2023-08-30

Sequential Importance Resampling Particle Filter for Ambiguity Resolution

Manzano Islas, Roberto Rene

UNIVERSITY OF CALGARY

Sequential Importance Resampling Particle Filter for Ambiguity Resolution

by

Roberto Rene Manzano Islas

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA
AUGUST, 2023

Abstract

In this thesis the sequential importance resampling particle filter for estimating the full geometry-based float solution state vector for Global Navigation Satellite System (GNSS) ambiguity resolution is implemented.The full geometry-based state vector, consisting on position, velocity, acceleration, and float ambiguities, is estimated using a particle filter in RTK mode. In contrast to utilizing multi-frequency and multi-constellation GNSS measurements, this study employed solely L1 GPS code and carrier phase observations. This approach simulates scenarios wherein the signal reception environment is suboptimal and only a restricted number of satellites are visible. However, it should be noted that the methodology outlined in this thesis can be expanded for cases involving multiple frequencies and constellations. The distribution of particles after the resampling step is used to compute an empirical covariance matrix P_k based on the incorporated observations at each epoch. This covariance matrix is then used to transform the distribution using the decorrelating Z transformation of the LAMBDA method [1]. The performance of a float solution based on point mass representation is compared to the typically used extended Kalman filter (EKF) for searching the integer ambiguities using the three common search methods described in [2]: Integer Rounding, Integer Bootstrapping, and Integer Least Squares with and without the Z transformation. As Bayesian estimators are able to include highly non-linear elements and accurately describe non-Gaussian posterior densities, the particle filter outperforms the EKF when a constraint leading to highly non-Gaussian distributions is added to the estimator. Such is the case of the map-aiding constraint, which integrates digital road maps with GPS observations to compute a more accurate position state. The comparison between the position accuracy of the particle filter solution with and without the map-aiding constraint to the solution estimated with the EKF is made. The algorithm is tested in different segments of data and show how the position convergence improves when adding digital road map information within the first thirty seconds of initializing the Particle Filter in different scenarios that include driving in a straight line, turning, and changing lanes. The assessment of the effect of the map-aiding algorithm on the ambiguity domain is carried out as well and it is shown how the convergence time of the float ambiguities improves when the position accuracy is improved by the constraint. The particle filter is able to weight the measurements according to any kind of distribution, unlike the EKF which always assumes a Gaussian distribution. The performance of the PF when having non-Gaussian measurements is assessed, such as when the measurements are distorted by multipath. Two additional steps are implemented, an outlier detection technique based on the predicted set of particles, and the use of a mixture of Gaussians to weight the measurements detected as outliers. The implemented outlier detection algorithm is based on the residual (or innovation) testing technique which is commonly applied into the EKF. The innovation and its covariance matrix are estimated from a predicted set of residuals using

the transitional prior distribution and the measurement model. Then, the innovation is compared against the critical value of $\mathcal{N}(0,1)$ at a level of significance α. The mixture of Gaussians is the weighted sum of two Gaussians, one from the measurement noise matrix, and the second being a scaled version of the first one describing the multipath error. This procedure de-weights the measurements with multipath, and reduces the bias in the position estimate. The proposed map-aiding algorithm improves the ambiguity convergence time by approximately 80%, while the deweighting process enhances it by around 25% for the segments of the vehicle dataset that were analyzed. This work serves as a demonstration of cases wherein the particle filter addresses the limitations of the EKF in estimating the float solution in ambiguity resolution. Such limitations include constraints that give rise to non-Gaussian probability density functions and the utilization of a distinct likelihood function for outlier measurements, as opposed to the Gaussian assumption made by the EKF. The proposed map-aided particle filter can be implemented in real-time to enhance the float ambiguity during the initial epochs after the filter has been initialized. This implementation proves beneficial in urban environments where there is a loss or complete obstruction of the GNSS signal.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Kyle O'Keefe, for his support, mentorship, and guidance throughout my graduate studies. I am truly grateful for all the time and effort you have dedicated to teaching and advising me. Your support in various aspects of my academic journey at the University of Calgary is deeply appreciated. I hold great admiration and respect for you, and I am thankful for all your contributions. I would also like to extend my thanks to my friend Kelly, as well as all my colleagues in the PLAN group, for their support during my studies. Your support and encouragement have made my academic experience more fulfilling.

To the person I love most in my life, my mother, I am forever grateful for your boundless love and support. Your love and belief in me has been a constant source of inspiration. I would also like to express my gratitude to my brother for his love and trust.

I am deeply grateful to my aunt Gabriela and uncle Rene for their support, unconditional love, and invaluable advice throughout my whole life. Special thanks to my grandparents Arcelia and Joel, who have always encouraged me to pursue my studies and professional growth while emphasizing the importance of being a good person.

I would like to acknowledge my undergraduate thesis supervisor, Bartolo Lara, who pursued his M.Sc. at the University of New Brunswick. His discussions about studying in Canada and sharing his knowledge with me were a great inspiration. He introduced me to GNSS topics and planted the seed for the idea of pursuing graduate studies in Canada.

I am also grateful to the Mexican government dependency CONACYT for providing financial support during my graduate studies. Their assistance has been crucial in enabling me to pursue my graduate degree.

I dedicate this thesis to my family: my mother, Ariadne; my brother, Roberto; my grandparents, Arcelia and Joel, who are watching over us from above; my aunt, Gabriela; and my uncle, Rene. I love you all so much.

Table of Contents

Abstract ii

Acknowledgements iv

Dedication v

Table of Contents viii

List of Figures xi

List of Tables xii

List of Symbols xiv

1 Introduction 1
- 1.1 Problem statement . . . 1
  - 1.1.1 Particle Filter in Ambiguity Resolution . . . 3
  - 1.1.2 Particle Filter and map-matching constraint in GNSS . . . 4
  - 1.1.3 GNSS in urban environments . . . 5
  - 1.1.4 Limitations of Previous Research . . . 6
- 1.2 Objectives . . . 6
- 1.3 Contributions . . . 7
- 1.4 Thesis Outline . . . 8

2 Ambiguity Resolution 10
- 2.1 Observation model . . . 10
- 2.2 Differential positioning . . . 11
- 2.3 Ambiguity Resolution . . . 12
  - 2.3.1 Integer bootstrapping . . . 14
  - 2.3.2 Integer least squares . . . 15
  - 2.3.3 LAMBDA . . . 17
  - 2.3.4 Ratio test . . . 19
  - 2.3.5 Success rates . . . 20

3 Nonlinear Filters 22
- 3.1 Sequential Least-Squares . . . 22
- 3.2 Extended Kalman Filter . . . 25
- 3.3 Unscented Kalman Filter . . . 27
- 3.4 Sequential Importance Resampling Particle Filter . . . 30
  - 3.4.1 Estimation problem in Bayesian framework . . . 30
  - 3.4.2 Minimum mean-square error estimate . . . 31
  - 3.4.3 Importance Sampling . . . 32
  - 3.4.4 Degeneracy Problem . . . 35
  - 3.4.5 Choice of importance density . . . 36

3.4.6 SIR Particle Filter with Local Linearization . . . 40
3.4.7 Unscented Based Particle Filter . . . 41
3.5 Comparison between non-linear filters . . . 42
3.6 Rao-blackwellization . . . 47
3.6.1 Rao-Blackwellized PF example . . . 49

4 SIR Particle Filter in float solution for ambiguity resolution 53
4.1 SIR PF implementation details . . . 53
4.1.1 Filter initialization . . . 55
4.1.2 Prediction . . . 55
4.1.3 Weighting . . . 56
4.1.4 Resampling . . . 58
4.1.5 Empirical variance-covariance matrix . . . 58
4.2 Data and Software . . . 59
4.2.1 Driving Dataset . . . 59
4.2.2 Software for data processing . . . 61
4.3 Results . . . 62
4.3.1 Probability of correct fix . . . 66
4.3.2 Computation load . . . 70

5 Map-aided Particle Filter for ambiguity resolution 72
5.1 Map-matching . . . 73
5.2 Digital Maps . . . 73
5.2.1 Open Street Map . . . 73
5.2.2 Software . . . 75
5.3 Map-aiding Algorithm . . . 76
5.3.1 Position domain . . . 78
5.3.2 Convergence of the Ambiguities . . . 86
5.4 Map-constrained EKF . . . 94

6 Non-Gaussian measurements 99
6.1 Multipath . . . 100
6.1.1 Signal and Measurement Quality Monitoring . . . 101
6.1.2 Code Minus Carrier . . . 102
6.2 Particle Filter in the presence of multipath . . . 105
6.2.1 Mixture of Gaussians . . . 107
6.3 Multipath environment data . . . 108
6.3.1 Robust Particle Filter . . . 110
6.3.2 Results . . . 114

7 Conclusions and recommendations for future work 121
7.1 Conclusions . . . 121
7.1.1 Nonlinear Filters . . . 121
7.1.2 SIR Particle Filter in Float Solution for Ambiguity Resolution . . . 122
7.1.3 Map-aided Particle Filter for Ambiguity Resolution . . . 123
7.1.4 Non-Gaussian Measurements . . . 124
7.1.5 Recommendations for Future Work . . . 125

References 126

A RTKLIB GNSS software 135
A.1 RNX2RTKP . . . 136

B Coordinate systems transformation **138**
B.0.1 Coordinate Systems 138
B.0.2 Transformation Equations 139

C Cholesky decomposition **142**

D Double Difference Transformation Matrix **144**

E Non-linear measurement models and non-Gaussian distributions **146**
E.1 Non-linear models 146
E.2 Non-Gaussian Distributions 149

List of Figures

2.1 Integer rounding two-dimensional pull-in regions. 13
2.2 Bootstrapping two-dimensional pull-in regions. 15
2.3 ILS two-dimensional pull-in regions. 16
2.4 Integer search space before decorrelation 18
2.5 Integer search space after decorrelation 19

3.1 Schematic description of the resampling process. Blue bars represent the particles with significantly larger weights than the rest, increasing their chance to be multiplied. 36
3.2 One-dimension estimation case. The *vehicle* (red triangle) is static, and the observations are ranges from the *source* (black circle) to the *vehicle*. 43
3.3 Comparison between the KF and UKF solutions at the first epoch of a one-dimension example. 44
3.4 Comparison between the KF and Particle filter solutions at the first epoch of a one-dimension example. 50 particles are used in this PF. The black line fits the updated particle histogram for the comparison between both solutions. 45
3.5 Comparison between the KF and Particle filter solutions at the first epoch of a one-dimension example. 5000 particles are used in this PF. The black line fits the updated particle histogram for the comparison between both solutions. 45
3.6 Comparison between the solutions estimated with the KF and Particle filter with local linearization using EKF at the first epoch of a one-dimension example. 500 particles are used in this filter. 46
3.7 Comparison between the solutions estimated with the KF and the Particle filter with the importance density as the transitional prior moved to the likelihood function using the Kalman gain from the UKF. 500 particles are used in this PF. 47
3.8 2-Dimension positioning example. 50
3.9 Comparison between the position and velocity states estimated with the SIR PF and the RBPF 52

4.1 Senors used in the data set collection. Photos from [3] 60
4.2 Drive trajectory of C. Huang dataset 61
4.3 Analyzed trajectory (red line). 63
4.4 Position state errors in ENU directions computed with EKF. 64
4.5 Position state errors in ENU directions computed with SIR particle filter. 64
4.6 Float-valued ambiguities differences with respect their integer values using the EKF. The shaded areas are the 2σ standard deviation of the DD float ambiguities. 65
4.7 Float-valued ambiguities differences with respect their integer values using the particle filter. The shaded areas are the 2σ standard deviation of the DD float ambiguities. 66
4.8 Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the EKF. The correct fixing for ILS (green), bootstrapping (light blue) and rounding (orange), 1 in case of correct fixing and 0 otherwise. 67
4.9 Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the SIR Particle Filter. Success rates of rounding (orange), bootstrapping (light blue) and ILS (green). 68
4.10 Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the EKF after LAMBDA decorrelation. The correct fixing for ILS (green), bootstrapping (light blue) and rounding (orange), 1 in case of correct fixing and 0 otherwise. 69

4.11 Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the SIR particle filter after LAMBDA decorrelation. Success rates of rounding (orange), bootstrapping (light blue) and ILS (green). 69
4.12 Processing time of Integer Least Squares applied to a set of particles before (red) and after (blue) decorrelation. 70

5.1 OSM original road map. 75
5.2 Example of a segment of digital road map modified for the purposes of this thesis. 75
5.3 Java OSM software interface. 76
5.4 Plan view of segments analyzed. Each segment describes a different vehicle maneuver. . . . 79
5.5 Estimated position using the SIR Particle Filter without map-aiding. 80
5.6 Estimated position using the SIR Particle Filter with map-aiding using the original digital map from OSM. 81
5.7 Estimated position state with map-aiding using the SIR Particle Filter. 82
5.8 Particles distribution of the estimated latitude positions without map-aiding for the second epoch of the *Turn* case shown in Figure 5.5. The EKF estimated position is plotted in green, described by a Gaussian distribution ($\mathcal{N}(\mu, \sigma^2)$) using the estimated position $\hat{x}_k$, and its covariance matrix $\hat{P}_k$ 83
5.9 Particles distribution of the estimated latitude positions with map-aiding for the second epoch of the *Turn* case. 84
5.10 Estimated position with the map-aiding implemented when the vehicle is not centred in a lane. 85
5.11 Single-Difference ambiguities without map-aiding. Histograms are shown during a 30 second time interval every 5 seconds for better although the filter updates every second second. Each distribution is Gaussian as no constraint is applied. 88
5.12 Single-Difference ambiguities estimated with the PF with map-aiding. The first histogram is no longer Gaussian, as the map-aiding algorithm results in a multimodal pdf. 89
5.13 Across-track position and float ambiguity distributions and histograms without map-aiding. . 90
5.14 Across-track position and float ambiguity distributions and histograms with map-aiding. . . . 91
5.15 Probability of correct rounding (orange) and bootstrapping (light blue) using the PF. Success rates of rounding (red), bootstrapping (blue) and ILS (green). 92
5.16 Probability of correct rounding (orange) and bootstrapping (light blue) and success rates of rounding (red), bootstrapping (blue) and ILS (green) using the PF with the map-aiding algorithm 93
5.17 Estimated position with the map-aiding EKF. 95
5.18 EKF distributions with and without the map-constraint for the second epoch of the *Turn* case. 96
5.19 Single-Difference ambiguities estimated wit the map-constrained. 97

6.1 2-dimension schematic example of NLOS and LOS multipath cases. 100
6.2 CMC metric of segments of data with measurements distorted by multipath. 102
6.3 Satellites skyplot. 103
6.4 Measurements without multipath. The fitted Gaussian (green curve) has a coefficient of determination R^2 of .96 104
6.5 Measurements with multipath error. The fitted Gaussian has a coefficient of determination of $R^2 = .78$ 104
6.6 Gaussian mixture noise. 108
6.7 Urban canyon environment example from satellite 6.7a and terrestrial 6.7b view. 109
6.8 Plan view of the segment with multipath analyzed in this chapter. 115
6.9 Fish-eye image of the analyzed segment. 116
6.10 ENU errors of the float solution obtained with the PF 117
6.11 ENU errors with the outlier detection and de-weighting process implemented on the PF. . . 117
6.12 Probability of correct rounding (orange) and bootstrapping (light blue) and success rates of rounding (red), bootstrapping (blue) and ILS using the SIR PF 119

A.1 RTPOST GUI application for GNSS post-processing analysis 137

A.2 RNX2RTKP CUI application for GNSS post-processing analysis 137

B.1 Local navigation system (East North Up system). 139

E.1 Linear function. Geometric distance ρ and x position state vector rate change. Due to the large distance between the satellite and the receiver, the function is almost linear. R^2 statistic is close to 1. 147
E.2 Nonlinear function. Angle Θ and $cos(\Theta)$ values. R^2 statistic is .85 depicting less *linearity*. . 148
E.3 Nonlinear function. Angle Θ and $cos(\Theta)$ values. R^2 statistic is .85 depicting less *linearity*. . 149
E.4 Gaussian distribution of code measurement errors without multipath effects. 150
E.5 Example of non-Gaussian distribution for code measurements distorted by mulitpath 151

List of Tables

4.1 Sensor Specification of the Data Collection Platform [3]. 59
4.2 RTKLIB input settings used in this work. 62

5.1 Estimated cross-track (CT) error (m) and variance σ^2 using the EKF, and the SIR Particle Filter with and without the map-aiding algorithm at 5, 15, and 30 seconds after the filter initialization for the different segments of data shown in Figure 5.4. The background colors of the first column match those shown in Figure 5.4. 86
5.2 Estimated probabilities of correct fixing of bootstrapping (PB) and empirical success rates of the ILS using the SIR Particle Filter with and without the map-aiding algorithm at 5, 15, and 30 seconds after the filter initialization for the different segments of data shown in Figure 5.4. The background colors of the first column match those shown in Figure 5.4. 94

A.1 Main RTKLIB applications. 136

List of Abbreviations

Symbol	Definition
cdf	Cumulative density function
EKF	Extended Kalman Filter
EKPF	Extended Kalman Particle Filter
ENU	East North Up
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
INS	Inertial Navigation System
MEMS	Micro-Electro-Mechanical System
MMSE	Minimum mean square error
MC	Monte Carlo
PF	Particle Filter
pdf	Probability density function
pmf	Probability mass function
PPP	Precise Point Positioning
RBPF	Rao-Blackwellized Particle Filter
SIR PF	Sequential Importance Resampling Particle Filter
SIS	Sequential Importance Sampling
SLS	Sequential Least Squares
SQM	Signal Quality Monitoring
MQM	Measurement Quality Monitoring
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
UT	Unscented Transform

List of Symbols

a	Semi-major axis
b	Semi-minor axis
$\check{a}$	Float ambiguity estimate
e	Eccentricity
f	Frequency
$f(x)$	Function of x
G	Coefficient matrix of random forcing function
D_k	Double-difference transformation matrix
H	Connection matrix between the measurement and the state
K	Kalman Gain
$\bar{K}$	Lagrange multiplier
L	Lower-triangular matrix from Cholesky decomposition
$\hat{P}$	Covariance matrix of the state
P_1, P_2, P_3	Reflection Matrices
$p(x_k\|y_k)$	Density function of state x given observations y at time k
$p(x_k\|x_{k-1})$	Transitional density
$p(y_k\|x_k)$	Likelihood function
$\mathbf{Q}_k$	Covariance matrix of the process noise
$q(x)_k$	Importance density
R_k	Covariance matrix of the measurement noise
R_1, R_2, R_3	Rotation Matrices
S_z	Pull-in region of vector z
x_k	State vector at time k
$\hat{x}_k^-$	Prior estimate of the state

$\hat{x}_k$	Estimate of the state
y_k	Measurement vector at time k
$\Delta\nabla$	Double Difference operator
δ	Dirac's Delta
λ	Longitude
Φ	Transition matrix
ϕ	Latitude
λ	Signal Wavelength
$\boldsymbol{\mathcal{Y}}$	Sigma points in the measurement domain
$\boldsymbol{\chi}_i$	ith sigma point for Unscented Kalman Filter

Chapter 1

Introduction

Global Navigation Satellite Systems (GNSS) are a network of satellites that enables positioning, navigation, and timing services worldwide. It encompasses various satellite constellations, such as GPS, GLONASS, Galileo and BeiDou. GNSS provides service to an unlimited number of users, providing accurate, continuous, worlwide, 3-dimensional positioning and veolicty information to users with the receiver equipment [4]. GNSS uses are varied, ranging from personal navigation devices to commercial transportation, agriculture, surveying and military purposes. Its importance for society lies in its ability to provide accurate positioning and timing information, facilitating efficient transportation and logistics. However, GNSS does have some limitations, including vulnerability to signal interference, limited accuracy in certain environments (dense urban areas), and susceptibility to atmospheric effects. Ongoing research aims to address these limitations and improve GNSS performance, focusing on areas such as signal processing, alternative positioning techniques which may use muti-sensor fusion and the implementation of different estimators.

1.1 Problem statement

In order to achieve centimetre-level precision with Global Navigation Satellite Systems, the integer number of cycles of the carrier wave needs to be estimated.

Many algorithms have been developed to address the ambiguity resolution problem. In [5], a threefold classification of different methods is made based on how they make use of the received measurements:

1. The first class uses C/A or P-code pseudoranges directly to determine the ambiguities of the corresponding carrier-phase observations.

2. The second class comprises the first ambiguity resolution method developed, the Ambiguity Function

Method [6]. This technique uses the fractional value of the instantaneous carrier-phase measurement, so the ambiguity function values are not affected by cycle slips.

3. The third class comprises a group of techniques that are based on the theory of ILS. Some of these techniques include the Least Squares Ambiguity Search Technique (LSAST) [7], the Fast Ambiguity Resolution Approach (FARA) [8], the modified Cholesky decomposition method [9], the null space method [10], the Fast Ambiguity Search Filter (FASF) [11], and the Optimal Method for Estimating GPS Ambiguities (OMEGA) [12].

In [13], the main characteristics of ambiguity resolution techniques in the third classification are discussed. The ambiguity resolution method presented in this thesis is similar to those in the third classification in that a float solution is estimated as input to the integer ambiguity resolution process. Generally, the ambiguity resolution process is divided into three main steps [14]:

1. Estimating a real-valued or float solution for the ambiguities, along with its covariance matrix at each epoch, which will be used for searching the correct integer values.

2. Mapping the ambiguity float parameters to integer values in the integer ambiguity estimation step.

3. Fixing the solution by successfully constraining the ambiguities to integers and adjusting the remaining unknown parameters based on their correlation with the ambiguities.

Most of the existing algorithms for ambiguity resolution utilize the ILS method [15] to determine the correct integer value from a set of integer candidates [13]. These algorithms rely on a previously computed float solution, typically derived through an EKF. However, ambiguity resolution becomes more challenging in environments with difficult signal reception conditions, such as urban canyons [16]. Consequently, achieving a faster convergence of float ambiguities is required. Certain ambiguity resolution algorithms leverage additional constraints or integrate sensor measurements to enhance the float solution [16, 17, 18]. Nevertheless, the inclusion of constraints or sensor measurements may lead to posterior distributions that are not Gaussian or require the use of non-linear measurement models. Describing such distributions and deriving such models becomes complex when considering the framework of the EKF. This estimator has limitations in dealing with highly nonlinear measurement models and highly non-Gaussian posterior densities (see Appendix E). This is especially the case when the receiver is in urban environments and multipath signals are a significant source of error. While typical GNSS observations are easily linearized about a point of expansion, additional sensor observations and constraints, such as short-range constraints or map-matching techniques, are not easily integrated using least-squares or EKF approaches. An alternative solution is to use an estimator that

addresses both non-linear observation models and non-Gaussian errors [19]. Sampling or Bayesian estimators are able to include highly non-linear elements and accurately describe non-Gaussian distributions. In this thesis, the Sequential Importance Resampling Particle Filter (SIR PF) is used as an alternative method to estimate the full geometry-based float solution state vector. The SIR PF is a sequential Monte Carlo method based on the point mass representation of probability densities [20]. The PF represents the a priori and posteriori probability density functions with the help of many samples in the state space. Therefore, the SIR PF outperforms the EKF when the system is highly nonlinear because it makes no assumptions about the shape of the observation or state probability density functions and does not require a linearization step.

1.1.1 Particle Filter in Ambiguity Resolution

The PF has been previously proposed for satellite navigation and positioning; however, it is not commonly applied to ambiguity estimation and resolution because one of the purposes of ambiguity resolution is to find the correct solution in the shortest possible time [1]. The main limitation of the PF is its computational cost, and therefore, it is generally used to estimate small state vectors or combined with another estimator. In [21], a Bayesian approach for ambiguity resolution is presented. This approach does not resolve ambiguities in the traditional way but instead sums over all possible ambiguities with proper weights directly derived from the likelihood function, considering the confidence region around the float estimates.

A PF-based method for ambiguity resolution is proposed in [22]. In this work, the integer ambiguity is treated as a random integer vector, and particles are generated according to the initial probability density of the rover position given the double difference measurement for the pseudorange observable. After incorporating the measurements and applying Bayes' rule, the probability mass function (pmf) evolves, and new particles are generated. However, since the GPS receiver is not considered in the PF, the method itself cannot be used for real-time applications without complementing it with other algorithms.

An adaptive Kalman filter-based method for ambiguity resolution is proposed in [23]. This work only focuses on determining the correct sets of possible ambiguities while using all data to improve the overall position. It consists of a blending of solutions from multiple filters, each of which hypothesizes a different ambiguity set. This method is divided into three steps: the first one uses an EKF to calculate the user's position, velocity, and carrier phase ambiguities estimates. The second step generates the candidate ambiguity sets using the Z transformation [2] from the Least Squares Ambiguity Decorrelation Adjustment (LAMBDA) method. The third step provides candidate ambiguity sets to a bank of Kalman filters in the Multiple-Model

Adaptive Estimation.

In [24], samples are drawn directly from the three-dimensional position space, and then integers consistent with the particle filter are constructed, allowing the estimates to be insensitive to cycle slips. However, this method does not follow the traditional steps of first estimating a float solution and then finding an integer one.

In [25], a nonlinear filter is developed to estimate the user position and all unknown parameters, such as integer ambiguities and ionospheric errors, using multi-frequency and multi-GNSS. This procedure is based on four steps: 1) computing the float solution at each epoch adaptively to the dynamic environment using the EKF and the particle filter, 2) using a carrier phase multipath indicator to derive a constraint that rejects integer candidates affected by multipath errors, 3) applying LAMBDA to search for integer ambiguities, and 4) validating the fixed solution using a statistical test.

A method for ambiguity resolution based on a mixture Kalman filter solution is proposed in [26]. The optimal proposal density for the ambiguities is leveraged, and the set of ambiguities is marginalized by employing a particle filter. Then, possible values are bounded to a dense set of integers, allowing the extraction of the integer solution as a maximum likelihood estimate from a mixture Kalman filter, where each Kalman filter is conditioned on a fixed ambiguity value for each satellite.

In [27], a PF is implemented to estimate the user position using pseudorange observations only, while Non-Line-of-Sight multipath signals are determined by a likelihood estimation that compares the position hypotheses of the receiver computed from the pseudoranges and the particles.

1.1.2 Particle Filter and map-matching constraint in GNSS

The PF is more suitable for incorporating constraints or sensor measurements as these additional observations may lead to a highly nonlinear measurement model or a highly non-Gaussian posterior distribution [28].

An approach to PF-based map matching is presented in [19]. This solution is given within the framework of Bayesian theory. The basic principle is to use particles to represent the posterior density of the vehicle in a dynamic state estimation framework where road map information is integrated. This is a probabilistic approach to map matching, which calculates the probabilities of the vehicle traveling on different road segments to select the correct one and then estimates the vehicle's position on the selected road segment. The

PF considers multiple candidate roads and is able to adapt if an initial guess at the proper road is found to be incorrect. The authors demonstrate in this work how map-matching using a nonlinear filter, such as the PF, can improve the car navigation system's position accuracy. This becomes very helpful in urban environments where GPS reception deteriorates due to obstacles and multipath.

As shown in [29], incorporating digital map information within the conventional EKF estimation method is not easy because this constraint leads to highly non-Gaussian posterior densities. The PF approach is an excellent option for map constraint since it presents no restrictions regarding non-Gaussian distributions and the nonlinearity of the system models.

1.1.3 GNSS in urban environments

Location information in urban canyons is needed for many applications such as land navigation, transportation efficiency, and vehicle or pedestrian localization. The most popular solution is the use of GNSS because they provide continuous coverage, and all-weather, worldwide information. However, this is still a challenge because GNSS signals can be easily blocked or disturbed by obstacles such as buildings or dense foliage. These features introduce multipath errors, which arise from the reception of reflected or diffracted signals [30]. Hence, standalone GNSS receiver architectures cannot provide suitable accuracy in an urban canyon scenario. To face this problem, many authors have considered aiding GPS with other sensor data, which could be Dead Reckoning, Inertial Navigation Systems (INS), and digital map information.

In [31] a PF-based method is proposed to integrate Micro-Electro-Mechanical Systems (MEMS) sensors with GPS for reliable navigation solutions in urban environments. The authors introduce a Mixture PF algorithm for integrating the Radio Inertial Satellite System (RISS) with GPS observations and handle non-Gaussian errors effectively.

In [32], an integration of Wi-Fi and GPS is presented. This solution comprises an integrated system that mainly depends on GPS positioning improved by a Kalman Filter. If GPS signal reception deteriorates, the estimation is assisted by Wi-Fi positioning and a PF.

An approach that uses high-sensitivity receivers in order to receive a degraded signal by an urban environment is proposed in [33]. However, since measurements with this kind of receiver are not always reliable and can produce large errors, Dead Reckoning is integrated into this navigation system.

In [34] a navigation system for precise urban vehicular positioning is presented. This proposed method is unaided by inertial or electro-optical sensors, instead, it uses an embeddable multi-frequency software-defined GNSS receiver which draws ephemeris data and GPS data bit estimates from a set of 8 reference GNSS stations in the city of Austin. This receiver feeds code and carrier observables to an RTK engine which produces a precise solution and, depending on an integer aperture test, it can be fixed or float.

1.1.4 Limitations of Previous Research

Particle filters have been used for state estimation since the early 1990s, and researchers have since explored various aspects of this approach. The particle filters are useful for tracking and navigation problems involving highly non-linear and non-Gaussian distributions. However, their main drawback is their high computational cost, which increases with the dimensionality of the state vector. Ambiguity resolution methods, on the other hand, prioritize efficiency and speed in resolving ambiguities, which is a limitation when using particle filters. As a result, researchers often combine particle filters with other estimators to estimate certain variables only. There are many unexplored aspects of using particle filters for ambiguity resolution, such as the behavior of particles in a float solution under different signal reception conditions or the effect of constraints, such as a map-matching constraint, on the filter.

Recent advances in hardware and software have increased computational capacity, allowing for more possibilities in the use of particle filters for ambiguity resolution. By computing a float solution with the SIR PF, many possible solutions or particles can be obtained instead of just one.

1.2 Objectives

An alternative solution was adopted in this research for estimating the float solution with an additonal map-aiding algorithm and an outlier detection algorithm for deweighting the measurements distorted by multipath: the Sequential Importance Resampling Particle Filter. This method can handle both non-linear observation models and non-Gaussian errors.

While the particle filter is not commonly applied to ambiguity resolution in GNSS positioning, as one purpose of ambiguity resolution is to find the correct solution in the shortest possible time, the main objectives of this thesis are the following:

1. Assess the performance of the particle filter when estimating the full geometry-based float solution for

a double-differenced short baseline.

2. Compare the float solution estimated with the PF to the typically used EKF for searching the integer ambiguities using the three common search methods: Integer Rounding, Bootstrapping, and ILS, with and without an application of LAMBDA decorrelation. These integer search methods are used to compare their probabilities of correct fixing and the empirically success rates obtained with the PF.

3. Demonstrate the particle filter's ability to describe arbitrary posterior distributions, beyond just Gaussian distributions, by implementing a map-matching algorithm. The effect of the map-constraint into the float solution is assessed in the position and ambiguity domains showing how much the position accuracy and the convergence of the float ambiguities are improved for faster ambiguity resolution.

4. Assess the performance of the PF for estimating the float solution when dealing with non-Gaussian errors, such as the case when signals are a significant source of error.

1.3 Contributions

The contributions of this thesis are outlined in the following list:

1. A detailed description of the most common non-linear estimators is provided. The explanation provided in this thesis follows a sequence where the simplest and most common non-linear estimator, the EKF, is first introduced, followed by the unscented Kalman filter (UKF) which provides a second-order Taylor series approximation to non-linear problems. Finally, we explain in detail each step of the Particle Filter, describing its main advantages and limitations for the state estimation problem and explicitly demonstrate the necessary conditions for the PF and the EKF to provide equivalent results. Throughout this explanation, simulated examples are carried out to facilitate the comparison of these nonlinear estimators, enabling straightforward analysis of the solutions generated by these.

2. A detailed description of the implementation of the SIR PF applied to the estimation of the full geometry-based float solution is presented. This has never been done before as the PF is commonly used to estimate small-dimension state vectors or combined with other estimators such as the EKF.

3. A full comparison of the available options for ambiguity resolution when having a float solution based on mass representation is made. The three integer search methods presented in [35] are compared using the EKF and the PF. An empirical probability of the correct fix of the three estimators is estimated as an alternative way to measure its success rate adn is compared with the formally estimated bound first shown by [36].

4. The map-matching constraint for improving the accuracy of the position state and the convergence of the real-valued ambiguities is analyzed. The ability of the PF to capture multi-modal distributions is shown by adding digital road map information to the estimation.

5. This work shows the ability of the PF to weight the particles according to a non-Gaussian distribution for the likelihood function. The PF is applied to estimate the float solution under an urban environment, where there is presence of multipath error and the measurements are not longer Gaussian. A mixture of Gaussians is used for the likelihood function, on measurements detected as outliers.

1.4 Thesis Outline

The structure of the thesis is as follows:

- Chapter 2 provides an explanation of background information about GNSS differential positioning and ambiguity resolution. It presents the code and phase measurement models for positioning, which are the basis for double-difference positioning. A description of the three admissible estimators presented in [35] (integer rounding, bootstrapping, and ILS) and the LAMBDA method [2] is provided.

- Chapter 3 is a full description of the Particle Filter. Prior to explaining the Particle Filter process and features, descriptions of the EKF and UKF are conducted. The local linearization and the Rao-Blackwellization are explained as alternatives to improve the estimation or reduce the number of particles needed. A simple case of state estimation is presented to show the similarities between the PF and the EKF.

- Chapter 4 presents the implementation of the SIR PF to estimate the full geometry-based solution using L1 GPS carrier phase and pseudorange observations. A comparison between the solutions estimated with the EKF and the PF is conducted using a kinematic land-vehicle data set described in [3].

- Chapter 5 provides the implementation of a map-matching algorithm for the estimation of the float solution using the SIR PF. This chapter analyzes the effect of implementing digital road map information in the position state estimate and in the convergence of the float ambiguities for a faster ambiguity fix. The algorithm is applied to different driving scenarios of the data set described in Chapter 4.

- Chapter 6 analyzes the effect of weighting the particles according to a distribution other than Gaussian for the likelihood function. The results were obtained from the estimation of the float solution for the segments of data where measurements are likely to be distorted by multipath. The outlier detection

step implemented is described, as well as the mixture of Gaussians distribution used as the likelihood function.

- Chapter 7 provides a discussion of the conclusions for each chapter of this thesis followed by suggestions for future work.

Chapter 2

Ambiguity Resolution

This chapter provides the theoretical background of the ambiguity resolution process, emphasizing the description of each of the three admissible integer estimators [15]. The chapter begins by introducing the observation model for code and carrier phase, followed by the single and double difference model. Then, it describes the ambiguity resolution methods, including Integer Least Squares, Bootstrapping, and Integer Rounding. Additionally, it introduces the Least Squares Ambiguity Decorrelation Method [1]. Finally, an explanation about the ratio test for validating the estimated integer ambiguity, as well as the success rate for bootstrapping and rounding is provided.

2.1 Observation model

In GNSS, the distance between the satellite and the receiver's antenna is measured by the pseudorange [37]. This refers to the distance between the satellite at the time of signal emission and the receiver at the time of signal reception. The pseudorange is different from the geometric distance because of the influence of the signal transmission mediums and the clock errors. The precision of this measurement has been traditionally about 1% of the chip length, thus, the C/A code has a precision of about 3 m, while the P-code is 30cm. The carrier phase is a measure of the phase of the received satellite signal relative to the receiver-generated carrier phase at the reception time [38]. The number of full carrier waves between satellite-receiver is not accounted for at the initial signal reception, hence, the carrier phase is a measurement of the fractional phase. The carrier phase is measured with a precision better than 1% of the wavelength, this means millimetre precision. The ambiguous integer number of cycles in the carrier phase measurement is called ambiguity. Therefore in order to use the carrier phase for precise positioning, the integer ambiguity must be determined.

The GNSS code pseudorange and carrier phase are formulated as:

$$P_r^i = \rho_r^i - (\delta t_r - \delta t_i)c + \delta_{ion} + \delta_{trop} + \varepsilon_c \tag{2.1}$$

and

$$\Phi_r^i = \rho_r^i - (\delta t_r - \delta t_i)c + \lambda N_r^i - \delta_{ion} + \delta_{trop} + \varepsilon_p \tag{2.2}$$

where P_r^i and Φ_r^i are the observed pseudorange and carrier phase between the satellite i and the receiver r respectively, δ_{ion} and δ_{trop} are the ionospheric and tropospheric delays respectively, δt_r and δt_i are the receiver and satellite clock error terms respectively and N is the ambiguity in cycles. In order to use the carrier phase for precise positioning, the integer ambiguity N must be determined.

Phase range model

GNSS high precision is possible by using carrier phase measurements from receivers. Pseudoranges can be obtained from phase measurements by noting that they differ only by the phase ambiguity and the sign of the ionospheric effect. The mathematical model for these measurements is given by [37]:

$$\Phi_r^i = \frac{1}{\lambda^i}\rho_r^i + N_r^i + \frac{c}{\lambda^i}\Delta\delta t_r^i \tag{2.3}$$

where ρ is the geometric distance between the observing point r and the satellite i, and can explicitly be written as:

$$\rho_r^i = \sqrt{(X^i - X_r)^2 + (Y^i - Y_r)^2 + (Z^i - Z_r)^2} \tag{2.4}$$

$\Phi_r^i(t)$ is the measured carrier phase expressed in cycles , λ^i is the wavelength, c denotes the speed of light and $\Delta\delta t_r^i$ is the combined receiver and satellite clock bias ($\delta t_r - \delta t^i$). N_r^i is the time-independent phase ambiguity number. Considering that $f^i = c/\lambda^i$, equation (2.4) can be rewritten as:

$$\Phi_r^i + f^i\delta t^i = \frac{1}{\lambda^i}\rho_r^i + N_r^i + f^i\delta t_r \tag{2.5}$$

2.2 Differential positioning

Differential positioning techniques using GNSS need two or more receivers. One receiver, usually static, is located at a reference station with known coordinates, and remote receivers, usually are roving and their coordinates are to be determined [39]. The objective of this technique is to reduce or eliminate spatially

correlated error sources such as the atmospheric, orbital and clock errors.

The most common difference is the between-receiver, which is obtained when the observations of a set of satellites are differenced between the rover receiver and the base station. An important feature of this difference is that the satellite clock error is canceled and atmospheric errors are reduced. This occurs because the satellite clock is very stable, making the satellite clock errors the same for these near-simultaneous transmissions. Denoting the receivers by A and B and the satellite by i, and considering equation (2.5) for each receiver, the single-difference equation using phase measurements is the following:

$$\Phi^i_{AB} = \frac{1}{\lambda^i}\rho^i_{AB} + N^i_{AB} + f^i \delta t_{AB} \tag{2.6}$$

If we assume two receivers A, B and two satellites i,j two single differences (equation 2.6) for each satellite may be formed. In order to obtain a double-difference (DD), these single differences are subtracted. The receiver clock bias is eliminated using double-differencing. Assuming equal frequencies for the satellites i and j, the DD equation is:

$$\Phi^{jk}_{AB} = \frac{1}{\lambda^{jk}}\rho^j_{AB} + N^{jk}_{AB} \tag{2.7}$$

The use of double-differences in ambiguity resolution is important. The reason is that by doing this, the clock terms are eliminated and the isolation of the ambiguities is possible.

2.3 Ambiguity Resolution

In order to leverage phase measurements, one has to compute the ambiguous number of cycles between satellites and receivers. Once this is done, carrier phase measurements act as if they were very precise pseudorange measurements. Many of the ambiguity resolution methods first estimate a float-valued solution, which will be the basis for an integer-valued estimation. Then, the integer ambiguity estimates are used to correct the float estimate values. The integer ambiguity computed from the float ambiguity estimate $\check{a}$ [40] is expressed as follows:

$$\check{a} = S(\hat{a}) \tag{2.8}$$

Where S is the mapping from the n-dimensional space of real numbers to the n-dimensional space of integers. This mapping must be many-to-one, so that real-valued ambiguity vectors a will be mapped to the same integer vector. Therefore a subset of real values S_z can be assigned to each integer vector z. This is called the pull-in region of z. Algorithms for ambiguity resolution are based on constructing a search space of integer ambiguities from integer values near the float solution. Usually, search space is bounded by the

uncertainty of the estimated float ambiguities, then the most likely correct integer value is selected. In [41], the conditions for an integer estimator to be admissible are given, which are the following three:

1. All pull-in regions should cover the complete space R^n, so that all real-valued vectors are mapped to an integer vector

2. There should be no overlap between the pull-in regions so that the float solution is mapped to just one integer vector.

3. If the float solution is perturbed by an amount, the corresponding integer solution is perturbed by the same amount.

In literature, we can find several authors who have studied and analyzed different integer estimators or strategies. The simplest one is to round each entry of the real-valued float solution to its nearest integer. The pull-in regions $S_{R,z}$ that corresponds to Integer rounding is as follows:

$$S_{R,z} = \bigcap_{i=1}^{n} \left\{ x \in R^n \mid |x_i - z_i| \leq \frac{1}{2} \right\} \tag{2.9}$$

The two-dimensional case of the integer rounding is visualized in Figure 2.1.

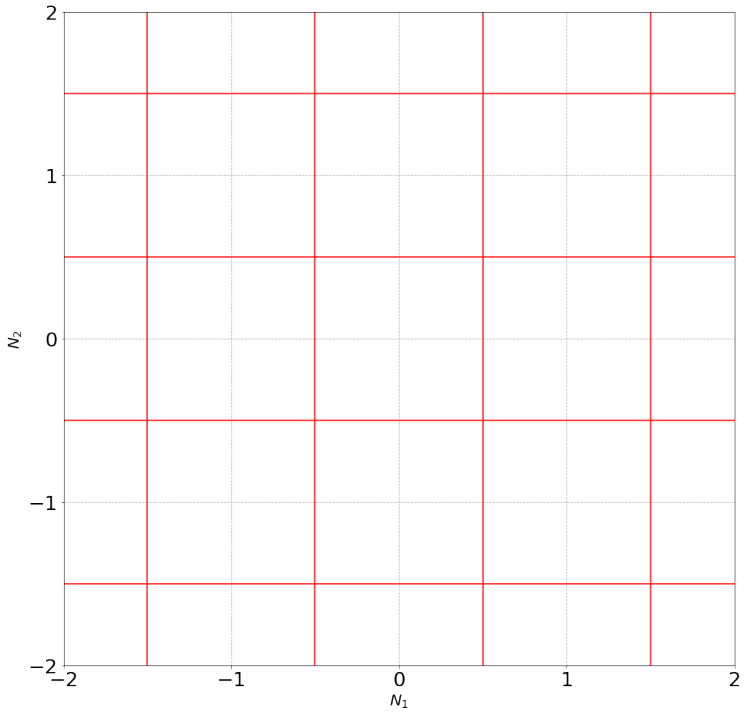


Figure 2.1: Integer rounding two-dimensional pull-in regions.

2.3.1 Integer bootstrapping

Integer bootstrapping or sequential conditional rounding occurs when the first ambiguity is rounded to its nearest integer value and then the remaining float ambiguities are corrected subject to the fixing of the first ambiguity [42]. This estimator makes use of the integer rounding, and also takes the correlation between the ambiguities into account [43]. It is the result from a sequential conditional least-squares adjustment. The bootstrapped estimator is described as follows: assume we have n ambiguities available, one starts with the first real-valued ambiguity $\hat{a}$, and rounds it to the nearest integer. Afterwards, having the first integer ambiguity, the real-valued of the remaining ambiguities are then corrected according to their correlation with the first ambiguity. Subsequently, the second (now corrected) real-valued ambiguity is rounded to its nearest integer. Having rounded the second ambiguity, the real-valued estimates of the remaining n-2 ambiguities are again corrected, but now based on their correlation with the second ambiguity. This process is continued until all ambiguities are corrected and rounded [44]. The nth component of the bootstrapped estimator $\check{a}_B$ is as follows:

$$\check{a}_{B,n} = \left[\hat{a}_{n|N}\right] = \left[\hat{a} - \sum_{i=1}^{n-1} \sigma_{\hat{a}_n \hat{a}_{i|I}} \sigma^{-2}_{\hat{a}_{i|I}} (\hat{a}_{i|I} - \check{a}_{B,i})\right] \tag{2.10}$$

where $\hat{a}_{i|I}$ stands for the ith least squares ambiguity conditioned on the previous sequentially rounded ambiguities. Two-dimensional example for bootstrapping is shown in Figure 2.2. This algorithm is closely related to the LDU triangular decomposition of the variance-covariance matrix of the ambiguities $Q_{\hat{a}} = LDL^T$, with L a unit lower triangular matrix and D a diagonal matrix. Having:

$$(\hat{a} - z) = L(a' - z) \tag{2.11}$$

with a' being the conditional least-squares solution obtained from a sequential conditioning on z entries, and a variance-covariance matrix given by D ($\sigma^{-2}_{\hat{a}_{i|I}}$). The pull-in region $S_{B,z}$ is given by:

$$S_{B,z} = \bigcap_{c_i \in Z^n} \left\{ \hat{a} \in R^n \mid |c_i^T D^{-1} L^{-1} (\hat{a} - z)| \leq \frac{1}{2} c_i^T D^{-1} c_i \right\} \tag{2.12}$$

with c_i the ith canonical unit vector, which has 1 as its ith entry and zeros otherwise.

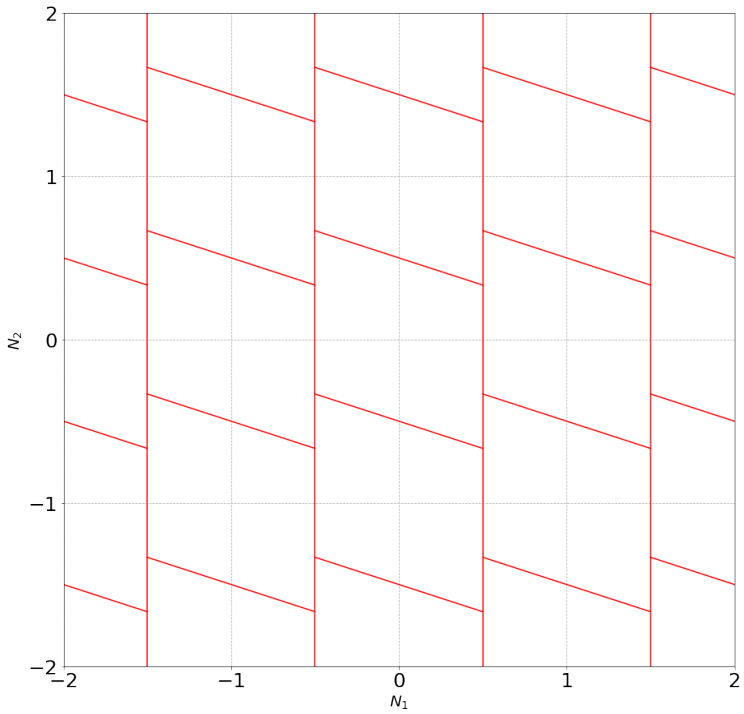


Figure 2.2: Bootstrapping two-dimensional pull-in regions.

2.3.2 Integer least squares

Least-squares problems that involve some integer parameters are non-standard [44] and are referred to as integer least squares (ILS). Having:

$$min \left\| \hat{a} - z \right\|^2_{Q_{\hat{a}}} \tag{2.13}$$

where $z \in Z^n$. Therefore, $\check{a}_{LSQ}$ is the integer least-squares solution when:

$$\left\| \hat{a} - \check{a}_{LSQ} \right\|^2_{Q_{\hat{a}}} \leq \left\| \hat{a} - z \right\|^2_{Q_{\hat{a}}} \tag{2.14}$$

for all $z \in Z^n$. Expression 2.14 can be written as:

$$(z - \check{a}_{LSQ})^T Q^{-1}_{\hat{a}} (\hat{a} - \check{a}_{LSQ})^T \leq \frac{1}{2} \left\| z - \check{a}_{LSQ} \right\|^2_{Q_{\hat{a}}} \tag{2.15}$$

Hence, the pull-in region $S_{LSQ,z}$ using the LDU decomposition is as follows:

$$S_{LSQ,z} = \bigcap_{c_i \in L^{-1}(Z^n)} \left\{ \hat{a} \in R^n \mid \left| c_i^T D^{-1} L^{-1} (\hat{a} - z) \right| \leq \frac{1}{2} c_i^T D^{-1} c_i \right\} \tag{2.16}$$

The two-dimensional case is visualized in Figure 2.3. Comparing the two pull-in regions 2.12 and 2.16 shows that they only differ in their choice of c_i vectors. This means that the two pull-in regions are identical when the matrix entries of L^- are all integer, which is the case when L is an admissible ambiguity transformation. It is also shown that subset 2.2 is reduced to the integer rounding subset 2.1 when L becomes diagonal. This happens when the variance-covariance is diagonal.

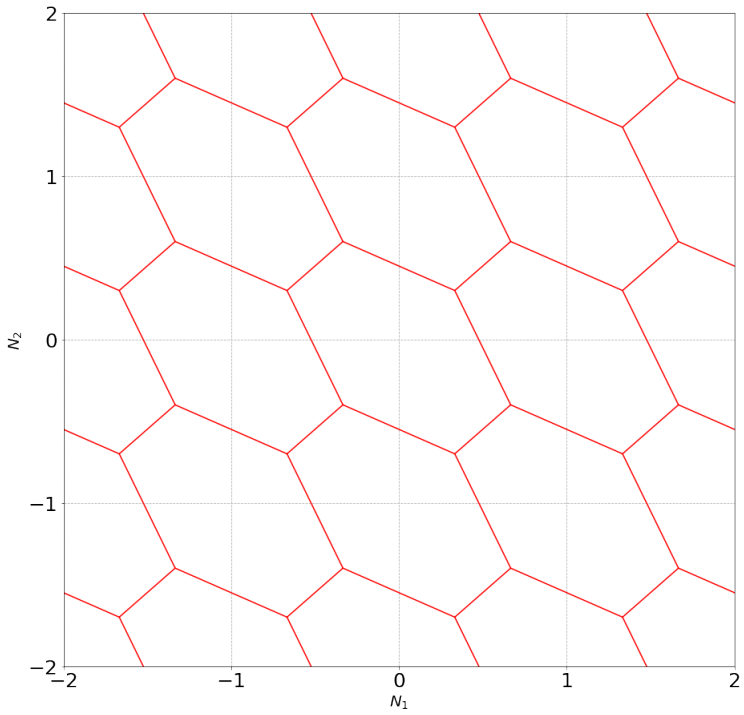


Figure 2.3: ILS two-dimensional pull-in regions.

Figures 2.1, 2.2, 2.3 show the two-dimensional pull-in regions for integer rounding, bootstrapping and ILS. The geometry of the pull-in regions depends on the nature of the integer estimator. The integer rounding, being the simplest of these methods, has squared 2D pull-in regions, as all the real valued ambiguities are mapped to their closest integer value. The bootstrapping pull-in regions are similar to the ones of the rounding method, with the difference that the slopes of the regions are due to the corrections of the float ambiguities considering the correlation between them. Finally, the ILS pull-in regions are constructed as intersecting half spaces, which are bounded by planes orthogonal to $\hat{a} - z$ [14]. For the 2D case three pairs of such half spaces are needed, as the integer vectors must be bounded by $2n - 1$ adjacent integers. Therefore the ILS pull-in regions for a 2D case are hexagons.

2.3.3 LAMBDA

The Least Squares Ambiguity Decorrelation Adjustment (LAMBDA) [1] method solves an integer least squares problem successfully solving the ambiguity resolution problem. The integer ambiguity solution is obtained by using one of the three search methods mentioned in the previous section. An integer search space is defined as [14]:

$$\Omega_a = a \in Z^n | (\hat{a} - a)^T Q_{\hat{a}}^{-1} (\hat{a} - a) \leq X^2 \tag{2.17}$$

The search space is an n-dimensional ellipsoid with center at $\hat{a}$, shaped by the variance-covariance matrix $Q_{\hat{a}}^{-1}$, and its size is defined by X^2. High correlation between the ambiguities result in extremely elongated search spaces, therefore, the integer search may take very long. The LAMBDA decorrelation method is necessary for transforming the search space to a more spherical shape. In this method, the ambiguities are decorrelated prior to integer estimation using the Z-transformation [45]:

$$z = Z^* a, \hat{z} = Z^* \hat{a}, Q_{\hat{z}} = Z^* Q_{\hat{a}} Z \tag{2.18}$$

This is only admissible if and only if Z^* is volume preserving and has integer entries only. Hence, the ILS problem would then transform into the equivalent minimization problem:

$$min(\hat{z} - z)^* Q_{\hat{z}}^{-1} (\hat{z} - z) \tag{2.19}$$

with $z \in Z^n$. The decorrelation realizes an ellipsoid that is very much a sphere or circular, therefore, it can be searched through efficiently. Using the LDL^T decomposition of $Q_{\hat{z}}$ the new ambiguity search space is then given by:

$$\sum_{i=1}^{n} (\hat{z}_{i|I} - z_i)^2 / \sigma_{i|I}^2 \leq X^2 \tag{2.20}$$

Its size can be controlled through the selection of the positive constant X^2, and it gives an indication of the number of candidates contained in the ellipsoid. A limited number of candidates will be the output of which one will be the least-square estimation for the vector of ambiguities. Finally, this solution is used to correct the real estimate of the baseline parameters and is known as the fixed baseline solution. In figures 2.4 and 2.5, it is shown a two-dimensional example of the search space before (2.4) and after (2.5) the decorrelating Z transformation. As mentioned before, the transformed space is less elongated, however, it has a volume identical to the volume of the original search space and the same number of candidate grid points are inside the ellipse in both cases. The difference is the number of candidates in the rectangular

search region. In this example, the next values were used:

$$\hat{a} = \begin{pmatrix} 27.6490 \\ 10.3038 \end{pmatrix} \quad \text{and} \quad Q_{\hat{a}} = \begin{pmatrix} 25.04 & 29 \\ 29 & 36.04 \end{pmatrix}$$

where $\hat{a}$ are the float values with its respective variance-covariance matrix $Q_{\hat{a}}$. The transformation matrix Z is:

$$Z = \begin{pmatrix} 2 & 1 \\ -1 & -1 \end{pmatrix}$$

After transformation, the float estimate and the covariance matrix become:

$$Q_{\hat{z}} = Z^T Q_{\hat{a}} Z = \begin{pmatrix} 20.2 & -0.88 \\ -0.88 & 3.08 \end{pmatrix} \quad \text{and} \quad z = Z^T \hat{a} = \begin{pmatrix} 44.9942 \\ 17.3452 \end{pmatrix}$$

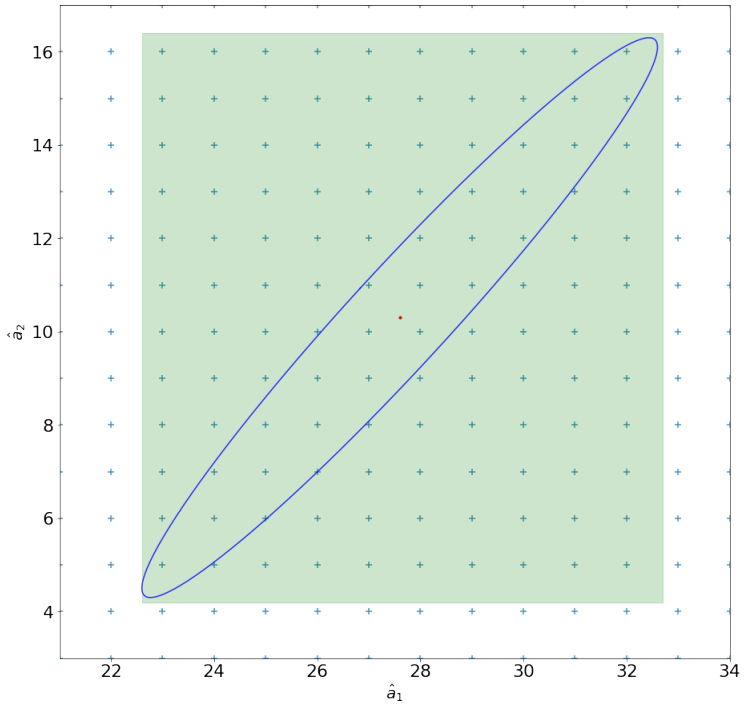


Figure 2.4: Original search space.

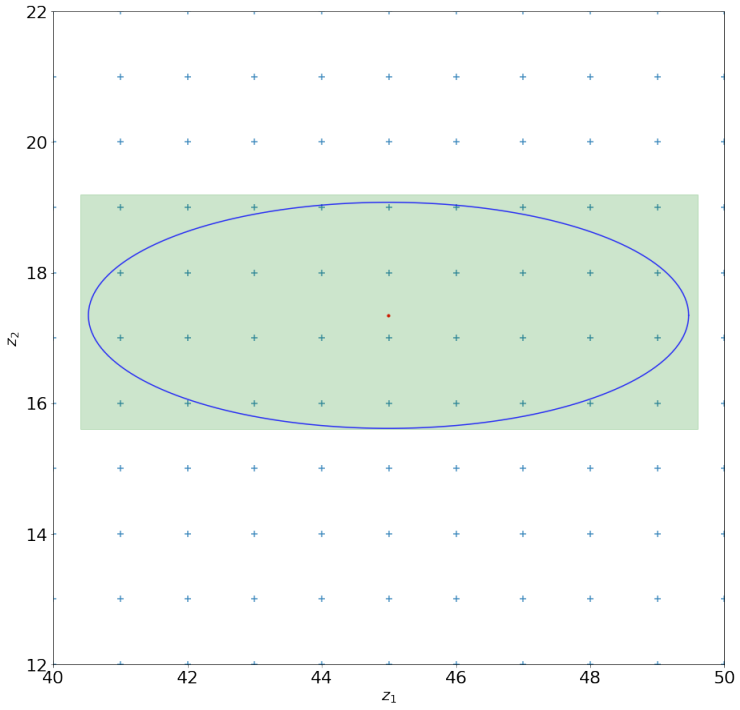


Figure 2.5: Search space after decorrelation.

2.3.4 Ratio test

After the identification of the most likely integer ambiguity combination, a process of validation must be made. This is a necessary step before fixing or constraining the ambiguities to the computed integers in a final baseline computation. The ratio test was introduced in order to decide whether the integer ambiguity solution can be considered sufficiently more likely than any other candidate [46]. Lets consider $\check{a}$ as the integer minimizer of $R(a)$:

$$R(a) = (\hat{a} - a)^T Q_{\hat{a}\hat{a}}^{-1} (\hat{a} - a) \tag{2.21}$$

Let a' be the integer that returns the second smallest value of $R(a)$. Then, the ratio test is as follows:

$$\frac{R(a')}{R(a)} > kc \tag{2.22}$$

where kc is the chosen critical value. Equation (2.22) evaluates how close is the float solution to its nearest integer vector. This will be the criteria for accepting or rejecting the ILS solution. If it is not close enough, the test leads to rejection in favor of the float solution [47].

2.3.5 Success rates

The success rate or probability of correct integer ambiguity estimation describes the frequency with which one can expect to have a successful ambiguity resolution. The ambiguity success rate depends on the observation equations, the precision of the observables, and the chosen method of integer estimation. The integer least-squares estimator is the most effective method used for maximizing the success rate[48]:

$$P(\check{a}_R = a) \leq P(\check{a}_B = a) \leq P(\check{a}_{LS} = a) \tag{2.23}$$

where $P(\check{a} = a)$ is the probability of correct integer estimation for integer rounding R, bootstrapped B and integer least-squares LS estimators. Hence, one can use the bootstrapped or the integer rounding success rates as a lower bound for the integer least-squares success rates. The bootstrapped probability of obtaining the correct integer ambiguity is [36]:

$$P(\check{a}_B = a) = \prod_{i=1}^{n} (2\phi\left(\frac{1}{2\sigma_{\hat{a}_{i|I}}}\right) - 1) \tag{2.24}$$

where ϕ is the normal distribution described by:

$$\phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}z^2)dz \tag{2.25}$$

Note the success rates described by Equation (2.24) depend only on the standard deviation of the float ambiguities $\sigma_{\hat{a}_{i|I}}$ and does not consider biases in the float solution, therefore, a high success rate does not always guarantee a correct integer search. In Chapter 4, the empirical success rates are compared to the probabilities of correct fixing, providing a more realistic metric for correct integer searching. The three integer estimators described in this chapter are applied with and without the Z transformation throughout this research with the objective of assess the performance of each method having the float solution represented by a set of particles in different cases. Having the original float ambiguities decorrelated through the Z transformation, enables the analysis of the particle filter solution performance using the empirical covariance matrix obtained from the cloud of particles as an input for LAMBDA method. applying The double-difference measurements are used to reduce the atmospheric effects and cancel the clock errors in order to estimate the ambiguities. Finally, the success rates are computed and then compared with the empirical probability of correct fixing obtained from the PF solution. The methods outlined in this chapter, regarding ambiguity resolution, require a float solution as an input. Typically, an EKF is employed to obtain this solution. However, these methods just require an estimation of the float solution along with its covariance matrix. In

the upcoming chapter, Kalman filtering and alternative approaches will be explored.

Chapter 3

Nonlinear Filters

In this chapter, a description of the most common non-linear estimator, the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), and the Particle Filter (PF) is provided. The Sequential Importance Resampling (SIR) Particle Filter is described in detail as it is the estimator used throughout this thesis. Afterwards, two methods to improve the accuracy and reduce the computational cost of the SIR particle filter by selecting a different importance density than the transitional prior are described. The Rao-Blackwellized PF is described as an alternative method to increase the efficiency and reduce the computational cost by splitting the state vector into linear and non-linear parts. Finally, a simple example with simulated data is provided showing the performance of each filter when estimating a one-dimensional state vector.

3.1 Sequential Least-Squares

Suppose we need to adjust a state vector according to a set of measurements observed at two sequential epochs, the portion of observations y_{k-1} at a certain epoch $k-1$ is used to adjust the state vector $\hat{x}$ which will be considered at a later epoch when the next set of measurements y_k is available. Instead of adjusting again the combined observations $y_{k-1} + y_k$ when the second set of measurements is available, y_k is used to provide a correction Δx to the adjusted state $\hat{x}_{k-1}$ based on the first batch of measurements [49]. In order to standardize the notation for the entire explanation of this chapter and make the comparison between the estimators, I will follow the *Kalman filter* notation used by most authors ([50], [51], [52]). Any special notation or exception will be pointed out. The adjusted observations in a parametric model of both epochs are given by:

$$\hat{y}_{k-1} = f(\hat{x}_{k-1}) \tag{3.1}$$

$$\hat{y}_k = f(\hat{x}_k) \tag{3.2}$$

The main idea of doing the sequential least-squares is to add the effects of the second set of observations, without having to readjust again both sets together as one. Considering v_{k-1} and v_k as the vectors of residuals of the observations, Equations (3.1) and (3.2) can be described as follows:

$$\hat{y}_{k-1} + v_{k-1} = f(\hat{x}_{k-1}) \tag{3.3}$$

$$\hat{y}_k + v_k = f(\hat{x}_k) \tag{3.4}$$

Linearizing these equations by a first-order Taylor series expansion and taking into account the least-squares criterion, the variation function is:

$$\phi = v_{k-1}^T R_{k-1}^{-1} v_{k-1} + v_k^T R_k^{-1} v_k - 2\bar{K}_k^T (H_{k-1}\hat{x}_{k-1} + w_{k-1} - v_{k-1}) - 2\bar{K}_k^T (H_k \hat{x}_k + w_k - v_k) = minimum \tag{3.5}$$

where R is the covariance matrix of the measurements noise for each set (inverse of the measurements weights), $\bar{K}$ is an undetermined constant called the *Lagrange multiplier*, H_k is the Jacobian matrix of partial derivatives of the function f with respect to x $H_k = \frac{\partial f_k}{\partial x}$ and $w_k = f(x^-) - y_k$ the innovation. Solving the set of normal equations derived from the variation function 3.5, the solution for the sequential least-squares is obtained, and can be summarized as follows [53]:

1. Compute the state $\hat{x}_{k-1}$ with:

$$\hat{x}_{k-1} = -(H_{k-1}^T R_{k-1}^{-1} H_{k-1})^{-1} H_{k-1}^T R_{k-1}^{-1} w_{k-1} \tag{3.6}$$

2. Compute the residual vector to be applied to measurements y_{k-1}:

$$v'_{k-1} = H_{k-1}\hat{x}_{k-1} + w_{k-1} \tag{3.7}$$

3. Compute the covariance matrix of the adjusted parameters:

$$\hat{P}_{k-1} = (H_{k-1}^T R_{k-1}^{-1} H_{k-1})^{-1} \tag{3.8}$$

4. When the second set of measurements is available, compute the final state vector:

$$\hat{x}_k = \hat{x}_{k-1} + \Delta\hat{x} \tag{3.9}$$

where $\Delta\ \hat{x}$ are the changes in corrections due to adjustments of measurements y_k:

$$\Delta\hat{x} = -(H_{k-1}^T R_{k-1}^{-1} H_{k-1})^{-1} H_k^T (R_k + H_k (H_{k-1}^T R_{k-1}^{-1} K_{k-1})^{-1} H_k^T)^{-1} (H_k \hat{x}_{k-1} + w_k) \tag{3.10}$$

5. Compute the residual vectors v_{k-1} and v_k as:

$$\begin{aligned} v_{k-1} &= v'_{k-1} + H_1 \Delta\hat{x} \\ v_k &= R_k M (H_k \hat{x}_{k-1} + w_k) \end{aligned} \tag{3.11}$$

with:

$$M = (R_k + H_k (H_{k-1}^T R_{k-1}^{-1} H_{k-1})^{-1} H_k^T)^{-1} \tag{3.12}$$

6. Compute the final covariance matrix:

$$\hat{P}_k = \hat{P}_{k-1} + (-((H_{k-1}^T R_{k-1}^{-1} H_1)^{-1}) H_k^T M H_k (H_{k-1}^T R_{k-1}^{-1} H_{k-1})^{-1}) \tag{3.13}$$

which considering Equation 3.8, can be rewritten as follows:

$$\hat{P}_k = \hat{P}_{k-1} - (\hat{P}_{k-1}) H_k^T M H_k (\hat{P}_{k-1}) \tag{3.14}$$

Substitution of Equation 3.8 in Equation 3.10 yields:

$$\Delta\hat{x} = K_k (H_k \hat{x}_{k-1} + w_k) \tag{3.15}$$

where K, known as Kalman gain is:

$$K_k = \hat{P}_{k-1} H_k^T (R_k + H_k \hat{P}_{k-1} H_k^T)^{-1} \tag{3.16}$$

then Equation 3.9 can be rewritten as:

$$\hat{x}_k = \hat{x}_{k-1} + K_k (y_k - \hat{y}_k) \tag{3.17}$$

where $\hat{y}_k = H_k \hat{x}_{k-1}$. Equations (3.17) and (3.14) are equivalent to the Kalman Filter update equation for the state vector and its covariance matrix. Therefore, the SLS constitutes a filtering form solution in a static mode which minimizes the squared error σ by the factor K in Equation (3.16). This means that the SLS is

considered a special case of filtering which does not considers a dynamic model as the EKF does.

3.2 Extended Kalman Filter

For estimating the state of a system changing over time described by state dynamics and observation models, the use of a recursive filter is a convenient solution and consists of essentially two stages, prediction, and update. In the prediction stage, the state pdf is predicted for the subsequent epoch using the state dynamics model. The update stage uses the incoming measurements at each epoch to modify the predicted pdf. Consider a discrete dynamical system defined by the following state dynamics:

$$x_k = \mathbf{\Phi}_k x_{k-1} + w_k \tag{3.18}$$

and the observation model is as follows:

$$y_k = H(x_k) + n_k \tag{3.19}$$

where Φ is the transition matrix of the dynamics model, $H(x_k)$ is the matrix giving the connection between the measurement with noise n_k and the state vector at time k, which for the nonlinear case will be the Jacobian matrix of partial derivatives of the measurement model just as in the sequential least-squares. The process state and measurement vectors at time k are denoted as x_k and y_k respectively. Having a prior estimate of the state $\hat{x}_k^-$ (the super minus $^-$ indicates the prior estimation while the hat denotes estimation only), the Kalman filter seeks to use the measurement y_k to improve the prior estimate. This is the update step of the filter and it is as follows:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{3.20}$$

where K_k is known as the *Kalman gain*, it minimizes the mean-square estimation error and follows the next equation:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{3.21}$$

where R_{kk} is the covariance matrix of the measurement noise and $\hat{P}_k$ is the covariance matrix associated with the optimal estimate and is described by the next equation:

$$P_k = (I - K_k H_k) P_k^- \tag{3.22}$$

We need now the best prior estimate for the state vector $\hat{x}_k^-$ and its covariance matrix P_k^-, which are computed in the prediction step of the filter. The prediction of the state is projected using the transition matrix Φ:

$$\hat{x}_{k+1}^- = \Phi \hat{x}_k \tag{3.23}$$

The expression for $\hat{P}_{k+1}^-$ is as follows:

$$\hat{P}_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \tag{3.24}$$

where $\mathbf{Q}_k$ is the variance of the process noise (assumed Gaussian).

In most of the applications of the Kalman filter, the dynamics are nonlinear, therefore the extended Kalman filter utilizes a first-order Taylor series approximation to the nonlinear function. Considering a nonlinear dynamic system, the process to be estimated and the observation model are described by the following equations [54]:

$$x_{k+1} = f(x) + w_k \tag{3.25}$$

and

$$y = h(x) + n_k \tag{3.26}$$

where f and h are known nonlinear functions. The system is linearized by approximating f and h with the first-order terms of Taylor's series expansions and then evaluating the partial derivative matrices at point x, which are the equivalent matrices of the linear case Φ and H:

$$\Phi = \left. \frac{\partial f(x, w)}{\partial x} \right|_{x,w} \tag{3.27}$$

$$H = \left. \frac{\partial h(x)}{\partial x} \right|_{x} \tag{3.28}$$

The EKF updates the state estimates resulting from the measurements as:

$$y_k - h(\hat{x}_k^-) \tag{3.29}$$

Then the update step in the EKF is as follows:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - h(\hat{x}_k^-)) \tag{3.30}$$

The recursive process of estimating the state of a system using the Kalman Filter is described in the next algorithm:

Algorithm 1 Extended Kalman Filter

Require: Initial state estimate $\hat{x}_0$, initial state error covariance matrix P_0, process noise covariance matrix Q, measurement noise covariance matrix R, measurement function $h(\cdot)$, state transition function $f(\cdot)$, and Jacobian matrices F_k and H_k

1: **for** $k = 1, 2, \ldots$ **do**

2: $\hat{x}_k^- = f(\hat{x}_{k-1})$ ▷ Predicted state estimate

3: $P_k^- = F_k P_{k-1} F_k^T + Q$ ▷ Predicted state error covariance

4: $H_k = \frac{\partial h}{\partial x}\Big|_{\hat{x}_k^-}$ ▷ Linearize measurement function around predicted state

5: $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$ ▷ Calculate Kalman gain

6: $\hat{z}_k = h(\hat{x}_k^-)$ ▷ Predicted measurement

7: $\hat{x}_k = \hat{x}_k^- + K_k(y_k - \hat{z}_k)$ ▷ Updated state estimate

8: $P_k = (I - K_k H_k) P_k^-$ ▷ Updated state error covariance

9: **end for**

10: **return** Filtered state estimates $\hat{x}_k$ with its covariance P_{xk}

3.3 Unscented Kalman Filter

As the EKF only provides a first-order Taylor series approximation to the recursive nonlinear estimation, it provides a limited performance when dealing with high nonlinearity [55]. Another drawback of the EKF is that differentiating complex models for generating the Jacobian matrices is not trivial in most cases. These limitations can introduce large errors in the estimation and sometimes the filter can diverge. The unscented Kalman filter (UKF) addresses nonlinearity using a deterministic sampling approach. This method provides estimates of the mean and its covariance based on a set of discrete samples projected exactly through the associated nonlinear transformation, hence, this estimator is derivative-free. The chosen set of points completely captures the true mean and covariance of the state, and the estimation results in approximations accurate to the third-order for Gaussian inputs, and at least second-order for non-Gaussian [56]. The UKF differs from the Monte Carlo methods (such as the Particle Filter) which require a large set of random samples with associated weights to describe the posterior density function, while UKF draws samples according to a specific deterministic algorithm.

The UKF is named after the *Unscented Transform* (UT), a method for computing the statistics of a random variable that undergoes a nonlinear transformation [57]. Lets consider that the random variable x of dimension n with mean $\bar{x}$ and covariance P_x can be approximated by $2n+1$ weighted samples (commonly

named *sigma points* $\boldsymbol{\chi}_i$)[58]:

$$\begin{aligned}
\boldsymbol{\chi}_0 &= \bar{x} \\
\boldsymbol{\chi}_i &= \bar{x} + (\sqrt{(n+\lambda)})_i \quad i = 1, ..., n \\
\boldsymbol{\chi}_i &= \bar{x} - (\sqrt{(n+\lambda)})_{i-n} \quad i = n+1, ..., 2n
\end{aligned} \tag{3.31}$$

where:

$$\lambda = \alpha(n + \kappa) - n \tag{3.32}$$

is a scaling parameter. The constant α determines the spread of the points around the mean and is usually set to a small positive value. The parameter κ is usually set to 0 [55]. $(\sqrt{(n+\lambda)})_i$ is the ith column of the matrix square root using the Cholesky factorization. Each point is then propagated through the nonlinear process model $f(x)$:

$$\boldsymbol{\mathcal{Y}} = f(\boldsymbol{\chi}) \tag{3.33}$$

The mean and covariance of the propagated points with associated weights W_i are computed through the $Unscented\ Transform$ as follows:

$$\bar{y} \approx \sum_{i=0}^{2n} W_i^{(m)} \boldsymbol{\mathcal{Y}}_i \tag{3.34}$$

$$P_y \approx \sum_{i=0}^{2n} W_i^{(c)} \{\boldsymbol{\mathcal{Y}}_i - \bar{y}\}\{\boldsymbol{\mathcal{Y}}_i - \bar{y}\}^T \tag{3.35}$$

where the weights are given by the next equations:

$$\begin{aligned}
\boldsymbol{W}_0^{(m)} &= \lambda/(n+\lambda) \\
\boldsymbol{W}_0^{(c)} &= \lambda/(n+\lambda) + (1 + \alpha^2 + \beta) \\
\boldsymbol{W}_i^{(m)} &= \boldsymbol{W}_i^{(c)} = 1/\{2(n+\lambda)\} \quad i = 1, ..., 2n
\end{aligned}$$

and β is set to 2 for Gaussian Distributions. At this step of the filter, we have a set of points that have been selected through equations (3.31) and have been propagated using the nonlinear function $f(x, w)$. The mean and its covariance of this cloud of particles are computed afterward using their associated weights and the UT (equations (3.34) and (3.35)). This step is analogous to the predict step in the EKF. As the Kalman Filter performs the update step in measurement space, we need to propagate the samples using the nonlinear

measurement model $h(x)$,

$$\mathbf{Z} = f(\boldsymbol{\mathcal{Y}}) \tag{3.36}$$

Having the sigma points transformed into measurements, we can compute their mean $\bar{\mathbf{z}}$ and its covariance P_z through the UT (equations (3.34) and (3.35)). With the incoming measurements y we compute the innovation r and then the Kalman gain,

$$r = y - \bar{\mathbf{z}} \tag{3.37}$$

$$K = P_{xz} P_z^{-1} \tag{3.38}$$

where P_{xz} is the cross-covariance matrix of the state and measurements, which is computed as follows:

$$P_{xz} = \sum_{i=0}^{2n} W_i^{(c)} \{\boldsymbol{\mathcal{Y}}_i - \bar{x}\}\{\mathbf{Z}_i - \bar{z}\}^T + R \tag{3.39}$$

Finally, we update the state and its covariance matrix:

$$\hat{x}_k = \hat{x}_k^- + Kr \tag{3.40}$$

$$P_x = P_x^- - K P_z K^T \tag{3.41}$$

As the EKF assumes the state distribution to be Gaussian which is propagated through the first-order linearization of the nonlinear system, it can introduce large errors in the true posterior mean and covariance of the transformed state, which may lead to sub-optimal performance of the filter. The UKF addresses this limitation by representing the state again by a Gaussian, but through a minimal set of deterministically chosen sample points which are propagated through the true nonlinear system, capturing the posterior mean and covariance accurately to the 3rd order Taylor series expansion, for any nonlinear model [55]. The UKF is an alternative solution when the measurement model is highly nonlinear and the EKF does not longer perform efficiently. The algorithm for estimating a dynamic state using the UKF is the following:

Algorithm 2 Unscented Kalman Filter

Require: Initial state estimate $\hat{x}_0$, initial state error covariance matrix P_0, process noise covariance matrix Q, measurement noise covariance matrix R, measurement function $h(\cdot)$, state transition function $f(\cdot)$, and scaling parameters α, β, and κ

$\lambda = \alpha(n + \kappa) - n$ ▷ Calculate scaling parameter

$\boldsymbol{W}_m = [\frac{\lambda}{n+\lambda}, \frac{1}{2(n+\lambda)}, \ldots, \frac{1}{2(n+\lambda)}]$ ▷ Calculate weight vector for mean

$\boldsymbol{W}_c = [\frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \frac{1}{2(n+\lambda)}, \ldots, \frac{1}{2(n+\lambda)}]$ ▷ Calculate weight vector for covariance

$\boldsymbol{\chi}_0 = \hat{x}_0$ ▷ Initialize sigma point matrix

$P_0 = P_0 + Q$ ▷ Add process noise to initial state error covariance

for $k = 1, 2, \ldots$ **do**

  $\boldsymbol{\chi}_{k-1}$ = sigma points around $\hat{x}_{k-1}$

  $\boldsymbol{\mathcal{Y}}_k^- = f(\boldsymbol{\mathcal{Y}}_{k-1})$ ▷ Predicted sigma points

  $\hat{x}_k^- = \sum_i \boldsymbol{W}_m(i)\boldsymbol{\mathcal{Y}}_{k,i}^-$ ▷ Predicted state estimate

  $P_k^- = \sum_i \boldsymbol{W}_c(i)(\boldsymbol{\mathcal{Y}}_{k,i}^- - \hat{x}_k^-)(\boldsymbol{\mathcal{Y}}_{k,i}^- - \hat{x}_k^-)^T$ ▷ Predicted state error covariance

  $\mathbf{Z}_k^- = h(\boldsymbol{\mathcal{Y}}_k^-)$ ▷ Predicted measurement sigma points

  $\hat{z}_k^- = \sum_i \boldsymbol{W}_m(i)\mathbf{Z}_{k,i}^-$ ▷ Predicted measurement mean

  $P_{xz,k} = \sum_i \boldsymbol{W}_c(i)(\boldsymbol{\mathcal{Y}}_{k,i}^- - \hat{x}_k^-)(\mathbf{Z}_{k,i}^- - \hat{z}_k^-)^T + R$ ▷ Cross-covariance

  $K_k = P_{xy,k}P_k^-$ ▷ Calculate Kalman gain

  $r_k = y_k - \bar{\mathbf{z}}$ ▷ Innovation

  $\hat{x}_k = \hat{x}_k^- + Kr$ ▷ Update state

  $P_{x,k} = P_x^- - KP_zK^T$ ▷ Update state covariance

end for

return Filtered state estimates $\hat{x}_k$ with its covariance P_{xkk}

3.4 Sequential Importance Resampling Particle Filter

3.4.1 Estimation problem in Bayesian framework

From a Bayesian perspective, the state estimation problem of a dynamic system described by dynamics and measurement models (3.18 and 3.19) seeks to construct the posterior density function of the state $p(x_k|Y_k)$, where Y_k are all measurements $\{y_1, y_2, ..., y_k\}$ up to time k [20]. Assuming the prior pdf at time $k-1$ is known, the prediction step consists in using the dynamics model 3.18 to obtain a predicted density, described

as follows:

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1} \tag{3.42}$$

where $p(x_k|x_{k-1})$ is known as transitional density and is defined by the dynamics model 3.18. Equation 3.42 is known as the Chapman-Kolmogorov equation and relates the joint probability distributions of different sets of coordinates. In the update stage, taking into account the measurement y_k, the predicted pdf $p(x_k|Y_{k-1})$ is updated through the Bayes' theorem:

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \tag{3.43}$$

where the normalizing term $p(y_k|Y_{k-1})$ depends on the likelihood function $p(y_k|x_k)$ defined by the measurement model 3.19:

$$p(y_k|Y_{k-1}) = \int p(y_k|x_k)p(x_k|Y_{k-1})dx_k \tag{3.44}$$

therefore:

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{\int p(y_k|x_k)p(x_k|Y_{k-1})dx_k} \tag{3.45}$$

Equations (3.42) and (3.45) are the predict and update step in the state estimation problem of a dynamic system within the Bayes framework.

3.4.2 Minimum mean-square error estimate

Having the posterior pdf $p(x_k|Y_k)$ we can compute the minimum mean-square error (MMSE) estimate through the conditional mean [59]. In probability and statistics, the expected value of a variable is its mean (first moment) [60]:

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} xp(x)dx \triangleq \bar{x} \tag{3.46}$$

where $p(x)$ is the probability of random variable x happening. This means that the expected value can be computed by summing the product of each value x by its own likelihood to occur $p(x)$. An MMSE estimator seeks to minimize the mean-square error:

$$\hat{x}^{MMSE} =_{argmin} \mathbb{E}[(\hat{x} - x)^2|Y] \tag{3.47}$$

where $(\hat{x} - x)^2$ is the loss function from the sum of squares of the errors of the estimate $\hat{x}$ and the unknown parameters x. This loss function leads to the least-squares method [61]. Deriving $\mathbb{E}[(\hat{x}-x)^2|Y]$ from Equation

3.47 from respect to $\hat{x}$ and setting to zero in order to find the minimum:

$$\begin{aligned} \mathbb{E}[2(\hat{x} - x)|Y] &= 0 \\ \mathbb{E}[x|Y] &= \hat{x} \end{aligned} \tag{3.48}$$

Applying the definition of expected value (Equation 3.46) of a random variable with pdf $p(x_k|Y_k)$ at time k, it is shown that the MMSE is the mean of x_k given the measurements Y_k (conditional mean):

$$\hat{x}_k^{MMSE} \triangleq \mathbb{E}[x_k|Y_k] = \int_{-\infty}^{\infty} x_k p(x_k|Y_k) dx \tag{3.49}$$

Equation 3.49 shows that the expected value $\mathbb{E}[x_k|Y_k]$ is computed according to 3.46, using the posterior density function $p(x_k|Y_k)$ for x_k given the values of Y_k. In the following subsection, it is demonstrated that the posterior can be described as a sum of random samples with associated normalized weights drawn from a similar density function called *importance density*.

3.4.3 Importance Sampling

The cumulative distribution function (cdf) gives the probability of a random variable X taking a value less than or equal to a given value x [62]. The cdf of a random variable X is defined by its relation to the pdf $p(x)$ for a variable value x as follows:

$$\begin{aligned} F(x) &= P(X \leq x) = \int_{-\infty}^{x} p(x) dx \\ p(x) &= \frac{dF(x)}{dx} \end{aligned} \tag{3.50}$$

If the state space is discrete and consists of a finite number of values for x with range $x_1, x_2, ..., x_n$ with associated probabilities $P(x_i)$ to occur, the cdf of x is then as follows:

$$F(x) = \sum_{i=1}^{n} P(x_i) U(x - x_i) \tag{3.51}$$

where $U(x)$ is the unitary step function also known as the $Heaviside$ step function, defined as

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

According to Equations (3.50), deriving Equation 3.51 to get the generalized pdf of x as follows [63]:

$$p(x) = \sum_{i=1}^{n} P(x_i)\delta(x - x_i) \tag{3.52}$$

where $\delta(x - x_i)$ is the Dirac delta function shifted by x_i.

Considering Equation 3.52 and a discrete state space of $x_{k-1}^i, i = 1, ..., N$ at time k, the posterior density 3.42 steps can be written as [64]:

$$p(x_k|Y_k) = \sum_{i=1}^{N} P(x_k^i|Y_k)\delta(x_k - x_k^i) \tag{3.53}$$

where the $P(x_k^i|Y_k)$ is the conditional probability of x_k given the observations Y_k. Equation 3.53 describes the posterior $p(x_k|Y_k)$ as the sum of N points or samples from the state space at time k with associated probabilities conditioned to measurements Y_k. The Monte Carlo (MC) methods associate normalized weights (proportional to $P(x_k^i|Y_k)$) to these samples, which are drawn from an importance density as it is not possible to sample effectively from the posterior distribution.

The basis of the Sequential Monte Carlo method is the Monte Carlo integration method, which states that in order to numerically evaluate a multidimensional integral of the form:

$$I = \int g(x)dx \tag{3.54}$$

we can factorize it as:

$$I = \int f(x)\frac{\pi(x)}{q(x)}q(x)dx \tag{3.55}$$

considering that $g(x) = f(x) \cdot \pi(x)$, where $\pi(x)$ is a probability density which satisfies $\pi(x) \geq 0$ and $\int \pi(x)dx = 1$, and in a Bayesian context, it is the posterior density. As it is not possible to generate samples from the posterior $\pi(x)$, the most reliable option is to generate the particles from the importance density $q(x)$. Assuming that it is possible to generate $N \gg 1$ samples from $q(x)$, then I in 3.54 is estimated by the following weighted sum:

$$I = \frac{1}{N}\sum_{i=1}^{N} f(x^i)\tilde{w}(x^i) \tag{3.56}$$

where $\tilde{w}(x^i)$ are the importance weights equal to:

$$\tilde{w}(x^i) = \frac{\pi(x^i)}{q(x^i)} \tag{3.57}$$

After normalizing the importance weights in equation 3.56, I is described by:

$$I = \sum_{i=1}^{N} f(x^i) w(x^i) \tag{3.58}$$

where $w(x^i)$ are the normalized importance weights equal to $\tilde{w}(x^i)/\sum_{j=1}^{N} \tilde{w}(x^j)$.

The particle filter as a realization of a Bayesian Filter is a Monte Carlo method whose main objective is to represent the required posterior density function $p(\mathbf{x}_k|Y_k)$ at time k with a set of random samples with associated weights and estimate the state vector $\mathbf{x}$ based on these samples and weights [65]. The weighted approximation of the posterior density at time k described in equation 3.53 is as follows [20]:

$$p(x_k|Y_k) \approx \sum_{i=1}^{N_p} w_k^i \delta(x_k - x_k^i) \tag{3.59}$$

where $\bar{w}_k^i$ is the normalized weight of the ith particle drawn from the importance density $q(x_k|Y_k)$:

$$w_k^i \propto \frac{p(x_k|Y_k)}{q(x_k|Y_k)} \tag{3.60}$$

where $q(x_k|Y_k)$ is chosen (for convenience) to be an augmentation of the prior density with the new state x_k^i and the incoming measurement y_k [66] :

$$q(x_k|Y_k) \triangleq q(x_k|X_{k-1}Y_k) q(X_{k-1}|Y_{k-1}) \tag{3.61}$$

Moreover, considering the Bayes Theorem 3.43, the posterior $p(x_k|Y_k)$ in 3.60 is as follows:

$$p(x_k|Y_k) \propto p(y_k|x_k) p(x_k|x_{k-1}) p(x_{k-1}|Y_{k-1}) \tag{3.62}$$

Substituting 3.62 and 3.61 into 3.60:

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i) p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tag{3.63}$$

Equation 3.59 is known as Sequential Importance Sampling (SIS) and forms the basis of most of the MC methods nowadays [67], including the particle filters. It consists in propagating the importance weights 3.63 recursively, drawing the samples x_k^i from the importance density $q(x_k|Y_k)$ and using the incoming measurements at each epoch k.

3.4.4 Degeneracy Problem

In [68] it is shown that after a few iterations of the Sequential Importance Sampling Filter (SIS) Filter, only a few particles or one particle will have an associated $w_k^i \neq 0$. This is analogous to the convergence of the covariance estimate in an EKF and means that only a few samples will gain weight, therefore a lot of computing effort will be wasted updating the weights of the remaining particles. This is called the degeneracy problem and is addressed by resampling the particles after computing the associated weights. In [69] a suitable measure of degeneracy is presented, called effective sample size:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (\bar{w}_k^i)^2} \tag{3.64}$$

It is noted that the effective sample size takes a value between $1 < \hat{N}_{eff} < N$, being N in the case that all the weights are uniform, meaning that all particles are contributing to the estimate (low degeneracy), and 1 in the case there are many particles with weight one and the rest with zero weight (high degeneracy). The idea is to perform a resampling algorithm when $\hat{N}_{eff}$ is below a certain threshold $N_T = N_s/2$. Many resampling algorithms have been introduced [70], the systematic resampling algorithm (often called minimum variance sampling) is one of the most efficient and popular resampling algorithms as it is simple to implement and its computational complexity is $O(N)$ [71]. This method treats the weights as continuous random variables in the interval (0,1), which are randomly ordered. The spacing interval (0,1) is divided into N divisions, then the number of samples is counted in each division using the weight's cumulative sum of the particles. In this way, large-weight particles are likely to be counted many times, and therefore replicated more. This process is schematically shown in Figure 3.1. The particles with large weights have a good chance of being selected and resampled, particles 3 and 7 (blue bars) have significantly larger weights than the rest, having more chance to be multiplied.

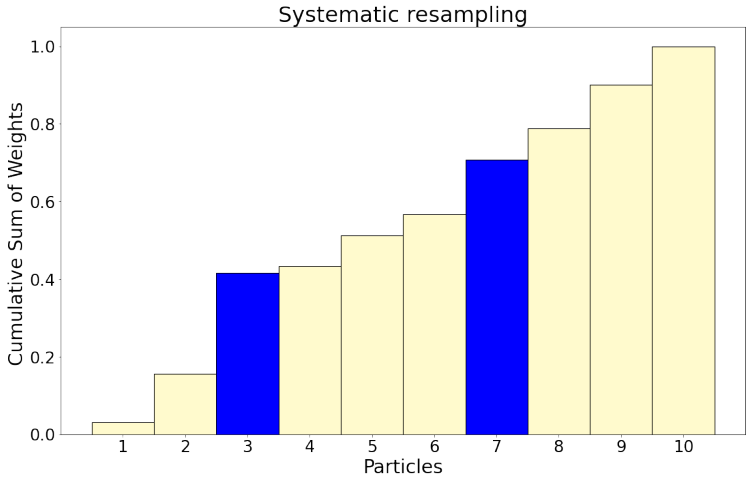


Figure 3.1: Schematic description of the resampling process. Blue bars represent the particles with significantly larger weights than the rest, increasing their chance to be multiplied.

3.4.5 Choice of importance density

For simplicity, the choice of importance density $q(x_k^i|x_{k-1}^i, y_k)$ is the transitional prior given by the dynamic system model 3.18 and the associated process noise Q [64]:

$$q(x_k^i|x_{k-1}^i, y_k) = p(x_k^i|x_{k-1}^i) \tag{3.65}$$

Considering this choice of importance density, the normalized importance weights in (3.63) are as follows:

$$w_k^i \propto w_{k-1}^i p(y_k|x_k^i) \tag{3.66}$$

The main limitation of the transitional prior as importance density is that the filter would not work if this distribution is much broader than the likelihood $p(y_k|x_k^i)$, resulting in rapid degeneration of the particles. The optimal choice of importance density was introduced in [72] and [73], it minimizes the variance of the importance weights conditional upon x_{k-1}^i and yk:

$$q(x_k^i|x_{k-1}^i, y_k) = p(x_k^i|x_{k-1}^i, z_k) \tag{3.67}$$

Applying the Bayes' theorem (3.43) to the optimal choice for importance density:

$$p(x_k^i|x_{k-1}^i, z_k) = \frac{p(y_k^i|x_k, x_{k-1}^i)p(x_k|x_{k-1}^i)}{p(y_k|x_{k-1}^i)} \tag{3.68}$$

then the importance weights are as follows:

$$w_k^i \propto w_{k-1}^i p(y_k|x_{k-1}^i) \tag{3.69}$$

This means that the importance weights can be computed before the particles are propagated through the system model (3.18), however, it has to be possible to draw samples from $p(x_k^i|x_{k-1}^i, z_k)$ which is not a straightforward task. If we consider a linear measurement model, non-linear dynamics model, and Gaussian pdfs for all the elements in the model, then the use of the optimal choice for the importance density as $p(x_k^i|x_{k-1}^i, z_k)$ is possible and would be Gaussian [74]. Applying the Bayes' rule to the optimal choice for importance density:

$$p(x_k^i|x_{k-1}^i, y_k) = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|x_{k-1})} \tag{3.70}$$

Considering Gaussian distributions and dynamics 3.18 and measurement 3.19 models, the transitional density and the measurements likelihood are as follows:

$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; f(x_{k-1}), P_{k-1})) \tag{3.71}$$

$$p(y_k|x_k) = \mathcal{N}(y_k; H_k xk, R_{k-1})) \tag{3.72}$$

Replacing equation (3.71) and (3.72) into equation (3.70), the numerator of the right side of the equation is:

$$\frac{1}{\sqrt{2\pi R_{k-1} P_{k-1}}} \exp\left(\frac{(y_k - H_k x_k)^T R_{k-1}^{-1}(y_k - H_k x_k)}{2} - \frac{(x_k - f(x_{k-1}))^T P_{k-1}^{-1}(x_k - f(x_{k-1}))}{2}\right) \tag{3.73}$$

Expanding the terms in the exponent, we have:

$$(y_k - H_k x_k)^T R_{k-1}^{-1}(y_k - H_k x_k) + (x_k - f(x_{k-1}))^T P_{k-1}^{-1}(x_k - f(x_{k-1})) \tag{3.74}$$

$$\begin{aligned} y_k^T R_{k-1}^{-1} y_k - x_k^T H_k^T R_{k-1}^{-1} y_k - y_k^T R_{k-1}^{-1} H_k x_k + x_k^T H_k^T R_{k-1}^{-1} H_k x_k + x_k^T P_{k-1}^{-1} x_k \\ -f(x_{k-1})^T P_{k-1}^{-1} x_k - x_k^T P_{k-1}^{-1} f(x_{k-1}) + f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1}) \end{aligned} \tag{3.75}$$

Factorizing :

$$\begin{aligned} x_k^T (H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1}) x_k - x_k^T (H_k^T R_{k-1}^{-1} y_k + P_{k-1}^{-1} f(x_{k-1})) \\ -(y_k^T R_{k-1}^{-1} H_k + f(x_{k-1})^T P_{k-1}^{-1}) x_k + y_k^T R_{k-1}^{-1} y_k + f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1}) \end{aligned} \tag{3.76}$$

Making the following replacements into Equation (3.76) :

$$\begin{aligned} A_k^{-1} &= H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1} \\ y_k &= y_k - H_k f(x_{k-1}) = y_k - b_k \\ a_k &= f(x_{k-1}) + A_k H_k^T R_{k-1}^{-1}(y_k - H_k f(x_{k-1})) \end{aligned} \tag{3.77}$$

we have:

$$x_k^T(A_k^{-1})x_k - x_k^T A_k^{-1} a_k + y_k^T R_{k-1}^{-1} y_k + f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1}) \tag{3.78}$$

completing the squared binomial by adding the term $a_k^T A^{-1} a_k$:

$$(x_k - a_k)^T A_k^{-1}(x_k - a_k) + m_k \tag{3.79}$$

where m_k:

$$m_k = -a_k^T A_k^{-1} a_k + y_k^T R_{k-1}^{-1} y_k + f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1}) \tag{3.80}$$

considering (3.77) and expanding, m_k is as follows:

$$\begin{aligned} &-(y_k - H_k f(x_{k-1}))^T R_{k-1}^{-1} H_k A^{-1} H_k^T R_{k-1}^{-1}(y_k - H_k f(x_{k-1})) - f(x_{k-1})^T H_k^T R_{k-1}^{-1} y_k \\ &+f(x_{k-1})^T H_k^T R_{k-1}^{-1} H_k f(x_{k-1}) - y_k^T H_k f(x_{k-1}) + f(x_{k-1})^T H_k^T R_{k-1}^{-1} H_k f(x_{k-1}) \\ &-f(x_{k-1})^T (H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1}) f(x_{k-1}) + y_k^T R_{k-1}^{-1} y_k + f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1}) \end{aligned} \tag{3.81}$$

substituting the squared binomial $(y_k - H_k f(x_{k-1}))^T R^{-1}(y_k - H_k f(x_{k-1}))$:

$$\begin{aligned} &-(y_k - H_k f(x_{k-1}))^T R_{k-1}^{-1} H_k A_k^{-1} H_k^T R_{k-1}^{-1}(y_k - H_k f(x_{k-1})) + (y_k - H_k f(x_{k-1}))^T R_{k-1}^{-1}(y_k - H_k f(x_{k-1})) \\ &+\cancel{f(x_{k-1})^T H_k^T R_{k-1}^{-1} H_k f(x_{k-1})} - \cancel{f(x_{k-1})^T (H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1}) f(x_{k-1})} + \cancel{f(x_{k-1})^T P_{k-1}^{-1} f(x_{k-1})} \end{aligned} \tag{3.82}$$

factorizing:

$$\begin{aligned} &= (y_k - H_k f(x_{k-1}))^T [R_{k-1}^{-1} - R_{k-1}^{-1} H_k A_k^{-1} H_k^T R_{k-1}^{-1}](y_k - H_k f(x_{k-1})) \\ &= (y_k - H_k f(x_{k-1}))^T [R_{k-1}^{-1} - R_{k-1}^{-1} H_k (H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1}) H_k^T R_{k-1}^{-1}](y_k - H_k f(x_{k-1})) \end{aligned} \tag{3.83}$$

According to the Woodbury matrix identity (or matrix inversion lemma [75] , [76]):

$$R_{k-1}^{-1} - R_{k-1}^{-1} H_k (H_k^T R_{k-1}^{-1} H_k + P_{k-1}^{-1}) H_k^T R_{k-1}^{-1} = H_k P_{k-1}^{-1} H_k^T + R_{k-1} = B_k^{-1} \tag{3.84}$$

then

$$m_k = (y_k - H_k f(x_{k-1}))^T B_k^{-1} (y_k - H_k f(x_{k-1})) \tag{3.85}$$

replacing m_k (3.85) in Equation (3.79):

$$(x_k - a_k)^T A_k^{-1} (x_k - a_k) + (y_k - H_k f(x_{k-1}))^T B_k^{-1} (y_k - H_k f(x_{k-1})) \tag{3.86}$$

Therefore, the exponential term (3.74) is equal to (3.86):

$$\begin{aligned} &(y_k - H_k x_k)^T R_{k-1}^{-1} (y_k - H_k x_k) + (x_k - f(x_{k-1}))^T P_{k-1}^{-1} (x_k - f(x_{k-1})) \\ &= (x_k - a_k)^T A_k^{-1} (x_k - a_k) + (y_k - H_k f(x_{k-1}))^T B_k^{-1} (y_k - H_k f(x_{k-1})) \end{aligned} \tag{3.87}$$

Which is Bayes' rule from 3.68:

$$p(y_k|x_k) p(x_k|x_{k-1}) = p(y_k|x_{k-1}) p(x_k^i|x_{k-1}^i, y_k) \tag{3.88}$$

Equation (3.88) shows that the optimal importance density and $p(y_k|x_{k-1})$ are Gaussian under the assumptions mentioned before [64]:

$$p(x_k^i|x_{k-1}^i, y_k) = \mathcal{N}(x_k; a_k, A_k)) \tag{3.89}$$

$$p(y_k|x_{k-1}) = \mathcal{N}(y_k; b_k, B_k)) \tag{3.90}$$

where, according to Equations (3.77) and (3.84), a_k,A_k, b_k and B_k are:

$$\begin{aligned} a_k &= f(x_{k-1}) + A_k H_k^T R_{k-1}^{-1} (y_k - b_k) \\ A_k &= P_{k-1} - P_{k-1} H_k^T B_k^{-1} H_k P_{k-1} \\ b_k &= H_k f(x_{k-1}) \\ B_k &= H_k P_{k-1}^{-1} H_k^T + R_{k-1} \end{aligned} \tag{3.91}$$

These expressions are the update recursions for the Kalman Filter. The analytic solution above is only possible under the assumptions of having Gaussian distributions and a linear observation model. For the rest of the cases, an analytic solution is not possible, nevertheless is possible to approximate the transitional prior (suboptimal choice) (3.66) to the optimal importance density using local linearization [56]. The local linearization encourages the particles to be in the regions of high likelihood by incorporating the observations at time k through a bank of EKFs or UKFs [66].

3.4.6 SIR Particle Filter with Local Linearization

Choosing the optimal importance density is only possible for linear measurement models and Gaussian distributions, and the solution under these assumptions turns out to be the Kalman Filter update step. As mentioned in section 3.2, the EKF approximates the nonlinear case through a first-order Taylor series expansion and computes the mean and the covariance of the state. Within the particle filter framework, the EKF propagates each particle according to its Gaussian solution resulting in a *better* importance density function, with particles closer to regions of high likelihood [59].

$$q(x_k^i|x_{k-1}^i, y_k) = \mathcal{N}(\hat{x}_k{}^i, \hat{P}_k^i) \tag{3.92}$$

Although this additional step incorporates the observations at time k and computes the EKF solution to each sample, this process is not equivalent to performing the update of the particles, it only approximates the importance density function to the optimal choice. It is important to note that performing the local linearization on the transitional prior density $p(x_k^i|x_{k-1}^i)$ introduces a different importance density rather than the prior, therefore the weights have to be computed through Equation (3.63) instead of (3.66). Under Gaussian assumptions and a linear measurement model case, the local linearization step will propagate the particles to describe accurately the posterior distribution. In this case, performing the weighting/resampling step afterward will result in the same distribution of particles, as the local linearization step propagates the particles to the solution. The proof of the fact that performing this additional step does not update the particles twice after weighting the particles is explained by replacing the importance density as the posterior in Equation 3.60:

$$w_k^i = \frac{p(x_k|Y_k)}{q(x_k|Y_k)} \approx \frac{p(x_k|Y_k)}{p(x_k|Y_k)} \approx 1_k^i \tag{3.93}$$

As the EKF is only a first-order Taylor series approximation to the posterior, it introduces inaccuracies due to linearization and Gaussian assumptions of the posterior. Due to this, applying the local linearization using the EKF in highly non-Gaussian cases and highly non-linear measurements may not be advantageous and the filter can lose performance. The UKF is a third-order approximation of the posterior and addresses the limitations of the EKF, therefore, the UKF is a better option to approximate the particles to the optimal choice of importance density. The particle filter that uses the UKF or the EKF to perform the local linearization is called the Unscented Particle Filter (UPF) or Extended Kalman Particle Filter (EKPF)[59].

3.4.7 Unscented Based Particle Filter

As mentioned in the previous Subsection 3.4.6, applying the local linearization propagates the particles towards the likelihood function $p(y_k|x_k^i)$, improving the performance of the SIR Particle Filter and allowing to reduce the number of particles used for the estimation. The additional computational of performing local linearization to the prior is often more than offset by the reduction in the number of particles [64]. Nevertheless, the local linearization focuses on improving the accuracy of the estimation of the particle filter by improving the importance density and not reducing the computational cost, i.e. the EKF or the UKF is performed for each particle. In [77], an alternative method is presented based on the same concept as Local Linearization, which significantly improves both the computational cost and the accuracy of the estimation. Suppose the distribution of particles at time $k-1$ is available from $p(x_{k-1}|Y_{k-1})$, therefore is possible to compute the estimate $\hat{\mathbf{x}}_{k-1}$ and its empirical covariance matrix $\hat{\mathbf{P}}_{k-1}$ through:

$$\begin{aligned} \hat{\mathbf{x}}_{k-1} &= \sum_{i=1}^{N_p} w_{k-1}^i \mathbf{x}_{k-1}^i \\ \hat{\mathbf{P}}_{k-1} &= \frac{\sum_{i=1}^{N_p} (\mathbf{x}_{k-1}^i - \hat{\mathbf{x}}_{k-1}) \otimes (\mathbf{x}_{k-1}^i - \hat{\mathbf{x}}_{k-1})}{N_p - 1} \end{aligned} \tag{3.94}$$

The idea is the same as the local linearization step, to move the particles towards the likelihood function, but the UKF (or EKF) is performed only once to the prior $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{P}}_{k-1}$ to compute the Kalman gain. The Kalman gain times the innovation $\mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$ is then used as a correction term to the transition prior density to move the particles towards the likelihood. The new set of particles is the importance density $q(x_k^i|x_{k-1}^i, y_k)$ and the associated weights for the particles are computed through Equation 3.63. This method incorporates the measurements y_k performing the UKF once instead of N_p times and approximates the importance density to the optimal choice. This method is more computationally efficient than the UPF or the EKPF and more accurate than the SIR Particle Filter. This estimator is known as the Unscented Based Particle Filter (UBPF) in the case the UKF is applied to compute the Kalman gain [77]. The SIR particle filter steps are shown in the following diagram: The complete process of estimating a dynamic state using the UKF is described as follows:

Algorithm 3 Particle Filter Algorithm

Require: Initial particles $S_0 = \{x_0^{(i)}, w_0^{(i)}\}_{i=1}^N$, process model f, choice of importance density q, measurement y_k, systematic resampling method

for $k = 1, 2, ...$ **do**
  for $i = 1, 2, ..., N$ **do**
    Sample particle $x_k^{(i)} \sim f(x_{k-1}^{(i)})$
    Calculate importance weight $w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)}$
  end for
  Normalize weights $\hat{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}$
  Resample particles $S_k = \{x_k^{(i)}, \hat{w}_k^{(i)}\}_{i=1}^N$ using resampling method
  Calculate estimated state $\hat{x}_k = \sum_{i=1}^N \hat{w}_k^{(i)} x_k^{(i)}$
  Calculate estimated state covariance $P_k = \sum_{i=1}^N \hat{w}_k^{(i)} (x_k^{(i)} - \hat{x}_k)(x_k^{(i)} - \hat{x}_k)^T$
end for
return estimated state $\hat{x}_k$ and estimated state covariance P_k

This algorithm describes the Sequential Importance Resampling Particle Filter, with the prior distribution as choice for the importance density. The prediction step is not included as it will be discussed in the next chapter.

3.5 Comparison between non-linear filters

In order to compare the output between the filters, I implemented them in a simple simulated case. This simple case is shown in Figure 3.2 and consists of the following:

- Is a static position estimation problem in one dimension. The state only consists of position in the x axis.
- The transition matrix is $\Phi = 1$, the initial position is $\hat{\mathbf{x}}_0^- = 10$,
- The measurement model is linear, the posterior of the state, and the measurement errors are Gaussian with measurement noise $R_k = 4$
- The observations are simulated ranges between a source and the stationary vehicle.
- All filters are initialized with the same values for the position state and its initial covariance matrix $\hat{\mathbf{P}}_0^- = 4$.

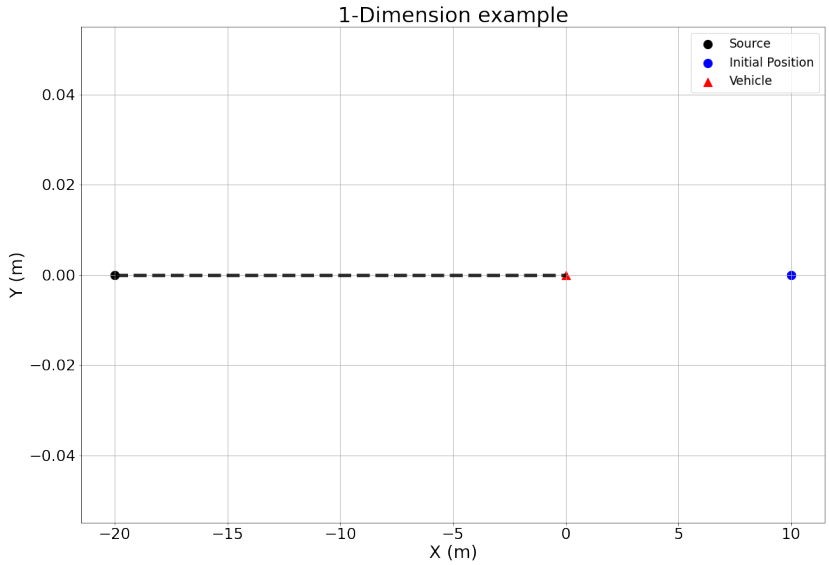


Figure 3.2: One-dimension estimation case. The *vehicle* (red triangle) is static, and the observations are ranges from the *source* (black circle) to the *vehicle*.

The measurements consist of simulated ranges from the Source (black dot) to the vehicle (red triangle), 20 metres at every epoch since the vehicle is not moving. In the particle filter case, the initial values are represented with a set of particles randomly sampled from a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_0^- = 10, \hat{\mathbf{P}}_0^- = 4)$. Showing the prior and the update step at the first epoch demonstrates the equivalence between the particle filter and the UKF to the Kalman filter under the simplest conditions mentioned above.

In Figure 3.3, the comparison between the UKF and the EKF is shown. The apriori of the state is shown in red, in the UKF case it can be noted that the mean and variance are described by a minimal set of deterministically chosen sample points. After the update step, both filters have the same output, the EKF solution is plotted in green and the UKF one in black. The UKF efficiently captures the prior and posterior distributions as the state is still represented as a Gaussian variable specified by the sigma points [59]. Both filters are equivalent under the conditions mentioned above.

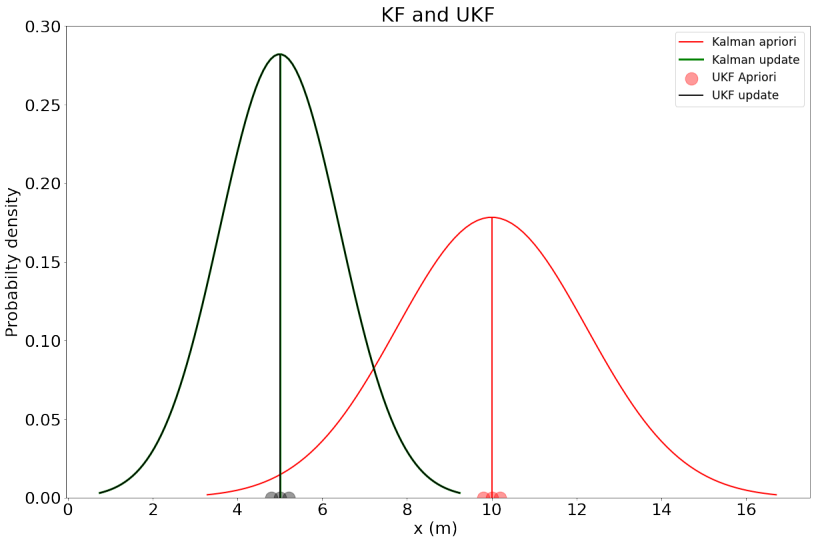


Figure 3.3: Comparison between the KF and UKF solutions at the first epoch of a one-dimension example.

Figures 3.4 and 3.5 show the comparison between the KF and particle filter solutions. In both plots, the histograms of the particle distributions are shown, the red histogram is the distribution before the update step, and the black histograms are for the particles after being weighted and resampled. Figure 3.4 shows the particle filter running with 50 particles, while the second figure 3.5 shows a 5000-samples particle filter. It can be noted that for this one-dimensional example, 50 samples are not enough amount to properly describe the pdf. Only a small number of particles are being resampled, which are the ones close to the likelihood area. For the second PF, the particles approximate better the posterior, therefore the estimation is equivalent to the KF solution. However, it can be noted on the left side of the distribution the lack of samples to describe perfectly the posterior. The performance of the particle filter is highly dependent on the number of particles. The Monte Carlo estimate of the integral described by equation 3.55, is the sample mean [64]:

$$I_N = \frac{1}{N} \sum_{i=1}^{N} f(x^i) \tag{3.95}$$

if the variance of $f(x)$ is finite, the following central limit theorem holds:

$$\lim_{N \to \infty} \sqrt{N}(I_N - I) \sim \mathcal{N}(0, \sigma^2) \tag{3.96}$$

Central limit theorem 3.96 states that the error of the Monte-Carlo estimate of the integral 3.54 is of order $N^{-1/2}$, therefore the rate of convergence of the MC integration decreases as the number of samples N

increases

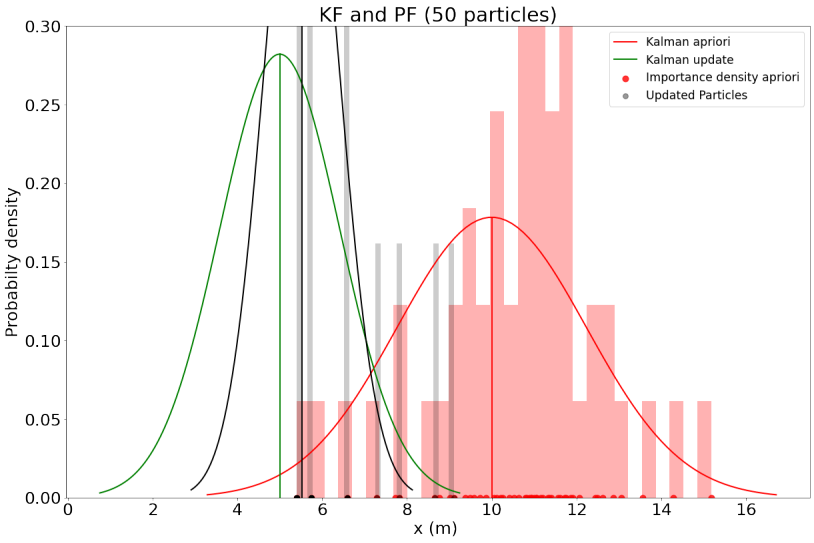


Figure 3.4: Comparison between the KF and Particle filter solutions at the first epoch of a one-dimension example. 50 particles are used in this PF. The black line fits the updated particle histogram for the comparison between both solutions.

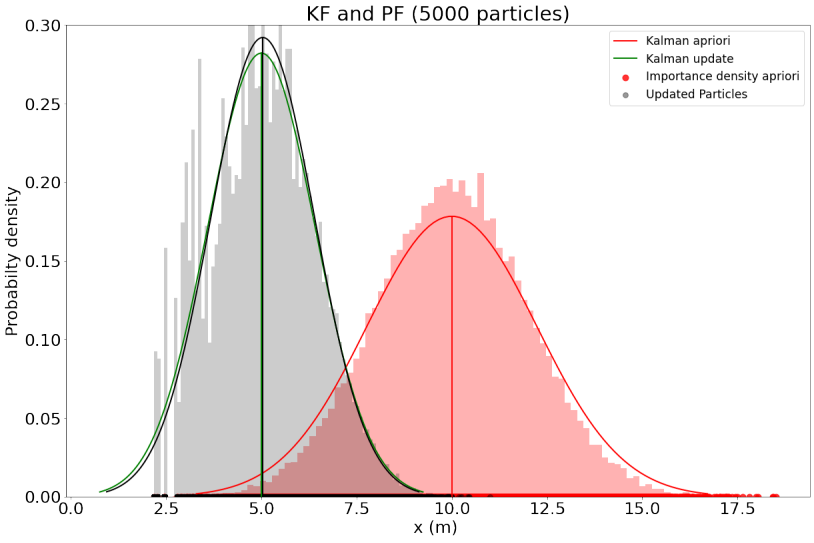


Figure 3.5: Comparison between the KF and Particle filter solutions at the first epoch of a one-dimension example. 5000 particles are used in this PF. The black line fits the updated particle histogram for the comparison between both solutions.

Figure 3.5 is a clear example of the main limitation of using the transitional prior as the importance

density, the lack of samples in the likelihood area, even when a large number of particles is used. In Figures 3.6 and 3.7, improved importance densities are used by performing local linearization and applying the Kalman gain as a correction term to the transitional prior respectively as mentioned in subsections 3.4.6 and 3.4.7. In both figures, the red histograms are the prior densities, the blue histograms are the improved importance densities, and the black ones are after the update step (weighting and resampling). In Figure 3.6 the importance density is approximate to the optimal choice by using local linearization applying the EKF to each particle. The resulting importance density is a set of particles already positioned around the solution, although it can be noted that the variance is larger than the posterior. By weighting this set of particles using Equation (3.63), we get the same solution as the Kalman filter, both Gaussians are overlapped.

In Figure 3.7, the UBPF is shown. The Kalman gain was computed once and used to move the particles from the prior density. It can be noted that the shapes of the blue and red histograms are the same, as the entire prior distribution was moved by $\mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$. The blue set is now the importance density which is the likelihood function area, resulting in a perfect fitting to the posterior after weighting and resampling the particles.

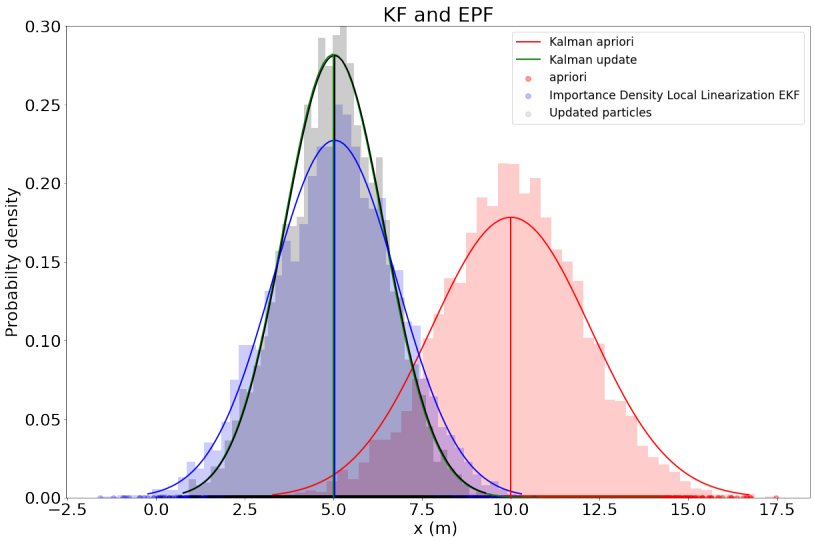


Figure 3.6: Comparison between the solutions estimated with the KF and Particle filter with local linearization using EKF at the first epoch of a one-dimension example. 500 particles are used in this filter.

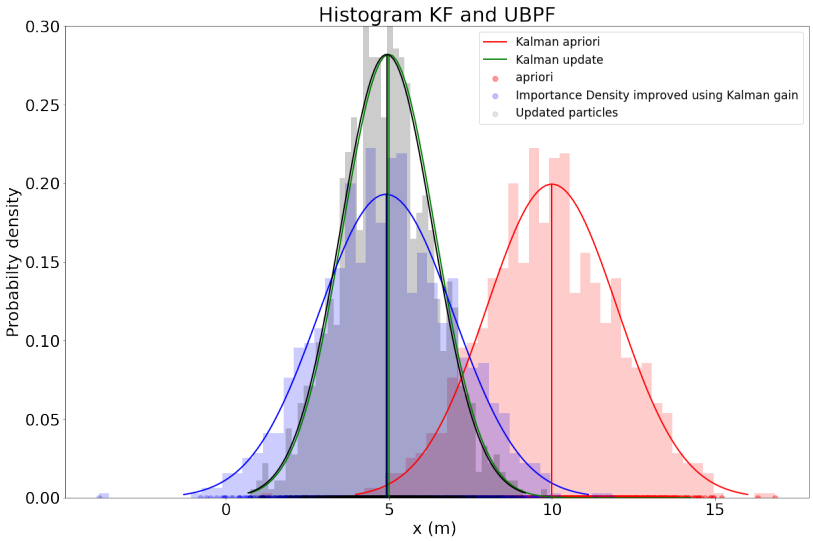


Figure 3.7: Comparison between the solutions estimated with the KF and the Particle filter with the importance density as the transitional prior moved to the likelihood function using the Kalman gain from the UKF. 500 particles are used in this PF.

In these comparisons, I have shown the performance of the SIR Particle Filter when using a few particles against a large number and the transitional prior as importance density for simplicity. As the PF is highly dependent on the number of particles to properly describe the posterior, the 5000-sample PF is significantly more accurate than the 50-sample one, although it is still limited when the likelihood function is far from the mean of the prior density. The solution is to improve the importance density by moving the prior towards the likelihood, two ways are explained and demonstrated to do so, local linearization and correcting the prior using the Kalman gain. The local linearization improves the accuracy of the estimation and allows for a reduction in the number of particles, although it is needed to perform N times either the EKF or the UKF. The UBPF improves the accuracy and the computational cost as well, as it performs the UKF (or EKF) only once on the estimate at time $k-1$, and uses the Kalman gain to move the entire set of particles from the transitional prior towards the likelihood function.

3.6 Rao-blackwellization

Local linearization improves estimation accuracy by moving the prior distribution towards the likelihood, resulting in a reduction of particles required for effective estimation. However, computing the Kalman gain for each particle and moving the particles based on it may cause a loss of the non-Gaussian distribution, which is one of the key advantages of particle filters. Another common approach is to use Rao-Blackwellization,

which involves reducing the size of the state space being computed by the particle filter and estimating the remaining states analytically or using other methods such as the Kalman Filter [70]. By reducing the dimension of the state space, fewer particles are needed and the computational cost decreases significantly. The first step is to make a partition of the state vector as follows:

$$x_k = \begin{bmatrix} x_k^l \\ x_k^{nl} \end{bmatrix} \tag{3.97}$$

where x_k^l and x_k^{nl} are the linear and nonlinear parts of the state respectively [78]. The partition considers that some of the states are linear and Gaussian distributed. In some satellite navigation applications, the nonlinear part of the state is the position state, and the rest are linear and Gaussian distributed [79]. Considering the partition 3.97, the state system and observation models can be reorganized as [80]:

$$\begin{aligned} x_{k+1}^{nl} &= f_k^{nl}(x_k^{nl}) + A_k^{nl}(x_k^{nl})x_k^l + w_k^{nl} \\ x_{k+1}^{l} &= f_k^{l}(x_k^{nl}) + A_k^{l}(x_k^{nl})x_k^l + w_k^{nl} \\ y_k &= h(x_k^{nl}) + C_k(x_k^{nl})x_k^l + n_k \end{aligned} \tag{3.98}$$

where the state noise w_k for both linear and nonlinear parts is Gaussian distributed $\mathcal{N}(0, Q_k)$ as:

$$Q_k = \begin{bmatrix} Q_k^l & Q_k^{lnl} \\ (Q_k^{lnl})^T & Q_k^{nl} \end{bmatrix} \tag{3.99}$$

The linear part of the state, described by the probability density function $p(x_k^l|x_k^{nl}, Y_k) = \mathcal{N}(\hat{x}_k^l, P_{k+1})$ (measurement update) is the solution to the Bayes rule under Gaussian assumptions described by the recursive set of Kalman Filter equations 3.91:

$$\begin{aligned} \hat{x}_k^l &= \hat{x}_{k-1}^l + K_k(y_k - h_k - C_k\hat{x}_k^l) \\ P_k &= P_{k-1} - K_k(C_kP_{k-1}C_k^T + R_k)K_t^T \\ K_k &= P_{k-1}C_k^T(C_kP_{k-1}C_k^T + R_k)^{-1} \end{aligned} \tag{3.100}$$

The prediction (time update) of the linear state described by the $p(x^l_{k+1}|x^{nl}_{k+1}, Y_k) = \mathcal{N}(\hat{x}^l_{k+1}, P_{k+1})$ pdf is as follows:

$$\begin{aligned}
\hat{x}^l_{k+1} &= \bar{A}^l_k \hat{x}^l_{k-1} + (Q^{lnl}_k)^T (Q^{nl}_k)^{-1} + L_k(x^{nl}_{k+1} - \hat{x}^{nl}_k - A^{nl}_k \hat{x}^l_k) \\
P_{k+1} &= \bar{A}^l_k P_k (\bar{A}^l_k)^T + \bar{Q}^l_k - L_k N_k L^T_k \\
L_k &= \bar{A}^l_k P_k (A^{nl}_k)^T N^{-1}_k \\
N_k &= A^{nl}_k P_k (A^{nl}_k)^T + Q^{nl}_k
\end{aligned} \tag{3.101}$$

where:

$$\begin{aligned}
\bar{A}^l_k &= A^l_k - (Q^{lnl}_k)^T (Q^{nl}_k)^{-1} A^{nl}_k, \\
\bar{Q}^l_k &= Q^l_k - (Q^{lnl}_k)^T (Q^{nl}_k)^{-1} Q^{lnl}_k
\end{aligned} \tag{3.102}$$

In the case where the measurement equation does not include any state from the linear partition, the Kalman Filter can not be used to update according to the received observation at the current epoch, this means that the Kalman measurement update step can be left out. However, all the observation information gets to the linear partition through the time update step, using the estimated states from the nonlinear part $\hat{x}^{nl}_k$ and their prediction $\hat{x}^{nl}_{k+1}$ as measurements. It can be noted from equations (3.101) and (3.102) that the correlation between the nonlinear and linear partitions are considered through the Q^{lnl}_k and Q^{nl}_k matrices. Both additional steps, measurement and time updates are applied after having the particles resampled (measurement update for the nonlinear part of the state) using Equations (3.59) and (3.63).

3.6.1 Rao-Blackwellized PF example

A simple 2-dimensional example is carried out to compare the performance of the SIR-PF and the Rao-Blackwellized Particle Filter (RBPF). The example consists of a 4-state estimation, 2 states for P_x and P_y positions, and 2 states for their corresponding velocities V_x and V_y. The measurements consist of ranges generated from three different locations as it is shown in Figure 3.8.

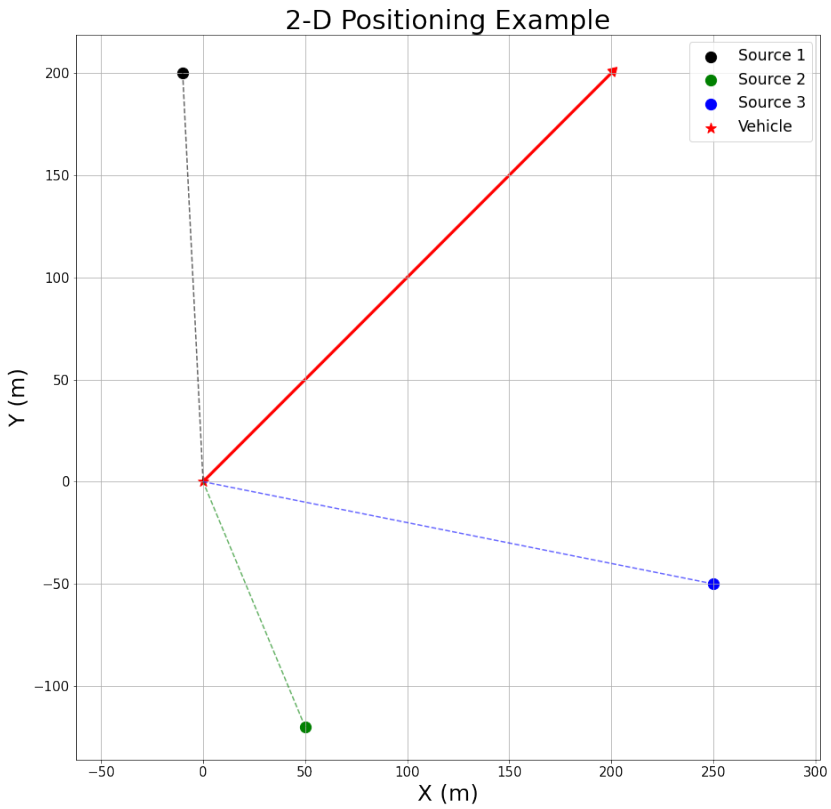


Figure 3.8: 2-Dimension positioning example.

The system model can be written as follows:

$$x_{k+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + w_k \tag{3.103}$$

and the measurement model as:

$$y_k = h(x_k) + n_k \tag{3.104}$$

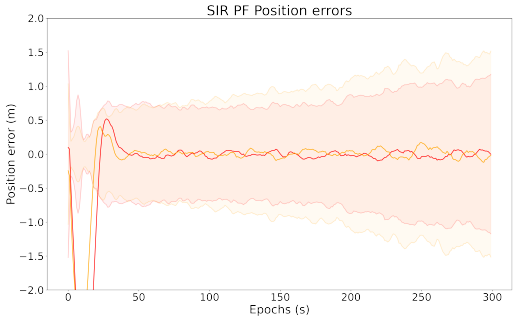
As the measurement model is nonlinear and only includes the position states, the velocity states are linear

and assumed Gaussian and can be estimated using the KF, therefore the state can be split as follows:

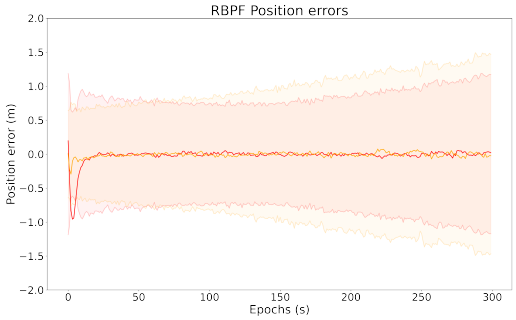
$$\begin{aligned} x_{k+1}^{nl} &= x_k^{nl} + A_k^{nl}(\Delta t) + w_k^{nl} \\ x_{k+1}^{nl} &= x_k^{l} + w_k^{l} \\ y_k &= h_k(x_k^{nl}) + n_k \end{aligned} \tag{3.105}$$

where the nonlinear part of the state x_k^{nl} are the positions P_x and P_y and the linear part x_k^l the velocity states V_x and V_y. The nonlinear states are being updated using the SIR PF, while the linear part is being estimated with the KF. As the measurement model does not involve any state from the linear partition, these states are being updated recursively by the prediction step (time update step) using Kalman filter Equations (3.102). In Equation 3.102, it can be observed that the predicted states for the nonlinear part $\hat{x}^{nl}k+1$, are treated as "measurements". As a result, the time update of the linear partition is performed subsequent to the prediction step of the particle filter. Additionally, since $A^{nl}k$ and $A^{l}k$ are not dependent on $x^{nl}k$, it is sufficient to compute the covariance matrix of the linear part of the states P_k only once, instead of computing it for each particle. In Figure 3.9 the differences between the states estimated using both filters and the true values are shown with their respective standard deviation 2σ "envelope". This example is intended to compare the SIR PF and the RBPF algorithms. As such, a small number of particles were used to estimate the results, in order to highlight the difference in performance between the two methods. As illustrated in Figure 3.9, when using the same number of particles, the RBPF provides more accurate estimates for the position and velocity states compared to the SIR PF. This is primarily due to the RBPF's ability to occupy a lower dimensional space and utilize the Kalman Filter for optimal estimation of linear and Gaussian states. The main purpose of the RBPF is to decrease the computational load (number of particles) that the PF has when estimating a large dimension state vector. It has been shown that the SIR PF represents accurately any distribution as long as sufficient number of particles are involved in the process, this means that it provides the same result as the RBPF if enough particles are being used. In [81], a comparison of multiple tests of the RBPF is carried out by varying the number of particles and using different partitions of the state. The author employed range and azimuth measurements from a radar system to estimate the position, velocity, and acceleration states in an aircraft navigation scenario. The assessment involved calculating the root mean square error (RMSE) while varying both the number of particles and the number of states within the nonlinear partition. For a complete state estimation utilizing the particle filter with 2393 particles, the RMSE for the position state is 7.07. Conversely, in situations where the particle filter is used to estimate the position state, while the Kalman filter estimates the remaining states with a mere 264 particles, the RMSE increases to 7.27. It's noteworthy that a comparable RMSE is achieved even with significantly fewer particles

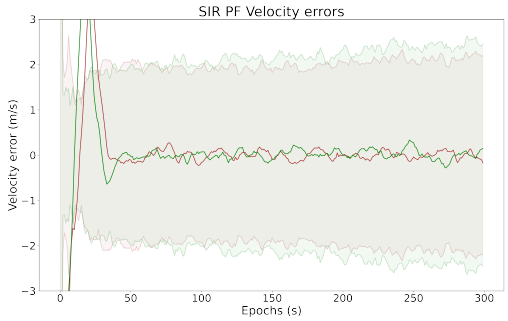
when marginalizing the velocity and acceleration states. This observation underscores the effectiveness of Rao-Blackwellization.



(a) Position states estimates using the SIR PF.



(b) Position states estimates using the marginalized PF.



(c) Velocity states estimates using the SIR PF

(d) Velocity states estimates using the marginalized PF.

Figure 3.9: Comparison between the position and velocity states estimated with the SIR PF and the RBPF

The most common non-linear estimators are explained in this chapter. The PF is described in detail, providing mathematical concepts and proofs to show its equivalence to a Minimum Mean-Square Error estimator. It has been shown that the optimal solution to the estimation problem, under a Bayesian framework and Gaussian assumptions, are the measurement update and predict steps of the Kalman Filter. By providing a mathematical description of the PF in a Bayesian context, the similarities to other estimators become easily appreciated. The SIR PF with the transitional prior as the importance density is used for Chapters 4 and 5. However, in Chapter 6, the Rao-Blackwellized particle filter is used due to the outlier detection step implemented, which generates a predicted distribution of measurements and increases the computational load.

Chapter 4

SIR Particle Filter in float solution for ambiguity resolution

Chapter 3 is a detailed description of the SIR particle filter, its features, and 3 available techniques in order to increase its efficiency; 2 based on choosing a better importance density, and the other method splits the state vector into linear and nonlinear parts, and estimates the linear part with the KF (Rao-Blackwellization). This chapter shows the implementation of the SIR PF to the estimation of the full geometry-based float solution. Firstly, the filter implementation details are explained in section 4.1 followed by a comparison of the results obtained with this filter and the ones estimated with the EKF. Then, in section 4.3, an assessment of the performance of the float solution estimated with each filter for searching the integer ambiguities is made by applying the three integer search methods described in Chapter 2. This thesis exclusively employs L1 code and carrier GPS measurements. This choice is driven by the objective of maintaining a minimized dimension for the float solution while also reducing computational load. The occurrence of a limited number of satellites in view primarily arises in challenging environments with poor signal reception. Consequently, the analysis conducted within this study simulate scenarios resembling such challenging conditions.

4.1 SIR PF implementation details

Recursive filters for state estimation consist of two stages: prediction and update. In the prediction stage, the state probability density function is forecasted for the subsequent epoch using the state dynamics model described by the dynamics model. The update stage then uses the observations at each epoch to correct the predicted pdf. According to Bayesian theory, the pdf of the state is constructed based on all available

information, including the set of received measurements [64]. The particle filter (PF) is a sequential Monte Carlo approach that represents the posterior density of the state with a set of random samples with associated weights, rather than assuming that the posterior density is Gaussian and parametrized by mean and variance-covariance matrix. For a discrete dynamical system defined by Equations 3.18 and 3.19, the Sequential Importance Sampling (SIS) PF recursively estimates the state $\mathbf{x}k$ at time k by constructing the pdf $p(\mathbf{x}k|Y_k)$ using Bayes' rule [65]:

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \tag{4.1}$$

where $p(y_k|Y_{k-1})$ is the normalizing constant depending on the likelihood function $p(y_k|x_k)$ defined by the observations model. The SIS recursive estimation propagates the importance weights w_k^i (Equation 3.63) associated with each particle as each measurement is received sequentially. The sample points x_k^i are then updated accordingly. To simplify computation of the weights, the importance density $q(x_k^i|x_{k-1}^i, y_k)$ is chosen to be the transitional prior $p(x_k^i|x_{k-1}^i)$, as demonstrated in the previous chapter.

To avoid the degeneracy problem, a systematic resampling algorithm is implemented at each epoch, and a large number of samples are used. This choice of importance density and the resampling algorithm derive the SIR PF from the SIS filter. In contrast, the extended Kalman filter (EKF) utilizes a first-order Taylor series approximation of the nonlinear function and approximates $p(x_k|Y_k)$ to be Gaussian. However, when the true density is highly non-Gaussian, the EKF may not describe it accurately.

In this Chapter, the SIR PF is used to estimate the full geometry-based float solution and evaluate and compare its performance with the commonly used EKF under different conditions. The full geometry-based state vector is defined as follows

$$\mathbf{x} = (r_i, v_i, a_i, N_{L1}) \tag{4.2}$$

where r_i, v_i, a_i are the rover station position, velocity, and acceleration respectively and N_{L1} are the between receiver single difference (SD) ambiguities. SD ambiguities are used to simplify the base-satellite book-keeping as well as the observation covariance matrix. However, the double difference (DD) equations (**??**) and (**??**) to weigh the particles and eliminate atmospheric effects and the clock biases [82]. Prior to the ambiguity resolution step, the between-satellite difference operator (see Appendix D) to transform the SD ambiguities into DD ambiguites is used. For this thesis, only GPS L1 was used in order to keep as small as possible the state vector, but this approach is easily generalized to multiple frequencies and constellations.

4.1.1 Filter initialization

The initial set of particles is sampled from a Gaussian distribution $\mathcal{N}(0, Q_k)$. The position and velocity of the receiver are computed from a single-point positioning using pseudorange and Doppler observations through a least-squares estimator as follows:

$$\hat{x}_0 = (H^T W H)^{-1} H^T W (y - h(x)) \tag{4.3}$$

where W is a diagonal matrix containing the standard deviations of the measurements errors, H is a partial derivatives matrix of the measurement vector function $h(x)$ with respect to x:

$$H = \frac{\partial h(x)}{\partial x}|_x \tag{4.4}$$

The initial real-valued ambiguities are computed by subtracting the carrier-phase observations minus the pseudoranges $\lambda\hat{N} = P - \Phi$.

4.1.2 Prediction

A constant acceleration dynamics model augmented by random walk ambiguity states is adopted to match the dynamics model used in [82]:

$$\Phi_{k+1} = \begin{bmatrix} I_{3x3} & I_{3x3}\Delta T & I_{3x3}\frac{\Delta T^2}{2} & 0 \\ 0 & I_{3x3} & I_{3x3}\Delta T & 0 \\ 0 & 0 & I_{3x3} & 0 \\ 0 & 0 & 0 & I_{mxm} \end{bmatrix} \tag{4.5}$$

where ΔT is the receiver sampling interval in seconds, I is an identity matrix and m is the number of SD observations.

We consider the covariance matrix of the process noise Q_k as:

$$Q_k = \begin{bmatrix} Q_{(p,v,a)9x9} & \\ & Q_{(N)mxm} \end{bmatrix} \tag{4.6}$$

where $Q_{(p,v,a)9x9}$ is the covariance matrix of the position, velocity and acceleration of the state defined

as:

$$Q_{(p,v,a)9x9} = G \cdot G^T \cdot \sigma_a^2 \tag{4.7}$$

with $G = [\frac{\Delta T^2}{2}{}_{1x3}, \Delta T^2_{1x3}, I_{1x3}]^T$, σ_a^2 as the acceleration process noise and $Q_{(N)mxm}$ as a diagonal matrix of process noise σ_N^2. For each epoch, the particles are updated according to the dynamics described by equation (4.5), and then random noise is added to the states of each particle according to the matrix L_k after applying the Cholesky decomposition to Q_k (see Appendix C):

$$Q_k = L_k \cdot L_k^T \tag{4.8}$$

The predicted state is then:

$$x_{k+1} = \Phi_{k+1} x_k + L_k \cdot \mathcal{N}(0,\, Q_k) \tag{4.9}$$

where $\mathcal{N}(0,\, Q_k)$ are random-generated values using the diagonal of Q_k.

This step ensures that the particles have correlated process noise added to them, rather than just adding uncorrelated variances of the diagonal values of the matrix Q_k.
It's important to note that this step is essentially the same as the prediction step in a Kalman Filter. In this step, the state for each particle is predicted using a dynamics model, and process noise is added to represent the error in the prediction.

4.1.3 Weighting

The GNSS code pseudorange and carrier phase are formulated as [38]:

$$P_r^i = \rho_r^i - (\delta t_r - \delta t_i)c + \delta_{ion} + \delta_{trop} + \varepsilon_c \tag{4.10}$$

and

$$\Phi_r^i = \rho_r^i - (\delta t_r - \delta t_i)c + \lambda N_r^i - \delta_{ion} + \delta_{trop} + \varepsilon_p \tag{4.11}$$

where ρ is the geometric distance between the satellite i and receiver r, δ_{ion} and δ_{trop} are the ionospheric and tropospheric delays respectively, δt_r and δt_i are the receiver and satellite clock error terms respectively and N is the ambiguity in cycles. In order to use the carrier phase for precise positioning, the integer ambiguity N must be determined.

For short baselines, the Double Difference (DD) method can be applied between receivers and satellites. This significantly reduces δ_{ion} and δ_{trop} and eliminates $\delta t_r - \delta t_i$ from equations (4.10) and (4.11), resulting

in DD code and phase measurements as follows:

$$\Delta\nabla P = \Delta\nabla\rho + \varepsilon_{\Delta\nabla P} \tag{4.12}$$

and

$$\Delta\nabla\Phi = \Delta\nabla\rho + \lambda\Delta\nabla N + \varepsilon_{\Delta\nabla\Phi} \tag{4.13}$$

where the operator $\Delta\nabla$ is $\Delta\nabla = \{(\cdot)_i - (\cdot)_j\}_r - \{(\cdot)_i - (\cdot)_j\}_b$, i and j are the indices for the receivers and r and b are indices for the satellites.

As stated in the previous section, we estimate SD ambiguities to simplify the base-satellite book-keeping, however, the observation models are defined with DD code and phase measurements as the atmosphere effects are significantly reduced and clock error terms are eliminated.

In the SIR PF, the measurement update step is accomplished by weighting each particle based on its agreement with the new observations. The misclosures of each particle given the code and phase observations at time k from equations (4.12) and (4.13) are:

$$\varepsilon_{\Delta\nabla P} = \Delta\nabla\rho - \Delta\nabla P \tag{4.14}$$

and:

$$\varepsilon_{\Delta\nabla\Phi} = \Delta\nabla\rho + \lambda\Delta\nabla N - \Delta\nabla\Phi \tag{4.15}$$

are evaluated for each particle. The weight of the particle is computed using the product of the probability of each observation's misclosure, based on the prior distribution of the observation errors. In this case, we have no particular prior distribution and assume Gaussian weights, however, other prior distributions of the observations could be used. Each observation's individual weight is computed as :

$$w^i_{k,P,\Phi} = \frac{1}{\sqrt{2\pi\sigma^2_{P,\Phi}(k+1)}} \exp\left(-\frac{\varepsilon^2_{\Delta\nabla P,\Phi}}{2\sigma^2_{P,\Phi}(k+1)}\right) \tag{4.16}$$

and then the particle's total weight is computed from the product of the weights for each observation:

$$w^i_k = \prod_{P,\Phi=1}^{N_{obs}} w^i_{k,P,\Phi} \tag{4.17}$$

The weights are normalized such that $\sum_{i=1}^{N_p} w_k^i = 1$ as:

$$\bar{w}_k^i = \frac{w_k^i}{\sum_{i=1}^{N_p} w_k^i} \tag{4.18}$$

This step is analogous to the measurement update step in an EKF but does not require linearization of the measurement model and is capable of using an arbitrary prior distribution of the measurement errors if one is available.

4.1.4 Resampling

Resampling the particles is suggested once the filter starts to high weigh just a few particles. We use the effective sample size N_{eff} as a suitable measure of degeneracy [69]. N_{eff} is estimated as follows:

$$\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s} (\bar{w}_k^i)^2} \tag{4.19}$$

where $\hat{N_{eff}} \leq N_s$, and a small $\hat{N_{eff}}$ indicates severe degeneracy. The resampling algorithm is performed when $\hat{N_{eff}}$ is below a certain threshold $N_T = N_s/2$. Many resampling algorithms have been introduced [70], in this research, the systematic resampling (also called minimum variance sampling) is employed. This method treats the weights as continuous random variables in the interval (0,1), which are randomly ordered. The space interval (0,1) is divided into N_s divisions, then the number of samples is counted in each division using the weight's cumulative sum of the particles. In this way, large-weight particles are likely to be counted many times, and therefore replicated more.

4.1.5 Empirical variance-covariance matrix

The EKF describes the posterior distribution with a mean and its covariance matrix, which is the input for the LAMBDA method to obtain the integer decorrelating matrix Z^T. While the SIR PF describes the posterior distribution of the particles, the mean and the covariance matrix can be easily obtained. The geometry-based float solution state vector x_k is estimated through the weighted mean of the particles as follows:

$$\overline{x_k} = \sum_{i=1}^{i=N_p} \bar{w}_k^i x_k^i \tag{4.20}$$

The variance-covariance matrix P_k of the state vector x_k is estimated based on the current distribution of the particles after the resampling step:

$$\hat{P}_k = \frac{\sum_{i=0}^{i=N_p} (x_k^i - \overline{x_k}) \otimes (x_k^i - \overline{x_k})}{N_p - 1} \tag{4.21}$$

4.2 Data and Software

4.2.1 Driving Dataset

The data set used in this thesis consists of just over one hour of driving in a mix of urban and suburban areas near the University of Calgary collected by C. Huang [3]. Although most of the route is suburban, there are some segments of data where the vehicle navigated through urban canyon environments. Although the same data set is utilized throughout this study, different segments of data were selected for analysis in each chapter. In Chapter 5, the focus is on evaluating the performance of a map-aided SIR PF in segments of data that represent various driving scenarios. In Chapter 6, the analyzed segments of data contain GPS observations which are likely to be distorted by multipath.

A data collection platform was built which consists of the following sensors (Table 4.1):

- CAN-BUS wheel odometer data logger with the output resolution of 1 km/h.
- Vision system originally developed by Bernhard Aumayer [83] consisting of two RGB cameras and a U-blox 6 receiver. The GPS receiver is used to ensure the shutter synchronization between the two cameras and provide time-tagged images [84].
- A reference trajectory consisting of a tightly-coupled Novatel SPAN-LCI RTK GNSS/INS forward and backward differential carrier-phase post-processing.

Table 4.1: Sensor Specification of the Data Collection Platform [3].

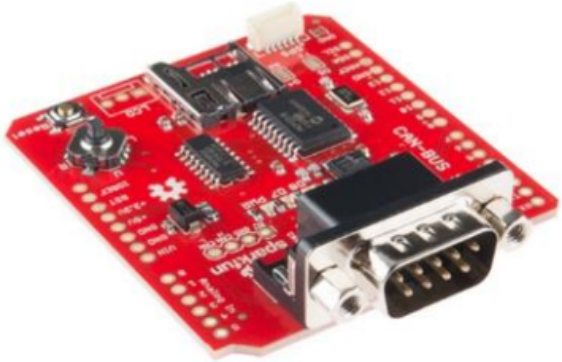
Sensor	**Manufacturer/Model**	**Accuracy**
Stereo RGB Cameras	PointGray / Blackfly-S GigE	
IMU	Xsens / MTi-600	12°/h
GPS/INS	Novatel / SPAN-LCI	0.06m +<1°/h
GPS	Ublox 6	2m
Wheel Odometer	Sparkfun - CAN-BUS Shield	1 km/h resolution

In Figure 4.1 a close-up look of the sensors used for collecting the data is shown.

(a) XSENS MTi-600 IMU.

(b) Novatel SPAN-LCI IMU.

(c) Novatel 702-gg Antenna.



(d) Sparkfun CANBUS Shield.

Figure 4.1: Senors used in the data set collection. Photos from [3]

The complete drive trajectory is shown in Figure 4.2.

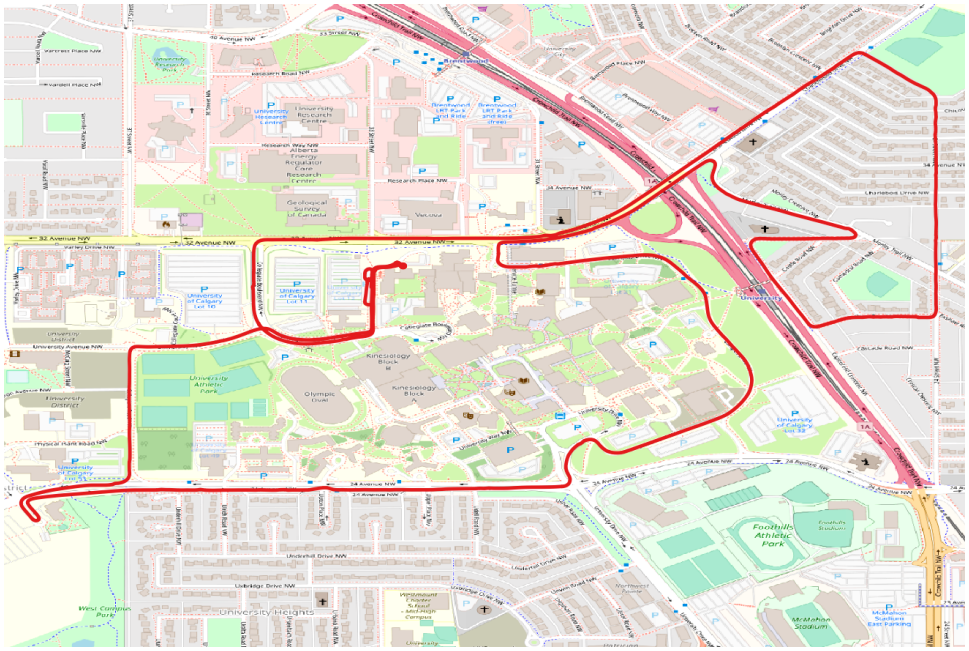


Figure 4.2: Drive trajectory of C. Huang dataset

4.2.2 Software for data processing

In order to process the GPS data and provide a RTK positioning solution, RTKLIB 2.4.2p (see Appendix A) was used. The application in its CUI version for performing post-processing analysis is RNX2RTKP. The SIR PF for computing the full geometry-based solution was programmed from scratch in Python programming language. As RTKLIB is developed in C, a script for integrating Python into a C program was written. Once the Python code is embedded in RNX2RTKP, the float solution can be estimated using the default EKF written in RTKLIB or the SIR PF developed for this thesis. The RTKLIB settings used in this thesis are displayed in Table 4.2.

Table 4.2: RTKLIB input settings used in this work.

Setting	**Value**
Positioning mode	RTK
Frequencies / Filter type	L1 / Forward
Elevation mask	15
SNR Mask	35 dBHz
Satelllite ephemeris clock	Broadcast
Constellations	GPS
Integer Ambiguity Resolution	Continuous
Reject treshold of GDOP	30
Reject treshold of innovation	1000

4.3 Results

The SIR particle filter described in the previous section was used to compute the full geometry-based float solution of the land-vehicle data collected by C. Huang [3].The data used for the estimation is logged from the U-BLOX 6 rover receiver, and the base station at the rooftop of Calgary Center for Innovation Technology (CCIT) building. The reference trajectory was obtained using the Novatel SPAN-LCI tightly coupled RTK GNSS/INS solution generated by the Inertial Explorer software. The main objective of this section is to compare the estimations obtained with the SIR PF and the EKF under the same optimal conditions for signal reception. In this way, equivalence between both filters is demonstrated, and further analysis of the particle filter performance in different scenarios or integration with other constraints is carried out in the subsequent chapters. In Figure 4.3 the 15 minutes analyzed segment of data is highlighted in red.

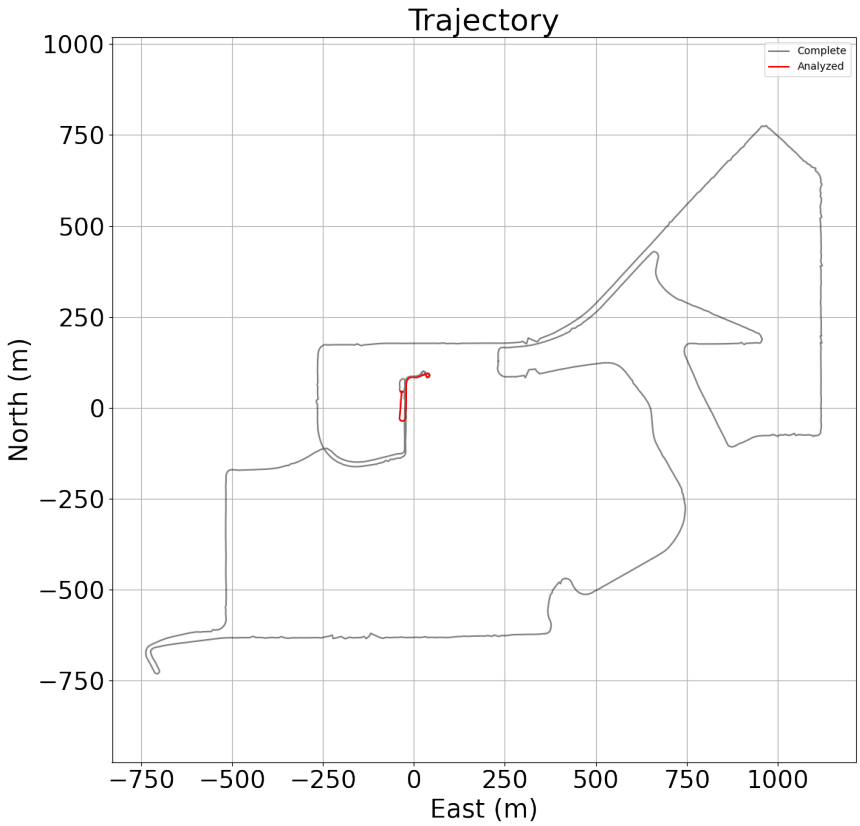


Figure 4.3: Analyzed trajectory (red line).

In order to show the position convergence through time, the differences between the reference trajectory and the float solutions estimated with the EKF and the PF are plotted in Figures 4.4 and 4.5 respectively in ENU directions (see Appendix B for coordinates transformation). The blue shaded areas wrapping the differences are the 2σ standard deviations of the computed solution.

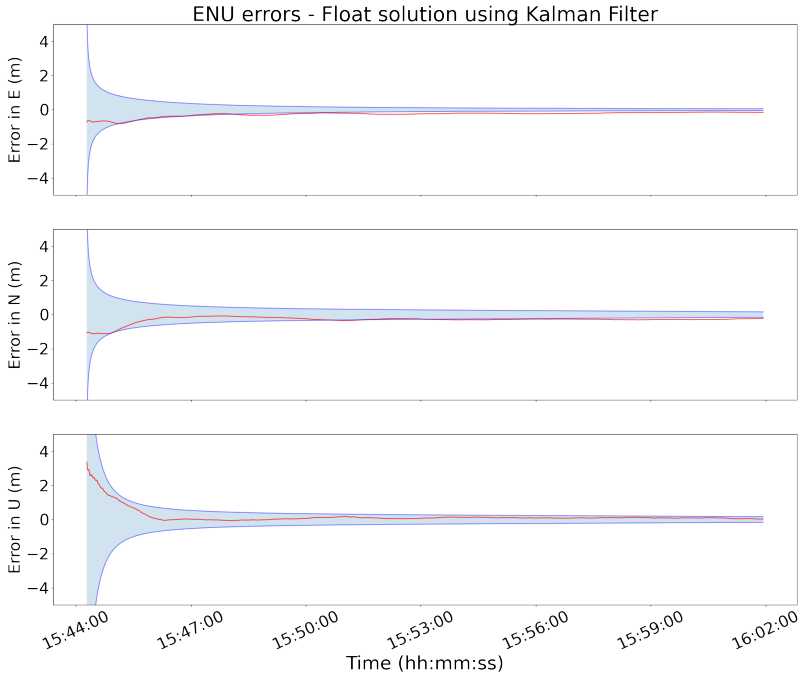


Figure 4.4: Position state errors in ENU directions computed with EKF.

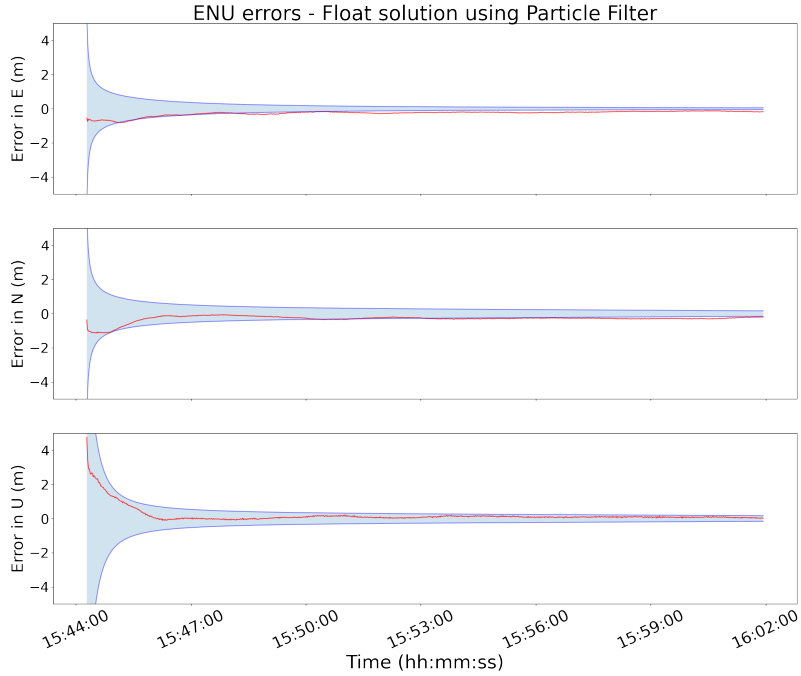


Figure 4.5: Position state errors in ENU directions computed with SIR particle filter.

In Figures 4.6 and 4.7 the Double-Difference float ambiguities estimated with the EKF and the PF respectively are shown. The plotted ambiguities are the difference between the real-valued ambiguities and their respective final integer values computed with LAMBDA using integer least-squares on the final epoch of the EKF float solution. It can be noted that the float ambiguities estimated with both filters are almost the same, however slight differences between their standard deviations can be noticed as these values, in the PF case, are estimated empirically using the particle distribution at each epoch using Equation (4.21). For both estimators, the float values do not converge to their respective integer values due to correlation and biases of the estimated float ambiguities.

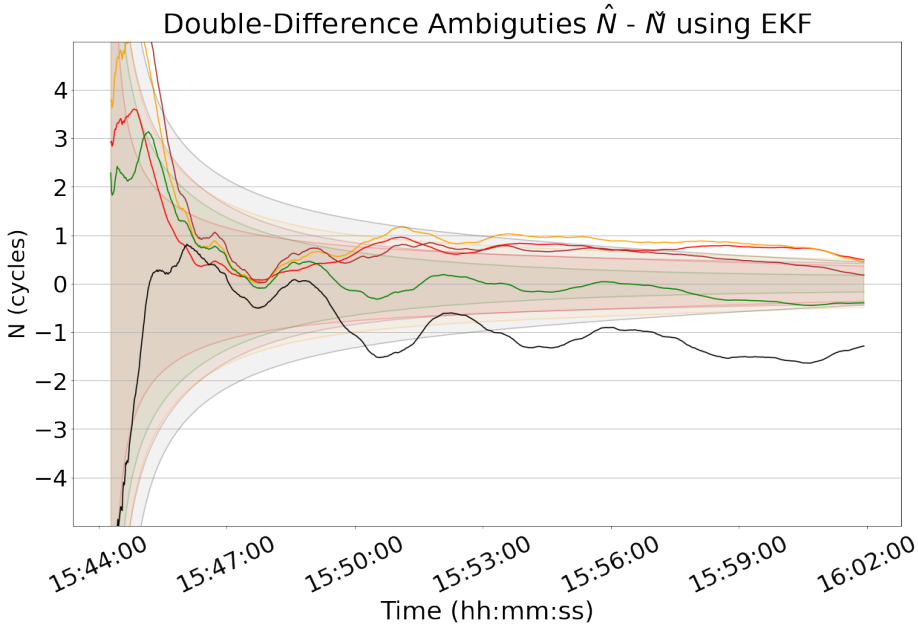


Figure 4.6: Float-valued ambiguities differences with respect their integer values using the EKF. The shaded areas are the 2σ standard deviation of the DD float ambiguities.

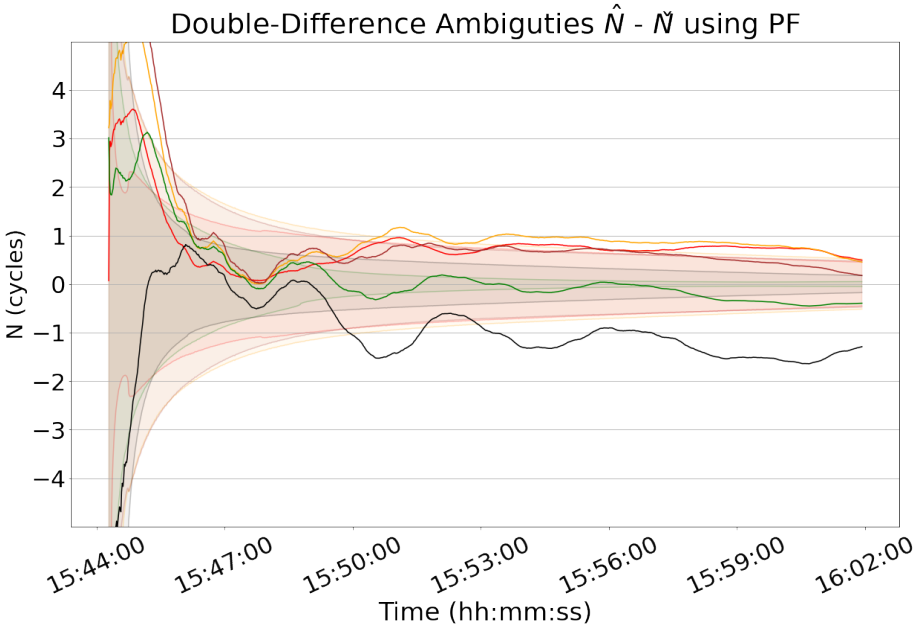


Figure 4.7: Float-valued ambiguities differences with respect their integer values using the particle filter. The shaded areas are the 2σ standard deviation of the DD float ambiguities.

4.3.1 Probability of correct fix

The probability of correct fixing for bootstrapping and rounding using the EKF and PF covariance matrices before and after decorrelating them using the LAMBDA transformation Z^T was computed to make a comparison between the levels of confidence of these two integer search methods. The probability of rounding $P(\check{N}_R = N)$ and bootstrapping $P(\check{N}_B = N)$ to the correct integer value is computed as follows [36]:

$$P(\check{N}_R = N) = \prod_{i=1}^{n} 2\Phi\left(\frac{1}{2\sigma_{\check{N}}}\right) - 1 \tag{4.22}$$

and

$$P(\check{N}_B = N) = \prod_{i=1}^{n} 2\Phi\left(\frac{1}{2\sigma_{\check{N}|I}}\right) - 1 \tag{4.23}$$

where

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}z^2)dz \tag{4.24}$$

From equations (4.22) and (4.23) can be noted that the success rate of both searching methods increases as the standard deviations $\sigma_{\check{N}}$ and the sequential conditional standard deviations $\sigma_{\check{N}|I}$ of the ambiguities become smaller. The computation of the success rates of the three ambiguity search methods using the SIR PF was carried out empirically by counting the success or failure of the method applied directly to each of the particles. This procedure can not be executed for the EKF as this filter provides only one solution along

with its covariance matrix, however, is possible to track the correct fix of the solution at each epoch using the three search methods (1 in case of correct fixing, 0 otherwise). In Figures 4.8 and 4.9, the success rates and the covariance based probability of correct fix of the three integer search methods using both filters are plotted. The probability of correct fixing for rounding and bootstrapping was computed through equations (4.22) and (4.23). As the probability of correct fix equation does not consider biases in the float solution, it will be greater than the success rate always assuming unbiased float estimates [85].

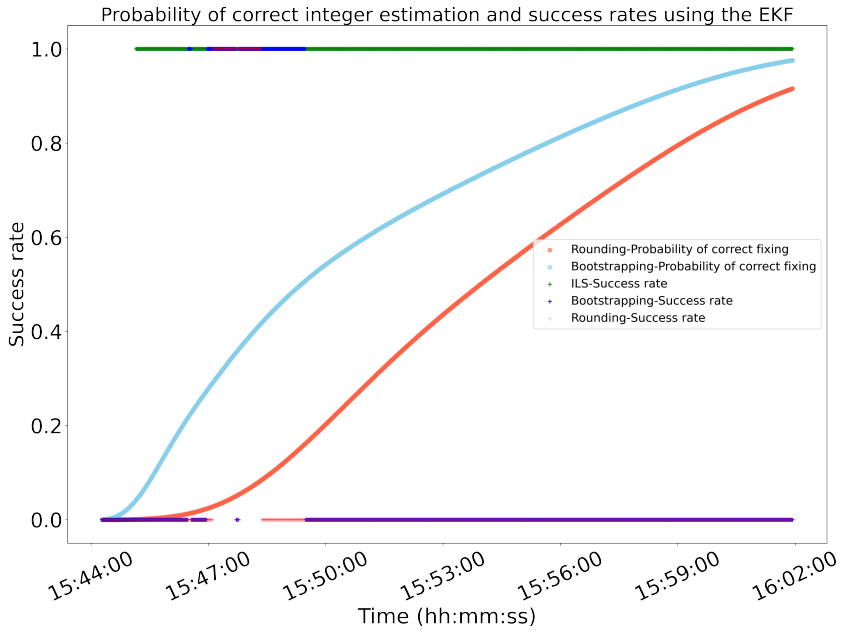


Figure 4.8: Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the EKF. The correct fixing for ILS (green), bootstrapping (light blue) and rounding (orange), 1 in case of correct fixing and 0 otherwise.

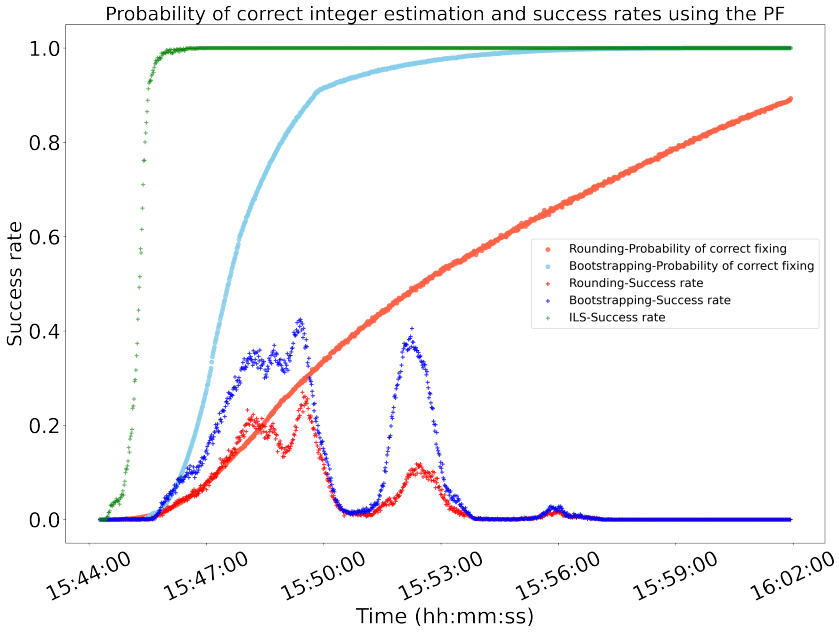


Figure 4.9: Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the SIR Particle Filter. Success rates of rounding (orange), bootstrapping (light blue) and ILS (green).

In Figures 4.10 and 4.11 the success rates and probabilities of correct fixing for both filters after decorrelation are shown. The performance of both rounding and bootstrapping improves considerably for both filters after decorrelation, however both are still impacted by biased ambiguity states. The ILS results are the same as those obtained without decorrelation, since the Z transformation does not change the ILS search criterion. However, the decorrelation significantly reduce the search space which is even more significant when using the SIR PF since each particle must be individually searched.

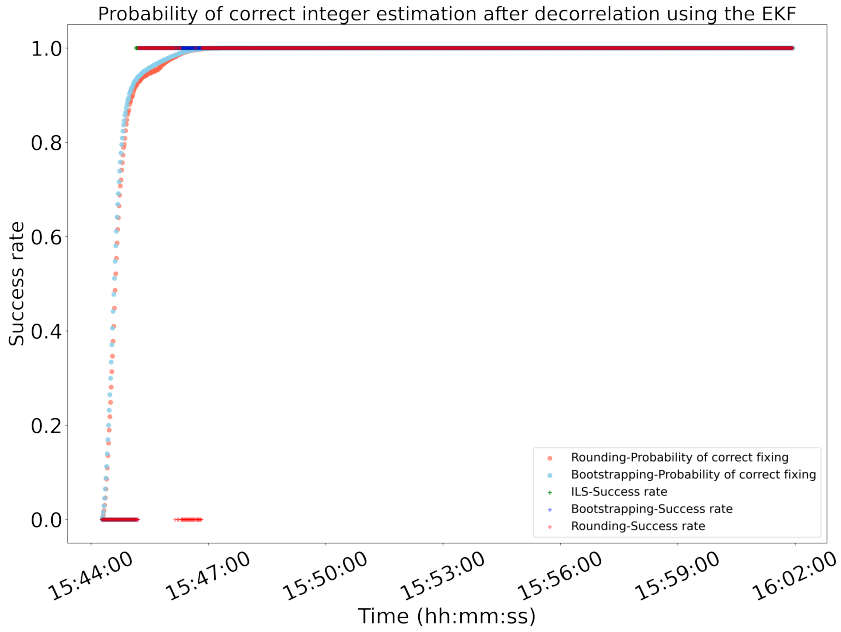


Figure 4.10: Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the EKF after LAMBDA decorrelation. The correct fixing for ILS (green), bootstrapping (light blue) and rounding (orange), 1 in case of correct fixing and 0 otherwise.

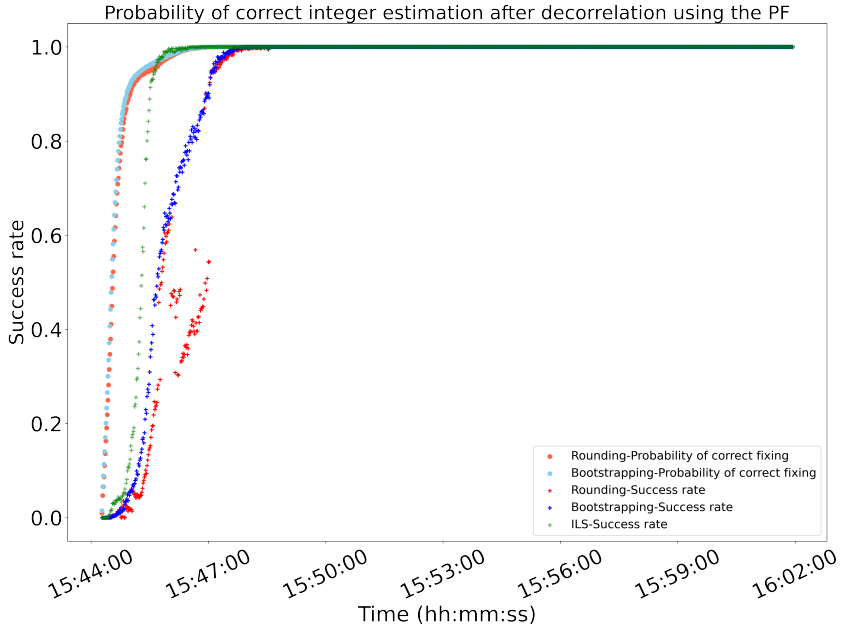


Figure 4.11: Probability of correct integer estimation of rounding (red) bootstrapping (blue) using the SIR particle filter after LAMBDA decorrelation. Success rates of rounding (orange), bootstrapping (light blue) and ILS (green).

4.3.2 Computation load

As described in Chapter 3, the particle filter estimates the state vector by representing the posterior density function through a set of weighted samples. As the pdf is characterized by a large number of samples, it is an equivalent representation to the usual functional description of the posterior pdf, and the particle filter will approach to the optimal Bayesian estimate [20]. For this comparison, 60,000 particles were used as the full geometry-based state vector estimated has a dimension of 9+ SD ambiguities. The particle filter implementation was written in Python code and embedded in the RTKLIB program A. The processing time in ambiguity resolution is of critical concern, being the most important limitation of a sample-based filter. In Figure 4.12, an estimation of he computational time when applying the ILS search method with and without the LAMBDA decorrelation is shown. The ILS was applied to a subset of 2,000 particles randomly selected from the 60,000 set of particles. This plot shows how the LAMBDA decorrelating transformation reduces considerably the computation time by reducing the search space, generally completing the process in .3 seconds while without the transformation takes up to 20 seconds at early epochs.

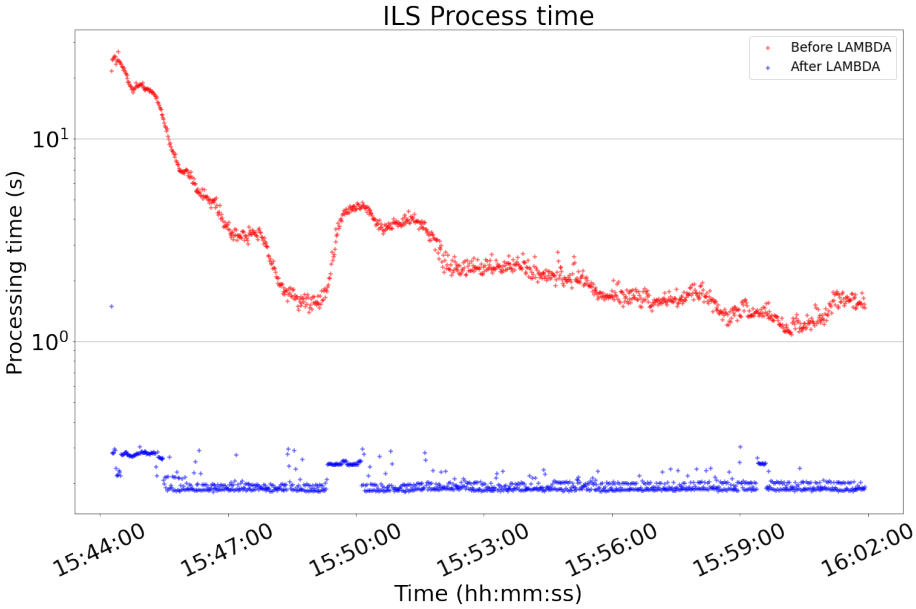


Figure 4.12: Processing time of Integer Least Squares applied to a set of particles before (red) and after (blue) decorrelation.

In this chapter, the implementation of the SIR PF for estimating the float solution is described. The results obtained are then compared to the ones obtained with the EKF. The three integer search methods were performed to both solutions and the success rates and an empirical probability of correct fixing were estimated. The success rates for the Bootstrapping and Integer rounding are computed using expressions 4.22 and 4.23, while the probability of correct fix is an actual count of the particles fixing to the correct

integer using the three integer search methods. It has been showed that the PF and the EKF obtain similar results under optimal conditions for signal reception. In Chapter 5, the SIR PF will be aided by a map-aiding algorithm using digital maps from the Open Street Map repository. This constraint leads no a non-Gaussian posterior distribution which is not possible to describe using the EKF. In Chapter 6 the performance of the PF in presence of non-Gaussian measurements is assessed. Such is the case when the GPS pseudoranges are distorted by multipath. The EKF assumes the measurements are Gaussian distributed, while the PF is able to weight the particles according to a any distribution.

Chapter 5

Map-aided Particle Filter for ambiguity resolution

In Chapter 3 an explanation of the most common nonlinear recursive filters for the state estimation problem is provided. Emphasis is placed on providing a detailed description of the Particle Filter with the resampling step implemented at each update and the transitional prior as the importance density (SIR Particle Filter). The previous chapter describes the implementation of the SIR PF for the estimation of the full geometry-based solution, followed by a comparison of the estimations obtained with the PF and the EKF. Both filters yield similar float solutions when the receiver is under optimal conditions for signal reception. However, the PF solution provides more information when resolving the ambiguities and offers an empirical probability of a correct fix. By using the PF as the float solution estimator, it becomes possible to empirically compute success rates by simply counting the success or failure of each particle. This also allows for the computation of a success rate for the ILS method. This chapter showcases the implementation of a map-aiding algorithm to improve the accuracy of the position state vector and the convergence rate of the float ambiguities within the PF. Section 5.2 provides a description of the digital road map information used in this thesis, specifically the Open Street Map (OSM) database and Java OSM, which is used for editing, creating road maps, and adding them to the OSM database. In Section 5.3, the implementation of the map-aiding algorithm is explained, followed by experimental results demonstrating its effect in both the position and ambiguity domains.

5.1 Map-matching

Map-matching is a process that improves navigation accuracy by integrating road maps with GNSS measurements to compute a more precise position state. In [19], a threefold classification of different map-matching algorithms is presented. They are categorized as semi-deterministic approaches [86], probabilistic approaches [87], and fuzzy-logic and belief theory-based approaches [88]. In this thesis, the map constraint falls into the probabilistic techniques group. Methods in this group calculate the probabilities of the vehicle traveling on different road segments and select the segment with the highest probability. By incorporating digital map information to enhance the navigation solution, highly non-Gaussian posterior densities are obtained, which can be accurately represented by the particle filter. The particle filter outperforms the EKF under such conditions [19].

One of the main advantages of using a sequential Monte Carlo-based estimator, such as the particle filter, with a map-matching constraint is its ability to capture multi-modal distributions. A multi-modal distribution occurs when there are multiple probable candidates for the road segments or lanes along which the vehicle is traveling. In Section 5.2, a description of the implemented multiple-lane road map, which provides an across-track position constraint, is provided. Once the across-track constraint is applied, the particles cluster around each candidate lane. The particle filter can quickly adapt if an initial prediction of the correct lane is found to be incorrect. However, it relies on the assumption that the vehicle is predominantly driven along the center of a lane.

5.2 Digital Maps

A digital road network is a representation of a set of nodes and edges organized into a directed graph [89]. Nodes in the graph correspond to intersections or dead-ends of road segments, while edges represent directed road segments connecting a starting node to a target node. These digital road maps are typically based on a two-dimensional geographical coordinate system, using latitude and longitude coordinates. In [90], the characteristics of a road network are further elaborated, involving nodes, edges, and shape points. Shape points are a sequence of nodes that approximate a curved road segment by representing it as a series of consecutive straight lines.

5.2.1 Open Street Map

In this thesis, we utilize the digital road maps available in Open Street Map (OSM) [91], which are stored in geographical coordinates. However, the navigable road segments in these maps do not include explicit

representations of multiple lanes as separate sets of edges or nodes. Instead, in OSM, roads with multiple lanes are typically represented by a single vector mapped along the center lane. Figure 5.1 illustrates an example of a T-intersection between divided suburban roads. In this example, the original OSM map describes multiple lanes within a road segment using a single vector. The intersection itself is modeled as a rectangle, with additional segments representing right-turning lanes.

Although the OSM database includes a field specifying the number of lanes, spatially it is represented by a single vector. When estimating the full geometry-based solution using carrier-phase and pseudorange observations, a more detailed road map is necessary. Utilizing the original map as a constraint to the solution can introduce bias compared to not using the map-aiding algorithm, particularly when the vehicle deviates from the edge representing the road. Digital road maps from OSM contain various topology features and rules (such as the representation of T-intersections or one vector representing multiple lanes within a road segment), which are useful for route finding or vehicle navigation when position accuracy is not critical. However, they may not accurately represent the actual paths taken by vehicles. For example, in the turning scenario depicted in Figure 5.1, a vehicle turning from Westbound to Southbound would follow a curved path, but the original OSM map represents the turn as a right-angle turn at the intersection of the road segments.

To address these limitations, a more detailed version of the map, as shown in Figure 5.2, is utilized in the estimation process. This enhanced map includes corresponding lanes added to the road segments where the estimation is performed, while the turning segments at the intersection are removed. Removing the turning segments improves the map-aiding algorithm described in Section 5.3, enabling the filter to select the most probable lane from accurately located candidates. However, this removal of turning segments eliminates the topology required for route finding.

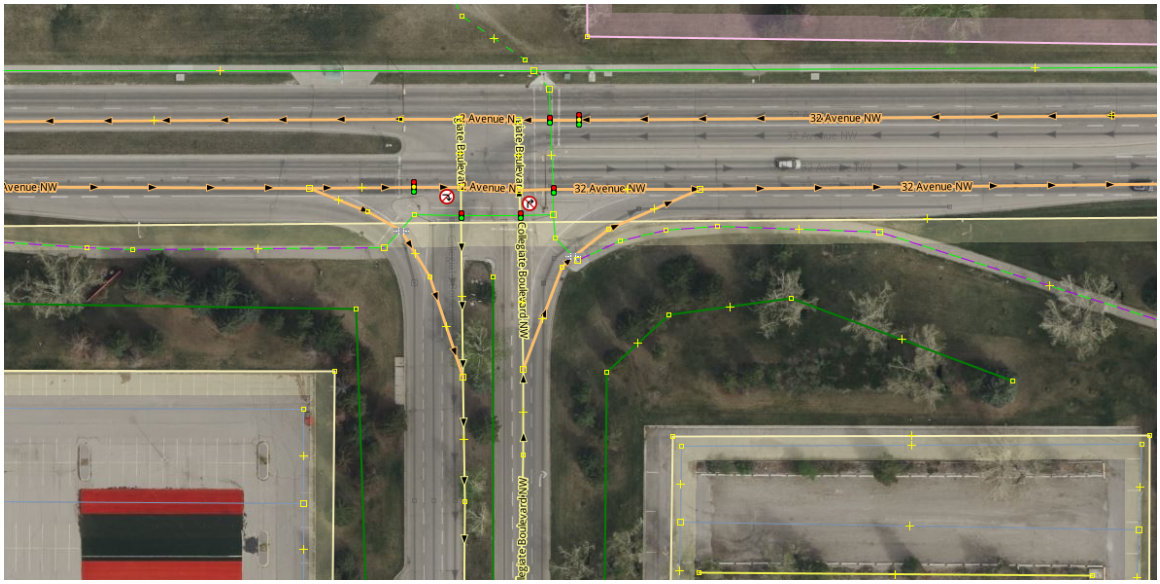


Figure 5.1: OSM original road map.

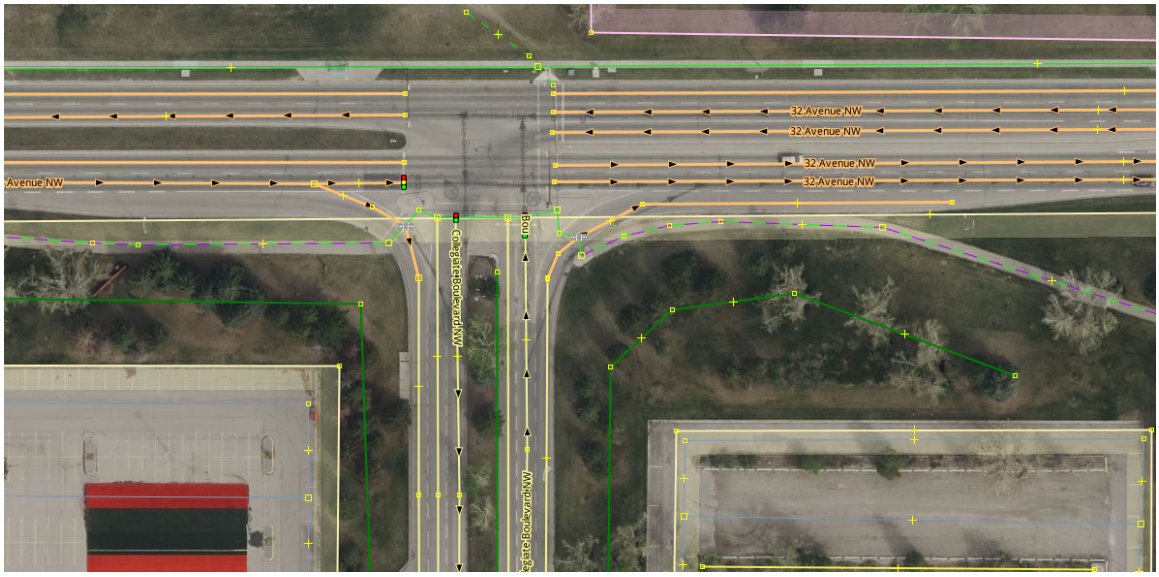


Figure 5.2: Example of a segment of digital road map modified for the purposes of this thesis.

5.2.2 Software

To modify, create, or delete digital maps in the OSM repository, the Java OSM (JOSM) software is required. JOSM is an extensible editor specifically designed for OSM, utilizing Java 8. It is an open-source software that supports various functionalities, including the loading of GPX tracks, background imagery, and OSM data from local or online sources. JOSM serves as the official software for editing digital maps in the OSM repository, allowing users to edit nodes, ways, relations, topology, and metadata information. It can be

compared to Geographic Information System (GIS) software, as both work with referenced geographic information layers and utilize basic geometry features such as points (nodes), vectors, and shapes, as illustrated in Figure 5.3. JOSM is a multiplatform software, and its installation requirements mainly depend on having an up-to-date version of Java. For detailed information about the functions and tools provided by JOSM, refer to the comprehensive documentation available in [92].

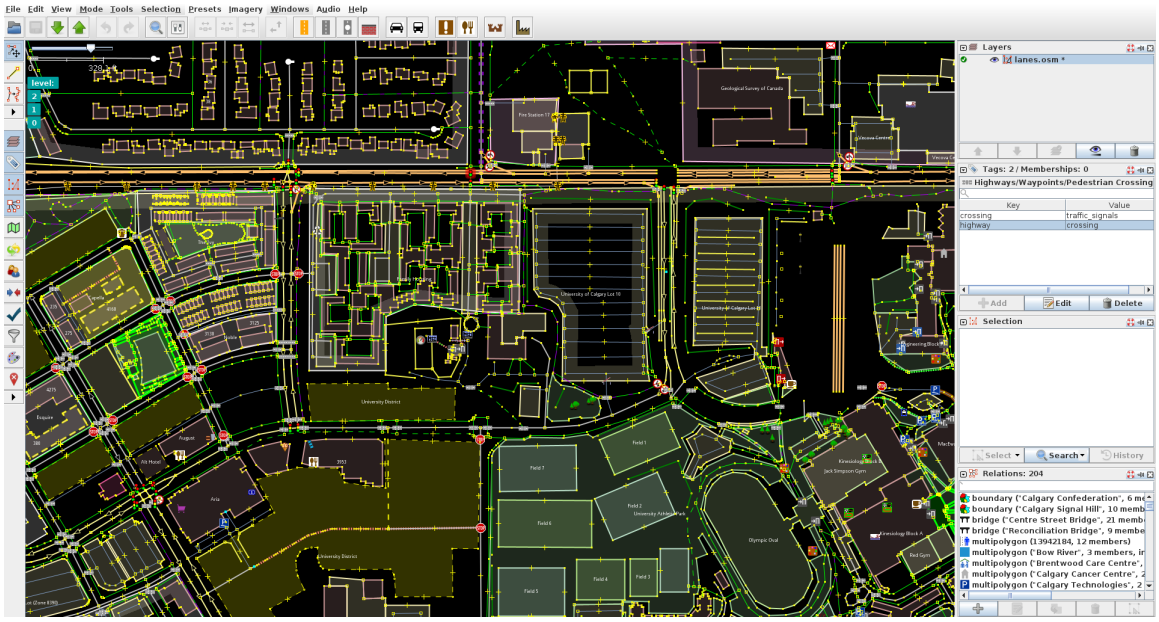


Figure 5.3: Java OSM software interface.

The map-aiding algorithm was implemented in the same Python particle filter routine used for the purposes of Chapter 4, which can be replaced for the standard EKF used in the **rnx2rtkp** module from RTKLIB.

5.3 Map-aiding Algorithm

The main objective of the PF is to approximate the posterior density of the state at each time step by using a set of random samples with associated weights, incorporating all available information, including the received measurements [64]. By applying Bayes' rule, the posterior can be recursively estimated using the likelihood function $p(y_k|x_k)$, which is defined by the observation model, the predicted probability density function $p(x_k|Y_{k-1})$, and a normalizing term $p(y_k|Y_{k-1})$ (Equation (3.43)). The posterior density at time k, represented by a set of random samples with associated weights, can be approximated using Equation (3.59), which is repeated here for convenience. Therefore, the posterior density is approximated by the following

expression:

$$p(x_k|Y_k) \approx \sum_{i=1}^{N_p} w_k^i \delta(x_k - x_k^i) \tag{5.1}$$

where the normalized weights are:

$$w_k^i \propto \frac{p(x_k|Y_k)}{q(x_k|Y_k)}$$

where $q(x_k|Y_k)$ is the convenient but suboptimal choice $q(x_k|X_{k-1}Y_k)q(X_{k-1}|Y_{k-1})$, which is an augmentation of the prior density with the new state x_k. Therefore, considering this choice for importance density, the normalized importance weights are as follows:

$$w_k^i \propto w_{k-1}^i p(y_k|x_k^i) \tag{5.2}$$

where $p(y_k|x_k^i)$ is the likelihood function which will depend on GPS observations and digital map information. The implemented map-aiding algorithm is designed to assign weights to the particles based on the most probable lane of the road segment on which the vehicle is traveling. This probability takes into account both the Euclidean distance between the vehicle and its nearest lane, as well as the difference in direction between the lane and the vehicle's trajectory. By weighting the particles according to their distance and direction difference from the lanes, the algorithm ensures that particles positioned on the most probable lane with the correct direction are assigned higher weights.

In Figure 5.1, the digital map does not explicitly describe the permitted turns at the intersection of two roads through a set of edges or nodes. Therefore, when the vehicle is making a turn, the map-aiding algorithm is not applied, and the particles are weighted solely based on the GPS observation misclosure using Equation (4.16):

$$w_{k,P,\Phi}^i = \frac{1}{\sqrt{2\pi\sigma_{P,\Phi}^2(k+1)}} \exp\left(-\frac{\varepsilon_{\Delta\nabla P,\Phi}^2}{2\sigma_{P,\Phi}^2(k+1)}\right) \tag{5.3}$$

To compute the Euclidean distances between the vehicle and its closest edge, the particles (estimated in earth-centered Cartesian coordinates) and the digital map (represented in curvilinear geodetic coordinates) were transformed into a local geodetic East-North-Up (ENU) frame. This transformation was performed using the nearby reference station as the center of the ENU frame. The details of this coordinate transformation can be found in Appendix B. The heading angle of the vehicle was obtained from its velocity vector and compared with the bearing of the nearest road segment. The bearing represents the direction of the road segment. Each particle weight considering the nearest lane and the correct direction of the edge according

to the vehicle's velocity is computed as follows:

$$w^i_{k,\Delta x,\Delta\alpha} = \frac{1}{\sqrt{2\pi\sigma^2_{\Delta x,\Delta\alpha}(k+1)}} \exp\left(-\frac{\Delta^2_{x,\alpha}}{2\sigma^2_{\Delta x,\Delta\alpha}(k+1)}\right) \tag{5.4}$$

where Δx and $\Delta\alpha$ are the distance between each particle to its closest lane and the difference between the directions of the lane and the particle. The particle's total weight considers both GPS measurements and the digital road map and is computed from the product of each GPS observation 4.16 and the map-aid 5.4:

$$w^i_k = w^i_{k,\Delta x,\Delta\alpha}\left(\prod_{P,\Phi=1}^{N_{obs}} w^i_{k,P,\Phi}\right) \tag{5.5}$$

The weights of the particles are then normalized using Equation (4.18). By incorporating equations (4.16), (5.4), and (5.5) to weigh the particles, the estimation of the vehicle's position is based on the GPS observations at each time step, aided by the map constraint that considers the most probable lane of a road segment, taking into account the correct direction and the distance between the vehicle and multiple lanes of the road. The Particle Filter utilized to assess the impact of the map constraint on the position states and the convergence of the ambiguities is the same Sequential Importance Resampling (SIR) PF described in Chapter 4. The only difference lies in the measurement update step, which incorporates the map-aiding algorithm.

5.3.1 Position domain

The map-aided Sequential Importance Resampling (SIR) Particle Filter described in this chapter was applied to compute the geometry-based float solution for the dataset discussed in Section 4.2 of Chapter 4. Figure 5.4 illustrates the road segments that were analyzed to demonstrate the effect of the map-aiding algorithm on the estimated position state. Each road segment represents a different scenario of the vehicle's trajectory. The complete dataset is displayed as a solid black line, while specific segments of data are highlighted for analysis. The red trajectory represents a segment where the vehicle makes a turn, the yellow trajectory represents a segment where the vehicle changes lanes, and the green, blue and brown trajectories represent segments where the vehicle travels in a straight direction, which is the simplest case. A circle is used to indicate the starting point of each analyzed segment.

The analyzed segments of data are characterized by an optimal environment for GPS signal reception, and the estimation of the position state tends to converge within the first minute of the GPS phase and pseudorange observations without the map-aiding. Since the objective of implementing a map constraint is to improve the convergence time, the analysis focuses on the first thirty seconds after initializing the filter for each segment

of data.

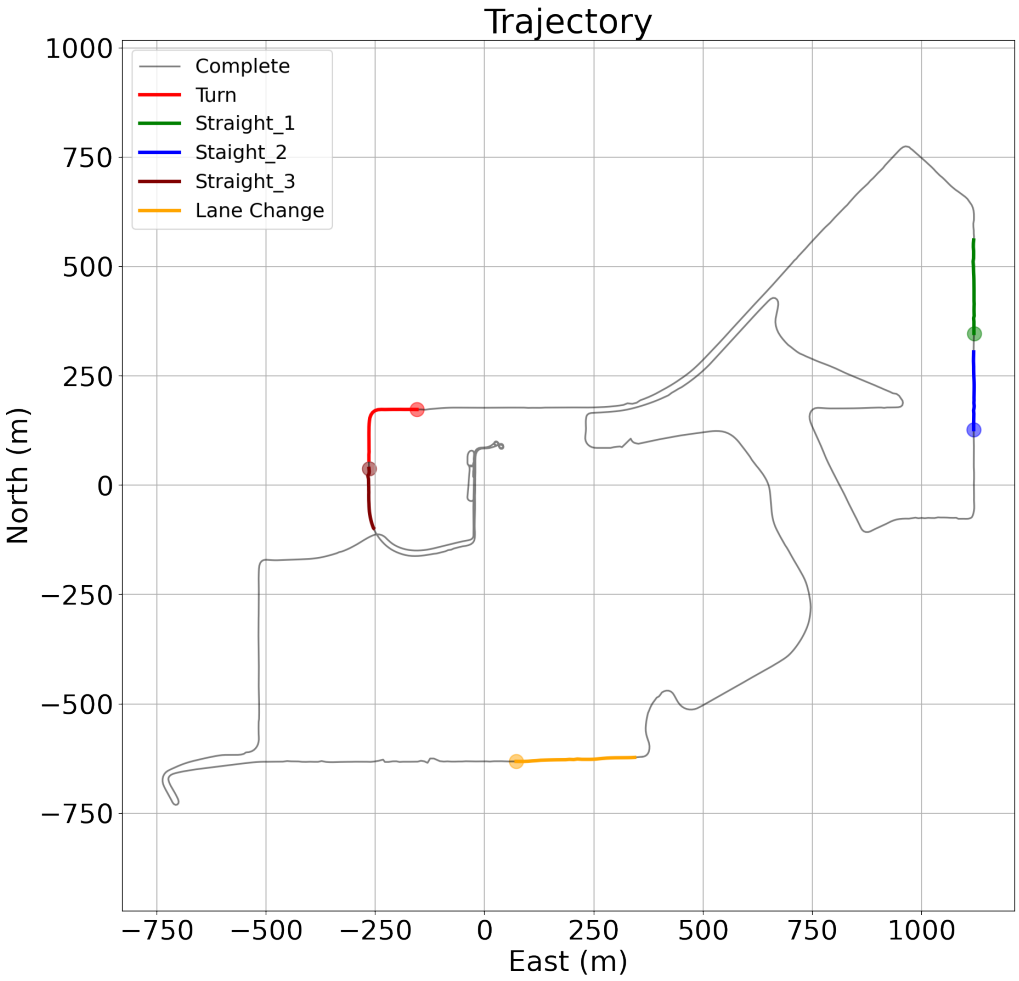


Figure 5.4: Plan view of segments analyzed. Each segment describes a different vehicle maneuver.

In Figure 5.5, the estimated position in an ENU local frame for the "Turn" segment (red line in Figure 5.4) computed with the SIR PF is shown. The EKF solution is plotted in green, showing its 2σ error ellipse derived from the covariance matrix at each epoch. The blue cloud of particles represents the prediction step of the PF, which is computed for each particle using the constant acceleration dynamics model described by Equation (4.5). The red cloud of particles represents the solution after the weighting and resampling steps, based on their agreement with the GPS observations at each epoch. Both filters were initialized at the top-right side of the plot, where the error ellipse is larger for the EKF solution, and the blue cloud of random samples is more spread out for the PF. The reference trajectory is plotted in orange.

In Figure 5.6, the biased solution resulting from using the original map as a constraint is illustrated. It can be observed that the estimated position is biased because the vehicle is not navigating on the segment of road represented by the corresponding edge in the digital map. In such cases, it is more favorable to not use the digital map information, as the cloud of particles is dragged towards the only edge in the map that represents the road.

In Figure 5.7, the solution estimated using the lane-level digital map is shown. During the turn, the map-aiding is not applied, and the state vector is solely estimated using the GPS observations. The particles without a map constraint are plotted in pink. It can be noted that the cross-track variance is significantly reduced in the early epochs of the estimation when using the modified digital map. The particles are distributed closely to their most probable lane based on their direction and distance to each lane. The particles capture a multi-modal distribution in the early stages of the Particle Filter. In the second epoch, some particles are distributed around the lane whose direction is opposite to the correct one; however, the Particle Filter quickly adapts in the following epoch as Equation (5.7) de-weights the particles going in the wrong direction. The map-aiding algorithm ensures that the majority of the particles are positioned on the road, unlike when there is no constraint, where many particles end up in non-navigable areas.

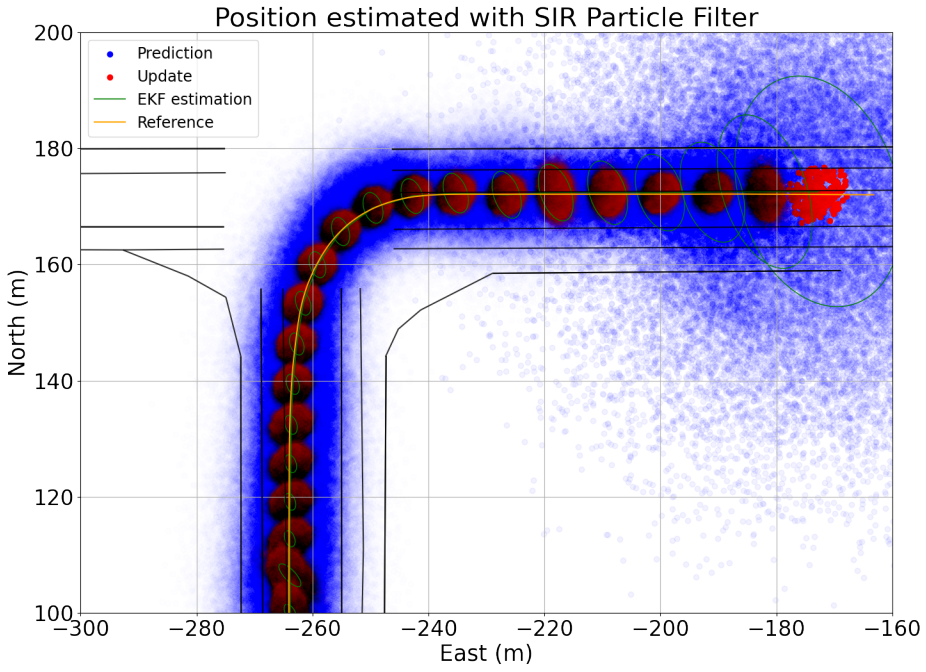


Figure 5.5: Estimated position using the SIR Particle Filter without map-aiding.

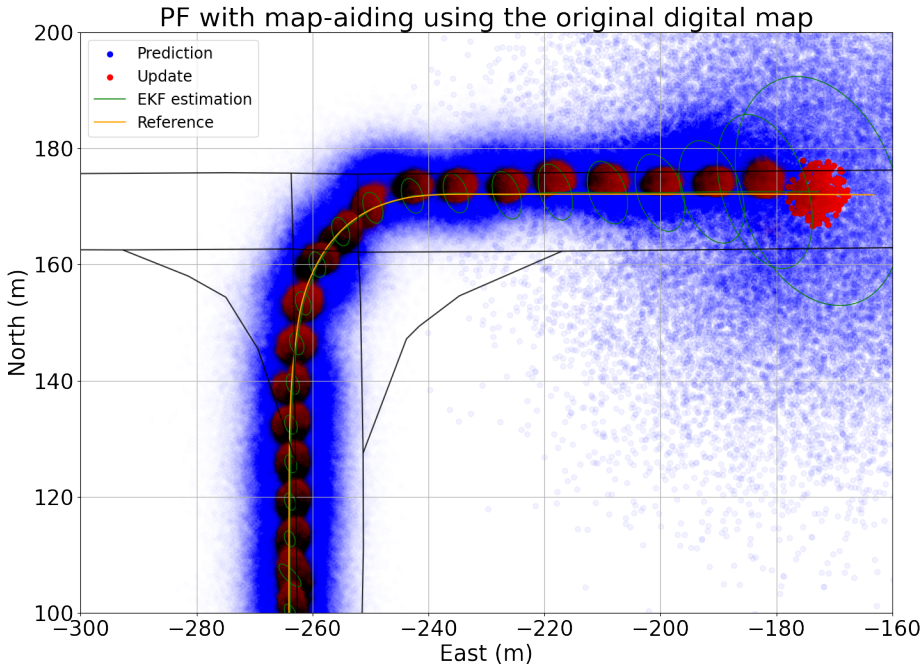


Figure 5.6: Estimated position using the SIR Particle Filter with map-aiding using the original digital map from OSM.

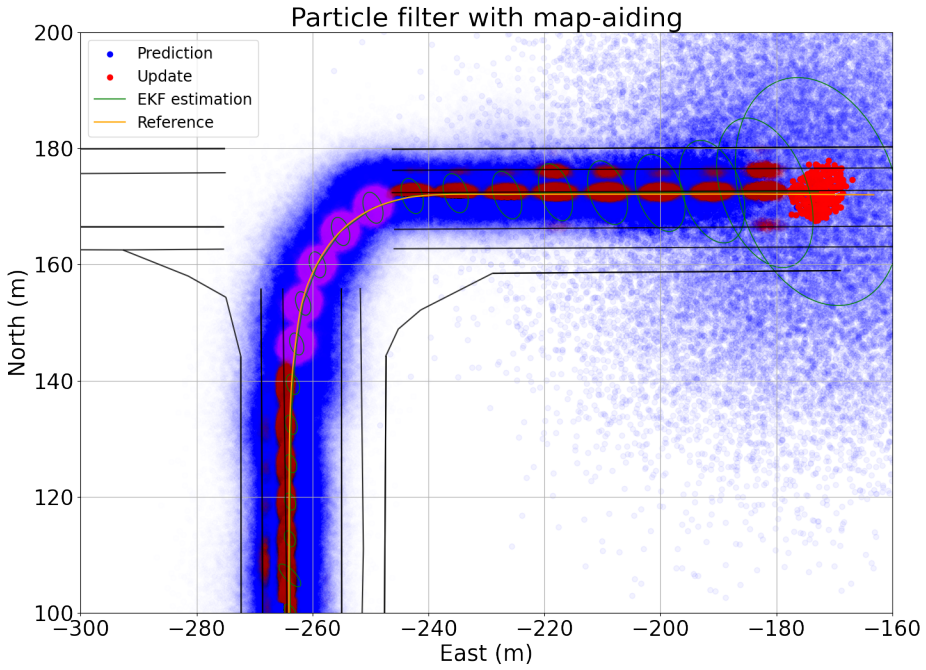


Figure 5.7: Estimated position state with map-aiding using the SIR Particle Filter.

In Figure 5.8, the histogram of the North-South errors of each particle with respect to the reference trajectory is displayed. The center median and curb positions are depicted as solid lines, while the lane boundaries are shown as dashed lines. This histogram corresponds to the updating step of the particle filter without the map-aiding (Figure 5.5) at the second epoch after the filter initialization. The EKF solution is represented by a Gaussian distribution shown in green, centered at the EKF estimate. The frequencies of the particles over the latitudes fit the Gaussian curve from the EKF solution and are normalized by the total number of samples to obtain the probability of the latitude. In this case, the histogram and the Gaussian curve are plotted over the lanes of the road. The estimated latitude at this epoch is in the first lane or turning lane of the Westbound side. Some particles are estimated on the median between both sides of the road, which is a non-navigable area.

In Figure 5.9, the histogram of the estimated latitude of the particles for the same epoch but with the map-aiding algorithm implemented is shown. The distribution now becomes multi-modal since the particles are weighted according to their distance to their closest lane and their heading. Unlike the example in Figure 5.8, all particles end up in navigable areas as the map-aiding assigns a large weight to the particles whose position is close to the lane centers. The number of particles clustered around each lane represents the estimated probability of the vehicle navigating in each lane. Without the map-aiding algorithm implemented

and using the EKF instead, this probability could be estimated by integrating the PDF across each lane. However, using the PF with the map constraint, the particles cluster near the lane centers. By comparing the two plots, the improvement in the position solution at the early stages of the filter convergence is evident, assuming the vehicle is indeed centered in a lane.

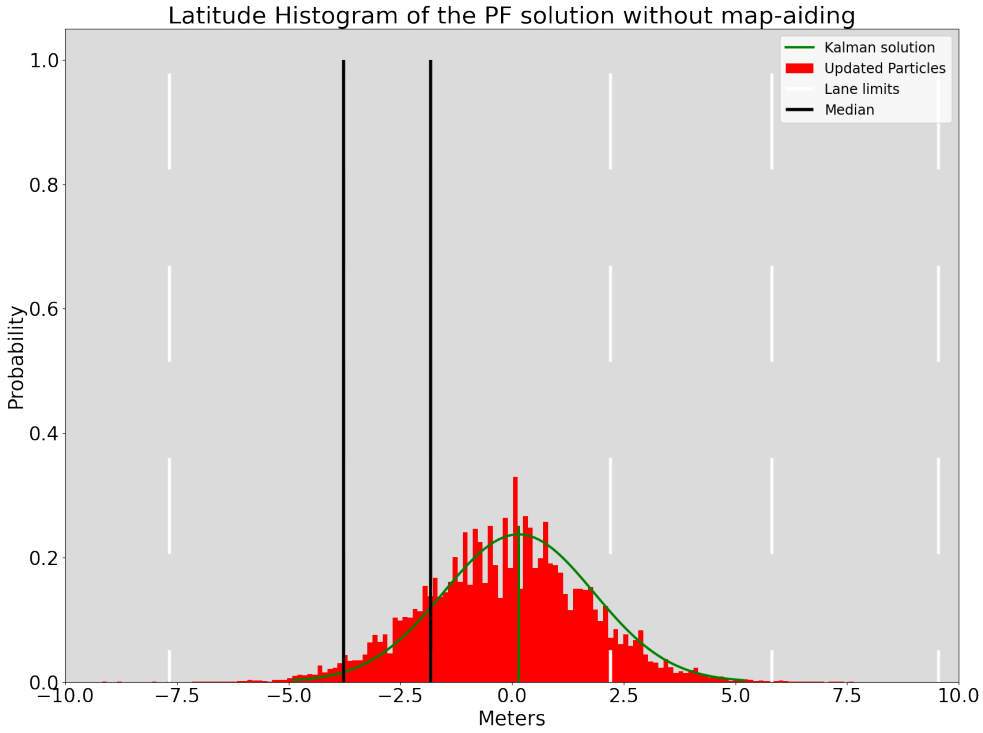


Figure 5.8: Particles distribution of the estimated latitude positions without map-aiding for the second epoch of the *Turn* case shown in Figure 5.5. The EKF estimated position is plotted in green, described by a Gaussian distribution ($\mathcal{N}(\mu, \sigma^2)$) using the estimated position $\hat{x}_k$, and its covariance matrix $\hat{P}_k$

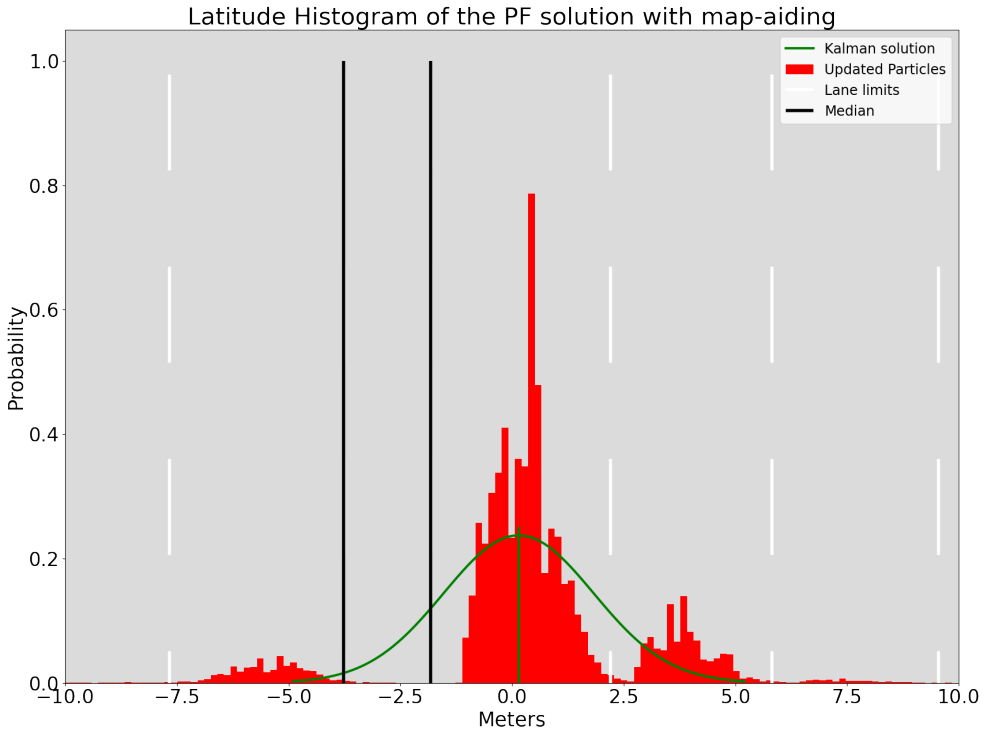


Figure 5.9: Particles distribution of the estimated latitude positions with map-aiding for the second epoch of the *Turn* case.

Figure 5.10 depicts a scenario where the vehicle is not centered within a lane but is navigating between two lanes. In this case, the particles cloud is split into two clusters, each clustered around its closest lane. The green line represents the EKF solution without any map constraint, indicating that the vehicle's position is not centered. It is important to note that the map-aiding algorithm is designed to improve the accuracy of the position estimation when the vehicle is navigating within a lane. It assumes that the vehicle is following a specific lane and weights the particles accordingly. However, in scenarios where the vehicle is changing lanes or not positioned within a lane, the map constraint may not be applicable or may lead to biased estimates. Therefore, the effectiveness of the map-aiding algorithm depends on the specific scenario and whether the vehicle's trajectory aligns with the lanes represented in the digital map.

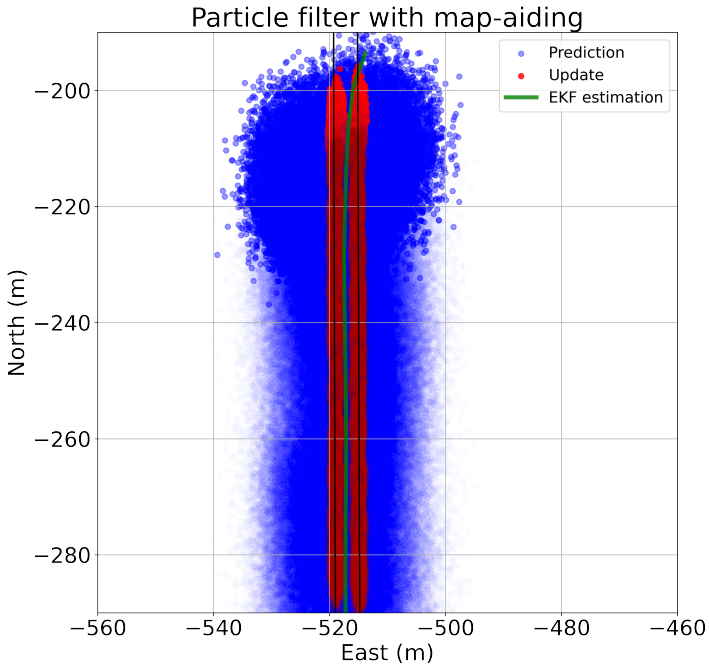


Figure 5.10: Estimated position with the map-aiding implemented when the vehicle is not centred in a lane.

Table 5.1 provides a comparison between the results obtained using the EKF and the Particle Filter (with and without map-aiding) at 5, 15, and 30 seconds after initializing each filter for the five segments shown in Figure 5.4. The estimated variance for the EKF is obtained from the covariance matrix $\hat{P}_k$ after the update step. The variance σ^2 for the PF is estimated empirically using the distribution of the particles to obtain a covariance matrix, as described in Equation (4.21) in Chapter 4. Although these empirical σ^2 values are larger than those of the EKF, the cross-track error is smaller even without map-aiding. The map-aided PF significantly reduces the estimated variance and slightly improves the accuracy. In the last row of the table, the mean of the variances and the cross-track errors for each filter at the compared epochs is shown. It can be observed that the PF with map-aiding estimates the most accurate position. However, the EKF variance is the smallest at the 15 and 30-second epochs, indicating that the effect of the map constraint on the variance is more significant at earlier epochs.

Table 5.1: Estimated cross-track (CT) error (m) and variance σ^2 using the EKF, and the SIR Particle Filter with and without the map-aiding algorithm at 5, 15, and 30 seconds after the filter initialization for the different segments of data shown in Figure 5.4. The background colors of the first column match those shown in Figure 5.4.

Time	5 seconds						15 seconds						30 seconds					
Case/Estimator	EKF		PF		PF/MA		EKF		PF		PF/MA		EKF		PF		PF/MA	
	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT	σ^2	CT
Turn	1.188	0.269	1.402	0.277	0.329	0.201	0.389	0.220	0.844	0.102	0.198	0.092	0.202	0.338	0.812	0.259	0.183	0.257
Change Lane	1.705	0.452	2.104	0.504	1.295	0.389	0.627	4.287	2.310	4.020	0.852	3.865	0.295	2.385	1.521	2.142	0.233	1.532
Straight 1	0.775	0.680	0.975	0.623	0.280	0.609	0.201	1.435	0.854	1.023	0.351	0.993	0.131	0.541	0.548	0.332	0.231	0.328
Straight 2	0.785	0.628	0.923	0.601	0.356	0.532	0.207	1.830	0.713	1.760	0.290	1.661	0.138	0.395	0.492	0.345	0.199	0.625
Straight 3	0.984	0.456	1.123	0.468	0.578	0.438	0.248	0.387	0.378	0.351	0.194	0.277	0.154	0.299	0.241	0.279	0.122	0.234
Mean	1.087	0.497	1.305	0.494	0.567	0.434	0.334	1.6312	1.019	1.451	0.377	1.377	0.184	0.791	0.722	0.671	0.194	0.595

It is important to note that turning lanes are represented by rectangles where the road segments intersect and do not accurately describe the vehicle's true trajectory. Additionally, when the vehicle is changing lanes, there is no edge between the lanes that describes this maneuver. In these cases, the map constraint can lead to a biased position estimation, as it assigns large weights to particles near a road segment. To address this, the map-aiding algorithm is designed to be applied only when the heading angle indicates that the vehicle is driving straight and not turning, or when changing lanes. The applied map-aiding algorithm improves the position estimation accuracy at early epochs by reducing the cross-track error and variance, as the particles are resampled along the most probable lane of the road where the vehicle is navigating.

The map-aided SIR PF improves the accuracy of the cross-track position solution. However, it is important to note that the digital map used for the constraint needs to be more detailed than the original maps provided by the OSM website in order to accurately capture turning lanes and lane changes.

5.3.2 Convergence of the Ambiguities

Some ambiguity resolution algorithms exploit additional constraints or integrate additional sensor measurements to improve float estimation. For example, in [16] the ambiguity resolution is improved by aiding the float solution estimation with LIDAR measurements. [18] proposed an aided ambiguity resolution technique based on a tight-coupling IMU integration using GPS and BeiDou navigation satellite system (BDS) data, showing a faster ambiguity recovery after signal blockage. In [17], an assessment of the performance of both a tightly and loosely coupled IMU/GPS integration for a faster ambiguity recovery is made.

In this section, the positive impact of the map-aiding algorithm on the ambiguity domain is demonstrated. The performance of three integer estimators is compared when using the SIR PF with and without map-aiding for the segments of data displayed in Figure 5.4. The success rate of the estimators is calculated to assess the effect of the map constraint on the convergence of the float solutions.

Figure 5.11 shows the histograms of the SD float ambiguities estimated with the PF without map-aiding during a thirty-second time interval after the filter initialization of the *Straight*3 segment of data (brown). The plotted values represent the differences between the float estimated ambiguities and their respective final integer values, which are computed using the LAMBDA transformation with the ILS on the final epoch of the EKF float solution using the same observations. The histograms are displayed at five-second intervals although the filter updates every second. The first histogram in red at the bottom represents the second epoch after the filter initialization, while the histogram at the top in blue shows the convergence of the float solution after thirty seconds. Since this example does not implement the map constraint, the distribution of the particles is Gaussian at each epoch. In Figure 5.12, the convergence of the float ambiguities is shown when the map-aiding algorithm is implemented. The first histogram in the plot is no longer Gaussian, as the Particle Filter captures a multi-modal distribution resulting from each particle being constrained to its nearest road segment (or lane). However, as the measurement updates are carried out through epochs, the distributions become Gaussian. Comparing these histograms to the solution without map-aiding (Figure 5.11), it can be observed that the variances are reduced, indicating a faster convergence of the filter.

These results demonstrate that implementing the map-aiding algorithm improves the convergence of the float ambiguities in the ambiguity domain. The multi-modal distributions observed when using map-aiding indicate that the particles are constrained to their nearest road segments or lanes, leading to a more accurate estimation of the integer ambiguities.

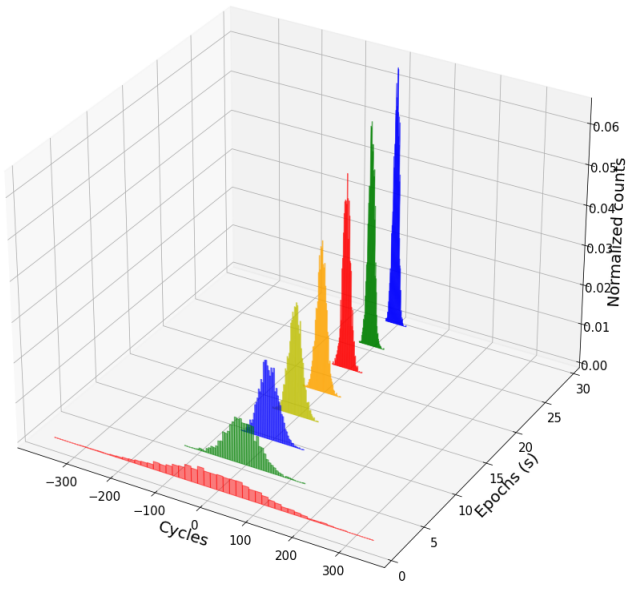


Figure 5.11: Single-Difference ambiguities without map-aiding. Histograms are shown during a 30 second time interval every 5 seconds for better although the filter updates every second second. Each distribution is Gaussian as no constraint is applied.

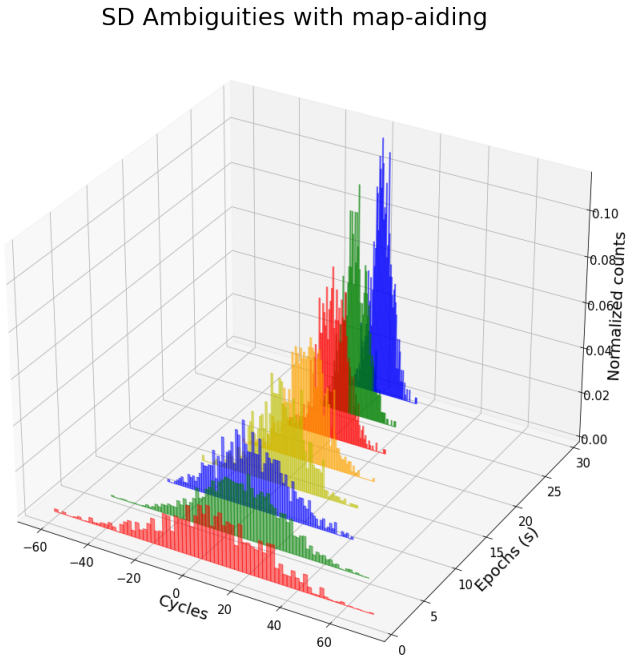


Figure 5.12: Single-Difference ambiguities estimated with the PF with map-aiding. The first histogram is no longer Gaussian, as the map-aiding algorithm results in a multimodal pdf.

The map-aiding algorithm, which weights the particles according to the digital map in addition to the GPS observations, tends to cluster the particles around the most probable lane that the vehicle is navigating on. This clustering effect is primarily observed in the position state, as the map constraint is directly applied to this state. In Figure 5.13, the particles after the resampling step of the second epoch after filter initialization are plotted in a 2-D plot, showing the cross-track position state and a float ambiguity. Without the map-aiding algorithm, both distributions, represented in blue for the position state and red for the float ambiguity, appear Gaussian as no map-matching algorithm is applied. Figure 5.14 shows the same estimation case but with the map-aiding algorithm implemented. In this case, the particles' distributions are no longer Gaussian, especially in the cross-track position state. The particles are clustered around the lanes due to the map constraint. However, in the float ambiguity domain, the distribution still exhibits a Gaussian shape with some irregularities resulting from the resampling and clustering of particles around the lanes. The EKF solution is shown in both cases in green in order to compare the effect of the map-matching technique. The map-matching algorithm applied to the position state improves the position accuracy by reducing the space of the dimension that needs to be described by the particles. This allows the particles to accurately represent

the remaining states. This effect is similar to the reduction of states achieved by marginalizing the state using an EKF, as described in Chapter 3.

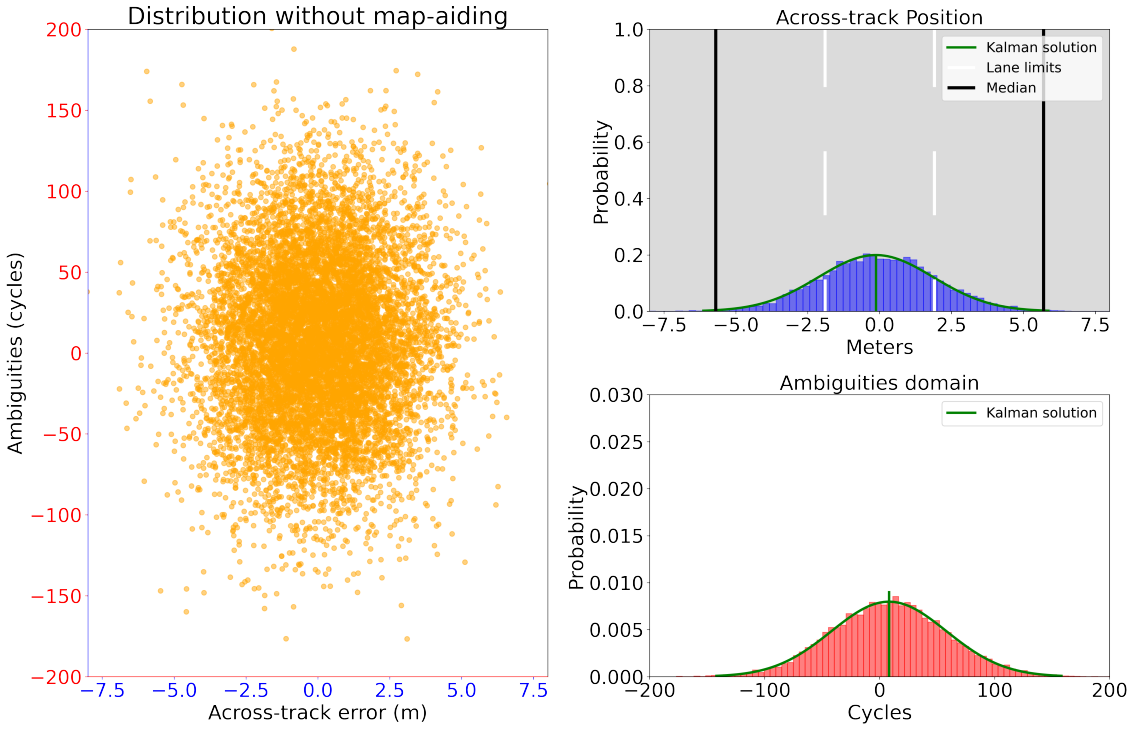


Figure 5.13: Across-track position and float ambiguity distributions and histograms without map-aiding.

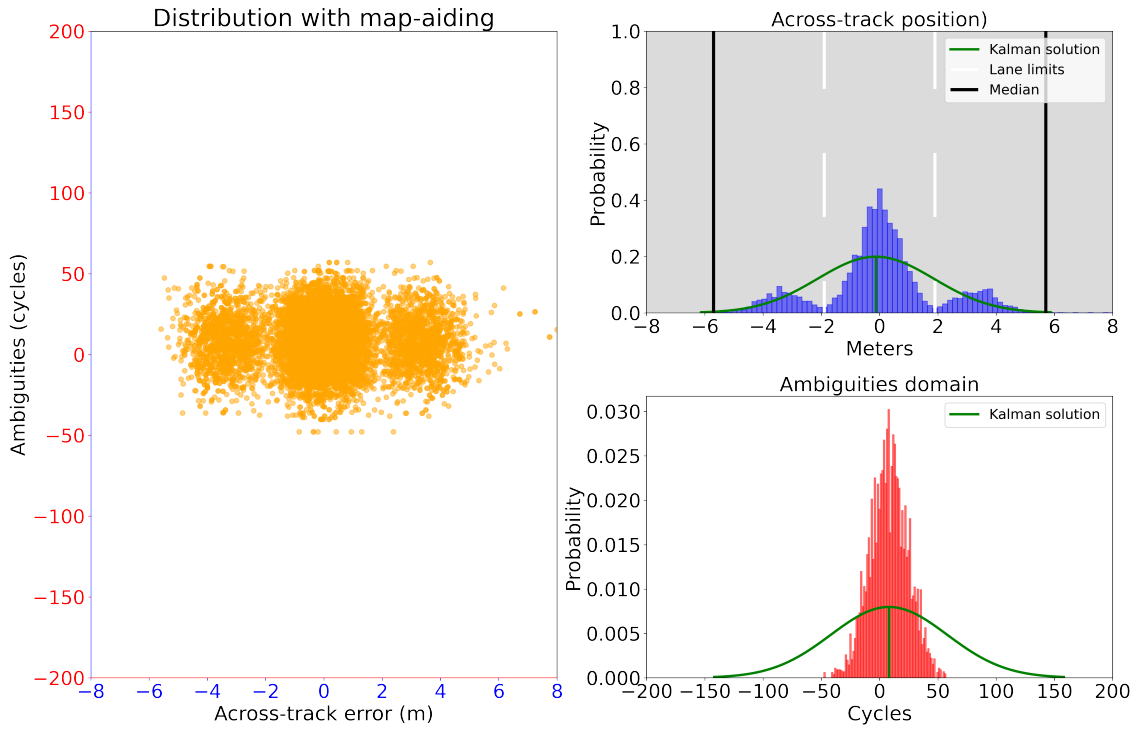


Figure 5.14: Across-track position and float ambiguity distributions and histograms with map-aiding.

In Chapter 4, the probability of correct fix [36] is computed at each epoch using the EKF and PF is computed and shown in figures 4.10 and 4.11. These plots show the comparison between the probabilities of correct fix using the rounding and bootstrapping methods for the correct integer search after the decorrelation transformation. A similar analysis is made in this section to show the effect of the map-aiding algorithm on the float solution when fixing the ambiguities using three integer estimators. However, for this purpose, the comparison is made using only the SIR Particle Filter and the 3 search methods after LAMBDA transformation, with and without the map-aiding algorithm . In Figure 5.15, the probabilities of correct fixing using bootstrapping and rounding as integer search estimators after decorrelation without the map constraint is shown. The empirical covariance matrix (Equation (4.21)) was used to compute the probability of correct fixing, while the success rates are estimated by counting the success or failure of the method applied directly to each of particles. In Figure 5.16, the probabilities of correct fixing and success rates of the search estimators are shown for the same segment of data in Figure 5.15. As the variance of the float ambiguities is reduced due to the map-aiding, the probabilities of correct fixing and success rates improve. Although the map-aiding is directly applied to the position state, it improves the convergence of the float ambiguities resulting in higher probabilities of correct fixing.

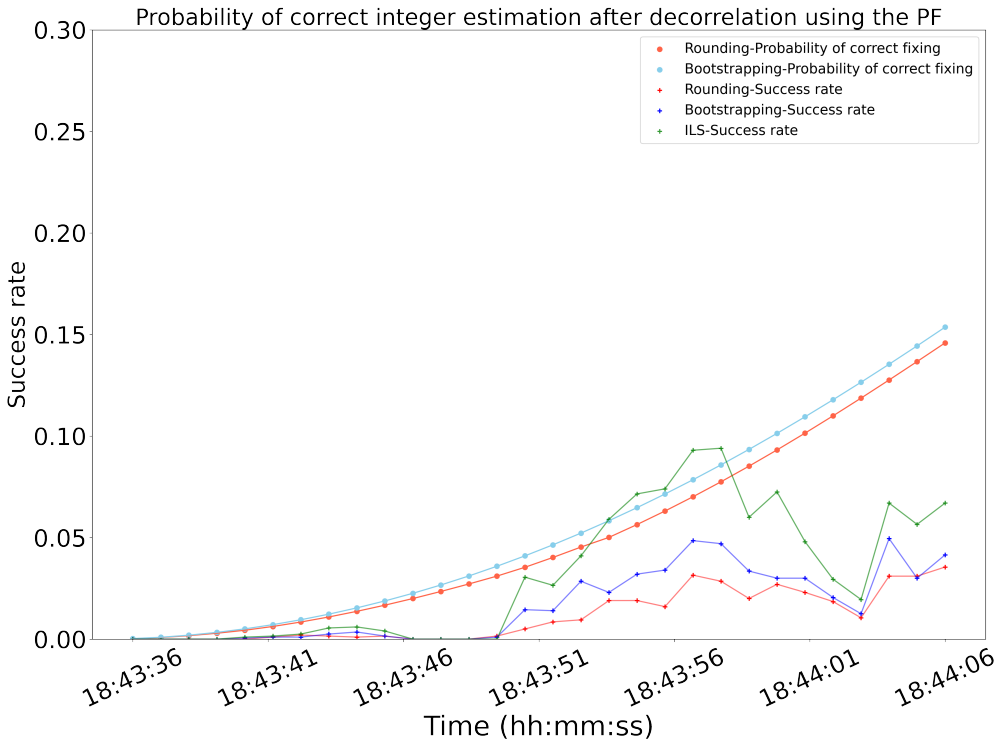


Figure 5.15: Probability of correct rounding (orange) and bootstrapping (light blue) using the PF. Success rates of rounding (red), bootstrapping (blue) and ILS (green).

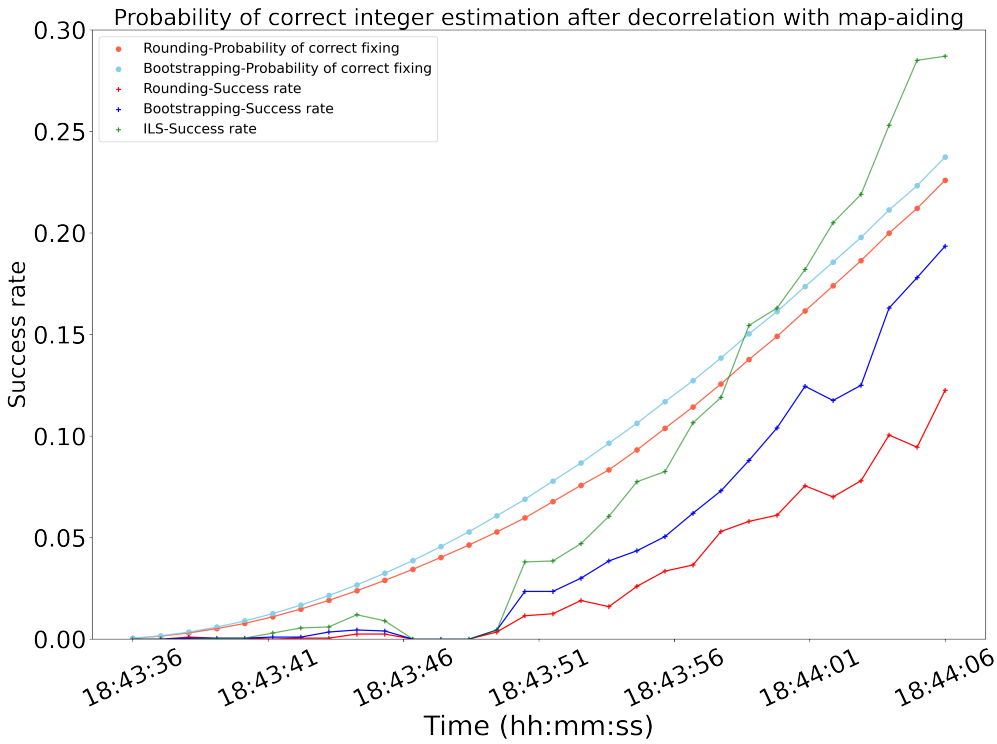


Figure 5.16: Probability of correct rounding (orange) and bootstrapping (light blue) and success rates of rounding (red), bootstrapping (blue) and ILS (green) using the PF with the map-aiding algorithm

Results from all segments

In Table 5.2, the success rate of the ILS and the probability of correct fix of the bootstrapping (PB) method obtained using the SIR PF with and without the map constraint are compared for each of the five segments shown in Figure 5.4 at 5, 15 and 30 seconds after initializing each filter. In the last row of the table, we show the mean of the probabilities of correct fix and success rates of the three epochs. As shown in Chapter 4, the map constraint improves the estimated position at early epochs, reducing the cross-track error and variance as the particles are being resampled along the most probable lane of the road where the car is navigating. Moreover, this improvement in position accuracy is reflected in the ambiguity domain as well. It can be noted that the probability of a correct fix improves as the variance of the float ambiguities is also reduced at each analyzed epoch. In consequence, there is an increase in the count of particles that are being fixed to their respective integer value.

Table 5.2: Estimated probabilities of correct fixing of bootstrapping (PB) and empirical success rates of the ILS using the SIR Particle Filter with and without the map-aiding algorithm at 5, 15, and 30 seconds after the filter initialization for the different segments of data shown in Figure 5.4. The background colors of the first column match those shown in Figure 5.4.

Time	5 seconds				15 seconds				30 seconds			
Case/Estimator	PF		PF/MA		PF		PF/MA		PF		PF/MA	
	PB	ILS	PB	ILS	PB	ILS	PB	ILS	PB	ILS	PB	ILS
Turn	.0038	0.0009	0.0066	0.0017	0.0331	0.0284	0.5156	0.0324	0.1316	0.2091	0.2260	0.2815
Change Lane	0.0041	0.0008	.0074	0.0026	0.0346	0.0142	0.0595	0.0189	0.1428	0.1562	0.2091	0.2001
Straight 1	0.0057	0.0015	0.0103	0.0089	0.0472	0.0303	0.0704	0.0468	0.1731	0.1854	0.2416	0.3251
Straight 2	0.0068	0.0021	0.0094	0.0028	0.0578	0.0462	0.0642	.0438	0.1895	0.0946	0.3461	0.4101
Straight 3	0.0059	0.0012	0.0089	0.0023	0.0413	0.0305	0.0697	.0344	0.1531	0.0678	0.2371	0.2848
Mean	.0052	0.0013	0.0085	0.0037	0.0428	0.0298	0.1556	0.0359	0.1500	0.1426	0.2529	0.3003

5.4 Map-constrained EKF

In order to integrate a digital map as a constraint in the EKF, it is required to represent this information as an additional row in the design matrix H, an additional element for uncertainty in R, and additional row in the innovation or misclosure. The map-constrained EKF follows the next steps:

1. Search the closest lane of a segment of road to the predicted position state $\hat{x}_k^-$
2. Compute the orthogonal vector $r_{\hat{x}}^{lane}$ between the position state and the lane.
3. Compute the magnitude $\left\|r_{\hat{x}}^{lane}\right\|$ and the unit vector $\hat{r}$.

The orthogonal vector is obtained by projecting the position state $\hat{x}_k^-$ into the lane vector $\vec{L}_{AB}$ with nodes A and B. For convenience, $\hat{x}_k^-$ is renamed V_{point} in this description. First, the projection of V_{point} onto the vector $\vec{L}_{AB}$ is as follows:

$$V_{pj} = A + \frac{\vec{AV} \cdot \vec{L}_{AB}}{\left\|\vec{L}_{AB}\right\|^2}(\vec{L}_{AB}) \tag{5.6}$$

where A is the first node (point) of the lane $\vec{L}_{AB}$ and $\vec{AV}$ is the vector from point A to V. Having computed the projection V_{pj}, the orthogonal vector $\vec{V}_{ortho}$ is formed between this point and V_{point}. The measurement model of the map constrain would then be the magnitude of the orthogonal vector $\left\|\vec{V}_{ortho}\right\| = \sqrt{(V_{pj,x} - V_{point,x})^2 - (V_{pj,y} - V_{point,y})^2 - (V_{pj,z} - V_{point,z})^2}$, therefore, getting the differentials for the matrix H is as follows:

$$H = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \frac{\partial Lane}{\partial x} & \frac{\partial Lane}{\partial y} & \frac{\partial Lane}{\partial z} & \cdots \end{bmatrix} \tag{5.7}$$

$$H = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \frac{-(V_{pj,x}-V_{point,x})}{\left\|\vec{V}_{ortho}\right\|} & \frac{-(V_{pj,y}-V_{point,y})}{\left\|\vec{V}_{ortho}\right\|} & \frac{-(V_{pj,z}-V_{point,z})}{\left\|\vec{V}_{ortho}\right\|} & \cdots \end{bmatrix} \tag{5.8}$$

The H matrix would have an extra row for the digital map measurement, where all values are 0 except for the position state. As the assumption of this *pseudo*-measurement is that the distance between the centre of the lane and the vehicle is 0, the innovation is the orthogonal distance $\left\|\vec{V}_{ortho}\right\|$. In Figure 5.17 the solution estimated with the map-aided EKF for the "Turn" segment in Figure 5.4. The solutions obtained with the EKF without and with the map-constraint are plotted in green and blue respectively. Both solutions are shown along with their 2σ error ellipse derived from the covariance matrices at each epoch. As expected, the accuracy is improved and the cross-track variance is reduced. However, an important feature of the map-aided PF is the ability to capture the multi-modal distribution, which is not possible when using the EKF.

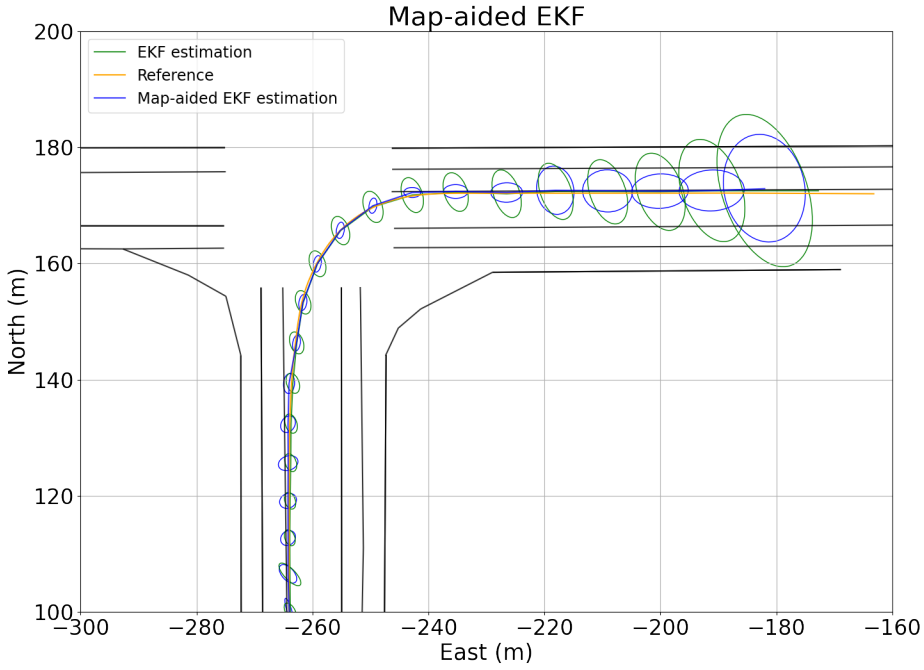


Figure 5.17: Estimated position with the map-aiding EKF.

In Figure 5.18, the probability density functions of the EKF solution are displayed both with and without the application of the map-constraint, specifically for the second epoch of the *Turn* segment (same as shown in Figure 5.8). Notably, the variance in across-track positioning is reduced, leading to improved accuracy. However, unlike the map-aided PF, the Gaussian curve of the EKF solution with the map-constraint still

partially overlaps with the median. This indicates a probability of the vehicle being located in non-navigable areas. This is a consequence from the EKF's consistent assumption of a Gaussian distribution for the posterior pdf. Given the nature of the map-constraint and the particle filter's capability to accommodate various posterior distributions, the particle filter is better suited for handling this type of constraint.

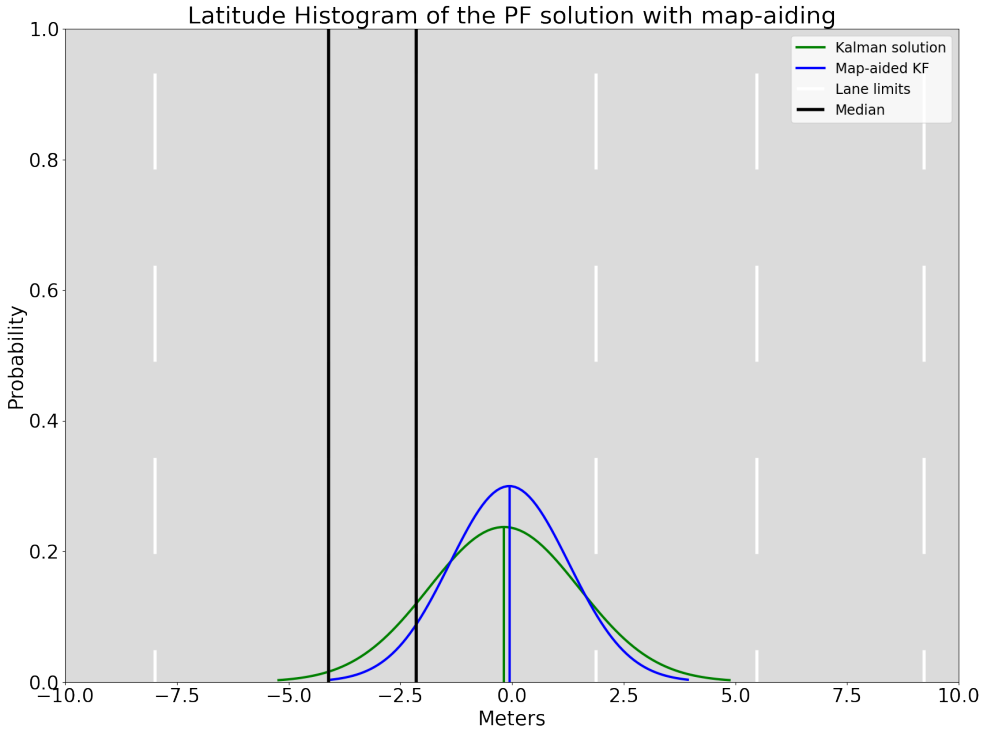


Figure 5.18: EKF distributions with and without the map-constraint for the second epoch of the *Turn* case.

Figure 5.19 illustrates the SD float ambiguities estimated using the EKF, both with and without the map-constraint. This corresponds to the same time interval and data segment as presented in Figure 5.11. The Gaussian distributions are presented at five-second intervals across a thirty-second duration, despite the filter updating every second. As anticipated, the influence of the map-constraint, when applied to the position state, leads to a reduction in the variance of the estimated ambiguities. This outcome arises from the direct correlation between the position state and the ambiguities. Note that the impact of the map-aiding algorithm becomes more pronounced when employed within the PF framework. This can be attributed to the inherent characteristics of sample-based estimators. In addition to the correlation between the position state and the ambiguities, particles more effectively depict the ambiguity posterior distribution when the map-constraint is implemented and the position state space is constrained.

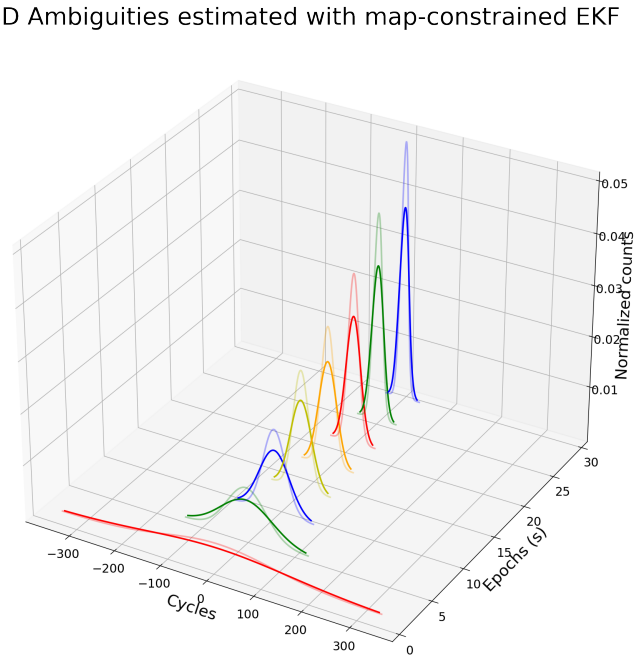


Figure 5.19: Single-Difference ambiguities estimated wit the map-constrained.

The root mean square errors (RMSE) of the across-track position state using the EKF, both with and without the map constraint, for all analyzed data segments, are 0.2561 m and 0.2001 m, respectively, signifying a 27% enhancement. When the PF is applied, the RMSE with and without the map-constraint is 0.2467 m and 0.1957 m, representing a 26% improvement in position accuracy. While the particle filter estimation offers superior accuracy, its key advantage lies in providing more comprehensive information contained within the posterior density when integrating the map-aiding algorithm. This comprehends factors like the probability of the vehicle traversing each lane within a road segment. Regarding the ambiguity resolution, the incorporation of map-matching improves the success rates of the ILS by 90%, demonstrating the algorithm's efficacy and the influence of a position constraint within the ambiguity domain.

In this chapter, the implementation of a map-aiding algorithm in the SIR Particle Filter has been demonstrated, showcasing its positive impact on position accuracy and ambiguity convergence. The constraint provided by the digital map improves the estimation in both the position and ambiguity domains, highlighting the particle filter's ability to handle non-Gaussian posterior distributions. The map-aiding algorithm is particularly useful in the early epochs of the filter, when the state variances are large and the convergence

is still ongoing. By weighting the particles according to the map information, the algorithm helps improve the accuracy and convergence speed of the filter. However, it should be noted that the original maps obtained from the OSM repository require modification. The current representation, where multiple lanes are described by a single vector, leads to biased estimations. Therefore, it is necessary to modify the maps by providing lane-level details to achieve more accurate results.

In the next chapter, an outlier detection algorithm will be implemented to identify observations distorted by multipath effects. Once these observations are detected, the particles will be weighted according to a distribution that accounts for these distortions, deviating from the typical Gaussian distribution.

Chapter 6

Non-Gaussian measurements

In the previous chapters, a detailed explanation about the implementation of the SIR PF for estimating the geometry-based float solution is carried out. A comparison between the EKF and the PF was conducted, revealing their equivalence under optimal conditions for signal reception and low nonlinearity in measurement models. However, the PF offers a more realistic probability of success rate for Integer Rounding and Bootstrapping during the ambiguity resolution process, as it directly counts the samples fixing to the correct integer solution. Chapter 5 introduced the implementation of a map constraint with the goals of improving position accuracy, enhancing convergence of float ambiguities, and showcasing the PF's ability to handle non-Gaussian distributions. Incorporating map-matching within the EKF is impractical due to the challenges of representing highly non-Gaussian posterior densities with a mean and variance parametrization.

In this chapter, we focus on evaluating the performance of the PF in estimating the float solution under urban and semi-urban environments. Specifically, we analyze the PF's behavior when confronted with non-Gaussian measurements, such as those obtained in urban canyon scenarios where GPS measurements are distorted by multipath effects. This analysis demonstrates the PF's capability to weight particles according to any likelihood function, in contrast to the EKF's assumption of a Gaussian distribution.

Section 6.1 provides an overview of Signal Quality Monitoring (SQM) and Measurement Quality Monitoring (MSM) techniques for detecting and mitigating multipath effects. Section 6.2 explains the methodology employed for detecting and mitigating multipath effects on measurements using the state PDF based on point mass representation. Section 6.3 describes the particle filter implementation with outlier detection and the application of Rao-Blackwellization to estimate the full geometry-based solution under urban canyon

environment conditions, along with the corresponding results. Finally, in Section 6.3.2, we present the obtained results with the robust PF, showcasing the impact of deweighting particles based on measurements likely to be distorted by multipath.

6.1 Multipath

The presence of reflective surfaces between GNSS spacecraft and the receiver gives rise to multipath signals, where the signal arrives at the receiver through paths other than the direct line of sight. These reflected signals can cause errors in GNSS measurements [4]. Multipath signals in GNSS are typically classified as Line of Sight (LOS) or Non-Line of Sight (NLOS) signals. LOS multipath occurs when both the direct and reflected signals are received, while NLOS multipath occurs when the received signal comes solely from reflections off surrounding buildings. LOS multipath errors can reach several metres, whereas NLOS errors can be on the order of tens of metres or more in pseudorange measurements [27]. Figure 6.1 provides a 2-dimensional schematic illustrating multipath effects resulting from the reflection of GNSS satellite signals by surrounding buildings.

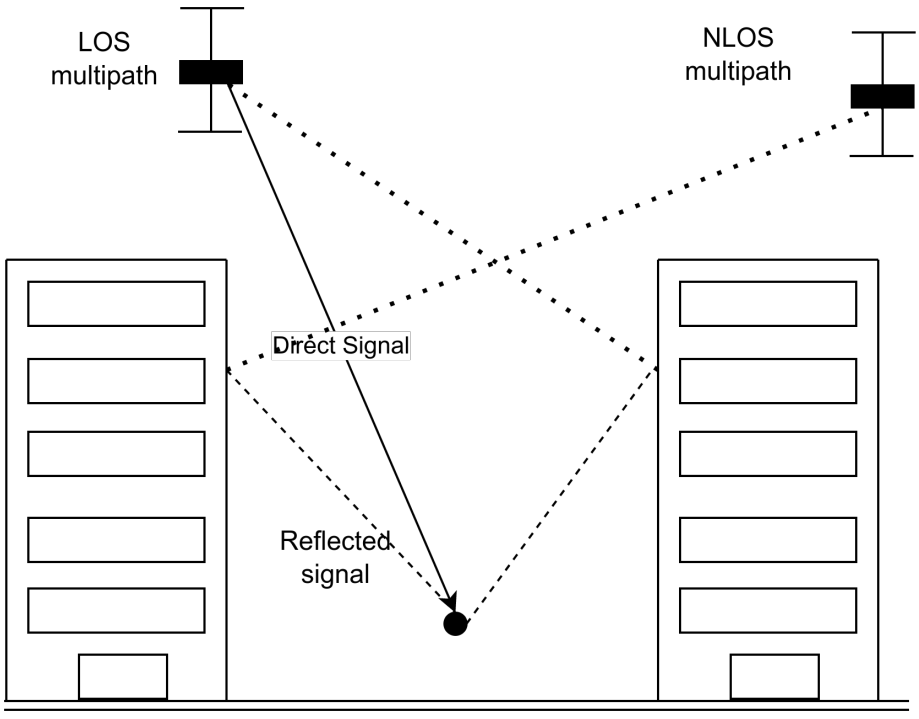


Figure 6.1: 2-dimension schematic example of NLOS and LOS multipath cases.

Multipath signals, being delayed compared to the direct signal, introduce distortions in the phase and pseudorange measurements due to the longer path they travel. Many techniques for the detection and

mitigation of Line of Sight (LOS) multipath have been developed and tested in recent years. These techniques are commonly threefold classified [93]:

- The first group of techniques focuses on isolating the receiver from multipath interference and modifying the antenna design. "Choke ring" antennas or the use of synthetic antenna arrays are within this classification. The "choke ring" suppresses the entry of small elevation signals. A popular technique in this group is polarization diversity which uses two receiver ports with orthogonal polarizations to obtain diversity gain [94].

- The second group attempts to jointly estimate the multipath parameters and remove or mitigate the errors due to multipath. Well-known algorithms in this group are the Narrow [95] and Strobe [96] Correlators. The Narrow Correlator reduces the chip space between the early and late correlator in order to cut down the impact of multipath on the Delay Lock Loop (DLL) [97].

- The third group focuses on detecting and excluding (or de-weight) measurements distorted by multipath. The techniques in this classification include signal and measurement monitoring techniques for detecting and mitigating multipath effects. In [93], it has been demonstrated that these multipath countermeasure techniques satisfy some level of navigation integrity, keeping the satellite geometry (if possible) in most of the cases.

6.1.1 Signal and Measurement Quality Monitoring

Signal Quality Monitoring (SQM) metrics involve the use of different tracking and monitoring correlators to detect any distortions on the overlapped correlation peaks during the receiver tracking stage [98]. Common SQM metrics include Signal-to-Noise Ratio (SNR) and Carrier-to-Noise (C/N_0) ratio, which are used to detect and mitigate multipath effects by employing stochastic weighting models based on metrics like Ratio SQM and Delta SQM [99]. In [100], a review of SQM methods for multipath detection and mitigation is provided. However, one limitation of SQM techniques is their inability to detect short-range multipath. To address this, Measurement Quality Monitoring (MQM) techniques have been developed, focusing on monitoring GNSS measurements (code and carrier phase, Doppler, and C/N_0) to detect and exclude (or deweight) measurements affected by multipath effects. In [93], three traditional MQM approaches are described, namely multipath error correction, stochastic weighting, and detection-exclusion. The Code Minus Carrier (CMC) metric is a popular monitoring metric that combines code phase and carrier phase measurements to provide an estimation of the code range multipath error.

6.1.2 Code Minus Carrier

CMC is a direct measurement of the code range multipath error which is computed by subtracting the carrier phase measurements from the corresponding pseudoranges [101]. Considering the code and carrier phase measurement models (Equation 4.10 and Equation 4.11), CMC is as follows:

$$\begin{aligned} CMC_k &= P_k^i - \Phi_k^i \\ &= \cancel{\rho_k^i - (\delta t_r - \delta t_i)c + \delta_{trop}} + \delta_{ion} + MP_P + \varepsilon_c \\ &\quad - \cancel{\rho_k^i + (\delta t_r - \delta t_i)_k c - \delta_{trop}} - \lambda N_k^i + \delta_{ion} - \cancel{MP_\Phi} - \varepsilon_p \\ CMC_k &= MP_P + 2\delta_{ion} - \lambda N_k^i - \varepsilon_p + \varepsilon_c \end{aligned} \tag{6.1}$$

where MP_P and MP_Φ is the code and carrier multipath error, δ_{ion} is the ionospheric error,λN_k^i is the wavelength times the ambiguity and ε_p, ε_c are the code and carrier noise errors. Given that MP_Φ is considerably smaller than MP_P, the carrier multipath can be neglected. Equation 6.1 describes the CMC metric as an indicator of the code multipath, commonly used to detect measurements distorted by multipath and remove them. If the ambiguity N and the ionospheric effect δ_{ion} are computed, then CMC would be equivalent to the code multipath and can be used to correct the multipath from the measurements and avoid excluding the measurement to keep the satellite geometry configuration [93]. However, considering that the ambiguities and twice the ionospheric effect are constant, these can be removed by computing the average. In Figure 6.2 the CMC metric computed for a segment of the dataset described in Section 4 is shown. In this plot, observations from satellites 18 and 12 present severe multipath effects, while the rest of the observations can be accurately approximates to a Gaussian pdf. This is due to the reason that both satellites have low elevations as shown in Figure 6.3 and the plotted segment of data is of the vehicle driving near buildings.

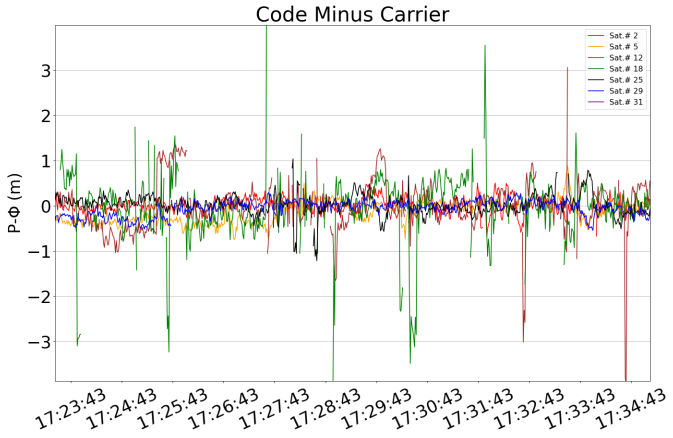


Figure 6.2: CMC metric of segments of data with measurements distorted by multipath.

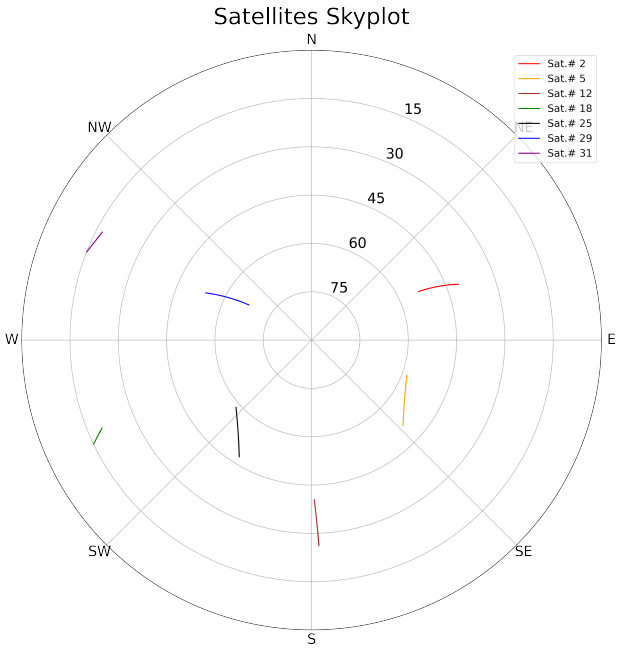


Figure 6.3: Satellites skyplot.

In [93], the CMC metric is utilized for the detection of multipath effects, and a stochastic weighting model is employed to deweight the measurements distorted by multipath. The approach aims to assign lower weights to the detected measurements instead of discarding them, however, the approach assumes Gaussian measurement errors. An important feature of the PF is the ability to weigh the particles according to any likelihood function, and not assuming Gaussian in cases where the measurements are likely to have different distributions. Figure 6.5 illustrates the non-Gaussian distribution of the measurements affected by multipath from the data set used in this thesis, in contrast to measurements without multipath distortion (Figure 6.4). To quantify the level of Gaussianity, a Gaussian curve was modeled using the least-squares method, and the R^2 statistic was computed for both histograms (see Appendix E). The R^2 value for the first histogram is 0.96, indicating that measurement errors without multipath follow a Gaussian distribution. However, in the multipath case analyzed in this thesis, the R^2 value is 0.74, depicting non-Gaussianity.

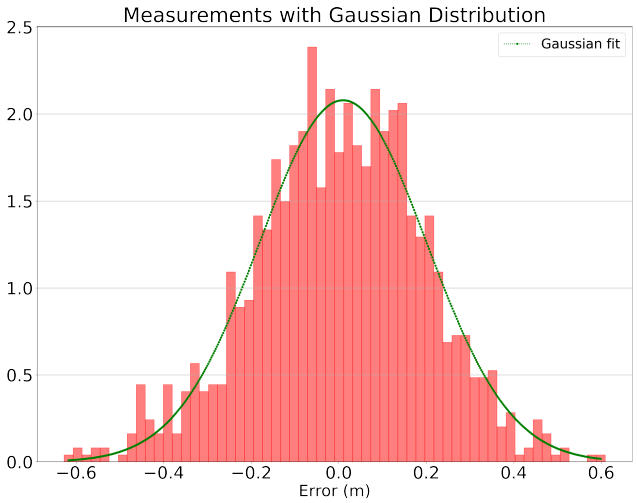


Figure 6.4: Measurements without multipath. The fitted Gaussian (green curve) has a coefficient of determination R^2 of .96

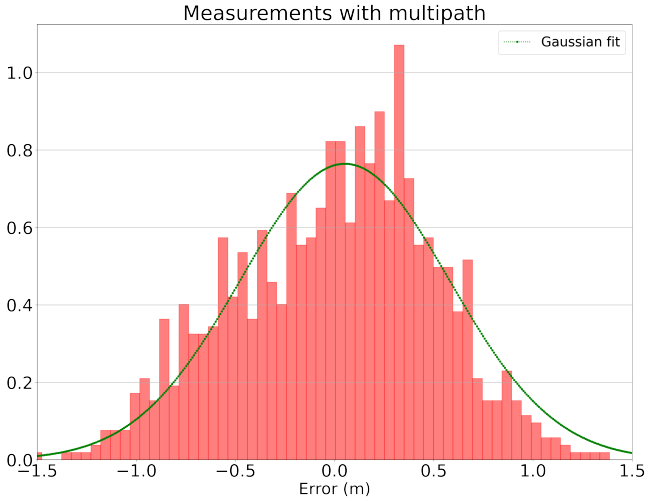


Figure 6.5: Measurements with multipath error. The fitted Gaussian has a coefficient of determination of $R^2 = .78$

A traditional approach for handling measurements distorted by multipath is to utilize the CMC metric to detect the presence of multipath, followed by applying a stochastic model to de-weight the affected measurements by increasing their variance in the filter [93]. However, a limitation of using the CMC metric for multipath detection is its reliance on carrier phase measurements, which requires a restart of the time averaging process in the event of a cycle slip. Figure 6.5 demonstrates that the error distribution of measurements

affected by multipath deviates from the Gaussian assumption, thereby impeding proper weighting of such measurements by the traditional EKF estimator. As discussed in Chapter 3, the Particle Filter (PF) not only addresses nonlinear measurement or system models but also accommodates non-Gaussian measurements.

6.2 Particle Filter in the presence of multipath

Using the double differencing technique reduces the atmospheric errors and the clock error terms are eliminated from the code pseudorange model, leaving only the measurement error:

$$\begin{aligned} \Delta\nabla P &= \Delta\nabla\rho + \varepsilon_{\Delta\nabla P} \\ \varepsilon_{\Delta\nabla P} &= \Delta\nabla P - \Delta\nabla\rho \end{aligned} \tag{6.2}$$

where the $\Delta\nabla$ operator represents the differencing operation between receivers (rover r and base b) and satellites (satellite rs and reference bs). In the presence of multipath error, the pseudorange measurement is significantly affected, leading to distortion in the error term $\varepsilon_{\Delta\nabla P}$. In particular, if the reference satellite bs is selected with the highest altitude and the base receiver b is situated in a location conducive to optimal signal reception, it can be assumed that $\varepsilon_{\Delta\nabla P}$ is predominantly influenced by the multipath error occurring in the pseudorange observation between the rover r and the satellite rs [27]:

$$\begin{aligned} \varepsilon_{\Delta\nabla P} &= \varepsilon_r^{rs} - \cancel{\varepsilon_r^{bs}} - \cancel{\varepsilon_b^{rs}} + \cancel{\varepsilon_s^{bs}} \\ \varepsilon_{\Delta\nabla P} &\approx \varepsilon_r^{rs} \end{aligned} \tag{6.3}$$

where the error $\varepsilon_{\{b\}}^{\{a\}} = P_b^a - \rho_b^a$. In this scenario, the detection of observations with multipath error is achieved through the identification of outliers. This work utilizes an algorithm based on the recursive validation method, also known as innovation testing, for outlier detection [102]. This approach allows for the identification and exclusion of measurements that deviate significantly from the expected values, indicating the presence of multipath error.

Recursive validation on the particle filter

The recursive validation implemented on the EKF consists on computing a predicted residual with its covariance matrix as follows [102]:

$$\begin{aligned} \upsilon_k &= y_k - H_k\hat{x}_{k|k-1} \\ Q_{\upsilon k} &= R_k + H_k P_{k|k-1} H_k^T \end{aligned} \tag{6.4}$$

Considering the local null hypothesis:

$$H_0 : E(\upsilon_k) = 0 \tag{6.5}$$

versus the alternative hypothesis :

$$H_a : E(\upsilon_k) \in \mathbb{R}^{m_k} \tag{6.6}$$

Then, the normally distributed statistic t_k can be used for data snooping as:

$$t_k = \frac{c_k^T Q_{\upsilon k}^{-1} \upsilon_k}{\sqrt{c_k^T Q_{\upsilon k}^{-1} \upsilon_k}} | \mathcal{N}(0,1) \tag{6.7}$$

where c is the canonical unit vector. The main idea of the recursive validation (or innovation testing) for the local identification of outlier measurements is to compare the normalized predicted residual, as defined in Equation (6.7), to a normal distribution at a specified confidence level, denoted as α. Within a Bayesian framework, the predicted distribution of the residuals can be obtained through a two-step procedure:

1. Samples are drawn from the transitional density $p(x_k|x_{k-1})$ by propagating the prior distribution $p(x_{k-1}|y_{k-1})$ through the dynamics model (prediction step).

2. Compute the residuals for each particle using Equation 6.2

From the distribution of predicted residuals $p(\upsilon_k|x_k, x_{k-1})$, the mean value along with its covariance matrix can be computed as (equations (4.20) and (4.21)):

$$\overline{\upsilon_k} = \frac{\sum_{i=1}^{i=N_p} \upsilon_k^i}{N_p} \tag{6.8}$$

and:

$$\begin{aligned} Q'_{\upsilon k} &= \frac{\sum_{i=0}^{i=N_p} (\upsilon_k^i - \overline{\upsilon_k}) \otimes (\upsilon_k^i - \overline{\upsilon_k})}{N_p - 1} \\ Q_{\upsilon k} &= R_k + Q'_{\upsilon k} \end{aligned} \tag{6.9}$$

Having the predicted residual $\overline{\upsilon_k}$ with its covariance matrix $Q_{\upsilon k}$, the residual testing is carried out using Equation 6.7. Detecting outliers using the DD pseudorange residuals overcomes the limitation of relying solely on carrier phase measurements, as in the case of the CMC metric. When an outlier pseudorange observation is detected at a particular epoch k, the particles in the particle filter are weighted using a different likelihood function, deviating from the assumption of a Gaussian distribution. This allows for more flexibility in incorporating non-Gaussian errors and handling outlier measurements effectively.

6.2.1 Mixture of Gaussians

Once the outlier measurement is detected, it can be weighted according to a different statistical distribution model than the Gaussian distribution. The particle filter has no restrictions on using any likelihood function for weighting the measurements. A typical distribution model used when having non-Gaussian measurement errors is the mixture of Gaussians [103]. The sum of two or more Gaussians can represent many error distributions according to different environments or conditions for the observations in one function, improving the accuracy and performance of the filter [80]. The statistical model used for computing the weights of detected outlier measurements at time k and mitigating the multipath effects follows the next mixture of Gaussians [104]:

$$w^i_{k,P,MP,b} = (1-MP)\frac{1}{\sqrt{2\pi\sigma^2_P(k)}}\exp\left(-\frac{\varepsilon^2_{\Delta\nabla P}}{2\sigma^2_P(k)}\right) + (MP)\frac{1}{\sqrt{2\pi b^2\sigma^2_P(k)}}\exp\left(-\frac{\varepsilon^2_{\Delta\nabla P}}{2b^2\sigma^2_P(k)}\right) \tag{6.10}$$

where MP is the probability of the measurement being distorted with multipath effects, b is a factor that accounts for the larger standard deviation associated with outliers [104].

The first term in the expression represents the Gaussian error of the measurement without considering multipath effects (Equation (4.16) from Chapter 4), scaled by $1 - MP$. The second term corresponds to the distribution associated with the measurements, scaled by the probability of being an actual outlier. By weighting both distributions, we ensure that the integral over the entire domain of the mixture of Gaussians remains equal to one. Figure 6.6 illustrates the mixture of Gaussians (Equation (6.10)) used in this analysis. In this distribution, the measurement noise variance is $\sigma^2 = 0.2$, the scaling factor for the variance of the second Gaussian is $b = 5$, the multipath probability MP is 0.8, and both Gaussians have a mean value of 0. The red line represents the measurement error distribution, the blue line represents the multipath effect, and the black line represents the mixture of both Gaussians. Note that the variance σ^2 is the only parameter that varies according to the measurement. This mixture of Gaussians is only used for detected outliers to appropriately de-weight them. Therefore, an MP value sufficiently large was chosen for the multipath distribution (with a larger variance) to have a greater impact on the weight computation and to fit the empirical distribution plotted in Figure 6.5. In [105], the multipath code delay is modeled using statistical models (or distributions) based on a large set of measurements in an urban environment. The author employs a Gamma distribution to model the distribution of multipath delays, considering that multipath delays result in larger pseudoranges. The measurements are grouped based on satellite elevation, resulting in a Gamma distribution model for each group set (divided into intervals of 15 degrees from 0 to 90 degrees). There is a strong correlation between multipath errors and satellite elevation. As satellite elevation decreases, the multipath error distribution tends to widen. However, even for high-elevation satellites, the multipath error

distribution can still exhibit significant width. An important conclusion from this research is that there is no single distribution that universally describes the multipath effects, as it depends on the specific environment, satellite, receiver, and reflecting surface. The motivation behind utilizing a Mixture of Gaussian distributions is to reduce the influence of measurements that are more likely to have multipath errors by combining the original probability density function and the one corresponding to measurements with multipath effects. The result of the mixture of Gaussians is a new distribution that accounts for the uncertainties introduced by multipath, which de-weights the measurements that are likely to have multipath, as determined by the outlier detection method described in the previous subsection. Furthermore, given that the variance of the multipath error is significantly larger than the measurement noise variance, it is crucial to consider a parameter b that adequately broadens the multipath distribution.

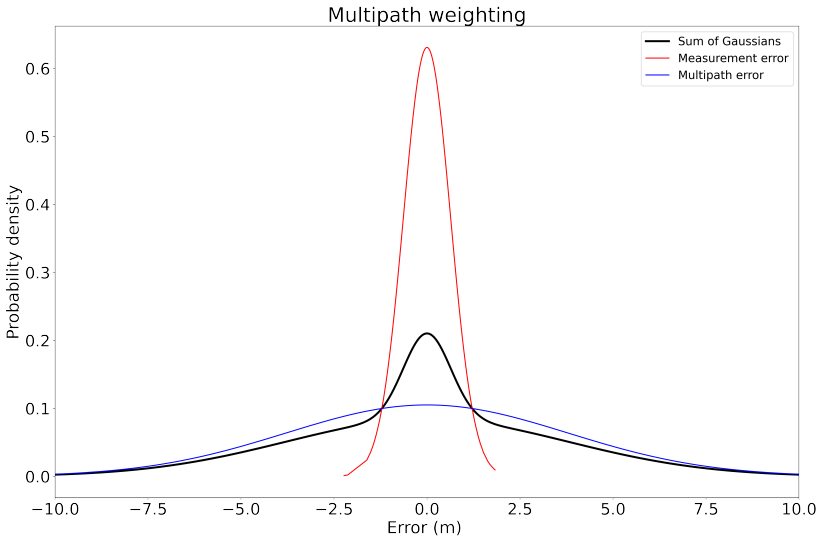


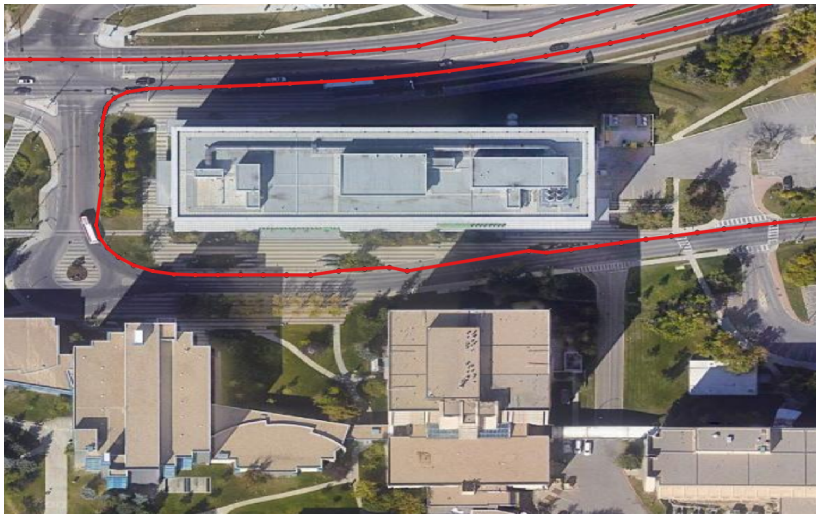
Figure 6.6: Gaussian mixture noise.

6.3 Multipath environment data

The main objective of this section is to evaluate the performance of the particle filter robustified by the outlier detection step and its ability to handle non-Gaussian errors. Chapter 5 demonstrated that incorporating a constraint such as the map-aiding algorithm into the position state improves position accuracy and enhances the convergence of float ambiguities. In this section, we demonstrate how de-weighting measurements affected by multipath enhances position accuracy, thereby also positively impacting the ambiguity domain.

While the majority of the dataset used in this thesis comprises drive-throughs in suburban areas, cer-

tain segments involve urban canyon environments or scenarios where the vehicle navigates near buildings, potentially causing reflections of the GPS signal. Additionally, the CMC metric was employed to identify segments of data that may contain measurements with multipath errors. Figure 6.7 illustrates a representative segment of the dataset corresponding to an urban canyon environment, where the vehicle travels between two buildings for 20 seconds, enough time to evaluate the performance of the outlier detection algorithm during the initial epoch of filter initialization. The objective of this evaluation is to assess the impact of the algorithm on the position and ambiguity convergence and compare the performance of three integer estimators using the float solution estimated with the robust particle filter (with the outlier detection algorithm implemented), the SIR PF, and the PF aided with digital map information.



(a) Multipath environment drive trajectory.

(b) Urban canyon ground-level view.

Figure 6.7: Urban canyon environment example from satellite 6.7a and terrestrial 6.7b view.

6.3.1 Robust Particle Filter

In order to apply the outlier detection step, Rao-Blackwellization (Chapter 3) was used to make the SIR Particle Filter more efficient and reduce the computational demand of the estimation. The Rao-Blackwell process consists on marginalizing the linear state variables and estimating them with a Kalman filter. This means the state vector is split into linear and nonlinear parts, using the SIR-PF for estimating the nonlinear state variables. The partition is as follows:

$$x_k = \begin{bmatrix} x_k^l \\ x_k^{nl} \end{bmatrix} \tag{6.11}$$

where $x_k^l = (v_k, a_k)$, $x_k^{nl} = (r_k, NL1_k)$. v_k, a_k are the rover station velocity and acceleration respectively, which are linear and Gaussian distributed, while the nonlinear part includes the position of the rover r_k and the $NL1_k$ between receiver single difference ambiguities.

Filter initialization

The filter is initialized with N number of particles samples from a Gaussian distribution $\mathcal{N}(x_0, P_0)$. Each particle contains the original complete state vector, however, the linear part is parametrized by a mean and its variance-covariance matrix, effectively reducing the dimensionality of the state space from $9 + NL1$ to $3 + NL1$. By reducing the dimensionality of the state space, fewer particles are needed to accurately represent the distribution of the state, resulting in improved computational efficiency. The initial real-valued ambiguities are computed by subtracting the carrier-phase observations minus pseudoranges $\lambda\hat{N} = P - \Phi$.

Outlier detection

Prior to the weighting step, the statistic t_k from Equation (6.7) is used to detect any outlier measurement. The distribution of the predicted residuals $p(\upsilon_k|x_k, x_{k-1})$ is generated computing the residual for each particle from the transitional density $p(x_k|x_{k-1})$. Estimating the mean value and the empirical covariance matrix from the residuals pdf, the statistic t_k is computed with Equation (6.7) and then compared against the critical value from the normal distribution $\mathcal{N}(0,1)$ at a level of significance $\alpha = .05$. If $t_k > \mathcal{N}(0,1)_{\alpha=.025}$ then the measurement is detected as an outlier.

Weighting

The measurement update follows the same procedure as the SIR PF explained in Chapter 4. However, if the measurement is identified as an outlier, the particles are weighted using the function described by Equation (6.10). Considering the DD measurement models for code and phase observations, the misclosures can be

calculated as follows:

$$\varepsilon_{\Delta\nabla P} = \Delta\nabla\rho - \Delta\nabla P \tag{6.12}$$

and:

$$\varepsilon_{\Delta\nabla\Phi} = \Delta\nabla\rho + \lambda\Delta\nabla N - \Delta\nabla\Phi \tag{6.13}$$

To compute the weight of each particle, the same probability-based approach as the SIR-PF described in Chapter 4 is used, which takes into account the distribution of observation errors. If the measurement is not detected as an outlier, we assume a Gaussian distribution. However, if the measurement is an outlier, we use a mixture of Gaussians (as described in Equation (6.10)). Each observation individual weight is computed as:

$$w^i_{k,P,\Phi} = \frac{1}{\sqrt{2\pi\sigma^2_{P,\Phi}(k+1)}} \exp\left(-\frac{\varepsilon^2_{\Delta\nabla P,\Phi}}{2\sigma^2_{P,\Phi}(k+1)}\right) \tag{6.14}$$

If the measurement is detected as an outlier, then the weight is computed using Equation (6.10) (repeated in this subsection for convenience):

$$w^i_{k,P,MP,b} = (1-MP)\frac{1}{\sqrt{2\pi\sigma^2_P(k)}} \exp\left(-\frac{\varepsilon^2_{\Delta\nabla P,\Phi}}{2\sigma^2_P(k)}\right) + (MP)\frac{1}{\sqrt{2\pi b^2\sigma^2_P(k)}} \exp\left(-\frac{\varepsilon^2_{\Delta\nabla P,\Phi}}{2b^2\sigma^2_P(k)}\right) \tag{6.15}$$

If the map constraint is implemented in the filter, then each particle weight must consider the nearest lane and the correct direction of the edge according to the vehicle's velocity (equation (5.4)):

$$w^i_{k,\Delta x,\Delta\alpha} = \frac{1}{\sqrt{2\pi\sigma^2_{\Delta x,\Delta\alpha}(k+1)}} \exp\left(-\frac{\Delta^2_{x,\alpha}}{2\sigma^2_{\Delta x,\Delta\alpha}(k+1)}\right) \tag{6.16}$$

The product of the weights for each measurement is the particle's total weight. After having computed all the weights, these are normalized such that their sum equals to 1.

After computing the weights for each particle, systematic resampling is carried out in order to discard particles with low weights and replicate those with higher weights. The systematic resample helps to maintain a diverse set of particles that represent the posterior distribution, while focusing on the regions of high probability.

Particle filter prediction

In this step, only the position states and float ambiguities (nonlinear partition x^{nl}) are predicted next epoch. Using the constant acceleration dynamics model augmented by random walk ambiguity states, the nonlinear part is predicted:

$$\Phi_{k+1}^{nl} = \begin{bmatrix} I_{3x3} & I_{3x3}\Delta T & I_{3x3}\frac{\Delta T^2}{2} & 0 \\ 0 & I_{3x3} & I_{3x3} & 0 \\ 0 & 0 & I_{3x3} & 0 \\ 0 & 0 & 0 & I_{mxm} \end{bmatrix} \tag{6.17}$$

where ΔT is the sampling interval in seconds and m is the number of SD observations. The process noise Q_k^{nl} for the nonlinear part is as follows:

$$Q_k^{nl} = \begin{bmatrix} Q_{(p)3x3} & \\ & Q_{(N)mxm} \end{bmatrix} \tag{6.18}$$

where the covariance matrix of the position, velocity and acceleration of the state is $Q_{(p)3x3} = G \cdot G^T \cdot \sigma_a^2$, with $G^{nl} = [\frac{\Delta T^2}{2}{}_{1x3}]^T$, σ_a^2 as the acceleration process noise and $Q_{(N)mxm}$ as a diagonal matrix of process noise σ_N^2. After getting the lower-triangular matrix L_k from the Cholesky decomposition to Q_k, the prediction of the nonlinear partition of the state is as follows:

$$x_{k+1}^{nl} = \Phi_{k+1}^{nl} x_k + L_k \cdot \mathcal{N}(0,\, Q_k) \tag{6.19}$$

It can be noted that the only states being modified by the given Φ_{k+1}^{nl} matrix are the position states. Furthermore, the only states to which process noise is added are the position and ambiguity states. At this point in the filter estimation process, the linear part has not been modified or updated, as the weighting, resampling, and prediction steps are only applied to the nonlinear partition.

Rao-Blackwellization

As the acceleration and the velocity states (linear partition x^l) are not included in the measurement model, the observation information gets to the linear partition through the KF time update step i.e. the prediction step of the linear part. Considering the process noise for the velocity and acceleration states as:

$$Q_k^l = \begin{bmatrix} Q_{(v,a)6x6} \end{bmatrix} \tag{6.20}$$

with $Q_{(p)3x3} = G \cdot G^T \cdot \sigma_a^2$, with $G^{nl} = [\Delta T_{1x3}^2, I_{1x3}]^T$, σ_a^2 as the acceleration process noise, the state noise involving both partitions can be formed as:

$$Q_k = \begin{bmatrix} Q_k^l & Q_k^{lnl} \\ (Q_k^{lnl})^T & Q_k^{nl} \end{bmatrix} \tag{6.21}$$

Using the set of Equations 3.101, the prediction of the states $\hat{x}_{k+1}^l$ is computed for each particle along with the covariance matrix P_{k+1}^l.:

$$\begin{aligned} \hat{x}_{k+1}^l &= \bar{A}_k^l \hat{x}_{k-1}^l + (Q_k^{lnl})^T (Q_k^{nl})^{-1} + L_k(x_{k+1}^{nl} - \hat{x}_k^{nl} - A_k^{nl}\hat{x}_k^l) \\ P_{k+1} &= \bar{A}_k^l P_k (\bar{A}_k^l)^T + \bar{Q}_k^l - L_k N_k L_k^T \\ L_k &= \bar{A}_k^l P_k (A_k^{nl})^T N_k^{-1} \\ N_k &= A_k^{nl} P_k (A_k^{nl})^T + Q_k^{nl} \end{aligned} \tag{6.22}$$

where:

$$\begin{aligned} \bar{A}_k^l &= A_k^l - (Q_k^{lnl})^T (Q_k^{nl})^{-1} A_k^{nl}, \\ \bar{Q}_k^l &= Q_k^l - (Q_k^{lnl})^T (Q_k^{nl})^{-1} Q_k^{lnl} \end{aligned} \tag{6.23}$$

Although the KF time update needs to be run for each particle, neither the update P_{k+1} nor the Kalman gain L_k have dependency with the nonlinear part estimated with the PF, this means that these terms are computed just once for the entire set of particles. Without the additional steps that include outlier detection and the Rao-Blackwellization step, the filter is the same as the SIR PF described in Chapter 4. The SIR PF with Rao-Blackwellization and the outlier detection process is summarized in algorithm 4:

Algorithm 4 Marginalized Particle Filter with outlier detection

Require: For i=1,...,N_p initialize the particles $x_{k0}^{nl(i)} = \mathcal{N}(x_0, P_0)$ and set $\{x_0^{l(i)}, P_0^{(i)}\} = \{\bar{x}_0^l, \bar{P}_0\}$, dynamic model f

for $k = 1, 2, ...$ **do**
  for $i = 1, 2, ..., N$ **do**
    Get transitional density $p(x_k|x_{k-1})$ from sampling $x_k^{(i)} \sim f(x_{k-1}^{(i)})$
    for $j = 1, 2, ..., m$ **do**
      Estimate the distribution of the predicted residuals $p(\upsilon_k|x_k, x_{k-1})$ and perform outlier detection test (Equation (6.7)) at each measurement y_k^j.
      if y_k^j is an outlier **then**
        Calculate the ith importance weight using the mixture of Gaussians for likelihood function (Equation (6.15))
      else
        Calculate the ith importance weight using the Gaussian distribution (Equation (6.14))
      end if
    end for
  end for
  Normalize weights $\hat{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
  Resample particles $x_k^{nl(i)} = \{x_k^{nl(i)}, \hat{w}_k^{(i)}\}_{i=1}^{N}$ using systematic resampling method
  for $i = 1, 2, ..., N$ **do**
    Perform Kalman Filter update for the linear partition of the state $x_k^{l(i)}$ using equations (6.22)
  end for
  Calculate the complete state $\hat{x}_k^{nl,l} = \sum_{i=1}^{N} \hat{w}_k^{(i)} x_k^{(i)}$
  Calculate estimated state covariance $P_k = \sum_{i=1}^{N} \hat{w}_k^{(i)} (x_k^{nl,l(i)} - \hat{x}_k^{nl,l})(x_k^{nl,l(i)} - \hat{x}_k^{nl,l})^T$
end for
return estimated state $\hat{x}_k$ and estimated state covariance P_k

6.3.2 Results

The particle filter with the outlier detection step discussed in the preceding section was implemented to estimate the complete geometry-based float solution for a segment of the dataset that presented pseudorange measurements affected by multipath errors, thereby facilitating an evaluation of the proposed algorithm's efficacy in improving position accuracy and ambiguity convergence. Figure 6.8 visually depicts the analyzed

segment, representing a trajectories within an urban environment. This scenario involve the vehicle traversing areas surrounded by buildings, which serve as potential reflecting surfaces for the GPS signals. Such settings increase the likelihood of encountering multipath-induced errors in the measurements, making it ideal for assessing the impact of the proposed algorithm.

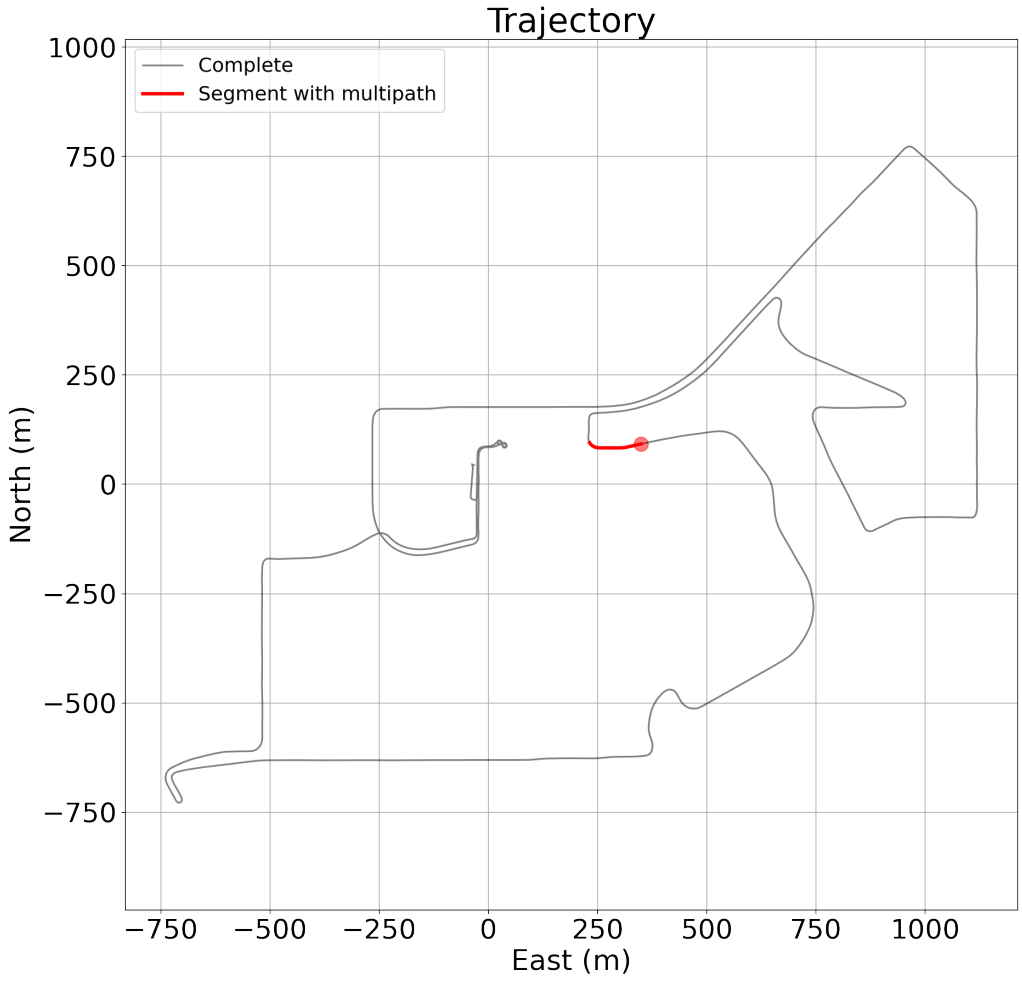


Figure 6.8: Plan view of the segment with multipath analyzed in this chapter.

The analyzed segment represents a 20-second narrow travel route characterized by the vehicle's proximity to three surrounding buildings. This particular environment is highly susceptible to multipath errors in the GPS measurements. Figure 6.7 illustrates the trajectory of this segment, showcasing a ground-level view of the path. Additionally, Figure 6.9 displays a fish-eye image captured at the location where the drive-through took place.

Figure 6.9: Fish-eye image of the analyzed segment.

In Figure 6.10, the discrepancy between the reference trajectory and the float solution estimated with the PF is depicted in the east, north, and up directions. It can be observed that the estimated positions exhibit a bias, which can be attributed to the presence of multipath errors. On the other hand, Figure 6.11 illustrates the ENU errors of the same subset of data estimated with the PF that incorporates the outlier detection algorithm. The detected outlier measurements were assigned the mixture of Gaussians likelihood function, as in the previous section (Equation 6.15), for weighting the particles. This modification leads to a reduction in the position bias.

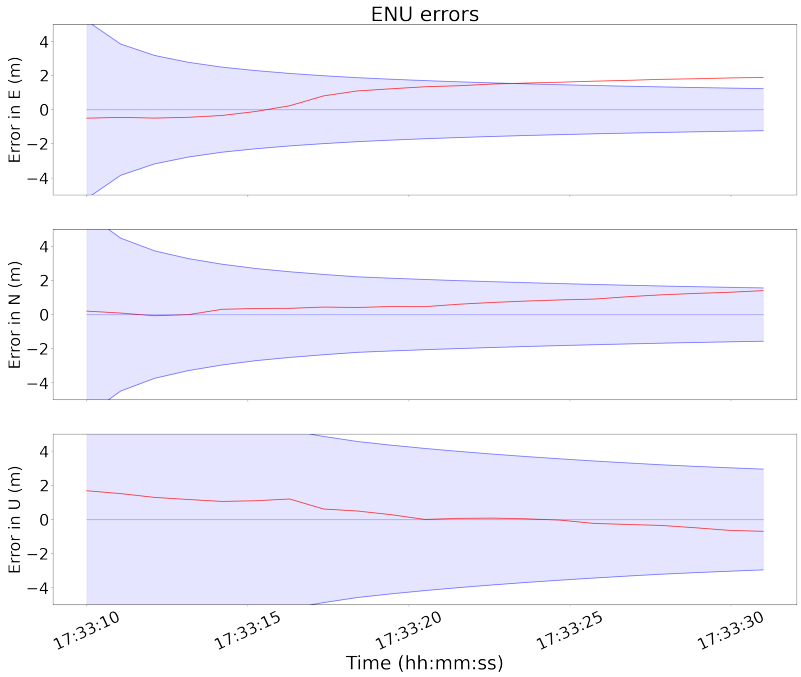


Figure 6.10: ENU errors of the float solution obtained with the PF .

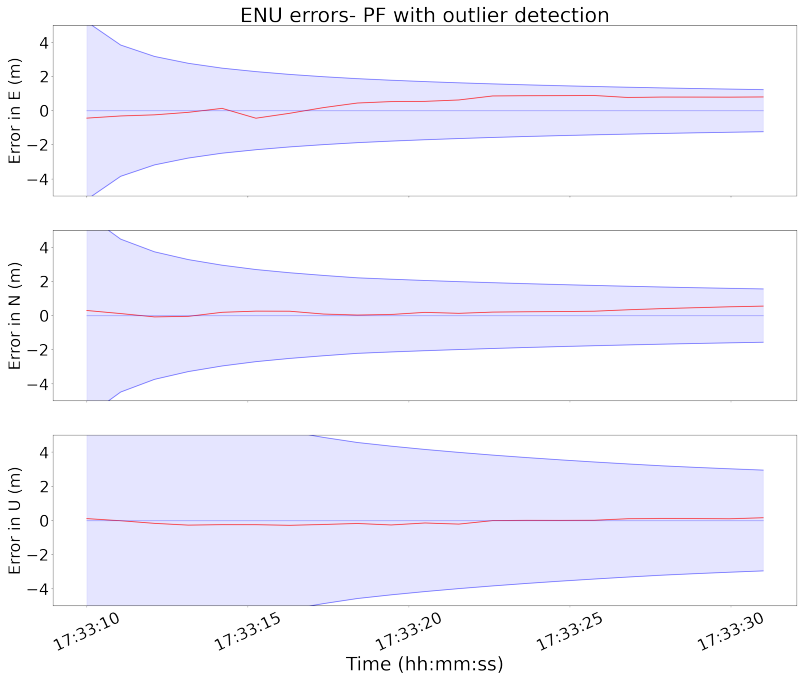
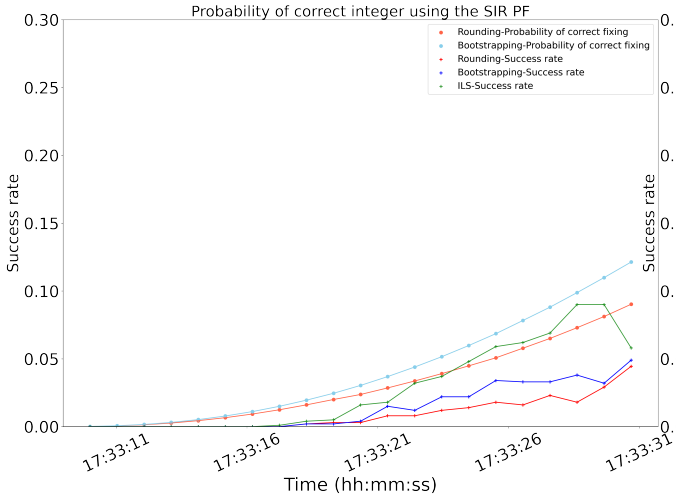
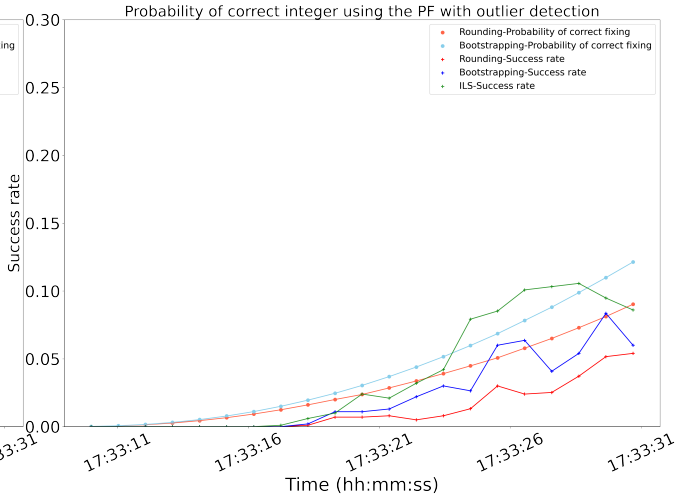


Figure 6.11: ENU errors with the outlier detection and de-weighting process implemented on the PF.

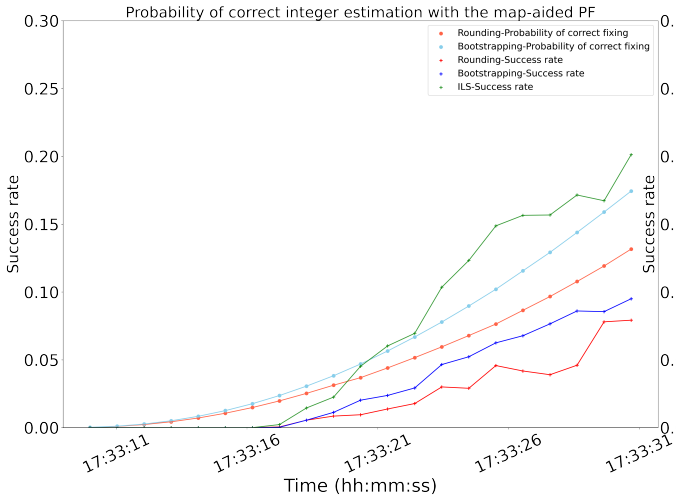
In Figure 6.12 the probabilities of correct fix and estimated success rates using the PF are shown. Figures 6.12a and 6.12b illustrate the impact of the outlier detection without map-aiding while figures 6.12c and 6.12d show the results with the map constraint. The outlier detection has a more noticeable effect when there is no map constraint, resulting in faster ambiguity convergence. However, as the map constraint impact is harder, the deweighting of the multipath measurements effect on the ambiguity domain is slightly appreciated. The map constraint has a stronger effect on the convergence of the float ambiguities enhancing the convergence of the float ambiguities by reducing the variance in the position state, which enables the particle to better describe the remaining states.



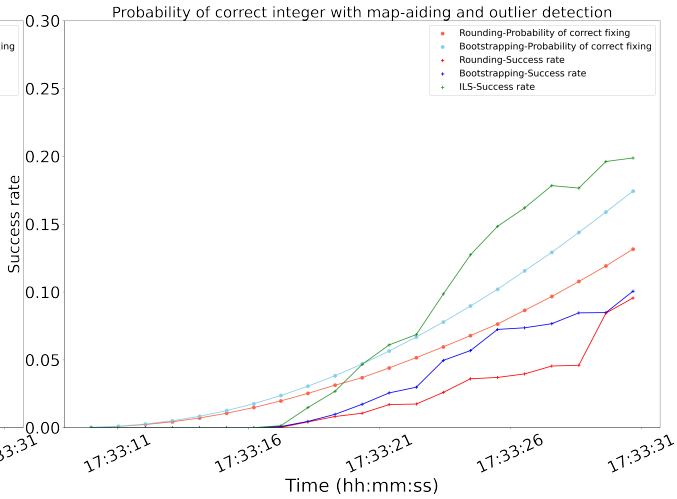
(a) Probabilities of correct fixing and success rates using the SIR PF



(b) Probabilities of correct fixing and success rates using the PF with outlier detection



(c) Probabilities of correct fixing and success rates using the map-aided PF



(d) Probabilities of correct fixing and success rates using the map-aided PF with outlier detection

Figure 6.12: Probability of correct rounding (orange) and bootstrapping (light blue) and success rates of rounding (red), bootstrapping (blue) and ILS using the SIR PF

The implementation of the outlier deweighting algorithm positively impacted the estimated probabilities of correct fixing, leading to a reduction in the variance of float ambiguities and higher success rates as more particles accurately converged to their correct integer values. On average, the outlier deweighting algorithm improved the success rate of the Integer Least Squares (ILS) by 30% and the probabilities of correct fixing of the bootstrapping method by 18% after 30 seconds of the filter initialization. This positive impact is due to reduction of bias in the position state. The RMSE's for the ENU positions, are 1.415m, .878 m. and 1.231m

respectively without deweighting the measurements distorted by multipath. When the outlier detection algorithm is implemented, the position accuracy improves and the RMSE values for the ENU positions are .856 m, .363 m. and .341 m. This results show the importance of detecting and deweighting (or rejecting) outlier measurements.

This chapter highlights the particle filter's capability to handle non-Gaussian measurements by utilizing a likelihood function other than the Gaussian distribution. This is particularly relevant when dealing with distorted pseudorange observations caused by multipath effects. An outlier detection methodology based on the innovation testing technique was introduced, leveraging the transitional prior to obtain the distribution of predicted residuals and calculating the normally distributed t_k statistic. The identified outlier measurements were subsequently deweighted using a mixture of Gaussians, effectively reducing bias in the position estimate and yielding a beneficial impact on the ambiguity domain. By utilizing the map constraint and incorporating the outlier detection algorithm, the position estimate is enhanced, leading to a reduction in bias and improved accuracy. This, in turn, has a beneficial effect on the convergence of ambiguities. These results reinforce the importance of considering and refining the position estimate in order to achieve more reliable and accurate ambiguity resolution.

Chapter 7

Conclusions and recommendations for future work

This work focuses on conducting a comprehensive assessment of the particle filter performance in estimating the geometry-based float solution. The research highlights the main advantages of the PF over the commonly used Kalman filter. The PF outperforms the KF in two cases: when the dynamic system has a highly nonlinear measurement model and when non-Gaussian posterior densities are present. Additionally, the PF has the flexibility to use any likelihood function to weigh the particles, unlike the KF which assumes Gaussian measurements. In this chapter, the major conclusions of this research are discussed, and recommendations for future work are suggested.

7.1 Conclusions

The conclusions provided in this section are sorted by chapter for the reader convenience.

7.1.1 Nonlinear Filters

The comparison between the most common nonlinear estimators, namely the extended Kalman filter, unscented Kalman filter, and particle filter, was conducted in this research. The EKF is a first-order Taylor series approximation to the nonlinear case, while the PF does not require any approximation and can directly use the measurement model. The UKF approximates the posterior probability density function (pdf) to a Gaussian using a set of defined samples, without approximating the nonlinear model itself. The UKF shares common features with both the EKF and PF.

In linear cases with Gaussian posterior densities (which are ideal scenarios for the Kalman Filter), the EKF, UKF, and PF estimate equivalent results, and there is no particular reason to choose one over the others. However, describing the full posterior pdf with particles in the PF requires a large set of samples, which leads to a heavy computational cost. This is a major limitation of the PF. To address this, two options were discussed to reduce computational costs without compromising estimation accuracy. The first option is the local-linearization step, which improveThe shaded areas are the 2σ standard deviation of the float ambiguities.s the importance density by applying the EKF to each particle before the weighting and resampling step. This step moves the particles towards the likelihood area, reducing the number of particles required. However, a downside is that the distribution becomes more Gaussian, and the original distribution of particles is lost. The second option is the Rao-Blackwellized or marginalized particle filter, which splits the state into nonlinear and linear parts. The nonlinear part is estimated using the PF, while the linear part is estimated using the Kalman Filter. This reduces the number of particles needed, as fewer state pdfs are described by the particles.

The implementation of the Sequential Importance Resampling (SIR) PF is described in detail, highlighting the equivalence between the measurement and prediction steps of both filters. The PF is presented as a Minimum Mean Square Error (MMSE) estimator, which provides a comprehensive understanding of its operation and capabilities.

7.1.2 SIR Particle Filter in Float Solution for Ambiguity Resolution

The SIR PF was used to estimate the full geometry-based float solution for ambiguity resolution using L1 GPS carrier phase and pseudorange observations, and the results were compared to those obtained with an EKF. The implementation details of the SIR PF were described in detail. The chosen data segment for analysis was selected to ensure optimal signal reception conditions, resulting in similar estimations from both filters. This aligns with the findings from Chapter 3, which demonstrated that the estimations from both filters are comparable when dealing with Gaussian posterior pdfs and low nonlinear measurement models.

The float solution obtained with the PF provides the necessary information to estimate the empirical success rate of three admissible integer estimators (ILS, bootstrapping, and rounding) based on the distribution of particles. This estimation involved counting the number of particles that were fixed to their correct integer values using each integer estimator. The estimated success rates were then compared to the probabilities of correct fix calculated using Equations 4.22 and 4.23 from a previous study [36] for rounding and bootstrapping. However, since these expressions do not account for biases in the float solution, the estimated success rates were lower compared to the probabilities of correct fix assuming unbiased float estimates.

The application of the LAMBDA transformation to the float ambiguities significantly reduced the computational complexity when performing the integer search.

In the optimal signal reception conditions considered, there was no significant difference between the solutions obtained with both estimators. However, the PF provided additional information for fixing the ambiguities, such as more realistic success rates for the integer fixing using any of the three integer estimators..

7.1.3 Map-aided Particle Filter for Ambiguity Resolution

In Chapter 5, a map-aiding algorithm was implemented in the SIR PF to estimate the float solution for ambiguity resolution. The algorithm uses digital maps available in the OSM repository and weights the particles based on the most probable lane of a road segment, considering the correct direction and the distance between the vehicle and the multiple lanes. It is important to note that using the original maps in the map-aiding algorithm would introduce bias since the multiple lanes of the road are typically described by a single vector. Therefore, a lane-level map was required for accurate results.

By applying a constraint that leads to highly non-Gaussian posterior distributions, the SIR PF demonstrated its capability to capture multi-modal distributions. In contrast, using the EKF would only provide an approximate estimation of the probabilities by integrating the Gaussian over each lane.

The map-aiding algorithm notably improved the accuracy (by %26)of the vehicle's across-track position, particularly in the early epochs where the solution had a larger variance. As the solution converged, the impact of the map-aiding algorithm diminished. Five segments of the dataset representing different driving maneuvers were analyzed. For each scenario, the results obtained using the PF with and without the map-aiding algorithm were compared to the results obtained with the EKF at different time intervals after the filter initialization. In all cases, the PF with the map-aiding algorithm demonstrated better position accuracy compared to the other methods. Furthermore, the convergence of the float ambiguities was improved when the PF was aided with the digital map. The success rates of the ILS had an improvement of %95. This highlights the positive impact of improving the position estimation on the ambiguity resolution. The particles being less dispersed in the position state allowed the filter to better describe the posterior distribution of the other states, similar to the effect of marginalizing states with the EKF.

The incorporation of a map constraint into the EKF becomes feasible when digital map information is incorporated as a measurement using the H design matrix. This process involves differentiating the measurement model, which corresponds to the magnitude of the orthogonal vector connecting the lane and the predicted position. Although a comparable enhancement is observed as with the map-aided PF (%26), the EKF is limited in capturing the multimodal posterior distribution. Additionally, one of the distinct advantages of

the map-aided PF is that particles are not resampled within non-navigable regions. In contrast, the Gaussian distribution derived from the EKF solution intersects with the median between roadways.

Chapter 5 showcased two significant advantages of the PF over the EKF: the ability to integrate highly nonlinear and non-Gaussian constraints into the estimation process and the ability to capture multi-modal distributions or other types of distributions beyond the Gaussian assumption.

7.1.4 Non-Gaussian Measurements

The assessment of the PF for estimating the float solution for ambiguity resolution in the presence of non-Gaussian measurements was conducted. Specifically, the analyzed data segment was selected to have measurements distorted by multipath, which do not follow a Gaussian distribution. To address this, two additional steps were implemented in the SIR PF prior to the measurement update step. The first step was the implementation of an outlier detection algorithm. The statistic t_k, which is a normalized predicted residual estimated from the transitional density, is firstly computed. This metric was then compared to the normal distribution at the desired level of significance α. Unlike the recursive validation (or innovation testing) usually implemented on the EKF, the predicted residual and its covariance for the PF are estimated using the particles themselves. The second step was the Rao-Blackwellization (or marginalization) of the velocity and acceleration states. This step aimed to reduce the dimensionality of the state being estimated by the PF, thereby reducing the computational load. By separating the nonlinear and linear states, the estimation of the nonlinear part was performed using the PF, while the linear part was estimated using the Kalman Filter. Once an outlier measurement was detected, a mixture of Gaussians was adopted to weight the particles in the PF. This distribution consisted of a sum of two Gaussians weighted according to the probability of the measurement being distorted by multipath. As a result, the weighted distribution became a skewed Gaussian that accounted for distortions in both directions. For the analyzed data where the outlier detection and the mixture of Gaussians were employed, improvements in position accuracy were observed, and the estimated float ambiguities were less biased. By de-weighting the measurements instead of excluding them entirely, the algorithm effectively mitigated the effects of multipath while maintaining the same satellite geometry. Overall, the implemented algorithm demonstrated its effectiveness in addressing non-Gaussian measurements and improving the accuracy of the float solution for ambiguity resolution, particularly in scenarios where multipath distortions were present.

7.1.5 Recommendations for Future Work

The assessment of the performance of the SIR PF for estimating the full geometry-based float solution was done in this thesis. This analysis demonstrates the most important features and advantages of using the Particle Filter. The main limitation of the PF is the computational cost as a large number of particles are needed to estimate a 9+N dimension state vector. However, The SIR PF with a map-constraint can be applied to help the states converge faster at the first 30 seconds after the filter has been initialized, and the EKF can be used afterwards. Moreover, if the vehicle is navigating through an urban canyon environment and, therefore, prone to receive observations with multipath, the SIR PF can be used to de-weight this measurements and improve the position accuracy. The PF allows to add any constraint despite the resulting distribution or the nonlinearity of the measurement model. Considering the limitations and unique features of the SIR PF for estimating the float solution, some suggestions for future work may include:

- Integrate other sensor data in order to assess which one has greater impact on the position accuracy and on the ambiguity domain.
- In terms of hardware, evaluate how the resources can be managed to increase efficiency of the Particle Filter.
- Test multiple kinds of distributions for weighting the measurements as there is no limitation in this aspect for the PF.
- This research was carried out using L1 GPS measurements in order to keep the state vector as small as possible, however, this research can be extended to the multi-constellation and multi-frequency case.
- The addition of other non-holonomic constraints to the estimation, regardless how non-Gaussian the resulting distribution becomes could be investigated.

The Particle Filter has gained popularity in recent years because of its advantages and the highly developed computers existing nowadays. Monte Carlo methods were only possible in specialized computers and were applied on particular robotics tracking and navigation cases. As the technology keeps increasing at a fast pace, the main limitation of the PF eventually will disappear and this estimator will become even more popular. The study and description of a filter of this kind applied to GPS navigation, and specifically to estimate the float solution for ambiguity resolution is pertinent.

References

[1] P. J. G. Teunissen, "A new method for fast carrier phase ambiguity estimation," *in Proceedings of the IEEE Position Location and Navigation Symposium, Las Vegas, NV (USA)*, pp. 562 – 573, 05 1994.

[2] P. J. G. Teunissen, P. J. De Jonge, and C. C. J. M. Tiberius, "Performance of the lambda method for fast gps ambiguity resolution," *Navigation*, vol. 44, no. 3, pp. 373–383, 1997.

[3] C. Huang, Y. Jiang, and K. O'Keefe, "Wheel odometry aided visual-inertial odometry for land vehicle navigation in winter urban environments," *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 2237–2251, 09 2020.

[4] E. D. Kaplan and C. Hegarty, "Understanding gps principles and applications," *Artch Housem, Norwood, MA, USA*, 2006.

[5] H. Euler and B. Schaffrin, "On a measure for the discernibility between different ambiguity solutions in the static-kinematic gps-mode," *Kinematic Systems in Geodesy, Surveying and Remote Sensing*, vol. 107, pp. 285–295, 05 1991.

[6] B. Remondi, *Using the Global Positioning System (GPS) Phase Observable for Relative Geodesy: Modeling, Processing, and Results.* U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Ocean Service, 1989.

[7] R. Hatch and H. J. Euler, "Comparison of several arof kinematic techniques," *Proceedings of ION GPS-94, Salt Lake City, Utah*, vol. September 20-23, pp. 363–370, 1994.

[8] E. Frei and G. Beutler, "Rapid static positioning based on the fast ambiguity resolution approach fara: Theory and first results," *Manuscripta Geodaetica*, vol. 15, no. 6, pp. 325–356, 1990.

[9] H. J. Euler and H. Landau, "Fast gps ambiguity resolution on-the-fly for real-time application," *in 6th h International Geodetic Symposium on Satellite Positioning, Columbus, Ohio, USA*, pp. 650–659, 01 1992.

[10] M. Martin-Neira, M. Toledo, and A. Pelaez, "The null space method for gps integer ambiguity resolution," *Proceedings of DSNS'95, Bergen, Norway*, vol. 24-28 April, no. No.31, p. 8, 1995.

[11] D. Chen and G. Lachapelle, "A comparison of the fasf and least-squares search algorithms for on the on-the-fly ambiguity resolution," *Journal of The Institute of Navigation,*, vol. 42, no. 2, pp. 371–390, 06 1995.

[12] D. Kim and R. Langley, "An optimized least-squares technique for improving ambiguity resolution performance and computational efficiency," *Proceedings of ION GPS'99, Nashville, TN, USA*, pp. 1579–1588, 09 1999.

[13] D. Kim and R. Langley, "GPS ambiguity resolution and validation: Methodologies, trends and issues," *7th GNSS Workshop - International Symposium on GPS/GNSS*, pp. 213–221, 11 2000.

[14] S. Verhagen, "The gnss integer ambiguities: Estimation and validation," *Delft Institute of Earth Observation and Space Systems, Delft University of Technology*, 01 2005.

[15] P. J. G. Teunissen, "Least-squares estimation of the integer gps ambiguities," *Technical Report; Delft Geodetic Computing Centre, Delft University of Technology*, 01 1993.

[16] C. Qian, H. Zhang, W. Li, and et al., "A lidar aiding ambiguity resolution method using fuzzy one-to-many feature matching," *Journal of Geodesy*, vol. 94, no. 98, p. 98, 2020.

[17] B. M. Scherzinger, "Precise robust positioning with inertial/gps rtk," *Proceedings of the 13th International Technical Meeting for the Satellite Division of the Institute of Navigation (ION GPS),Utah, USA*, pp. 115–162, 2000.

[18] H. Han, J. Wang, J. Wang, and A. H. Moraleda, "Reliable partial ambiguity resolution for single-frequency gps/bds and ins integration," *GPS Solutions*, vol. 21, pp. 251–264, 2017.

[19] P. Davidson, J. Collin, and J. Takala, "Application of particle filters to map-matching algorithm," *17th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2010 - Proceedings*, vol. 2, pp. 285–292, 10 2011.

[20] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 02 2002.

[21] B. Betti, M. Crespi, and F. Sansò, "A geometric illustration of ambiguity resolution in gps theory and a bayesian approach," *Manuscripta geodetica*, vol. **18**, pp. 317–330, 1993.

[22] B. Azimi-Sadjadi and P. Krishnaprasad, "Integer ambiguity resolution in gps using particle filtering," *Proceedings of the American Control Conference, Arlington, VA, USA*, vol. 5, pp. 3761 – 3766, 02 2001.

[23] P. Henderson, J. Raquet, and P. Maybeck, "A multiple filter approach for precise kinematic dgps positioning and carrier-phase ambiguity resolution," *Navigation*, vol. 49, pp. 149–160, 09 2002.

[24] S. S. Hwang and J. Speyer, "Relative gps carrier-phase positioning using particle filters with position samples," *Proceedings of the American Control Conference, ST. Louis, MO, USA*, pp. 4171 – 4177, 07 2009.

[25] M. Sahmoudi and R. J. Landry, "A nonlinear filtering approach for robust multi-gnss rtk positioning in presence of multipath and ionospheric delays," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, pp. 764 – 776, 11 2009.

[26] B. Karl, W. Avishai, and D. C. Stefano, "Gnss ambiguity resolution by adaptive mixture kalman filter," *International Conference on Information Fusion (FUSION)*, vol. **TR2018-103**, 2018.

[27] T. Suzuki, "Mobile robot localization with gnss multipath detection using pseudorange residuals," *Advanced Robotics*, vol. 33, pp. 1–12, 05 2019.

[28] B. Azimi-Sadjadi and P. Krishnaprasad, "Approximate nonlinear filtering with applications to navigation." *Institute for systems research, Institute of Maryland, PhD thesis*, vol. 5, 2001.

[29] F. G. Toro, D. E. D. Fuentes, Debiao Lu, U. Becker, H. Manz, and Baigen Cai, "Particle filter technique for position estimation in gnss-based localization systems," *2015 International Association of Institutes of Navigation World Congress (IAIN)*, pp. 1–8, 2015.

[30] P. Xie and M. G. Petovello, "Measuring gnss multipath distributions in urban canyon environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 366–377, 2015.

[31] J. Georgy, A. Noureldin, M. J. Korenberg, and M. M. Bayoumi, "Low-cost three-dimensional navigation solution for riss/gps integration using mixture particle filter," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 599–615, 2010.

[32] H. Lu, S. Zhang, Y. Dong, and X. Lin, "A wi-fi/gps integrated system for urban vehicle positioning," *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 1663–1668, 2010.

[33] P. Xie, Z. He, and M. G. Petovello, "Performance evaluation of integration of on- board vehicle sensors with high sensitivity re- ceivers in urban canyons," *Proceedings of Chinese Satellite Navigation Conference*, 05 2014.

[34] T. Humphreys, M. Murrian, and L. Narula, "Low-cost precise vehicular positioning in urban environments," 04 2018, pp. 456–471.

[35] P. J. G. Teunissen, "Least-squares estimation of the integer gps ambiguities," *Technical Report; Delft Geodetic Computing Centre, Delft University of Technology*, 01 1993.

[36] P. J. G. Teunissen, "Success probability of integer gps ambiguity rounding and bootstrapping," *Journal of Geodesy*, vol. 72, pp. 606–612, 01 1998.

[37] B. Hoffman, L. Herbert, and E. Wasle, *GNSS – Global Navigation Satellite Systems.* Springer-Verlag Wien, 2008.

[38] J. Spilker, A. Penina, W. Parkinson, and P. Enge, *Global positioning system: Theory and applications, Vol 1.* American Institute of Aero-nautics and Astronautics, 1996.

[39] A. Leick, L. Rapoport, and D. Tatarkinov, *GPS Satellite Surveying*, 4th ed. Wiley, 2015.

[40] P. J. G. Teunissen, "On the integer normal distribution of the gps amibiguities." *Artificial Satellites*, no. 33(2), pp. 49–64, 1998.

[41] P. J. G. Teunissen, "Class of unbiased integer gps ambiguity estimators," *Artificial Satellites*, vol. 33, no. 1, pp. 1–10, 1998.

[42] D. Dong and Y. Bock, "Global positioning system network analysis with phase ambiguity resolution applied to crustal deformation studies in california," *Journal of Geophysical Research, Nashvillem Tennessee*, vol. 94, pp. 3949–3966, 1989.

[43] G. Blewitt, "Carrier-phase ambiguity resolution for the global positioning system aoolied to geodetic baselines up to 2000 km." *Journal of Geophysical Research, Nashvillem Tennessee*, vol. 94 (B8), pp. 10 187–10 302, 1989.

[44] P. J. G. Teunissen, "The probability distribution of the gps baseline for a class of integer ambiguity estimators," *Journal of Geodesy*, no. 73(5), pp. 275–284, 1999.

[45] P. De Jonge and C. Tiberius, "The lambda method for integer ambiguity estimation: implementation aspects," *Delft Geodetic Computing Centre LGR Series*, vol. 12, 1998.

[46] H. Euler and B. Schaffrin, "On a measure for the discernibility between different ambiguity solutions in the static-kinematic gps-mode," *Kinematic Systems in Geodesy, Surveying and Remote Sensing*, vol. 107, pp. 285–295, 05 1991.

[47] S. Verhagen, "Integer ambiguity validation: an open problem?" *GPS Solutions*, vol. 8, pp. 36–43, 2004.

[48] P. J. G. Teunissen, "An optimality property of the integer least-squares estimator," *Journal of Geodesy*, no. 73, pp. 587–593, 1999.

[49] J. Ogundare, "Understanding least squares estimation and geomatics data analysis," *New Jersey: John Wiley*, 11 2019.

[50] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2015.

[51] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[52] S. Sarkka, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[53] D. Wells and E. Krakiwsky, "The method of least squares," *Lecture Notes, University of New Brunswick*, 5 1971.

[54] R. Brown and Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering : With MATLAB Exercises*, 4th ed. Wiley, 2015.

[55] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, 1995, pp. 1628–1632 vol.3.

[56] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[57] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Defense, Security, and Sensing*, 1997.

[58] E. Wan and R. V. D. Merwe, "Chapter 7 the unscented kalman filter," in *Kalman Filtering and Neural Networks*. Wiley, 2001, pp. 221–280.

[59] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds.,

vol. 13. MIT Press, 2000. [Online]. Available: https://proceedings.neurips.cc/paper/2000/file/f5c3dd7514bf620a1b85450d2ae374b1-Paper.pdf

[60] Y. Bar-Shalom, X. Li, and T. Kirubarajan, in *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York: John Wiley & Sons, 2001.

[61] K.-R. Koch, *Introduction to Bayesian Statistics*, 1st ed. Springer-Berlag, 2000.

[62] J. O. Ogundare, in *Understanding Least Squares Estimation and Geomatics Data Analysis*. New Jersey: John Wiley, 2019.

[63] H. Pishro-Nik, "Introduction to probability, statistics and random processes." Kappa Research, 2014.

[64] B. Ristic, S. Arulampalam, and N. J. Gordon, "Beyond the kalman filter: Particle filters for tracking applications." Artech House, 2004.

[65] Z. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, 01 2003.

[66] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[67] A. Doucet, N. de Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*, 2001.

[68] A. Kong, J. S. Liu, and W. Wong, "Sequential imputations and bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 03 1994.

[69] J. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 04 1998.

[70] A. Doucet and A. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, 01 2009.

[71] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/10618600.1996.10474692

[72] V. Zaritskii, V. Svetnik, and L. Shimelevich, "Monte carlo technique in problems of optimal data processing," *Automation and Remote Control*, vol. 10, no. 3, pp. 197–208, 1975.

[73] H. Akashi and H. Kumamoto, "Random sampling approach to state estimation in switching environments," *Automatica*, vol. 13, no. 4, pp. 429–434, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109877900280

[74] D. Salmond, "Tracking in uncertain environments," *Royal Aircraft Establishment Farnborough*, 09 1989.

[75] M. A. Woodbury, *Inverting modified matrices*. Princeton, NJ: Department of Statistics, Princeton University, 1950.

[76] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Cambridge: Academic Press, 1970.

[77] J. Zuo, Y. Jia, W. Zhang, and Q. Gao, "Particle filter with importance density function generated by updated system equation," *Journal of Central South University*, vol. 20, no. 10, pp. 2700–2707, 10 2013.

[78] A. Doucet, N. Gordon, and V. Krishnamurty, "Particle filters for state estimation of jump markov linear systems." *IEEE Transactions on Signal Processing,*, vol. 49(3), pp. 613–624, 2001.

[79] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[80] T. Schon, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[81] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity analysis of the marginalized particle filter," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4408–4411, 2005.

[82] T. Takasu and A. Yasuda, "Development of the low-cost rtk-gps receiver with an open source program package rtklib," *Laboratory of Satellite Navigation, Tokyo University of Marine Science and Technology*, 04 2013.

[83] B. Aumayer, M. Petovello, and G. Lachapelle, "Stereo-vision aided gnss for automotive navigation in challenging environments," *26th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2013, Nashville, TN*, vol. 1, pp. 511–520, 01 2013.

[84] P. V. Gakne and M. Petovello, "Assessing image segmentation algorithms for sky identification in gnss," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–7.

[85] P. J. G. Teunissen, "Integer estimation in the presence of biases," *Journal of Geodesy*, vol. 75, pp. 399–407, 2001.

[86] B. W. Parkinson and J. J. S. Jr., "Land vehicle navigation and tracking," in *Global Positioning System: Theory and Applications, Volume II.* Reston, VA: American Institute of Aeronautics and Astronautics, 2004, ch. Land Vehicle Navigation and Tracking, Robert L. French.

[87] S. Dmitriev, O. Stepanov, B. Rivkin, and D. Koshaev, "Optimal map-matching for car navigation system," *Proc. of 6th International Conference on Integrated Navigation Systems, St. Petersburg*, vol. 2, 1999.

[88] S. Syed and M. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," *Proc. of ION National Technical Meeting, San Diego, CA*, pp. 26–28, 6 2004.

[89] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden markov model with precomputation," *International Journal of Geographical Information Science*, vol. 0, pp. 1–24, 11 2017.

[90] Y. Zhao, "Vehicle location and navigation system," *Artech House*, pp. 83–103, 1997.

[91] "Open Street Map, url = https://www.openstreetmap.org,."

[92] "Java Open Street Map Documentation, url = https://josm.openstreetmap.de/,."

[93] A. Pirsiavash, A. Broumandan, G. Lachapelle, and K. O'Keefe, "Detection and de-weighting of multipath-affected measurements in a gps/galileo combined solution," in *2019 European Navigation Conference (ENC)*, 2019, pp. 1–11.

[94] M. Zaheri, A. Broumandan, and G. Lachapelle, "Comparing detection performance of polarization and spatial diversity for indoor gnss applications," in *IEEE/ION Position, Location and Navigation Symposium*, 2010, pp. 737–744.

[95] A. J. Van, "Theory and performance of narrow correlator spacing in a gps receiver," *NAVIGATION: Journal of The Institute of Navigation*, vol. 39, 1992.

[96] L. Garin and R. J.M., "Enhanced strobe correlator multipath rejection for code and carrier," *Proceedings of the 10th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, pp. 559–568, 1992.

[97] Lee, Jin Hyuk, Lim, Deok Won, Noh, Jae Hee, Jo, Gwang Hee, Park, Chansik, Ahn, Jae Min, and Lee, Sang Jeong, "A gps multipath mitigation technique using correlators with variable chip spacing," *E3S Web Conf.*, vol. 94, p. 03006, 2019. [Online]. Available: https://doi.org/10.1051/e3sconf/20199403006

[98] A. Pirsiavash, A. Broumandan, and G. Lachapelle, "Characterization of signal quality monitoring techniques for multipath detection in gnss applications," *Sensors*, vol. 17, no. 7, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/7/1579

[99] A. Pirsiavash, A. Broumandan, G. Lachapelle, and K. O'Keefe, "Gnss measurement weighting to improve navigation integrity in multipath environments," *6th ESA International Colloquium on Scientific and Fundamental Aspects of the Galileo*, p. 8, 10 2017.

[100] A. Pirsiavash, A. Broumandan, and G. Lachapelle, "How effective are signal quality monitoring techniques for gnss multipath detection?" *Inside GNSS Magazine*, vol. 13, pp. 40–47, 01 2018.

[101] A. Beitler, A. Tollkuehn, D. Giustiniano, and B. Plattner, "Cmcd: Multipath detection for mobile gnss receivers," *Institute of Navigation International Technical Meeting 2015, ITM 2015*, pp. 455–464, 01 2015.

[102] P. J. G. Teunissen and O. Monterbruck, "Batch and recursive model validation," in *Springer Handbook of Global Navigation Satellite Systems*. Springer, 2017, ch. Batch and Recursive Model Validation, Peter Teunissen, pp. 711–723.

[103] K. Morad, B. R. Young, and W. Y. Svrcek, "Rectification of plant measurements using a statistical framework," *Computers & Chemical Engineering*, vol. 29, no. 5, pp. 919–940, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135404002029

[104] T. Chen, J. Morris, and E. Martin, "Dynamic data rectification using particle filters," *Computers & Chemical Engineering*, vol. 32, no. 3, pp. 451–462, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135407000634

[105] X. Chen, "Statistical multipath model comparative analysis of different gnss orbits in static urban canyon environment," *Advances in Space Research*, vol. 62, 06 2018.

[106] D. Wells and E. Krakiwsky, "Coordinate systems in geodesy," *Lecture Notes, University of New Brunswick*, 5 1971.

[107] B. R. Bowring, "The accuracy of geodetic latitude and height equations," *Survey Review*, vol. 28, pp. 202–206, 1985.

[108] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012.

[109] J. K. Blitzstein and J. Hwang, *Introduction to Probability*, 2nd ed. CRC Press, 2019.

Appendix A

RTKLIB GNSS software

RTKLIB is an open source program developed by Tomoji Takasu in 2007 [82]. This software consist of several application programs and libraries developed in C and C++ which can be executed either on a Graphics User Interface (GUI) application or a Command-line User Interface (CUI). The system requirements for executing an application of RTKLIB are minimal, as the library functions are developed in C, the libraries or applications can be executed on any operating system as long as a C/C++ compiler is installed. However, it should be considered that in order to run any GUI app, a Microsoft Windows operating system is required. The CUI applications can be executed in any operating system, and is the suggested way to do it when modifications are being made to any of the routines. Although the algorithms written in RTKLIB are complex, the simplicity for modifying the code for experiments is the main reason why it is popular for processing GNSS data. Within the libraries and applications, RTKLIB provides a solution for standard and precise positioning. The main features of RTKLIB are the following:

- It supports multi-constellation: GPS, Galileo, GLONASS, QZSS, BeiDou and SBAS
- It provides a real time and post-processing solution for: Single point positioning, Differential positioning, kinematic with static and moving baseline, PPP-kinematic, PPP-static and PPP-fixed.
- Supports many standard formats and protocols for GNSS
- Supports external communication via serial, TCP/IP, NTRIP, local log file and FTP/HTTP.

RTKLIB library functions include several algorithms and mathematical functions in order to process GNSS data and perform the computations. This functions may include matrices operations, positioning models, coordinates transformation, PPP algorithms, integer ambiguity resolution and can be used in any

independent program by importing the header files that include the function needed. The main applications of RTKLIB are in Table A.1

Table A.1: Main RTKLIB applications.

GUI	**CUI**	**Description**
RTK-NAVI	RTKRCV	Real-time positioning
STRSVR	STR2STR	Communication Server
RTKPOST	RNX2RTKP	Post-processing analysis
RTKCONV	CONVBIN	RINEX converter
RTKPLOT	-	Plot solutions
RTKGET	-	Downloader for GNSS data

A.1 RNX2RTKP

The RTKLIB CUI module for post processing analysis RNX2RTKP was used in this thesis. This application inputs standard RINEX observation data for the base-station and rover and the navigation file when performing a differential positioning. Several options about the entire process can be modified, including the time lapse of the analysis, time interval, coordinate system of the solution, the minimal ratio to fix the ambiguities. In Figure A.1 the GUI application RTPOST is showed. The command version is compiled in a better way on a UNIX environment, i.e. Linux or MacOS, because it is easier to inlcude compilation flags to make the execution more customized. When compiling the application, it is possible to set the same options as flags, for example: **./executable input -m 15 -p mode output** where **-m** and **-p** are the option flags for elevation mask and positioning mode (single point, PPP-kinematic, fixed, static, moving-base and Kinematic). However, it is suggested to include the configuration file in the compile command: **./executable input file.conf output**. In Figure A.2 it is shown an example of a command line for executing RNX2RTKP application, where $file.conf$ is the configuration file containing all the options required, *commandresultpy.pos* is the output file, then a time interval is passed in order to analyze just a segment of data, finally the observation and navigations data files are input.

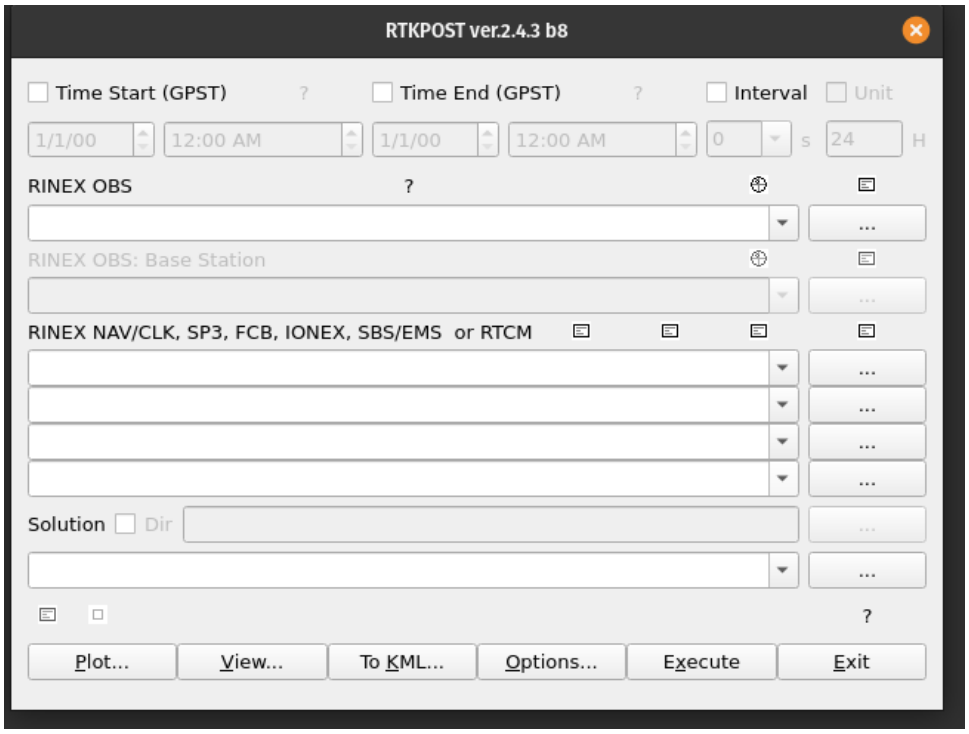


Figure A.1: RTPOST GUI application for GNSS post-processing analysis

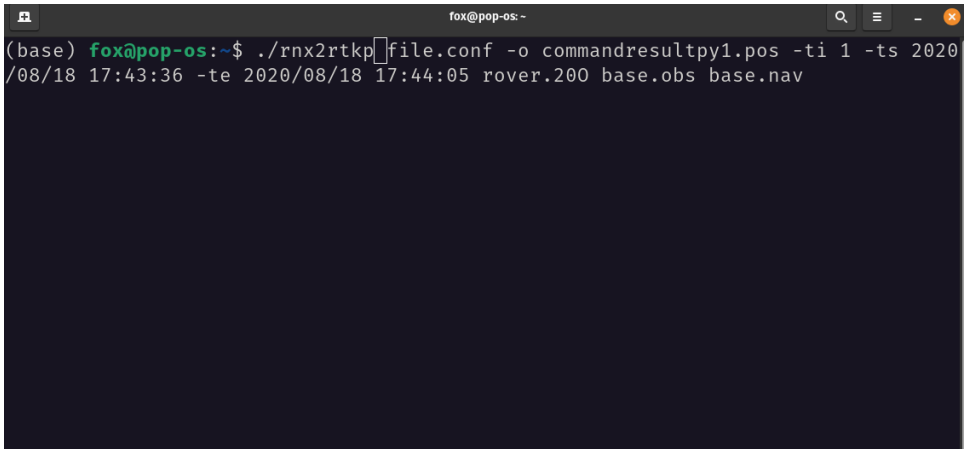


Figure A.2: RNX2RTKP CUI application for GNSS post-processing analysis

Appendix B

Coordinate systems transformation

In this appendix, a description about the transformation from a geographic coordinate system to a local geodetic frame is provided. This transformation is useful when estimating position and velocity vectors from GNSS measurements that are referenced to a geodetic coordinate system but a description about the navigation trough a specific are or project site is needed. The standard or root mean square errors of the estimated position in a local geodetic system are useful for interpreting the results.

B.0.1 Coordinate Systems

The geodetic coordinate system is defined as follows [106]:

- The origin is at the center of the reference ellipsoid.
- The primary pole or z axis is the semi-minor axis of the ellipsoid. The plane perpendicular to this axis is called equatorial plane.
- The secondary axis or x axis is the intersection of the equatorial plane and the Greenwich meridian plane. Passes through the pojnt P_0 ($\lambda = 0$, $\phi = 0$).
- The y axis is chosen to form a right-handed system.

However, a local geodetic frame (LGF) is a local coordinate system that is referenced to a specific location. The local geodetic frame or East North Up (ENU) system is defined as follows [106]:

- The origin is at any point P (λ , ϕ , h) referred to a given ellipsoid.
- The Up axis (U) is normal to the surface of the reference ellipsoid. It is assumed that the gravity vector coincides with this axis. Primary plane is defined as the plane containing the origin and perpendicular

to the primary pole.

- The North axis (N) (primary axis) is the intersection of the primary plane and the plane containing the semi-minor axis of the ellipsoid and the origin.
- The East axis (E) is directed east to complete a left-handed system.

Figure B.1 is a graphical description of the relation between the geodetic and local geodetic coordinate frames.

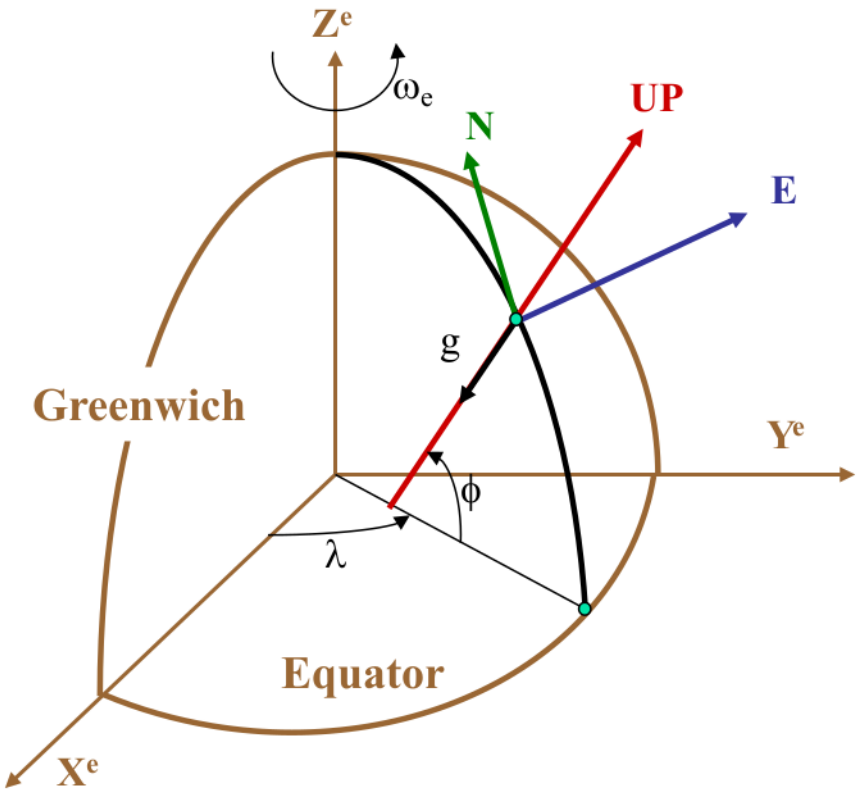


Figure B.1: Local navigation system (East North Up system).

B.0.2 Transformation Equations

The transformation of coordinates from a curvilinear geodetic system to a cartesian frame is described by the following equations.

$$e^2 = 2f - f^2 \tag{B.1}$$

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \tag{B.2}$$

$$x = (N + h)\cos\phi\cos\lambda \tag{B.3}$$

$$y = (N + h)\cos\phi\sin\lambda \tag{B.4}$$

$$z = [(1 - e^2)N + h]\sin\phi \tag{B.5}$$

where a is the semi-major axis of the WGS84 ellipsoid, f is the flattening factor, e^2 is the eccentricity squared, N is the radius of curvature in the prime vertical, h is the height above the ellipsoid, and x, y, and z are the geodetic cartesian coordinates in meters. The inverse transformation from the cartesian to the curvilinear geodetic frame is as follows (non-iterative method [107]):

$$\phi = \arctan\left(\frac{z}{\sqrt{x^2+y^2}(1-\frac{e^2 b z}{a\sqrt{x^2+y^2}})}\right) \quad \lambda = \arctan\left(\frac{y}{x}\right) \quad h = \frac{\sqrt{x^2+y^2}}{\cos\phi} - N \tag{B.6}$$

The transformation from the geodetic to the local geodetic frame is described by the following expression:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Geodetic} = P_2 R_2(\phi - \pi/2) R_3(\lambda - \pi) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{LocalGeodetic} \tag{B.7}$$

and the inverse transformation is as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{LocalGeodetic} = R_3(\pi - \lambda) R_2(\pi/2 - \phi) P_2 \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Geodetic} \tag{B.8}$$

with rotational matrices R_3 and R_2 as:

$$R_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{B.9}$$

and:

$$R_3 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.10}$$

$P2$ is the reflection matrix described as:

$$P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.11}$$

Appendix C

Cholesky decomposition

The Cholesky decomposition is a transformation or factorization of a positive-definite Hermitian matrix into the product of a lower triangular matrix and its conjugate transpose. It is useful in sampling methods such as Monte-Carlo estimators (Particle Filter) to produce a set of samples with the covariance properties of the system. Consider having a set of uncorrelated samples s and the lower-triangular matrix L from the Cholesky decomposition of matrix A, then is possible to compute a sample vector Ls correlated according to L. The algorithm is to compute the Cholesky decomposition of a positive-definite matrix A is as follows [108]:

Algorithm 5 Cholesky Decomposition

Require: A positive-definite matrix A of order n.

Ensure: The lower triangular Cholesky factor L such that $A = LL^{\mathrm{T}}$.

1: Let $a_{11} = \sqrt{A_{11}}$ and $L_{11} = a_{11}$.
2: **for** $j = 2$ to n **do**
3: Compute $L_{j1} = \frac{A_{j1}}{L_{11}}$.
4: Compute $s = \sum_{k=1}^{j-1} L_{jk}^2$.
5: Compute $L_{jj} = \sqrt{A_{jj} - s}$.
6: **for** $i = j + 1$ to n **do**
7: Compute $L_{ij} = \frac{A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk}}{L_{jj}}$.
8: **end for**
9: **end for**

The Cholesky decomposition is used in this thesis to ensure that the correlated process noise Q_k is added to the particles at the prediction step instead of just adding the uncorrelated variances of the diagonal values

of Q_k:

$$x_{k+1} = \Phi_{k+1} x_k + L_k \cdot \mathcal{N}(0,\, Q_k) \tag{C.1}$$

where L_k is the lower-triangular matrix after applying the Cholesky decomposition to Q_k.

Appendix D

Double Difference Transformation Matrix

A description of the algorithm used to compute the double difference transformation matrix to change the state quantity from single to double difference is provided in this appendix. This algorithm is implemented in RTKLIB software [82] to simplify the base-satellite book-keeping, as well as the observation covariance matrix. The DD transformation matrix is applied for the ambiguity resolution step. The algorithm is as follows:

1. Set all the diagonal elements of the matrix to 1 of the state vector that represent the position, velocity and acceleration of the receiver.

2. Iterate over each satellite observed and used for the estimation. The index of the first satellite that meets the criteria to be selected as base is saved. The satellite elevation needs to meet the elevation mask, the signal needs to be tracked for a sufficient amount of time with no cycle slips.

3. Iterate over each satellite, skipping the base satellite. Check the visibility of the satellite according to mask angle and if the signal has been tracked for a sufficient amount of time with no cycle slips. If the criteria are met, the corresponding element is set to 1 for the base satellite and -1 for the current satellite.

The result of the algorithm described is a matrix of n rows by n columns, n being the dimension of the state vector. The matrix is identity matrix for the first 9 states (position, velocity and acceleration) and the rest elements are filled according to the algorithm described. The transformation from SD to DD is as

follows:

$$x_{DD} = D_k^T x_{SD} \tag{D.1}$$

and the transformed covariance matrix as:

$$P_{DD} = D_k^T P_{SD} D_k^T \tag{D.2}$$

where x_{SD} is the state vector with the SD float ambiguties with its covariance matrix P_{SD} and D_k is the DD transformation matrix. Assuming the state vector includes only the position state and three SD float ambiguities, and example of the 6x6 DD transformation matrix would be as follows:

$$D_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

It can be noted that the product of the DD matrix and the SD vector reduces the state vector by 1 dimension due to the base satellite.

Appendix E

Non-linear measurement models and non-Gaussian distributions

E.1 Non-linear models

In the context of state estimation, measurements serve to update or correct the predicted state vector. The measurement model h establishes a connection between the observables $y = (y_1, y_2, y_3, ..., , y_n)$ and the unknown state vector $x = (x_1, x_2, x_3, ..., , x_m)$ [49]. This mathematical representation assumes a linear form for any of the states x_m if the plot of $h(x_m)$ forms a straight line. It's important to note that a model satisfying this criterion takes the shape $h(x) = ax + b$ (line equation). However, in practice, most state estimation problems are nonlinear, causing the graph of $h(x_m)$ to deviate from a straight line.

For the carrier and code measurement models (as depicted in equations (2.1) and (2.2)), the introduction of the position state induces nonlinearity in the model, particularly within the geometric distance term ρ_r^i. Despite this nonlinearity, the substantial separation between satellite i and receiver r yields a $h(x)$ plot with relatively mild nonlinear characteristics. Illustrated in Figure E.1, the graph shows the geometric distance ρ varying along the x coordinate from the position state vector. To further assess the degree of linearity within the model, a linear approximation has been achieved using least squares, yielding the slope and y-intercept of the line equation. A method to quantify the extent of linearity in the model is by computing the coefficient of determination R^2 between the model and the linear equation. The coefficient R^2, is a statistical measure that indicates how well the independent variables explain the variability of the dependent variable [109]. R^2 is described as follows:

$$R^2 = 1 - \frac{SSR}{SST} \tag{E.1}$$

where SSR (Sum of Squares Residual) is the sum of squared differences between the actual values of the dependent variable and the predicted values from the regression model and SST (Total Sum of Squares) is the sum of squared differences between the actual values of the dependent variable and the mean of the dependent variable. Note that R^2 ranges from 0 to 1, where:

- $R^2 = 0$ indicates that the model does not explain any variability in the dependent variable.
- $R^2 = 1$ indicates that the model explains all the variability in the dependent variable.

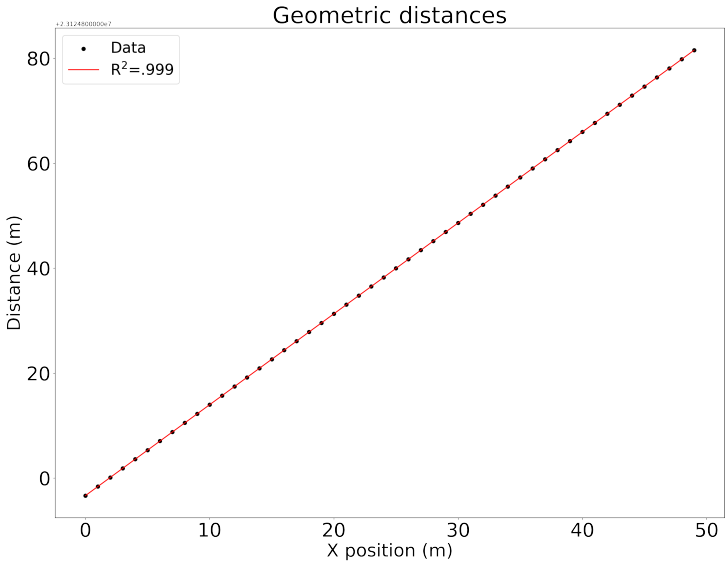


Figure E.1: Linear function. Geometric distance ρ and x position state vector rate change. Due to the large distance between the satellite and the receiver, the function is almost linear. R^2 statistic is close to 1.

Measurement models from additional sensors to aid the position estimation usually involve trigonometric functions to describe different maneuvers. For example, the *cosine* function is shown in Figure E.2 along with a line fitted to it. The R^2 value is lower than the previous example, depicting more nonlinearity.

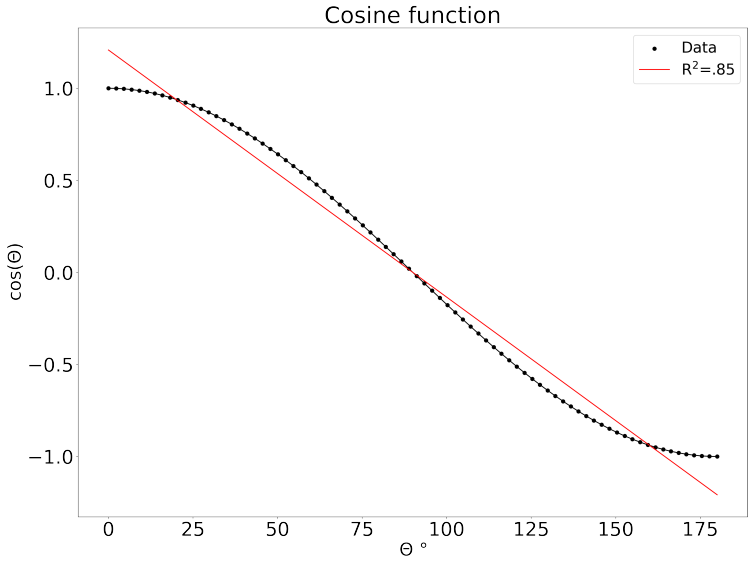


Figure E.2: Nonlinear function. Angle Θ and $cos(\Theta)$ values. R^2 statistic is .85 depicting less *linearity*.

The least-squares and EKF estimators use a first-order point of expansion Taylor series to approximate a non-linear model into its linear form. From a geometric perspective, the Jacobian matrix containing the derivatives $\frac{\partial h(x)}{\partial x}$ represents the slope of a tangent line to the nonlinear function h at the point $h(x)$. Two limitations arise when the measurement model is highly non-linear and the first-order Taylor series approximation is used for linearization. Firstly, computing the derivatives for the Jacobian matrix H is a complex task. Secondly, the approximation does not adequately describe the relationship between the unknown states and the measurements, which can result in filter divergence. In Figure E.3 a cosine function with its linear approximation evaluated in 4 different points. As mentioned, the Jacobian contains the tangent lines to the cosine function evaluated at x.

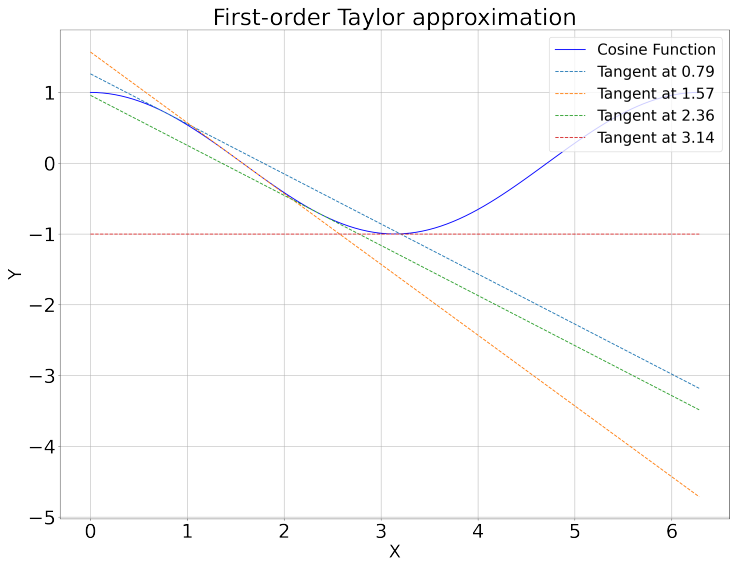


Figure E.3: Nonlinear function. Angle Θ and $cos(\Theta)$ values. R^2 statistic is .85 depicting less *linearity*.

In Chapter 5, the map-aiding algorithm implemented in the EKF is described. As explained, when digital map information is used to weigh the particles, each particle's weight is determined by the Euclidean distance and directional difference to the closest lane of a road segment. Similar to the code and carrier measurement models, the position state enters the model nonlinearly when estimating the Euclidean distance between each particle and its nearest lane. However, these distances are generally shorter than ρ_r^i, making the model more nonlinear than the code and carrier GNSS equations. Moreover, multiple lanes are considered, further complicating the model's approximation in its linear form.

E.2 Non-Gaussian Distributions

In GNSS positioning, measurements are treated as continuous random variables assumed to have Gaussian-distributed errors [49]. Constructing histograms of these measurements provides insight into the probability distribution of the errors inherent in the measurements. Similarly, histograms of measurement errors directly depict the probability distribution of these errors. In most cases, the Gaussian assumption about measurement errors holds true. However, when measurements are influenced by external sources of error, such as when satellite signals are reflected or refracted by nearby surfaces, the distribution no longer conforms to a Gaussian pattern [80]. Graphical analysis entails plotting the histogram of measurement errors to assess whether it adheres to a Gaussian curve. Additionally, a quantitative assessment of the distribution's "Gaussianity" level can be conducted by fitting the equation of the normal distribution curve to the probability

distribution of measurements and estimating the R^2 metric.

In Figures E.4 and E.5, the distributions of pseudorange measurements are shown both without and with multipath errors, respectively. This distributions correspond to segments of the data used in this thesis. As illustrated, the Gaussian curve fitted to the pdf of measurements without multipath effects offers a better description. The R^2 metric for the first distribution is .95, and for the second distribution, it is .45. As anticipated, the R^2 value in the presence of multipath errors is low, signifying a certain degree of non-Gaussianity. Furthermore, the statistical analysis conducted in [105] demonstrates that the non-line-of-sight (NLOS) multipath error distribution adheres to a Gamma distribution model. Given that the Kalman filter always assumes Gaussian distributions for measurement errors, the estimation becomes biased when measurements are distorted by multipath effects. Chapter 6 discusses the particle filter's capability to utilize any likelihood function for weighting measurements which are detected as outliers and do not follow a Gaussian distribution.

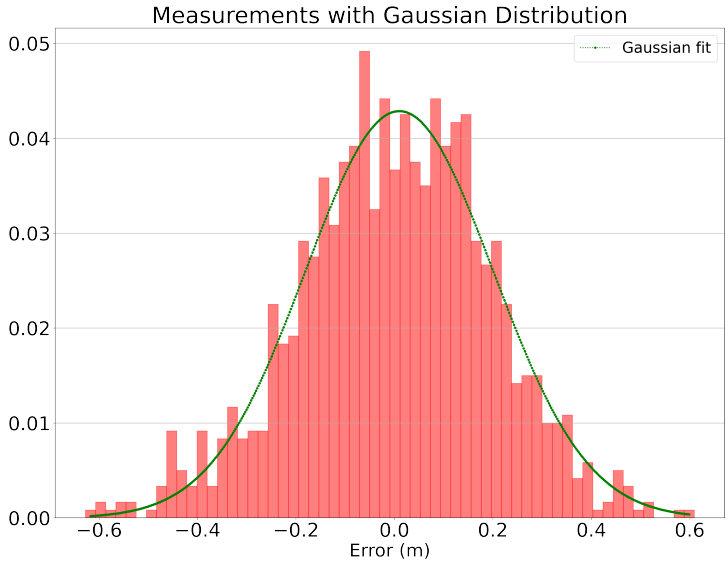


Figure E.4: Gaussian distribution of code measurement errors without multipath effects.

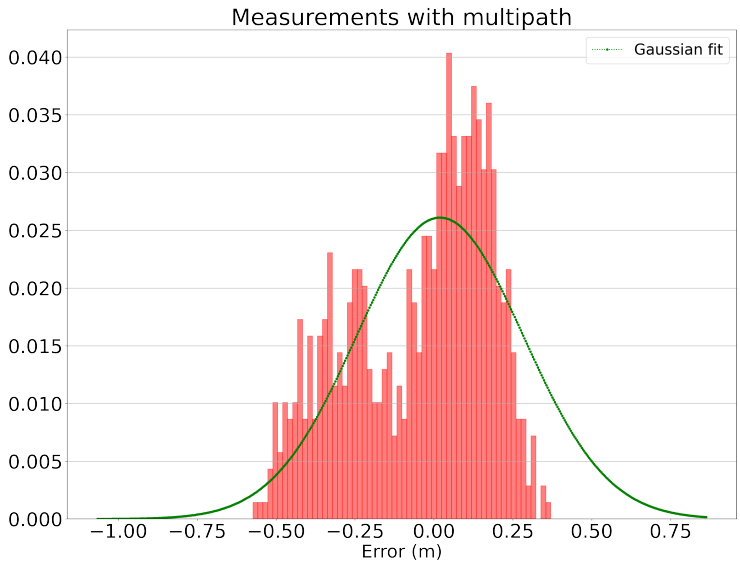


Figure E.5: Example of non-Gaussian distribution for code measurements distorted by mulitpath .