, a careful analysis of the Collaborative Ranging Unit (CRU) performed with knowledge acquired so far about the relationship between the motion of the agents and the bias in the ranging measurements, has highlighted an implementation issue that was causing the bias to occur. In its current implementation, when an agent receives the collaborative measurements from another agent, since the two receivers are asynchronous in collecting their observables, these measurements need to be corrected. If agent j performs its pseudorange measurement at time t_k^j , while agent i does so at t_k^i

(which we assume to be before t_k^j), then the pseudoranges of the latter need to be corrected so they are as if they were measured at time t_k^j instead, as explained in Section 2.1.1. This is needed due to the fact that both the satellites and the agent move in the meantime. In order to perform this operation, the Doppler measurement with respect to each satellite are used to correct the respective pseudorange. The difference between this two time instants is the inter-epoch misalignment as computed in (2.2).

Afterwards, the corrected measurements are used to compute the baseline length through DD method, the result of this operation is the distance between the two agents as if both took their measurements at t_k^j

$$d_k^{ij} = \|\mathbf{p}_i(t_k^j) - \mathbf{p}_j(t_k^j)\|. \quad (4.7)$$

Eventually, it is compared with the true range to obtain the ranging error. Differently from the measured baseline, the true range is obtained according to

$$r_k^{ij} = \|\mathbf{p}_i(t_k^i) - \mathbf{p}_j(t_k^j)\| \quad (4.8)$$

which is the Euclidean distance between the positioning solutions, but these points correspond to the position of agent i in time t_k^i and the one of agent j at t_k^j . This definition of true range is thus inconsistent with what is actually being measured. Essentially, the bias comes from the fact that particles are weighted based on the baseline length measurement d^{ij} , but the anchor point used to compute the distance of the particles is the old position of the aiding agent $\mathbf{p}_i(t_k^i)$.

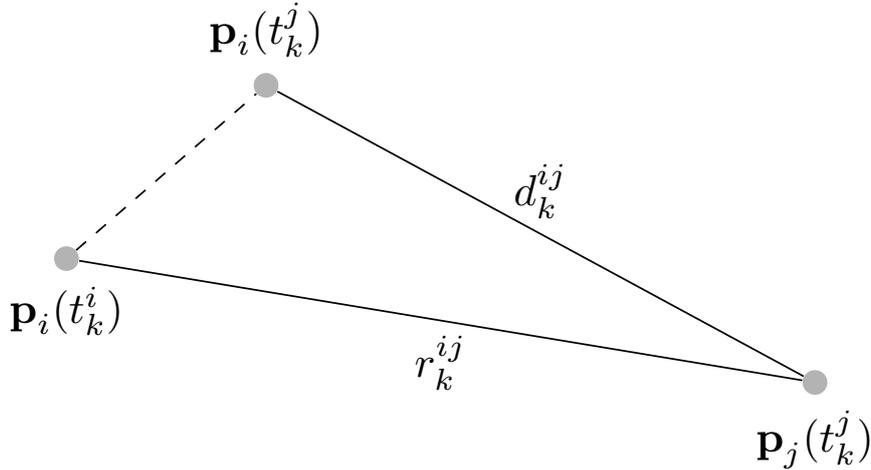


Figure 4.14: The bias on the ranging error is originated by an inconsistency in the measurement of the distance between agents.

Given the scenario drawn in Figure 4.14, a definition of the bias can be derived as

$$\phi_k^{ij} = d_k^{ij} - r_k^{ij}. \quad (4.9)$$

A synchronization of the aiding agent j is performed in the CRU in order to make the definition of the true range consistent with what is actually being measured.

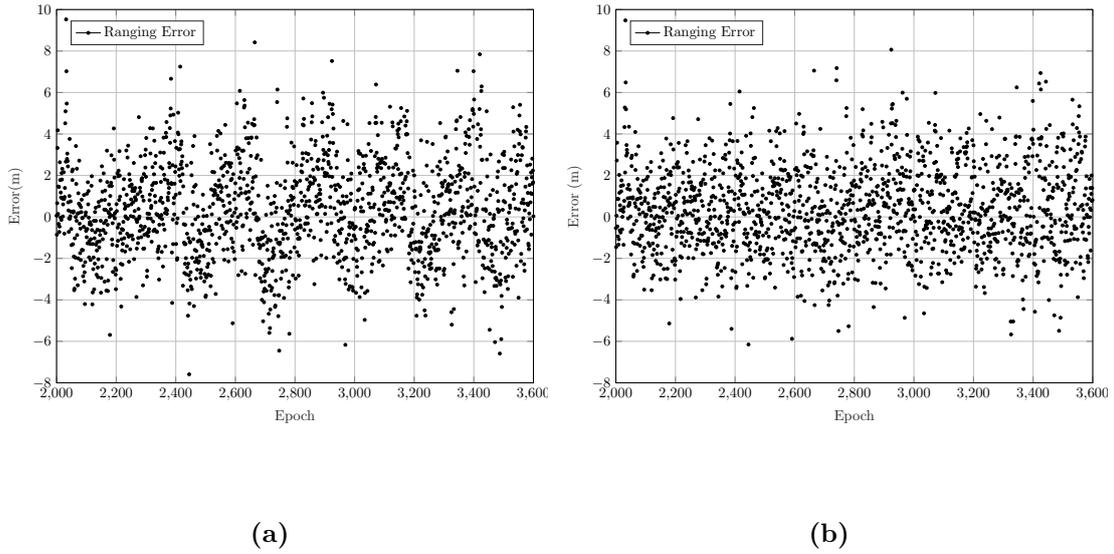


Figure 4.15: Ranging error before (a) and after (b) correcting the true range computation.

As shown in Figure 4.15b, the distribution of the error no longer shows a bias that evolves over time. Therefore, when the position of the aiding agent is correctly synchronized, no bias prediction model needs to be used. This result is crucial, as it shows that a correct estimation and integration of the cooperative measurements does not introduce any bias.

4.4 Aiding Agent Position Synchronization

The discrepancy between the measured baseline length and the anchor point, generates a systematic error which compromises the quality of the solution. Given this inconsistency that is responsible for the *ranging error bias*, its maximum value

at any given epoch is equal to

$$\max \phi_k^{ij} = \mathbf{v}_k^i \tau_k^{ij}. \quad (4.10)$$

It is directly proportional to the velocity of the collaborative agents whose measurements have to be synchronized, and the time misalignment between the two receiver. The actual bias then depends on the direction of the motion as well, and is maximum when the aiding agent is moving directly towards or away from the aided agent, but these two quantities can be used for a quick estimation of the worst-case bias that can be experienced.

Previous work on cooperative positioning focused mostly on scenarios including agents with low dynamics, and since the observed bias is directly proportional to the velocity of the agents, there was no need to synchronize the position of the aiding agents. Furthermore, since previous studies did not focus on precise positioning (unlike what is done here), the effect of the bias from low-dynamic agents was possibly hidden in the high variance of the estimation. To overcome this issue, two possible position synchronization strategies are now discussed.

Velocity-Based Position Synchronization

First, a method based on the velocity of the aiding agent is discussed. When agent j exchanges its measurements obtained at time t_k^j , the velocity obtained from the PVT is also transmitted, along with the position. This additional information can be used to predict the position of the aiding agent at time t_k^i , according to

$$\mathbf{p}_i(t_k^j) = \mathbf{p}_i(t_k^i) + \mathbf{v}_k^i \tau_k^{ij}. \quad (4.11)$$

Since this correction is a linear projection of the motion of the agent, it is not an exact correction of the position, and only holds under the assumptions that τ_k^{ij} is small enough and that the velocity of the agent changes slowly. It also has to be considered that, in scenarios of high dynamics, it is needed greater synchronization between agents for (4.11) to hold. It should be added that the availability of inertial data collected by the aiding agent could enable a better estimation of his trajectory, but would require considerably more data to be exchanged between the cooperating agents.

Pseudorange-Based Position Synchronization

As already mentioned, when aiding agent i transmits its pseudorange measurements obtained at time t_k^i , those are synchronized by aided agent j in order to perform the DD method. The idea is to use these synced pseudoranges as if they were measured at time t_k^j to re-obtain the synchronized position of agent i by means of position estimation algorithms. In particular, the LMS was used in this study to compute the synchronized position. It should be noted that, differently from the

computation of DD which uses raw pseudorange measurements, the LMS requires pseudoranges to be corrected, and thus additional computations on the side of the aided agent. The upside of this method is that, since the two agents are assumed to be close enough, pseudoranges being exchanged can be corrected using the aided agent’s parameters, and therefore no additional information has to be transmitted.

4.4.1 Performance of Position Synchronization methods

The performance of the previously described methods is evaluated with the intent to quantify the improvement in the positioning solution that can be achieved with respect to the previous implementation of the integration of collaborative measurements. Since the issue of synchronizing the aiding agent position is critical especially in high-dynamics situations, when agents are moving at considerable speeds, these methods are compared over a portion of the simulation where the agents are moving. The CDF of the error on the positioning solutions are shown in Figure 4.16.

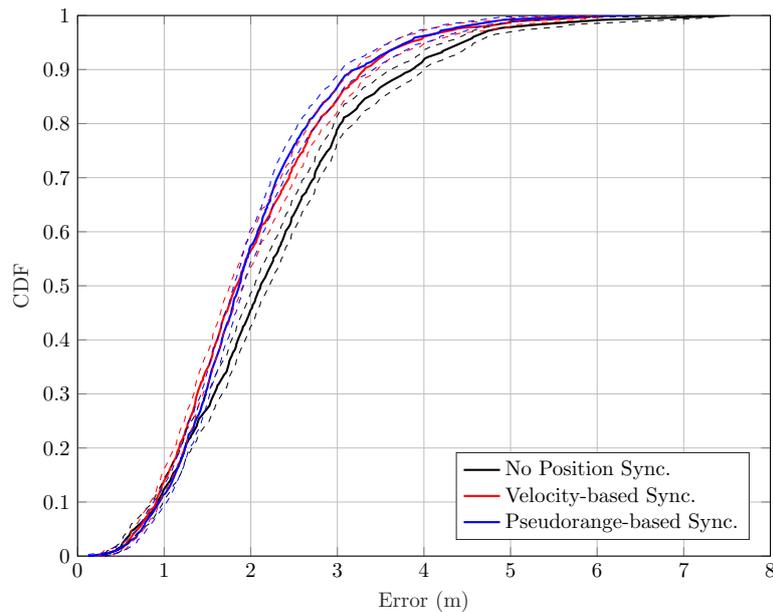


Figure 4.16: Performance comparison of the presented synchronization techniques.

The percentage improvement in terms of CDF of both methods is then evaluated at three different percentile points with respect to the past implementation where the position of the aiding was not synchronized.

Sync. method	50-th percentile	75-th percentile	95-th percentile
Velocity-based	13.1 %	9.8 %	14.2 %
Pseudorange-based	11.7 %	14.4 %	14.5 %

Table 4.2: Percentage improvement when synchronizing the position of the aiding agent.

The results in Table 4.2 show that, even in situations where $\tau_k^{ij} < 0.1\text{ s}$, in a dynamic scenario where agents are moving at relatively high speeds (i.e. approximately 90 km h^{-1}), the improvement in the positioning solution obtained using the synchronization methods previously discussed is considerable.

Chapter 5

Automatic Adaptive Likelihood Switch

In section 3.2.1, it was mentioned how according to the PF algorithm, weights are assigned to the particles based on a pre-defined likelihood distribution. Furthermore, the PF allows for different likelihoods to be used for different input measurements, and also relaxes the Gaussian constraint of the EKF and UKF on the measurement noise. GNSS pseudorange measurements are expected to have a measurement error distributed as Gaussian random variables, and so that distribution is chosen as the likelihood for those measurements.

On the other hand, for inter-agent distances, the proximity between the agents exchanging measurements can affect the shape of the distribution of the error of the baseline length.

To avoid a degradation of the performance due to the mismodeling of errors, it is necessary to dynamically use different likelihood models to account for the non-stationarity of measurements.

Eventually, an algorithm is proposed to automatically switch between different families of likelihood distributions depending on the proximity of agents. Then, its performance is evaluated over a dataset containing kinematic agents.

5.1 Baseline Length Error Distribution

As mentioned in Section 2.1.2, the inter-agent distances being integrated in the PF is obtained as the norm of the baseline vector, whose error on the components can be modelled as a multivariate Normal distribution, since the baseline vector is estimated by means of W-LMS. As such, when the distance between two agents cooperating approaches zero, the measurement error on the baseline length becomes

more and more skewed towards positive values, since negative distances can't obviously be measured. On the other hand, when the distance between agents tends to infinite, the distribution of the error on the distance tends to a more symmetric zero-mean distribution.

The non-Gaussian nature of the cooperative ranging measurement (due to the Euclidean norm) justifies the use of PF as choice for the estimation of the position, as it allows to model the PDF of the errors according to other well-known distributions. To demonstrate this behaviour, a Monte-Carlo simulation was set up to randomly generate points according to multivariate Normal distribution with covariance matrix C around a point at a given distance from the origin. Then, the error of the distance w.r.t. to the origin between the generated points and the mean of the distribution is computed. The experiment is repeated twice for different distances of the center of the distribution. The histogram of the error in the two cases is shown in Figure 5.1a and 5.1b.

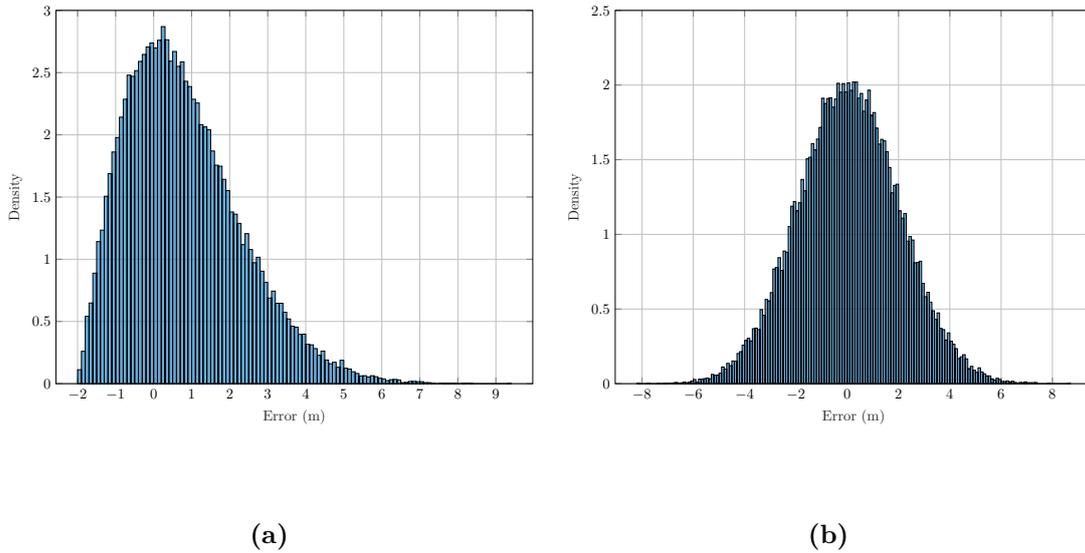


Figure 5.1: Baseline length error distribution at distances 2m (a) and 20m (b) w.r.t. a reference point.

The fact that the distribution of the error of the measurements changes shape as the distance between the agents changes, requires an adaptive estimation of its distribution, in order to guarantee a proper weight of the particles in the PF. Given the dynamic nature of the scenario under study, where the distance between moving agents is constantly changing, it is not a suitable strategy to store the estimation of the ranging errors over time to then decide the distribution based

on a best-fit technique. Instead, the most suitable family of distribution for both the cases in which agents are close or far w.r.t. each other, should be evaluated comparing the performance of some well-known families when used as likelihood distributions for the PF.

It should be added that the error distribution being studied varies continuously as the distance between the agents changes, but the choice of the likelihood distribution can be discretized based on models known a-priori and chosen via a best-fit approach.

5.2 Likelihood Distribution Families

In the previous section, the problem of the skewness of the error distribution for close agents was introduced. In order to find the best performing likelihood for a reliable computation of the particle weights to use in that scenario, some well-known families of distribution will be introduced in this section in order to later test their performance. When evaluating the baseline length w.r.t. to a multivariate distribution, the distribution of the error can be modelled as a Normal distribution only under the assumptions that the value of the baseline length is large enough, and that the covariance matrix of the multivariate distribution is diagonal. In the general and most frequent case in which the off-diagonal terms are instead not negligible, it has been shown through Monte-Carlo simulation of DD measurements, that the GEV distribution can be used to approximate the distribution of the error [19]. In Figure 5.1a, it was shown how the distribution of the baseline length error can be skewed, to account for this scenario, the Rayleigh distribution will be introduced as a possible likelihood.

5.2.1 GEV Distribution

The GEV distribution is a family of continuous probability distribution. It takes as input three parameters known as location μ , scale σ and shape ξ . For the purposes of this investigation, the only value of the location parameter that will be used is zero, since the error distributions observed so far have shown probability distribution centered around zero. The scale parameter σ , used to obtain the likelihood function for the weights assignment in the PF, corresponds to the estimated value of the variance of the input measurements. Therefore, only the shape parameter ξ will be changed to test the performance of this family of distributions in different scenarios.

Moreover, as parameter ξ changes, so does the skewness of the GEV distribution changes. Therefore, this family of distributions will be tested as likelihood for both the cases in which the cooperating agents are close or far, possibly by varying the shape parameter accordingly. The probability density function of the distribution

is

$$f(x; \mu, \sigma, \xi) = \frac{1}{\sigma} t(x)^{\xi+1} e^{-t(x)} \quad (5.1)$$

where

$$t(x; \mu, \sigma, \xi) = \begin{cases} \left(1 + \xi \left(\frac{x-\mu}{\sigma}\right)\right)^{-\frac{1}{\xi}}, & \text{for } \xi \neq 0 \\ e^{-\frac{(x-\mu)}{\sigma}}, & \text{for } \xi = 0. \end{cases} \quad (5.2)$$

Figure 5.2 shows an example of a GEV distribution used as PDF of a single element of the state vector inside the PF, where each particle is given a certain weight based on its distance w.r.t. the aiding agent.

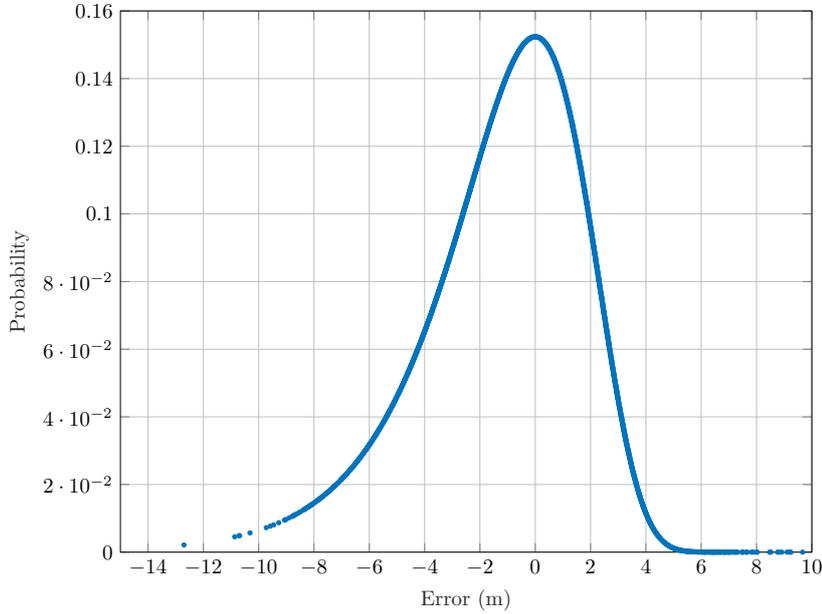


Figure 5.2: GEV likelihood distribution with $\xi = 0$.

5.2.2 Rayleigh Distribution

The Rayleigh distribution is a continuous probability distribution, and is defined for non-negative values only. It can be defined to be the magnitude of a bivariate Normal distribution whose components are uncorrelated and zero mean, and it is modelled through the scale parameter σ . For this distribution, σ is the mode as well, and so both the shape and the mean value of the distribution change according to the value of the scale parameter. For this reason, this distribution will

be tested using fixed values of the σ parameter, instead of using the estimation of the variance of the error as done for the GEV distribution. Because of its support, this family of distribution will only be tested as likelihood when the agents are close to each other. The probability density function is

$$f(x; \sigma) = \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}. \quad (5.3)$$

As shown in Figure 5.3, the distribution is flipped with respect to the vertical axis, this is done in order to reflect the way distances are defined inside the PF. As mentioned in Section 5.1, when agents are close to each other, it is expected that the error on the ranging measurements is not zero-mean and instead skewed towards positive values. Hence, to provide the best performance, the likelihood distribution should give more weight to particles that are closer to the aiding agents with respect to the measurement.

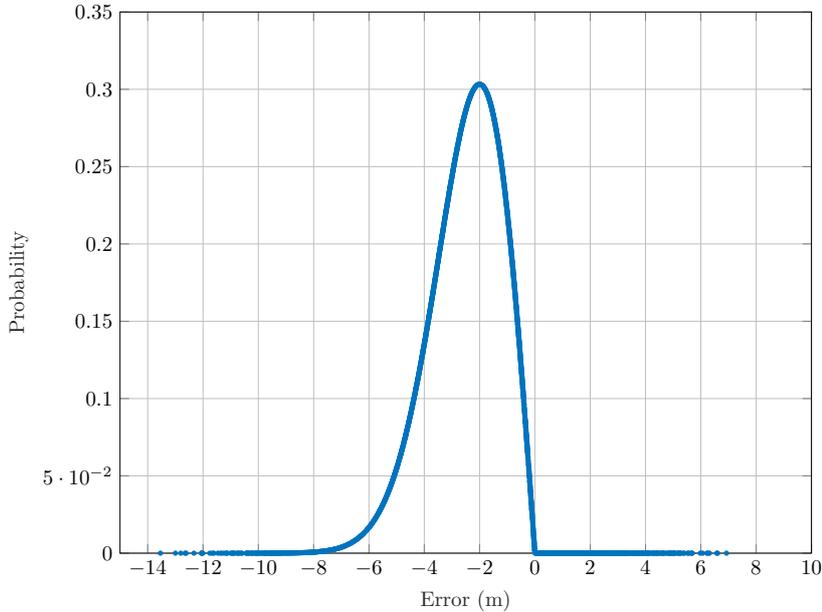


Figure 5.3: Rayleigh likelihood distribution.

5.3 Mahalanobis Distance

In section 5.1, the necessity to switch between different likelihood distribution families according to the distance between the agents was introduced. In order to identify which likelihood is the best suited at each epoch of the simulation, a

metric needs to be introduced in order to evaluate how close the two cooperating agents are.

The reason why the Euclidean distance is not used, is that it does not take into account the variance of the error, whose value is critical in deciding whether the two agents are close enough to perform the switch. Furthermore, the covariance matrix of the error on the baseline vector is not an identity matrix, meaning that the variance might be greater in some given directions.

Because of these reasons, a geometrical information about the distance between the agents is not sufficient to establish when to switch between distribution families. Therefore, a statistical distance has to be defined, namely the Mahalanobis distance.

5.3.1 Mahalanobis Distance Definition

The Mahalanobis distance can be defined as the statistical distance between a point P and a distribution D . In particular, it can be seen as a restriction of the Bhattacharyya distance, which instead computes the distance between two distributions. In case the covariance of D is a unit matrix, the Mahalanobis distance reduces to the Euclidean distance. This metric is widely used in many different applications, and in particular in collision avoidance [20] [21] and motion planning [22] for vehicles. Its value can be computed according to the formula

$$D_M(\mathbf{P}) = \sqrt{(\mathbf{P} - \boldsymbol{\mu})^T C^{-1} (\mathbf{P} - \boldsymbol{\mu})} \quad (5.4)$$

where \mathbf{P} and $\boldsymbol{\mu}$ are the vectors containing the coordinates of point P and the coordinates of the center of distribution D , respectively. It can be observed from (5.4) that because of its definition, the Mahalanobis distance is dimensionless. Moreover, it is also scale-invariant, meaning that if the Euclidean distance between P and the mean of D is doubled, but each term of the covariance matrix of D is also multiplied by two, then the Mahalanobis distance does not change.

When computing the Mahalanobis distance between two cooperative agents, the vectors containing their coordinates would be used as \mathbf{P} and $\boldsymbol{\mu}$. C is instead the covariance matrix of the distribution, and corresponds to the estimated covariance matrix of the baseline vector in the case under study.

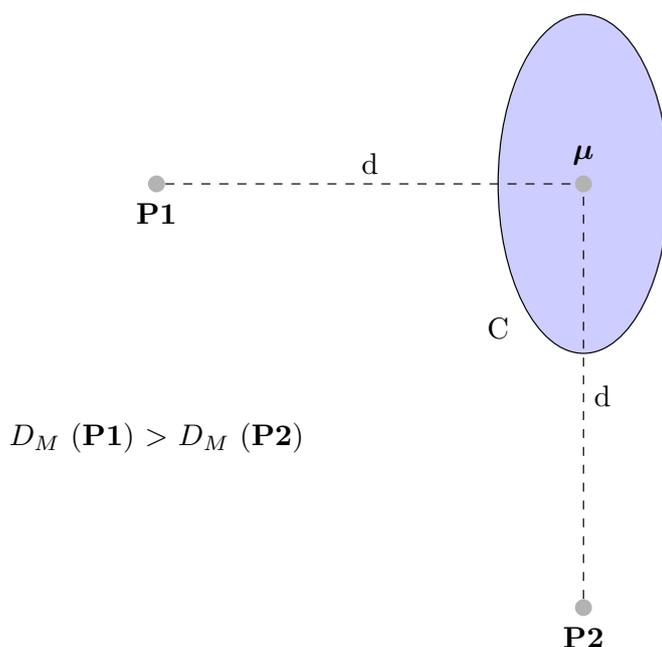


Figure 5.4: Distance of two points with respect to a distribution.

As it can be seen from the plot in Figure 5.4, even though the two points **P1** and **P2** are at the same Euclidean distance d from the mean value of the distribution, **P2** is closer in terms of Mahalanobis distance, since the variance of the distribution is greater in the direction pointing towards it.

Another way of looking at the problem, would be to measure the probability that a given point in space **P** is a realization of the distribution D , but since the Mahalanobis distance can be computed in matrix form, it is easier to be implemented in navigation filters without increasing their computational complexity.

5.4 Automatic Adaptive Likelihood Switch Algorithm

As mentioned in Section 5.1, the shape of the error distribution for the magnitude of the baseline vector depends on its value, thus it is needed to properly choose the likelihood distribution that best fits the expected distribution of the error. In this section, an algorithm to perform an automatic switch between different families of likelihood is proposed. In Algorithm 1, S corresponds to the number of aiding agents, while `DistrType` is a variable used to choose between the different likelihood distribution families.

Algorithm 1 AALS

```

1: if  $S > 0$  then
2:   for  $i = 1 : S$  do  $D_M = f(\mathbf{p}_i(t_k^i), \mathbf{p}_j(t_k^i), C)$ 
3:     if  $D_M > T_M$  then DistrType  $\leftarrow$  1; ▷ GEV
4:     else DistrType  $\leftarrow$  2; ▷ Rayleigh
5:     end if
6:     ProbCOOP( $:, i$ ) = Likelihood(DistrType, ...)
7:   end for
8: end if

```

At each iteration of the PF, for each cooperative agent, the Mahalanobis distance is computed and a choice of distribution type is made based on its value. The likelihood is then computed on line 6 of Algorithm 1 by also providing the function any parameter necessary for the chosen distribution. It should be added that, in case a real-time estimation of the covariance matrix of the baseline vector is not performed, but instead only an estimate of the variance of the baseline length is available, the algorithm has to be modified to work with the baseline length instead of computing the Mahalanobis distance, since it is not possible to compute the covariance matrix from only an estimate of the variance. In such case, the value should be compared to a different threshold $T'_M = T_M * \sigma$, where σ is the estimated variance of the baseline length. This alternative is a simplification of the real scenario, as it does not fully consider the orientation of the covariance matrix. In particular, the information of the orientation of the covariance matrix with respect to the baseline vector is not taken in consideration, and only the variance of the baseline length is used instead. For this reason, it should be preferred the implementation given in Algorithm 1.

5.4.1 AALS Threshold Value

In order to benefit the most from the likelihood switch, it is of critical importance a correct choice of the threshold value T_M . For this reason, a Monte-Carlo simulation is set up to find a candidate best fit distribution. A grid of points with different values of variance and distance of a distribution from a reference point is set up. The best fit distribution for each point in the grid is decided based on the Bayesian Inference Criterion (BIC), a popular tool used to compare the PDF of a given distribution to a set of well known models [23] [24].

The BIC uses the optimal likelihood function value, while also penalizing more complex models with multiple parameters. This approach is suitable for real implementations, since it limits the complexity of the likelihoods that have to be

generated inside the PF algorithm. The BIC can be computed as

$$\beta_i = -2\Lambda_i + W_i \log(N) \quad (5.5)$$

where Λ_i is the Log-Likelihood of a given model and W is the number of parameters of its PDF.

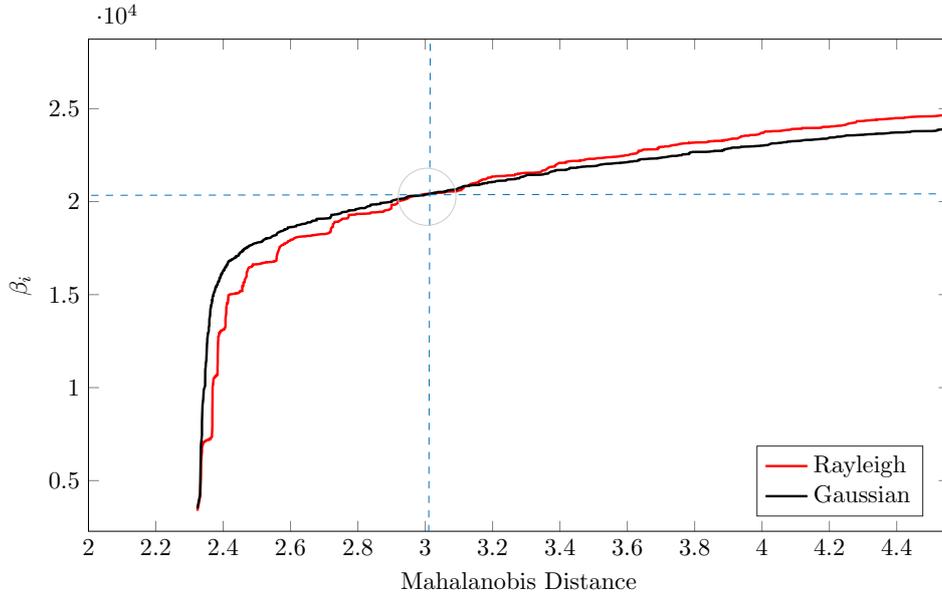


Figure 5.5: BIC computed at different Mahalanobis distances for two distribution families.

As can be seen in Figure 5.5, the Rayleigh distribution minimizes the BIC for values of the Mahalanobis distance roughly smaller than 3, meaning that it is the PDF with the best fit for those cases. It should be added that the results shown are obtained from an ideal scenario.

In reality, the covariance matrix of the baseline vector is not diagonal, and the off-diagonal terms are in general not negligible, although no assumption can be made on the correlation of the errors, since it depends on the geometry of the satellites. Furthermore, no assumptions can be made on the correlation between errors on the components of the baseline vectors, as in general it depends on the geometry of the problem. Because of these reasons, the distributions obtained here are not necessarily the best fit in a real scenario, but the simulation provides a good estimate of the Mahalanobis distance in which the switch happens. Therefore, a threshold value for the automatic likelihood switch is chosen as $T_m = 3$. It should be clarified that this choice of value is only an heuristic. An accurate estimation of

the threshold value would require an accurate estimation of the covariance matrix, and on the magnitude of its off-diagonal terms. A generalization of this is out of the scope of this thesis and is foreseen as a future investigation.

As for the likelihood distributions to be used in the a real implementation, simulations are run over different families of distributions, possibly with varying parameters as well, in order to evaluate their performance.

5.4.2 AALS Implementation

Considering the block diagram of the PF shown in Figure 3.1, Algorithm 1 is implemented as a further block between the input measurements and the weight assignment. Based on the position of the cooperating agents, the algorithm computes the Mahalanobis distance between them and uses its value to choose between two different likelihood distributions.

A quick simulation was set up to check whether the effectiveness of the proposed AALS algorithm, and so to evaluate if it switches between the two distribution types as intended. During the simulation, the measured ranges between peers are collected separately, based on the scenario that was chosen according to the AALS algorithm, meaning that a set of true baseline lengths is collected for all the epoch where peers were considered to be close enough to perform a switch of the likelihood function, and another set of measurements is collected for the complementary case.

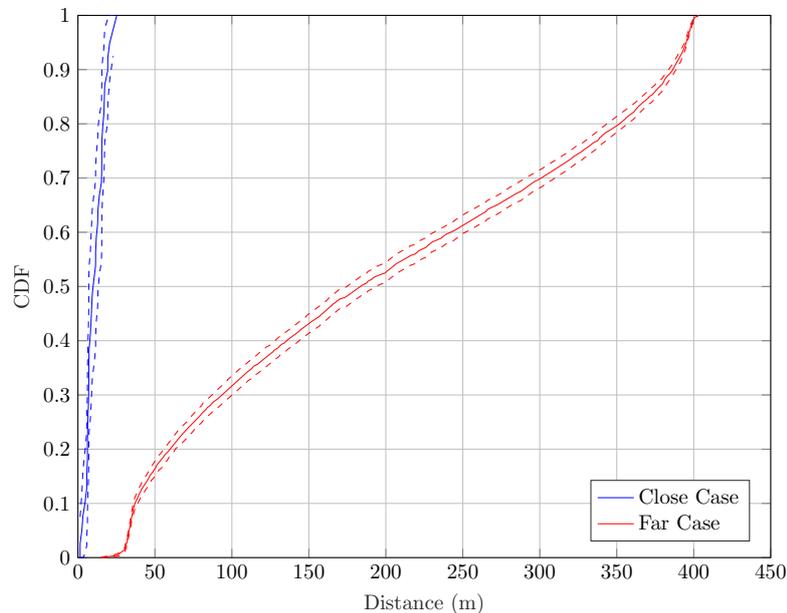


Figure 5.6: CDF of the range measurements collected for the two cases.

In Figure 5.6, it is shown the CDF of the collected values of true baseline length for the two cases described. As it can be seen, the algorithm works as intended, as it switches to a different distribution only for epochs were peers are close to each others, and for those cases only. Since the computation of the Mahalanobis distance, used to perform the switch, is based on values of distance and estimated covariance matrix, whose value might vary over the duration of the simulation, the two curves might overlap slightly, as there is no fixed value of the distance at which the switch is performed. It can be nonetheless claimed that the implementation of the AALS algorithm previously introduced in Section 5.4 works as intended.

5.5 AALS Performance Evaluation

In this section, the performance of different distribution families previously introduced will be evaluated both when they are used as distribution types 1 and 2, as defined in Algorithm 1. First, the best-performing likelihood will be evaluated for the case in which the cooperating agents are far from each other. Then, that distribution will be kept fixed and the same type of analysis will be repeated for the complementary case.

5.5.1 Distribution Families for Distant Peers

Since it is of interest to consider the far case first, the second distribution type, used as likelihood when the cooperating agents are close, is kept fixed for all tests performed as a GEV distribution with shape parameter $\xi = 0.5$. Then, the performance of the likelihood switch is evaluated when the first distribution type is a GEV with varying shape parameter $\xi = 0, 0.5, 1$.

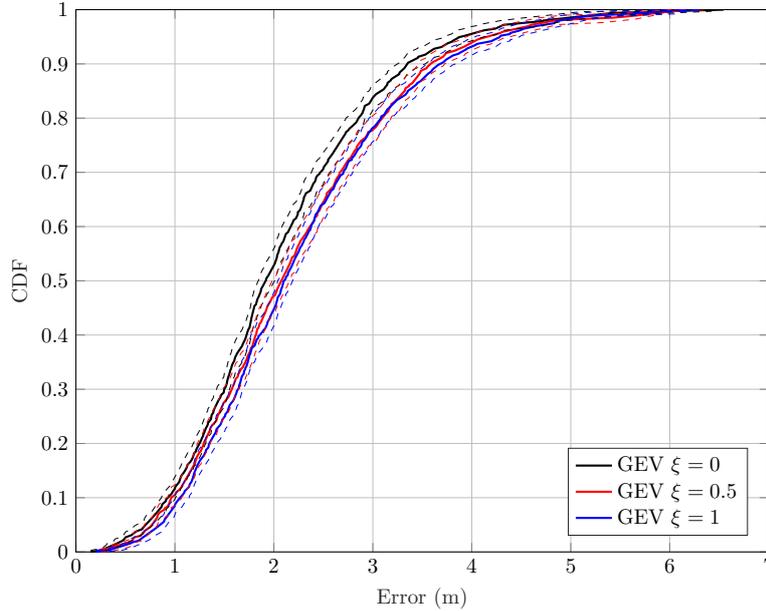


Figure 5.7: Comparison of different GEV distributed likelihood for distant peers.

As shown in Figure 5.7, the performance in terms of CDF of the positioning solution is best when the shape parameter $\xi = 0$, as expected since the GEV distribution is more symmetric for this value of ξ .

Shape Parameter	50-th percentile	75-th percentile	95-th percentile
$\xi = 0$	1.92 m	2.66 m	3.92 m
$\xi = 0.5$	2.07 m	2.85 m	4.20 m
$\xi = 1$	2.11 m	2.88 m	4.28 m

Table 5.1: Error of different GEV likelihoods evaluated at three percentiles.

As can be seen from Table 5.1, the difference between the cases is small, but it can still be claimed that the performance degrades for increasing values of ξ , meaning we are getting further away from the ideal likelihood distribution.

5.5.2 Distribution Families for Close Peers

In the previous section, a GEV distribution with $\xi = 0$ was found to be the best-performing when used as a likelihood for when the agents are far away from each other. As such, it is now kept fixed for all the tests performed in this section as distribution type 1, while different families will be tested as type 2.

Rayleigh Families

First, Rayleigh distributions with different scale parameters $\sigma = 0.5, 1, 2$ are tested and the result of this comparison is shown in Figure 5.8. Then, a similar comparison with varying parameters will be performed for the GEV.

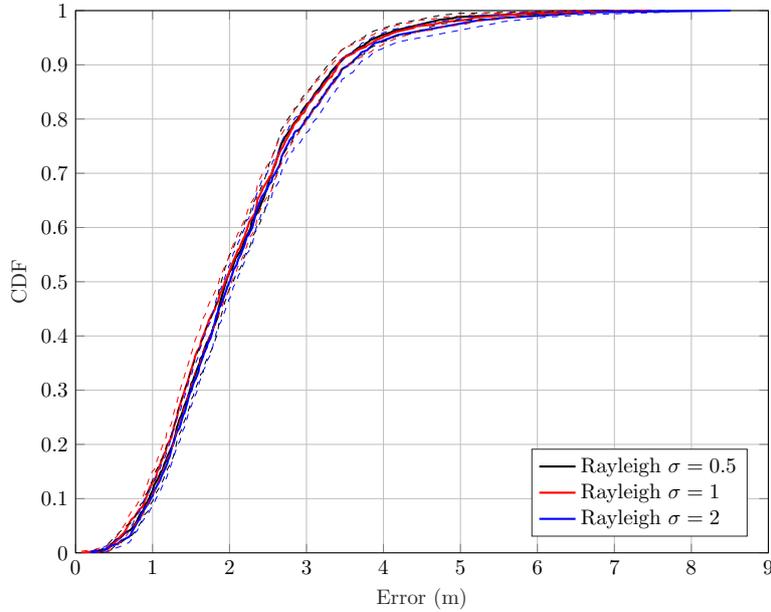


Figure 5.8: Comparison of different Rayleigh distributed likelihood for close peers.

In this simulation, a Rayleigh likelihood with $\sigma = 0.5$ has shown to provide the better performance, although the performance of the different cases is very similar, as can be seen from Table 5.2. This is due to the fact that, in the dataset being used for simulation, the cooperating agents moving over the track are close to each other for very short time-spans, and thus the AALS algorithm switches to the second distribution type rarely. Since these families of distribution are used as likelihood only on few epochs w.r.t. the entire simulation, the difference in overall performance of the PF is very small.

Scale Parameter	50-th percentile	75-th percentile	95-th percentile
$\sigma = 0.5$	1.96 m	2.66 m	3.91 m
$\sigma = 1$	1.94 m	2.70 m	3.99 m
$\sigma = 2$	2.01 m	2.75 m	4.08 m

Table 5.2: Error of different Rayleigh likelihoods evaluated at three percentiles.

Even though the difference between the three experiments is small, it can be observed that the best performance is obtained for small values of the shape parameter σ .

GEV Families

As previously done for the Rayleigh distribution, different GEV likelihoods with varying parameter $\xi = 0, 0.5, 1$ are tested, the result can be seen in Figure 5.9.

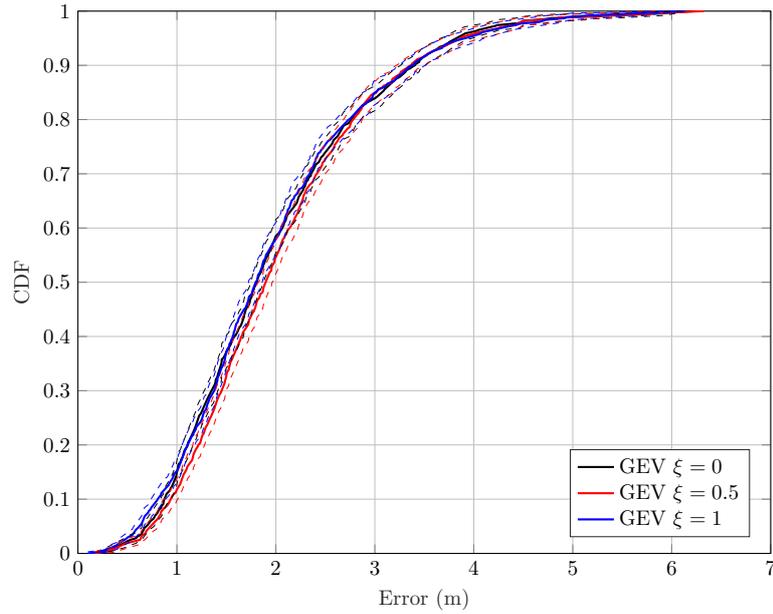


Figure 5.9: Comparison of different GEV distributed likelihood for close peers.

Once again, for the aforementioned reasons, the performance of the PF is very close for different ξ values of the GEV. Differently from the tests performed for the Rayleigh distribution though, there is no value of ξ which clearly provides the best performance.

Shape Parameter	50-th percentile	75-th percentile	95-th percentile
$\xi = 0$	1.81 m	2.55 m	3.82 m
$\xi = 0.5$	1.90 m	2.59 m	3.86 m
$\xi = 1$	1.80 m	2.49 m	3.95 m

Table 5.3: Error of different GEV likelihoods evaluated at three percentiles.

5.5.3 Performance Comparison

Comparing the results obtained in the previous two tests, it can be seen from Tables 5.2 and 5.3 that the performance in terms of CDF of the GEV distribution, for any tested value of ξ , is better than the one obtained for Rayleigh for any tested value of σ . Given the extremely skewed nature of the PDF of the Rayleigh distribution, it is reasonable to assume that its performance as a likelihood would be most suited only for cases in which the two cooperating agents are extremely close to each other (e.g. $D_M < 1$).

Comparing instead the different values of shape parameter of the GEV distribution, their performance is very similar, as shown in Table 5.3. This might be due to the fact that the current choice of threshold for the likelihood switch is greater than the actual value in which the best-fit distribution changes from one value of shape parameter to another. If that is the case, the current implementation includes cases in which different shape values perform best, thus leading to an overall similar performance between them. A possible strategy would be decreasing the value of the threshold T_M of the AALS algorithm, in order to only account for cases in which the agents are extremely close, and thus the distribution of the error is expected to be more skewed. The benefit obtained when the algorithm switches to a different likelihood to account for this skewness could be greater, but it would be counteracted by the fact that the number of epochs in which such switch would trigger would greatly decrease, as it becomes more and more unlikely that the agents are close enough to trigger the likelihood switch.

Chapter 6

Conclusions

In the context of vehicle positioning applications, GNSSs have a remarkable role since they allow receivers to estimate their own position. However, these applications have very strict safety requirements, thus needing an improvement in the performance of such positioning systems. This necessity has led to the development of Cooperative Positioning methods, which exploits the exchange of GNSS-only measurements among a network of vehicles (i.e. agents). These additional measurements allow agents to compute inter-agent distances through differential methods, and use other agents as additional anchor points w.r.t. which relative ranging measurements are used. The integration of inter-agent distances has led to the use of navigation filters such as the PF, since it is able to handle the non-stationary behaviour of such measurements. A proper integration of the auxiliary measurements can lead to a significant improvement in the quality of the solution, since the differential methods used to compute inter-agent distances allow for cancellation of common error terms between agents, thus possibly reducing the uncertainty of these auxiliary measurements. Furthermore, cooperative measurements provide information that is geometrically relevant, especially in scenarios of limited sky visibility, by reducing the GDOP (a scaling factor for the uncertainty on the positioning solution).

The work carried out during this thesis highlighted the importance of synchronizing in time the position of the aiding agent, especially for scenarios in which the velocity of the agents changes rapidly relative to the time-misalignment between the timestamp of their PVT solution. If the position of the aiding agent is not updated, the weights assigned to particles inside the PF are based on an anchor point which has a bias w.r.t. the true position of the agent, leading to a degradation of the performance. Two different strategies were proposed in order to perform this correction. The first involves the use of the velocity of the aiding agent to linearly project its most recent position solution to be synchronized with the one of the aided agent. This strategy is very simple, but holds only under the assumptions

that the motion of the agent is uniform w.r.t. the timestamp misalignment between the two agents. This means that the acceleration should be close to zero and there should not be any high-order term such as jerk. Furthermore, it requires the transmission of additional information, whose possible impact on network delays is not discussed in the thesis. The second strategy proposed requires the aided agent to use the synchronized pseudoranges (already used by the DD method) to compute the updated position of the aiding agent by means of LMS algorithm. This method requires no additional information to be transmitted, but involves more computational effort on the side of the aided agent since it needs to passively track the aiding agent. The performance of the two strategies is very similar and can yield a significant improvement in accuracy in cases of high dynamics (around 14% improvement at 95-th percentile), thus justifying the extra effort needed to perform the synchronization of the position.

Furthermore, the thesis investigated a solution to deal with the non-stationarity of the error on the baseline length. Since this quantity is computed as the Euclidean norm of a multivariate vector, the distribution of the error suffers from skewness when the norm of the baseline vector (the distance between the agents) is small w.r.t. to its variance. In scenarios involving kinematic agents, their distance changes continuously, and so does the statistical distribution on the baseline length. Therefore, a real-time method to adjust the likelihood distribution accordingly is needed. A metric known as the Mahalanobis distance is introduced, in order to quantify the statistical distance between agents. This metric is then used by a newly proposed algorithm (AALS algorithm) in order to automatically switch between different predefined likelihood distributions inside the PF. The ability to perform an automatic switch between different likelihoods depending on the proximity of agents is expected to improve the performance, since it reduces the mismodeling of the error on the measurements. The available datasets were not conceived to work in scenarios of proximity of the agents, and as such an improvement on the quality of the solution cannot be credited to a precise model of likelihood distribution. It is therefore necessary to work with new dedicated scenarios, and a new dataset is being worked on, which includes more complex and realistic trajectories of vehicles in a urban environment with more situations of proximity between them. Possible future work is expected to focus on the fine-tuning of the threshold used by the AALS algorithm to perform the switch, as well as possibly extend the experimentation to different families of distribution or different values of parameters for such distributions. Furthermore, future experimentation on the improvement obtained with multi-agent cooperation is expected.

Bibliography

- [1] P. Misra and P. Enge. «Global Positioning System: signals, measurements and performance second edition». In: *Global Positioning System: Signals, Measurements And Performance Second Editions* 206 (2006) (cit. on pp. 1–3, 5–9, 11, 14).
- [2] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007 (cit. on pp. 2–5).
- [3] GPS.gov. *GPS Space Segment*. URL: <https://www.gps.gov/systems/gps/space/> (cit. on p. 2).
- [4] GPS.gov. *GPS Control Segment*. URL: <https://www.gps.gov/systems/gps/control/> (cit. on p. 3).
- [5] Navipedia. *GPS Ground Segment*. URL: https://gssc.esa.int/navipedia/index.php/GPS_Ground_Segment (cit. on p. 3).
- [6] M. O. Karaim. *Real-time cycle-slip detection and correction for land vehicle navigation using inertial aiding*. Queen’s University (Canada), 2013 (cit. on p. 4).
- [7] GPS.gov. *GPS New Civil Signals*. URL: <https://www.gps.gov/systems/gps/modernization/civilsignals/> (cit. on p. 4).
- [8] Navipedia. *GPS Signal Plan*. URL: https://gssc.esa.int/navipedia/index.php/GPS_Signal_Plan (cit. on p. 6).
- [9] Navipedia. *GPS Navigation Message*. URL: https://gssc.esa.int/navipedia/index.php/GPS_Navigation_Message (cit. on p. 7).
- [10] M. Tamazin. «High Resolution Signal Processing Techniques for Enhancing GPS Receiver Performance». PhD thesis (cit. on p. 7).
- [11] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. C. S. Lee. «Improving Positioning Accuracy Using GPS Pseudorange Measurements for Cooperative Vehicular Localization». In: *IEEE Transactions on Vehicular Technology* 63.6 (July 2014), pp. 2544–2556 (cit. on pp. 15, 20, 22).

- [12] D. Dardari, E. Falletti, and M. Luise. *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2012 (cit. on p. 15).
- [13] F. de Ponte Müller. «Survey on ranging sensors and cooperative techniques for relative positioning of vehicles». In: *Sensors* 17.2 (2017), p. 271 (cit. on p. 15).
- [14] M. Tahir, S. S. Afzal, M. S. Chughtai, and K. Ali. «On the Accuracy of Inter-Vehicular Range Measurements Using GNSS Observables in a Cooperative Framework». In: *IEEE Transactions on Intelligent Transportation Systems* (2018), pp. 1–10 (cit. on pp. 15, 18).
- [15] K. Lassoued, I. Fantoni, and P. Bonnifait. «Mutual Localization and Positioning of Vehicles Sharing GNSS Pseudoranges: Sequential Bayesian Approach and Experiments». In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 1896–1901. DOI: 10.1109/ITSC.2015.307 (cit. on p. 15).
- [16] K. Lassoued and P. Bonnifait. *Cooperative Localization for Autonomous Vehicles Sharing GNSS Measurements*. 2019 (cit. on p. 15).
- [17] A. Minetto, G. Falco, and F. Dovis. «On the Trade-Off between Computational Complexity and Collaborative GNSS Hybridization». In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. Sept. 2019, pp. 1–5 (cit. on p. 16).
- [18] D. Yang, F. Zhao, K. Liu, H. B. Lim, E. Frazzoli, and D. Rus. «A GPS Pseudorange Based Cooperative Vehicular Distance Measurement Technique». In: *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*. May 2012, pp. 1–5 (cit. on p. 22).
- [19] A. Minetto. «GNSS-only Collaborative Positioning Methods for Networked Receivers». PhD thesis. Mar. 2020 (cit. on p. 53).
- [20] N. Das and M. Yip. «Learning-Based Proxy Collision Detection for Robot Motion Planning Applications». In: *IEEE Transactions on Robotics* (2020), pp. 1–19 (cit. on p. 56).
- [21] S. Alfano and D. Oltrogge. «Probability of Collision: Valuation, variability, visualization, and validity». In: *Acta Astronautica* 148 (2018), pp. 301–316 (cit. on p. 56).
- [22] F. Kanehiro, K. Fujiwara, H. Hirukawa, S. Nakaoka, and M. Morisawa. «Getting up Motion Planning using Mahalanobis Distance». In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 2540–2545 (cit. on p. 56).

BIBLIOGRAPHY

- [23] H. S. Bhat and N. Kumar. «On the derivation of the Bayesian Information Criterion». In: *School of Natural Sciences, University of California* 99 (2010) (cit. on p. 58).
- [24] A. A. Neath and J. E. Cavanaugh. «The Bayesian information criterion: background, derivation, and applications». In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.2 (2012), pp. 199–203 (cit. on p. 58).