POLITECNICO DI TORINO

Corso di Laurea in Communications and Computer Networks Engineering

Tesi di Laurea Magistrale

Integration of Cooperative GNSS Measurements by Particle Filter

Relatore
prof. Fabio Dovis

Studente
Alessandro Gurrieri
matricola: 239478

Anno accademico 2019-2020

Abstract

Global Navigation Satellite Systems (GNSSs) play an important role in vehicle-positioning applications. However, these systems have limited accuracy and availability to satisfy the increasingly strict safety-related requirements of this kind of applications. To overcome this issue, the recent evolution of Vehicle-to-Vehile (V2V) communication has led to the development of Cooperative Positioning methods, based on a network of vehicles able to exchange among them relative ranging information data to improve their positioning accuracy and precision. This thesis addresses the problem of integrating these additional relative measurements with the GNSS stand-alone solution. In particular, it focuses on Bayesian Filters that are generally considered as a powerful statistical tool to jointly combine data measurements from heterogeneous sources to enhance positioning performance. Despite of the non-linearity of the positioning problem (i.e. trilateration), many solutions approach it by means of linearization (i.e. Least Mean Square, Extended Kalman Filter). This methods also approximate general probability distributions of the input measurements through Gaussian distribution, for simplicity. In particular, when precise positioning is addressed, the mismodelling error due to non-Gaussian nature of relative measurements induces sever reduction in the accuracy of the solution. Among the Bayesian Filters, Particle Filter (PF) has the capability to natively handle non-Gaussian and non-linear measurement models, thus, being suitable for the aforementioned measurements integration. The integration of cooperative ranging measurements is performed by compensating for the lack of GNSS measurements in urban environment. By addressing the collaborative positioning of moving vehicles immersed in a urban scenario, an Agent Network (AN) is implemented with the intent of locating and tracking a vehicular target T by employing some static collaborative agents C_i. The static location of the collaborative agents does not imply a lack of generality in terms of relative dynamics with respect to the GNSS satellites. In this cooperative scenario, the agents interact with the target providing an estimate of their position at each time instant, along with the set of the available GNSS measurements (i.e. Doppler and pseudoranges). Target-agent distances are computed by means of Weighted Least Square Double Difference method. At first, the general Particle Filter model is optimized for a GNSS-only scenario by focusing on the resampling procedure and the particle generation process. Then, by analyzing a cooperative scenario where an agent provides relative ranging measurements to the target at each time instant, it is shown how the optimized Particle Filter can achieve higher performance with respect to the suboptimal one, by applying non-Gaussian distribution models for the likelihood of the auxiliary measurements.

Contents

Chapter 1

GPS Overview

1.1 GPS Principles

Global Position System (GPS) is one of the Global Navigation Satellite Systems (GNSSs) currently operating worldwide. An exhaustive definition of GPS can be given as: a space-based radionavigation system, broadcasting synchronized timing signals to provide estimates of Position, Velocity and Time (PVT) based on passive, one-way ranging to satellites. PVT measurements are collected by applying a spherical trilateration method, starting from the identification of the visible satellites and the estimation of satellite-user distances called *pseudoranges* [1].

1.1.1 System Architecture

GPS consists of three segments: the Space Segment, the Control Segment and the User Segment (Figure 1.1). The Space Segment includes the satellites while the Control Segment deals with the management of the satellite operations. Instead, the User Segment refers to a wide range of possible receivers, with different performance levels and additional functionalities [1].

Satellite Segment

The fully operational GPS constellation includes 32 active satellites approximately uniformly distributed around six circular orbits with four or more satellites each (Figure 1.2). The orbits are inclined at an angle of 55° with respect to the equator and are separated from each other by multiples of 60° right ascension. The orbits are non-geostationary and approximately circular, with radii of 26,560 km and orbital periods around 12 hours. Theoretically, a minimum of four GPS satellites will always be visible from most of the points

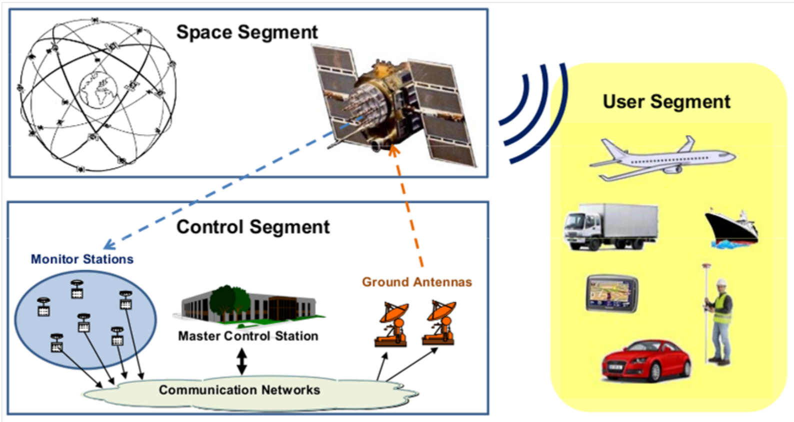

Figure 1.1. GPS Segments. Figure taken from [3].

on the earth's surface, and this number of satellites is also the minimum number needed to determine an user's position by measuring the pseudorauges and estimating the position in a passive, listen-only mode [1] [2].

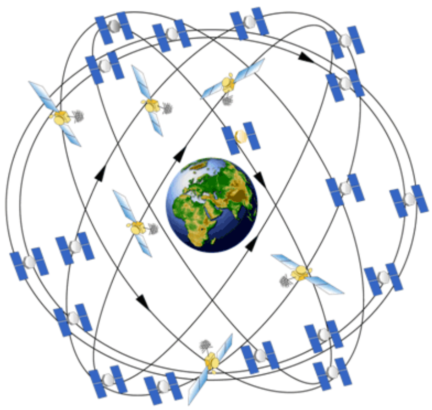

Figure 1.2. GPS Space Segment. Satellites are distributed in six orbital planes [22].

Control Segment

Control Segment is managed by the Master Control Station (MCS) which operates the system and provides command and control functions (Figure 1.3). Some of the MCS functions are based on monitoring the satellite orbits and health and commanding small maneuvers to maintain the orbit and relocations to compensate for failures. Other Control Segment functions consist on mantaining the GPS time, predicting the satellite ephemerides and

clock parameters and updating satellite navigation messages.
Besides the MCS, twelve unmanned monitor stations spread around the globe are operated remotely directly from the MCS. Equipment at a monitor station consists essentially of GPS receivers with cesium standards, meteorological instruments and communication gears to transmit information to the MCS via ground and satellite links. This information refers to estimation and prediction of the satellite orbits, satellite clock biases and GPS Time.
The ephemeris and clock parameters are computed by the MCS and uploaded to the satellites through one of the dedicated ground antennas for communications with the satellites [1].

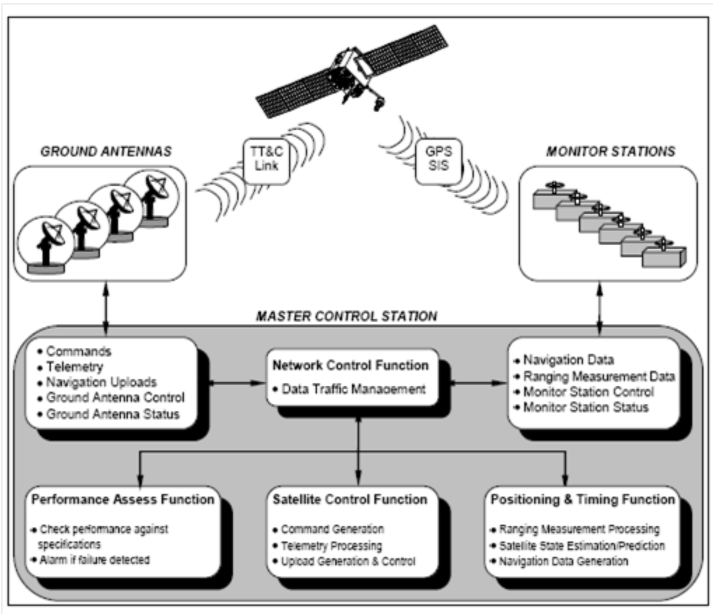

Figure 1.3. GPS Control Segment [23].

User Segment

User Segment considers all the activities related to the development and usage of military and civil GPS equipments. The revolution of integrated circuits gave a great contribution to the large-scale civil employment of GPS receivers and services, together with the benefits coming from the military GPS receivers [1] [2].

1.1.2 GPS Signal Structure

GPS satellite signals are transmitted continuously by each satellite and consist of three structural components: a carrier, a ranging code and the navigation data.

Carrier

Each GPS satellite transmits two spread spectrum L-band carrier signals: a L1 signal with carrier frequency f_1= 1575.42 MHz and a L2 signal with carrier frequency f_2= 1227.6 MHz. These two frequencies are integral multiples (f_1=1540f_0 and f_2=1200f_0) of a base frequency f_0=1.023 MHz. The signals transmitted on L1 are for Standard Positioning Service (SPS), i.e. civil service, while those transmitted on L2 are for Precise Positioing Service (PPS), i.e. the fully accuracy positioning service provided to United States and its allied military organizations. In the last years, the L2 band has started been granted also for some civil applications to provide compensation for propagation delays. Since delay varies approximately as the inverse square of signal frequency f (delay $\propto f^{-2}$), the measurable differential delay, computed by exploiting together the two carrier frequencies L1 and L2, can be used to compensate for the delay in each carrier [2].

Ranging Code

Ranging codes are associated with the provided service (SPS or PPS) and refers to a family of binary codes called Pseudorandom Noise (PRN) sequences or PRN codes [1].
L1 signal from each satellite uses Binary Phase-Shift Keying (BPSK), modulated by two PRN codes in quadrature, defined as the Coarse Acquisition code (C/A-code) and Precision code (P-code). Instead, the L2 BPSK signal is modulated by only the P-code. The signal generation scheme is shown in Figure 1.4.

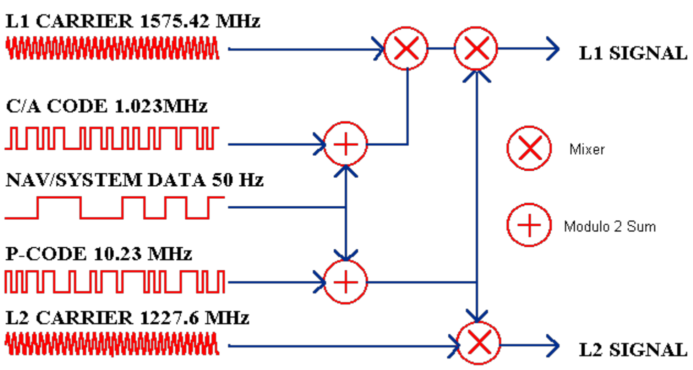

Figure 1.4. GPS Signal Generation.

The P-code is a relatively long and fine-grained code having an associated clock or chip rate of $10f_0$=10.23 MHz. The full P-code has a length of 259 days ($\approx 10^{14}$ chips), during which each satellite transmits a unique portion of the full P-code. This portion of P-code has a length of precisely one week before this code portion repeats.
The C/A-code is instead employed to simplify the rapid satellite signal acquisition and the handover to the P-code. It is a relatively short and coarse-grained code having an associated clock or chip rate f_0=1.023 MHz. The C/A-code for any GPS satellite has a length of 1023 chips before it repeats.
A panoramic of the GPS signal spectra is displayed in Figure 1.5.
Users have knowledge of the PRN codes and they can be generated and stored in the GPS satellite signal receivers. The signal transmitted by a particular GPS satellite can be acquired by generating and matching, by correlation, the local PRN code of that particular satellite [2].

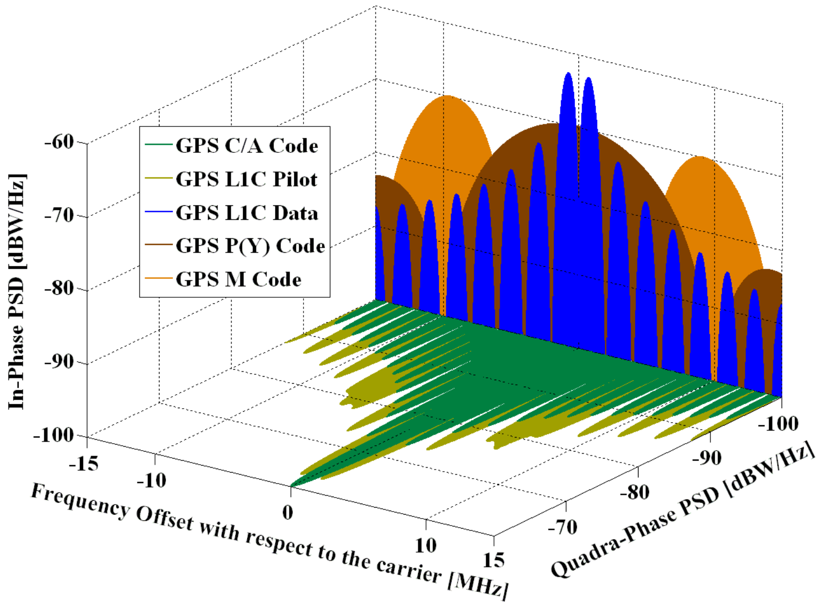

Figure 1.5. Spectra of GPS Signals [24].

Navigation Data

The navigation signal is a binary-coded message including navigational information on the ephemeris of the transmitting GPS satellite, the clock bias parameters and an almanac for all the GPS satellites, providing approximate corrections for the ionospheric signal propagation delays and for the offset time between satellite clock time and true GPS time. The navigational information is transmitted at a rate of 50 Baud [1].

1.2 Receiver Design

The first function of a GPS receiver (a generic architecture is shown in Figure 1.6) is to collect, pre-amplify and filter all the Radio Frequency (RF) signals transmitted by the satellites. This is performed by a hemispherical antenna and the RF front-end. Then, given a recent almanac and a rough idea of the user location, the receiver determines which satellites are visible, separating the different signals from satellites in view. At this point, knowing the structure of the C/A codes being transmitted by the satellites, the receiver begins the *acquisition stage* where satellites are detected and a rough estimation of their parameters is performed. Once the acquisition is complete, the *tracking stage* is executed by performing fine measurements of the signal transit time and the Doppler shift by code synchonization. At last, it decodes the navigation message determining the satellite position, velocity, and clock parameters needed to finally solve the PVT equations [1].

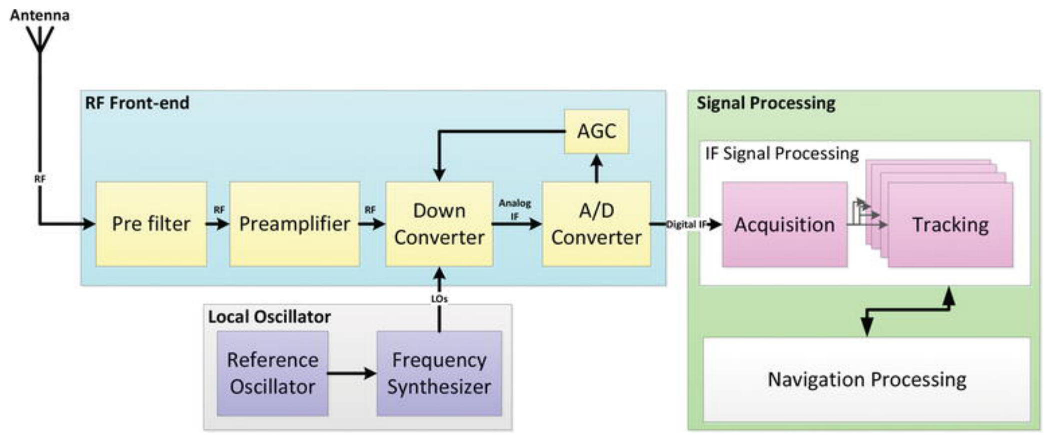

Figure 1.6. Generic GPS Receiver Architecture [25].

1.2.1 Acquisition Stage

To acquire a satellite signal, the receiver generates a replica of the known C/A-code, and attempts to align it with the incoming one by sliding the replica in time and computing the correlation. From the auto-correlation property of the signal, the correlation function exhibits a sharp peak when the code replica is aligned with the code received from the satellite. The uncertainty in matching the replica with the incoming code is limited to only 1023 code chips, and the process of aligning them is generally quick. Direct acquisition of a P(Y)-code is difficult by design due to the length of the code. The signal acquisition is accomplished in two steps and is based on the known timing relationship between the

C/A- and P(Y)-codes. First, the receiver acquires the C/A-code and then, with the aid of the timing information contained in the navigation message, acquires also the P(Y)-code [1].

1.2.2 Tracking Stage

Code tracking is implemented as a feedback control loop, called Delay Lock Loop (DLL), which continuously adjusts the replica code to keep it aligned with the code in the incoming signal. After the alignment is accomplished, the PRN code is removed from the signal, leaving the carrier modulated by the navigation message.
This signal is now tracked with another feedback control loop called Phase Lock Loop (PLL). Essentially, the receiver generates a sinusoidal signal to match the frequency and phase of the incoming signal, and in the process extracts the navigation message. The Doppler shift is measured in the phase-lock loop.
The measurement of the transit time for a signal modulated by a C/A-code is conceptually quite simple. The time shift required to align the receiver-generated code replica and the signal received from the satellite is the apparent transit time of the signal modulo 1 ms. The PRN code chips are generated by the satellite at precisely known instants according to the satellite clock, therefore the receiver can read the satellite clock time to determine when a chip was generated. Instead, the time of reception can be easily determined from the receiver clock.
If the apparent transit time is multiplied by the speed of light a measure of pseudorange is obtained. Pseudoranges measured from four, or preferably more, satellites are used to compute the position estimate. The Doppler shift, caused by the relative motion of a satellite and the user, is the projection of the relative velocity on the Line-of-Sight (LoS) and can be converted into pseudorange rate. Given the pseudorange rates corresponding to four or more satellites and the satellite velocity vectors (derived from the navigation message), a user can compute his velocity. A GPS receiver does this automatically, continuously, and virtually instantaneously [1].

1.3 PVT Computation

1.3.1 Quality Factors for PVT Estimate

The quality of the PVT estimates obtained by a user GPS receiver depends basically on two factors: the number and spatial distribution of the visible satellites and the quality of the range and range rate measurements.
The spatial distribution of the satellites relative to the user is referred to as satellite geometry. The satellite geometry changes with time as the satellites rise, move across the

sky and set. An optimal satellite spatial distribution corresponds to satellites dislocated on all sides of the user, some high in the sky and several low. If a significant part of the sky is somehow blocked (LoS not available), the user may still be able to compute PVT estimates if four or more satellites are in view, but there would generally be an accuracy penalty for poor geometry.
The second factor determing the quality of the PVT estimates is the quality of the pseudorange and Doppler measurements. There are several sources of biases and random errors, which affect the measurements. Typical errors occur in the navigation message on parameters related to satellite position and signal transmission time, affecting the pseudorange measurements. These errors are often referred to as Signal-In-Space (SIS) errors. Other errors depends on propagation delays due to ionosphere and troposphere and signal distortion due to multipath and receiver noise. In particular, the ionospheric propagation delay can be quite large, ranging from several meters to several tens of meters, depending upon the state of the ionosphere and the elevation of the satellite. Nevertheless, this error can be considerably removed by means of a dual-frequency receiver [1].

1.3.2 Position Estimation with Pseudoranges

The pseudorange measurement from the k-th satellite at epoch t (GPS time) can be modeled as

$$\rho^{(k)}(t) = r^{(k)}(t, t-\tau) + c[\delta t_u(t) - \delta t^{(k)}(t-\tau)] + I^{(k)}(t) + T^{(k)}(t) + \epsilon_\rho^{(k)}(t) \tag{1.1}$$

where $k = 1,2,\ldots,K$. $r^{(k)}(t, t-\tau)$ is are actual distance between the receiver antenna at signal reception time t and the satellite antenna at signal transmission time $(t-\tau)$; $\delta t_u(t)$ and $\delta t^{(k)}(t-\tau)$ are the receiver and satellite clock offsets respectively, relative to GPS Time (GPST); $I^{(k)}(t)$ and $T^{(k)}(t)$ are the ionospheric and tropospheric propagation delays; $\epsilon_\rho^{(k)}(t)$ accounts for modeling errors, orbit prediction errors and unmodeled effects (e.g., receiver noise and multipath).
A user can correct each measured pseudorange for the known errors using parameter values in the navigation message from the satellite. The main corrections available for a civil user are: satellite clock offset relative to GPST, relativistic effect and ionospheric delay, using the parameter values for the Klobuchar model.
The corrected pseudorange obtained by accounting for the satellite clock offset and compensating for the remaining errors in the measurements can be rewritten as

$$\rho_c^{(k)}(t) = r^{(k)} + c \cdot \delta t_u + \tilde{\epsilon}_\rho^{(k)} \tag{1.2}$$

where the error term $\tilde{\epsilon}_\rho^{(k)}$ denotes the combined effect of the residual errors.
Let vectors $\mathrm{x} = (x, y, z)$ and $\mathrm{x}^{(k)} = (x^{(k)}, y^{(k)}, z^{(k)})$, for $k = 1,2,\ldots, K$, represent respectively the position of the user at the time of the measurement and the position of the k-th satellite at the time of signal transmission. The user-to-satellite geometric range is

$$r^{(k)} = \sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} = \|\mathrm{x}^{(k)} - \mathrm{x}\| \tag{1.3}$$

By replacing (1.3) in (1.2):

$$\rho_c^{(k)}(t) = \|\mathrm{x}^{(k)} - \mathrm{x}\| + b + \tilde{\epsilon}_\rho^{(k)} \tag{1.4}$$

where the the user clock bias term is replaced with a simpler b.
The pseudorange measurements from K satellites are modeled as nonlinear equations corresponding to (1.4). Each equation involves four unknowns: b and three components of x. Clearly, four pseudorange measuremens (and equations) are required from at least four satellites to estimate the user's instantaneous position.
A simple approach to solve the K equations is to linearize them about an approximate user position and solve iteratively. The idea is to start with rough estimates of the user position and clock bias and refine them in stages so that the estimates fit better the measurements.
The generic pseudorange $\rho_c^{(k)}$ can be approximated through the Taylor expansion around a known location $\hat{\mathrm{x}} = (\hat{x},\hat{y},\hat{z},\hat{b})$ to

$$\hat{\rho}^{(k)} = \|\mathrm{x}^{(k)} - \hat{\mathrm{x}}\| + \hat{b} \tag{1.5}$$

The true position and the true clock bias can be represented as $\mathrm{x} = \hat{\mathrm{x}} + \delta\mathrm{x}$ and $b = \hat{b} + \delta b$, where $\delta\mathrm{x}$ and δb are the unkwown correction to be applied to the initial estimates, so that (the error is considered negligible for the following computations):

$$\delta\rho^{(k)} = \rho_c^{(k)} - \hat{\rho}^{(k)} = a_x^{(k)}\delta x + a_y^{(k)}\delta y + a_z^{(k)}\delta z + \delta b \tag{1.6}$$

where the cofficient are:

$$a_x^{(k)} = \frac{x^{(k)} - \hat{x}}{\hat{r}^{(k)}}, a_y^{(k)} = \frac{y^{(k)} - \hat{y}}{\hat{r}^{(k)}}, a_z^{(k)} = \frac{z^{(k)} - \hat{z}}{\hat{r}^{(k)}} \tag{1.7}$$

with $\hat{r}^{(k)} = \sqrt{(x^{(k)} - \hat{x})^2 + (y^{(k)} - \hat{y})^2 + (z^{(k)} - \hat{y})^2}$ defined as the geometrical distance between the linearization point and the satellite.

The set of K linear equations (1.6) can be written in matrix notation as:

$$\delta\rho = \begin{bmatrix} \delta\rho^{(1)} \\ \delta\rho^{(2)} \\ \vdots \\ \delta\rho^{(K)} \end{bmatrix} = \begin{bmatrix} a_x^{(1)} & a_y^{(1)} & a_z^{(1)} & 1 \\ a_x^{(2)} & a_y^{(2)} & a_z^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_x^{(K)} & a_y^{(K)} & a_z^{(K)} & 1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta b \end{bmatrix} \tag{1.8}$$

or more compactly as:

$$\delta\rho = \mathbf{H}\delta\mathrm{x} \tag{1.9}$$

If $K = 4$, the solution can be easily found directly as:

$$\delta\mathrm{x} = \mathbf{H}^{-1}\delta\rho \tag{1.10}$$

Otherwise if $K > 4$, least square solution is tipically used. The solution is given by the value of $\delta\mathrm{x}$ which minimizes the square of the residual:

$$R_{SE}(\delta\mathrm{x}) = (\mathbf{H}\delta x - \delta\rho)^2 \tag{1.11}$$

The solution can be obtained by differentiating with respect to $\delta\mathrm{x}$ to compute the gradient of R_{SE}:

$$\nabla R_{SE} = 2(\delta\mathrm{x})^T\mathbf{H}^T\mathbf{H} - 2(\delta\rho)^T\mathbf{H} \tag{1.12}$$

This gradient is set to zero and solved for $\delta\mathrm{x}$ to find a minimum value. By taking the transpose and setting it to zero, (1.12) becomes:

$$2\mathbf{H}^T\mathbf{H}(\delta\mathrm{x}) - 2\mathbf{H}^T(\delta\rho) = 0 \tag{1.13}$$

If $(\mathbf{H}^T\mathbf{H})^{-1}$ is non-singular, the solution is:

$$\delta\mathrm{x} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\delta\rho \tag{1.14}$$

This equation can be commonly solved by recursive methods or Kalman Filter, by assuming pseudorange errors as independent and identically distributed Gaussian random variables [1].

Chapter 2

Cooperative Positioning in GNSS

2.1 Overview of Auxiliary Ranging Systems for GNSS

GNSSs are exploited by a large number of Intelligent Transportation System (ITS) applications for positioning measurements. However, the evolution of vehicular applications such as collision avoidance, speed advisory, road reservation, etc. requires more and more accuracy and reliability regarding positioning data information [4].
Despite a stand-alone GNSS model represents the most suitable solution for this kind of applications, in some contexts GNSS suffers from inaccuracy of varying degrees and errors up to tens of meters [4] and this is not acceptable due to strict safety-related constraints. De facto, some scenarios, such as urban canyons, significantly degrade the received navigation signal due to the increase of multipath and interference, affecting the GNSS positioning accuracy. This led to the development of auxiliary systems to enhance positioning performance by cooperating with the GNSS [11].
A first typology of auxiliary systems refers to relative positioning sensors, such as radar sensors, which have been available in commercial vehicles for almost a decade. Besides radar sensors, 3D laser scanners can be exploited to obtain an accurate representation of the surrounding environment, while camera systems have found an application for collision avoidance, lane-keeping assistance in-vehicle and traffic sign recognition. The high precision of these devices makes possible for vehicles to obtain a detailed representation of the external scenario including the exact position of buildings, vegetation, other vehicles and further obstacles. In this way, the vehicles are able to self-localize themself and navigate through traffic [12].

The development of the Vehicle-to-Vehicle (V2V) communication has contributed to extend the capabilities of ranging sensors towards Cooperative Positioning applications by employing the exchange of ranging information among neighbor vehicles. Therefore, in a cooperative context, each vehicle is able to periodically broadcast cooperative messages towards other vehicles regarding position, speed and heading. In particular, position information can be exploited to compute ranging measurements in GNSS coordinates, providing support to the classic GNSS receiver [12].
Since sensors are very expensive and highly increase the system computational complexity, Cooperative Positioning methods based on GNSS-based ranging measurements have started being proposed. They are focused on the exchange of GNSS-only information between vehicles (or more generally agents) to provide better positioning accuracy and reliability, without the employment of any sensor. [6].

2.2 Sensor-base Cooperative Positioning

Before discussing the Cooperative Positioning, a general overview on the Sensor-based cooperative approaches for relative positioning is presented. When exploited in a non-cooperative scenario, a common problem with the sensor systems is their line-of-sight characteristic. In many situations these sensors are not able to offer a relative position and velocity estimate due to limited range, limited field of view or sight obstruction by other vehicles or surrounding objects (Figure 2.1). To overcome this issue, the relative positioning information provided by ranging sensors can be handled in a collaborative way, by making sensor-equipped vehicles able to communicate directly with each other with the purpose to improve their relative positioning estimates.

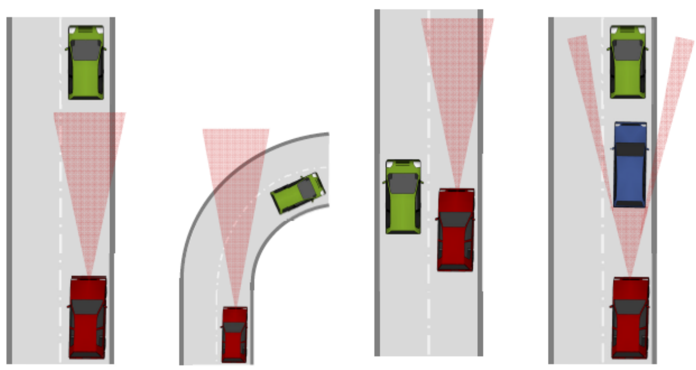

Figure 2.1. Situations of limited ranging sensor performances [12].

Radar, laser scanners and vision-based systems tipically work by first performing the measurement step, where they extract raw measurement data from a raw signal by applying signal processing algorithms. These raw measurement data consist on a cloud of reflecting points for radar and laser scanners or pixel matrices for vision-based systems. In the detection step, the raw measurement data are segmented and clustered into objects, which are then classified into different classes such as vehicles, bicycles and pedestrians. After the detection, the position of the vehicle needs to be estimated over time by a tracking step. Usually, movement models that predict the future path of the vehicle are employed. In the last step, a semantic classification provides an extensive knowledge and awareness of the surrounding environment [12].
In cooperative approaches based on V2V communication the measurement, detection and classification steps disappear, since these kinds of information are provided directly from the vehicles by the exchange of ranging data among them. [12].

2.2.1 Radio Ranging

The first descripted technique to estimate the range with respect to another vehicle consists of using electromagnetic waves and measuring the Received Signal Strength (RSS) of a signal transmitted by another vehicle. The level of this signal is proportional to the distance between the transmitter and receiver antenna. Considering a power decay model over distance given by the path loss exponent, a rough estimate of the range between two vehicles can be computed.
However, the measurement uncertainty for the RSS measurements is a key issue for this kind of application, since shadowing and multipath cause large variations in the received signal strength. Additionally, the RSS method is very sensitive to the estimated path loss exponent.
Radar sensors exploit high-frequency electromagnetic waves to measure the range and the relative speed of target objects. They have already found application for forward collision warning, lane change assistance or automatic cruise control. Two radar technologies are mainly employed in ITSs: impulse radar and Frequency-Modulated Continuous Wave (FMCW) radar.
The impulse radar works by measuring the time needed for a short pulse to travel from the radar sensor to the object, reflect and travel back to the sensor.
Instead, the FMCW radar transmits a frequency-modulated signal with a constant power envelope. The frequency difference between the outgoing and incoming waves is directly proportional to the relative distance to the target object.
The relative speed is computed by exploiting the Doppler effect. Radars are robust against

environmental conditions, such as changes in light or fog and rain, but suffers from multipath and shadowing effects [12].

2.2.2 Laser Scanners

A first model of lased-based ranging system refers to Light Detection and Ranging (LIDAR) devices. Similarly to radars, they rely on the time-of-flight of reflected light pulses to measure the distance towards an object. They usually work in the near infrared region of the electromagnetic spectrum at 905nm. However, their transmit power is limited due to eye-safety regulations, which impose a practical limit on the measuring range capability of the sensor.
A second model is instead given by laser scanners, which consist of a rotating device with few laser light sources to scan points in space. 2D laser scanners measure points in a plane, while 3D laser scanners are able to also take into account the elevation angle. They are principally implemented for obstacle detection, collision mitigation, stop-and-go assistance and recently also automatic cruise control. A first drawback of laser scanners is that they are not able to measure the relative speed between vehicles directly, but they need to differentiate the range of successive scans. Moreover, the availability of laser scanners is sensibly limited by environmental conditions such as fog, rain, dust, dirt and water which heavily degrade the performance of the sensor. Incident sunlight in the morning and afternoon hours can also affect the laser performance. At last, their cost is very high [12].

2.2.3 Time-Of-Flight Cameras

Time-Of-Flight (TOF) or 3D cameras have many fields of application. For what regards automotive applications they are principally used for driver assistance and safety applications, such as pedestrian recognition or pre-crash detection. Unlike laser scanners, the TOF camera captures the entire environment with just one single light pulse, since each camera pixel measures the time delay of modulated infrared light by comparing the phase of the outgoing and the incoming signal. Therefore the distance information is captured simultaneously for the entire surrounding scenario.
A typical value for the ranging accuracy is around 1 cm considering a maximum working range of 10 meters. TOF cameras have higher frame update rates, ranging from 20 fps up to 200 fps. The main advantage of TOF cameras is that they do not make use of mechanical components unlike laser scanners. On the contrary, TOF camera sensors, just like laser scanner sensors, are highly affected by incident sunlight, which represents a great drawback for automotive applications. Additionally, they are expensive in terms of cost [12].

2.2.4 Vision-Based Methods

Machine vision techniques are able to detect and localize objects by processing the images drawn from an imaging device like a camera. For automotive vision sensors, image processing of road scenario can provide highly valuable information which other sensors fail to obtain. For instance, vision-based solutions was recently applied for pedestrian detection, in the cases where radar sensors or laser scanners would fail. Cameras work usually at frame rates between 15 fps and 25 fps, therefore they have measuring rates comparable to radar sensors and laser scanners. However, relative velocity cannot be directly measured, but has to be differenced from successive images. Camera sensors are also sensitive to adverse lighting and atmospheric conditions. Furthermore, image processing techniques are complicated, computationally expensive and still under research [12].

2.2.5 Ultra-Wideband

Another possible Sensor-based Cooperative Positioning solution is to employ radar sensor models at lower frequencies. However, due to the larger wavelength, vehicles appear rather small and cannot provide a sufficiently big surface to reflect radio waves. For this reason, an antenna can be installed on vehicles so that the signals are amplified and reflected back to the transmitter vehicle. The relative distance from a vehicle can be estimated by measuring the Round Trip Delay (RTD) and multiplying it by the speed of light. The main issue with RTD is that the delay caused by the vehicle which reflects the signal back has to be estimated [13].

Many applications based on Ultra-Wideband (UWB) technology was implemented for collaborative radio ranging, providing estimations with precision of decimeters for a maximum distance of 300 meters [13]. In general, UWB solutions allow to achieve high localization accuracy, due to its ability to resolve multipath and penetrate obstacles [14]. Since the wide bandwidth of UWB allows very fine time resolution, several localization techniques exploiting UWB rely on Time-of-Arrival (ToA) estimation of the first path to measure the range between a receiver and a transmitter. In a cooperative scenario, the anchor nodes placed in known location and emitting UWB signals perform ranging measurements among them and not only with the target node, improving the overall accuracy and coverage. The accuracy of range-only localization systems depends mainly on the geometric configuration of the system and the quality of the range measurements, which typically degenerate in presence of multipath, LOS blockage and excess propagation delays through materials. Another drawback of UWB methods is the limited working range [14].

2.3 GNSS-based Cooperative Positioning Methods

Most of the recent studies about Cooperative Positioning has been directed towards Sensor-based models, while few contributions was focused on GNSS-based Cooperative Positioning models and measurements, obtained by means of non-Line-of-Sight network communication for absolute (vehicle self-localization on Earth) and relative (between two vehicles) positioning [15]. The previous overview of Sensor-based ranging solutions led to the conclusion that relying on Cooperative Sensor-based meeasurements requires high computational complexity and cost. Moreover, the sensors lose most of their ranging accuracy in a difficult environment.
The proposal of this thesis consists on the integration of GNSS-only auxiliary ranging measurements with the GNSS pseudorange measurements, without the deployment of any sensor. The goal is to re-evaluate the utilization of raw GNSS data in the form of auxiliary relative ranging information to improve both accuracy and precision of the navigation solution [15]. With respect to Sensor-based collaborative methods, GNSS-based ones requires less cost and less computational complexity. Moreover, since the ranging information is exchanged directly among GNSS receivers, there is no need to identify the type of the object, unlike ranging sensors which have to understand the type of object for each detection. In general, GNSS-based Cooperative Positioning methods consist in obtaining solutions from a network of peers/agents which collaborate by exchanging GNSS data, such as Inter-Agent Ranges (IARs). In a V2V communication, for instance over a Vehicular Ad-hoc Network (VANET), the GNSS-based relative localization relies on the information sent over this communication link, in particular about vehicle absolute positions in Earth coordinates, their speed and heading along with their associated uncertainty [12].

2.3.1 Belief Propagation

A possible Cooperative Positioning GNSS-based model regards the belief propagation method. It has developed for critical wireless scenario where GNSS is not available or affected by a large uncertainty. Nevertheless this application can be exploited as auxiliary ranging system of GNSS, creating a Hybrid GNSS + terrestrial localization model. The Belief Propagation technique relies on location information messages (beliefs) which reduce the uncertainty inside the network [8].
In this hybrid model the nodes deal with pseudoranges (distances from satellites) and terrestrial ranging measurements (tipically Peer-to-Peer distances) [11].

2.3.2 Differential Methods

The methods to estimate IARs are similar to Differential GPS, with the unique difference that both the involved nodes are mobile and no one works as base station. GNSS observables can be classified in raw code measurements and carrier measurements, however the second category is not preferred due to cycle slipping phenomenons and the need to resolve carrier phase ambiguity [6]. Four different methods exploiting GNSS code pseudoranges can be proposed to estimate IARs: absolute position provided by GNSS receivers, raw code pseudoranges, Single Difference (SD) and Double Differences (DD) of raw code pseudoranges. It was shown in literature that DD method is better in condition of negligibly small uncorrelated error term contribution on range measurements and assuming no or small multipath error [6].

2.3.3 Classic Double Difference Pseudorange Solution

Considering a satellite i and an user a in a GPS system and taking into account all the error contribution sources in the measurements, the *pseudorange* can be expressed as follows

$$PR_a^i = R_a^i + t_a + x^i + \varepsilon_a^i \tag{2.1}$$

where R_a is the true range between satellite i and user a; t_a is the errore due to user a's clock bias; x^i is the common noise related to satellite i; and ε_a^i is the non-common noise related to both receiver of user a and satellite i. Then, if we take the difference between the pseudoranges of two users a and b with respect to the same satellite i, the common noise due to satellite i can be removed as

$$S_{ab}^i = PR_a^i - PR_b^i = \Delta R_{ab}^i + (t_a - t_b) + (\varepsilon_a^i - \varepsilon_b^i) \tag{2.2}$$

where S_{ab}^i is the single difference of pseudorange measurements and ΔR_{ab}^i is the difference between the true ranges.
As the ranges from satellite i to users a and b are much larger than the distance between a and b, the two vectors pointing from a to i and from b to i are practically parallel to each other. As shown in Figure 2.2, ΔR_{ab}^i can be appoximated by

$$\Delta R_{ab}^i = \vec{e}^i \cdot \vec{r}_{ab} \tag{2.3}$$

where $\vec{e}^i$ is the unit vector pointing from user a or b to satellite i, and $\vec{r}_{ab}$ is the distance vector between users a and b. If a new satellite j is available to both of the users, also the clock bias of a and b can be removed by taking the double difference based on 2.2 and 2.3

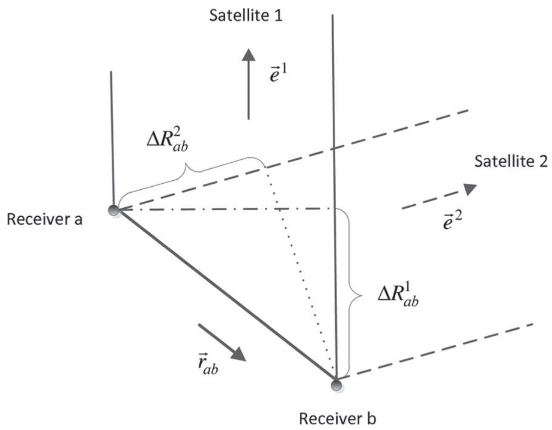

Figure 2.2. Pseudorange Double Difference [4].

as follows:

$$D^{ij}_{ab} = S^i_{ab} - S^j_{ab} = [\vec{e}^i - \vec{e}^j] \cdot \vec{r}_{ab} + [(\varepsilon^i_a - \varepsilon^i_b) - (\varepsilon^j_a - \varepsilon^j_b)]. \quad (2.4)$$

Given a set of satellites $\{0, 1, ..., n\}$ shared by the users a and b, and suppose satellite 0 is selected as the reference satellite for double difference (i.e. $j = 0$), then (2.4) can be generalized as

$$D_{ab} = \mathbf{H}\vec{r}_{ab} + \varepsilon \quad (2.5)$$

where D_{ab} is the column vector of pseudorange double differences with respect to the satellites 0 and i $(1 \leq i \leq n)$ $(D_{ab} = [D^{10}_{ab}\, D^{20}_{ab} \cdots D^{n0}_{ab}]^T)$, $\mathbf{H}$ is the column vector of the difference between two unit vectors $(\mathbf{H} = [(\vec{e}^1 - \vec{e}^0)(\vec{e}^2 - \vec{e}^0) \cdots (\vec{e}^n - \vec{e}^0)]^T)$, and ε is the column vector of aggregated non-common noise $(\varepsilon = [((\vec{\varepsilon}^1_a - \vec{\varepsilon}^1_b) - (\vec{\varepsilon}^0_a - \vec{\varepsilon}^0_b)) \cdots ((\vec{\varepsilon}^n_a - \vec{\varepsilon}^n_b) - (\vec{\varepsilon}^0_a - \vec{\varepsilon}^0_b))]^T)$. Assuming ε zero mean and equal variance, then $\vec{r}_{ab}$ can be computed by the linear least square estimator

$$\vec{r}_{ab} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{D}_{ab}. \quad (2.6)$$

When computing the unit vector of $\vec{e}^i$ $(0 \leq i \leq n)$, the positioning error by the GPS can be ignored since its negligible with respect to the satellite-Rx distance. Therefore, $\vec{e}^i$ can be obtained by the GPS corrections and the corresponding ephemeris [4].

2.3.4 Weighted Least Square Double Difference

In (2.5), ε is referred to as the aggregated non-common noise of pseudorange measurements, including both multipath and random code acquisition errors. The i-th element of ε represents the non-common noise when computing the double difference using satellite i

$(1 \leq i \leq n)$ and satellite 0 (i.e., the reference satellite) and it is defined as:

$$\varepsilon^i = (\varepsilon_a^i - \varepsilon_b^i) - (\varepsilon_a^0 - \varepsilon_b^0). \tag{2.7}$$

In a scenario where the multipath effect is not serious, such as highways, suburbs, rural areas, etc., ε is dominated by code acquisition errors, which is directly linked to the Carrier-to-Noise Ratio (CNR) of the received satellite signal. To mitigate the impact of ε, only the pseudorange measurements with high CNR values can be selected for double difference processing. A possible implementation of this is to set a threshold for selecting satellites with a good signal [5]. Although choosing satellites by setting a high CNR threshold could improve the accuracy, the number of valid samples of pseudorange measurements would be reduced significantly. Lack of valid samples for double difference processing will result in substantial delays between successive distance detections, which is not tolerable in highly dynamic vehicular networks [4].

Therefore, a new method to improve the accuracy without causing unacceptable delays in successive distance detections is needed. A Weighted Least Square Double Difference (WLS-DD) ranging technique is proposed in [4]. Assuming a non-equal variance of each ε^i, the weighted least squares estimator, i.e.

$$\vec{r}_{ab} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\mathbf{D}_{ab} \tag{2.8}$$

can be implemented as the best linear unbiased estimator, where $\mathbf{W}$ is the weight matrix corresponding to the inverse of the covariance matrix of ε.

The first operation of WLS-DD is the *selection of the reference satellite*. Given the reference satellite 0, if $(\varepsilon_a^0 - \varepsilon_b^0)$ can be kept small, then ε^i can be approximated as:

$$\varepsilon^i \approx (\varepsilon_a^i - \varepsilon_b^i). \tag{2.9}$$

Assuming independent and zero-mean random variables for ε_a^0 and ε_b^0, (2.9) holds only if the standard deviations of both ε_a^0 and ε_b^0 ε_a^0 and ε_b^0 are kept small. Therefore, the chosen reference satellite is expected to have high CNR values to both the GPS receivers a and b. By denoting Φ_a^0 and Φ_b^0 as the CNR values from satellite 0 to receivers a and b, it is required that $\Phi_a^0 \geq \mathrm{CNR}_{th}$ and $\Phi_b^0 \geq \mathrm{CNR}_{th}$, where CNR_{th} is the threshold. If more than one satellite satisfies this condition, the reference one is randomly selected among them. On the contrary, if no satellites satisfy the condition, the operation is terminated.

The second operation of WLS-DD consists in the *computation of the weight matrix*. From

(2.9), each ε_i is assumed to be uncorrelated, so that $\mathbf{W}$ can be simplified as

$$\mathbf{W} = diag\left(\frac{1}{(\sigma^1)^2}, ..., \frac{1}{(\sigma^n)^2}\right) \tag{2.10}$$

where $diag(\dot{)}$ denotes a diagonal matrix and σ^i is the standard deviation of ε^i. However, σ^i cannot be directly measured, unlike Φ_a^i and Φ_b^i. Therefore, assuming the noise is reversely proportional to the CNR, the variance can be expressed as

$$(\sigma^i)^2 \propto \frac{1}{(\Phi_a^i)^2} + \frac{1}{(\Phi_b^i)^2}. \tag{2.11}$$

Subsequentially, the Weight Matrix $\mathbf{W}$ is obtained by

$$\mathbf{W} = diag\left(\frac{(\Phi_a^1)^2 \cdot (\Phi_b^1)^2}{(\Phi_a^1)^2 + (\Phi_b^1)^2}, ..., \frac{(\Phi_a^n)^2 \cdot (\Phi_b^n)^2}{(\Phi_a^n)^2 + (\Phi_b^n)^2}\right). \tag{2.12}$$

The third operation regard the *selection of the candidate satellites*. (2.8) can be applyed only when there are at least four satellites shared by the two receivers. Therefore, in addition to the reference satellite, at least other three candidate satellites have to be selected. First of all, a minumum threshold CNR_{min} is set to prevent the use of pseudorange measurements with very high noise. Thus, only the satellites with CNR values higher than CNR_{min} can be selected. If the number of candidate satellites is less than three, the operation stops. It has to be notices that $\text{CNR}_{min} \leq \text{CNR}_{ref}$ should hold, since the reference satellite is expected to have the highest CNR value.

Chapter 3

Bayesian estimation for Hybrid PVT: Extended Kalman Filter and Particle Filter

3.1 Bayesian estimation for Hybrid PVT

In the previous chapter it was discussed how a stand-alone GNSS suffers difficult environments, such as urban canyons, under dense foliage and indoors, where the LoS between satellites and user receiver is often obstructed and so the positioning degrades or fails completely. To overcome this issue, hybrid positioning models have been proposed to improve availability and positioning accuracy by integrating auxiliary measurements. Thus, a proper receiver design able to combine the standard GNSS solution with the auxiliary one for the PVT estimation is required.

Bayesian Filters are a powerful statistical tool to jointly combine data measurements from heterogeneous sources to increase positioning accuracy and precision. The Bayesian estimation follows a probabilistic approach to determine a Probability Density Function (PDF) recursively over time by processing the incoming measurements. Then, they estimate a dynamic state (or set of states) which in a positioning context can be the location and the velocity of a target [11].

More specifically, given a set of observations z, a Bayesian estimator propose to estimate the state vector x as a realization of an unknown random variable X with a priori distribution $p_X(x)$. The state vector x corresponds to the position and other quantities of interest at a discrete time instant k, while z identify the measurements coming from a GNSS receiver or other auxiliary systems (intertial sensors, BSs, etc.). We define $p(x_k|z_{1:k})$ as

the *a-posteriori density function* which allows to determine the most probable state x_k at time k, given the history of observations $z_{1:k}$. It can be computed recursively through two stages: **prediction** and **update** [11].
The integration of cooperative measurements is tipically performed by means of the Kalman Filter (KF), which can properly perform the PVT computation of a GNSS receiver with a good compromise in terms of accuracy and computational complexity [15].
Recent studies have instead started proposing a Hybrid PVT implementation based on Particle Filter (PF), since it can achieve higher accuracy by implementing a sequential Monte Carlo approach, becoming a serious alternative to the KF Hybrid PVT models [15].

3.2 Extended Kalman Filter

3.2.1 Kalman Filter and its limit for Non-Linear Applications

Bayesian Filters start by considering the problem of estimating the sequence of hidden states $X_k = [x_0, ..., x_k]$, with initial distribution $p(x_0)$ and given the set of the observation $Z_k = [z_1, ..., z_k]$ which is related to the hidden states, at the time instant k. Following a first-order Markov chain approach, the *discrete-time state-space model* can be expressed by the two equations:

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, v_{k-1}) \tag{3.1}$$

$$z_k = h_k(x_k, u_k, w_k) \tag{3.2}$$

where $x_k \in \Re^{N_x}$ is the target state vector of order N_x at time k, u_{k-1} is the set of known system input at time index $k-1$ and v_{k-1} is the model noise sequence. Similarly, $z_k \in \Re^{N_z}$ is the observation vector of order N_z at time k and w_k is the observation noise. f_{k-1} and h_k are respectively the state transition function and the observation function and they can be linear or non-linear. Equation 3.1 is called *state (or process) equation*, while (3.2) is known as *observation (or measurement) equation* and models the relationship between the measurements and the state. The two noise contribution v_k and w_k are assumed with known statistic and mutually independent [11].
KF is an optimal Bayesian recursive filter that estimates the state of a linear dynamic system from a sequence of noisy measurements. To be applied, the following assumption must hold for (3.1) and (3.2): v_{k-1} and w_k are described by Gaussian densities of known parameter and they are additive; f_{k-1} is a known linear function of (x_{k-1},u_{k-1}) and v_{k-1}; h_k is also a known linear function of x_k and w_k. The KF achieves an optimal solution only if these constraints are verified.
However, a positioning system is non-linear by nature as well as most of the real-world systems, hence this assumption may not hold in some applications. Moreover, noise cannot

always be considered as normally distributed [11].
To overcome this limitation, suboptimal Bayesian filters has been adopted. Among the suboptimal filters, the Extended Kalman Filter is the most exploited for navigation in vehicular mobility and for integration of cooperative measurements [15], since it was designed to deal with system with non-linear state update and measurements equations.
The main idea of EKF is to linearize the state transition and observation equations around the mean of the relevant Gaussian RV by means of a Taylor-series expasion and then apply the linear KF to this linearized model. Thus, the performance of EKF depends mainly on how the system dynamics and measurements are modeled [11].
A general EKF algorithm following this model is described in the following paragraph.

3.2.2 Extended Kalman Filter Algorithm

We assume to be at a certain time instant k. The algorithm starts with the **Prediction** step, where the predicted state is estimated as follows:

$$\hat{x}_{k|k-1} = F_k \cdot \hat{x}_{k-1|k-1} + B_k \cdot u_k \tag{3.3}$$

where B_k is the input matrix, u_k is the input to the system and F_k represents the linearized state transition matrix defined as:

$$F_k = \nabla_x f_k(x, u)|_{x=\bar{x}} \tag{3.4}$$

where ∇_x is the Jacobian operator with respect to the vector x and $\bar{x}$ is the point where the linearization occurs. The next step regards the estimation of the predicted covariance matrix $\hat{P}_{k|k-1}$ related to the current a priori state vector $\hat{x}_{k|k-1}$:

$$\hat{P}_{k|k-1} = F_k \cdot \hat{P}_{k-1|k-1} \cdot F_k^T + Q_k \tag{3.5}$$

Now, the algorithm starts the **Update** procedure, by computing the *innovation vector* as the residual between the observed measurement z_k and the predicted measurement $h(\hat{x}_{k|k-1})$:

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}). \tag{3.6}$$

Next, The Kalman Gain is computed as:

$$K_k = \hat{P}_{k|k-1} \cdot H_k^T \cdot (H_k \cdot \hat{P}_{k|k-1} \cdot H_k^T + R_k)^{-1} \tag{3.7}$$

where H_k is the linearized observation matrix and R_k the covariance matrix related to the observation vector. The observation matrix H_k is given by:

$$H_k = \nabla_x h_k(x) \tag{3.8}$$

Then, the a-posteriori state estimate is computed as:

$$h(\hat{x}_{k|k}) = h(\hat{x}_{k|k-1}) + K_k \cdot \hat{y}_k \tag{3.9}$$

Similarly for the a-posteriori state covariance matrix:

$$\hat{P}_{k|k} = (I_n - K_k \cdot H_k) \cdot \hat{P}_{k|k-1} \tag{3.10}$$

where I_n is the identity matrix of dimension n.

3.3 Particle Filter

3.3.1 Particle Filter Overview

Previous works [15] has shown how a suboptimal implementation of PF achieves better performance with respect to EKF in terms of more accuracy in the state estimate by making use of a sequential Monte Carlo method. Nevertheless, the actual strength of PF is the capacity to handle Non-Gaussian and Non-Linear measurement models, when an optimized implementation is given.

At each iteration, PF is able to focus its resources (particles) on the region of the state space with high probability. However, since the complexity grows esponentially with the dimensions of the state space, we need to be very careful when dealing with high-dimensional estimation problems [11].

More specifically, it works approximating the discrete a-posteriori distribution of a generic state vector x_k at time t_k by generating and propagating a set of N particles to which a set of weights $\{\hat{x}_k^i, w_k^i\}_{i=1}^N$ is associated. The estimated a-posteriori distribution is given by:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - \hat{x}_k^i) \tag{3.11}$$

where $z_{1:k}$ represents the observation up to t_k, w_k^i is the weight associated to the i-th particle, $\delta(x)$ is the Dirac delta function and $\hat{x}_k^i$ is a propagated particle [16].

The weights are recursively set at each iteration according to a certain Likelihood function pre-defined as

$$p(z_k|\hat{x}_k^i) \tag{3.12}$$

and updated by means of a Resampling procedure [16].

3.3.2 Particle Filter Algorithm

A generic PF algorithm tipically consists on five steps (Figure 3.1). The first is the **Initialization**, where a set of particles $\hat{x}_k^i$ is generated according to

$$\hat{x}_k^i \sim p(x_k|\hat{x}_{k-1}^i) \tag{3.13}$$

where $p(x_k|\hat{x}_{k-1}^i)$ is the a priori density.
The second step is represented by the **Prediction**, where the all the generated particles are propagated following the dynamic system model

$$\hat{x}_{k,pred}^i = \Phi(\hat{x}_k^i, v_k) \tag{3.14}$$

where Φ denotes the state transition matrix and v_k the noise affecting the states.
At this point the **Weights Computation** is performed, based on the likelihood function described in (3.12). The weights are defined as:

$$w_k^i = \frac{\prod_n p(z_{n,k} - z_{n,k}^i)}{\sum_{i=1}^N \prod_n p(z_{n,k} - z_{n,k}^i)} \tag{3.15}$$

The following step is the **Resampling**, where the particles are recombined according to the weights previously assigned. This is a highly critical operation, since it has to deal with *degeneracy problem* and *sampling impoverishment* (exhaustive discussion about Resampling in Chapter 4.2).
Finally, the **Bayesian Estimation** of the state is given by the average of the resampled particle set as:

$$\hat{x}_k \approx \sum_{i=1}^N \hat{x}_{k,res}^i \tag{3.16}$$

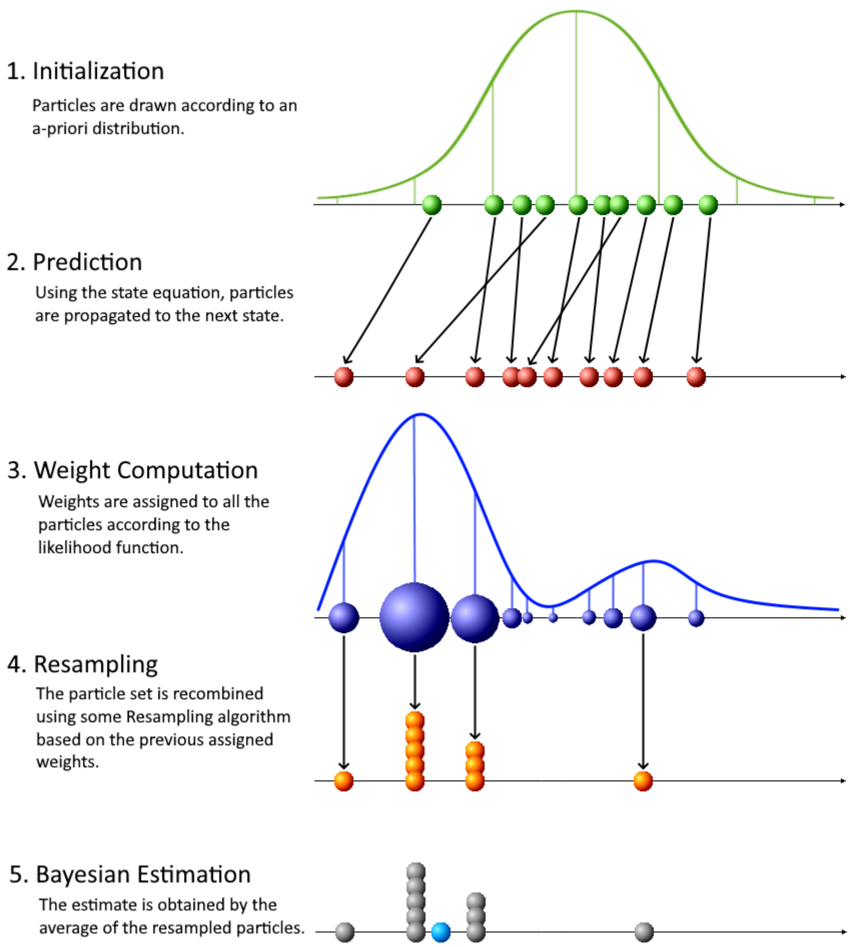

Figure 3.1. Particle Filter steps.

Chapter 4

Particle Filter Implementation for GNSS-based Cooperative Positioning

4.1 Particle Filter for Agent Network Models

4.1.1 Agent Network Implementation

By addressing the collaborative positioning of moving vehicles immersed in a urban scenario, an Agent Network (AN) is implemented with the intent of locating and tracking a vehicular target T by employing some static collaborative agents C_i. The static location of the collaborative agents does not imply a lack of generality in terms of relative dynamics with respect to the GNSS satellites.

The intent of this kind of AN consists in locating and tracking the target, by estimating its state vector x^k_{target} at each time instant k. An integration of cooperative ranging measurements is performed, since the local target pseudorange measurements may not be sufficient to obtain accurate and reliable position estimates.

In this cooperative scenario, the agents interact with the target providing an estimate of their position at each time instant, along with the set of the available GNSS measurements (i.e. Doppler and pseudoranges). Target-agent distances are computed by means of WLD-DD method (previously discussed in Chapter 2.3.3). The geometry of the network must be certainly taken into account when considering the reliability of the agents, since the accuracy of the target-agent distances depends strongly on the local position of the target with respect to the agents at each time instant.

To evaluate the system performance a Bernoullian trajectory is considered, along which the vehicular target T travels according to a uniformly accelerated dynamics. The collaborative agents C_i consist on stationary peers located in different points of the trajectory, as shown in Figure 4.1. A set of 8 visible satellites is applied to simulate the GPS constellation. The

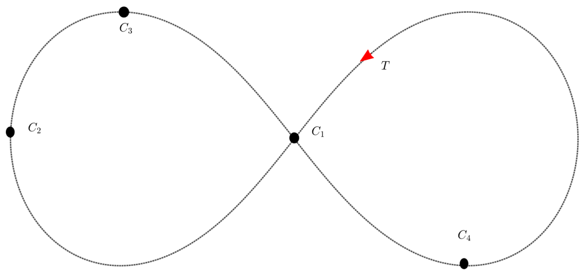

Figure 4.1. Agent positions on the trajectory

corresponding RF signals were generated by means of a high-end, professional IFEN™ NavX Signal Generator. The choice of a reduced number of satellites aim at addressing poor sky visibility for harsh environment. RF signals were then digitalized using an SDR USRP™ B210 Front-end and then processed through a propretary MATLAB™ GNSS software receiver.

4.1.2 Particle Filter Solution

As pointed out in Chapter 3.2.1, positioning systems include non-linear elements, i.e. the state equation (3.1) and the observation equation (3.2), and non-Gaussian elements, i.e. the probability distribution of the GNSS-based ranging measurements. As shown in Figure 4.2, one of the error distributions of the target-agent relative distance is not Gaussian.

In Chapter 3.3 it was stated how Particle Filter povides better performance for non-linear systems and non-Gaussian measurements compared to other Bayesian Filters. In general, PF works well in many situations where KF-based methods diverge. The drawback is an increase of the computationally complexity with respect to the EKF [17] [15], as the number of particles increases.

Given a Hybrid PVT model based on PF, when auxiliary ranging measurements are integrated (at least one agent collaborates with the target) and an optimized implementation of PF is provided, the algorithm is able to process simultaneously the pseudoranges (target-satellite distances) and the auxiliary ranges (target-agent distances) by applying respectively and separately a Gaussian distributed and a non-Gaussian distributed likelihood. Then they are combined and employed for the weight computation. The set of particles is generated over a region the state space according to a Gaussian model:

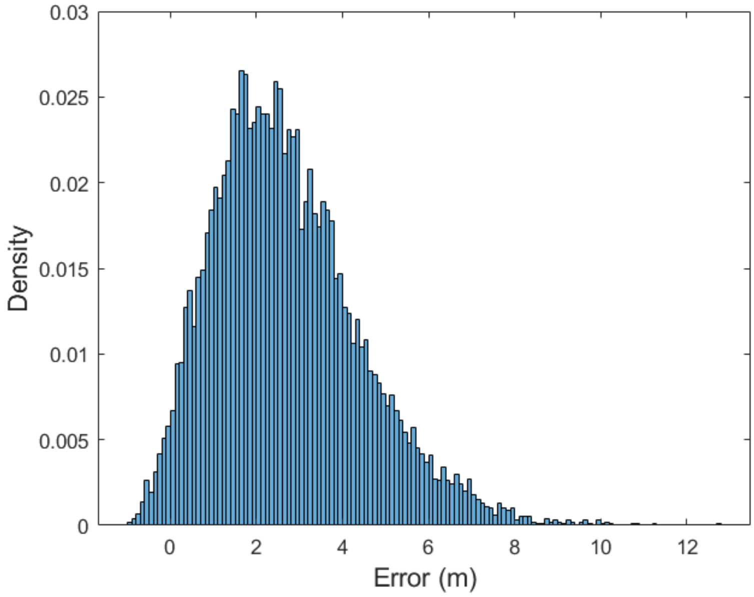

Figure 4.2. Error Distribution for one of the agents.

$$\hat{x}_k^i \sim \mathcal{N}(x_{k-1}, \sigma_x) \tag{4.1}$$

where x_{k-1} is the previous state estimate and σ_x is the state covariance matrix. A particle i represent a possibile realization of the state space vector at time k, defined as:

$$\hat{x}_k^i = [\mathrm{x}_k^i \quad \mathrm{y}_k^i \quad \mathrm{z}_k^i \quad \mathrm{b}_k^i \quad \dot{\mathrm{x}}_k^i \quad \dot{\mathrm{y}}_k^i \quad \dot{\mathrm{z}}_k^i \quad \dot{\mathrm{b}}_k^i] \tag{4.2}$$

where $[\mathrm{x}_k^i \quad \mathrm{y}_k^i \quad \mathrm{z}_k^i]$ refers to the spatial coordinates, $[\dot{\mathrm{x}}_k^i \quad \dot{\mathrm{y}}_k^i \quad \dot{\mathrm{z}}_k^i]$ to the velocity components, while b_k^i and $\dot{\mathrm{b}}_k^i$ are respectively the bias and the drift of the local clock. After each particle i is predicted according to the algorithm described in Chapter 3.3, the nominal measurement vector z_k^i is computed. At this point, the auxiliary ranges are integrated by adding them to z_k^i, creating a new nominal measurements vector $z_{k,int}^i$ as

$$z_{k,int}^i = [z_k^i \quad z_{k,aux}^i]. \tag{4.3}$$

If a suboptimal PF is considered, the likelihood function is forced to be Gaussian even if the auxiliary ranges are not normally distributed. Differently, the proposed optimized PF has the capability to separately process the pseudoranges as Gaussian distributed and the auxiliary ranges by adopting a different distribution (GEV, Rayleigh, etc.), generating two different likelihood function $p(z_k|\hat{x}_k^i)$ and $p_{aux}(z_{k,aux}|\hat{x}_k^i)$:

$$p(z_k|\hat{x}_k^i) \sim \mathcal{N}(0, \sigma_{LH}) \tag{4.4}$$

$$p_{aux}(z_{k,aux}|\hat{x}_k^i) \sim \mathcal{D}(0, \sigma_{LH}) \tag{4.5}$$

where σ_{LH} is the so called *observation noise covariance* matrix and $\mathcal{D}(0, \sigma_{LH})$ is a generic non-Gaussian distribution. The two likelihoods are then combined and employed for weight computation.

4.2 Resampling Methods

4.2.1 Degeneracy Problem and Sample Impoverishment

A common problem of PF is the *degeneracy* phenomenon. If a resampling procedure is not considered, after few iterations a particle weight will converge to 1, while all the other particles will have negligible weight [19]. This implies that a large part of computational resources are dedicated to update particles whose contribution is nearly zero [18]. Furthermore, PF will collaps to a single point processing losing its peculiarity. Previous studies [21] have shown that the variance of the importance weights can only increase over time, therefore the degeneracy problem cannot be avoided.

Degeneracy problem represents the key point which highlights why the resampling step is required to prevent high concentration of the probability mass on a few particles. The resampling step consist in generating a new set of particle $\hat{x}_{k,res}^i$ by statistically replacing time by time, each particle from the original set $\hat{x}_k^i$ according to the a-posteriori distribution shown in (3.11), so that particles with higher weights have an higher probability to be drawn in the resampled set. This corresponds to the most easy and common resampling algorithm defined as **Sampling Importance Resampling**, based on:

$$Pr(\hat{x}_{k,res}^i = \hat{x}_k^j) = w_k^j \tag{4.6}$$

where $Pr(\hat{x}_{k,res}^i = \hat{x}_k^j)$ describes the probability that a particle j from the original set is resampled as the i-th particle of the resampled one and this probability corresponds to the weight w_k^j of the particle j. The resulting resampled particles are unweighted, since the weights are reset to $1/N$. Other resampling algorithms, such as stratified, residual and systematic resampling may be applied as alternatives to this algorithm (see next paragraphs).

Although the resampling step efficiently reduces the effects of degeneracy, it introduces a new problem, defined as *sample impoverishment*. Due to resampling, the particles having high weights are statistically selected many times. This leads to a loss of diversity among the particles since the resampled particle set will contain many repeated points. The risk is again that all particles will collapse to a single point after few iterations [18]. The proposed solution is to use resampling only when a significant degeneracy is detected. A reliable measure of degeneracy is given by the effective number of particles N_{eff}, expressed as [18] [19]

$$N_{eff} = \frac{1}{\sum_i (w_k^i)^2} \tag{4.7}$$

This N_{eff} is compared with a threshold, set as $N_{th} = \frac{2N}{5}$. A small value of N_eff means high degeneracy, therefore the resampling is perormed only if $N_{eff} \leq N_{th}$.

4.2.2 Stratified and Systematic Resampling Algorithm

The main principle of these two resampling algorithms is to set N numbers from the uniform distribution

$$u_i \sim \mathcal{U}[0,1), \qquad i = 1, ..., N \tag{4.8}$$

and selecting particle x_k^j for replication, if

$$u_i \in \left[\sum_{p=1}^{j-1} w^p, \sum_{p=1}^{j} w^p \right) . \tag{4.9}$$

Stratified resampling algorithm works by considering a division into strata (layers) according to the number of particles. Inside each stratum, the resampling is performed simultaneously. The algorithm, which can be easily implemented with complexity $O(N)$, assumes that the range [0,1) is subdivided into equal parts, and the draw occurs in each such stratum

$$u_i \sim \mathcal{U} \left[\frac{i-1}{N}, \frac{i}{N} \right) \tag{4.10}$$

Particles x_j are then selected for replication according to (4.9). The main Stratified algorithm advantage is the possibility to be implemented by means of parallel computing [20]. Systematic resampling algorithm follows the same principle of Stratified one, the only difference is that the random number u_{start} is drawed only once and remains constant for that whole resampling step:

$$u_{start} \sim \mathcal{U} \left[0, \frac{1}{N} \right) \tag{4.11}$$

$$u_i = \frac{i-1}{N} + u_{start} \tag{4.12}$$

Again, particles x_j are selected for replication according to (4.9). Both of the algorithms have a very high operation speed [20].

4.2.3 Residual Resampling Algorithm

Residual Resampling (RR) starts from the assumption that the particles with large weight can be assigned to the resampled set without drawing. The algorithm proceeds at first computing the number of residual particles R. Then, it performs two different procedures: in the first part, the first $N-R$ particles are assigned to those with $w^j > \frac{1}{N}$ in deterministic way; in the second part, the weights of the already assigned particles are reduced and a new weight set is computed, then the residual R particles are assigned in stochastic way.

4.2.4 Resampling Methods Comparison

To perform the comparison among the previously discussed Resampling algorithm, the same simulation conditions are assumed for each one of them. In particular, a PF implementation with 10000 particles is exploited, considering 5 minutes of PVT computation along the trajectory shown in Figure 4.1, with no cooperative agents involved. The comparison analyzes the overall performance in terms of ECDF and computational time.

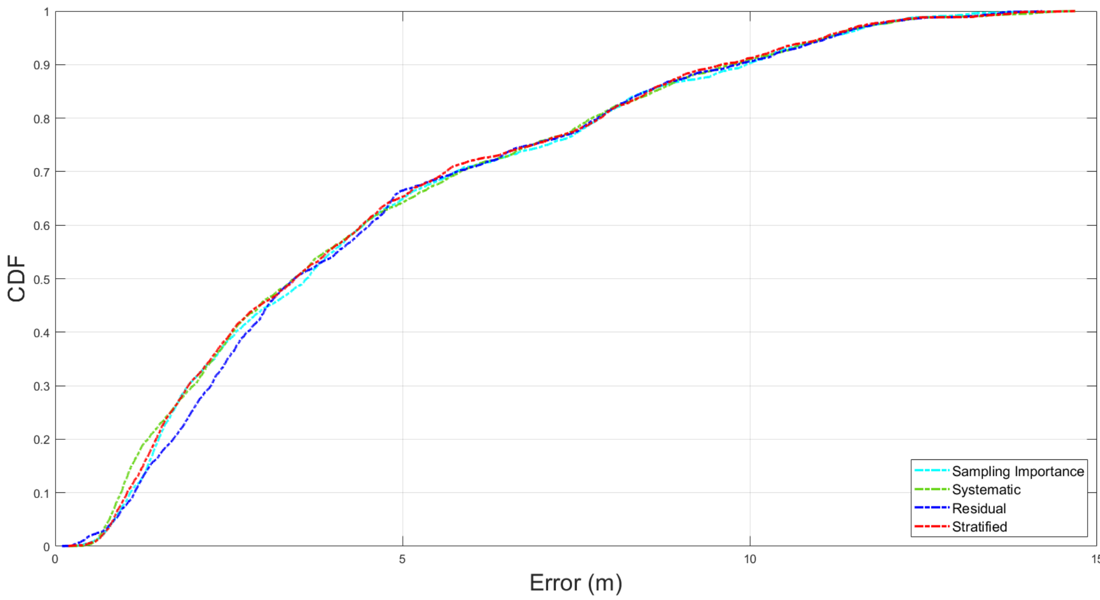

Figure 4.3. CDF comparison of Resampling methods

The result represented in Figure 4.3 reveals how the performance in terms of CDF slightly changes depending on the selected Resampling method, with errors that differ in the order of centimeters. Since a very large number of particles is considered, the effects of each

method becomes less noticeable.
A more interesting result emerges from the analysis of the execution time of the four considered methods.

Resampling Algorithm	Execution time (s)
Sampling Importance	1196
Systematic	68
Residual	115
Stratified	76

Table 4.1. Execution time comparison of Resampling methods.

The Table 4.1 points out how the three proposed methos extremely reduce the computational time of the overall simulation, in particular the Systematic and Stratified algorithms. The last one is preferred, due to the possibility to be implemented with parallel computing for future works.

4.3 A Real-Time Adaptive Covariance Model for Particle Generation Control

4.3.1 Static Covariance Model Case

From literature [19], a well-known problem with the Particle Filter is that its performance degrades quickly when a large number of states has to be estimated. From (4.2) it is known that the considered PF model estimates 8 states, thus a positionin error increase is predictable as the duration of the PVT computation increases.
From simulations, while the vehicular target is travelling along a curved trajectory (see Figure 4.4), a loss of performance in terms of divergence from the true trajectory is evaluated. When approaching a curve, the target starts to brake performing a rapid decrease of the velocity. On the contrary, at the end of the curve, it accelerates increasing rapidly the velocity. It can be assumed that PF is not able to track in a sufficiently reactive way these fast state variations, due to the high number of state dimensions it has to estimate time by time.

From (4.1), a particle is defined as a possibile realization of the state space vector x_k according to a Gaussian model with mean x_{k-1} and a predefined state covariance matrix

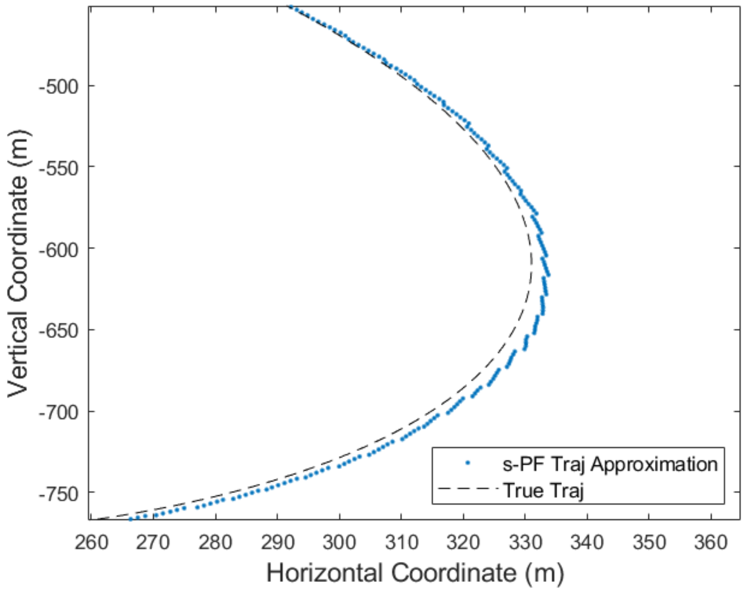

Figure 4.4. Curve detail of a PF trajectory approximation

$\sigma_{x,stat}$ set as:

$$\sigma_{x,stat} = \begin{bmatrix} 200 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} \tag{4.13}$$

Each value corresponds to the variance of one state according to (4.1). At each time instant and at each position along the trajectory, the state covariance matrix remains static and so the generated particle cloud $\hat{x}^i_{k,stat}$ as shown in Figures 4.5 and 4.6. On this particular case, a section of the trajectory where the target is moving highly along the dimension y_k and less on the dimension x_k is considered.

To mitigate this problem and make the PF more reactive to the state evolution over time, a control on the particle cloud generation can be proposed.

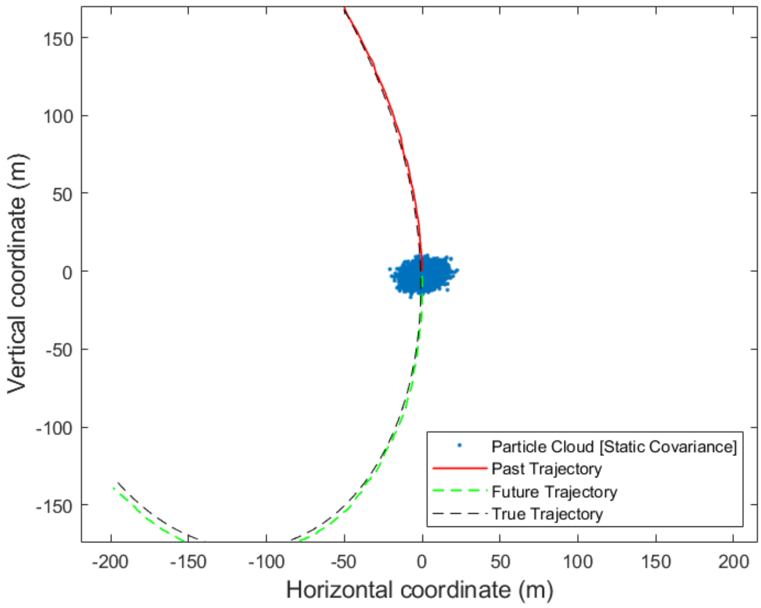

Figure 4.5. Particle cloud generated with static covariance matrix

4.3.2 Real-time Adaptive Covariance Model Proposal

The proposal discussed on this section consists on the possibility to implement a new model of the state covariance matrix $\sigma_{x,adap}$ which is able to dinamically adapt itself to the state evolution over time. At each time instant, the covariance of each state dimension is computed by relying on the previous state estimates $x_{k-4:k-1}$ (the last four are considered for this study) which provide a projection of how much the states are varying. Accordingly to this model, PF can generate the particle cloud $\hat{x}^i_{k,adap}$ by adapting it to the sector of the trajectory that the target is traveling at that time. In particular the covariance of the state x_k and y_k depends on $\dot{\mathrm{x}}_k$ and $\dot{\mathrm{y}}_k$ previous values, since they provide an estimates of the space variation along the two dimensions. To analyze how this new model affects the particle cloud, we consider the same section of the trajectory evaluated for the static case. By implementing a real-time adaptive model for σ_x, the evaluated covariance values on a

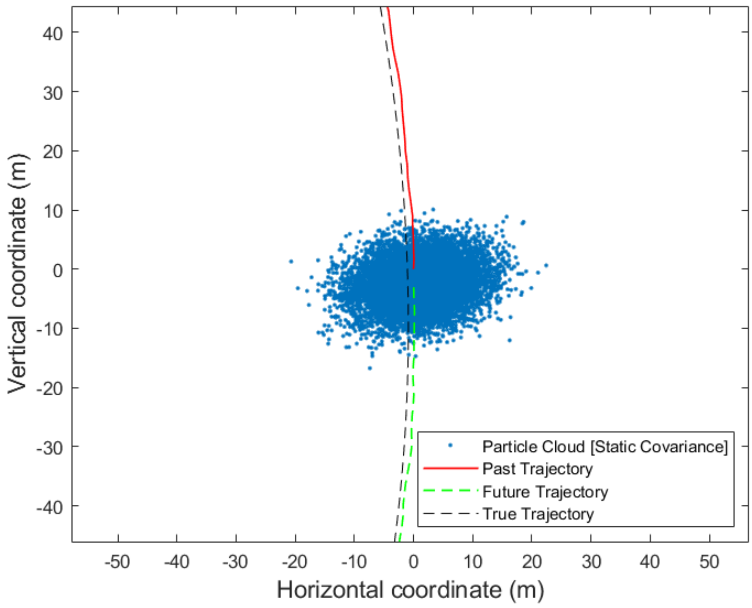

Figure 4.6. Detail of Particle cloud generated with static covariance matrix

point of the trajectory sector are:

$$\sigma_{x,adap} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix} \tag{4.14}$$

Since we are moving mainly along the y_k dimension, an higher value of covariance is evaluated and assigned to it ($\sigma_{x,adap}(\mathrm{y}_k) = 42$). On the contrary, x_k obtains a low covariance, since the target is moving slowly along this dimension ($\sigma_{x,adap}(\mathrm{x}_k) = 2$). The corresponding particle cloud modification is shown in Figure 4.7 and Figure 4.8.

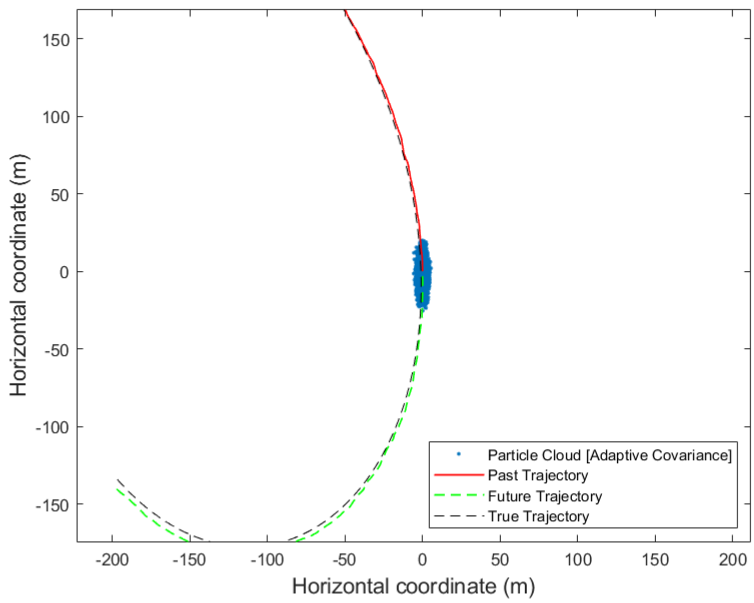

Figure 4.7. Particle cloud generated with adaptive covariance matrix

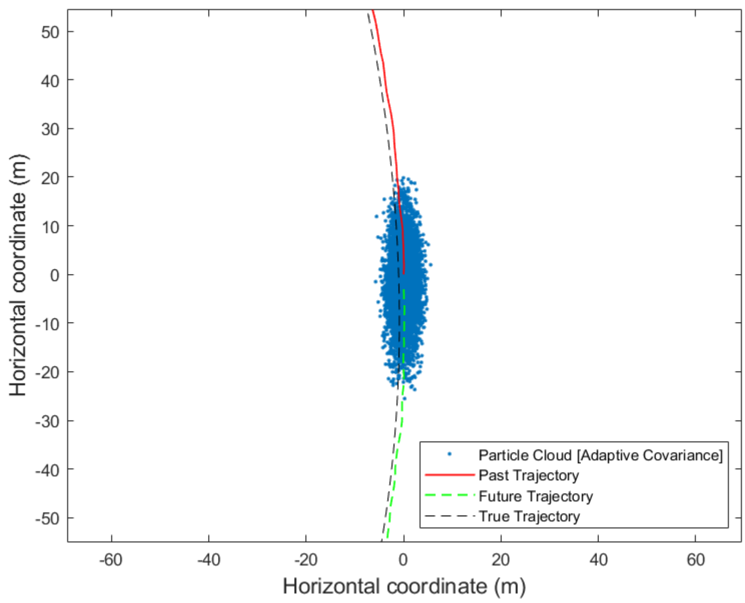

Figure 4.8. Detail of Particle cloud generated with adaptive covariance matrix

4.3.3 Static vs Adaptive Covariance Model Comparison

In conclusion, an error CDF performance comparison between the two cases is performed, considering as simulation settings: 10000 particles, 5 minutes of PVT computation along the trajectory and no cooperative agents involved (Figure 4.9).

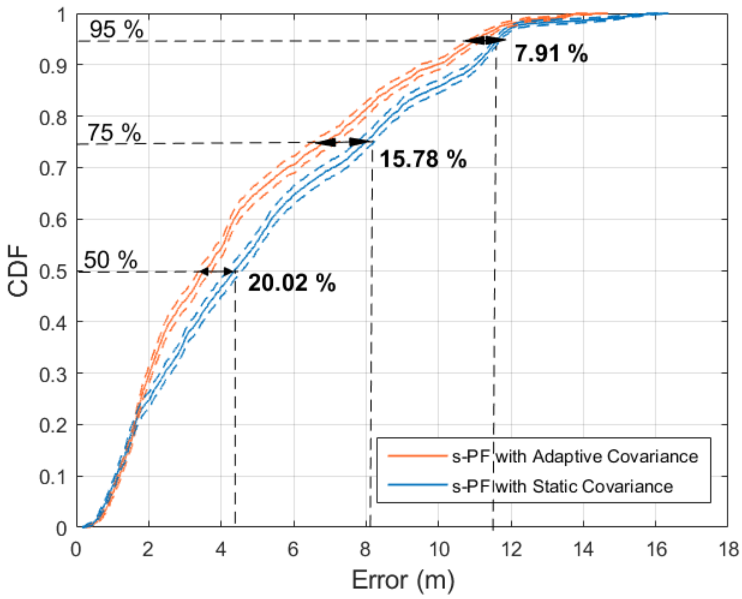

Figure 4.9. CDF Comparison between static and adaptive covariance model

An average accuracy improvement of 20.02% is achieved at the 50-th percentile (Table 5.7),

Covariance Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Static Model	4.34	8.03	11.84
Adaptive Model	3.47	6.76	10.90
Improvement	20.02	15.78	7.91

Table 4.2. Error evaluation for static and adaptive covariance model.

while a general reduced but positive improvement is evaluated for the error distribution on the other percentiles by looking at Figure 4.9.
To better understand how this adaptive model works, a detail of the trajectory section considered in Figures 4.5 and 4.7 is given in Figure 4.10. From this figure it can be noticed how, considering two passages along the same curve, the real-time adaptive model for the state covariance matrix produces a smoother trajectory approximation, with respect to

static one, and it allows PF to better track the state variations, even in case of rapid state changes. A higher divergence on the second passage along the curve is present, since the PF accuracy degrades as the PVT computation time increases, due to the high number of estimated dimensions.

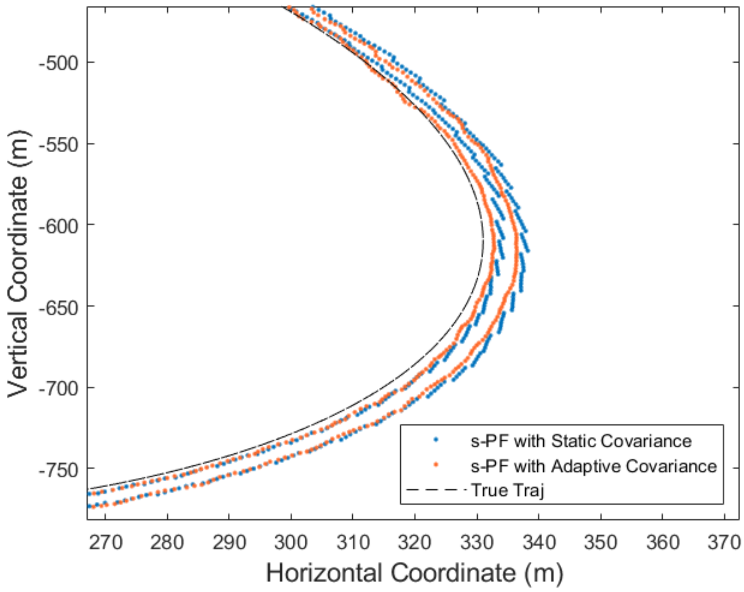

Figure 4.10. Curve detail comparison between Static and Adaptive Covariance models

Chapter 5

Results

In the previous chapter, the general PF model was enhanced to improve the accuracy and precision of the estimates when dealing with GNSS measurements, without considering the cooperative scenario. The computational complexity was sensibly reduced by means of a computationally-efficient resampling algorithm, while the positioning accuracy was enhanced by making PF able to adapt the particle cloud shape to the current trajectory.
At this point, the auxiliary ranging measurements are integrated in the model. A first set of result is obtained by considering a suboptimal implementaion of the PF, which processes the non-Gaussian auxiliary measurements by approximating their statistical distribution through a Gaussian likelihood model. A comparison between non-Cooperative and Cooperative case is given. Then, an optimized implementation of PF is realized by applying non-Gaussian models for the computation of the likelihood of the auxiliary measurements, according to (4.5). More specifically, General Extreme Value (GEV) and Rayleigh distribution models are considered according to previous investigations on the statistics of the DD ranging method, requiring an analysis of their performance behaviour by varying their parameters (k for GEV and σ for Rayleigh). Finally, a comparison between these two models with respect to the Gaussian one is shown. An additional proposal for an a-posteriori adaptive implementation for the likelihood distribution of the auxiliary measurements is presented.
For all the simulation described in this chapter they were considered: 10000 particles, Stratified Resampling algorithm and Adaptive covariance model for the particle generation.

5.1 Non-Cooperative Case vs Cooperative Case

This first result compares the accuracy of s-PF without cooperation and s-PF with the integration of the auxiliary measurements by considering a different agent per each simulation. Performances are estimated by ECDF evaluation. A first comparison is performed by considering only 90 seconds of PVT computation along a rectilinean trajectory (Figure 5.1). Instead, the second one is extended up to 3 minutes covering a full lap along the trajectory (Figure 5.2).

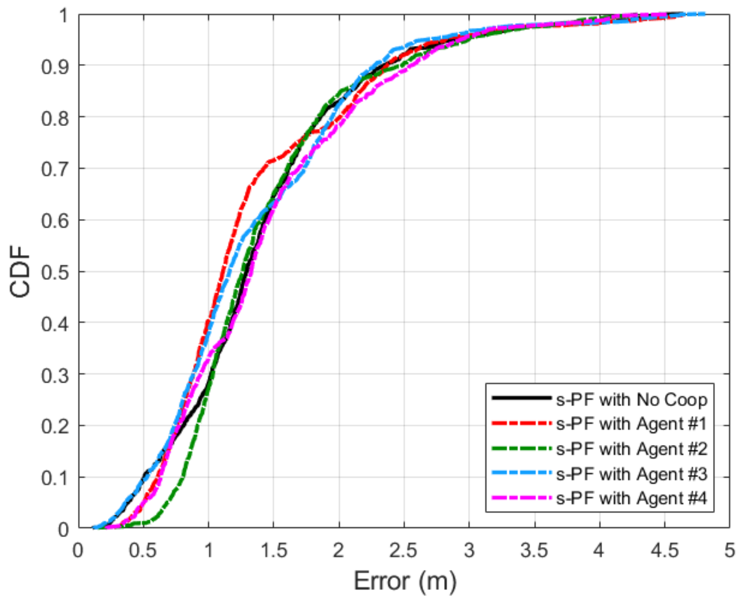

Figure 5.1. ECDF of Non-Cooperative case vs Cooperative case assuming Gaussian auxiliary measurement model, along a rectilinean trajectory.

Aiding Peer	50-th percentile Improvement (%)	75-th percentile Improvement (%)	95-th percentile Improvement (%)
Agent #1	14.07	1.89	2.31
Agent #2	2.35	-0.24	-3.92
Agent #3	10.17	-6.6	6.90
Agent #4	-2.60	-8.64	-1.9

Table 5.1. Improvement for each agent assuming Gaussian auxiliary measurement model, along a rectilinean trajectory.

In both cases the evaluated errors differ in the order of centimeters, therefore on this considered scenario the cooperation provides only a slight improvement, although for some

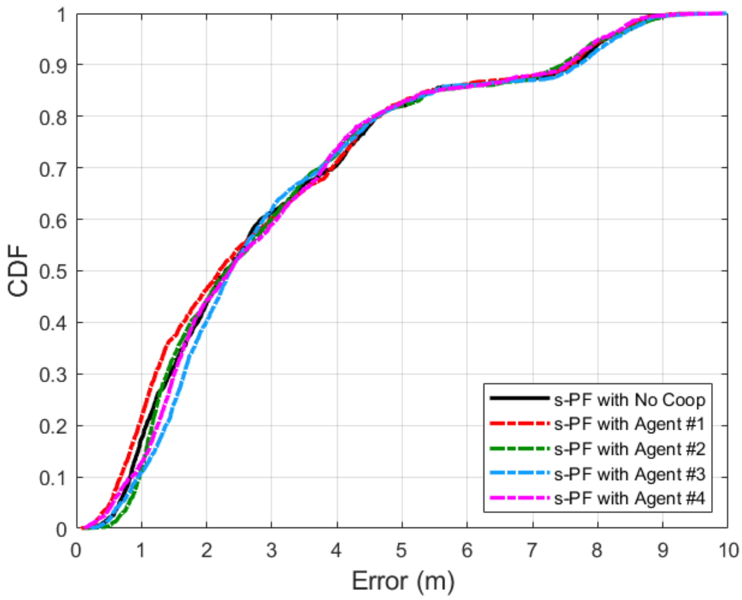

Figure 5.2. ECDF of Non-Cooperative case vs Cooperative case assuming Gaussian auxiliary measurement model, along a full lap of the trajectory

agents and percentiles it is negative (Table 5.1). For what concerns the first scenario, a rectilinean trajectory is easier to be predicted and tracked, since the applied propagation model of the motion is linear. Therefore cooperation does not necessarely produce a sensibly higher improvement. When considering the full trajectory, the geometry of the system and the relative position of the target with respect to the agents have to be taken into account, since they affect the accuracy and precision of the auxiliary ranging measurements. These results are obtained by employing a suboptimal PF, which forces the Non-Gaussian error distributed ranging measurements to match Gaussian likelihood models. This justified the investigation of a more accurate model for the statistics of the auxiliary measurements.

5.2 Optimized PF Implementation for Cooperative Positioning

To enhance Cooperative Positioning by PF, an optimized implementation of the filter is required. It can be performed by exploiting non-Gaussian distribution models for the generation of the likelihood function of the auxiliary measurements. Preliminary investigations about the error distribution of DGNSS range measurements provided they can follow GEV

or Rayleigh distributions when they are obtained from Gaussian-distributed pseudoranges. Both of them depends on two parameters (k for GEV and σ for Rayleigh) which modify the shape of their PDF. When the two distribution models are integrated inside the PF, these parameters affect the generation of the likelihood hence the accuracy of the estimation. Before comparing the performance of the optimized PF model with respect to s-PF, an analysis of these two distributions by varying the shape parameters is needed.
For these measurements, only the Agent #1 is considered.

5.2.1 General Extreme Value Distribution

The GEV distribution is a family of continuous probability distributions developed within extreme value theory, which provides the statistical framework to make inferences about the probability of very rare or extreme events. The GEV distribution combine together the Gumbel, Fréchet and Weibull distributions into a single family to allow a continuous range of possible shapes. These three distributions are also known as type I, II and III extreme value distributions. Its characteristic parameters are the shape paramater k, the location parameter μ and the scale parameter σ and it belongs to the type I, II and III respectively when the shape parameter is equal to 0, greater than 0 and lower than 0. Based on the extreme value theorem the GEV distribution is the limit distribution of properly normalized maxima of a sequence of independent and identically distributed random variables. Thus, the GEV distribution is used as an approximation to model the maxima of long (finite) sequences of random variables [26].
The PDF of the GEV distribution, given the previous parameters is

$$f(x|k,\mu,\sigma) = \frac{1}{\sigma} e^{-(1+k\frac{x-\mu}{\sigma})^{-\frac{1}{k}}} (1+k\frac{x-\mu}{\sigma})^{-1-\frac{1}{k}} \tag{5.1}$$

for

$$1+k\frac{x-\mu}{\sigma} > 0$$

. For $k=0$ the PDF becomes:

$$f(x|0,\mu,\sigma) = \frac{1}{\sigma} e^{-e^{-\frac{x-\mu}{\sigma}}} - \frac{x-\mu}{\sigma} \tag{5.2}$$

In Figure 5.3, a generic GEV-shaped likelihood for $k=0$ is shown.
To evaluate what is the best value (or values) of the parameter k for the GEV distribution when employed to model the statistics of the auxiliary measurements, four different portion of the Bernoullian trajectory are considered: starting from the rectilinean one traveled in the first 90 seconds of PVT computation, the duration of the PVT computation is extended simulation by simulation as well as the traveled trajectory up to 5 minutes which

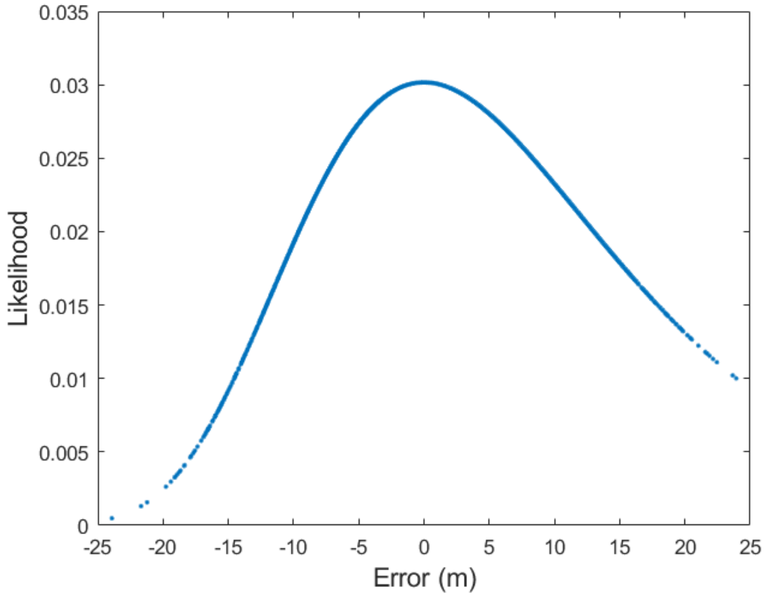

Figure 5.3. Likelihood function applying a GEV Distribution with $k = 0$.

corresponds to almost two laps. The considered set for the k values is $k = \{-0.5, 0, 0.5, 1\}$. The results are evaluated in terms of ECDF curves by varying the k parameter according to the chosen set of values.

From Figures 5.4, 5.5, 5.6 and 5.7, the best value of k in terms of positioning error performance changes depending on the considered trajectory. GEV Distribution with $k = 0$ better approximates the error distribution of the auxiliary measurements along a rectilinear trajectory, while $k = 0.5$ and $k = 1$ starts providing better accuracy when the trajectory becomes longer and more complicated, therefore when the target-agent distance varies a lot and quickly.

5.2.2 Rayleigh Distribution

Rayleigh distribution is defined as a continuous probability distribution for nonnegative-valued random variables. Its PDF can be expressed as:

$$f(x|\sigma) = \frac{x}{\sigma} e^{\frac{-x^2}{2b^2}} \tag{5.3}$$

In Figure 5.8, a generic likelihood obtained by applying a Rayleigh Distribution with $\sigma = 0.5$ is shown.

The same sequence of simulations used for the GEV distribution is performed considering

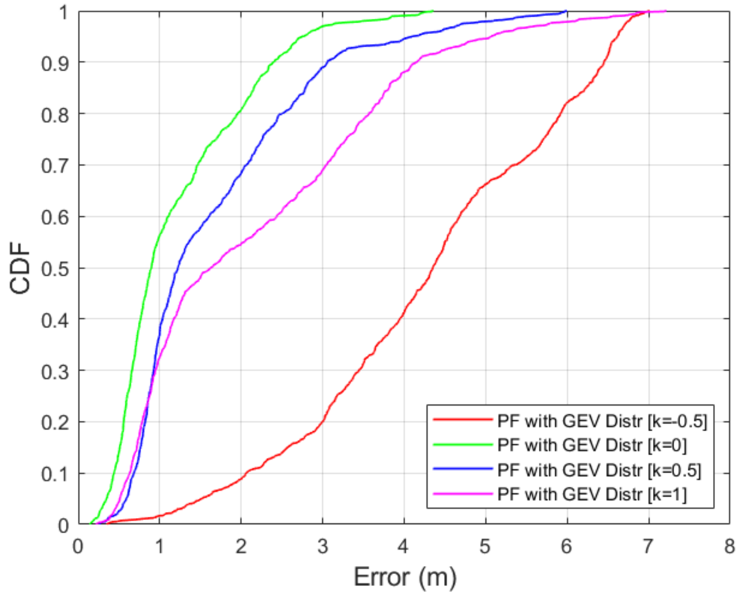

Figure 5.4. ECDF of optimized PF assuming GEV distributed likelihood for auxiliary measurements, after 90 seconds of PVT computation along a rectilinean trajectory.

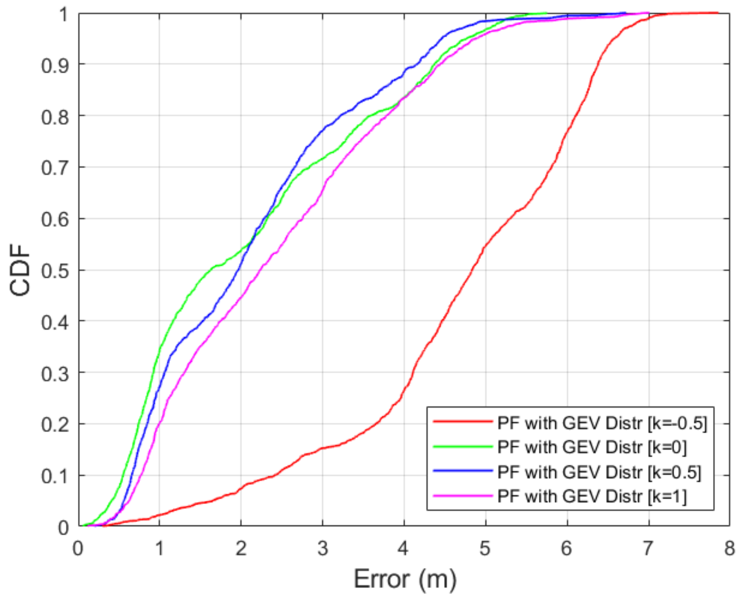

Figure 5.5. ECDF of optimized PF assuming GEV distributed likelihood for auxiliary measurements, after 2 minutes of PVT computation.

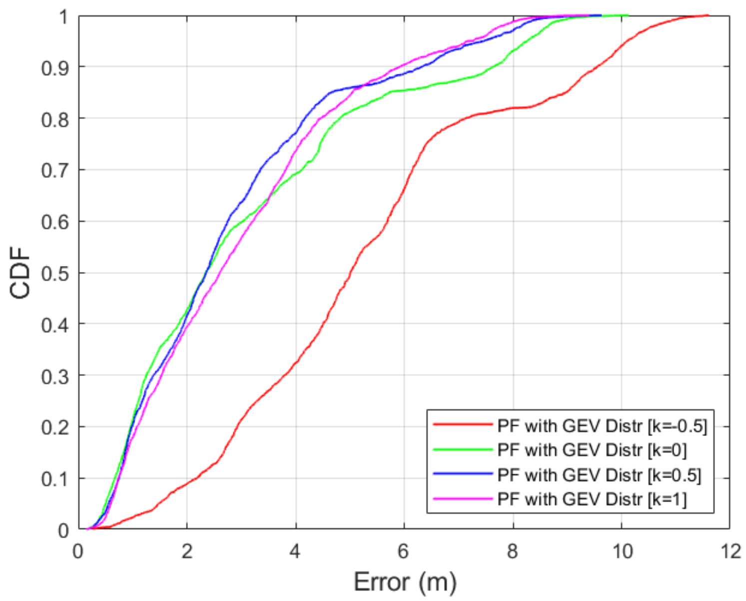

Figure 5.6. ECDF of optimized PF assuming GEV distributed likelihood for auxiliary measurements, after 3 minutes of PVT computation.

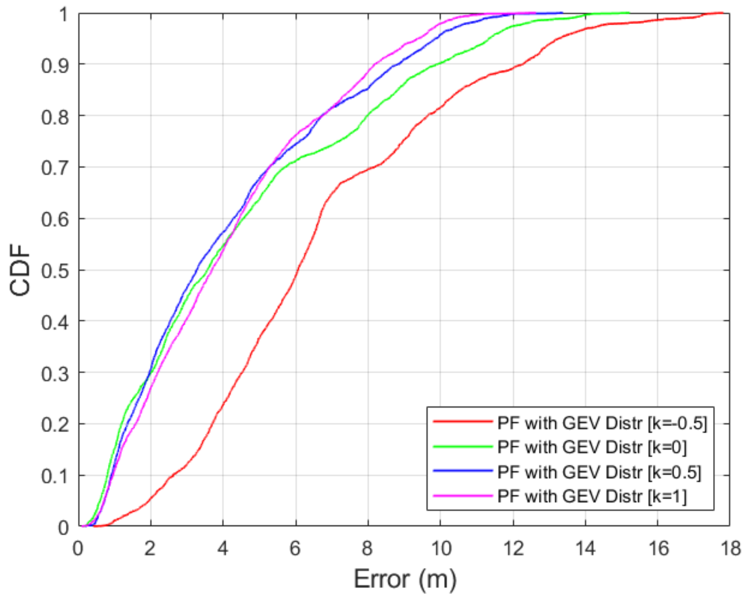

Figure 5.7. ECDF of optimized PF assuming GEV distributed likelihood for auxiliary measurements, after 5 minutes of PVT computation.

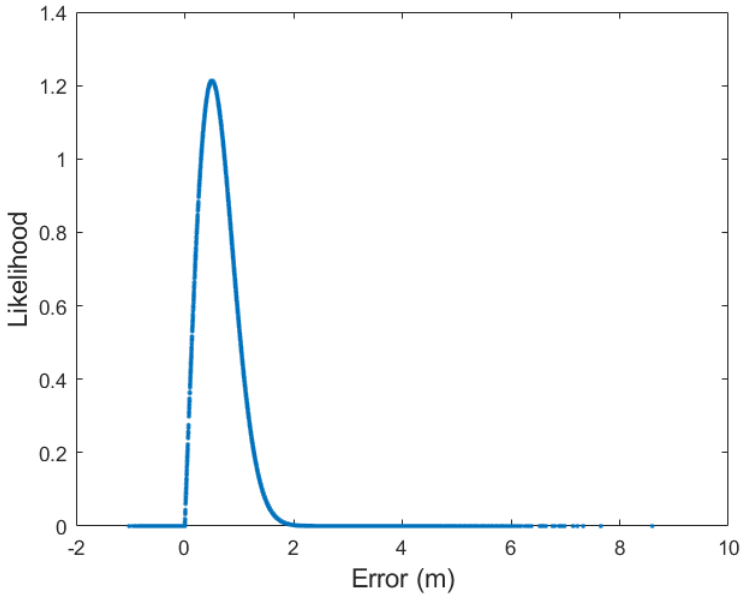

Figure 5.8. Likelihood function applying a Rayleigh Distribution with $\sigma = 0.5$.

instead the Rayleigh distribution for the set $\sigma = \{0.5, 1, 2, 3\}$. Only the last one simulation (5 minutes of PVT computation is shown in Figure 5.9, since the performance behaviour by varying σ is practically the same along the different portions of the Bernoullian trajectory.

5.2.3 Distribution Comparison

According to the previous results, $k = \{0, 0.5, 1\}$ for the GEV Distribution and $\sigma = 0.5$ for the Rayleigh one correspond to the values which provide the best likelihood models for the auxiliary measurements in an optimized PF implementation. At this point, a comparison between GEV and Rayleigh distributions (setting the collected optimal k and $sigma$ values) with respect to the Gaussian one is performed by considering the same four durations of PVT computation and related trajectory lengths. The performance comparison in terms of ECDF is evaluated in Figures 5.10, 5.11, 5.12 and 5.13 considering respectively 90 seconds, 2, 3 and 5 minutes of PVT computation. Error estimates at the percentiles of interests are shown in Tables 5.2, 5.3, 5.4, 5.5.

The first clear result is that Rayleigh distribution is not a good model for this kind of measurements, since the related PF performance is highly below the Gaussian model for each considered case.

Regarding the GEV Distribution, in general there is always at least one ECDF curve which is better than the Gaussian one. In the rectilinear portion of the Bernoullian trajectory

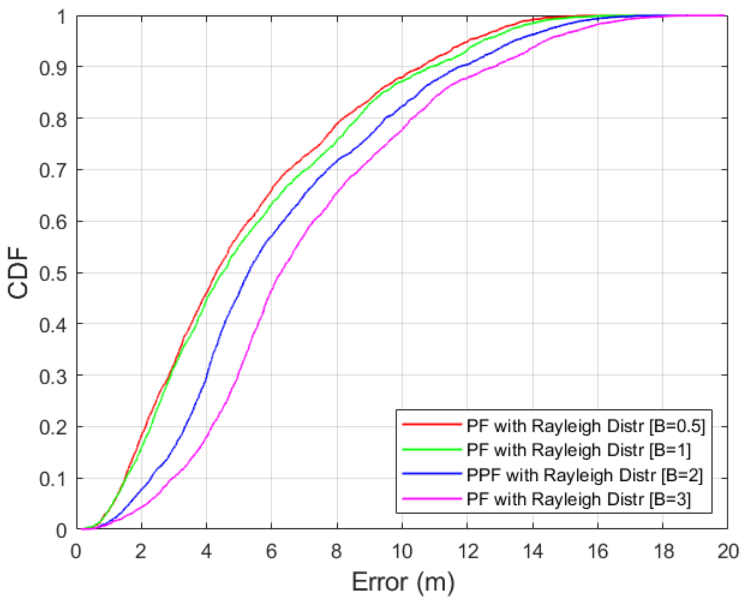

Figure 5.9. ECDF of optimized PF assuming Rayleigh distributed likelihood for auxiliary measurements, after 5 minutes of PVT computation.

(Figure 5.10), a slight improvement is provided by the GEV Distribution with $k = 0$, with a maximum one corresponding to 7.70% at the 75-th percentile. When a full lap of trajectory is considered (Figure 5.12), GEV Distribution with $k = 0.5$ and $k = 1$ provides better improvement for all the percentiles higher than the 50-th one, varying from 1.9% to 13%. In a full 5 minutes of PVT computation (Figure 5.13), the maximum improvement increases up to 16.56%, considering the 95-th percentile of the ECDF curve for $k = 1$.
An important consideration on these results is that the general GEV Distribution achieves a better accuracy and precision with respect to the Gaussian Distribution, for this typology of auxiliary measurements. However, the performance behaviour of the single GEV distributions by varying the k parameter are quite different when different parts of the trajectory are considered. Thus, an adaptive implementation for the distribution of the likelihood of the auxiliary measurements according to the specific portion of the traveled trajectory can be proposed.

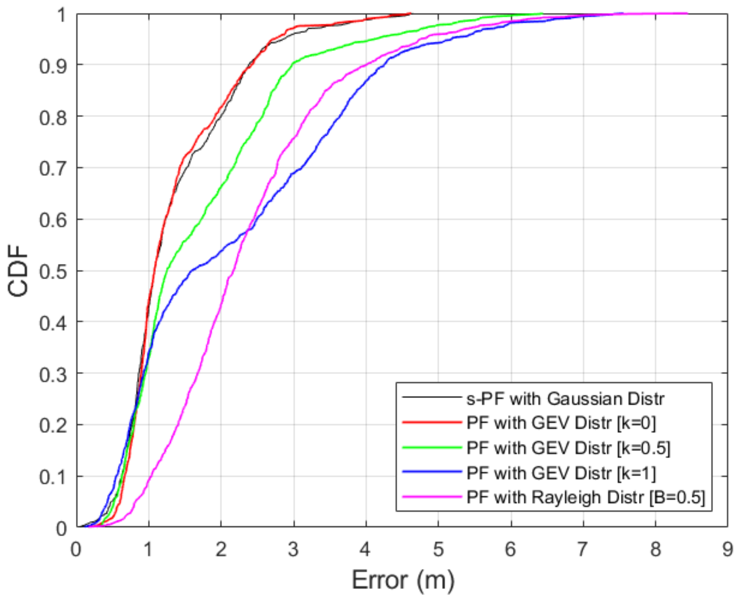

Figure 5.10. Comparison among the different distributions for auxiliary measurements, after 90 seconds of PVT computation.

Distribution Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Gaussian Distr.	1.08	1.79	2.86
GEV Distr. $[k = 0]$	1.08	1.65	2.86
GEV Distr. $[k = 0.5]$	1.25	2.34	4.15
GEV Distr. $[k = 1]$	1.60	3.36	5.18
Rayleigh Distr. $[\sigma = 0]$	2.17	2.96	4.75

Table 5.2. Error evaluation for different models of the probability distribution of the auxiliary measurements considering 1 minute of PVT computation.

5.3 A-Posteriori Adaptive Implementation for the Likelihood Distribution of the Auxiliary Measurements

To better exploit the potentiality of the GEV Distribution for this particular agent network and vehicular scenario, an adaptive selection of the best likelihood distribution of the auxiliary measurements is proposed in this section. The full trajectory is divided into six sectors as shown in Figure 5.14. For each sector, a PVT computation is performed and the distribution for the likelihood of the auxiliary measurements which best matches the error distribution of the auxiliary measurements along that sector is selected. In Figure 5.15,

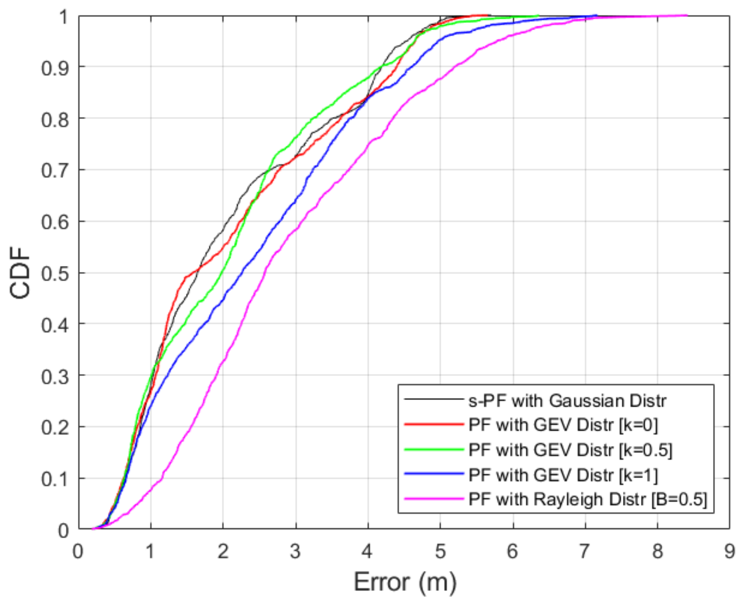

Figure 5.11. Comparison among the different distributions for auxiliary measurements, after 2 minutes of PVT computation.

Distribution Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Gaussian Distr.	1.64	3.12	4.53
GEV Distr. $[k = 0]$	1.58	3.25	4.65
GEV Distr. $[k = 0.5]$	1.98	2.92	4.65
GEV Distr. $[k = 1]$	2.27	3.48	4.97
Rayleigh Distr. $[\sigma = 0]$	2.57	4.02	5.78

Table 5.3. Error evaluation for different models of the probability distribution of the auxiliary measurements considering 2 minutes of PVT computation.

the error distributions of the auxiliary ranging measurements are showed sector by sector. Table 5.6 shows how the GEV Distribution with $k = 0$ provides better positioning accuracy along rectilinear trajectories, while it achieves better accuracy along curved trajectories with the values $k = 0.5$ and $k = 1$, in particular with the last one.
The results obtained so far are then combined and a final simulation is performed by applying for each sector the a-posteriori selected distribution. A comparison with the Gaussian Distribution is presented in Figure 5.16. From the provided ECDFs, it can be noticed how the new model accounting for all the considered GEV distributions exhibits a general better accuracy with respect to the Gaussian one. An improvement of 19.48% is

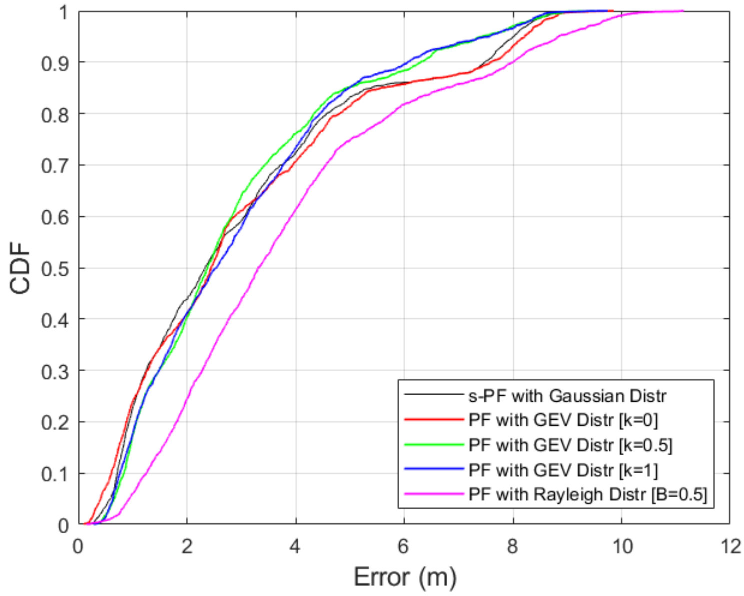

Figure 5.12. Comparison among the different distributions for auxiliary measurements, after 3 minutes of PVT computation.

Distribution Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Gaussian Distr.	2.29	4.31	8.05
GEV Distr. $[k = 0]$	2.40	4.38	8.21
GEV Distr. $[k = 0.5]$	2.25	3.71	7.50
GEV Distr. $[k = 1]$	2.59	4.33	7.29
Rayleigh Distr. $[\sigma = 0]$	3.25	5.29	8.56

Table 5.4. Error evaluation for different models of the probability distribution of the auxiliary measurements considering 3 minutes of PVT computation.

achieved at the 50-th percentile (around 40 centimeters for errors of the order of 2 meters), while reduced improvements of 17.41% (around 70 centimeters for errors of the order of 4 meters) and 6.23% (around 50 centimeters for errors of the order of 8 meters) are obtained respectively for the 75-th and 95-th percentiles (Table 5.6).

This final result shows how PF with an optimized implementation can achieve higher performance with respect to the suboptimal one, by exploiting GEV distributed models for the generation of the likelihood function of the auxiliary measurements, when cooperative ranging measurements present non-Gaussian error distributions.

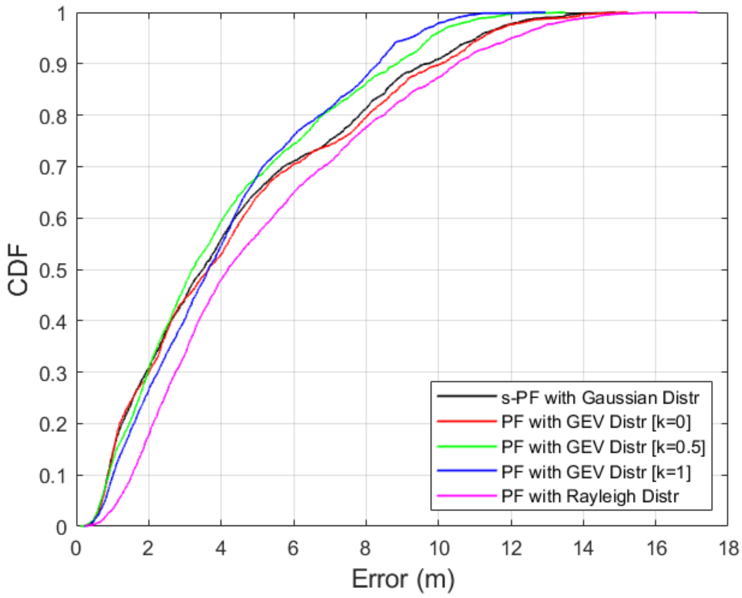

Figure 5.13. Comparison among the different distributions for auxiliary measurements, after 5 minutes of PVT computation.

Distribution Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Gaussian Distr.	3.50	6.98	11.04
GEV Distr. $[k = 0]$	3.66	7.21	11.18
GEV Distr. $[k = 0.5]$	3.19	6.13	9.78
GEV Distr. $[k = 1]$	3.64	5.90	9.21
Rayleigh Distr. $[\sigma = 0]$	4.18	7.57	12.02

Table 5.5. Error evaluation for different models of the probability distribution of the auxiliary measurements considering 5 minutes of PVT computation.

Trajectory Sector	Selected Distribution Model
Sector 1	GEV Distr. $[k = 0]$
Sector 2	GEV Distr. $[k = 1]$
Sector 3	GEV Distr. $[k = 0.5]$
Sector 4	GEV Distr. $[k = 0]$
Sector 5	GEV Distr. $[k = 1]$
Sector 6	GEV Distr. $[k = 1]$

Table 5.6. A-posteriori adaptive selection of the distribution models for the likelihood of the auxiliary measurements

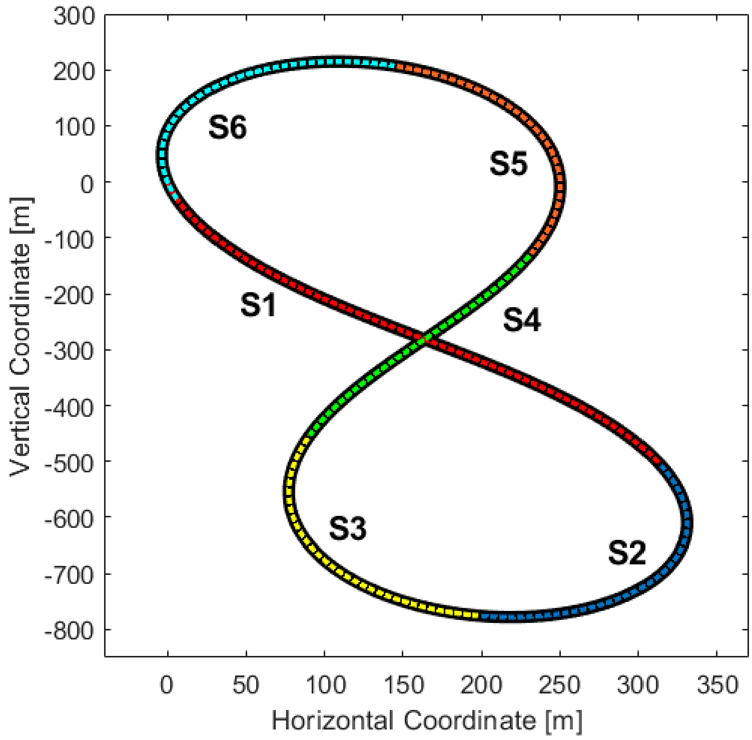

Figure 5.14. Trajectory divided in sectors.

Distribution Model	50-th percentile Error (m)	75-th percentile Error (m)	95-th percentile Error (m)
Gaussian Model	2.31	4.25	8.02
Adaptive GEV Model	1.86	3.51	7.52
Improvement	19.48	17.41	6.23

Table 5.7. Error and improvement evaluation applying a-posteriori adaptive distribution model selection.

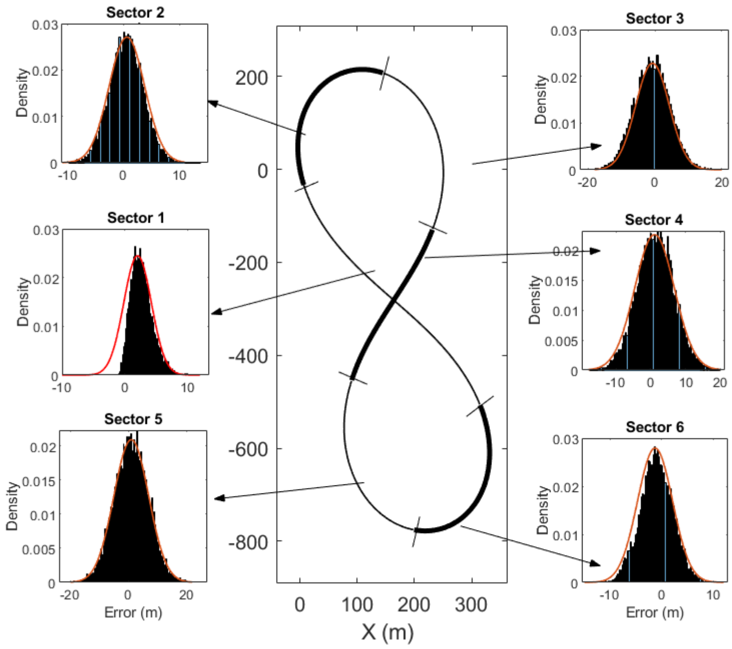

Figure 5.15. Sectorization of the trajectory according to a-priori knowledge of the noise statistical distribution of the inter-receiver distances.

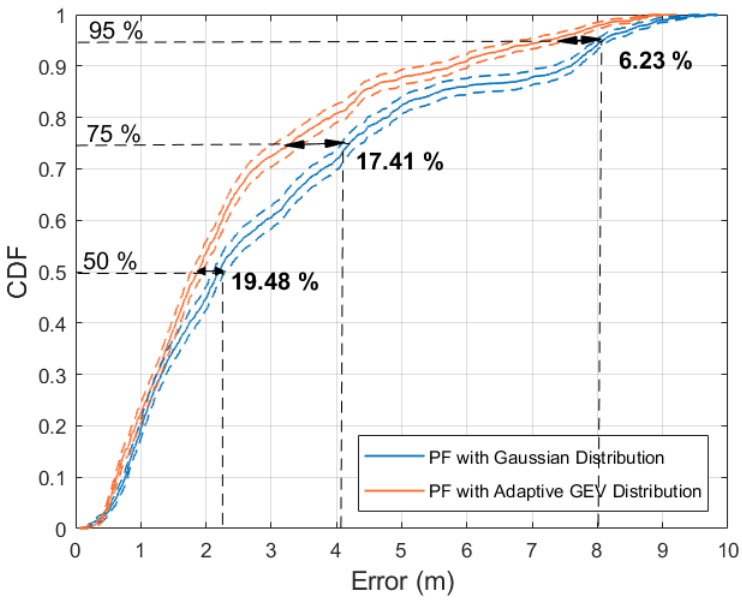

Figure 5.16. ECDF curve of PF with adaptive distribution implementation and s-PF with Gaussian distribution.

Chapter 6

Conclusions

This thesis investigated the possible advantages of exploiting an optimized PF model to integrate GNSS-based auxiliary ranging measurements for Cooperative Positioning in a vehicular scenario.
The evaluation of the performance along the considered trajectory pointed out the limit of the PF in tracking fast state variations in reactive way, when a high number of states has to be estimated, and how a real-time adaptive covariance model for the particle generation can provide a valid support for avoiding high divergency from the actual trajectory.
It was shown how the cooperation does not always work properly when s-PF is considered, due to the forcing of non-Gaussian error distribution of the auxiliary measurements to match Gaussian likelihood function. By implementing an optimized PF model and comparing its performance with s-PF, an enhancement of the accuracy and precision along all the portions of the trajectory was observed, in particular by exploiting GEV distribution with some k parameters for the generation of the likelihood function of the auxiliary measurements.
However, the single GEV distributions, by varing k, provide different performance behaviours depending on the traveled trajectory sector. Thus, an a-posteriori adaptive implementation of the likelihood distribution is performed by choosing sector by sector the best performing GEV distribution by varying k, obtaining a solid model which is able to achieve higher accuracy and precision performance with respect to s-PF, when cooperative ranging measurements show non-Gaussian error distributions, as in the case of weighted double differences ranging.
Furthermore, if a parallel architecture can be implemented to compensate for the increased complexity of the sequential filter, PF can replace other Bayesian estimation filters such as EKF/UKF. Possible future works are expected to focus on a possibile real-time implementation of the dynamic adaptation of the likelihoods related to the auxiliary measurements,

if an estimation of the noise probability distribution can be performed about these measurements. Such estimated distributions can hence be used to dynamically build and use proper likelihood functions within the position estimation routine.

Bibliography

[1] P. Misra, P. Enge, "Global Positioning System: Signals, Measurements, and Performance", Ganga-Jamuna Press, 2011.

[2] M. S. Grewal, L. R. Weill, and A. P. Andrews, "Global Positioning System, Inertial Navigation, and integration". John Wiley & Sons, 2007.

[3] J. Sanz Subirana, JM. Juan Zornoza and M. Hernandez-Pajares, "Global Navigation Satellite Systems: Volume I: Fundamentals and Algorithms".

[4] 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,* vol. 63, no. 6, July 2014, pp. 2544-2556.

[5] D. Yang, F. Zhao, K. Liu, H. Lim, E. Frazzoli, and D. Rus, "A GPS pseudorange based cooperative vehicular distance measurement technique" *in Proc. 75th IEEE VTC-Spring*, 2012, pp. 1-5.

[6] 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*, vol. 20, no. 2, Feb. 2019, pp. 682-691.

[7] M. A. Hossain, I. Elshafiey and A. Al-Sanie, "Cooperative vehicular positioning with VANET in urban environments", *2016 IEEE Asia-Pacific Conference on Applied Electromagnetics (APACE),* Langkawi, 2016, pp. 393-396.

[8] H. Kim, S. W. Choi and S. Kim, "Connectivity Information-Aided Belief Propagation for Cooperative Localization", *in IEEE Wireless Communications Letters,*vol. 7, no. 6, Dec. 2018, pp. 1010-1013.

[9] M. A. Caceres, F. Penna, H. Wymeersch and R. Garello, *"Hybrid GNSS-Terrestrial Cooperative Positioning via Distributed Belief Propagation"*, 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, 2010, pp. 1-5.

[10] F. Shen, J. W. Cheong and A. G. Dempster, "A DSRC Doppler/IMU/GNSS Tightly-coupled Cooperative Positioning Method for Relative Positioning in VANETs", *Journal of Navigation, 70(1)*, pp. 120-136.

[11] Mauricio A. Caceres Duran, Pau Closas, Emanuela Falletti, Carles Fernandez-Prades,

Montse Najar, Francesco Sottile, "Satellite and Terrestrial Radio Positioning Techniques", Chapter 6 - "Signal Processing for Hybridization" Pages 317-382, Editor(s): Davide Dardari, Emanuela Falletti, Marco Luise, *Academic Press, 2012*

[12] de Ponte Muller, Fabian. "Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles." *Sensors (Basel, Switzerland)* vol. 17,2 271. 31 Jan. 2017, doi:10.3390/s17020271

[13] Petovello, M.G.; O'Keefe, K.; Chanv, B.; Spiller, S.; Pedrosa, C.; Xie, P.; Basnayake, C. "Demonstration of Inter-Vehicle UWB Ranging to Augment DGPS for Improved Relative Positioning." *J. Glob. Position. Syst.* 2012, 11, 11–21

[14] A. Conti, D. Dardari and M. Z. Win, "Experimental results on cooperative UWB based positioning systems," *2008 IEEE International Conference on Ultra-Wideband*, Hannover, 2008, pp. 191-195.

[15] A. Minetto, G. Falco and F. Dovis, "On the Trade-Off between Computational Complexity and Collaborative GNSS Hybridization", *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall),* Honolulu, HI, USA, 2019, pp. 1-5.

[16] F. Sottile, H. Wymeersch, M. A. Caceres and M. A. Spirito, "Hybrid GNSS-Terrestrial Cooperative Positioning Based on Particle Filter", *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011,* Houston, TX, USA, 2011, pp. 1-5.

[17] O. Hlinka, F. Hlawatsch and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison", *in IEEE Signal Processing Magazine,* vol. 30, no. 1, pp. 61-81, Jan. 2013.

[18] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", *in IEEE Transactions on Signal Processing,* vol. 50, no. 2, Feb. 2002, pp. 174-188.

[19] F. Gustafsson et al., "Particle filters for positioning, navigation, and tracking", *in IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425-437, Feb. 2002.

[20] P. Kozierski, M. Lis, J. Zietkiewicz, "Resampling in particle filtering-comparison", *Poznanskie Towarzystwo Przyjaciol Nauk,* 2013.

[21] A. Doucet, "On sequential Monte Carlo methods for Bayesian filtering", *Dept. Eng., Univ. Cambridge, UK, Tech. Rep., 1998.*

[22] Navipedia, "GPS Space Segment", URL: https://gssc.esa.int/navipedia/index.php/GPS_Space_Segment

[23] Navipedia, "GPS Ground Segment", URL: https://gssc.esa.int/navipedia/index.php/GPS_Ground_Segment

[24] Navipedia, "GPS Signal Plan", URL: https://gssc.esa.int/navipedia/index.php/GPS_Signal_Plan

[25] M. Tamazin, M. Karaim and A. Noureldin, "GNSSs, Signals, and Receivers", URL: https://www.intechopen.com/

books/multifunctional-operation-and-application-of-gps/gnsss-signals-and-receivers

[26] "Generalized Extreme Value distribution and calculation of Return value" URL: https://gmao.gsfc.nasa.gov/research/subseasonal/atlas/GEV-RV-html/GEV-RV-description.html