Aalborg Universitet

Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks

Figueiras, Joao

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Figueiras, J. (2008). Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks. Aalborg: Department of Electronic Systems, Aalborg University.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: May 01, 2017

AALBORG UNIVERSITET

Department of Mobile Communications

Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks

by

João Carlos Prazeres Figueiras, MsC

Dissertation

Presented to the Faculty of Engineering, Science and Medicine of the Graduate School of

Aalborg Universitet

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Aalborg Universitet

November 2007

Copyright

by

João Carlos Prazeres Figueiras

2007

The Dissertation Committee for João Carlos Prazeres Figueiras
certifies that this is the approved version of the following dissertation:

Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks

Committee:

Associate professor Hans-Peter Schwefel, M.Sc.EE, Ph.D.
AAU, Aalborg, Denmark (Supervisor)

Associate professor Atílio Gameiro, M.Sc.EE, Ph.D.
Campus Universitário de Santiago, Aveiro, Portugal

Professor Mogens Blanke, M.Sc.EE, Ph.D.
DTU, Lyngby, Denmark

Associate professor Henrik Schiøler, M.Sc.EE, Ph.D.
AAU, Aalborg, Denmark (Chairman)

Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks

Publication No. ____________________

João Carlos Prazeres Figueiras, Ph.D.
Aalborg Universitet, 2007

Abstract:

Positioning of mobile devices in wireless communication networks is nowadays being intensively investigated due to the combined benefit of location information and communication. Typical solutions for such scenario rely on robust algorithms that estimate position from indirect measurements of the physical length of the communication links. Since these solutions do not require integration of additional hardware into the mobile nodes, they are cheap and simple to implement. As a price to pay, accuracy is typically lower in comparison to dedicated positioning systems. Thus, an important challenge to solve, that this dissertation aims at, is to enhance the accuracy of position estimation in wireless communication networks without the use of additional hardware. To reach this goal, the adopted approach is divided in three phases: (i) give a deeper insight on the problems by analyzing current solutions; (ii) propose solutions for enhancing accuracy of position information; and (iii) exploit the potential of such position strategies in cooperative schemes within wireless communications.

The dissertation starts by analyzing common localization and tracking solutions generally used in positioning systems. Since the measurement sampling is also important for tracking moving devices, the first analysis concerns the latency time for obtaining location information in short-range ad-hoc networks. Bluetooth is the elected technology since it is an actual example of such kind of networks. Simulations of the communication protocols defined in the specifications tell us that typical mechanisms present in current wireless communications, such as collision avoidance, can deeply influence the latency time. For accuracy analysis an experimental indoor setup was implemented, as a typical scenario where short-range communications are used in. The physical length of the communication links was estimated by using the measurements of signal strength permitted by the Bluetooth specifications. The final results lead us to the conclusion that due to the high noise measurements and its inherent low frequency of fresh measurements, tracking wireless devices is challenging problem. In order to get better results, we propose a solution which restricts the movement models of the mobile devices. This restriction, by relying on the assumption that users typically move in segment-wise linear trajectories, outperforms the standard solutions.

Additionally, this dissertation develops the idea of mobile cooperative positioning. Instead of localizing mobile devices in cellular networks in an individualist manner, the proposed solutions exploit the physical proximity of other mobile devices. By fostering the cooperation and subsequent communication among users, cooperative positioning strategies aim at localizing devices as a group and not as individuals. In order to reach this goal it is necessary to combine measurements from two domains: device-to-device links and cellular links. Since this combination of information is not a straightforward task, this dissertation proposes two different solutions for solving the problem. The main approach is to use a Bayesian filtering framework design in such a way that both types of measurements can contribute in a balanced manner to a single estimation of positions for each cooperative device. While in the first approach all the measurements are combined in a single estimation procedure, in the second approach, the short- and long-range are separately treated and combined in a subsequent step. The proposed solutions of combining additional information given by the device-to-device links with the cellular links, results in considerably higher performance than standard non-cooperative solutions.

Dedicated to my family & my friends

Acknowledgments

For their work and support, my advisor Associate Prof. Hans-Peter Schwefel at Aalborg University, Prof. Ali H. Sayed at University of California Los Angeles, PhD Adel Youssef and PhD Zhengrong Li at Google Inc, USA.

My direct work colleagues for their collaboration. For the sake of not forgetting anyone I do not list the names. For the discussion of ideas and market vision the folks from Blip Systems. For their support, the WANDA partners. My students, for their patience and whole the work enriching my personal knowledge.

All my friends from IEEE and the Aalborg University IEEE Student Branch. All the international friends I have met along the way in Denmark and USA. All my friends back in my home country, Portugal.

Thanks to my family for paving the road where I've been walking on. Finally, thanks to my Danish *family* for keeping me alive! ;)

João Carlos Prazeres Figueiras

Aalborg Universitet
November 2007

Contents

Abstract **iv**

Acknowledgments **vii**

List of Tables **xii**

List of Figures **xiii**

Acronyms . . . xx

I Introduction **xxiii**

Chapter 1 Introduction **1**

1.1 Problem Definition . . . 2

1.2 Problem Delimitation . . . 3

1.3 Scenario Description . . . 3

1.4 Contributions . . . 5

1.5 Thesis Outline . . . 6

II Background **9**

Chapter 2 Location Information in Wireless Networks **11**

2.1 Introduction . . . 11

2.2 The Localization Stack . . . 11

2.3 Obtaining Measurements to Extract Positioning Information . . . 13

2.3.1 Received Signal Strength . . . 13

2.3.2 Time of Arrival . . . 14

2.3.3 Time Difference of Arrival . . . 15

2.3.4 Angle of Arrival . . . 15

2.4 Radio-localization Techniques for Data Fusion . . . 16

2.4.1 Triangulation . . . 16

2.4.2 Hyperbolic Localization . . . 17

2.4.3 Angulation . . . 18
2.5 Positioning Techniques Based on User Proximity . . . 19
2.5.1 Physical Contact . . . 19
2.5.2 Cell Identification . . . 19
2.6 State-of-Art on Localization Solutions for Wireless Networks . . . 19

Chapter 3 Data Association for Maneuverable Target Tracking 21
3.1 The Bayesian Framework . . . 21
3.1.1 The Kalman Filter and Kalman Smoother . . . 23
3.1.2 The Extended and Unscented Kalman Filter for Nonlinear Estimation . . . 25
3.1.3 The Bootstrap Particle Filter for Nonlinear Non-Gaussian Estimation . . . 27
3.2 Maneuverable Target Tracking . . . 28
3.2.1 Maneuver Detection . . . 29
3.2.2 Multiple Model Solution for Handing Maneuvers . . . 29
3.3 Using Exponential Smoothing for Post-Processing . . . 31
3.3.1 Single Exponential Smoothing . . . 31
3.3.2 The Double Exponential Smoother . . . 32

III Analysis 33

Chapter 4 Latency of Location Information in Short-Range Networks 35
4.1 Introduction . . . 36
4.2 Measuring the Latency Time in Ad-Hoc Short-Range Communications . . . 37
4.2.1 Scenario Description . . . 37
4.2.2 Simulating the Inquiry Procedure . . . 38
4.2.3 Simulation Results of Inquiry Time . . . 39
4.3 Approximating the Inquiry Time . . . 41
4.3.1 Analytical approach of the Inquiry Procedure in Bluetooth . . . 43
4.3.2 A fast and simple approach for simulating the Inquiry Procedure in Bluetooth 45
4.4 Conclusions . . . 49

Chapter 5 Tracking of Wireless Devices in Infrastructure-Based Networks Using Received Power Measurements 51
5.1 Problem Definition . . . 52
5.2 Performance Metrics . . . 54
5.3 System Implementation for Performance Analysis . . . 56
5.3.1 Scenario Description . . . 56
5.3.2 Experimental Procedures . . . 56
5.4 The problem of Obtaining a Single Measurement at a Time . . . 57
5.5 Tracking Using the Extended Kalman Filter/Smoother . . . 60

5.5.1 Design of the filter . . . 60
5.5.2 Performance Analysis . . . 61
5.5.3 Process Noise Adaptation for Solving Unknown Initial State Problem . . . 69
5.5.4 Joint Estimation of Position and unknown Parameters . . . 70
5.6 Modeling Maneuvers Using Multiple-Models . . . 72
5.6.1 Design of the Filters used in the Switching Extended Kalman Filter (SEKF) Framework . . . 72
5.6.2 Switching Scheme for Filter Selection . . . 72
5.6.3 Performance of the Multi-Model Filters . . . 75
5.7 Particle Filters for Modeling the Measurement Noise Provoked by the Propagation Effects . . . 76
5.7.1 Modeling the Uncertainties in the Filter . . . 76
5.7.2 Initial Design and Testing of the Particle Filter . . . 78
5.7.3 Applying the Sequential Double Exponential Smoothing . . . 79
5.7.4 Performance Results . . . 81
5.8 Conclusions . . . 81

Chapter 6 Modeling Target Trajectory Based on a Segment-Wise Linear Approach 84
6.1 Using an Extended Kalman Filter with the Standard Constant Velocity Model . . . 86
6.1.1 Modeling and Extended Kalman Filter with Maneuver Detector . . . 86
6.1.2 Noise Adaptation . . . 87
6.2 An Alternative Approach: The Drifting Points Model . . . 87
6.2.1 The Algorithm . . . 87
6.2.2 Mathematical Formulation of the Model . . . 89
6.2.3 Modeling the Maneuver Detector . . . 90
6.2.4 The transformation of moments of the process . . . 90
6.3 Performance Analysis . . . 91
6.3.1 Simulation Framework for Performance Analysis . . . 91
6.3.2 Constant Velocity Model . . . 93
6.3.3 The Drifting Points Model . . . 97
6.4 Comparing the Drifting Points Model and the Constant Velocity Model . . . 100
6.5 Additional Batch Optimization for Off-line Processing . . . 102
6.5.1 Solution based on the Newton-Gauss Method . . . 102
6.6 Applying the Drifting Point Model in a Experimental Scenario . . . 104
6.7 Conclusions . . . 105

Chapter 7 Enhancing Localization Accuracy in Cellular Systems by Exploiting User-to-User ad-hoc links 107
7.1 Statistical Models for the Propagation in Short- and Long-Range Channels . . . 109

7.1.1 Statistical Models for measurements of Time of Arrival (ToA) and Angle of Arrival (AoA) . . . 109
7.1.2 Statistical Models for measurements of Received Signal Strength (RSS) . . . 111
7.2 Localization of Static Devices Using the Extended Kalman Filter (EKF) for Data Association . . . 112
7.2.1 Design of the Cooperative EKF . . . 112
7.2.2 Performance Analysis . . . 115
7.3 The Two-Level Data Fusion for Cluster Mobility and Individuals Mobility . . . 120
7.3.1 Decoupling and Coupling . . . 122
7.3.2 Relative Localization . . . 123
7.3.3 Absolute Localization . . . 124
7.3.4 Performance Analysis . . . 125
7.4 Conclusions . . . 127

IV Conclusions 129

Chapter 8 Summary and Conclusions 131
8.1 Conclusions . . . 131
8.2 Outlook . . . 133

V Appendixes 135

Appendix A Propagation Effects in the Wireless Channel 137

Appendix B Bluetooth survey for location purposes 139

Appendix C Analytical Approximation of the Inquiry Procedure in Bluetooth 146

Appendix D Additional evaluation of the EKF and EKS 150

Appendix E Localization in Wireless Networks Using Least Square Approach 153

Bibliography 157

Vita 163

List of Tables

5.1 Initial state used during the experiment. 57
5.2 Default parameters used in the design of the EKF and the Extended Kalman Smoother (EKS). Shaded row represents EKS specific parameters. 62
5.3 Overview of the problems and possible solutions of EKF and EKS algorithms when applied to positioning applications. 66
5.4 Parameters used in the design of the Particles Filter (PF). Shaded row presents the PF specific parameters. 78

6.1 Simulation parameters used for generating measurements of RSS. 93
6.2 Parameters used in the design of the EKF 93
6.3 Parameters used in the χ^2 maneuver detector. 93
6.4 Parameters used in the χ^2 maneuver detector when applied in the experimental setup. 104

7.1 Parameters used in the generation of measurements of AoA and ToA for localization in cellular communications. 111
7.2 Parameters used in the simulation of the ad-hoc cellular system 115

B.1 Set of frequencies used in the Inquiry procedure: sorted in ascendant manner and in the way they appear in the sequence 144

List of Figures

1.1 General scenario description: Users enabled with one short-range and one long-range technologies are able to communicate among them or/and with an infrastructure of fixed nodes. The infrastructure is assumed to be of either technology. 4

2.1 The location stack: An abstract layered framework envisaging seamless and ubiquitous localization systems. The shaded layers are the focus of the present dissertation. 12
2.2 Triangulation Technique 16
2.3 Hyperbolic Localization Technique 17
2.4 Angulation Technique 18
2.5 CellID Technique 19

3.1 Bayesian filtering cycle: cyclic sequence of prediction and correction. 22
3.2 Example of execution of the multiple model framework. Two filters run in parallel during two units of time. The result is 4 histories. 30

4.1 Description of the scenario used for studying the latency time of localization in short-range networks. 37
4.2 Inquiry Time: a) Single AP scenario; b) Multi AP scenario. 38
4.3 *Inquiry Time* PDF for a *Single AP scenario* with Bluetooth specifications V2.0 . . . 39
4.4 *Inquiry Time* PDF for a *Single AP scenario* with Bluetooth specifications V1.1 [PBK04]. 39
4.5 Inquiry Time PDF to a *Multiple Access Point Scenario* ($n = 1, 2, 4, 7$) 40
4.6 Inquiry Time CDF to a *Multiple Access Point Scenario* ($n = 1, 2, 4, 7$) 40
4.7 Simulated Inquiry time Cumulative Distribution Function (CDF) to *Multiple Access Point Scenario* (n=1 up to 20) and 1 Wireless Device (WD) (obtained simulating 10^6 values of inquiry time for each settings) 41
4.8 Inquiry Time duration depending on the number of Access Point (AP)s for different percentiles 41
4.9 Simulated and approximated Inquiry Time CDF to a *2, 4, 7 AP scenario* (obtained simulating 10^6 values of inquiry time) 42

4.10 Probability Density Function (PDF) of the inquiry time for a scenario with 2 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. 45
4.11 CDF of the inquiry time for a scenario with 2 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. . . 45
4.12 Execution of the ball-urn game that approximates the inquiry time. 46
4.13 CDF of the inquiry times from the Bluetooth procedure (InqSim - dashed line) in comparison to the abstracted model (InqSimpl - solid line) for $N = 1, 2, 5, 10, 15, 20, 30, 40$ and 50 APs. 48
4.14 Average time to generate one samples of the inquiry time. The results were obtained out of 1000 runs of each case and each number of nodes. 49

5.1 Example of a wireless short-range technology used for estimating user location information. 51
5.2 Example of a segment-wise trajectory described by the WD 53
5.3 Setup of the Bluetooth network and layout of the indoor scenario used to run the localization experiment. 56
5.4 Straight trajectory defined for running the experiments. 57
5.5 Squared trajectory defined for running the experiments. 57
5.6 Estimation of the target path using the EKF, when the target follows the straight trajectory. 62
5.7 Estimation of the target path using the EKF, when the target follows the squared trajectory. 62
5.8 Estimation of the target velocity using the EKF, when the target follows the squared trajectory. 62
5.9 Estimation of the target velocity using the EKF, when the target follows the squared trajectory. 62
5.10 Estimation of the target path using the EKS, when the target follows the straight trajectory. 63
5.11 Estimation of the target path using the EKS, when the target follows the squared trajectory. 63
5.12 Estimation of the target velocity using the EKS, when the target follows the straight trajectory. 63
5.13 Estimation of the target velocity using the EKS, when the target follows the squared trajectory. 63
5.14 Performance of the estimation in the straight trajectory when using the EKF and the EKS. Performance according to the Pointwise Mean Distance Error (PMDE) and the Interpolated Pointwise Mean Distance Error (IPMDE) metrics 64
5.15 Performance of the estimation in the squared trajectory when using the EKF and the EKS. Performance according to the PMDE and the IPMDE metrics 64

5.16 Performance of the estimation in the straight trajectory when using the EKF and the EKS. Performance according to the Covered Distance Error (CDE) metric. . . . 65
5.17 Performance of the estimation in the squared trajectory when using the EKF and the EKS. Performance according to the CDE metric. 65
5.18 Performance of the EKF for several values of initial state for the straight trajectory. 65
5.19 Performance of the EKF for several values of initial state for the squared trajectory 65
5.20 PMDE performance of the EKS in order to the α parameter for the straight trajectory. 67
5.21 PMDE performance of the EKS in order to the α parameter for the squared trajectory. 67
5.22 Example of path estimation for the straight trajectory when $\alpha = -42dBm$. 67
5.23 Example of path estimation for the squared trajectory when $\alpha = -42dBm$. 67
5.24 Performance, according to the PMDE, of the EKS with respect to the variance of the acceleration for the straight trajectory. 68
5.25 Performance, according to the PMDE, of the EKS with respect to the variance of the acceleration for the squared trajectory. 68
5.26 Example of the estimated straight trajectory when $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ are both equal to $5 \times 10^{-3} m/s^2$. 68
5.27 Example of the estimated squared trajectory when $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ are both zero. 68
5.28 Performance of the EKF for several values of initial state for the straight trajectory when it is used process noise adaptation. 70
5.29 Performance of the EKF for several values of initial state for the squared trajectory when it is used process noise adaptation. 70
5.30 Joint parameter and state estimation of an EKS for the straight trajectory 71
5.31 Joint parameter and state estimation of an EKS for the squared trajectory 71
5.32 Single run of the SEKF in the squared trajectory using a switching scheme based on the error covariance matrix. 73
5.33 Model selection performed by a switching scheme based on the error covariance matrix. 73
5.34 Single run of the SEKF in the squared trajectory using a switching scheme based on the innovation process. 75
5.35 Model selection performed by the switching scheme using the innovation process. . . 75
5.36 PMDE and IPMDE performance of the SEKF for straight trajectory 75
5.37 PMDE and IPMDE performance of the SEKF for squared trajectory 75
5.38 Probability Density Function for the measurement of the received power for the parameters $\sigma_p = 4dBm$ 78
5.39 Two independent runs of trajectory estimation with $\sigma_{\ddot{x}} = 0.001 m/s^2$ for the PF. The only difference between the two setups is the seed used in the generation os particles in the PF. 79
5.40 Two independent runs of trajectory estimation with $\sigma_{\ddot{x}} = 0.01 m/s^2$ (i.e. 10 times higher than in Fig. 5.39) for the PF. The only difference between the two setups is the seed used in the generation os particles in the PF. 80

5.41 PF coupled with a Double Exponential Smoothing (DES) time series. 80
5.42 PMDE for the PF coupled with a DES when estimating the straight trajectory. . . . 81
5.43 PMDE for the PF coupled with a DES when estimating the squared trajectory. . . . 81
5.44 CDE for the PF coupled with a DES when estimating the straight trajectory. 82
5.45 CDE for the PF coupled with a DES when estimating the squared trajectory. 82

6.1 Example of a noisy estimation of the user trajectory: the black dashed line represents a true trajectory and the red solid line represents an example of the estimated trajectory. 85
6.2 Constant Velocity Algorithm. The solid lines correspond to flow of data, while the dashed lines correspond to flow of control signals. 86
6.3 Drifting Points Algorithm 88
6.4 Example of a run of the Drifting Points model. 88
6.5 Illustration of the DP model in operation. 91
6.6 Straight trajectory used in the simulations. 92
6.7 Cross trajectory used in the simulations. 92
6.8 Staircase trajectory used in the simulations. 92
6.9 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 94
6.10 Maneuver detector for $f_s = 15Hz$. The signal represents the inverse of the χ^2 PDF as given by eq.(3.53). 94
6.11 Empirical CDF of the PMDE metric for the straight trajectory. 94
6.12 Empirical PDF of the number of detected maneuvers in the estimation of the straight trajectory. 94
6.13 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 95
6.14 Maneuver detector for $f_s = 15Hz$. 95
6.15 Empirical CDF of the PMDE metric for the cross trajectory. 96
6.16 Empirical PDF of the number of detected maneuvers in the estimation of the cross trajectory. 96
6.17 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 96
6.18 Maneuver detector for $f_s = 15Hz$. 96
6.19 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 100Hz$. 97
6.20 Maneuver detector for $f_s = 100Hz$. 97
6.21 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 98
6.22 Maneuver detector for $f_s = 15Hz$. 98
6.23 Empirical CDF of the PMDE metric for the straight trajectory. 98

6.24 Empirical PDF of the number of detected maneuvers in the estimation of the straight trajectory. 98
6.25 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 99
6.26 Maneuver detector for $f_s = 15Hz$. 99
6.27 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$. 99
6.28 Maneuver detector for $f_s = 15Hz$. 99
6.29 Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 100Hz$. 100
6.30 Maneuver detector for $f_s = 100Hz$. 100
6.31 Gain vs. maneuver threshold for straight trajectory 101
6.32 Gain vs. maneuver threshold for cross trajectory 101
6.33 Gain vs. maneuver threshold for staircase trajectory 101
6.34 Example of the Drifting Point (DP) model with additional minimization for measurement frequency of $f_s = 15Hz$. Straight trajectory. 103
6.35 Example of the DP model with additional minimization for measurement frequency of $f_s = 15Hz$. Cross trajectory. 103
6.36 Empirical CDF of the PMDE metric for the straight trajectory after minimization. . 103
6.37 Empirical CDF of the PMDE metric for the cross trajectory after minimization. . . 103
6.38 Comparison in a experimental setup between the Constant Velocity (CV) and the DP model. 104
6.39 Performance comparison in a experimental setup between the CV and the DP model, in terms of the PMDE and Scaled Root Mean Square Error (SRMSE) metrics. . . . 105
6.40 Performance comparison in a experimental setup between the CV and the DP model, in terms of the CDE metric. 105

7.1 Mobile Cooperative Localization: Users in the close proximity able of communicating through short-range links cooperate in order to enhance the location accuracy of the overall cooperative users. 108
7.2 LOS probability vs. distance between transmitter and receiver. 110
7.3 Localization results for the Hybrid TOA/AoA Positioning (HTAP) for 3 Base Station (BS)s 116
7.4 Localization results for the Hybrid Line of Position (HLOP) for 3 BSs 116
7.5 Localization results for the Hybrid Time Difference of Arrival (HTDOA) for 3 BSs . 116
7.6 Performance for the HTAP when σ is known 117
7.7 Performance for the HTAP when σ is constant. 117
7.8 Performance for the HLOP when σ is known 118
7.9 Performance for the HLOP when σ is constat. 118
7.10 Performance for the HTDOA when σ is known 118

7.11 Performance for the HTDOA when σ is constant. 118
7.12 Gain of cooperation for the 67% percentile with σ known 119
7.13 Gain of cooperation for the 67% percentile with σ unknown 119
7.14 Gain of cooperation for the 95% percentile with σ known 119
7.15 Gain of cooperation for the 95% percentile with σ unknown 119
7.16 Schematic representation of the proposed algorithm. Note that the arrows represent quantities to be estimated. 120
7.17 Operational representation of the proposed algorithm. 121
7.18 Empirical CDF of the Root Mean Square Error (RMSE) metric for the HTAP localization technique, when the proposed Two Level Kalman Filter (2LKF) is used for data fusion. 126
7.19 Empirical CDF of the RMSE metric for the HLOP localization technique, when the proposed 2LKF is used for data fusion. 126
7.20 Empirical CDF of the RMSE metric for the HTDOA localization technique, when the proposed 2LKF is used for data fusion. 126
7.21 Gain of the 2LKF cooperative algorithm with respect to the non-cooperative method for 67% of all the tested cases 127
7.22 Gain of the 2LKF cooperative algorithm with respect to the non-cooperative method for 95% of all the tested cases 127

B.1 Bluetooth Scatternet 140
B.2 Inquiry Procedure in a macro point of view 141
B.3 The Inquiry Substate in detail 142
B.4 Frequency shifting that occurs in the master node 145

C.1 PDF of the inquiry time for a scenario with 1 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. . . 147
C.2 CDF of the inquiry time for a scenario with 1 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. . . 147
C.3 PDF of the inquiry time for a scenario with 3 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. . . 149
C.4 CDF of the inquiry time for a scenario with 3 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications. . . 149

D.1 PMDE performance of the EKS with respect to the β parameter for the straight trajectory. 150
D.2 PMDE performance of the EKS with respect to the β parameter for the squared trajectory. 150
D.3 The influence of the lag in the PMDE performance of the EKS. Results for the 1st run of the straight trajectory. 151

D.4 The influence of the lag in the PMDE performance of the EKS. Results for the 1st run of the squared trajectory. 151
D.5 Performance of the EKS when the velocity is emulated by a factor of 10 in the straight trajectory. The velocity is $0.23m/s$ and $\sigma_{\ddot{x}} = 2.5m/s^2$. 152
D.6 Performance of the EKS when the velocity is emulated by a factor of 10 in the squared trajectory. The velocity is $0.44m/s$ and $\sigma_{\ddot{x}} = 2.5m/s^2$. 152
D.7 Performance of the EKS when the velocity is emulated by a factor of 30 in the straight trajectory. The velocity is $0.70m/s$ and $\sigma_{\ddot{x}} = 50m/s^2$. 152
D.8 Performance of the EKS when the velocity is emulated by a factor of 10 in the squared trajectory. The velocity is $1.35m/s$ and $\sigma_{\ddot{x}} = 50m/s^2$. 152

E.1 Single run of a Least Squares (LS) solution when $\sigma_Z = 1dB$. 154
E.2 Single run of a LS solution when $\sigma_Z = 6dB$. 154
E.3 Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 1dB$. . . 155
E.4 Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 6dB$. . . 155
E.5 Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 1dB$. . . 156
E.6 Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 6dB$. . . 156

Acronyms

2D	2-Dimensional
2LKF	Two Level Kalman Filter
3D	3-Dimensional
ACL	Asynchronous Connection Oriented
AoA	Angle of Arrival
AP	Access Point
BER	Bit Error Rate
BS	Base Station
CDE	Covered Distance Error
CDF	Cumulative Distribution Function
CV	Constant Velocity
DES	Double Exponential Smoothing
DIAC	Dedicated Inquiry Access Code
DP	Drifting Point
EKF	Extended Kalman Filter
EKS	Extended Kalman Smoother
FCC	Federal Communications Commission
FHS	Frequency Hop Synchronization
FHSS	Frequency Hopping Spread spectrum
GFSK	Gaussian Frequency Shift Keying

GIAC General Inquiry Access Code

GPB Generalized Pseudo-Bayesian

GPS Global Positioning System

GSM Global System for Mobile communications

HLOP Hybrid Line of Position

HTAP Hybrid TOA/AoA Positioning

HTDOA Hybrid Time Difference of Arrival

ID IDentity

IPMDE Interpolated Pointwise Mean Distance Error

ISM Industrial, Scientific and Medical

LAP Lower Part Address

LLS Linear Least Squares

LoS Line of Sight

LS Least Squares

NLoS Non Line of Sight

OLS Ordinary Least Square

OSI Open Systems Interconnection

PDF Probability Density Function

PF Particles Filter

PMDE Pointwise Mean Distance Error

RMSE Root Mean Square Error

RSS Received Signal Strength

RSSI Received Signal Strength Indicator

SCO Synchronous Connection Oriented

SEKF Switching Extended Kalman Filter

SIG Special Interest Group

SNR Signal to Noise Ration

SRMSE Scaled Root Mean Square Error

TDOA Time Difference Of Arrival

ToA Time of Arrival

UKF Unscented Kalman Filter

UMTS Universal Mobile Telecommunications System

UWB Ultra Wideband

WD Wireless Device

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

Part I

Introduction

Chapter 1

Introduction

Location information is one of the research topics that has been receiving a continuously increasing attention in both academia and industry. Primarily considered as a vital information for vehicle tracking and military strategies, it has nowadays been quickly introduced in common wireless communication networks. In contrast to the former dedicated solutions that were designed to simply provide position, the new solutions for wireless networks are able to provide a combined benefit of both communication and positioning. As a consequence, all the network as well as the end-user and the service providers can profit with the position enabled communication capabilities. While the network operator is able to more efficiently manage the resources of the network, the service provider is able to offer location based services and the user is allowed to enjoy of those personalized location dependent services. In fact, it is already found in research documents that location information has been used as a basic requirement for deployment of new protocols (e.g. routing, clustering) [AKK04], technologies (e.g. cooperative systems) [Soc03] and applications (e.g. navigation, location aware advertising) [Küp05]. From the industry point of view, the use of location information has been mainly stimulated by applications such as navigation [Nav], location dependent search [LS] and social networks [Dod]. Furthermore, since wireless communication networks are nowadays always present anywhere at any time, every location dependent networking enhancements, service or application can be rapidly spread and globally used.

Location information is most of the times possible to obtain by using a positioning system or a positioning solution. A positioning system means, in this dissertation, the hardware and software necessary to measure the indirect readings of user position and process them in order to obtain an estimation of the user position itself. In contrast, a positioning solution is typically a software based implementation, where the indirect measurements of the user position are obtained by a second party technology or a closed piece of hardware. Basically the difference is that in a positioning system, both hardware and software are open for design; while in a positioning solution, all the design must be in the software side, since the hardware is a closed entity that does not permit redesigning.

The possible scenarios and conditions where location information is needed are numerous and no actual localization system is able to perform localization under any condition. For this reason, several localization systems and solutions are available and many others are still under research. Possibly the most famous localization system is the satellite-based Global Positioning System (GPS) [K$^+$96]. Although the GPS technology represents a solution with worldwide coverage, it is not usable under determined situation, such as dense urban places, indoors, underwater and underground scenarios. For these cases, many other systems are being developed or even already available on the market [HB01b].

Enabling location information in current wireless communication networks can be done by either integrating a positioning system in the network or by implementing a positioning solution that extracts location information by exploiting the potential of the networks. By integrating a positioning system in the network it is typically possible to obtain better accuracy than what is obtained by a direct implementation in the network. However, integrating additional hardware represents a great disadvantage due to the fact that it necessarily implies additional costs, power consumption and complexity. For this reason, a direct software based positioning implementation, that opportunistically exploits the existent hardware and extracts location information is commonly preferred. Typical positioning solutions make use of mechanisms present in most of the current advanced wireless technologies such as power control or synchronization schemes. The requirement is that the mechanism makes measurements of the wireless channel that can be used as indirect observations of the user position. Examples of such measurements are the received signal strength or the propagation delay of the signals.

1.1 Problem Definition

Implementing positioning solutions in wireless networks is a challenging task due mainly to two major factors: the behavior of the communication channels, and the typically unpredictable movement behavior of the mobile users.

Propagation Conditions

The problem relative to the communication channel is that the signal propagation conditions are dependent on the scenario and often have a dependency with time. The factors influencing the propagation conditions are numerous. Some examples of these factors are the design of the antennas and its subsequent integration in the end-user equipment, orientation of those antennas, scenario partitions such as walls and furniture. In particular for handsets, a great problem is on the position of the hand of the user while holding the device. All these factors result in a wireless channel strongly influenced by undesirable destructive propagation effects such as signal blocking, multipath, dispersive attenuations and shadowing. These effects harden the tasks of measuring the wireless channel and also modeling it. Furthermore, since in positioning solutions (which are typically the strategies for obtaining location information in wireless networks), there is no access to the internal design of the hardware, the problem of modeling the errors in the measurements gets even more complex. For this reason it is not straightforward to design an accurate model for the communication channel envisaging its usage in localization algorithms.

User Mobility

Another problem with positioning solutions for wireless networks is the mobility of the user. The user is typically characterized by high mobility and highly unpredictable behavior. This fact makes common solutions for stochastic filtering more difficult to implement, because the models required by those frameworks are, itself, difficult to obtain, mainly due to the constantly changing dynamics of the user.

1.2 Problem Delimitation

In the present dissertation, the propagation models are assumed rather simple. The work will mainly focuses on the mobility models and on the algorithmic frameworks for estimating location information. The reason for such choice is that modeling propagation conditions is itself a big field of research that has an extensive literature. Furthermore, in current wireless communications, the end-user has only access to readings of the propagation channel having minimal or no control on how those readings are obtained. Thus, instead of only look at propagation of wireless signals, the choice was to design a whole localization solution upon actual communication networks and evaluate its accuracy. Existent models and algorithms are tested and compared. Two major pieces of the work concern on one had tracking wireless users in indoor short-range communications and, on the other hand, cooperative cellular localization via additional ad-hoc short-range links.

Out of the scope of this dissertation are problems such as management, power consumption and privacy. Regarding the propagation conditions only a short overview is made in order to obtain a simple model for integrating in the localization solutions. Although in Section 5.7 the propagation conditions are more carefully studied, the entire dissertation considers a channel corrupted by additive white gaussian noise. Also there is no specific attention given to multi-platform solutions. Only in Section 5.5.4 this problem is indirectly addresses by the joint parameter estimation. Regarding multi-technology, it is only addresses in the context of cooperative localization as it is done in Chapter 7. Problems regarding the integration of those technologies is not addressed.

1.3 Scenario Description

This dissertation assumes throughout its chapters the general scenario of Fig. 1.1 which is narrowed down according to each specific chapter. In a environment where 2 technologies coexist, one short-range and one long-range technology, localization is performed in 2 dimensional using the channel reading permitted by each technology. Thus, it is assumed that measurements are obtained by the technology specific mechanisms and no control is possible on how the wireless channel is measured. Let assume N^{wd} WDs with mobility behavior, i.e. its physical position changes along time. The WDs are equipped with two different technologies, a short-range technology such as Wireless Local Area Network (WLAN) 802.11 or Bluetooth, and a long range technology such as Global System for Mobile communications (GSM) or Universal Mobile Telecommunications System (UMTS). The

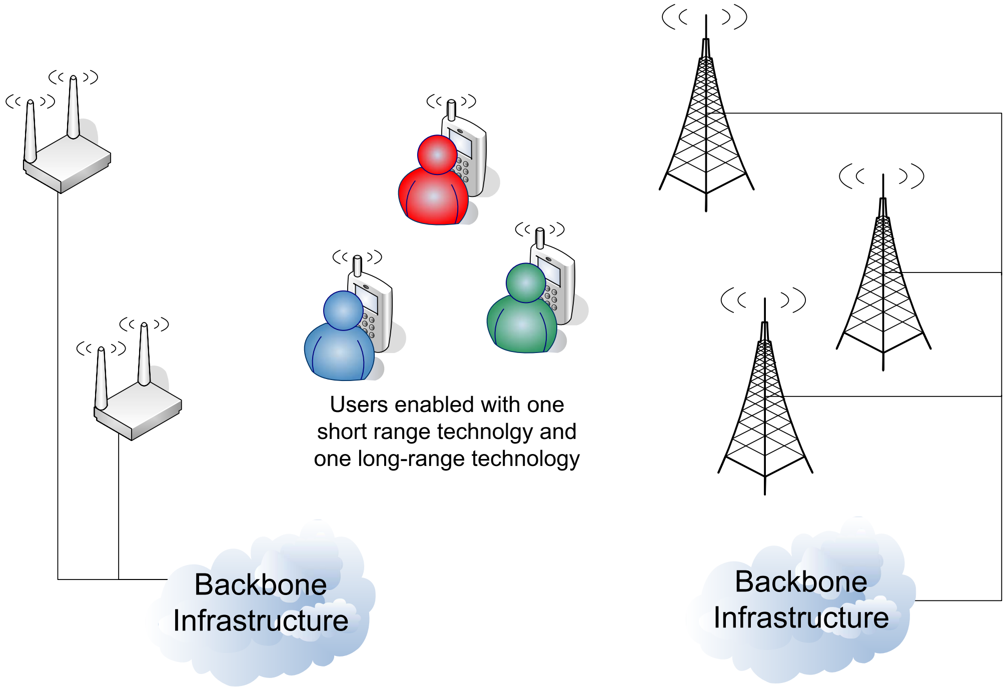


Figure 1.1: General scenario description: Users enabled with one short-range and one long-range technologies are able to communicate among them or/and with an infrastructure of fixed nodes. The infrastructure is assumed to be of either technology.

short-range technology can consider two types of communication: ad-hoc peer-to-peer links, i.e. links established among WDs with a time-life enough for transferring communication data; (ii) infrastructure-based links, which allow the WDs to communicate with a backbone infrastructure by accessing to APs at fixed positions. Regarding the long-range technology, it uniquely permits infrastructure-based links. In the context of long-range communications, the same concept of AP nodes is instead named BS nodes. The short-range and the long-range technology are assumed to be respectively composed by N^{ap} and N^{bs} fixed nodes of access to the backbone network.

Estimating location information in the scenario of Fig. 1.1 is possible by using algorithms that associate, in a single estimator, indirect measurements of the physical length of the several communication channels. While the short-range technology commonly uses observations of signal strength for estimating the link length, the long-range technology uses observations of propagation delays or angles of arrival. The reason is that while for short range links, the readings that carry more accurate information of the communication channel are the signal strength measurements, in long-range communications are the measurements of signal propagation delays. The angle readings are commonly used as a rough indication of the position since the errors have higher magnitude than what happens to the propagation delays.

Along this dissertation, the scenario of Fig. 1.1 is narrowed down depending on the issues address by each chapter. Mainly two major scenarios derived from Fig. 1.1 are considered: infrastructure-based communication using the short-range technology; and combined infrastructure-based long-range communication and short-range ad-hoc communication. The first scenario, evaluated in an indoor experimental setup allows the existent algorithms to be tested and the problems to be identified. The indoor scenario is used as a typical case of where short-range communications operate on. Concerning the second scenario, a computer tool simulates a cellular network where users cooperate envisaging an enhancement on their location estimation. The propagation conditions of the long range links are simulated according to stochastic models for the signal propagation conditions proposed in the literature. The users are assumed in line of sight, i.e. no obstructions in the propagation of the short-range links, where the models assume simple gaussian channels.

1.4 Contributions

The contributions to the state-of-art in positioning solutions for wireless networks are:

Latency of positioning solutions The latency of obtaining location information in wireless networks is studied. The analysis give an understanding on the statistical behavior of that time in current ad-hoc communications. The Bluetooth technology was used as a target technology to represent a use case of a typical ad-hoc technology that implements randomized link layer protocols in the establishment of communication among nodes. The latency time in Bluetooth networks was evaluated and an algorithm for generating samples of that time is proposed. The results were published in [FSK05, FS06].

Evaluation of existent stochastic filtering techniques Existent solutions for stochastically filter noisy measurements in order to produce an estimator of position were studied. Algorithms such as Kalman filters, and Particle filters were studied. Advanced techniques for estimating jumping processes were studied. The problems were identified and some solutions were proposed. Preliminary results were introduced in [FSK05, MMF$^+$05]. Concerning the comparison among the several algorithms, results will be soon published.

Drifting Points model A segment-wise linear model for the user dynamics is proposed. It assumes that the user moves according to piecewise linear trajectories, where the size of each segment is unknown and subsequently estimated. The model is evaluated while assuming that the user moves at constant velocity along the entire trajectory, but can be easily extended to consider other types of piecewise linear models. The results are being compiled in one or two articles to submit for publication.

Cooperative localization schemes Two algorithms for cooperative localization are proposed. The users are enabled with two technologies, a long-range cellular technology and a short-range ad-hoc technology. Each user measures its long-range wireless channel and the cooperative ad-hoc links with other users. The information from all the users is combined in a single framework for estimating position. The first proposed algorithm applies an extended Kalman filter for combining all the measurements. The second algorithm, the 2 Level Kalman Filter, by decoupling the several sources of measurements based on their types, is more scalable and modularizes the sources of information for independent execution. The extended Kalman filter was published in [FF07, LFFS06] and further used in [MRW$^+$07, RWM$^+$07]. The 2 Level Kalman Filter will be published later.

1.5 Thesis Outline

This dissertation is divided in 4 main parts: the present introduction, a background part; proposal and analysis of positioning mechanisms; and final conclusion. Each chapter is here summarized:

Background Chapters

Chapter 2 starts by explaining a general framework of positioning systems. Then, it describes the common basic methods for obtaining location information and mentions the common sources of location information in wireless communications.

Chapter 3 explains in detail the mechanisms of estimating position and it gives an overview on the available solutions for estimating position of WDs. The biggest emphasis is given to Bayesian filtering due to the fact that is an adequate solution to be used in such scenarios where noisy measurements are used to estimate a sequence of states of an invisible process, i.e. the user position.

Analysis Chapters

Chapter 4 analyzes the timing aspects of obtaining location information in wireless networks. The main assumption is that location is possible as long as the link layer protocol establishes a connection. Current short-range communication networks tend to operate in a ad-hoc fashion, where the links between the several nodes are commonly established according to randomized protocols. This subsequently means that the latency of positioning solutions in such networks is not of simple evaluation. For this reason the link layer protocols are evaluated in a use case technology, Bluetooth, which implements a connection establishment mechanism of a so called ad-hoc network.

Chapter 5 evaluates several of the available solutions for tracking WDs in short-range communications. The scenario is an indoor experimental setup that represents a typical scenario where current short-range communications nowadays operate on. The analyzed solutions range from a single filter approach to a multi model filtering approach. The main purpose of this chapter is to evaluate the accuracy of each one of the positioning algorithms, identify their problems and propose solutions for some for those problems. The entire chapter is based on Bayesian filtering implementations since these algorithms are adequate for estimating position information based on noisy measurements of the communication channel.

Chapter 6 proposes an alternative model for estimating the trajectory described by a WD within the coverage range of a wireless network. The solution relies on a segment wise linear model, where sporadic changes on the movement patterns are considered as discontinuities on the estimation algorithm and thus estimated using a jump detector.

Chapter 7 applies the concept of cooperative localization used in Wireless Sensor networks on the heterogeneous multi-technology scenario. The WDs, enabled with two different technologies are enforced to cooperate among themselves by using their short-range links in order to enhance their localization accuracy in cellular networks. The evaluation is based on a computer tool that simulates the measurements obtained from both the cellular links and the ad-hoc links as stochastic samples generated from models present in the literature.

Conclusion Chapter

Chapter 8 concludes the thesis and gives some guidelines for future work.

Part II

Background

Chapter 2

Location Information in Wireless Networks

This background chapter presents the basic methods and concepts behind common radio-localization solutions and techniques. An initial overview of a general localization system is introduced in a conceptual layered approach. Then, it is explain the common measurements and subsequent basic techniques that are used in current solutions for localizing WDs. Additional positioning solutions, out of the scope of this dissertation, can be found in survey presented in [HB01a].

2.1 Introduction

Location information in wireless networks is commonly obtained by observing the properties of the wireless channel. The observations where localization solutions rely on are typically obtained by measuring the propagation conditions in the wireless link. Measurements to be used can be for instance RSS, ToA, AoA, Bit Error Rate (BER) or Signal to Noise Ration (SNR). These observations, are required to have a relation with the distance or angle between transmitters and receivers. This way, using the measurements of the propagation channel it is possible to estimate distances and/or angles and subsequently estimate, based on certain algorithms further explained in Chapter 3, a single position for the mobile device.

2.2 The Localization Stack

Similarly to the Open Systems Interconnection (OSI) framework used in telecommunication systems, the Location Stack is a conceptual framework of well defined layers [HBB02, Hig04]. The interest of such type of design is that it allows localization solutions to be modular, more dynamic, cooperative and above all, ubiquitous. As it is shown in Fig. 2.1, the stack is divided in 7 layers, each of which with clearly defined properties.

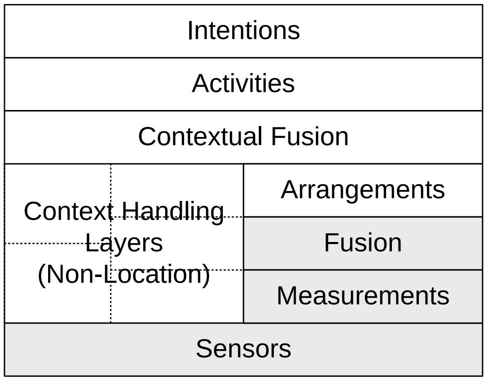


Figure 2.1: The location stack: An abstract layered framework envisaging seamless and ubiquitous localization systems. The shaded layers are the focus of the present dissertation.

Sensors: This layers consists on the necessary hardware and software to obtain measurements from the wireless channel. Output data can be for example measurements of RSS.

Measurements: The measurements layer concerns the models to translate sensor observations into distance, angle or position itself, and the inherent uncertainty representations of the sensors responsible for the raw data. The estimated distance between transmitter and receiver is an example of the output of this layer. Section 2.3 explains some of the models in more detail.

Fusion: This layers is responsible for gathering the flow of information obtained from the measurement layer and produce an estimator for the position of the WD. This layer is also responsible for handling possible redundancies in order to produce more accurate estimators of position. Triangulation, presented in Section 2.4.1, is an example of such fusion layer.

Arrangements: This layer handles the dependencies and relationships that the several WDs have among themselves. Additionally this layer must be able to easily interchange between local and global coordinate systems. Grouping users in proximity clusters can be considered as a technique belonging to this layer.

Contextual Fusion: This layer is responsible for mapping the positioning information into a non-position contextual information. Example can be the application that manages a “todo list” depending on the position of the WD.

Activities: The gathering of all positioning and non-positioning information is done in this layer. The purpose is to learn activities of the WD as a representation of the state of the whole amount of information. An example can be an application that looks at the todo list” and user position and understands that the user is for instance running out of time in order to accomplish all the tasks in that todo list”.

Intentions: This layer relates the cognitive desires of the users and the expectable behavior of a ubiquitous localization system. An example can be the desire of the user to accomplish all the tasks in his todolist" by using a determined trajectory. Given this desire, the system adapts itself to comply with it.

Context Handling Layers: The handling of non-positioning information is done in this layer. In a vertical approach, the information is exchanged by several os the location stack layers.

This dissertation exclusively focuses on the sensors, measurements and fusion layers. The generic concepts related to the measurements and fusion layers are explained respectively in Section 2.3 and Section 2.4. Then, in Chapter 3, advanced algorithms which combine both layers are surveyed. Concerning the sensors layer, only latency time for obtained location information is studied, what is done is Chapter 4. In the rest of the dissertation, i.e. in Chapter 5, Chapter 6 and Chapter 7, both the measurements and the fusion layers are jointly studied in frameworks that integrate both layers.

2.3 Obtaining Measurements to Extract Positioning Information

This section presents some of the models of the measurements layer defined in the location stack. Since it is not the aim of the present work to directly study how the measurements are obtained (i.e. the physical layer aspects relative of obtaining those measurements), the types of measurements will be presented as simple models. These models approximately represent the relation between those measurements and user position.

2.3.1 Received Signal Strength

The RSS is a measure of the magnitude of the power of the signal that the receiver sees in its terminals. Due to the destructive propagation effects (presented in Appendix A), this measure is typically subject to a noise component with variations of tens of dB for the typical scenarios of interest, i.e. indoor short-range localization or outdoor long-range localization.

The RSS can be modeled by a constant component and a variable component. Generally, the constant component is modeled by the path loss propagation model and the variable component is modeled by a range of different complex propagation effects. The propagation effects can be for instance signal attenuation, shadowing, multipath, scattering and diffraction. For an overview of these propagation effects see Appendix A and for a deeper insight see for example [Par00, Gol05]. For modeling the path loss propagation effect, it is considered the following equation:

$$p = \alpha - 10\beta \log(d) \tag{2.1}$$

where p is the received signal strength, α a parameter dependent on the transmission power, additional losses and other system component dependent constants, β is the path loss exponent, and d the physical length of the communication link, i.e. distance between transmitter and receiver. Regarding the variable component, it is complex to obtain a model that takes into account the stochastic behavior of the involved propagation effects. The reason is that those effects strongly depend on the scenario. Thus, the most generic approach is to consider all those effects as a single additive Gaussian distributed noise component with a standard deviation of σ_v.

$$p = \alpha - 10\beta \log(d) + \mathbf{v} \quad \mathbf{v} \sim \mathrm{Norm}(\sigma_v) \tag{2.2}$$

Although this noise component is generally considered Gaussian throughout the present work, in Section 5.7 a more detailed model is considered. The purpose is to compare the impact of the modeling of this component in the localization accuracy.

2.3.2 Time of Arrival

The ToA is the timestamp that the receiver sees on its internal clock when the signal is received at its terminals. To this type of measurements, one of the factors that most influences the error corrupting ToA observations is the granularity of the internal clock. Given that the clock increments only in the instant of a clock tick, whatever is the instant in between two consecutive clock ticks, when the signal triggers the terminal of the receiver, it will see the same timestamp. Thus, for every instant within this period, the estimated link length will be the same. In terms of distance this error can be easily calculated as the quotient of the speed of light by the clock granularity. When this error has the same magnitude as the typical length of the communication channel, it gets unfeasible to estimate the link length based on ToA measurements.

Although the fact that the ToA is strongly dependent on the technology, it is common to consider it, in general, as modeled by the sum of a constant component and a variable noise component.

The constant component t_a of the ToA model is equivalent to the sum of the transmission time t_d (timestamp in the transmitter clock when the signal is transmitted) and the propagation delay (time that the signal needs to travel from the transmitter until the receiver).

$$t_a = t_d + \frac{d}{c} \tag{2.3}$$

In eq.(2.3), c is the velocity in the medium, commonly the speed of light, and d the physical length of the communication link. Note that eq.(2.3) requires the transmitter and receiver to be clock synchronized, otherwise the transmitted timestamp would not have the same reference as the received timestamp. This can represent a strong disadvantage of the ToA observations because clock synchronizing a whole network is commonly complex to implement and expensive to maintain.

Regarding the variable component, it is for simplicity assumed Gaussian distributed. In Chapter 7, this component is studied in deeper detail.

2.3.3 Time Difference of Arrival

The Time Difference Of Arrival (TDOA) is the difference of two ToA measurements obtained from two different copies of the same signal. This technique was proposed as a solution for dropping the requirement that ToA measurements impose on clock synchronization between transmitter and receiver. Since TDOA determines the difference between two copies of the same signal, the dependency shown in eq.(2.3) with the time of transmission of the signal is lost. Thus, assuming that at time t_d two copies of the same signal are sent from BS1 and BS2. Given that the WD is d_1 meters from BS1 and d_2 meters from BS2, the TDOA T_a seen at the WD is given by:

$$T_a = t_d + \frac{d_1}{c} - \left(t_d + \frac{d_2}{c}\right) = \frac{d_1 - d_2}{c} \tag{2.4}$$

This technique assumes two possible modes: uplink or downlink. In the uplink mode, the WD produces a signal which is received at 2 different BSs. This mode may have scalability problems since all the operations are made in the networks. In opposite, in the downlink, two BSs transmit, at the same time, a copy of the same signal, which is received by the WD. This mode is more scalable than the uplink mode since calculations are performed in the WD. Whatever is the mode of operation, the BSs need to be clock synchronized, though BSs and WDs do not need to be synchronized.

Concerning the modeling of the variable component of TDOA, in the present dissertation it is considered as gaussian distributed. Additional insight on this variable component is given in Chapter 7.

2.3.4 Angle of Arrival

The AoA is a measure of the angle from where the signal arrives to the receiver. The technical procedure for obtaining AoA measurements requires directional antennas and for this reason its implementation may represent additional cost and system complexity. In general, the AoA is related to an internal reference which is commonly chosen based on the topology of the network. For instance in multi BSs scenarios, the angles are commonly referenced to the virtual line that connects two of the BSs. A major problem with this type of measurements is the shadowing effect (Appendix A) present in the propagation channel. Since the shadowing effect biases the trajectory described by the main signal component, the signal will be received as arriving from a different angle than the one described by the physical positions of the transmitter and the receiver. Due to the complexity or even impossibility to detect the influence of this effect, it is acceptable to use the estimation of the angle as a rough indication of position in terrestrial systems.

A general and simple model of the AoA is to assume an internal reference to which all the angles and coordinates are related to, i.e. all the angles and the entire coordinate system is relative to that reference. Then, assuming the receiver placed at position (x_a, y_a) and the transmitter at position

(x_d, y_d), the AoA θ_a can be modeled by:

$$\theta_a = \text{atan}\left(\frac{y_d - y_a}{x_d - x_a}\right) \tag{2.5}$$

Note that the model of eq.(2.5) does not give a unique solution for (x_a, y_a). For this reason, it is necessary to implement some fusion techniques for estimating a single position for the WD, what is surveyed in Section 2.4.

Similarly to the models described for the previous channel measurements, the model for the AoA can also consider an additive noise component, which will be studied in more detail in Chapter 7.

2.4 Radio-localization Techniques for Data Fusion

Radio-localization techniques are those techniques that rely on the propagation channel in order to estimate the position of the WDs. As expected, these techniques have a strong dependency with the propagation conditions (Appendix A). In the location stack framework of Fig. 2.1, the techniques presented in this section belong to the fusion layer.

2.4.1 Triangulation

Triangulation is a technique that performs localization making use of distances among BSs and WDs. The calculation of the predicted location can be seen as the problem of determining the 3 vertices of a triangle given the sizes of the three edges and the location of 2 of those vertices. As Fig. 2.2 show, the two known vertices of the triangle are given by the position of two BSs with

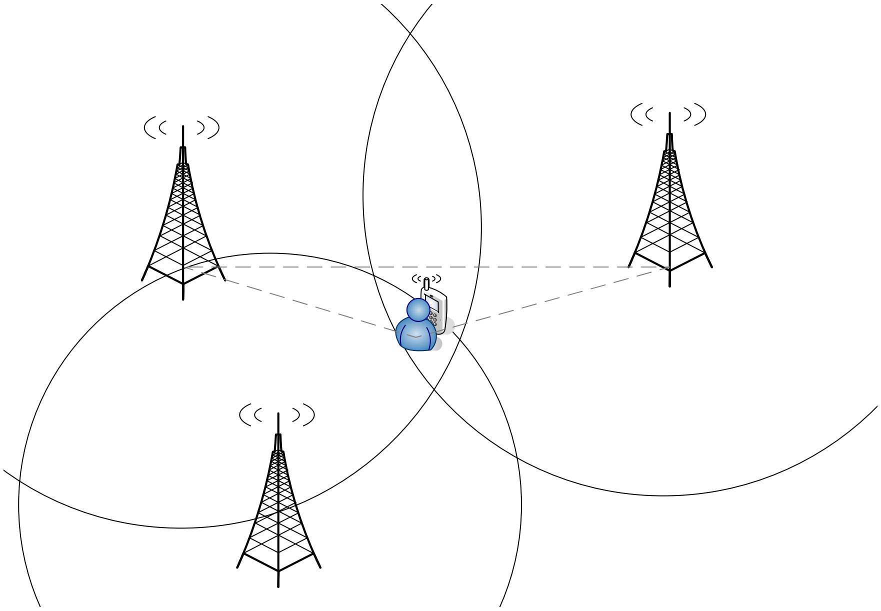


Figure 2.2: Triangulation Technique

known location, one of the edges is given by the distance between the 2 BSs and the remaining 2 edges are given by the estimated length of the WD-BS communication channel. Although 2 BSs define the concept of triangulation, there is still a flipping problem around the axis connecting both BSs. For this reason, with only two BSs, the predicted location may result in two points, for a 2-Dimensional (2D) space. Thus, the minimum number of necessary BSs is 3 (non collinear points), where the third BS decides which of the two positions is more probable. Similar idea can be described to motivate 4 non coplanar points as the minimal number of BSs for 3-Dimensional (3D) localization. The physical length of the communication channels is estimated using the range observations presented in Section 2.3.

As Fig. 2.2 shows, the triangulation method requires, in a 2D scenario, 3 BSs with known location. However, due to the noisy behavior of the measurements it is not possible to determine a single estimation for the WD position using trigonometric manipulation. The problem is that the link length estimators are corrupted by noise and subsequently they do not intercept in a single location. Thus, it is necessary to use an algorithm able of handling this noisy component in the link length estimations. Some of the available solutions to solve this problem are surveyed in Chapter 3 and can be divided in least squares approaches or stochastic approaches [KAB00, SN03]. Since stochastic approaches have been proven to be more robust in terms of localization accuracy, the further work introduced from Chapter 5 to Chapter 7 will use this approach. Additionally as a reference for further comparison the least squares approach is studied in Section E.

2.4.2 Hyperbolic Localization

Hyperbolic Localization is an alternative to the triangulation technique when link lengths are estimated using TDOA measurements. Due to the fact that these type of measurements do not depend on the link length but instead on the difference between two measures of link length, the TDOA measurements describe an hyperbola as the possible location for the user position (Fig. 2.3). For a 2D space, it is possible to estimate the (x, y) position of the WD from a system of equations

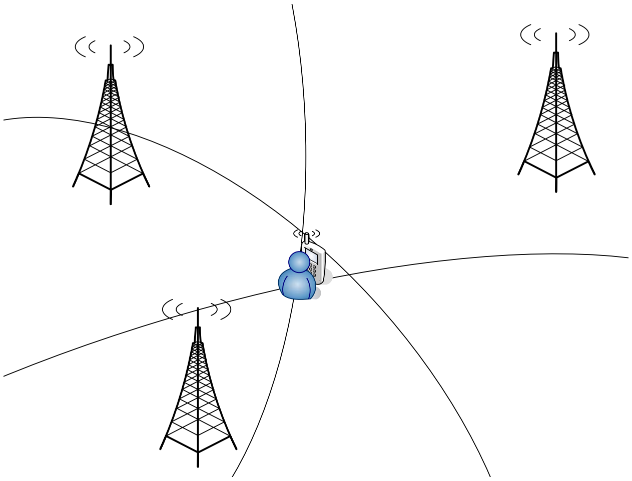


Figure 2.3: Hyperbolic Localization Technique

with 3 different TDOA readings. For the same reason that we need a third BS in order to have a unique solution using triangulation (Section 2.4.1), hyperbolic localization requires a third BS. Similarly to the triangulation technique, hyperbolic localization is subject to the problem of non-intercepting curves. The same approaches used to solve the problem in triangulation can be used in this case. Chapter 3 introduces these approaches in detail.

2.4.3 Angulation

Angulation is a technique that makes use of measurements of AoA. The algorithms relies, for a 2D scenario, on two BSs at known locations capable of measuring AoA (Fig 2.4). For a 3D space it is possible to use this technique if another measurement of azimuth is available. In this technique it is common to pre-define one of the directions of the vector that connects 2 of the BSs as the magnetic north. This vector is defined as the reference at 0°. Subsequently, the magnetic south is pre-defined at 180°. Each BS must be able to identify the angle from which the emission was originated, referenced to the pre-defined magnetic north. Several methods can be used to measure this angle such as directive antennas [VPP97] or methods making use of the signal phase properties [oDoT01]. After measuring the angles and knowing the location of the BSs, it is possible to predict the location of the WD.

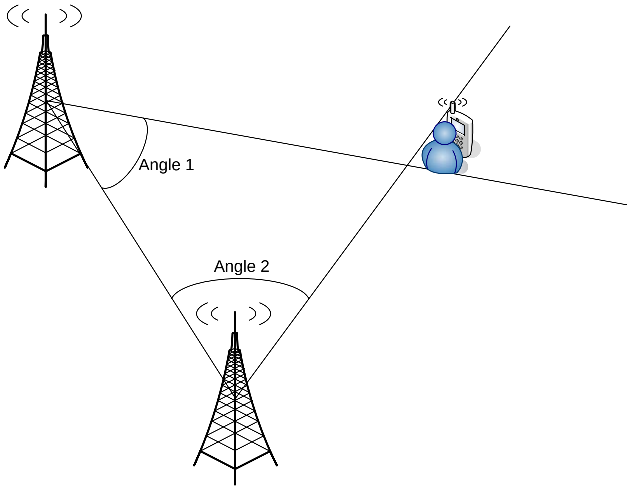


Figure 2.4: Angulation Technique

Due to the fact that Angulation uses AoA, the problem of systematic errors introduced by propagation shadowing is here present. Since this problem may be not noticeable by the BSs, the errors may be prohibitive for pinpointing a single estimated position. However, for a rough indication of the area where the WD is placed, Angulation can be a good candidate solution of localization. In Chapter 7, this technique is applied in a cooperative scenario.

2.5 Positioning Techniques Based on User Proximity

In proximity techniques, the idea is to identify the WD when it is “near” the BS. Using a physical phenomenon such as contact or coverage range, the BS is able to cover a certain area, identifying the WD when it is inside that area.

2.5.1 Physical Contact

This is the simplest technique, which localizes the WD when it establishes a physical contact with the BS. Here the definition of WD and BS may be somehow abusive, since this techniques relies basically on sensors of pressure, contact or distance.

2.5.2 Cell Identification

Cell Identification, also called CellID, assumes that the location of the BSs and the size of the cell are known or at least possible to estimate. This technique consists in determining to which BS the WD is connected to. The predicted location equals the area defined by the position of the BS and the size of the cell (Fig. 2.5).
In Cell Identification, the estimation of location cannot be pinpointed in terms of location (x, y) but in terms of an area where the WD can be found in.

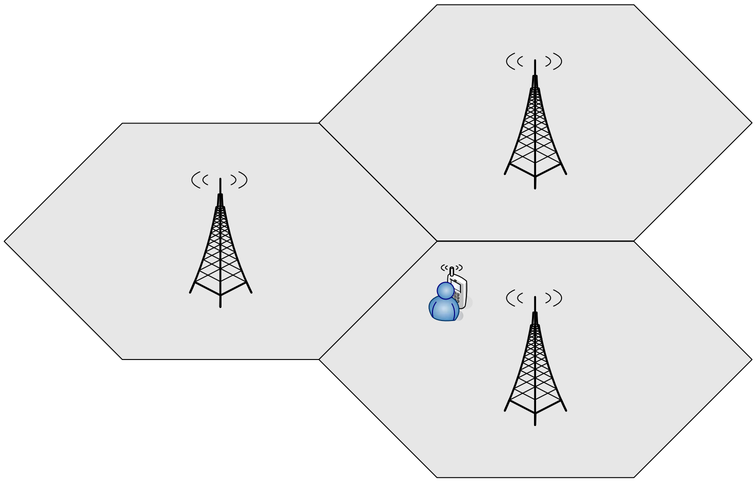


Figure 2.5: CellID Technique

2.6 State-of-Art on Localization Solutions for Wireless Networks

Positioning solutions for wireless communication networks are nowadays widespread around the market and around research laboratories all over the world. Commercially available, we can find for example Ubisense [Ubi]. This Ultra Wideband (UWB) based solution has well defined localization protocols and algorithms as stated by the technology specifications [Sah06]. Using a ToA technique, this solution can obtain typical values of a few centimeters of accuracy. For WLAN 802.11 technology, which contrarily to UWB does not have standard positioning solution in its specifications, there are several available solutions. MSR Radar [BP00] is one of the pioneer works in this WLAN localization, which approach has henceforth been followed in several research works such as [YASN02, HFL$^+$04, CCKM01, LBR$^+$02, PKC02], where Ekahau [EPS] is a commercially available solution. The common solution makes use of fingerprinting techniques [HB01a] and simple models of radio propagation in order to estimate user localization. Typical accuracy levels are lower than $10m$. Alternative solutions that use cellID localization methods in WLAN are Place Lab [LCC$^+$05, Lab], Skyhook Wireless [WFPS] or Navizon [Nav]. These solutions localize WDs with some tens of meters of error. In general, WLAN cellID based solutions require a database with knowledge on all the APs used for localization. The major problem is that, when aiming at global localization (such as the case of Navizon and Skyhook Wireless), it can be extremely costly, technically complex and virtually impossible to build and maintain, at a global scale, an entire database containing information concerning all the deployed APs. For Wireless Personal Area Network (WPAN) such as Bluetooth, there is a cellID solution called Blip Zones [BZ]. Also for the Blip Zones solution, errors are dependent on the size of the coverage area and the density of APs. For cellular solutions, there are commercially available solutions such as True Position [Pos] or Navizon [Nav], but majority is found in research literature [CSC$^+$06, OVLdL05, TV04, LAK$^+$01, CS98]. The general approach is to run a time consuming training phase where signals are captured in a strategy similar to the one specified by fingerprinting techniques. Typical values of accuracy range from $50m$ to $500m$ depending on the scenario conditions.

In general, for every wireless localization technique, there is a strong tradeoff between accuracy and user-friendliness. On one hand, enhancing accuracy implies a loss on user-friendliness, but on the other hand by loosing on accuracy it is possible to obtain a more user friendly system. For the sake of better understand this tradeoff, Chapter 3 will present some of the methods and techniques for data processing behind some of the most advanced localization solutions.

Chapter 3

Data Association for Maneuverable Target Tracking

The previous chapter has explained common basic methods for obtaining position information in wireless networks. Contrarily to the more conceptual approach used in the previous chapter, the present chapter explains practical implementations of those localization methods. The basic assumption is that the measurements, from where location information is estimated, are unavoidably corrupted by noise. For this reason, the current chapter surveys those data association algorithms able of handling the noisy measurements and reduce/minimize the effect of that noise in the estimation of the location of the WD. Additionally, these algorithms also aggregate information obtained from several sources, such as different APs, in order to provide a single output estimation. According to the location stack framework presented in Section 2.2 the algorithms in this chapter fall into the definition of the fusion layer.

3.1 The Bayesian Framework

Bayesian filtering techniques are techniques that estimate the state of a dynamic system based on noisy sensors measurements [AMG$^+$02, MG01]. These techniques are widely used within several contexts, such as Business & Economics, Signals & Systems, Image Processing or Mobile Tracking. To the specific case of location information estimation in scenarios with static or moving devices, Bayesian filtering techniques have been successfully applied in order to solve the problem of noisy measurements present in localization applications [FHL$^+$03].

The Bayes filter is an abstract concept that describes a probabilistic framework for recursive state estimation. Its main idea consists in estimating the distribution of the uncertainty, called *belief*, over the possible state space (i.e. range of possible user kinetics) given the sensor measurements obtained until the time of the estimation. Defining t_k as the instant in time when the $k-th$ measurement is obtained, x_k as the random variable that represents the state space at time t_k and

$z_{1:k}$ as the set of all the sensor measurements until time t_k, the *belief* is defined as:

$$\text{Bel}(x_k) = \text{pr}(x_k|z_{1:k}) \tag{3.1}$$

where $\text{pr}(x_k|z_{1:k})$ is the probability density function of x_k given $z_{1:k}$. Note that in localization applications, x_k often represents the user kinetics. However in this section, x_k is considered as an arbitrary state space.

As Eq.(3.1) shows, the belief has a dependency with all the measurements obtained from time t_0 until present time t_k, meaning that complexity increases with time. To solve this dependency, it is common to consider that the Bayesian filter has Markovian properties of first order, i.e. all the information necessary to compute the estimation of position at time t_k is given by the information available at time t_{k-1}.

As shown in the big picture of Fig. 3.1, the Bayesian filter consists in two main steps:

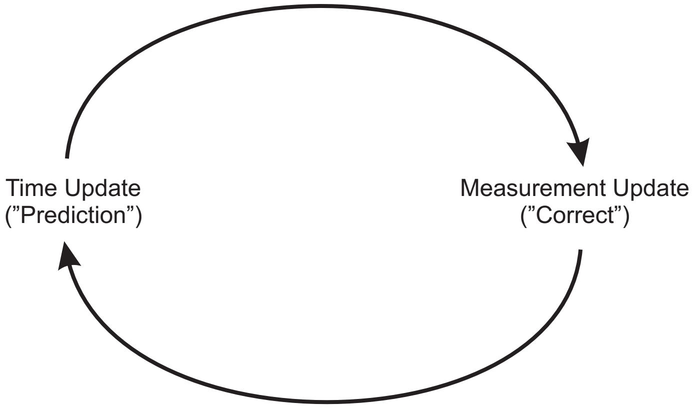


Figure 3.1: Bayesian filtering cycle: cyclic sequence of prediction and correction.

- **Prediction**: At each time, the current state x_k is updated based on previous states x_{k-1}:

  $$\text{Bel}^-(x_k) = \int \text{pr}(x_k|x_{k-1})\text{Bel}(x_{k-1})dx_{k-1} \tag{3.2}$$

  In Eq.(3.2) the term $\text{pr}(x_k|x_{k-1})$ describes the *system dynamics*, i.e. how the systems evolves from the state x_{k-1} to the current state x_k. The belief at time t_0 is usually defined as a reasonable density function within the context of the system in use.

- **Correction**: The predicted belief is corrected everytime a new sensor measurement is obtained:

  $$\text{Bel}(x_k) = \varrho_k \text{pr}(z_k|x_k)\text{Bel}^-(x_k) \tag{3.3}$$

  In Eq.(3.3) the term ϱ_k represents a normalization constant to ensure that $\int \text{Bel}(x_k)dx_k = 1$, while $\text{pr}(z_k|x_k)$ represents the *perceptual model*. The term $\text{pr}(z_k|x_k)$ describes the probability of having a sensor measurement z_k given that the system is in state x_k.

For localization purposes, the term $\mathrm{pr}(x_k|x_{k-1})$ in Eq.(3.2) is determined by the motion model, while the term $\mathrm{pr}(z_k|x_k)$ in Eq.(3.3) is dependent on the technology of the sensor and the location method.

3.1.1 The Kalman Filter and Kalman Smoother

The Kalman filter [MG01, KAB00, Kal60] is one of the most widely used implementations of a Bayesian filter. The basic assumptions of this filter are: (i) the models $\mathrm{pr}(x_k|x_{k-1})$ in Eq.(3.2) and $\mathrm{pr}(z_k|x_k)$ in Eq.(3.3) are linear; (ii) the noise statistics are Gaussian distributed. The filter defines five equations (set of expressions given by eq.(3.8) to eq.(3.12)), which in a recursive manner can estimate the state of an unknown noisy process. It is a very powerful filter because the amount of processing is considerably reduced compared with other Bayesian filters (such as the Particle filter in Section 3.1.3) and because it estimates the past, present and even future states with approximations to the true systems models. Since this filter was invented, it has been successfully applied in several areas of research. An example of such applications is tracking of moving devices. In the subsequent Chapters 5, 6 and 7, Kalman filters are used in the context of positioning in wireless networks.
The Kalman filter equations are able to predict the state vector X of a discrete-time process that is governed by the following linear system in state space form:

$$X_k = AX_{k-1} + BU_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{w} \sim \mathrm{Norm}(0, Q) \tag{3.4}$$

$$Z_k = HX_k + \mathbf{v}_k, \quad \mathbf{v} \sim \mathrm{Norm}(0, R) \tag{3.5}$$

In eq. (3.4), the matrix A defines the relation between the previous state and the current one, matrix B relates the previous external input vector U_k with the actual state, and $\mathbf{w}$ is a random variable representing the process noise. Note that the control input U can be known or unknown. In eq. (3.5), the matrix H represents the relation between the present state X_k and the measurement Z_k expected for that state X_k. The random variable $\mathbf{v}$ represents the measurement noise. In eq. (3.4) and eq. (3.5), the random variables $\mathbf{w}$ and $\mathbf{v}$ are assumed to be independent of each other, white, and have a normal distribution with zero mean and covariance matrices Q and R respectively. Having defined the expressions in eq.(3.4), eq.(3.5) and subsequent parameters, it is still necessary to define the starting values for the state $X_{0|0}$ and the error covariance estimator $P_{0|0}$. These parameters are system dependent. For the state $X_{0|0}$ the value shall be the expected value of the state space at time t_0. Concerning the $P_{0|0}$ it is commonly initialized with the covariance matrix Q. Based on the system of eq.(3.4) and eq.(3.5) the Kalman filter can be defined as:

Initialization:

$$\hat{X}_{0|0} = E[X_0] \tag{3.6}$$

$$P_{0|0} = E[(\hat{X}_{0|0} - X_0)(\hat{X}_{0|0} - X_0)^T] \tag{3.7}$$

Prediction:

$$\hat{X}_{k|k-1} = A_k \hat{X}_{k-1|k-1} + B_k U_{k-1|k-1} \tag{3.8}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \tag{3.9}$$

Correction:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{3.10}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k (Z_k - H \hat{X}_{k|k-1}) \tag{3.11}$$

$$P_{k|k} = (I - K_k H) P_{k|k-1} \tag{3.12}$$

From eq.(3.10) and eq.(3.11), the innovation process is identified as the difference from the obtained measurements Z_k and the expected measurements $H\hat{X}_{k|k-1}$. This process, by definition gaussian distributed, defines how much each individual measurement z_k contributes to the overall estimation of the state space $\hat{X}_k$. The innovation process and its subsequent covariance are defined as:

$$\tilde{Z}_k = Z_k - H \hat{X}_{k|k-1} \tag{3.13}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{3.14}$$

In Section 3.2.1, the innovation process is used as a solution for estimating abrupt changes in the dynamics of the WD.

Regarding the Kalman smoother, this dissertation uses the so called fixed-lag Kalman smoother, obtained from [TCS97, CST94]. This smother runs a Kalman filter similar to the one presented above in this section and then a set of additional equations named by [TCS97] as *retrospective analysis*. A smoothed estimation $\hat{X}_{k-\ell|k}$ at past time $t_{k-\ell}$, $\ell > 0$ can be obtained by running the expressions of the ordinary Kalman filter until time $t_{k-\ell}$ and then, beyond that time until current time t_k, run the *retrospective analysis* equations. The smoothed estimator $\hat{X}_{k-\ell|k}$ of the hidden process beyond time $t_{k-\ell}$, given observations until time t_k is calculated by:

Smoothing:

$$\hat{X}_{k-\ell|k} = \hat{X}_{k-\ell|k-1} + K_{k-\ell|k} \tilde{Z}_k \tag{3.15}$$

$$K_{k-\ell|k} = \left(P^{fa}_{k,k-\ell|k-1}\right)^T H_k^T S_k^{-1} \tag{3.16}$$

$$P^{fa}_{k,k-\ell|k-1} = A_k P^{aa}_{k-1,k-\ell|k-1} \tag{3.17}$$

$$P^{aa}_{k,k-\ell|k} = P^{fa}_{k,k-\ell|k-1} - K_k H_k P^{fa}_{k,k-\ell|k-1} \tag{3.18}$$

where the quantities $\tilde{Z}_k$ and S_k are respectively given by eq.(3.13) and eq.(3.14). The matrix $P^{fa}_{k,k-\ell|k-1}$ is called the forecast-retrospective analysis error cross-covariance matrix and $P^{aa}_{k,k-\ell|k}$ is the filter analysis-retrospective analysis error cross-covariance matrix. Basically, the matrices $P^{fa}_{k,k-\ell|k-1}$ and $P^{aa}_{k,k-\ell|k}$ propagate along time the uncertainty in the estimation at time $t_{k-\ell}$, given the measurements until time t_k. The covariance error $P_{k-\ell|k}$ at time $t_{k-\ell}$ is given by:

$$P_{k-\ell|k} = P_{k-\ell|k-1} - K_{k-\ell|k} H_k P^{fa}_{k,k-\ell|k-1} \tag{3.19}$$

The set of expressions from eq.(3.15) to eq.(3.19), together with the Kalman filter expressions from eq.(3.8) to eq.(3.12) define the Kalman filter.

3.1.2 The Extended and Unscented Kalman Filter for Nonlinear Estimation

As we have seen in the previous section, the Kalman filter is an algorithm that requires the models to be linear. The main problem with this approach is that in typical cases where such an algorithm could be useful, the models of the system are nonlinear, i.e.:

$$X_k = \mathcal{F}(X_{k-1}, U_{k-1}, \mathbf{w}_{k-1}), \quad \mathbf{w} \sim \mathrm{Norm}(0, Q) \tag{3.20}$$

$$Z_k = \mathcal{H}(X_k, \mathbf{v}_k), \quad \mathbf{v} \sim \mathrm{Norm}(0, R) \tag{3.21}$$

Researchers have proposed several extensions of the Kalman filer for the nonlinear cases. The most used solutions are the EKF, in detail for example in [MG01], and the Unscented Kalman Filter (UKF) proposed in [JU97]. These solutions deal with the nonlinearity by approximating it with Taylor expansion. While the EKF approximates the models to their first derivative, the UKF approximates to the second derivative.

The EKF is a common solution to use when the nonlinearity is not “very” strong, meaning that the error in this approximation is several orders of magnitude lower than the typical errors introduced by the noise on the system. The solution uses a Taylor expansion and truncates the models in the linear terms, obtaining expressions for the models equivalent to eq.(3.4) and eq.(3.5). The difference is that the terms A and H are now the Jacobian matrices instead of model specific parameters. Thus, those Jacobian matrices can be defined as:

$$A = \frac{\partial \mathcal{F}}{\partial X}(X_{k-1}, U_{k-1}, 0) \tag{3.22}$$

$$H = \frac{\partial \mathcal{H}}{\partial X}(X_k, 0) \tag{3.23}$$

Based on the nonlinear eq.(3.20), eq.(3.21) and the parameters from eq.(3.22) to eq.(3.23), the EKF equations can be defined as:

Initialization:

$$\hat{X}_{0|0} = E[X_0] \tag{3.24}$$

$$P_{0|0} = E[(\hat{X}_{0|0} - X_0)(\hat{X}_{0|0} - X_0)^T] \tag{3.25}$$

Prediction:

$$\hat{X}_{k|k-1} = \mathcal{F}(\hat{X}_{k-1|k-1}, U_{k-1|k-1}, 0) \tag{3.26}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_{k-1} \tag{3.27}$$

Correction:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{3.28}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k(Z_k - \mathcal{H}(\hat{X}_{k|k-1}, 0)) \tag{3.29}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{3.30}$$

The EKF as defined by eq.(3.26) to eq.(3.30), is extensively used throughout the Chapters 5 to Chapter 7. Additionally, to the EKF, the expressions from eq.(3.15) to eq.(3.18) are used in order to defined the EKS, which is used throughout the Chapter 5.
One of the drawbacks of the EKF approach for stochastic filtering is, as previously stated, the linearization of the models. This linearization depending on the application can have a high impact concerning the errors in the estimated mean and covariance. The problem gets more and more evident with the increasing importance that higher order terms have in the nonlinear model. For a more accurate approximation of the models the UKF was proposed.

Similarly to the EKF, the UKF assumes the noise in the model to follow a gaussian random distribution. The difference is that the distributions are represented by a minimal set of sample points chosen in such a way that the true mean and covariance of the errors are captured. As it was shown by [JU97], these samples, when propagated through the nonlinearity, accurately capture the posterior mean and covariance to the third order Taylor expansion of the nonlinearity. This procedure is known as the unscented transformation. Since this dissertation uses the UKF as a tool, the algorithm is presented but it is not given deep focus on the formulation details. The algorithm augments the hidden space X_k with the noise processes such that the hidden space becomes $X^a = [X^T \quad \mathbf{w}^T \quad \mathbf{v}^T]^T$. Then the algorithm can be described as:

Initialization:

$$\hat{X}_{0|0} = E[X_0] \tag{3.31}$$

$$P_{0|0} = E[(\hat{X}_{0|0} - X_0)(\hat{X}_{0|0} - X_0)^T] \tag{3.32}$$

$$\hat{X}^a_{0|0} = E[X^a_0] = [\hat{X}^T_{0|0} \quad \mathbf{0} \quad \mathbf{0}]^T \tag{3.33}$$

$$P^a_{0|0} = E[(\hat{X}^a_{0|0} - X^a_0)(\hat{X}^a_{0|0} - X^a_0)^T] = \begin{bmatrix} P_{0|0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \end{bmatrix} \tag{3.34}$$

where $\mathbf{0}$ is the zero matrix of adequate size, and Q and R are the noise covariance matrices of the hidden process (in eq.(3.20)) and observable process (in eq.(3.21)), respectively.

Prediction:

$$\mathsf{X}^a_{k-1|k-1} = [\hat{X}^a_{k-1|k-1} \quad \hat{X}^a_{k-1|k-1} \pm \sqrt{(L+\lambda)P^a_{k-1|k-1}}] \tag{3.35}$$

$$\mathsf{X}^x_{k|k-1} = \mathcal{F}(\mathsf{X}^x_{k-1|k-1}, 0, \mathsf{X}^w_{k-1|k-1}) \tag{3.36}$$

$$\hat{X}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \mathsf{X}^x_{k|k-1;i} \tag{3.37}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} W_i^{(c)} [\mathsf{X}^x_{k|k-1;i} - \hat{X}_{k|k-1}][\mathsf{X}^x_{k|k-1;i} - \hat{X}_{k|k-1}]^T \tag{3.38}$$

$$\mathsf{Z}_{k|k-1} = \mathcal{H}(\mathsf{X}^x_{k|k-1}, \mathsf{X}^v_{k|k-1}) \tag{3.39}$$

$$\hat{Z}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \mathsf{Z}_{k|k-1;i} \tag{3.40}$$

In eq.(3.35) $\mathsf{X}^a = [(\mathsf{X}^x)^T \quad (\mathsf{X}^w)^T \quad (\mathsf{X}^v)^T]$ are the sigma points, i.e. the points where the noise distributions are sampled. The notation $\sqrt{(L+\lambda)P^a_{k-1|k-1}}$ corresponds to the Cholesky factorization and its summation with $\hat{X}^a_{k-1|k-1}$ correspond to a column-wise sum. The parameters L and λ are the respectively the dimension of X and a scaling parameter, which is set to $\lambda = (1e3)^2 L - L$ as advised in [JU97]. In eq.(3.36) and eq.(3.39), $\mathcal{F}$ and $\mathcal{H}$ are respectively the evolution and the observation model as defined by eq.(3.20) and eq.(3.21). In eq.(3.37) and eq.(3.40), the weights $W_i^{(m)}$ are calculated as in [JU97] and the notation $\mathsf{X}^x_{k|k-1;i}$ corresponds to the column i of the matrix $\mathsf{X}^x_{k|k-1}$

Correction:

$$P_{z_k z_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathsf{Z}_{k|k-1;i} - \hat{Z}_{k|k-1}][\mathsf{Z}_{k|k-1;i} - \hat{Z}_{k|k-1}]^T \tag{3.41}$$

$$P_{x_k z_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathsf{X}_{k|k-1;i} - \hat{X}_{k|k-1}][\mathsf{Z}_{k|k-1;i} - \hat{Z}_{k|k-1}]^T \tag{3.42}$$

$$K_k = P_{x_k z_k} P^{-1}_{z_k z_k} \tag{3.43}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k (Z_k - \hat{Z}_{k|k-1}) \tag{3.44}$$

$$P_{k|k} = P_{k|k-1} + K_k P_{z_k z_k} K_k^T \tag{3.45}$$

where $W_i^{(c)}$, similarly to $W_i^{(m)}$ are calculated as in [JU97]. The matrix $P_{z_k z_k}$ corresponds to the innovation covariance and $P_{x_k z_k}$ corresponds to the cross correlations between the hidden and the observable process. The UKF, as presented in this section, is used in Chapter 7 in order to solve nonlinear problems.

3.1.3 The Bootstrap Particle Filter for Nonlinear Non-Gaussian Estimation

The Particles Filter (PF) is the name given to a Monte Carlo implementation of the Bayesian framework. This filter is adequate for nonlinear and non-gaussian estimation. It assumed several different approaches depending on the different problems existing on filtering a noisy set of data. As stated in [AMG$^+$02] these algorithms have their basis in the Sequential Importance Sampling. From the available solutions, we have decided to implement, due to its simplicity, the so called Bootstrap filter introduced by [GSS93].

Suppose that, at time t_{k-1}, M random samples $X^{(m)}_{k-1|k-1}$ (commonly named as particles) are available and that they are drawn from a probability distributions $\text{pr}(X_{k-1}|Z_{1:k})$, i.e. the belief $\text{Bel}(X_{k-1})$ as expressed in eq.(3.1). The bootstrap filter is an algorithm that propagates those samples through the system models (i.e. the models of eq.(3.20) and eq.(3.21)) in order to obtain the random samples $X^{(m)}_{k|k}$ that approximate the belief $\text{Bel}(X_k)$ at time t_k. Similarly to any Bayesian filter, the bootstrap filter runs cyclicly the prediction and the correction step:

Initialization:

$$X^{(m)}_{0|0} \sim \text{pr}(X_0) \tag{3.46}$$

Prediction:

$$X^{(m)}_{k-1|k-1} \sim \text{Bel}(X_{k-1}) \tag{3.47}$$

$$X^{(m)}_{k|k-1} \sim \mathcal{F}(X^{(m)}_{k-1|k-1}, U_{k-1|k-1}, \mathbf{w}^{(m)}_{k-1}) \quad \mathbf{w}^{(m)}_{k-1} \sim \text{pr}(\mathbf{w}_{k-1}) \tag{3.48}$$

where $\text{pr}(\mathbf{w}_{k-1})$ are samples drawn from the noise component $\mathbf{w}$ in the system dynamics.

Correction:

$$W^{(m)}_k = \frac{\text{pr}(Z_k|X^{(m)}_{k|k-1})}{\sum_{j=1}^{M} \text{pr}(Z_k|X^{(m)}_{k|k-1})}, \quad \text{pr}(Z_k|X^{(m)}_{k|k-1}) = \int \delta\Big(Z_k - \mathcal{H}(X^{(m)}_{k|k-1}, \mathbf{v}_k)\Big)\text{pr}(\mathbf{v}_k)d\mathbf{v}_k \tag{3.49}$$

$$\hat{X}_{k|k} = \sum_{j=1}^{M} W^{(m)}_k X^{(m)}_{k|k-1} \tag{3.50}$$

$$\text{Bel}(X_k) = \sum_{j=1}^{M} W^{(m)}_k \delta(X_k - X^{(m)}_{k|k-1}) \tag{3.51}$$

where $\hat{X}_{k|k}$ is the estimator of $X_{k|k}$ at time t_k given measurements until time t_k. The PF is used in Section 5.7, in order to study the influence of the assumptions made in the observation noise. The particle filter is there compared with the EKF.

3.2 Maneuverable Target Tracking

More complex algorithms attempt to estimate the trajectory of target WD with maneuvering properties. A maneuver is defined as an abrupt change in the movement dynamics. The additional problem in tracking maneuvering targets is that not only its position changes along time, but also the dynamic system governing the movement of the WD can undergo an abrupt change. Depending on the strategy for estimating the instants when maneuvers occur, there are two main approaches, either by using maneuver detection algorithms of multiple model algorithms.

3.2.1 Maneuver Detection

In the maneuver detection algorithm, the strategy is to evaluate the innovation process (given by eq.(3.13)) along time. As mentioned in [BSLK01], an adequate maneuver detection scheme for using in the present application is the so called White Noise Model with Adjustable Level. The detection of a maneuver is verified when a sequence of "large" innovations happen, what can be discovered by using the normalized innovations squared:

$$\varepsilon_k = \tilde{Z}_k^T S_k^{-1} \tilde{Z}_k \tag{3.52}$$

where $\tilde{Z}_k$ and S_k are respectively given by eq.(3.13) and eq.(3.14).

Since $\tilde{Z}_k$ is by assumption gaussian distributed, it is possible to prove that eq. (3.52) is a chi-squared random variable with l degrees of freedom (where l is the dimension of the observation vector Z).

Based on the assumption that ε_k is chi-squared, a possible solution for detecting a maneuver would be when the ε_k exceeds a predetermined threshold. However, due to the stochastic properties of ε_k, maneuvers can be incorrectly detected. To minimize this problem, an exponential discounted average can be used with a forgetting factor η.

$$\varepsilon_k^\eta = \eta \varepsilon_{k-1}^\eta + \varepsilon_k, \quad 0 < \eta < 1, \quad \varepsilon_0^\eta = 0 \tag{3.53}$$

where ε_k^η is, by first moment approximation, chi-squared distributed with $l/(1-\eta)$ degrees of freedom. In the case that $l/(1-\eta)$ is not an integer, the gama function shall be used. For additional details see [BSLK01].

When a maneuver is estimated, i.e., ε_k^η has exceeded the predefined threshold, the level of process noise is momentarily increased, meaning that uncertainty in the estimation increases. Inversely, when ε_k^η gets below the threshold, the low process noise is reused.

This maneuver detector with the exponential discounted average, given by eq.(3.53) is used in Chapter 6 in order to estimate maneuvering points in a segment-wise trajectory.

3.2.2 Multiple Model Solution for Handing Maneuvers

An alternative approach for tracking maneuvering users is to use a bank of r different models, running in parallel. Then a decision is required on how they can be combined. The r models can be considered for example as modeling different movement dynamics of the WD, e.g. static vs. moving, straight direction vs. transversal direction or horizontal movement vs. vertical movement.

The simplest multiple model approach assumes that the model that the system obeys to is fixed, but unknown. Thus, the problem is to decide which one of the r models is the one that produces the output which is observed. This solution is adequate when the model does not change along time. The final estimation can be considered as a decision of which is the most probable filter.

In a more advanced approach, the model that describes the system can undergo a switching scheme along time. These systems, also known as jump-linear systems or system jumping processes can be modeled by:

$$X_k = A(M_k)X_{k-1} + V(M_k)\mathbf{w}_k \tag{3.54}$$

$$Z_k = H(M_k)X_k + W(M_k)\mathbf{v}_k \tag{3.55}$$

where M_k denotes the model at time k. The model that is active at time t_k is assumed to be among the possible r models. Assuming that the r filters are running in parallel, a possible decision for combining all the filters can be taken by a switch variable Y_k. This variable specifies which movement pattern to use at each time instant t_k. For the case of Kalman filters, this switching variable decides which $\{A, Q, H, R\}$ matrices to use. The combination between the models is based on a weighted scheme where weights are given by $Pr(Y_k = i|Z_{1:k})$, i.e. the probability of using model i given measurements $Z_{1:k}$ from time t_1 until time t_k. This is called a "soft switching" scheme.
The *l-th* model history (or sequence of models) through time k is denoted as:

$$M^{k,l} = \{M_{i_{1,l}}, \dots M_{i_{k,l}}\}, \quad l = 1, r^k \tag{3.56}$$

where $i_{\kappa,l}$ is the model index at time κ from history l and

$$1 \leq i_{\kappa,l} \leq r, \quad \kappa = 1, \dots, k \tag{3.57}$$

Figure 3.2 gives a schematic representation of a multiple model running with 2 filters in parallel.

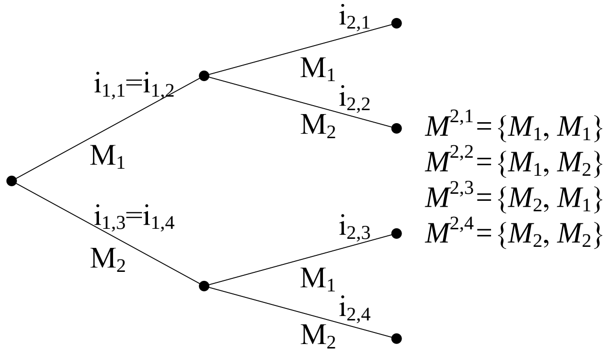


Figure 3.2: Example of execution of the multiple model framework. Two filters run in parallel during two units of time. The result is 4 histories.

A fundamental problem of this multi model framework is that running r filters in parallel represents a geometric complexity cost (see Fig. 3.2). The reason is that at at each time step, every single estimator output of the parallel filters shall be input into all the r filters.
For simplification and easier tractability of the mathematical formulation, it is common to assume that the mode switching has Markovian properties of first order.

$$m_{ij} \triangleq \Pr\{M(k) = M_j | M(k-1) = M_i\} \tag{3.58}$$

The system of equations defined by eq.(3.54), eq.(3.55) and eq.(3.58) is a generalized hidden Markov model. Given this system, it is possible to obtain the probability for each history as given by [BSLK01]:

$$\mu^{k,l} = \frac{1}{c} p[Z_k | M^{k,l}, Z_{1:k-1}] m_{ij} \mu^{k-1,s} \tag{3.59}$$

where $i = s_{k-1}$ is the last model of the parent sequence s. As it is shown in eq.(3.59), there is still the needed to condition on the entire history, even though the model switching has Markovian properties.

As it was shown, accounting with the entire history means manipulating an algorithm that grows exponentially in time. This behavior is clearly untractable after some few time steps, even for the smallest system with only 2 models. In order to solve this scalability problem, there are several approaches. The Generalized Pseudo-Bayesian (GPB) [BSLK01] approach is an algorithm that keeps track of recent history, collapsing older part of the history in a single estimator. Thus, the first order GPB collapses the history for time $k-1$ when the estimation is at time k. The second order GPB collapses the history until time $k-2$. Respectively the first and second order approaches use r and r^2 filters running in parallel. An alternative solution for the second order GPB is the Interacting Multiple Model, which reduces the number of parallel filter from r^2 to r. Additional alternatives can be see in the summary of [Mur88].

The multiple model framework is used in Section 5.6 for estimating different directions of movement. The filter assumed a switching scheme designed based on a Kalman filter.

3.3 Using Exponential Smoothing for Post-Processing

Exponential smoothing techniques are techniques often used in the business and economics literature [NIS07, BJ76]. The application on the localization context is possible and has been used in previous work [LaV03, AK07]. Time series techniques suit applications where data points corrupted by noise, present an internal structure such as trends or seasonal variations.

3.3.1 Single Exponential Smoothing

The Single Exponential Smoothing is the simplest technique of the family of exponential smoothing [NIS07]. This technique uses an exponential weighted averaging law in order to give more emphasis to more recent observations. This technique assumes no trends and thus it is best suit for localizing static nodes or nodes moving at velocities low enough to consider the position constant all over the averaging time window.

Thus, assuming that at time t_k, it is available an indirect observation of the position x_k of the WD, the smoothed estimator of the location of the WD is given by:

$$s_k = \gamma_s x_k + (1-\gamma_s) s_{k-1}, \quad 0 \le \gamma_s \le 1 \tag{3.60}$$

where the variable x_k is the measured position at time t_k and γ_s the smoothing forgetting factor. In absence of other information, the initial condition is commonly set as $s_1 = x_1$. Regarding the measurement x_k, it can be obtained by using the models presented in Chapter 2.

3.3.2 The Double Exponential Smoother

When the data points present a trend, a second exponential smoothing equation can be used for estimating that trend. The idea is to add a second equation that estimates the constant increment that each data point is subject to along time.

$$\begin{cases} s_t = \gamma_s x_t + (1-\gamma_s)(s_{t-1} + b_{t-1}) & 0 \le \gamma_s \le 1 \\ b_t = \gamma_b (s_t - s_{t-1}) + (1-\gamma_b) b_{t-1} & 0 \le \gamma_b \le 1 \end{cases} \tag{3.61}$$

where the parameter γ_b is the forgetting factor of the incremental trend b_k. In practice, s_k is the smoothed estimator of the position data points and b_k is the smoothed estimator of the increment that data points are subject to at each time step. It is important to notice that the first expression of eq.(3.61) includes the trend b_{t-1} at previous time, meaning that it accounts with the lag between two subsequent data points. Regarding the initial conditions of the eq.(3.61), they are commonly set as:

$$\begin{cases} s_1 = x_1 \\ b_1 = x_2 - x_1 \end{cases} \tag{3.62}$$

Forecasting the data points into future is possible using the following expression:

$$s^f_{t+m} = s_t + m b_t \tag{3.63}$$

where m is the number of time steps that forecast is looking at.

The double exponential smoother is used in Section 5.7 as a post-processing method for reducing the variability of the trajectory estimators, i.e. filter out undesirable noise.

Part III

Analysis

Chapter 4

Latency of Location Information in Short-Range Networks

Within the scope of the current dissertation, this chapter analyzes the timing aspects for obtaining location information. Timing aspects, in special relative to the latency while obtaining location information, is an important factor in a positioning system. The reason is that this latency time can be a limitation for a third party application making use of this location information. Moreover, for the cases of real time position systems, latency is generally a major problem which shall be minimized whenever it is possible. Thus, it is important to study the latency in wireless communications regarding the mechanisms to obtain location information.

For short-range communications, which represents the main focus of this dissertation, the study of the latency time strongly depends on the mechanism used for extracting location information. In the particular case of short-range communications, the most appropriate type of measurements are generally RSS, due to the fact that it is available in most technologies. Additionally, RSS is, for short-range narrow band systems, the most accurate information to be used for location. Mechanisms for obtaining RSS measurements in short-range communications are generally available integrated in power control mechanisms. However, since it is necessary to know the transmission power in order to measure the link length, localization methods shall preferably run when that mechanism is not in operation. This generally happens during the initial stages before connections are actually established.

Additionally, short-range networks tend to operate using ad-hoc links, where the devices operate according to its internal clock without mutual clock synchronization scheme. This means that the link layer protocols commonly require operational modes obeying to randomized processes. As a consequence, the latency time is not a straightforward problem, which needs to be studied as a stochastic process.

As an example of such short-range communication networks, this chapter will look at Bluetooth [Blu04]. The motivation for choosing this technology is that, on one hand, it is a well established technology that operates using asynchronous links, has a power control mechanisms, and implements randomized link layer protocols. On the other hand, Bluetooth is an under-used technology when compared with the levels of market penetration, meaning that any additional use of this technology can rapidly spread in the market. The most suitable procedure, to obtain location information, according to the aforementioned characteristics, is the inquiry procedure that Bluetooth runs during the initial steps for establishing a communications [Blu04]. During this procedure, power control is still not on operation. For all these reasons, the inquiry procedure is the mechanisms studied in more detail along this chapter.

The complexity of the inquiry procedure is rather high, meaning that a solid knowledge on the mechanism is required, specially, during the discussion of the results. For an initial overview, the reader is advised to read Appendix B. For a more detailed description, the specifications [Blu04] are an adequate reference, as well as [PBK06].

4.1 Introduction

Bluetooth is a short-range technology with typical range of $10m$ up to $100m$ depending on the class of the devices. Although its primary deployment envisages replacement of cables among small devices communicating through ad-hoc links, Bluetooth is also nowadays used as an infrastructure based network [BZ]. Enabling Bluetooth devices with positioning capabilities is possible by exploiting the power control feature present in the Bluetooth specifications [Blu04]. As part of the power control feature, devices are able to measure the received power in their communication channels. A problem in this approach is that power control is used for adjusting the transmission power when the receiver reports too high or too low power readings. Thus, it is necessary to use the power readings once the power control feature is not active. This is possible during the Inquiry Procedure of Bluetooth. The Inquiry Procedure is the mechanisms that is used for a device to search for neighbor devices and possibly connect to them in subsequent mechanisms. In short, the master, i.e. the device searching for neighbors, sends IDentity (ID) packets as a request for other devices to connect to it. If a second device, called slave, is at that same time listening for masters, it has the opportunity to answer by transmitting a Frequency Hop Synchronization (FHS) packet and proceed to establish a link between both devices. For a more detailed description of the Inquiry Procedure, see Appendix B.

Throughout this chapter, the latency time that we are interested to measure is often identified as the Inquiry Time. The reason is that when the Inquiry procedure is finished, the devices have already the available RSS measurements with which perform localization. In here it is assumed that the processing of the RSS information in order to obtain location information is ignorably lower than the inquiry time. The inquiry time metric stands for the time interval since a set of N APs is requested to start searching for one WD, until the moment that all the APs get to know about that WD. As it was explained in the background Section 2.4, the localization methods typically require multiple sources of information in order to estimate a single estimation of position, what explains the definition of inquiry time as given in the present chapter.

4.2 Measuring the Latency Time in Ad-Hoc Short-Range Communications

As previously mentioned, the use case for measuring latency time is based on the Bluetooth technology. The underlying feature which will support localization in Bluetooth is the inquiry procedure. The problem can be described as the time since the procedure of localization is started until the time when location information is available. By assuming that the time for processing the RSS measurements is ignorable when compared with the overall duration for obtaining those measurements, the latency time can be defined as the duration time of the inquiry procedure. As a randomized process as it is in the case of Bluetooth, the results envisage a generalization for ad-hoc short-range technologies.

For studying the inquiry time of Bluetooth it was decided to implement a simulator of the entire procedure instead of running an experimental setup. The main reason for adopting this approach is to have a clearer understanding on the sequence of events happening in this mechanisms and subsequently have a higher control on the evaluation. Additionally, evaluating the procedure by simulation allows to: (i) scale the problem to high number of APs; (ii) have high control and understand on the sequence of events; (iii) run the sequence of events faster than what would be possible with real equipment.

4.2.1 Scenario Description

Let us assume a Bluetooth network with N APs and a single static WD. The APs, supported by a backbone wired network, are controlled by a central server responsible for requesting position information (Fig. 4.1). Note that the scenario of Fig. 4.1 is a particular case of the general scenario described in Fig. 1.1, where only the short-range technology is taken into account. The APs, by request from the server start sending ID packets for searching the WD. Once each AP discovers the WD, it notifies the entity responsible for measuring the inquiry time, i.e. the server. The server, after receiving the notification from all the APs, logs the inquiry time and restarts the process

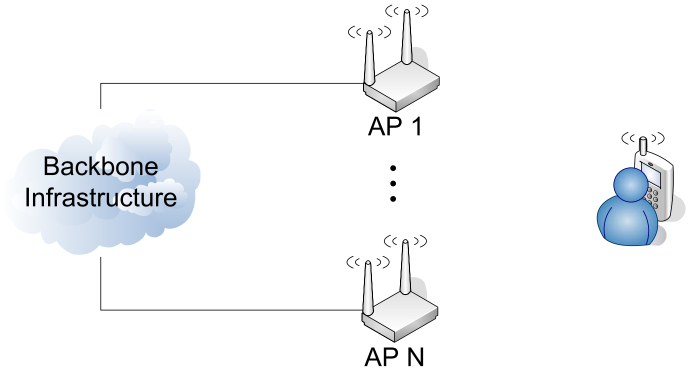


Figure 4.1: Description of the scenario used for studying the latency time of localization in short-range networks.

all over again. For the sake of a better understanding and a more detailed control of the Inquiry Procedure, the analysis started with a simple scenario with only a single AP and then it was extend to the case of N AP. Thus, it is considered:

- *Single AP scenario* - A single AP inquires the only WD. The inquiry time is defined as the elapsed time since the AP starts inquiring until when it discovers the WD.
- *Multiple AP scenario* - All the N APs inquire the WD. The inquiry time is defined as the time since the APs start inquiring until all of them have discovered the WD.

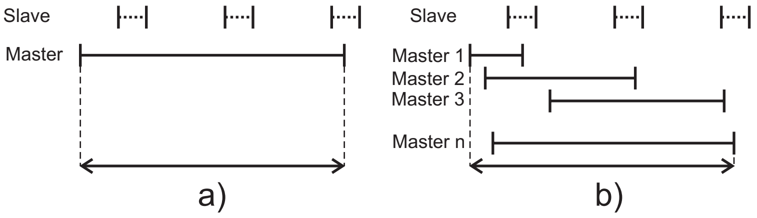


Figure 4.2: Inquiry Time: a) Single AP scenario; b) Multi AP scenario.

Figure 4.2 shows the definition of the inquiry time for the single and multiple AP scenario.

4.2.2 Simulating the Inquiry Procedure

In order to evaluate the *Inquiry Time* a simulator of the *Inquiry Procedure* was built. The protocol of the inquiry procedure was implemented as stated in the Bluetooth specifications V2.0 [Blu04]. Regarding the scenario, the additional approach was considered in the simulator:

- There is no interference with other technologies in the same frequency band (as WLAN or microwaves), meaning that apart of Bluetooth packet collisions, all transmitted packets are correctly received.
- Collisions among ID packets or between ID and FHS packets are treated as lost packets.
- There is no additional data packet traffic in the network; only the inquiry ID and FHS packets. This means that there is no additional delays introduced by other communication in the network.
- Every time a FHS packet is received, the corresponding AP stops sending ID packets, until the server request a new search.
- None of the APs or the WD are time synchronized. Whenever an AP enters the *Inquiry Substate*, it is delayed for a period uniformly distributed between $0s$ and $625\mu s$, the duration of one timeslot.
- The WD starts scanning in a period uniformly distributed between $0s$ and $1.28s$.

4.2.3 Simulation Results of Inquiry Time

Single AP-WD scenario

For the scenario of Fig. 4.1, using the above assumptions, the PDF of the *Inquiry Time* was computed for the Bluetooth specifications V2.0. The result for a single AP scenario is plotted in Fig. 4.3. To compare the simulation results, it was included Fig. 4.4, which can be found in the literature [PBK04]. Figure 4.4 besides representing the same PDF for a *Single AP scenario*, it was obtained for Bluetooth specifications V1.1 [SIG03].

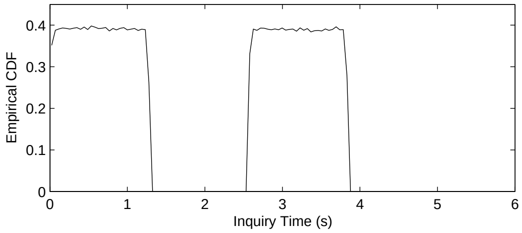


Figure 4.3: *Inquiry Time* PDF for a *Single AP scenario* with Bluetooth specifications V2.0

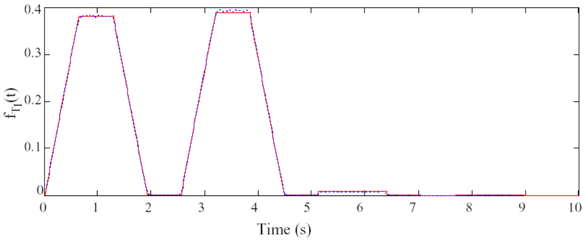


Figure 4.4: *Inquiry Time* PDF for a *Single AP scenario* with Bluetooth specifications V1.1 [PBK04].

Looking only to Fig. 4.3 it is possible to see that the empirical PDF presents two peaks: one starts at time $t = 0s$ and the other starts at time $t = 2.56s$. If we look into the Bluetooth specifications we can see that in fact, the time $t = 2.56s$ correspond to the exchange of inquiry trains from A to B that happens in the master (see detail in Appendix B). This means that, the set of 16 frequencies that have been used since $t = 0s$ to send ID packets, is exchanged at $t = 2.56s$ with the set of 16 frequencies that had not been used until the moment. On the other hand, since the slave can only scan one frequency each $1.28s$, the probability of this frequency being either in train A or B is 50%, what provokes the 2 peaks with the same probability density. Besides that we know that the scan interval for the slave is $1.28s$ with a scanning window of only $11.25ms$. Thus, since master and slave clocks are totally independent, when the master starts inquiring, the time at which the slave starts scanning is uniformly distributed between $0s$ and $1.28s - 11.25ms = 1.26875s$. This is the reason of the width of each peak being approximately $1.28s$.

Comparing now Fig. 4.3 with Fig. 4.4 we can see that besides the curves look similar, some differences are observed. In Fig. 4.4 the slope of the peaks is tilted and some small additional peaks appear for values of time higher than $t = 5s$. Both effects are due to the specification of Bluetooth that specifies that after the slave receives the first ID packet:

- in V2.0, the device first answers to the ID packet and then enters in back-off
- in V1.1, the device enters in back-off and then answers to the next ID packet received

As stated in Appendix B the back-off time is uniformly distributed between 0 and 1023 timeslots (approximately $639ms$). The fact that the back-off is before the answer, as in V1.1, influences the

Inquiry Time in the way that this back-off time has also to be considered in it. This results in the sum of two random variables: the time to get the ID packet; and the back-off time. The sum of another random variable (back-off time) represents in the distribution a convolution, resulting in the slopes observed in Fig. 4.4. On the other hand, the additional peaks are due to the case when the slave wakes up from back-off and the scanning frequency is not in the current train anymore.

Multiple APs inquiring a single WD

For further understanding the *Inquiry Time* behavior in a common network, the scenario is extended to the *Multiple AP scenario*. The evaluation consisted in determining the PDF of the *Inquiry Time* as defined in Section 4.2.1, for $n > 1$ APs and considering the assumptions stated in Section 4.2.2.

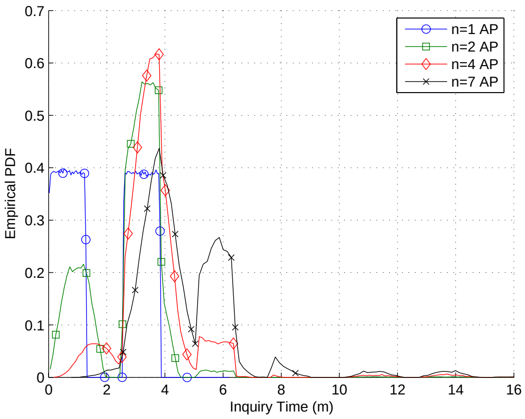


Figure 4.5: Inquiry Time PDF to a *Multiple Access Point Scenario* ($n = 1, 2, 4, 7$)

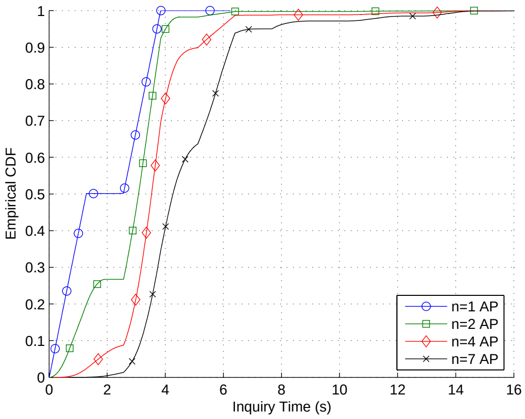


Figure 4.6: Inquiry Time CDF to a *Multiple Access Point Scenario* ($n = 1, 2, 4, 7$)

As Fig. 4.5 and Fig. 4.6 show, the *Inquiry Time* depends strongly on the number of APs: the bigger the number of APs, larger is the *Inquiry Time*. Furthermore, two effects are observed in Fig. 4.5: the first is that the relation existent in the *Simple AP Scenario* between *Inquiry Time* peaks and inquiry trains fades away when the number of APs grows. The second effect is the additional peaks that start to appear in the PDF for larger values of *Inquiry Time*. The CDF in Fig. 4.6 shows that the Inquiry time considerably increases with the number of APs.

Figure 4.7 shows the CDF for a *Multiple AP Scenario*, where the number of APs varies between $n = 1$ and $n = 20$. As it can be seen the *Inquiry Time* strongly depends on the number of APs. For scenarios with $n = 1, 2, 3, 4$ APs, it is possible to obtain a considerable percentage of *Inquiry Time* values in the first inquiry train ($t < 2.56s$). For scenarios with more APs, the percentage of *Inquiry Time* values obtained in the first inquiry is ignorable, but not in the second inquiry train ($2.56s < t < 5.12s$). For example for scenarios with $n = 5, 6, 7, 8, 9$ APs, the *Inquiry Time* is mainly obtained in the second inquiry train. For scenarios with n larger than 10 APs, more than 50% of the total number of *Inquiry Time* values are longer than the end of the second inquiry train, i.e.,

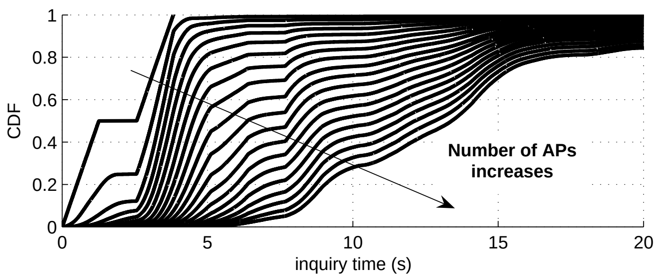


Figure 4.7: Simulated Inquiry time CDF to *Multiple Access Point Scenario* (n=1 up to 20) and 1 WD (obtained simulating 10^6 values of inquiry time for each settings)

longer than $5.12s$.

A different representation of the time behavior for localization procedures is presented in Fig. 4.8, which shows the *Inquiry Time* percentiles of 50%, 75%, 90% and 95% for a *Multiple AP Scenario*. These percentiles were chosen as reasonable boundaries for measuring performance. As Fig. 4.8 shows, for $n \leq 3$ APs, the *Inquiry Time* is lower than 5s even considering a high percentile such as 95%, i.e. the inquiry time is obtained before the end of the second train ($5.12s$). For $n = 4$ the percentile of 95% is higher than the duration of the first 2 trains.

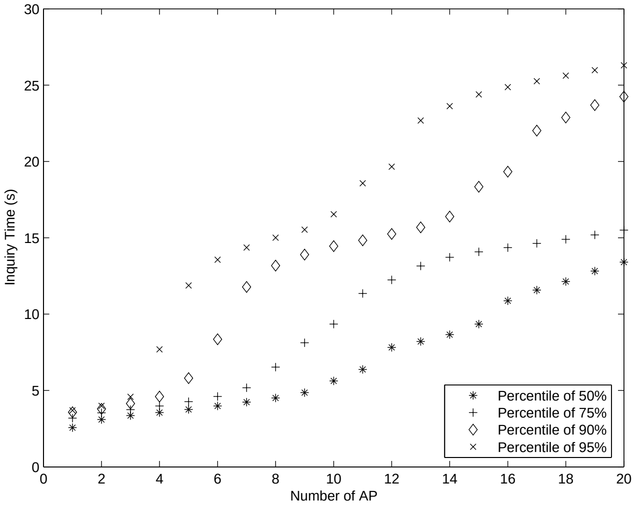


Figure 4.8: Inquiry Time duration depending on the number of APs for different percentiles

4.3 Approximating the Inquiry Time

Nowadays there are several frameworks where location information or, more generally, any context-aware factor is necessary for advanced networking and applications [BOJ+06, DA00]. Commonly one of the most influencing factors is the latency time for establishing communication between

nodes. Although the specific details of the protocol are not of major importance for the study of those frameworks, they need to be implemented in order to approximate the latency time. For this reason, a simple approximation of the distribution of this latency time can considerably ease the design of those frameworks and allow one o concentrate on the specific application. The Bluetooth technology is here used as an example of a ad-hoc technology. The solutions further presented only regard the inquiry procedure and not the entire protocol for establishing a connection between nodes. There are two main reasons for this decision. The first is that the connection latency in Bluetooth is mostly dominated by the latency of the inquiry procedure. Secondly, due to the fact that the current dissertation regards application of location, only the protocols permitting location information are studied, i.e. only the inquiry procedure is evaluated.
The first approximation considers the APs totally independent and non influencing each others. Thus, the PDF of Fig. 4.5 is compared with the distribution of the random variable of Eq. (4.1).

$$Y = \max_{i=0}^{n} X_i \tag{4.1}$$

In Eq. (4.1), the random variable Y represents the maximum of n independent random variables X_i, each one representing the *Inquiry Time* for the *Single AP Scenario.* Assuming that the n random variables X_i are independent and identically distributed, the random variable Y represents the *Inquiry Time* in a *Multiple AP Scenario* when APs do not influence each others. The distribution of Eq. (4.1) is compared with the CDF in Fig. 4.6 and the results are plotted in Fig. 4.9.

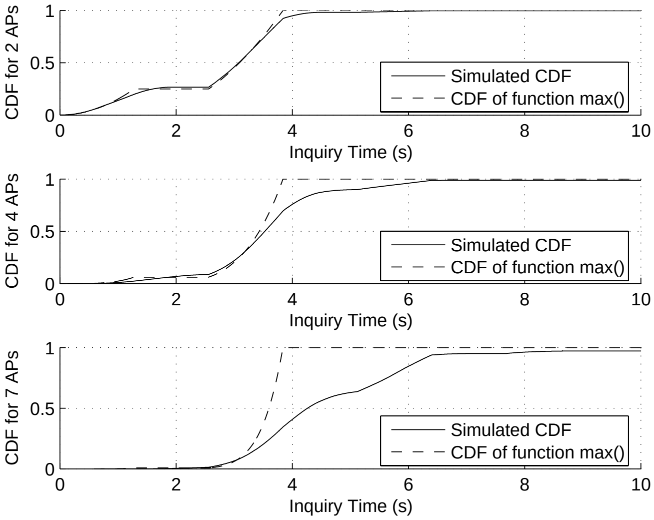


Figure 4.9: Simulated and approximated Inquiry Time CDF to a *2, 4, 7 AP scenario* (obtained simulating 10^6 values of inquiry time)

As observed in Fig. 4.9 increasing the number of APs result that both distributions start to diverge more and more. This fact let us conclude that there is in fact a strong AP-to-AP influence, which is more and more visible when the number of APs gets larger. There are two main reasons for this fact: the first because ID packets sent by the APs may collide with each others or even with FHS packets sent by the WD; the second reason is the back-off time. When the device receives an ID, it answers with an FHS packets and then it enters in a back-off time. This period makes the device totally unavailable to be discovered by another APs, what corresponds to the main explanation of influence among APs.

4.3.1 Analytical approach of the Inquiry Procedure in Bluetooth

Envisaging a simple generation of latency samples for the inquiry procedure, an analytical approach is considered in this section. For the case of a single AP a detailed analytical description is presented in [PBK06, DKNP06]. However, the presented analysis are complex and only regard one AP. For this reason, we introduce in this chapter a considerably simpler approach, which ignores some of the less influencing factors, and we extend it to 3 APs.
For the single AP scenario, a simple approximation is given in Appendix C due to its simplicity and straightforwardness. Extending the analytical formulation for a scenario with 2 APs implies the consideration of effects not present in a single AP scenario. As previously mentioned, the major influence among APs is due to the back-off time experienced by the WD after answering to inquiries. Thus, given that both APs have in their inquiry trains A, the frequency used by the WD during scanning, the sequence of events is:

1. the WD wakes up for scanning at a time uniformly distributed between 0 and $1.28s$;
2. the WD answers to one of the APs given that both are inquiring and have no packet collision problems;
3. the WD goes standby for a period uniformly distributed between 0 and 1023 timeslots;
4. The WD wakes up again and answers to the second AP. At this time the Inquiry Procedure is stopped and time measured.

Summing up the worse case when the scanning window and the back-off time are sampled at their maximum, the inquiry time takes $1.919s$. This time is lower than the $2.56s$ time when exchange of trains happen in the APs, meaning that the whole discovery is done within the time span of the inquiry train A.
When assuming that the scanning frequency belong to train A for one AP and train B for the other AP, the sequence of events is:

1. The steps from 1 to 3 still happen in the same way as it happens for the case when scanning frequency is in the same train for both APs.
2. When the WD wakes up, there are no more APs inquiring using the scanning frequency, because the second AP has the scanning frequency in its train B.
3. The WD goes standby for consecutive fixed periods of $1.28s$, until waking up after the second AP has changed the inquiry train to its B train.

4. The WD wakes up and answers to the second AP. This period since the train of frequencies is exchanged, i.e. $2.56s$, until the WD wake up is uniformly distributed between 0 and $1.28s$. When the WD answers, the inquiry time is finished.

At this point, and based on the just presented situations, it is possible to define a random variable describing which are the trains where the scanning frequency is in. This random variable has 3 possible outcomes:

- C_{2A} - The frequency used by the WD for scanning is for both APs in their train A.
- C_{1A1B} - The scanning frequency is in train A for one AP and in train B for the other AP.
- C_{2B} - The scanning frequency is in the train B of both APs.

Reasonably, the probability function of this random variable is given by $\Pr(C_{2A}) = \Pr(C_{2B}) = 1/4$ and $\Pr(C_{1A1B}) = 1/2$. Based on this random variable, PDF of the inquiry time can be described as:

$$\mathrm{pr}(T) = \Pr(C_{2A})\mathrm{pr}(T|C_{2A}) + \Pr(C_{2B})\mathrm{pr}(T|C_{2B}) + \Pr(C_{1A1B})\mathrm{pr}(T|C_{1A1B}) \tag{4.2}$$

$$\mathrm{pr}(T) = \frac{1}{4}\mathrm{pr}(T|C_{2A}) + \frac{1}{4}\mathrm{pr}(T|C_{2B}) + \frac{1}{2}\mathrm{pr}(T|C_{1A1B}) \tag{4.3}$$

Additionally, based on the sequence of events and given that the inquiry trains including the scanning frequency is know, the conditional distributions of eq.(4.3) are:

$$\begin{cases} \mathrm{pr}(T|C_{2A}) = Unif(0, 1.28) * Unif(0, 0.639) \\ \mathrm{pr}(T|C_{2B}) = Unif(2.56, 3.84) * Unif(0, 0.639) \\ \mathrm{pr}(T|C_{1A1B}) = Unif(2.56, 3.84) \end{cases} \tag{4.4}$$

Applying the eq.(4.4) in eq.(4.3), it is possible to obtain a theoretical approximation for the inquiry time for a scenario with 2 APs.

$$\mathrm{pr}(T) = \frac{1}{4}\Big[Unif(0, 1.28) + Unif(2.56, 3.84)\Big] * Unif(0, 0.639) + \frac{1}{2}Unif(2.56, 3.84) \tag{4.5}$$

The PDF and CDF of eq.(4.5) are respectively plotted in Fig. 4.10 and Fig. 4.11. As the visual results show, the approximation of the inquity time for a 2 AP scenario is worse then for the single AP scenario (Fig. C.2 in Appendix C). The reason is that with 2 APs the existence of actual collision between packets is already present. This can be clearly seen in the additional small peak at time between $5s$ and $6s$ in the PDF of Fig. 4.10. Additionally, besides packet collisions, the mismatch seen between both distributions in Fig. 4.11 is also due to the fact that the analytical result of eq.(4.5) is a simplified approach for the actual procedure described in the Bluetooth specifications.

Obtaining a similar analytical approach for scenarios with 3 APs is still possible and it is presented in Appendix C. However, extending for scenarios with more than 3 APs implies a dramatic increase on complexity. The main reason is that the random variable C (in eq.(4.2)) cannot be defined. To

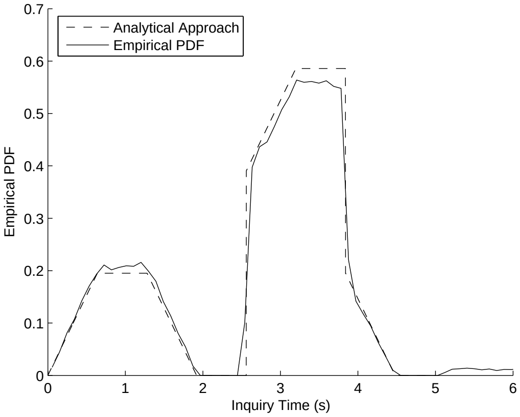


Figure 4.10: PDF of the inquiry time for a scenario with 2 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

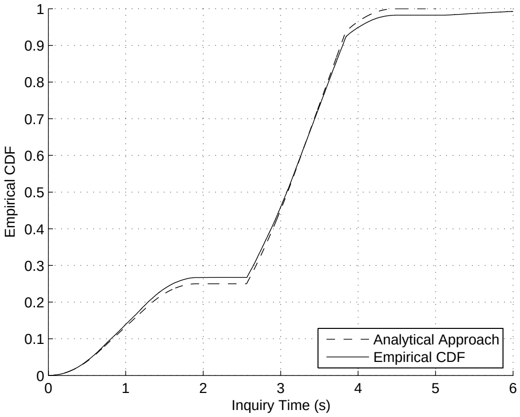


Figure 4.11: CDF of the inquiry time for a scenario with 2 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

understand the source of the problem, it is necessary to look at the case when all the APs have the scanning frequency in the same inquiry train. When this case happens, the inquiry time can be longer than the time span before exchange of inquiry trains, meaning that some of the APs that had the scanning frequency in its train A, start inquiring with train B. This additional problem considerably increases the complexity of the problem. Thus, for higher number of APs a different approach is given. Instead of obtaining an analytical formulation, the inquiry time is approximated by a monte carlo algorithm.

4.3.2 A fast and simple approach for simulating the Inquiry Procedure in Bluetooth

The *Inquiry Time* can be approximated by a ball-urn problem. In order to understand how can the problem be connected with the game, we decided to follow a process of reverse-engineering and present first the rules of the game and then make the connections with the Bluetooth Inquiry Procedure.

4.3.2.1 Inquiry Time as a ball-urn problem

The game requires N balls, 2 initially empty urns, and a counter with a cycle time of a as well as constant time values, T_i, T_f, and T_b. The game has the following steps:

1. Initialization: n_A balls out of N are picked up and inserted in Urn A, while the remaining balls ($n_B = N - n_A$) are inserted in Urn B. The variable n_A shall be randomly determined by a binomial distribution with bernoulli probability $p = 1/2$. Furthermore, the counter is set to value 0. This counter will be incremented after every time-interval of length a (in parallel to and independent of the subsequent steps).

2. We shall wait for a period of time t_i randomly determined by a uniform distribution between 0 and T_i.

3. We look at the counter and if the counter is even we pick up a ball from Urn A, if is odd we pick up a ball from Urn B as we can see in Fig. 4.12. If the corresponding urn does not contain any more balls, we move to Step 5.

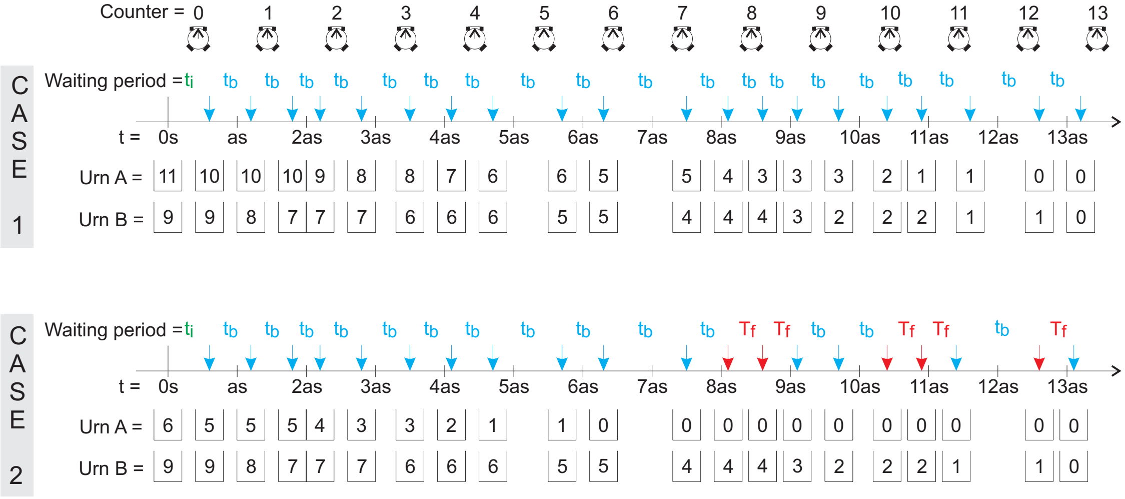

Counter = 0 1 2 3 4 5 6 7 8 9 10 11 12 13
CASE 1
Waiting period = ti tb tb tb tb tb tb tb tb tb tb tb tb tb tb tb tb tb tb tb
t = 0s as 2as 3as 4as 5as 6as 7as 8as 9as 10as 11as 12as 13as
Urn A = 11 10 10 10 9 8 8 7 6 6 5 5 4 3 3 3 2 1 1 0 0
Urn B = 9 9 8 7 7 7 6 6 6 5 5 4 4 4 3 2 2 2 1 1 0
CASE 2
Waiting period = ti tb tb tb tb tb tb tb tb tb tb tb Tf Tf tb tb Tf Tf tb Tf
t = 0s as 2as 3as 4as 5as 6as 7as 8as 9as 10as 11as 12as 13as
Urn A = 6 5 5 5 4 3 3 2 1 1 0 0 0 0 0 0 0 0 0 0 0
Urn B = 9 9 8 7 7 7 6 6 6 5 5 4 4 4 3 2 2 2 1 1 0

Figure 4.12: Execution of the ball-urn game that approximates the inquiry time.

4. We shall wait a period of time t_b randomly determined by a uniform distribution between 0 and T_b and then we return to Step 3.

5. If an attempt to take a ball from an empty urn is performed, the number of balls in both urns is unchanged, and we wait for a fixed period of time T_f before returning to Step 3.

The game ends when both urns are empty and the inquiry time is the time shown in the clock. Figure 4.12 illustrates two example runs of the game. Only in Case 2, an attempt to access an empty urn occurs.

4.3.2.2 The relation between the game and the Inquiry Procedure

In this section we explain how the Bluetooth Inquiry Procedure can be approximated by the ball-urn game presented in Section 4.3.2.1.

As previously introduced in Section 4.2.1 the *Inquiry Time* is defined, for a *Multiple AP Scenario*, as the time since the first AP starts inquiring until the last AP notices that there is a WD around. Additionally and as it was assumed in Section 4.2.3, the APs are forced to start inquiring nearly at the same time, which is defined as the starting time of the process (time $t = 0$ in Fig. 4.12). Considering that there is a change of frequency each $1.28s$ after starting inquiring, the APs will suffer, all nearly at the same time, a change in its trains every $1.28s$. Because the sequence of frequencies is pre-determined and unique in the Inquiry procedure, all the APs will follow the same sequence, though they may be at different offsets in that sequence. From the point of view of the WD, the new frequency scanned $1.28s$ after a determined scan also follows the same sequence, meaning that all the system can be considered as if there was dome sort of synchronization on the way the frequency selection evolves in time. Because of this synchronization we can build 2 groups of APs: the group A that includes the APs that have in its train A the frequency scanned by the WD; and the group B that includes those that have the scanning frequency in their tain B. This corresponds in our game to the existence of 2 urns: A and B.

As all the APs and the WD have a free running clock, the probability of the scanning frequency be in train A is 1/2, what is known as a Bernoulli distribution with parameter $p = 1/2$. For a number of cases n, the probability of having n_A cases succeeding in train A follows, by definition, a binomial distribution, which corresponds to the first part of the game.

As the APs interchanges the inquiry train every $2.56s$, we can say that if the scanning frequency was belonging previously to the group A, after the interchange it starts to belong to the group B, corresponding this, in our game, to the fact that instead of picking up a ball from the urn A, we start picking up from urn B, being the cycle a equal to $2.56s$.

When the Inquiry Procedure starts, the WD may or not be in inquiry scan substate. If it is not already in this substate, the WD will start scanning in a period of time uniformly distributed between $0s$ and $1.28s$. Concerning the game, the period of time corresponds to the random variable t_i and the time limit of $1.28s$ corresponds to T_i.

After the WD starts scanning, the WD searches for a determined frequency for $11.25ms$. During this period of time, if the WD is triggered with an ID packet, the WD enters in a back-off time uniformly distributed between 0 and 1023 time slots, what means roughly between $0s$ and $0.639s$. In fact, correlating with our game, the period of time of $11.25ms$ that the WD is scanning corresponds to the time spend picking up a ball, that for simplification we ignore because it is much lower than the other time values involved in the inquiry procedure. The back-off time corresponds to the variable t_b and the maximum back-off time of 1023 times slots corresponds to T_b.

If all the APs in group A already discovered the device and if the device scans for a frequency that does not belong to the used train of any of the remaining APs, we have the case 2 in Fig. 4.12, where T_f is the fixed time of the scan window of $1.28s$.

In the Inquiry Procedure, when all the APs discovered the device we have the same solution as when we picked up all the balls in our game.

4.3.2.3 The Algorithm

In this section we present the algorithm to generate samples from the inquiry time distribution for N APs, which is directly derived from the ball urn game explained in Section 4.3.2.1.

```
InqTime = 0;
Urns[0] = Bin(1/2, N); // $n_A$
Urns[1] = N - Urns[0]; // $n_B$
NextTime = Unif(0,1) * $T_i$;
while (Urns not [0,0])
    InqTime = InqTime + NextTime;
    UrnNumb = Int(InqTime/$a$) mod 2;
    if (Urns[UrnNumb] is 0)
        NextTime = $T_f$;
    else
        NextTime = Unif(0,1) * $T_b$;
        Urns[UrnNumb] = Urns[UrnNumb] - 1;
return InqTime;
```

4.3.2.4 Simulation Results

In order to evaluate our algorithm we compare it with a simulator of Bluetooth that implements the entire inquiry procedure from the specifications [Blu04]. The simulator and our algorithm were both implemented in C++ language under Linux environment.

In the first step of the evaluation aims at comparing the distribution of the inquiry time obtained via the ball-urn model (InqSimpl) with the full Bluetooth procedure (InqSim). For that 100.000 samples of the inquiry time were generated via the two different methods. The empiric CDF is plotted in Fig. 4.13. While comparing Fig. 4.13 with Fig. 4.9 that shows an initial attempt to

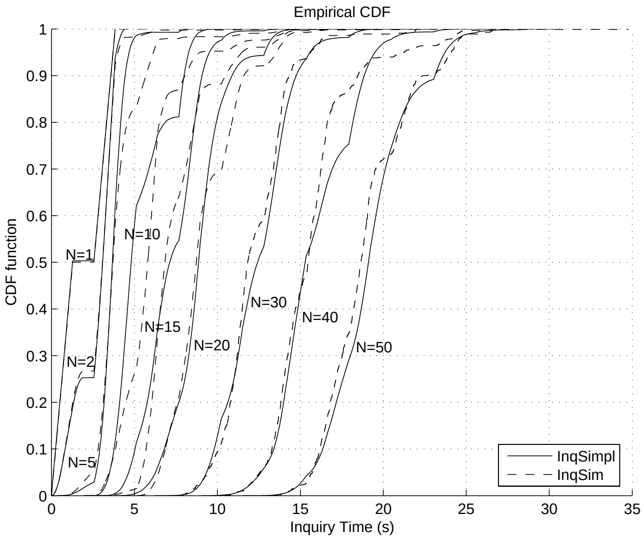


Figure 4.13: CDF of the inquiry times from the Bluetooth procedure (InqSim - dashed line) in comparison to the abstracted model (InqSimpl - solid line) for $N = 1, 2, 5, 10, 15, 20, 30, 40$ and 50 APs.

approximate the Inquiry Time, the proposed algorithm clearly outperforms the previous attempt. The second phase of the evaluation aims at evaluating the execution time of the algorithm in Section 4.3.2.3 in comparison with the simulator of Bluetooth. For this comparison, both simulators were implemented in C++ and were run in the same machine. The executions time (processing time) to generate one samples of the inquiry time is measured. The average of 1000 of these runs is shown in Fig.4.14 for the standard Bluetooth simulation (InqSim) and for the ball-urn simulation (InqSimpl). As Fig.4.14 show, the processing time of the proposed algorithm is approximately 10^4 times faster than the Bluetooth Simulator. Moreover, our algorithm shows an approximately linear complexity (approximately $O(N)$) with respect to the number of APs. By inspection of the log-log plot, an approximate polynomial complexity $O(N^\alpha)$ with exponent $\alpha \approx 2$ can be observed. In summary, both with respect to absolute values of execution as well as with respect to asymptotic growth for large number of APs, the proposed algorithm largely outperforms a full simulation model.

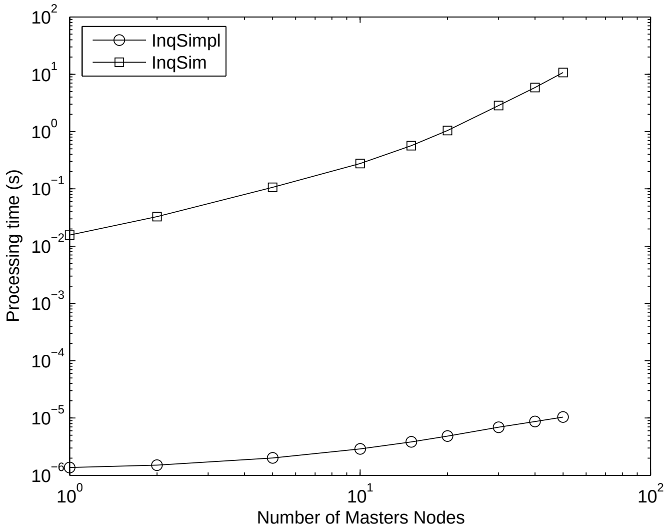


Figure 4.14: Average time to generate one samples of the inquiry time. The results were obtained out of 1000 runs of each case and each number of nodes.

4.4 Conclusions

In this chapter the latency time for obtaining location information was analyzed in ad-hoc short-range networks. Obatining location information in ad-hoc networks is possible by exploiting the mechanisms providing reading of the wireless channel, which can be relatable with user position. The latency time of the inquiry time, where power control was in use at, was analyzed in Bluetooth networks. The purpose of studying Bluetooth is because it is an actual example of an ad-hoc short-range communication that operates using asynchronous links, has a power control mechanisms, and implements randomized link layer protocols.

In this chapter, the Inquiry Time was analyzed. The first evaluation concerned a scenario where a single AP inquires a single WD. The results show that the inquiry trains have a visible influence in the Inquiry Time PDF. However, it is also possible to see that, for higher number of APs inquiring the same WD, the influence of the inquiry trains get less visible. Regarding the Inquiry Time for a single AP, the PDF presents two peaks with equal density. These peaks with a width of $1.28s$ correspond to the inquiry scan duration of the WD. Moreover the two peaks start one at $0s$ and the other at $2.56s$, where 2.56 is the time spend scanning with one train set before change trains. The analysis of the Bluetooth specifications V2.0 have shown that the Inquiry Time in V2.0 is faster than in V1.1. For multiple APs scenarios, the results show that the APs have a high influence on the overall Inquiry Time, what was shown to be caused by the non responsive period of the AP when experiencing the back-off time. In particular for a scenario with 3 APs, which is the common minimum to perform a triangulation method of localization, the mean time for obtaining responses from all the 3 APs is $3.5s$. For scenarios with a single or two APs, it was given an analytical approximation of the Inquiry Time distribution. However, for a larger number of APs, the inter-influence among APs results in a considerably high complex problem to handle. Thus, an algorithm for approximating the Inquiry Time for multiple APs was introduced. The algorithm presents a total abstraction of the actual protocol stated in the specifications V2.0, by considering the Inquiry Time as a ball-urn event. The proposed algorithm is simpler and faster to run or implement than an implementation of the entire Bluetooth specifications.

Generalizing, it was observed that the procedures that run pseudo-arbitrary mechanisms to avoid collision between communication packets have a deep impact on the latency time, when a single devices is being requested by several nodes for a connection. Thus, the statistics of the latency time are expected to vary depending on the number of nodes in the networks. In particular, for localization applications using signal strength due to the fact that multiple sources of data are required.

Chapter 5

Tracking of Wireless Devices in Infrastructure-Based Networks Using Received Power Measurements

In this chapter, the problem of tracking WDs in wireless short-range communications is addressed. The scenario is defined as a set of short-range technology APs supported by a backbone infrastructure of wired-communication (Fig. 5.1). The wireless technology is assumed to support a mechanisms such as power control, from where it is possible to obtain readings of RSS from the wireless channel. Thus, the present chapter intends to use RSS information for estimating the position of the WD. An inherent problem is that the measurements do not carry full information about the position of the WDs. Instead they give information about the AP-WD distance. To overcome this problem, data fusion techniques presented in Chapter 3 must be applied in order to obtain position information and subsequently tracking information.

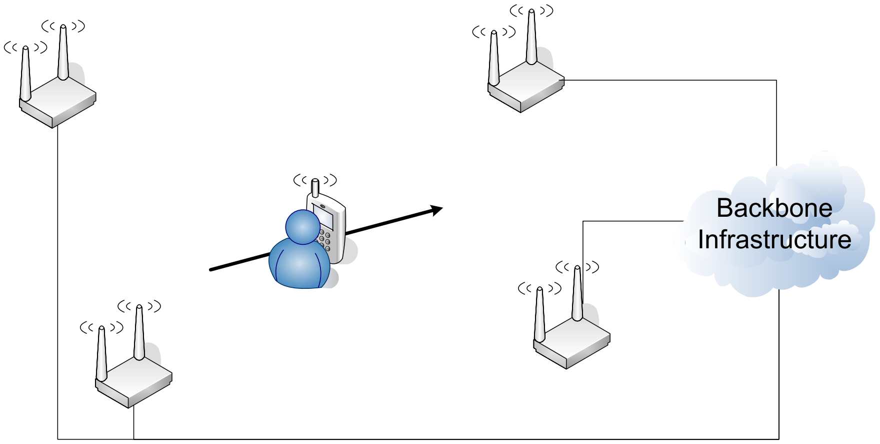


Figure 5.1: Example of a wireless short-range technology used for estimating user location information.

The contributions of this chapter are:

- Real-life experimental results on positioning accuracy obtained in Bluetooth networks and indoor scenarios. The Kalman filtering and smoothing approaches explained in Section 3.1.1 and Section 3.1.2 commonly used in the target tracking literature are evaluated in the context of wireless communication networks.

- Data fusion solutions based on multi-models of the target trajectory (explained in Section 3.2.2), e.g. SEKF, are analyzed for solving the problem of abrupt changes in the trajectory parameters. A switching scheme for model selection is also presented

- For modeling the non-Gaussian noise of the measurements, the PF explained in Section 3.1.3 is used and compared with the other solutions. Additional fast smoothing of the estimators is proposed by using an exponential averaging.

5.1 Problem Definition

Consider a network, such as in Fig. 5.1, composed by a single WD and N wireless APs. The APs are wired-connected to a larger infrastructure capable of providing communication services to the client. Due to the communication protocol between each AP and the WD, only a pair AP-WD is possible at a time. At each time step the active AP is uniformly chosen from the entire set of APs. The position of each AP n $(n = 1, ..., N)$ is static and known as $\Lambda_n = [a_n \quad b_n]^T$.

Consider now that the WD is a maneuverable target moving along an unknown trajectory $\mathcal{T}$. The full characterization of the entire trajectory is possible by knowing the dynamics $\mathcal{F}$ of the target and its initial condition X_0. Additionally, the WD is subject of discrete changes in its dynamics, defined as an unknown control input U_i. Describing the kinetics of the WD by $X(t)$, the trajectory $\mathcal{T}$ can be defined as:

$$\mathcal{T} \triangleq \mathcal{F} : \Big(X(t), U_i, t\Big) \to \dot{X}(t), \quad X(t_0) = X_0 \tag{5.1}$$

Note that in eq.(5.1), the dependency with time is explicitly stated in the definition of the trajectory. The control input is assumed piecewise constant with $U_k = U(t)$, $t_{k-\ell} \leq t < t_k$, (with $\ell > 0$ and unknown), when the continuous-time system is discretized. The same control input U_k is assumed to be the only responsible for changes in the direction of the movement of the WD. Thus, the discretization of U_k results in considering that the trajectory is segment-wise linear as shown in Fig. 5.2. Considering the existence of process noise $\mathbf{w}$ to model some uncertainty in the trajectory estimation, eq.(5.1) can be rewritten as:

$$\dot{X}(t) = \mathcal{F}\Big(X(t), U_k, t\Big) + \mathbf{w}(t) \tag{5.2}$$

where the noise $\mathbf{w}$ is assumed zero-mean Gaussian distributed with covariance matrix Q. Finally the trajectory is assumed 2-dimensional and the entire movement is described by constant velocity

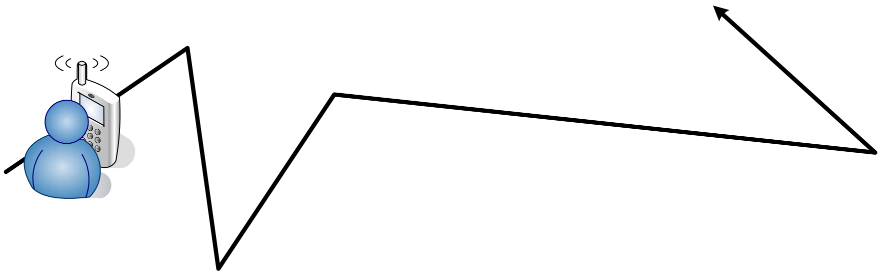


Figure 5.2: Example of a segment-wise trajectory described by the WD

within each segment. Note that even though a change in velocity may happen from segment to segment, acceleration is not estimated due to the nature of the measurements and the complexity of the problem. Since the measurements are highly corrupted by noise, it is practically impossible to detect small variations in higher order moments than the velocity. Thus, for simplicity it is assumed that in the worst case, there is a jump in the velocity which can be due to the control input U_k.

Assume that for localization purposes, the APs can measure values of RSS in the communication WD-AP. These measurements Z are however corrupted by a noise component $\mathbf{v}$, which is assumed to have known statistics. Thus, given that at time t_k there is only a single measurement of power, Z equals the path loss expression in eq.(2.2).

$$Z_k = \alpha - 10\beta \log\left(\left|\left|X_k, \Lambda_{n_k}\right|\right|\right) + \mathbf{v}(t) \tag{5.3}$$

$$Z_k = \alpha - 5\beta \log\left((x_k - a_{n_k})^2 + (y_k - b_{n_k})^2\right) + \mathbf{v}(t) \tag{5.4}$$

where Λ_{n_k} is a random variable that expresses the position of the AP that is active at time t_k.

$$\Lambda_{n_k} \sim \text{Unif}\Big(\Lambda_n \quad : \quad n = 1, ..., N\Big) \tag{5.5}$$

Note that while $\Lambda_n = [a_n \quad b_n]^T$ is the position of AP n, $\Lambda_{n_k} = [a_{n_k} \quad b_{n_k}]^T$ is the position of the AP responsible for the measurement obtained at discrete time t_k. Equivalently to eq.(2.2), in eq.(5.3), the parameter α depends on the transmission power as well as structural parameters of the communication system. The parameter β, known in the literature as the path loss exponent, is strongly influenced by the scenario where the system is operating.

As eq.(5.3) shows, it is possible in fact to obtain an estimation of the AP-WD distance by using observations of received power strength. However, in order to obtain an estimation of position, it is necessary to associate data from several sources. Otherwise a problem of system non-observability may occur, i.e. under no noise assumption, it may be impossible to unambiguously map observations of received power strength to a single estimation of position. This is the reason that, for example, it is 3 the minimum number of APs required in the triangulation method explained in Section 2.4.1. Given that no changes in the movement model occur along a certain period, i.e. U_k keeps constant, the system gets observable after some few measurement observations. The minimum number of observations required to guaranty observability of the system is studied in Section 5.4. Once that U_k changes in the movement model, the system looses observability, resulting that it needs to obtain again a minimum set of observations in order to regain observability. In the worse case, when U_k changes so rapidly that the system is not able to obtain a minimum set of measurements before U_k change again to guaranty observability, the system may never get observable or rely on wrong estimators. For this reason, it is necessary to determine the minimal relation between the rate of obtaining measurements and the rate of changes in the control input U_k.

5.2 Performance Metrics

In order to compare the performance of the position algorithms, three metrics are introduced in this section: the CDE, the PMDE and the IPMDE. The CDE metric, without any dependency with time, only measures the length of the trajectory. Instead the PMDE and IPMDE metrics take into account the time dependency and give two different definitions for the trajectory mismatches.

Covered Distance Error (CDE)

The CDE metric gives a measure of how close the length of the estimated trajectory is compared to the length of the true trajectory.

$$\varepsilon_{cde} = \frac{\hat{D}_k - D_k}{D_k} \tag{5.6}$$

with

$$D_k = \sum_{i=1}^{k} ||X_i, X_{i-1}|| \tag{5.7}$$

$$\hat{D}_k = \sum_{i=1}^{k} ||\hat{X}_i, \hat{X}_{i-1}|| \tag{5.8}$$

where X_i is the true position at discrete time t_i when received signal strength measurements are obtained, and $\hat{X}_i$ is the position estimator at the same corresponding discrete time.

From eq.(5.6) it is possible to see that: (i) when the true and estimated trajectories are equally long, the error is zero; (ii) when the estimated path is shorter than the real one, the error varies in the interval $[-1;0[$; (iii) when the estimated path is longer than the real one, the error varies in the interval $]0;\infty[$.

Pointwise Mean Distance Error

The PMDE is a measure of performance that takes into account the time dependency. It gives a notion of distance between the estimated trajectory and the true one.

$$\varepsilon_{pmde} = \frac{1}{t_k - t_0} \sum_{i=1}^{k} (t_i - t_{i-1}) ||X_i, \hat{X}_i|| \tag{5.9}$$

where t_i is the discrete time value when the $i - th$ received power measurements is obtained. Equation (5.9) shows that the error is weighted by the elapsed time since the previous measurement has occurred. This weighting solution is done in order to give less relevance to measurements obtained after short elapsed time values and favor those obtained after larger elapsed time values.

Interpolated Pointwise Mean Distance Error (IPMDE)

The IPMDE is a measure of performance that takes into account the position errors regardless of the time they have occurred. The metric attempts to measure the proximity of the estimated trajectory with respect to the real one. It assumes that it is required time $t_k - t_0$ to walk both trajectories. Then, is assumes that both trajectories were walked at constant velocity (though different velocity in each trajectory). Under these assumptions, it determines the time instants when the sequence of positions (either true or estimated) would be expected, i.e.

$$t'_{0:k} = \left\{ (t_k - t_0) \frac{D_l}{D_k} + t_0 \quad : \quad l = 1, ..., k \right\} \tag{5.10}$$

Then, given that the position sequence $X_{0:k}$ have actually occurred at time $t_{0:k}$, the sequence of positions $X_{0:k}$ is interpolated at times $t'_{0:k}$.

$$X'_{0:k} = \text{interp}\Big(t'_{0:k} \quad ; \quad X_{0:k} \quad ; \quad t_{0:k} \Big) \tag{5.11}$$

In eq.(5.11), the function $x_{ref} = \text{interp}(t_{in}, x_{in}, t_{ref})$ performs one-dimensional interpolation of the variable x_{in} dependent on t_{in} at the points defined by t_{ref}, in order to produce the result of x_{ref}. Note that t_{in} and x_{in} must have the same units as t_{ref} and x_{ref} respectively.

In the last step, the common PMDE metric is used with the position coordinates dictated by the interpolation of eq.(5.11)

$$\varepsilon_{spmde} = \frac{1}{t_k - t_0} \sum_{i=1}^{k} (t_i - t_{i-1}) ||X'_i, \hat{X}'_i|| \tag{5.12}$$

Note that eq.(5.12) indirectly depends on the estimated length of the trajectory $\hat{D}_k$. This length since is calculated as the sum of the segments between subsequent estimators of position (see eq.(5.8), it may be severely influenced by a noisy estimator of position.

5.3 System Implementation for Performance Analysis

In order to analyze the performance of the reference localization model and further proposed solutions, an experimental setup was performed. The experiment aims at measuring, in a real scenario, the performance of the positioning methods used in this chapter. The propagation conditions are uncontrollable, as it is so in common positioning applications.

5.3.1 Scenario Description

The layout of the indoor scenario used for experiments of localization is depicted in Fig. 5.3. This scenario was chosen as an example of a potential application for a positioning solution. The scenario is a $20m \times 15m$ animal stable, where all the animals, equipment and scenario obstructions were in the area. As Fig. 5.3 shows, 15 APs of Bluetooth class 1 were placed in the area. The decision on the number of APs was to counterbalance the highly erroneous measurements and their low frequency in Bluetooth. These APs were programmed to continuously measure the Received Signal Strength Indicator (RSSI) from the WD and report it back to a server which was responsible for estimating position. The WD used in the experiment was also a Bluetooth class 1 tag, which is assumed to move along pre-determine trajectories.

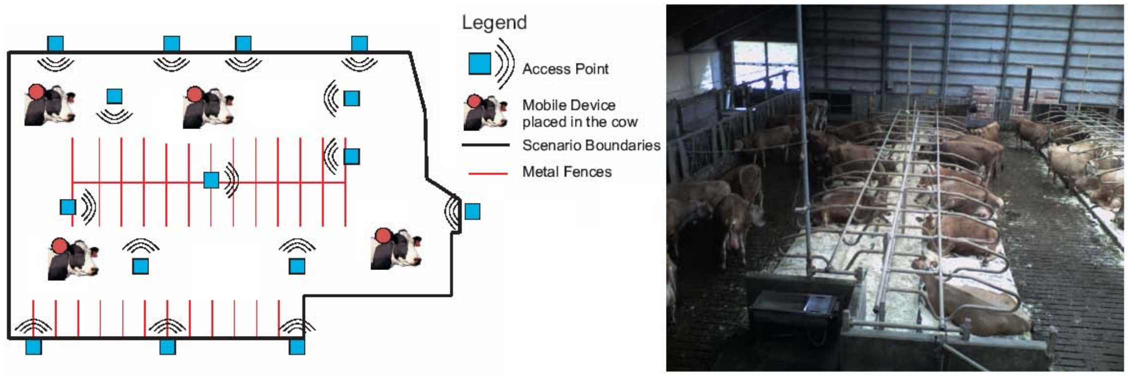


Figure 5.3: Setup of the Bluetooth network and layout of the indoor scenario used to run the localization experiment.

5.3.2 Experimental Procedures

The procedure for running the experiment was performed according to the following steps:

- In an initial stage, the trajectory of the device was defined and clear path-descriptive marks were made in the scenario.
- The WD carried by a person was forced to describe the pre-defined trajectory at relatively slow speed of the order of some few cm/s.
- During the walk, the APs were continuously requested to record measurements of RSSI taken from the links AP-WD, as well as the timestamps when each measurement was obtained.

This data was used later to run the localization algorithms and subsequently estimate the trajectory of the WD.

- Whenever the device was walking by the pre-defined marks existent in the scenario, the timestamps and their true positions were manually recorded. This information is later used as the true trajectory for comparison with the estimated trajectory (previous step).

It is important to mention that the the notions of time are relative to the internal clock of the server responsible for storing all the data.

In the current scenario, two different trajectories were defined. The first one, Fig. 5.4, describes a straight line movement with constant speed and constant direction. This scenario approximates the simplest case of movement where no abrupt changes on any parameter happen. The second trajectory, Fig. 5.5, instead describes a movement where some changes of direction happen, but the speed is still maintained constant. For both trajectories, 4 runs of the same experiment were carried out. Table 5.1 shows the values used in the initial state of the experiment, i.e., the position from where the WD has started its movement and the constant velocity of the WD.

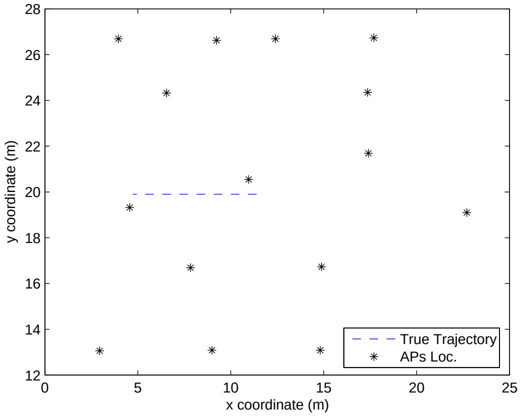


Figure 5.4: Straight trajectory defined for running the experiments.

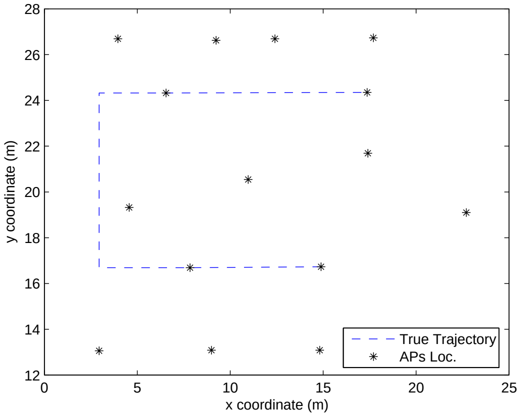


Figure 5.5: Squared trajectory defined for running the experiments.

Table 5.1: Initial state used during the experiment.

Parameter	Straight Trajectory	Squared Trajectory
(x_0, y_0) (m)	(11.4 ; 19.9)	(14.9 ; 16.7)
velocity (m/s)	0.023	0.045

5.4 The problem of Obtaining a Single Measurement at a Time

As briefly mentioned in the introduction of the current chapter, system observability must be verified. Estimating an hidden state X_i based on an observable process Y_i requires the system to be observable [Che99, KAB00]. A system is considered observed (see [Che99]) if for an unknown initial state X_0, the information given by the outputs Y_l and the inputs U_l suffices to uniquely determine X_0 in a finite number of time steps $l > 0$.

As eq.(5.4) shows, power measurements allow a unique estimation of WD-AP distance, but not a unique estimation for the position of the WD. In order to obtain this unique solution for the position of the WD, there are two possibilities: either by associating measurements from several sources (section 2.4), i.e. several APs; or by assuming that the movement dynamics of the WD behave in such a way that only a single position is possible for each measurement [Son99]. For instance if the movement is linear and the direction is known, each single observation of RSS can be mapped to a single position. See [Son99] for a more complex example.

Assuming that the relation between position and the measurements is linear and that the matrix relating the observations and hidden states is invertible, the system is observable according to the aforementioned definition. Thus, by obtaining a linear expression relating power measurements and position, observability can be shown to the present system. The eq.(5.4) shows that at time t_k the position to be estimated (x_k, y_k) can be related with the active AP n_k at position (a_{n_k}, b_{n_k}) and the power measurement z_k as:

$$(x_k - a_{n_k})^2 + (y_k - b_{n_k})^2 = 10^{(\alpha - z_k)/(5\beta)} \tag{5.13}$$

$$z'_k = (x_k - a_{n_k})^2 + (y_k - b_{n_k})^2 \tag{5.14}$$

Let us assume, for simplification purposes, constant speed along the entire experiment. Given that the timestamps of the measurements were obtained at discrete times t_k, the initial position (x_0, y_0) and the constant speed components $(\dot{x}, \dot{y})$, it is possible to determine the position at time t_k as:

$$\begin{cases} x_k = x_0 + t_k \dot{x} \\ y_k = y_0 + t_k \dot{y} \end{cases} \tag{5.15}$$

where t_0 is assumed zero without loss of generality.

Assuming that measurement z'_k is available at time t_k, it is possible to relate that measurement with the position and speed at time t_0 by:

$$z'_k = (x_0 + t_k \dot{x} - a_{n_k})^2 + (y_0 + t_k \dot{y} - b_{n_k})^2 \tag{5.16}$$

At this point, the concept of a macro is introduced in order to simplify the subsequent equations. A macro is a conceptual and compact representation of an expression. Whereas in a function the parameters are values, in a macro they are symbols to be replaced in the expression. With this definition, the following macro is specified:

$$\mathcal{M}\langle\mu\rangle\langle\nu\rangle \triangleq (t_{\nu-1}^2 - t_{\nu-2}^2)\mu_\nu - (t_\nu^2 - t_{\nu-2}^2)\mu_{\nu-1} + (t_\nu^2 - t_{\nu-1}^2)\mu_{\nu-2} \tag{5.17}$$

where μ and ν represent arbitrary symbols to be replaced in the macro. Combining eq.(5.16) and eq.(5.17), it is possible to obtain:

$$\mathcal{M}\langle z'\rangle\langle k\rangle = 2\mathcal{M}\langle t\rangle\langle k\rangle(x_0\dot{x}+y_0\dot{y})-2\mathcal{M}\langle a\rangle\langle k\rangle x_0-2\mathcal{M}\langle b\rangle\langle k\rangle y_0-2\mathcal{M}\langle a'\rangle\langle k\rangle\dot{x}+2\mathcal{M}\langle b'\rangle\langle k\rangle\dot{y}+\mathcal{M}\langle c\rangle\langle k\rangle \tag{5.18}$$

where

$$a'_k = a_{n_k}t_k, \quad b'_k = b_{n_k}t_k, \quad c_k = a_{n_k}^2 + b_{n_k}^2 \tag{5.19}$$

As we can see in eq.(5.18) the only nonlinearity with respect to x_0, y_0, $\dot{x}$ or $\dot{y}$ was reduced to the term $x_0\dot{x} + y_0\dot{y}$. Attempting to cancel out that nonlinearity, let us define the following macro:

$$\mathcal{N}\langle\mu\rangle\langle\nu\rangle \triangleq (t_{\nu-1}^2-t_{\nu-2}^2)\Big[t_{\nu-3}\mathcal{M}\langle\mu\rangle\langle\nu\rangle-t_\nu\mathcal{M}\langle\mu\rangle\langle\nu-1\rangle\Big]-(t_{\nu-1}-t_{\nu-2})\Big[t_{\nu-3}^2\mathcal{M}\langle\mu\rangle\langle\nu\rangle-t_\nu^2\mathcal{M}\langle\mu\rangle\langle\nu-1\rangle\Big] \tag{5.20}$$

Combining eq.(5.18) and eq.(5.20) it is possible to obtain a linear relation between the power and the target kinetics at time t_0.

$$\mathcal{N}\langle z'\rangle\langle k\rangle = -2\mathcal{N}\langle a\rangle\langle k\rangle x_0 - 2\mathcal{N}\langle b\rangle\langle k\rangle y_0 - 2\mathcal{N}\langle a'\rangle\langle k\rangle\dot{x} - 2\mathcal{N}\langle b'\rangle\langle k\rangle\dot{y} + \mathcal{N}\langle c\rangle\langle k\rangle \tag{5.21}$$

Given that there are 4 variables and that eq.(5.21) is linear, we can conclude that by having 4 linearly independent equations (for different times t_k), it is possible to show that the system is observable. Thus, a system of linear equations based on the relation of eq.(5.21) between the measurements and the state space $X = [x_0, y_0, \dot{x}, \dot{y}]^T$ can be written in a matrix form as:

$$Z = HX \tag{5.22}$$

where

$$Z = \frac{1}{2}\begin{bmatrix} \mathcal{N}\langle c\rangle\langle k\rangle - \mathcal{N}\langle z'\rangle\langle k\rangle \\ \mathcal{N}\langle c\rangle\langle k+1\rangle - \mathcal{N}\langle z'\rangle\langle k+1\rangle \\ \mathcal{N}\langle c\rangle\langle k+2\rangle - \mathcal{N}\langle z'\rangle\langle k+2\rangle \\ \mathcal{N}\langle c\rangle\langle k+3\rangle - \mathcal{N}\langle z'\rangle\langle k+3\rangle \end{bmatrix} \tag{5.23}$$

$$H = \begin{bmatrix} \mathcal{N}\langle a\rangle\langle k\rangle & \mathcal{N}\langle b\rangle\langle k\rangle & \mathcal{N}\langle a'\rangle\langle k\rangle & \mathcal{N}\langle b'\rangle\langle k\rangle \\ \mathcal{N}\langle a\rangle\langle k+1\rangle & \mathcal{N}\langle b\rangle\langle k+1\rangle & \mathcal{N}\langle a'\rangle\langle k+1\rangle & \mathcal{N}\langle b'\rangle\langle k+1\rangle \\ \mathcal{N}\langle a\rangle\langle k+2\rangle & \mathcal{N}\langle b\rangle\langle k+2\rangle & \mathcal{N}\langle a'\rangle\langle k+2\rangle & \mathcal{N}\langle b'\rangle\langle k+2\rangle \\ \mathcal{N}\langle a\rangle\langle k+3\rangle & \mathcal{N}\langle b\rangle\langle k+3\rangle & \mathcal{N}\langle a'\rangle\langle k+3\rangle & \mathcal{N}\langle b'\rangle\langle k+3\rangle \end{bmatrix} \tag{5.24}$$

Although observability is proven, it is still not known how many measurements are needed and from how many APs they must be obtained. Inspecting eq.(5.24) and its entries, one can easily understand that the index k cannot take the values $\{0, 1, 2\}$. The reason is that the corresponding row of H would be 0. From this we can conclude that the minimum number of measurements are 7, the first 3 used to define the eq.(5.21) and the remaining 4 which alow H to be invertible. Concerning the number of APs it is also possible to conclude that the minimum number is 3.

5.5 Tracking Using the Extended Kalman Filter/Smoother

The first method used to solve the problem of tracking WDs in a wireless network is the EKF, for real-time estimation, and the EKS, for smoothing the pre-determined trajectory in past time.

5.5.1 Design of the filter

As it was explained in Section 3.1.1 and Section 3.1.2, the EKF and the EKS require a clear definition of the hidden process and the observable process. Since we are interested to localize devices in a 2 dimensional plane, moving at a constant speed, we can define the hidden state as the x and y coordinates as well as the velocity in each respective coordinate. The acceleration instead is not considered due to the fact that it is assumed to be zero. Thus, for discrete time t_k:

$$X_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix}^T \tag{5.25}$$

The evolution of the system is modeled according to the physical laws of a constant velocity straight line movement as in eq.(5.2), i.e.

$$\hat{X}_k = A_k \hat{X}_{k-1} \quad \text{with} \quad A_k = \begin{bmatrix} 1 & 0 & T_k & 0 \\ 0 & 1 & 0 & T_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.26}$$

where $T_k = t_k - t_{k-1}$.

The noise process is considered as uniquely introduced in the acceleration. Although the mean value of the acceleration is considered zero, the noise is assumed to be in the acceleration in order to give a more realistic physical meaning for the model. Manipulating the kinematics of such sort of movement, similarly to what is done in [Sta01], it is possible to obtain the covariance matrix defined as:

$$Q_k = T_k \begin{pmatrix} \sigma_{\ddot{x}}^2 T_k^2/3 & 0 & \sigma_{\ddot{x}}^2 T_k/2 & 0 \\ 0 & \sigma_{\ddot{y}}^2 T_k^2/3 & 0 & \sigma_{\ddot{y}}^2 T_k/2 \\ \sigma_{\ddot{x}}^2 T_k/2 & 0 & \sigma_{\ddot{x}}^2 & 0 \\ 0 & \sigma_{\ddot{y}}^2 T_k/2 & 0 & \sigma_{\ddot{y}}^2 \end{pmatrix} \tag{5.27}$$

where the parameters $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ correspond to the standard deviation of the zero mean gaussian noise in the acceleration coordinates x and y respectively.

Regarding the observable state, the measurements from each one of the APs shall be used as the entries of the observation vector Z. However, due to the assumed link layer protocol characteristics, the APs are not able to obtain measurements of RSSI all at the same time. Moreover, the design of the protocol tells us that measurements from different sources never happen at the same time. For this reason, the observation vector is a single-entry vector, which will contain the measurement obtained at the current time step t_k. As a consequence of the eq.(5.4), the estimated measurement $\hat{Z}_k$ is given by:

$$\hat{Z}_k = \alpha - 5\beta \log \left((\hat{x}_k - a_{n_k})^2 + (\hat{y}_k - b_{n_k})^2 \right) \tag{5.28}$$

Note however, that this decision implies that the model to describe position and power relation has to change at every time step in order to include the position of the currently active AP $\Lambda_{n_k} = [a_{n_k}, b_{n_k}]^T$.

Since the EKF and the EKS require a linear model, we linearize the relation position-power by the jacobian matrix of eq.(5.28).

$$H_k = \left. \frac{\partial Z}{\partial X} \right|_{X=X_k} = -\frac{10\beta \ln^{-1}(10)}{(x_k - a_{n_k})^2 + (y_k - b_{n_k})^2} \begin{bmatrix} (x_k - a_{n_k}) & (y_k - b_{n_k}) & 0 & 0 \end{bmatrix} \tag{5.29}$$

Regarding the measurement noise, it is assumed zero mean gaussian distributed with covariance R. If the noise is not gaussian, the filter still works as an approximation to the optimality. Since the measurement vector is a single entry vector, R has to be defined as a one entry matrix of the form:

$$R = \sigma_p^2 \tag{5.30}$$

where σ_p^2 is the standard deviation of the noise present in the measurements of received power.

As we can see in the design of the filter, several parameters are unknown. They are the parameters α and β in eq.(5.28) and the standard variations $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ in eq.(5.27). In the coming sections we will study the performance of the EKF and EKS, modeled the way presented above.

5.5.2 Performance Analysis

In order to analyze the performance of the localization solutions based on the EKF and EKS, we have analyzed the influence of several of their design parameters. It was decided to use throughout this section the values stated in Table 5.2, unless something is said in contrary. The values of the Table 5.2 were found through an iterative process of manual tuning in order to determine the minimum PMDE in the first run of the squared trajectory. In the initial stage, we have taken the parameters from Table 5.2 and we run the EKF and the EKS for obtaining position estimations and subsequently an estimation for the trajectory. As we can see, in Fig. 5.6 and in Fig. 5.7, the estimation of the position is considerably noisy, what is due to the artificial noise that was introduced in eq.(5.2) into the evolution model (see speed estimation in Fig. 5.8 and Fig. 5.9 respectively). This behavior can be explained by the fact that in the design of the EKF, changes of direction are generally modeled by introducing an artificial noise in the process. Thus, the entire trajectory presents a noisier estimation, but on the other hand the filter is able to model changes in direction. If this would not be done, the EKF would try to estimate a straight line movement,

Table 5.2: Default parameters used in the design of the EKF and the EKS. Shaded row represents EKS specific parameters.

Parameter	Default Value
α (dBm)	-50
β (unitless)	2.5
$\sigma_{\ddot{x}}$ (m/s^2)	0.001
$\sigma_{\ddot{y}}$ (m/s^2)	0.001
σ_p (dB)	6
Initial State	true
Initial Covariance Matrix	Identity matrix
Lag Duration (Numb. Obs)	300

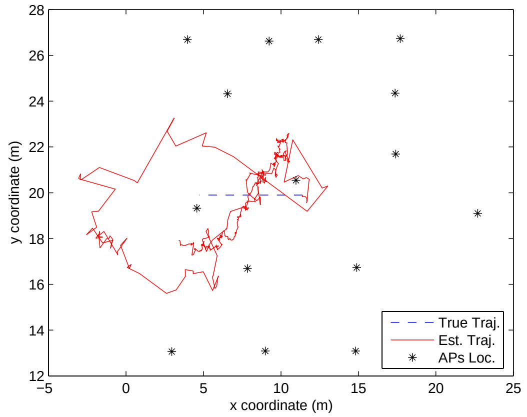


Figure 5.6: Estimation of the target path using the EKF, when the target follows the straight trajectory.

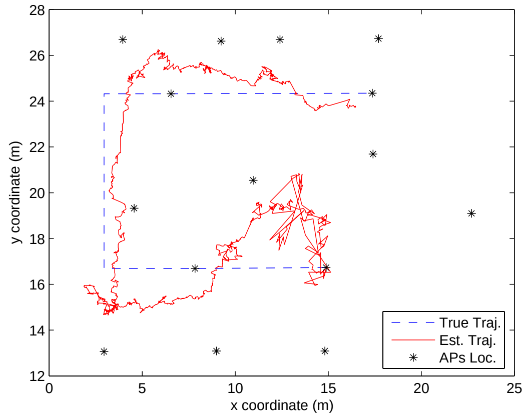


Figure 5.7: Estimation of the target path using the EKF, when the target follows the squared trajectory.

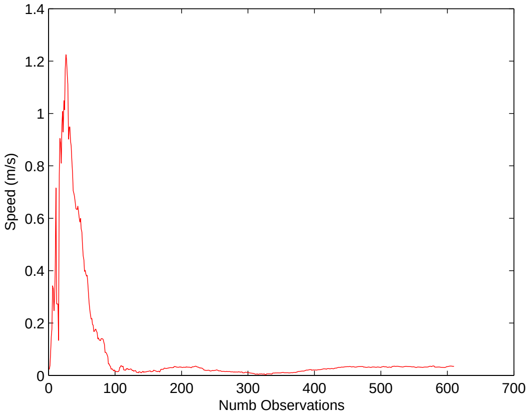


Figure 5.8: Estimation of the target velocity using the EKF, when the target follows the squared trajectory.

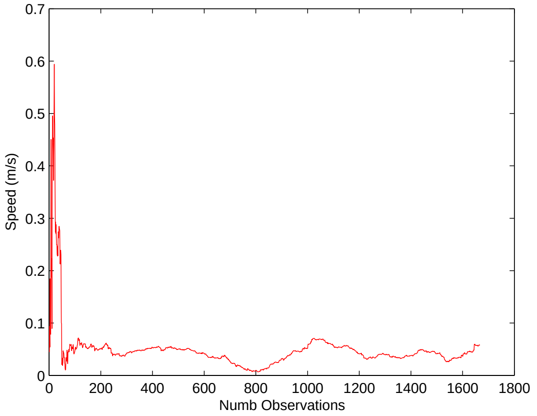


Figure 5.9: Estimation of the target velocity using the EKF, when the target follows the squared trajectory.

having long response times when changes in direction actually happen (see for instance Fig. 5.27 further in this section where this parameter is studied in more detail). A common solution to reduce this noise is to apply a EKS. The EKS, by using future measurements is able to better filter the noise of the estimations in an offline or in a online fashion with lag delay. This fact is clearly observed when Fig. 5.6 and Fig. 5.7 are respectively compared with Fig.5.10 and in Fig.5.11. Apart of some initial high noise, not only the position estimators have a considerably lower noise component, but also the speed has better estimators with lower noise (see Fig. 5.12 and Fig. 5.13).

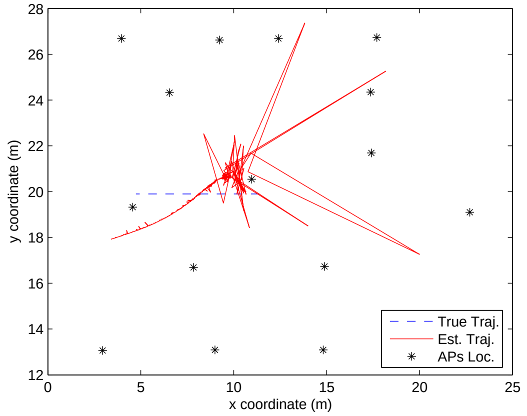


Figure 5.10: Estimation of the target path using the EKS, when the target follows the straight trajectory.

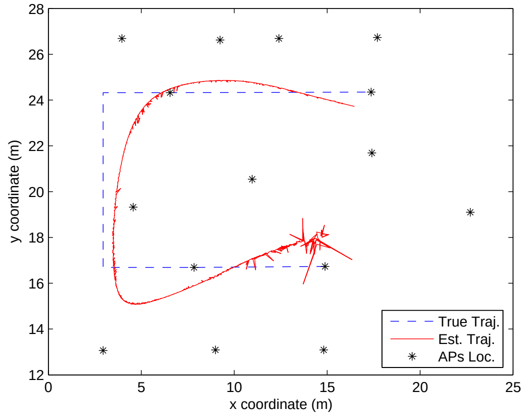


Figure 5.11: Estimation of the target path using the EKS, when the target follows the squared trajectory.

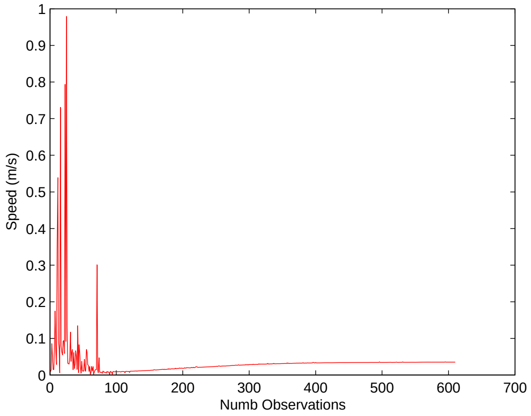


Figure 5.12: Estimation of the target velocity using the EKS, when the target follows the straight trajectory.

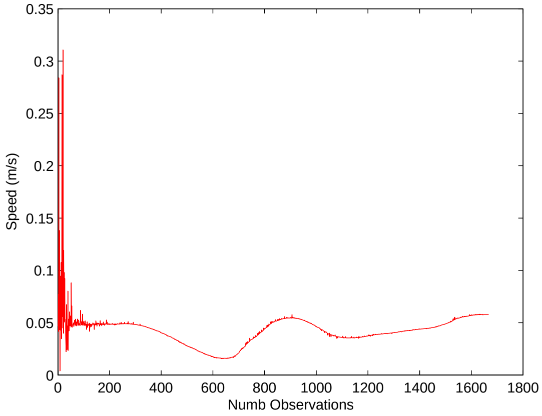


Figure 5.13: Estimation of the target velocity using the EKS, when the target follows the squared trajectory.

To measure the performance of the filtering and the smoothing solution, the 4 runs of the same experiment were evaluated according to the metrics defined in Sections 5.2. Inspecting the results

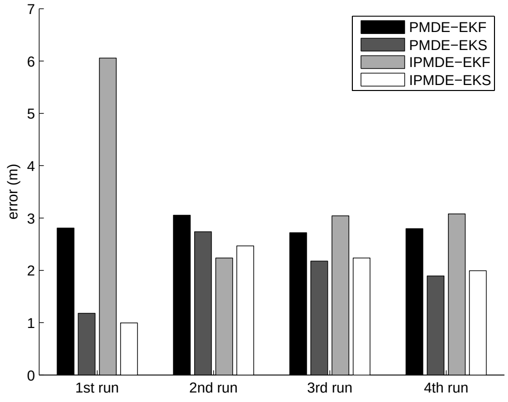


Figure 5.14: Performance of the estimation in the straight trajectory when using the EKF and the EKS. Performance according to the PMDE and the IPMDE metrics

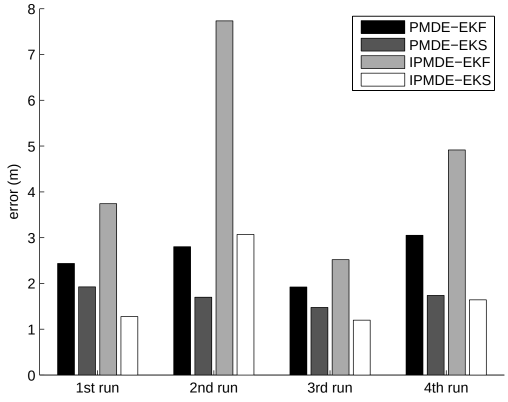


Figure 5.15: Performance of the estimation in the squared trajectory when using the EKF and the EKS. Performance according to the PMDE and the IPMDE metrics

presented in Fig. 5.14 and Fig. 5.15, it can be seen that for every case (except the second run of the straight trajectory) the performance of the smoother is better than the performance of the filter, whatever is the considered metric. This is explained by the simple fact that the EKS by using future information to update the current state, uses more information and subsequently it produces a better estimation for the target position. The disadvantages are that the estimation with a EKS cannot be done in real time and, also, the EKS itself presents a quadratic complexity. While comparing the metrics PMDE and IPMDE, we can see that the errors are not equivalent. The reason is that since the speed is commonly not estimated as a constant parameter, the PMDE metric tends to gradually approximate or gradually drift away from the true trajectory depending on the current value for the speed. On the other hand, since IPMDE depends on the length of the estimated trajectory as defined in eq.(5.8), it subsequently suffers with the noise on the position estimators. Thus, the performance of this metric is very dependent on the incremental sum of the errors on the estimators. Also, when comparing the performance of both straight and squared trajectory, one can see that generally the errors are worse for the squared trajectory. The reason is that since for the squared trajectory there is some uncertainty introduced by the changes of direction, the filter has additional problems to closely estimate position. When looking at the CDE metric in Fig. 5.16 and Fig. 5.17, it is clear that for all the cases, the smoother outperforms the filter. The reason is the same that provokes the aforementioned influence on the IPMDE metric, i.e. the dependency that the CDE metric has with eq.(5.8) and the subsequent dependency with the noise in the positions estimators. When directly comparing the performance according to CDE for both trajectories, we can clearly see that the straight trajectory has higher errors specially for the case of the filter. The reason is that since this metric is inversely proportional to the real covered distance (see eq.(5.6)), the longer the path is, lower tends the CDE metric to be.

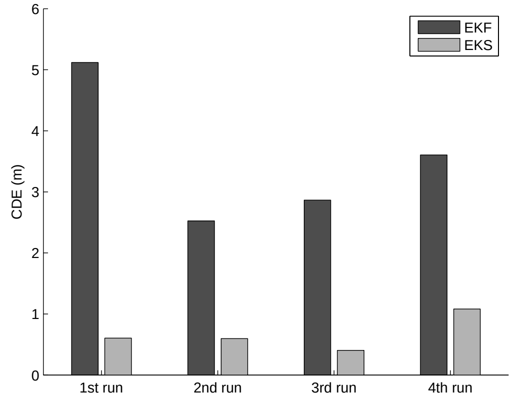


Figure 5.16: Performance of the estimation in the straight trajectory when using the EKF and the EKS. Performance according to the CDE metric.

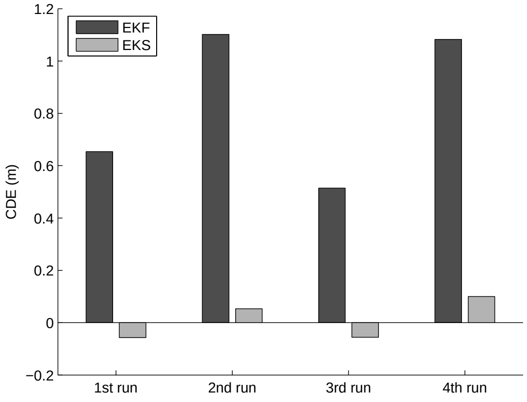


Figure 5.17: Performance of the estimation in the squared trajectory when using the EKF and the EKS. Performance according to the CDE metric.

At this point, in order to ease the reading of the further sections, Table 5.3 gives an overview of the problems to address concerning the EKF and EKS when applied to positioning in wireless networks.

One of the factors that we intend to evaluate in this section is the initial guess used during the design of the filter. In reality, the real initial state is very often unknown, meaning that assuming it known may be unrealistic. Consider now that the initial guess instead of being always the true one, is taken from one of the following positions: the origin of the coordinate system $(0, 0)$; close to the center of the test scenario $(12, 20)$; one of the four corners of the test scenario $(2, 12)$; $(22, 12)$; $(22, 27)$; $(2, 27)$. For the initial guess regarding the starting velocity it is assumed zero. As Fig.

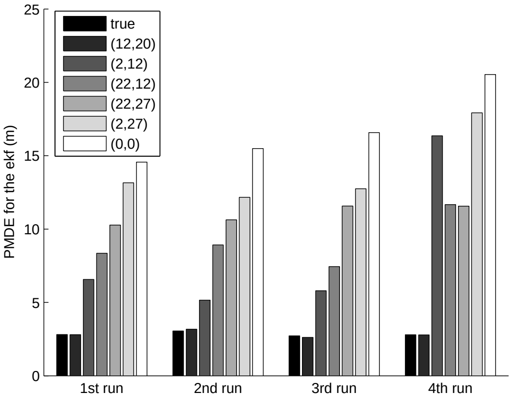


Figure 5.18: Performance of the EKF for several values of initial state for the straight trajectory.

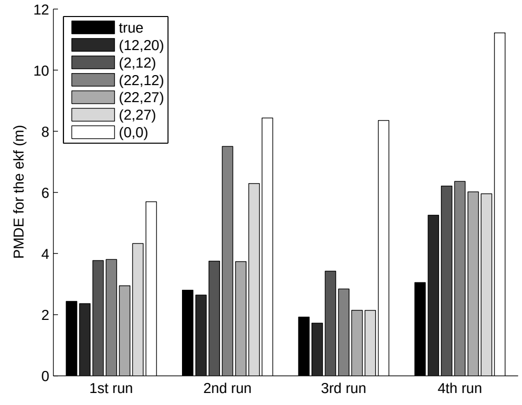


Figure 5.19: Performance of the EKF for several values of initial state for the squared trajectory

Table 5.3: Overview of the problems and possible solutions of EKF and EKS algorithms when applied to positioning applications.

Problem	Analysis	Solution
Initial guess for initial user dynamics X_0 and P_0	Current Section: Given that there is no a-priori knowledge on the user dynamics, X_0 and P_0 need to be guesses. The consequence is that this guess has a strong influence on the overall accuracy of the estimation algorithm.	Section 5.5.3: The process noise is increased in the beginning of the estimation as an indication of high uncertainty. Then in a recursive fashion the process is reduced along time.
Unknown propagation constants α and β	Current Section & Appendix D: Since the propagation constants are unknown, they shall be estimated either by a calibration phase or a more advanced solution.	Section 5.5.4: Joint estimation of the unknown parameters and the dynamics of the WD is performed.
Tuning of process noise $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$	Current Section: In order to model changes in direction, the process noise shall be increased. However, the result is that also the magnitude of the noise component in the final estimation gets higher.	Section 5.6: A multiple model approach is used in order to permit the modeling of changes in directions.
Noise statistics of the process noise $\mathbf{w}_k$	Current Section: The Kalman family of filters assume Gaussian noise. However assuming that the wireless channel produces RSS reading with Gaussian distribution is a rough approximation.	Section 5.7: The process noise is modeled based on statistical models of propagation and a PF is used for estimating the kinetics of the WD.
Lag time of the EKS	Appendix D: The lag time of the EKS is a tradeoff between processing time and estimation accuracy. Thus selecting an adequate value shall be studied.	Only analysis were done.
Speed of the WD	Appendix D: The speed of the WD influences the accuracy of the system, i.e. the higher the speed is, more complex is to track that WD due to the fact that the number of effective measurements per unit of covered distance is lower.	Only analysis were done.

5.18 and Fig. 5.19 show, the initial guess strongly influences the overall accuracy of the localization solution. As a consequence of its definition, the filter “believes” that the difference between the initial guess $\hat{X}_0$ and its true state X_0 is given by a Gaussian distribution with covariance Q (subsequently obtained from $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$). When the magnitude of the error introduced by the initial guess is higher than the process noise of the user dynamics, the filter presents convergence problems. Thus, the reason for this behavior is on the relative magnitude of the process noise of the user dynamics (i.e. $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$) in comparison to the error introduced by the initial guess (i.e. $\hat{X}_0 - X_0$).

Another result of interest is the influence of the path loss parameters on the performance of the data fusion algorithms. For analyzing this effect, the α and the β parameters are varied, once at a time, around the default values mentioned in Table 5.2. Equivalent results for the parameter β are presented in the Appendix D. As Fig. 5.20 and Fig. 5.21 show, the PMDE is higher, the higher the parameter α is. The reason is that by increasing the parameters α, the estimated physical length of the communication channel increases exponentially (see eq.(5.3)), what represents a strong bias introduced in the measurements (see example in Fig 5.22 and Fig 5.23). Contrarily, the lower the parameter α is, the lower is the destructive influence. The reason is that in this case, when α decreases, it approaches the order of magnitude of the measurements, meaning that distances tend to be very short and though the bias will not influence the result in the same way as in the case of high values of α.

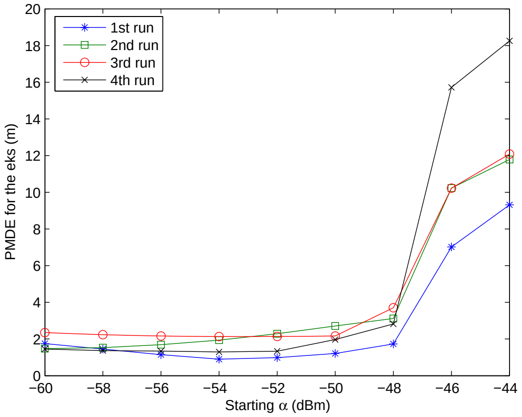


Figure 5.20: PMDE performance of the EKS in order to the α parameter for the straight trajectory.

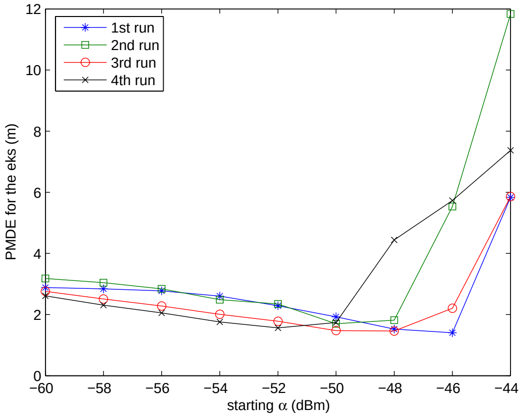


Figure 5.21: PMDE performance of the EKS in order to the α parameter for the squared trajectory.

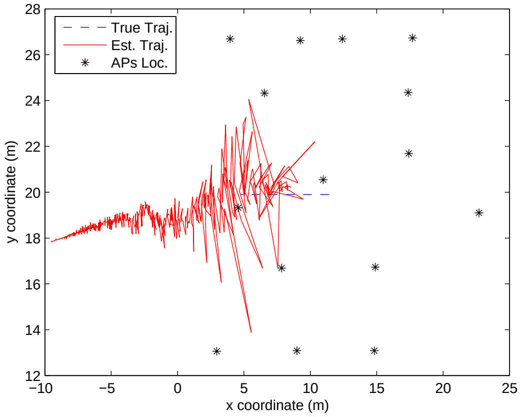


Figure 5.22: Example of path estimation for the straight trajectory when $\alpha = -42dBm$.

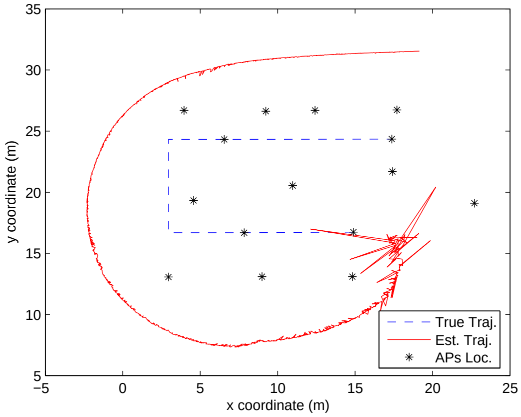


Figure 5.23: Example of path estimation for the squared trajectory when $\alpha = -42dBm$.

In order to understand the influence of the process noise added in the acceleration of x and y

coordinates, we have decided to plot the PMDE of the estimated position when the parameters $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ vary from $0m/s^2$ until $5 \times 10^{-3} m/s^2$. As we can see in Fig. 5.24 and Fig. 5.25, there is no equivalence in the trends of the curves of both trajectories. In fact this behavior is easily explained by inspecting the shape of the described trajectories. For the straight trajectory, the direction and the velocity are constant at every time, meaning that the more constrained the movement model is, the better the result gets, meaning that the best result can be obtained for $\sigma_{\ddot{x}} = \sigma_{\ddot{y}} = 0m/s^2$. On the other hand, the squared trajectory has two changes of direction, meaning that even though the norm of the speed is constant, its orthogonal components are varying. For the case of $\sigma_{\ddot{x}} = \sigma_{\ddot{y}} = 0m/s^2$ in the squared trajectory (see Fig. 5.27), the filter can hardly model the changes in direction, resulting in a worse estimation than in Fig. 5.11. Furthermore,

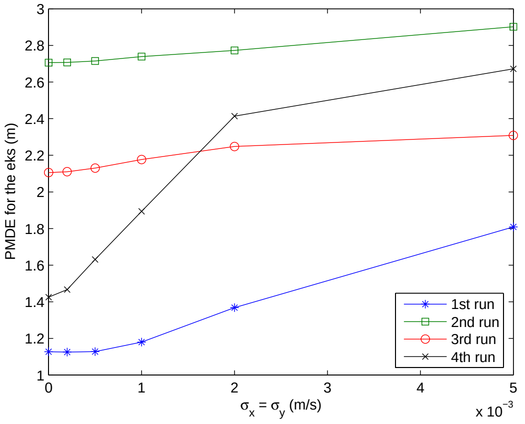


Figure 5.24: Performance, according to the PMDE, of the EKS with respect to the variance of the acceleration for the straight trajectory.

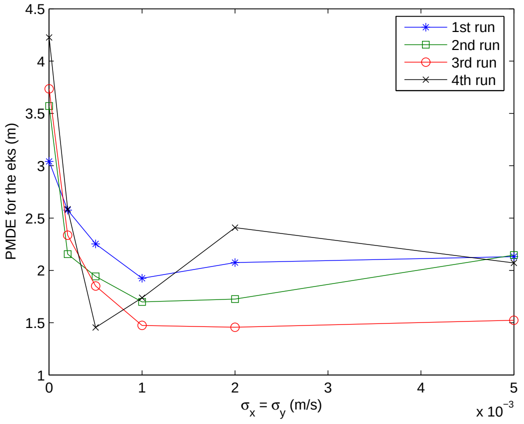


Figure 5.25: Performance, according to the PMDE, of the EKS with respect to the variance of the acceleration for the squared trajectory.

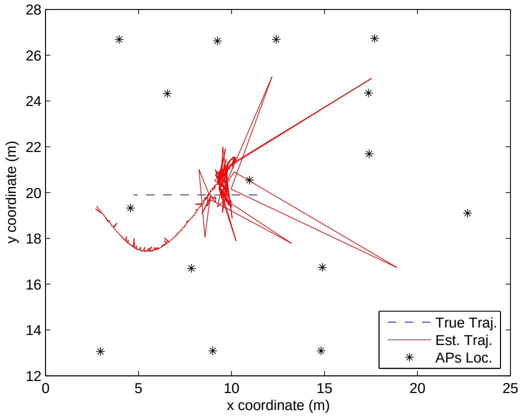


Figure 5.26: Example of the estimated straight trajectory when $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ are both equal to $5 \times 10^{-3} m/s^2$.

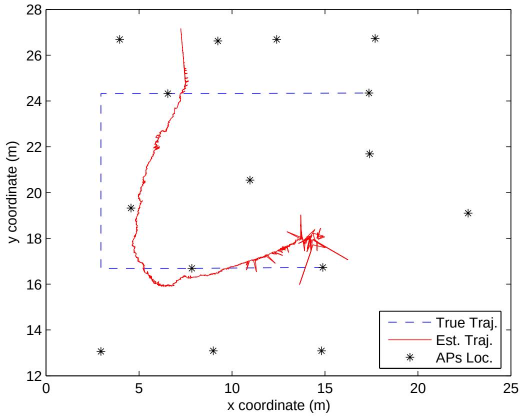


Figure 5.27: Example of the estimated squared trajectory when $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ are both zero.

when the process noise is increased by increasing $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$, the filter is gradually able to handled the changes in direction. Finally, when the noise exceeds a certain value, the filter starts to have too high freedom to estimate changes in direction even though they do not happen in the true trajectory (see Fig. 5.26).

5.5.3 Process Noise Adaptation for Solving Unknown Initial State Problem

As it was seen in the previous section, the initial guess strongly influences the performance of the system. For this reason, a solution for solving this problem is required. This section studies the performance of an adaptation scheme commonly used in applications where the EKF is chosen as the estimation algorithm: annealing optimization. The annealing optimization is a meta-algorithm that aims at solving problems of global optimization. In short, this solution attempts to evaluate a wide neighborhood of a determined state space and chose the state within the neighborhood which maximizes the probability to the system be in that state. In the scope of the present applications for positioning, the algorithm defines a upper bound threshold σ_{high}, a lower bound threshold σ_{low} and a decay parameter $0 < \varsigma < 1$. The value σ_{high} represents the maximum value of $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ that initializes the matrix Q_0 at time t_0, the value σ_{low} is the lowest value allowed for $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ and ς dictates how fast $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ decay from σ_{high} to σ_{low} along time. Thus:

$$\sigma_{\ddot{x}_k} = \max\left(\sigma_{low}, \varsigma\sigma_{\ddot{x}_{k-1}}\right) \qquad \text{where} \qquad \sigma_{\ddot{x}_0} = \sigma_{high} \tag{5.31}$$

where for $\sigma_{\ddot{y}_k}$ similar approach can be used. In order to obtain the process noise Q_k, the value of eq.(5.31) has to be included in eq.(5.27). In eq.(5.31), the initial threshold identifies the initial value for the process noise. Then, at every time step, by using eq.(5.31), the noise is gradually reduced according to the decay parameter ς. Once the process noise reaches the threshold σ_{low}, it keep that value for the rest of the estimation procedure. With this technique, the process noise can be initially increased, meaning that there is higher uncertainty in the state space. This fact will allow the EKF to augment the set of possible states at the beginning of the estimation and gradually reduce that set of states.

Using this simple solution one can see in Fig. 5.28 and Fig. 5.29 that it is possible to decrease the sensibility of the system to the initial guess. Comparing Fig. 5.19 and Fig. 5.29, which represent exactly the same setup but respectively without and with this optimization, we can see that: (i) for the true initial guess, the accuracy is slightly degraded; (ii) the localization solutions get more robust to an initial guess values different from the true one. However, as we can see in Fig. 5.28, for instance in the 4th run with initial guess $(2, 27)$, this optimization does not work. In some cases, the accuracy may be degraded, what is due to the considerable sensibility that the annealing method experiences to changes in its parameters. For obtaining these results we have used the values $\sigma_{high} = 0.5 m/s^2$, $\sigma_{low} = 0.001 m/s^2$ and $\varsigma = 0.975$. It shall be mentioned here that the choice of the values for these parameter is, in fact, system dependent and has influence in the final accuracy of the system.

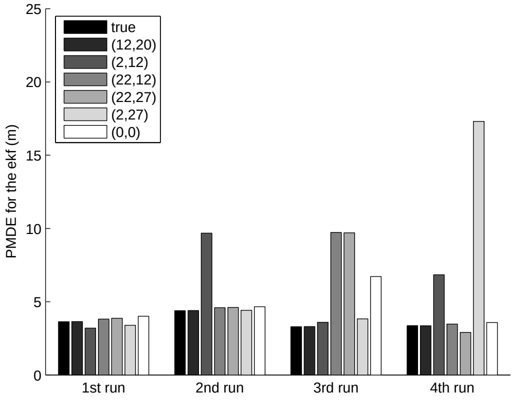


Figure 5.28: Performance of the EKF for several values of initial state for the straight trajectory when it is used process noise adaptation.

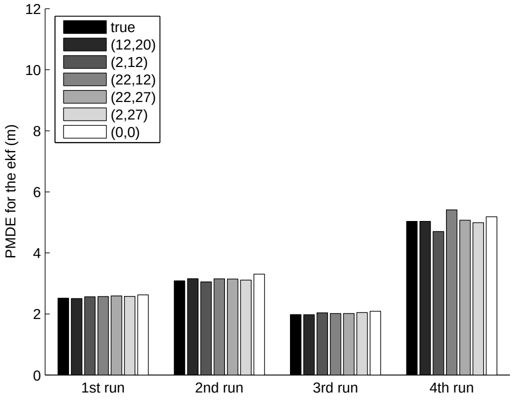


Figure 5.29: Performance of the EKF for several values of initial state for the squared trajectory when it is used process noise adaptation.

5.5.4 Joint Estimation of Position and unknown Parameters

Another problem of the EKF is that unknown parameters strongly influence localization accuracy. The straightforward solution for this problem is to proceed to a calibration phase, a training phase or a set of trial-and-error tests. A more advanced solution could be done by considering the unknown parameters as variables belonging to the hidden state space. In our setup, since the propagation parameters α and β (of eq.(5.28)) are unknown, the state space of eq.(5.25) shall be augmented to:

$$X_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k & \alpha_k & \beta_k \end{bmatrix}^T \tag{5.32}$$

As a consequence of the augmentation of the state space, also the transition and the process noise covariance shall be redesigned. Regarding the transition matrix, that in the ordinary EKF has the form of eq.(5.26), in the joint EKF, it must included the additional augmentation due to the parameters α and β. Since these parameters are constant along time, the transition matrix is unitary for the row/column entries corresponding to those parameters.

$$A_k = \left(\begin{array}{cccc|cc} 1 & 0 & T_k & 0 & 0 & 0 \\ 0 & 1 & 0 & T_k & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \tag{5.33}$$

Assuming that the noise in the parameters α and β is independent from the noise in the kinetics of the WD, the eq.(5.27) is augmented by simply including the variance of the noise in the respective diagonal entries.

$$Q = T \left(\begin{array}{cccc|cc} \sigma_{\ddot{x}} T^2/3 & 0 & \sigma_{\ddot{x}} T/2 & 0 & 0 & 0 \\ 0 & \sigma_{\ddot{y}} T^2/3 & 0 & \sigma_{\ddot{y}} T/2 & 0 & 0 \\ \sigma_{\ddot{x}} T/2 & 0 & \sigma_{\ddot{x}} & 0 & 0 & 0 \\ 0 & \sigma_{\ddot{y}} T/2 & 0 & \sigma_{\ddot{y}} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \sigma_\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\beta \end{array} \right) \qquad (5.34)$$

where σ_α and σ_β represent respectively the standard deviation of the noise component of the parameters α and β. Additionally, also the H matrix of eq.(5.29) has to include two additional columns, representing the derivatives of the received power with respect to the α and the β parameters respectively. These parameters are set to a value some orders smaller than the magnitude of the corresponding parameters. The reason is to give some flexibility to those parameters to have some variations along time envisaging a contribution to the convergence of those parameters. As Fig. 5.30 and Fig. 5.31 show, when comparing with 5.20 and Fig.5.21, the joint estimation of parameters and states gives an increased performance to the system in terms of accuracy. However, the systems gets instable for some of the runs resulting in a divergence of the estimators. As in the 2nd run of the squared trajectory, the estimation drifts away from the true trajectory. For this reason that run was not included in the resulting plot of Fig. 5.31. Similarly, for the straight trajectory, the 1st and 3rd run have diverged for some subset of starting values of α.

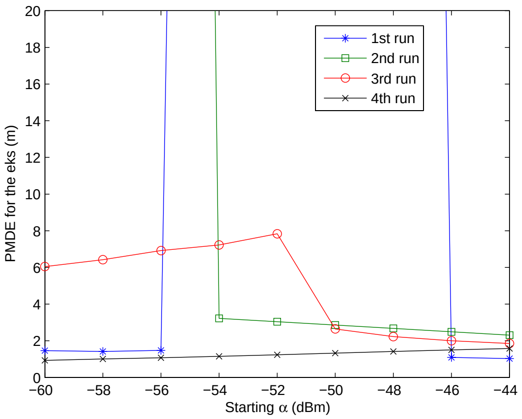


Figure 5.30: Joint parameter and state estimation of an EKS for the straight trajectory

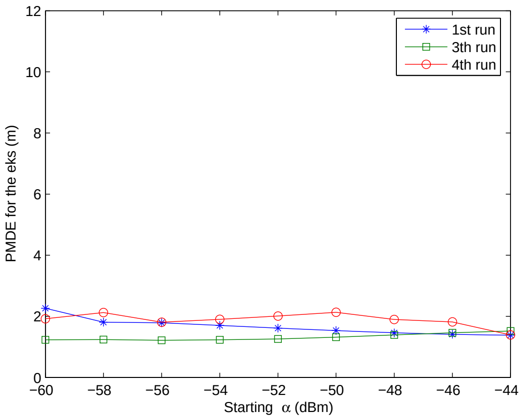


Figure 5.31: Joint parameter and state estimation of an EKS for the squared trajectory

Regarding the parameter β the results were not plotted since they do not give additional information for better understanding the behavior of the system.

5.6 Modeling Maneuvers Using Multiple-Models

As it was show throughout Section 5.5, one of the main problems of the EKF is to handle maneuvers, i.e. abrupt changes in the direction of the trajectory. A possible solution for solving the problem relies on the use of multiple models running in parallel, e.g. the SEKF framework (see Section 3.2.2). After this multiple estimation, the combination of the estimators produced by the different models have to be associated in a single and final estimation for the current state. This data association is not, in fact, straightforward and depends on the application.

5.6.1 Design of the Filters used in the SEKF Framework

In the proposed framework we have decided to implement four parallel filters. Each one of the filters models a movement along each one of the four cardinal directions: North, South, East and West. The correspondence of these filters and the reference of the coordinate systems is that: North is equivalent to positive y and East the positive x coordinates. Each one of the filters assumes constant velocity along the coordinate that it is modeling and assumes no transversal component. Due to this reason, we can decrease the size of the state space X in eq.(5.25), by simply considering both x and y coordinates and the norm of the speed v. The direction is indirectly modeled by the matrix A_k of each one of the filters. Thus the state space to be used by all the filters is:

$$X_k = \begin{bmatrix} x_k & y_k & v_k \end{bmatrix}^T \tag{5.35}$$

The transition matrices for each one of the filters can be generally modeled as:

$$A_k = \begin{pmatrix} 1 & 0 & T_k \sin(\theta) \\ 0 & 1 & T_k \cos(\theta) \\ 0 & 0 & 1 \end{pmatrix} \tag{5.36}$$

where $\theta = \{0, \pi/2, \pi, 2\pi/3\}$ respectively for the North, East, South and West filters. Regarding the covariance matrix it can be written as (similarly to what happened in eq.(5.27)):

$$Q_k = T_k \begin{pmatrix} \sin^2(\theta)\sigma_{\dot{v}}^2 T_k^2/3 & 0 & \sin^2(\theta)\sigma_{\dot{v}}^2 T_k/2 \\ 0 & \cos^2(\theta)\sigma_{\dot{v}}^2 T_k^2/3 & \cos^2(\theta)\sigma_{\dot{v}}^2 T_k/2 \\ \sin^2(\theta)\sigma_{\dot{v}}^2 T_k/2 & \cos^2(\theta)\sigma_{\dot{v}}^2 T_k/2 & \sigma_{\dot{v}}^2 \end{pmatrix} \tag{5.37}$$

where the parameters $\sigma_{\dot{v}}$ is the standard deviation of the error in the acceleration.
Concerning the measurement model, all the filters follow the same design decisions taken in the normal EKF in Section 5.5.

5.6.2 Switching Scheme for Filter Selection

When multiple models are included in an estimation framework, the association of the filters needs to be addressed as well since it is not a straightforward problem. As stated in literature [Mur88], there are several conceptual alternatives of associating the filters. For the present application, we are interested in selecting the filter that best performs according to a pre-determined metric. This model selection is also known as Multiple Hypothesis Tracking [BS88].

Switching Scheme Based on the Error Covariance Matrix

A possible switching scheme can be done by evaluating the commonly used performance metric of each one of the parallel Kalman Filters, the trace of the covariance matrix. The switching scheme is: at each time the filter with the lowest metric is selected as the best filter and its estimation is considered as the final estimation. After implementing this scheme, we can see (in Fig. 5.32) that the output of the SEKF does not produce an adequate estimation for the given trajectory. In order to understand the reason for such behavior we have decided to plot the sequence of filter selection (Fig. 5.33). As we can see from Fig. 5.33, the problem is that the selection of the filter is constantly switching and thus producing an undesirable output. Note that in Fig. 5.33 the switching of filters is so frequent that often it is not possible to distinguish segments where the same filter is selected along several estimations.

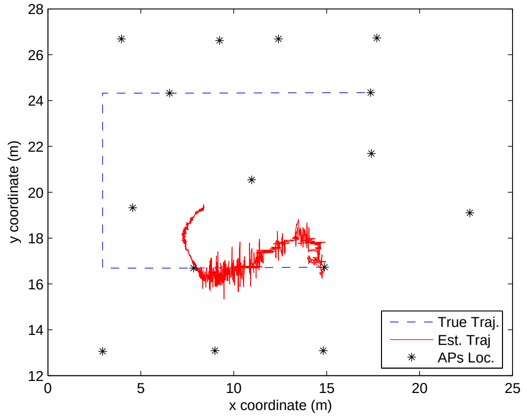


Figure 5.32: Single run of the SEKF in the squared trajectory using a switching scheme based on the error covariance matrix.

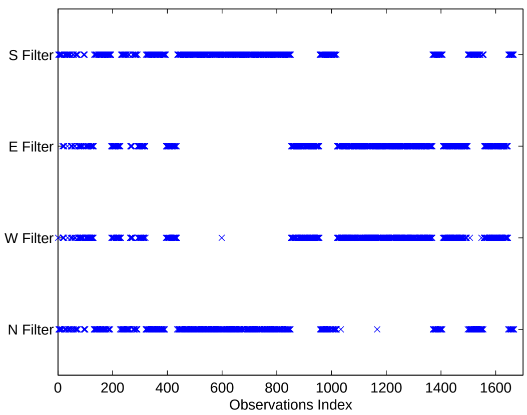


Figure 5.33: Model selection performed by a switching scheme based on the error covariance matrix.

Switching Scheme Based on the Innovation Process

Attempting to obtain a switching scheme which gives a better performance than the previous one, we have decided to reformulate the definition of best filter. The best filter is the one which produces the lowest absolute expected value of the innovation process explained in Section 3.1.1. This definition is supported in the assumption that there exists a filter that has an output closer to the true trajectory, than all the others. Subsequently, that filter is expected to show an expected value for the innovation process closer to zero. In order to determine the expected value of the innovations (for simplicity we call it bias), we can use a common filtering framework, given that the covariance matrix of the innovation process is known. From the Kalman filtering equations, it is possible to see that the innovation process is given by eq.(3.13) while its covariance matrix is given by eq.(3.14).

In order to determine the evolution of the bias (i.e. the expected value of the innovation process) along time, we design another filter which determines the expected value for the innovation based on the covariance matrix S (in eq.(3.14)) of each one of the filters. Thus, we define the switching scheme as a filter with state space as the innovation bias $b^{(i)}$ to be estimated, observations as the innovation residuals $\tilde{Z}_k^{(i)}$ and the respective covariance matrices $S_k^{(i)}$ for each one of the models. Since in the current application, the observations are scalars, the variables $\tilde{Z}_k^{(i)}$ and $S_k^{(i)}$ are scalars as well. This permits the filter to be defined as:

$$X_k^* = \begin{bmatrix} b_k^{(1)} & b_k^{(2)} & b_k^{(3)} & b_k^{(4)} \end{bmatrix}^T \tag{5.38}$$

$$Z_k^* = \begin{bmatrix} \tilde{Z}_k^{(1)} & \tilde{Z}_k^{(2)} & \tilde{Z}_k^{(3)} & \tilde{Z}_k^{(4)} \end{bmatrix}^T \tag{5.39}$$

$$Q^* = 0 \tag{5.40}$$

$$R_k^* = \begin{pmatrix} S_k^{(1)} & 0 & 0 & 0 \\ 0 & S_k^{(2)} & 0 & 0 \\ 0 & 0 & S_k^{(3)} & 0 \\ 0 & 0 & 0 & S_k^{(4)} \end{pmatrix} \tag{5.41}$$

Based on the previous definitions and given that biases are assumed to be constant in time, we can rewrite the Kalman filter equations and obtain a simplified system of equations:

$$K_k^* = P_{k-1|k-1}^*(P_{k-1|k-1}^* + R_k^*)^{-1} \tag{5.42}$$

$$X_{k|k}^* = X_{k-1|k-1}^* + K_k^*(Z_k^* - X_{k-1|k-1}^*) \tag{5.43}$$

$$P_{k|k}^* = (I - K_k^*)P_{k-1|k-1}^* \tag{5.44}$$

Based on this equation, the filter selected at time k is the one that minimizes the estimated bias. Once the filter is selected, its estimation is used as the overall estimation of the SEKF and further used as input in the following iteration for every filter.

$$i = \arg\max_l \left\{ b_k^{(l)} \quad : \quad l = 1, 2, 3, 4 \right\} \tag{5.45}$$

This decision implies that in Fig. 3.2, the history of the switching scheme concerns only one unit of time into the past.

After implementing the aforementioned switching scheme for determining the best filter, we have obtained better results than with the primarily explained scheme. As we can see in Fig. 5.34, the current switching scheme gives an estimation for the trajectory that is better than the equivalent in Fig. 5.32. When looking at Fig. 5.35, which represents the model selection, we can see that in fact

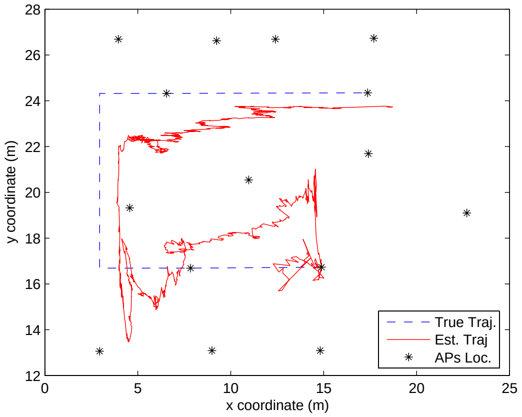


Figure 5.34: Single run of the SEKF in the squared trajectory using a switching scheme based on the innovation process.

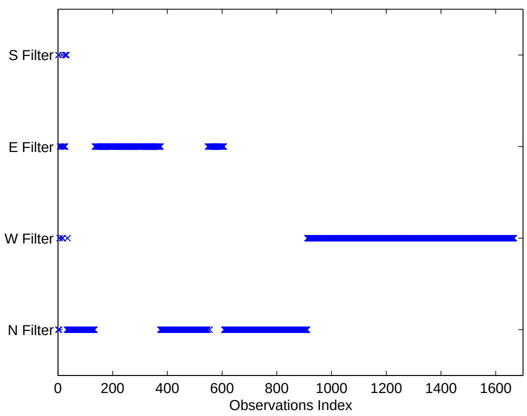


Figure 5.35: Model selection performed by the switching scheme using the innovation process.

there is a more accurate decision regarding the model than what happens in Fig. 5.33. Apart from some errors at the beginning of the estimation we can see that the switching scheme has chosen the sequence East, North and West filters. It is important to notice that the first and third segments of the trajectory are exchanged. This is due to the fact that East/West models are symmetric (as well as North/South), meaning that East model is equivalent to the West model with symmetric value for the velocity (see eq.(5.36) and eq.(5.37)).

5.6.3 Performance of the Multi-Model Filters

For analyzing the performance of the multi-model framework and compare it with the previously presented solution using an EKF, we have run the SEKF algorithm for the 4 runs of each trajectory. When we compare the SEKF in Fig. 5.36 and Fig. 5.37 and the EKF in Fig. 5.14 and Fig. 5.15,

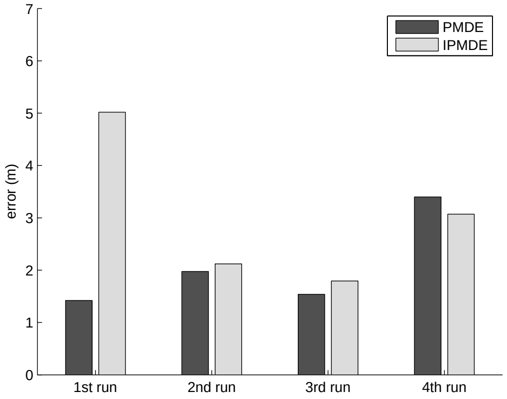


Figure 5.36: PMDE and IPMDE performance of the SEKF for straight trajectory

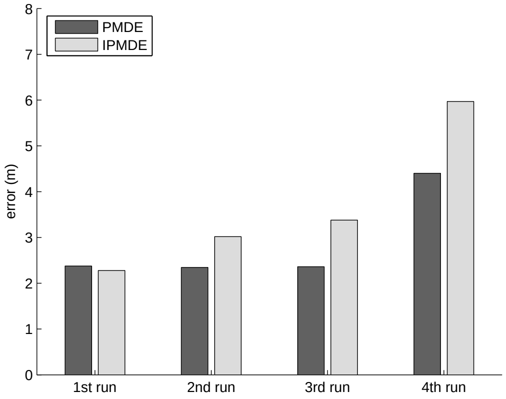


Figure 5.37: PMDE and IPMDE performance of the SEKF for squared trajectory

we can see that, either in terms of PMDE or IPMDE, the SEKF does have a better performance for some of the runs, such as runs 1, 2 and 3 of the straight trajectory, but it does not have reasonably better performance in the remaining cases. An important drawback of the SEKF is that it increases the computational complexity due to the fact that 4 filters plus the filter embedded in the switching scheme are continuously running in parallel.

5.7 Particle Filters for Modeling the Measurement Noise Provoked by the Propagation Effects

As we have seen in Section 3.1.3, a strong candidate for replacing the EKF in the current application is the PF. The latter, in opposite of the EKF is capable of modeling any kind of process statistics given that the PDF of the hidden process uncertainty is known (the noise $\mathbf{v}$ in eq.(5.4)). By using an approach that propagates samples of the involved distributions along time, the PF commonly results in better performance for non-Gaussian processes. For this reason we intend to use one of the available implementations (Section 3.1.3) [GSS93] of the particle filter and compare its performance with the one from the EKF.

5.7.1 Modeling the Uncertainties in the Filter

The Uncertainties in the system may be identified as: process noise; measurement noise; initial state uncertainty. For the initial state, we will simply assume that it is known; regarding the process noise, we will simply model it as the Gaussian process defined for the EKF in Section 5.5; concerning the measurement noise, we will assume the process as given by non-Gaussian distribution obtained from a mathematical formulation of the probabilistic behavior of the propagation effects. Thus, modeling the noise is possible by following one of the following alternatives: proceed to a pre-calibration phase; use advanced computer simulation tools given information about the scenario; approach the problem in a theoretical framework. The first alternative of pre-calibrating the system has the advantage that the noise statistics are closely approximated. However, this procedure can be complex and time-consuming to accurately run without subsequent external destructive effects, e.g. interference or body shadowing effect. Moreover, due to the dynamic behavior of the communication channels, the calibrated information can be quickly outdated and subsequently harm the proper operability of the system. On the other hand, by using advanced computer simulations capable of simulating channel dynamics, this problem can be solved. The drawback is that this tools can be considerably expensive and require technical supervision. The last solution relies on describing the channel characteristics in a theoretical and probabilistic approach. This approach, may not be the most accurate, but does not suffer to the same disadvantages as the other candidate approaches and at the same time it is more realistic than a gaussian distribution.

To model the dynamics of the channel in a probabilistic framework, we will consider some of the most common propagation effects present in common narrow-band communication systems [Gol05]:

- Attenuation - due to the distance between transmitter and receiver.
- Shadowing - due to the obstacles existent along the communication in an indoor scenario.
- Multipath - due to the several paths that the signal can travel along, from the transmitter to the receiver.

Note that in this work, body shadowing is not considered.

In a Rayleigh channel we know that if either the device is moving with low speed or the device is stopped but the scenario, surrounding it, is dynamic, the received signal envelope has a Rayleigh distribution. Since the relation between the signal envelope and the received power z' (in linear unities) is quadratic, the Rayleigh distribution for the envelope results in an exponential distribution for the received power z' [Gol05].

$$\mathrm{pr}(z'|m') = \frac{1}{m'}e^{-\frac{z'}{m'}} \quad [mW] \tag{5.46}$$

where m' (also in linear units) is the expected value for the exponential distribution. Translating into logarithmic units, eq.(5.46) get in the form:

$$\mathrm{pr}(z|m) = \frac{\ln(10)}{10}10^{\frac{z-m}{10}}e^{-10^{\frac{z-m}{10}}} \quad [dBm] \tag{5.47}$$

where $z = 10\log(z')$ and $m = 10\log(m')$.

On the other hand, for a fixed position of the transmitter, if we measure the mean received power at many locations that have the same distance to the transmitter, we expect m to have a normal distribution in dBm units.

$$\mathrm{pr}(m|\bar{z}) = \frac{1}{\sigma_p\sqrt{2\pi}}e^{-\frac{(m-\bar{z})^2}{2\sigma_p^2}} \quad [dBm] \tag{5.48}$$

where $\bar{z}$ corresponds to the path loss expression in eq(2.1) presented in the background Section 2.3.1. The parameter σ_p defines the standard deviation of the variable m.

Given previous definitions, the problem can be defined now as obtaining the probability density of the received power z given that the path loss value $\bar{z}$ is known. Mathematically, this means

$$\mathrm{pr}(z|\bar{z}) = \int_{-\infty}^{\infty} \mathrm{pr}(z|m,\bar{z})\mathrm{pr}(m|\bar{z})dm \tag{5.49}$$

In eq.(5.49), the term $\mathrm{pr}(z|m,\bar{z})\mathrm{pr}(m|\bar{z})$ turns into $\mathrm{pr}(z|m)\mathrm{pr}(m|\bar{z})$ due to the fact that once that m is known, $\bar{z}$ gives no additional information for the received power z. Then, applying eq.(5.47) and eq.(5.48) into eq.(5.49) and do the transformation of variables $\ell = m - \bar{z}$ and $\mathbf{v} = z - \bar{z}$, we can obtain:

$$\mathrm{pr}(\mathbf{v}) = \frac{\ln(10)}{10\sigma_p\sqrt{2\pi}}\int_{-\infty}^{\infty} 10^{\frac{\mathbf{v}-\ell}{10}}e^{-10^{\frac{\mathbf{v}-\ell}{10}}-\frac{\ell^2}{2\sigma_p^2}}d\ell \tag{5.50}$$

Note that the noise $\mathbf{v}$ in eq.(5.50) corresponds to the additive noise of eq.(5.3). As we can see from eq.(5.50), the distribution is not trivial due to the fact that it is not possible to obtain a closed-form solution for the involved integral. For this reason an approximation to the integral was performed by using numerical methods. Figure 5.38 shows an example of the distribution of eq.(5.50). As we can see from Fig. 5.38, the distribution is not gaussian as previously assumed in Section 5.5.

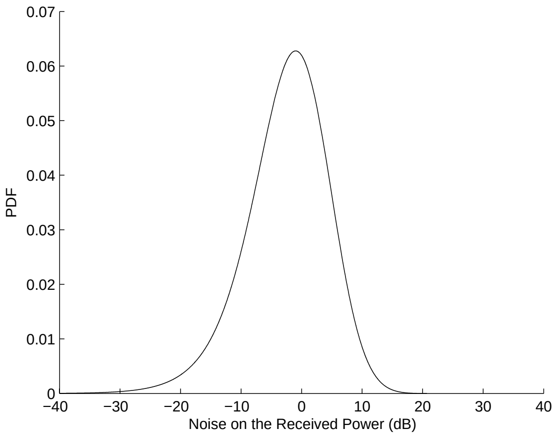


Figure 5.38: Probability Density Function for the measurement of the received power for the parameters $\sigma_p = 4dBm$

5.7.2 Initial Design and Testing of the Particle Filter

In order to perform a comparison with the previously proposed solution using an EKF, the filter was designed using the same parameters stated in Table 5.2. To these parameters we have included the additional PF specific parameters in Table 5.4. Regarding the distribution of the measurement noise we have used the one of eq.(5.50). Since eq.(5.50) is independent on the mean transmission power dictated by the path loss equation, we have numerically calculated the distribution. This numerical calculation, performed a single time in the initialization of the algorithm, resulted in a double vector with correspondence between noise attenuation values and the corresponding PDF values. Running the PF algorithm with the parameters of Table 5.4, we have noticed that the resultant trajectory estimation was strongly dependent of the initialization of the vector of particles. Given that the only difference is the seed value used in the random generation of particles which approximate eq.(5.50), the results are presented in Fig. 5.39. The problem, as exposed in [AMG$^+$02], is that the

Table 5.4: Parameters used in the design of the PF. Shaded row presents the PF specific parameters.

Parameter	Default Value
α (dBm)	-50
β (unitless)	2.5
$\sigma_{\ddot{x}}$ (m/s^2)	0.001 (later 0.01)
$\sigma_{\ddot{y}}$ (m/s^2)	0.001 (later 0.01)
σ_p (dB)	6
Numb. Particles (Numb. Obs)	5000

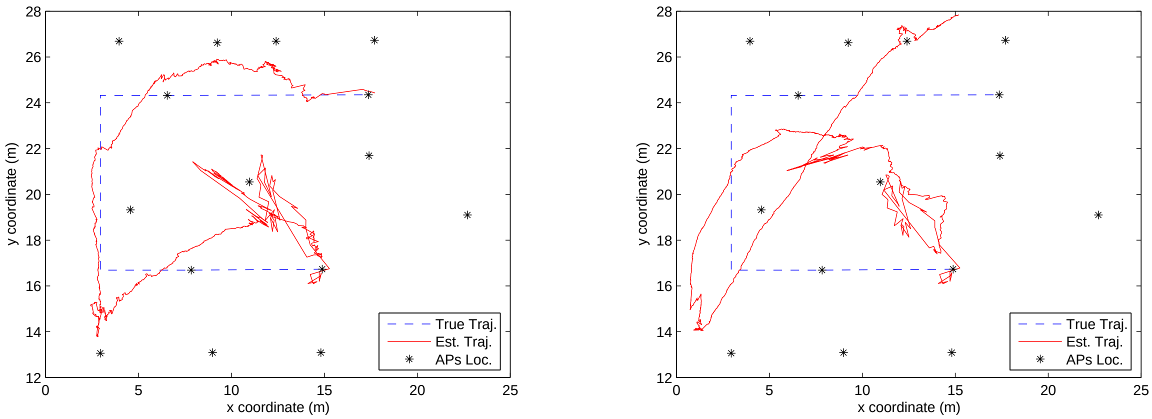


Figure 5.39: Two independent runs of trajectory estimation with $\sigma_{\ddot{x}} = 0.001m/s^2$ for the PF. The only difference between the two setups is the seed used in the generation os particles in the PF.

low magnitude of the process noise implies that the number of particles needs to be prohibitively high in order to avoid the particles to collapse in a single one after some few iterations. Thus, since for each different seed, the sequence of generated particles is different, the output of the filter may surely show, with higher or lower evidence, that random behavior. In our present case the influence is so big that even the trends of the trajectory change with each estimation. A solution for the problem could be to increase the number of particles. The consequence is that the filter could result in a computationally inefficient solution and thus, not interesting. Another solution is to primarily increase the process noise in order to decrease the tendency for the particles to collapse in a single one; secondly proceed to a post processing strategy where the noise would be canceled out from the estimated trajectory. Based on the latter proposed solution, we have decided to plot the same setup as in Fig. 5.39, but now using a standard deviation for the process noise that is $\sigma_{\ddot{x}} = 0.01$, i.e. 10 orders of magnitude higher than the case presented in Fig. 5.39. As we can visually see from Fig. 5.40, the estimated trajectory is not so sensitive to the randomness inherent to the PF, but on the other hand it is much noisier. As we can see, the collapsing of particles in a single one can be solved by increasing the process noise.

5.7.3 Applying the Sequential Double Exponential Smoothing

To solve the problem of noisy estimation of the trajectory, we could use a particle smoother. The problem with this approach is that smoothing increases enormously the need of processing, what in a particle type of algorithm may result in an untractable solution. For that reason, we have decided to look into time series and make use of one backwards DES as it was previously presented in the background Section 3.3.2. Due to the propagation of information represented by the forgetting factor in the DES algorithm, the DES time series is considered a smoothing algorithm. To better understand the impact of such algorithm we have plotted the same result as in Fig. 5.40, but post-processing the data with the DES approach. In Fig. 5.41, we can see that

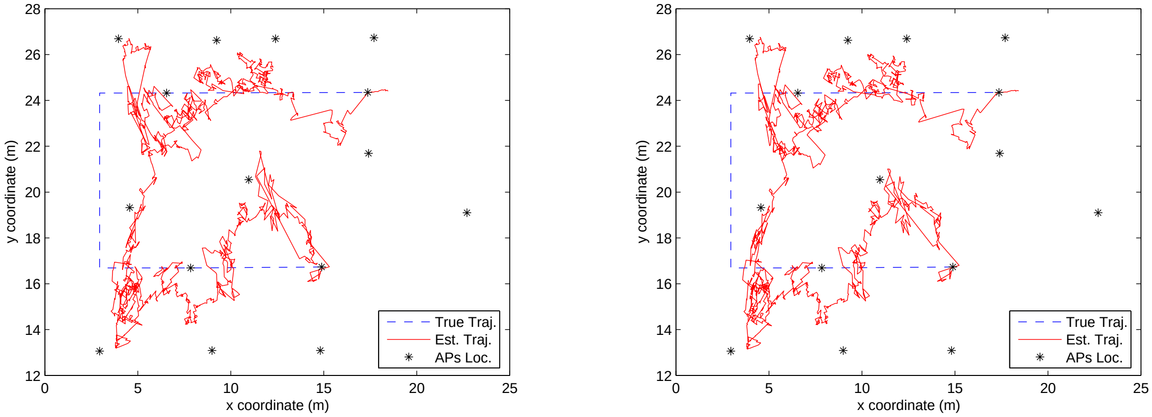


Figure 5.40: Two independent runs of trajectory estimation with $\sigma_{\ddot{x}} = 0.01m/s^2$ (i.e. 10 times higher than in Fig. 5.39) for the PF. The only difference between the two setups is the seed used in the generation os particles in the PF.

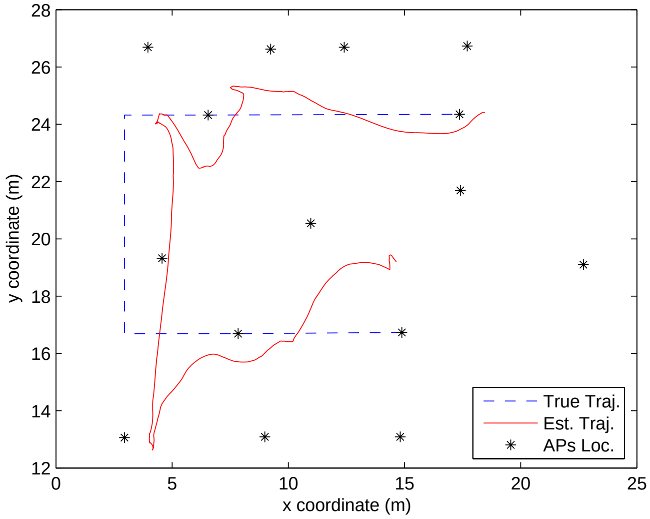


Figure 5.41: PF coupled with a DES time series.

with the forgetting factors $\gamma_s = \gamma_b = 0.01$, the result is considerably less noisier. Since the present solution is much more efficient than a particle smoothing algorithm regarding the computational effort of the involved calculations, and still very few dependent on the randomness of the particles, it is the elected solution for comparing with the EKF solution. However, one must bear in mind that direct comparison between this PF/DES and the EKF solution may not be fair in terms of the IPMDE metric. The reason is that the DES approach used to post-process the output of the PF, cancels out most of the noise in the estimators, what EKF by itself is not able to do because of the artificial nois introduced in the system to model changes of direction.

5.7.4 Performance Results

In order to study the performance of the current PF based solution, we have plotted the PMDE and the IPMDE metrics. Figures 5.42 and 5.43 show the performance for the four runs of the straight and the squared trajectories, respectively. When comparing the PMDE metric for the present filter and the EKF (Figures 5.14 and 5.15), we can see that the PF generally presents a slightly better performance than the EKF. However, the computational burden of using the PF was clearly verified during the data analysis. In opposite, when the EKS is compared with the present approach, it is clear to see that the EKS has a better performance. An equivalent smoothing solution could have been implemented which would expectably present better performance than the EKS. This alternative has been discarded due to the high computational effort already involved in the PF approach. Regarding the IPMDE metric, we can see that the PF clearly outperforms the rival EKF. This is due to the fact that the PF is followed by a DES, which cancels out most of the noise of the PF output and subsequently reduces the length of the estimated trajectory as it is defined in eq.(5.8). As we can see in Fig. 5.44 and Fig. 5.45, the length of the trajectory if considerably reduced when the PF is compared with the EKF (Figures 5.16 and 5.17).

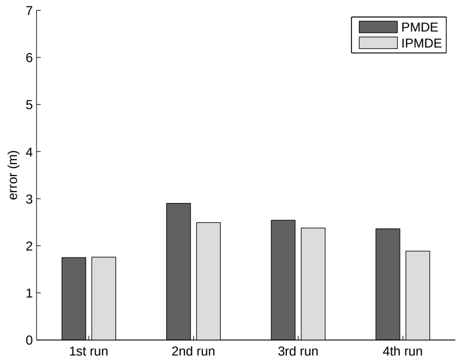


Figure 5.42: PMDE for the PF coupled with a DES when estimating the straight trajectory.

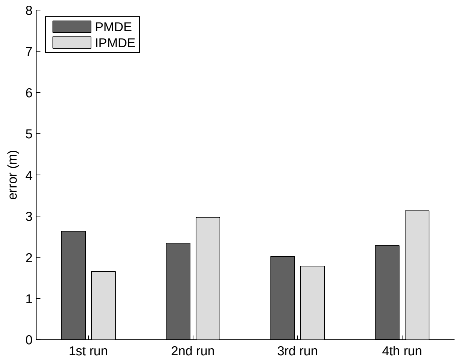


Figure 5.43: PMDE for the PF coupled with a DES when estimating the squared trajectory.

5.8 Conclusions

In this chapter, the problem of wireless tracking was studied in an experimental setup of a short-range network. The chosen technology was Bluetooth and the source of positioning information was the received power measured in the communication links. The algorithms for performing positioning were the EKF, the SEKF and the PF for filtering in real time and the EKS for smoothing. For measuring performance, the mostly used metric was the PMDE, which is defined as a weighted mean of the distances between the sequence of all the estimated positions and the true positions within the considered trajectories.

For every tested scenario, the EKS has a higher performance than the EKF, however it has also

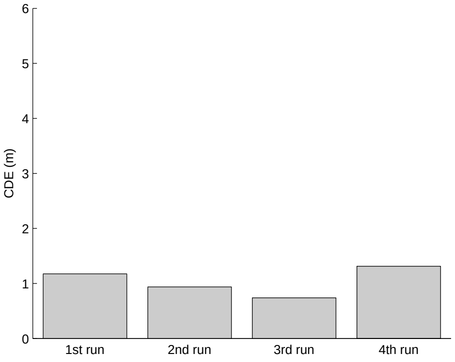


Figure 5.44: CDE for the PF coupled with a DES when estimating the straight trajectory.

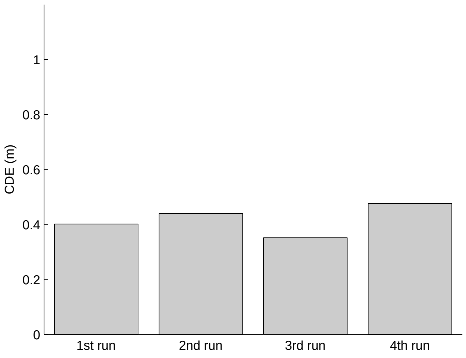


Figure 5.45: CDE for the PF coupled with a DES when estimating the squared trajectory.

higher complexity, i.e. squared complexity in contrast to the linear complexity of the EKF. Additionally, the length of the trajectory is more accurately estimated by an EKS than an EKF approach. The Extended Kalman approaches have shown strong dependency on the setup parameters of the involved evolution and observation models. The artificial noise added to the hidden process was shown as an alternative for estimating jumping processes, such as changes in direction. The consequence of this approach is that the final estimation of the trajectory is noisier. Thus, it is necessary to reach a tradeoff, between flexibility on modeling changes in direction and noisy trajectory estimations. An alternative to the artificial process noise, is to model changes in direction as a jumping process. The multi-model framework, e.g. the SEKF, was used and it has shown higher performance then the EKF for some of the tested scenarios, and similar performance for some others. Thus, given the higher computational burden of a multi model approach, the EKF may be preferred.

Regarding the propagation parameters used in the observation model, it was verified that they have an enormous impact in the accuracy of the trajectory estimation. The filtering approaches were shown to be extremely sensitive to these parameters. By using a joint estimation, we have shown that this influence can be reduced at the price of instability, i.e. for some cases the estimated trajectory greatly diverges from the true one. An additional parameter that considerably influences the performance is the initial guess for the starting state of the target. This influence could be reduced, for some of the tested cases, by using an annealing approach for adapting the process noise.

Since, in the literature, the PF is often a cnadidate for replacement of the EKF, we have decided to test it with a more accurate model for the noise in the measurements. According to a theoretical definition of the received power distribution, the observation model was redesigned and integrated in the PF, as a solution for enhancing the accuracy of the localization system. After running the first experiment, the PF has given an output trajectory that was presenting great randomness with respect to the seed used in the generation of the particles. This was due to the low magnitude of the hidden process noise. In order to reduce this influence, the process noise was increased and later cleared by a DES. This approach has resulted in a slightly higher accuracy then the EKF, but it has at the same time increased the computational burden. A particle smoothing approach has been discarded due to the computational effort already presented by the PF approach.

Chapter 6

Modeling Target Trajectory Based on a Segment-Wise Linear Approach

The previous chapter evaluated some of the existent solutions for tracking WDs in wireless networks. For all the solutions used in Chapter 5, the kinetics of the user were estimated at each time step. Instead, in the present chapter the movement of the user is assumed piece-wise linear and the algorithms attempt to estimate each segment of linear movement as a whole.

As it was explained in Section 3.1.1 and verified in Section 5.5, the Kalman family of filters provides algorithms for estimating the state space of an hidden process corrupted by noise based on indirect observations of that process also corrupted by noise. A particularity of these filters is that they are not designed to support parameters modeled by jumping processes. Thus, using Kalman filters in such cases, requires the use of some additional strategies. As done in Section 5.5 the simplest solution is to add an artificial noise component in the process to be estimated. This obviously increases the noise in the output estimations, but gives at the same time higher flexibility for the filter to model jumps in its parameters. Another solution is to use multiple models as done in Section 5.6. This solution, besides having more flexibility to model jumping processes than the ordinary EKF, it is computationally heavier than the EKF. The last solution presented in this dissertation is the integration in the EKF of a jump detection algorithm as the one explained in Section 3.2.1. This detector when estimates a jump, momentarily increases the process noise in the user dynamics until the end of the jump is estimated.

Although the WDs are characterized by high mobility, their motion patterns are not characterized by such noisy evolution in time (as shown in Fig. 6.1), but instead by frequent changes in direction and/or velocity. In between changes of direction and/or velocity, the motion pattern follows a linear model, with unknown parameters and corrupted by low noise when compared with the magnitude of the kinetics of the user. For instance, let us imagine a pedestrian walking in the street. While moving, the pedestrian has the tendency of setting up a goal and walk towards that goal using the closest path, i.e. straight walk. If an obstruction exists, the pedestrian targets that obstruction as a goal, and after arriving to that obstruction, a new goal is chosen. Thus, the pedestrian walk

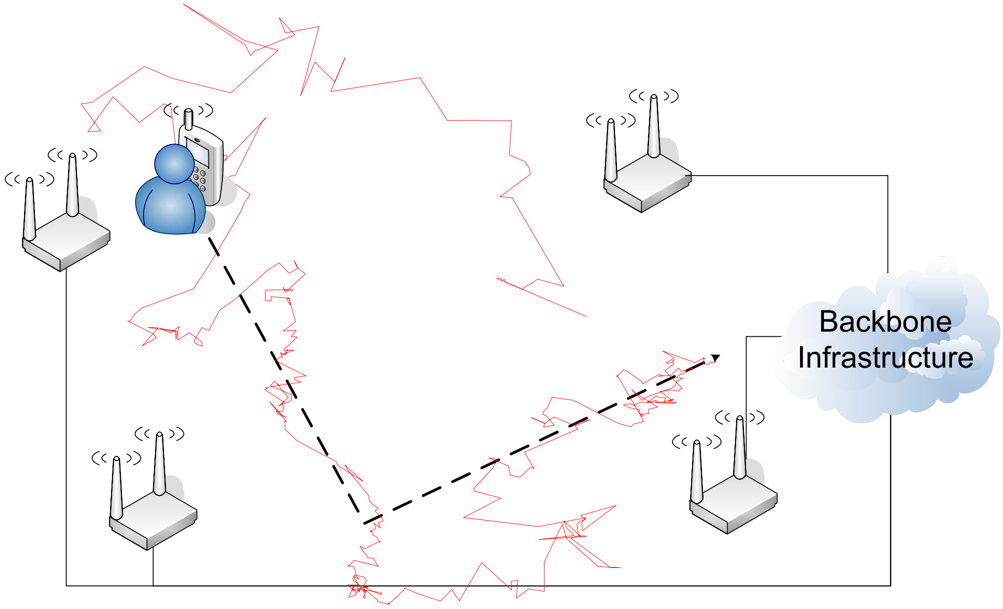


Figure 6.1: Example of a noisy estimation of the user trajectory: the black dashed line represents a true trajectory and the red solid line represents an example of the estimated trajectory.

will be characterized by a piecewise linear trajectory. Note however that although the obstructions usually introduce additional problems concerning for instance signal blocking or shadowing in the propagation channels, we do not deal with these problems in this chapter.

For trajectory estimation in applications such the one in the present scenario, the most suitable model is the so called CV model (which was extensively used in Chapter 5). This model assumes a constant velocity continuously corrupted by noise. In the eq.(5.2) of the previous section, it is possible to see that in fact the noise component $\mathbf{w}$ in the hidden process noise X_k is defined for every discrete time t_k.

According to the segment-wise linear approach of this chapter, the model of eq.(5.2) is not the most suited approach. The reason is that within a segment the trajectory is assumed non noisy, existing noise only in the points where maneuvers happen. Thus, given the position and respective timestamps where maneuvers have actually happened one can estimate the whole trajectory, simply by interpolating those points at the desired time. The problem can be now defined as the problem of estimating the position and time of those maneuvers.

This chapter proposes an alternative solution to the standard one for estimating the user trajectory as a piece-wise linear movement. The solution relies on a recursive Bayesian filter and it proposes a new model for the target mobility, called Drifting Points Model. Additionally, as it is shown in Section 6.2 the algorithm requires an external decoupling of the movement moments.

6.1 Using an Extended Kalman Filter with the Standard Constant Velocity Model

The common solution for solving such a trajectory estimation problem is to use a Bayesian filter coupled with a CV model. Assuming for instance an EKF as the filter responsible for the trajectory estimation, we can define the localization system as in Fig. 6.2. Figure 6.2 shows the structure of a localization system with maneuver detection. The system is divided in 3 main blocks, the filter itself, the maneuver detector and a noise adaptation block. Additionally we have decided to include a quarantine block. This block is activated every time a maneuver is detected. Then, it deactivates the maneuver detector until the system gets observable (see Section 5.4). The filter is responsible for calculating in a recursive fashion, the kinetics of the target. The maneuver detector is in charge of estimating maneuvers and the noise adaptation is responsible for controlling the artificial noise added to the state estimation of the filter. This artificial noise permits the filter to model uncertainty in its estimations.

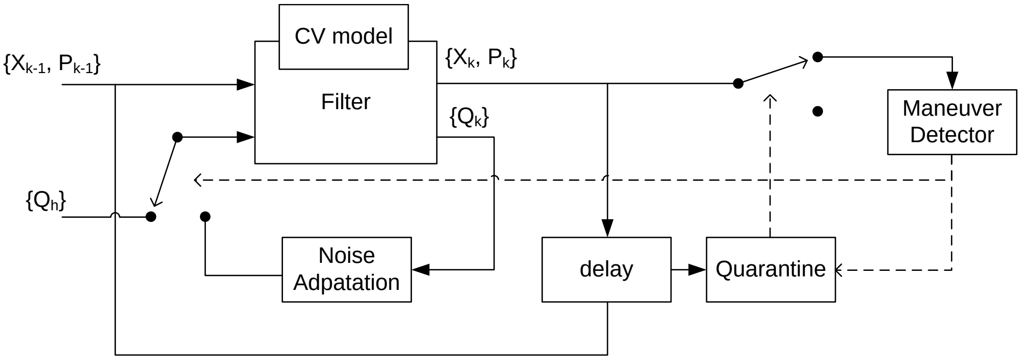


Figure 6.2: Constant Velocity Algorithm. The solid lines correspond to flow of data, while the dashed lines correspond to flow of control signals.

6.1.1 Modeling and Extended Kalman Filter with Maneuver Detector

For modeling an EKF by using the common constant velocity model, it can be done similarly to what was done in Section 5.5. Concerning the mobility model, the current situation considers a much higher velocity than the one used in Section 5.5, meaning that some adjustment may be necessary. In fact, that is the case of the standard deviations $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$, which need to be increased.
As we have seen in the background Section 3.2.1, one of the possible solutions for tracking maneuverable targets is to use a maneuver detector to adapt the system. A common solution in the literature is the χ^2 method, which works upon the innovation process (as it is defined in Section 3.1.1) of the estimation algorithm. This method assumes gaussianity in the innovation process and its implementation is totaly independent on the models.

In further sections we will use the χ^2 maneuver detector in order to find out when the target has experienced a change in the direction of its constant velocity movement. As a way of minimizing the individual influence of each measurement in the maneuver decision, the exponential average windows as given by eq.(3.53) is used. This solution gives commonly more robust results because it allows to cancel out the effect of occasionally high peaks provoked by possible outliers.

6.1.2 Noise Adaptation

For adapting the process noise in the system we adopt the same strategy as the one used in eq.(5.31). Since at the initial state we have not enough knowledge about the process to be estimated, we assume the standard deviations $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ to be initialized with a "high" value, i.e. σ_{high}. The definition of "high" in this context is a rule of thumb, i.e., the noise is tuned in order to reach a balance between fast convergence of the estimator and lower variations on the same estimator. Then, along time, the process noise is reduced according to an exponential decay with reason ς. This approach let us tend to more and more confidence along time, thus, reducing the artificial noise introduced in the system.

When a new maneuver is detected, it means that the confidence on the estimator shall be decreased, due to the fact that the direction chosen by the WD is unknown. In this case, the value of σ_{high} for the process noise is reestablished. Afterwards, the same exponential decay restarts to actuate upon the new value of the process noise.

6.2 An Alternative Approach: The Drifting Points Model

This section introduces a new model which complies with the main idea of basing the localization solution on the points where maneuvers occur. Due to the nature of the model presented in this section, this maneuvering point will be named as Drifting Points (DP), and the model is called Drifting Points Model.

6.2.1 The Algorithm

The DP model is a solution that instead of estimating position in a time-wise and point-wise fashion, it estimates linear segments of the trajectory. During the estimation of each segment, the more information we have available, the higher is the confidence on the estimation of that segment. Since the segments consider that the velocity is constant in every point, we can define the segment only based on its initial and final position. Subsequently, defining the trajectory is a matter of knowing all the positions and respective timestamps where maneuvers have occurred. Thus, as we can see in the schematic representation of the algorithm in Fig. 6.3, the output of the algorithm is the estimated maneuver positions. The main differences between the present algorithm and the standard one in Fig. 6.2 is twofold:

- The DP model itself is attached to the filter, instead of the CV model.
- A transformation block controlled by the maneuver detector.

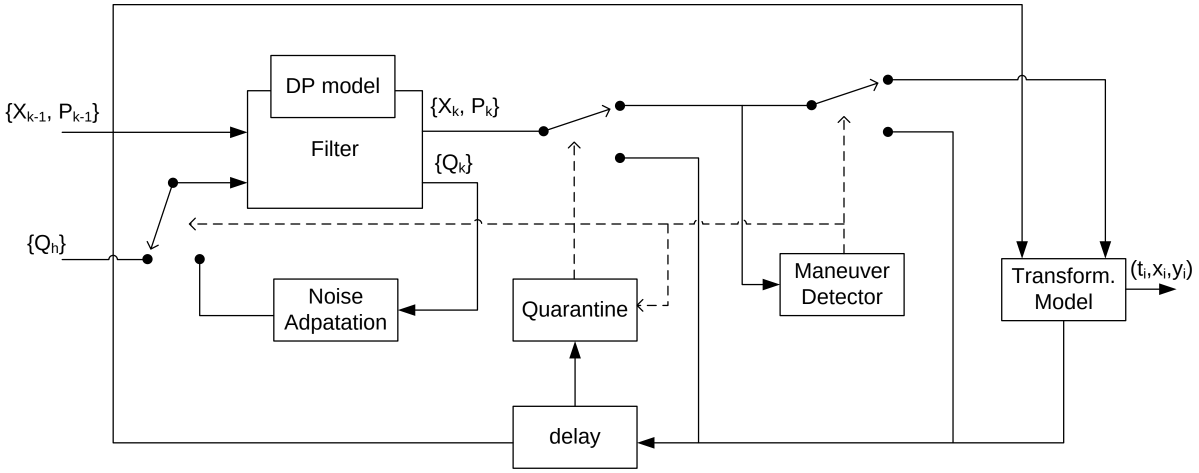
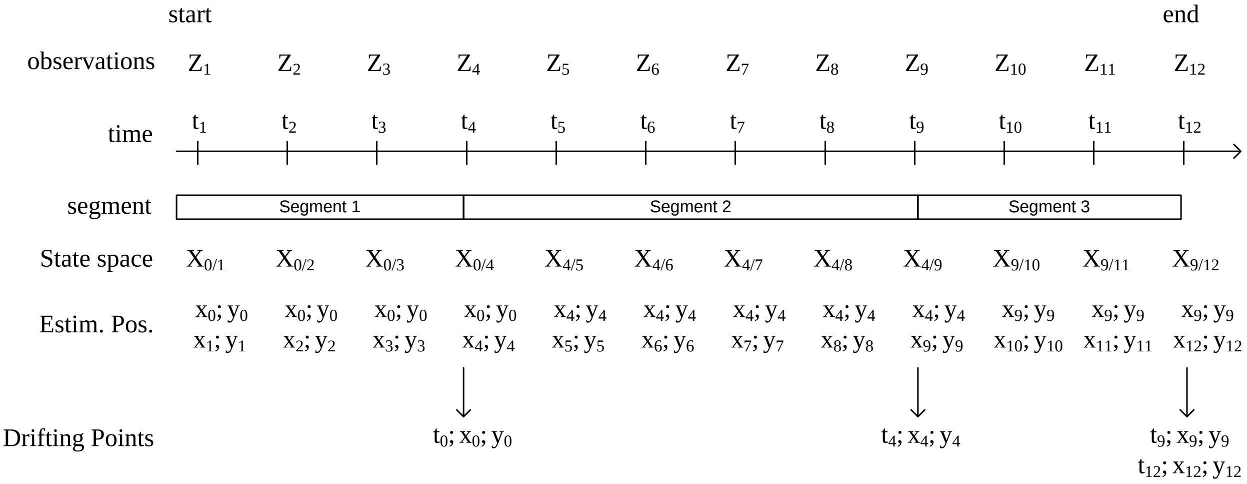


Figure 6.3: Drifting Points Algorithm

The DP model represents the process of estimation the starting and ending point of each segment. Due to this reason, when a new segment starts, what is made by the maneuver detector, a transformation in the state space and covariance matrix is necessary. As Fig. 6.4 shows, when a new segment is detected, the starting position of the old segment is identified as the drifting point contributing to the final trajectory estimation. The ending position of the old segment plays the role of starting position in the new segment, and a new brand point is added as the ending of the new segment (see Section 6.2.4).



	start											end
observations	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_{10}	Z_{11}	Z_{12}
time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
segment	Segment 1				Segment 2					Segment 3		
State space	$X_{0/1}$	$X_{0/2}$	$X_{0/3}$	$X_{0/4}$	$X_{4/5}$	$X_{4/6}$	$X_{4/7}$	$X_{4/8}$	$X_{4/9}$	$X_{9/10}$	$X_{9/11}$	$X_{9/12}$
Estim. Pos.	$x_0; y_0$ $x_1; y_1$	$x_0; y_0$ $x_2; y_2$	$x_0; y_0$ $x_3; y_3$	$x_0; y_0$ $x_4; y_4$	$x_4; y_4$ $x_5; y_5$	$x_4; y_4$ $x_6; y_6$	$x_4; y_4$ $x_7; y_7$	$x_4; y_4$ $x_8; y_8$	$x_4; y_4$ $x_9; y_9$	$x_9; y_9$ $x_{10}; y_{10}$	$x_9; y_9$ $x_{11}; y_{11}$	$x_9; y_9$ $x_{12}; y_{12}$
Drifting Points				$t_0; x_0; y_0$					$t_4; x_4; y_4$			$t_9; x_9; y_9$ $t_{12}; x_{12}; y_{12}$

Figure 6.4: Example of a run of the Drifting Points model.

As we can infer, this approach of transforming the moments of the movement opposes to the inherent optimality property of the Kalman filter theory.

6.2.2 Mathematical Formulation of the Model

Assume that the estimation using the DP model is running at present time t_k and that the previously detected maneuver happened at time t_i, $i < k$. Let us start by assuming no noise and later consider it when appropriate. Given that the velocity is constant along the entire segment, the state space of eq.(5.25) can be written as:

$$X_{i/k} = \begin{bmatrix} x_k & y_k & \dot{x}_{i/k} & \dot{y}_{i/k} \end{bmatrix}^T \tag{6.1}$$

where the notation $\dot{x}_{i/k}$ means that $\dot{x}$ is being estimated at time t_i given the observations until time t_k. If no new maneuver happens at time t_{k+1}, eq.(6.1) results by inference in:

$$X_{i/k+1} = A_{k+1} X_{i/k} \quad , \quad A_{k+1} = \begin{bmatrix} 1 & 0 & t_{k+1} - t_k & 0 \\ 0 & 1 & 0 & t_{k+1} - t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

Let us now define the state space $Y_{i/k}$ as:

$$Y_{i/k} = \begin{bmatrix} x_k & y_k & x_{i/k} & y_{i/k} \end{bmatrix}^T \tag{6.3}$$

where, at time t_i, instead of estimating the velocity, we estimate the position. Since the velocity is constant, eq.(6.1) and eq.(6.3) have a linear relation.

$$X_{i/k} = \Xi_k Y_{i/k} \quad , \quad \Xi_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{t_k - t_i} & 0 & -\frac{1}{t_k - t_i} & 0 \\ 0 & \frac{1}{t_k - t_i} & 0 & -\frac{1}{t_k - t_i} \end{bmatrix} \tag{6.4}$$

By applying eq.(6.4) in eq.(6.2), it is possible to obtain a model which defines at each time step, the position where the last maneuver has been estimated and the current position of the WD.

$$Y_{i/k+1} = \Phi_{k+1} Y_{i/k} \quad , \quad \Phi_{k+1} = \Xi_{k+1}^{-1} A_{k+1} \Xi_k \tag{6.5}$$

where the transition matrix is given by:

$$\Phi_{k+1} = \begin{bmatrix} \phi_{k+1} & 0 & 1 - \phi_{k+1} & 0 \\ 0 & \phi_{k+1} & 0 & 1 - \phi_{k+1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad , \quad \phi_{k+1} = \frac{t_{k+1} - t_i}{t_k - t_i} \tag{6.6}$$

Note however that this model requires one to memorize the time when the previous DP had occurred. Obtaining the final trajectory is of the responsibility of the transformation block described in Section 6.2.4.

Concerning the process noise Q, similar annealing approach was used as in Section 6.1.2. Note however that, since in the DP model the state space includes only positions, Q is defined as:

$$Q = \sigma_x^2 I \tag{6.7}$$

where σ_x is the standard deviation of the process noise and I is the identity matrix of proper size.

6.2.3 Modeling the Maneuver Detector

Regarding the maneuver detector we have decided to use exactly the same used in the previous CV model (see Section 6.1.1). For the sake of a fair comparison between models we have decided to keep the same maneuvering detector parameters in both models in every run.

6.2.4 The transformation of moments of the process

Based on the proposed DP model, we can see in eq.(6.3) that the state space is constantly composed by the previous DP and the location of the target at time t_k. For this reason, when a change of direction happens, a new DP is created and thus the state space $Y_{i/k}$ has to be manipulated. Additionally, the covariance matrix $P_{i/k}$ has to be redesign as well. In Fig. 6.3, this manipulation is made in the block "Transformation Model" and in Fig. 6.4 it happens at time t_4 and t_9. Assuming that at time t_k a maneuver is detected and a new DP is added to the system, the state space is changed such that:

$$Y_{k-1/k} = \begin{bmatrix} 0_2 & 0_2 \\ I_2 & 0_2 \end{bmatrix} Y_{i/k-1} + \begin{bmatrix} I_2 & 0_2 \\ 0_2 & 0_2 \end{bmatrix} Y_{i/k} \tag{6.8}$$

where I_2 is the 2×2 identity matrix and 0_2 is the zero matrix of dimension 2×2. The covariance matrix instead suffers the following transformation:

$$P_{k-1/k} = \begin{bmatrix} 0_2 & 0_2 \\ I_2 & 0_2 \end{bmatrix} P_{i/k-1} \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} + \sigma_x^2 \begin{bmatrix} I_2 & 0_2 \\ 0_2 & 0_2 \end{bmatrix} \tag{6.9}$$

Similarly, the process noise Q is transformed as:

$$Q_{k-1/k} = \begin{bmatrix} 0_2 & 0_2 \\ I_2 & 0_2 \end{bmatrix} Q_{i/k-1} \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix} + \sigma_x^2 \begin{bmatrix} I_2 & 0_2 \\ 0_2 & 0_2 \end{bmatrix} \tag{6.10}$$

When a maneuver is detected, the knowledge of the second moment (i.e. covariance) of the current position is erased. This is because at the discrete time t_k when the maneuver is detected, we do not know the direction chosen by the target.

After these transformations, the position $Y_i^{dp} = [x_i, y_i]^T$ at time t_i is identified as a drifting point (see example in Fig. 6.4). It is very important to mention here that the final trajectory estimation will be given by all the Y_ℓ^{dp}, where Y_ℓ^{dp} is the estimated drifting point at time t_ℓ. To determine the position of the target at time $t_m \neq t_\ell$, it is only necessary to interpolate the points Y_ℓ^{dp}. The result of the interpolation is defined as X_m^{dp}.

$$X_m^{dp} = \text{interp}(t_m \quad ; \quad Y_{1:\max(\ell)}^{dp} \quad ; \quad t_{1:\max(\ell)}) \tag{6.11}$$

where the function interp() has been previously introduce in eq.(5.11). The result of the DP model is that the entire trajectory is uniquely defined by the points where maneuvers were estimated. See Fig. 6.5 for an illustration of the aforementioned interpolation of the DPs.

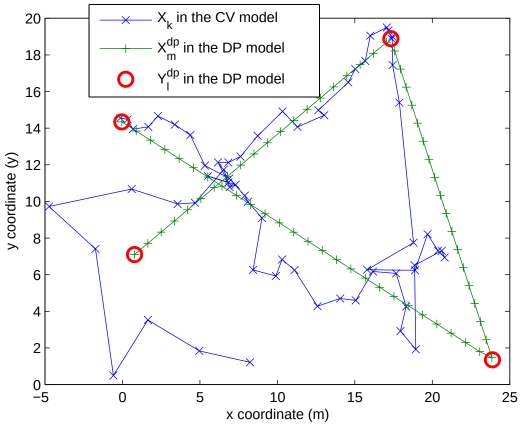


Figure 6.5: Illustration of the DP model in operation.

6.3 Performance Analysis

6.3.1 Simulation Framework for Performance Analysis

For analyzing the performance of the solutions presented in this section, we have developed a simulator in MATLAB [mat04]. The scenario is assumed to be an open area with 9 APs placed along a grid, $10m$ apart from its closest neighbors. This setup attempts to represent a typical scenario where short-range APs track a WD. The localization was performed in a 2D plane, despite the fact that the APs were elevated $1.5m$ from that same plane. Out of the 9 APs, the decision upon which would be active at each time step was made by a random generator able of producing an output of integers from 1 to 9 uniformly distributed. The measurement rate f_s has assumed different values such as: $5Hz$, $15Hz$, $50Hz$ and $100Hz$. The purpose of this is to simulate different levels of information, i.e. the higher the frequency is, higher is the amount of information about the target position. To the path loss expression dictated by eq.(2.1) it was added a zero-mean Gaussian distributed noise component with standard deviation $\sigma_p = 6dB$. For the path loss, it was setup $\alpha = -50dBm$ and $\beta = 2.5$ in order to comply with the same values used in Chapter 5.

On the target side, the simulator was built as follows: an initial position (x_0, y_0) and velocity v was set and three different trajectories were defined. They are the straight trajectory (Fig. 6.6), the cross trajectory (Fig. 6.7) and the staircase trajectory (Fig. 6.8). All the trajectories were

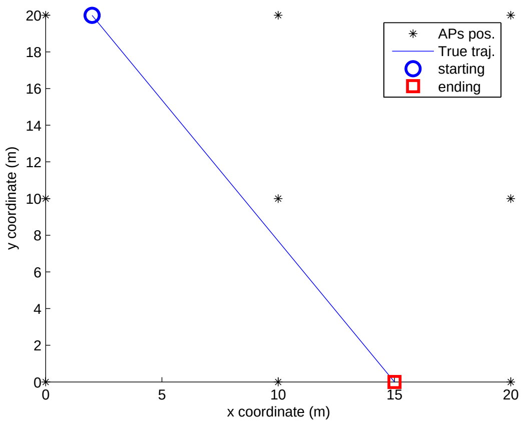


Figure 6.6: Straight trajectory used in the simulations.

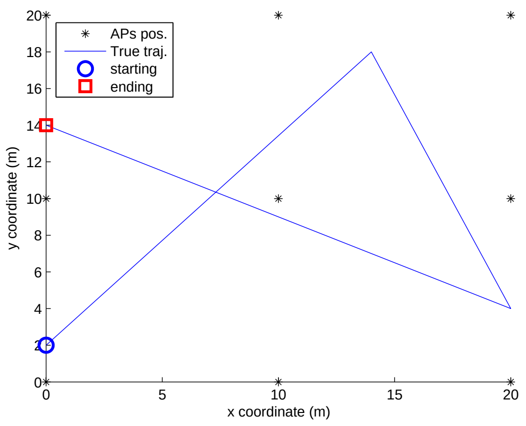


Figure 6.7: Cross trajectory used in the simulations.

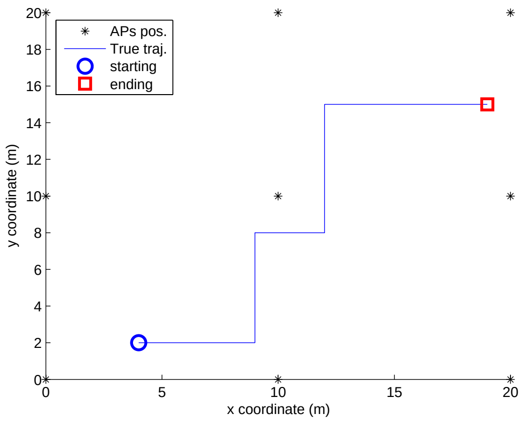


Figure 6.8: Staircase trajectory used in the simulations.

characterized by a piecewise linear movement, with sharp corners of changing of direction. While the straight trajectory is used to simulate movement with no changes in direction, the remaining trajectories have different number and angles of changes in direction. The final staircase trajectory is used to simulate movement with frequent changes in direction. The velocity was assumed constant from the starting to the ending point. To simulate a common human walk velocity, we have setup $v = 1.4m/s$. For summarizing all the parameters used in the generation of measurements, we have included Table 6.1. Regarding the estimation phase and the maneuver detector, Table 6.2 and Table 6.3 respectively summarize the involved parameters. Besides these parameters we have decided that for determining the CDFs of the errors in the estimation, each trajectory would need to be subject of 1000 independent runs.

Table 6.1: Simulation parameters used for generating measurements of RSS.

Parameter	Straight	Squared	Staircase
α (dBm)	-50		
β	2.5		
σ_p (dB)	6		
f_s (samples/s)	$\{5, 15, 50, 100\}$		
v (m/s)	1.4		
$(x_0; y_0)$ (m)	(2 ; 20)	(0 ; 2)	(4 ; 2)
Trajectory Length (m)	23.8	58.8	28
Number of Maneuvers	0	2	4

Table 6.2: Parameters used in the design of the EKF

Parameter	CV model	DP model
α (dBm)	-50	-50
β (unitless)	2.5	2.5
initial $\sigma_{\ddot{x}}$ (m/s^2)	5	—
initial $\sigma_{\ddot{y}}$ (m/s^2)	5	—
initial σ_x (m)	—	10
σ_p (dB)	6	6
Initial State	true	true
Initial Covariance Matrix	Identity matrix	Identity matrix

Table 6.3: Parameters used in the χ^2 maneuver detector.

Parameter	Value
η	0.9
Threshold (%)	95

6.3.2 Constant Velocity Model

As a reference case, this section testes the CV model. Before evaluating the performance of the standard CV model we have decided to look at a single run of the estimation of a straight trajectory. Figure 6.9 shows the result of the final trajectory estimation when the frequency of measurements is $15Hz$. The initial guess is calculated, as it is along the entire chapter, based on the triangulation of the initial three measurements. As we can see from Fig. 6.9, the initial guess has a great impact on the final estimation of the trajectory, what is clearly undesirable. Moreover, we can see that the trajectory is estimated with considerable noise, what do not represent the true trajectory. Although the estimation is undesirably noisy during the first half of the movement, it gets better and better the closer the targets gets to its final location. Figure 6.10 shows the signal of the inverse of the χ^2 PDF given by eq.(3.53). As we can see at the beginning of the estimation there exists a clear detection of a maneuver, what is due to the error in the initial guess. Mathematically, this systematic error can be considered as a bias, i.e. a sequence of "large" innovations in the estimation process. For this reason, the maneuver detector is triggered and a maneuver is estimated. In order to have an idea concerning performance we have decided to run the same simulation 1000 times,

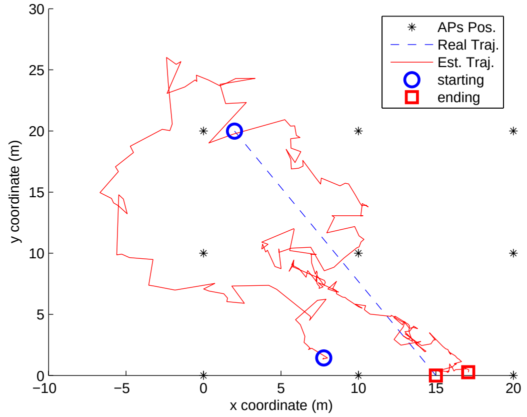


Figure 6.9: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

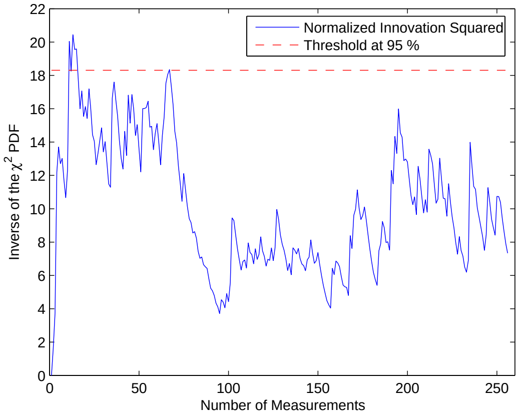


Figure 6.10: Maneuver detector for $f_s = 15Hz$. The signal represents the inverse of the χ^2 PDF as given by eq.(3.53).

where the only difference among runs was the seed used in the block of measurements generation. In Fig. 6.11 the empirical CDF of the PMDE metric as introduced in Section 5.2 is plotted for

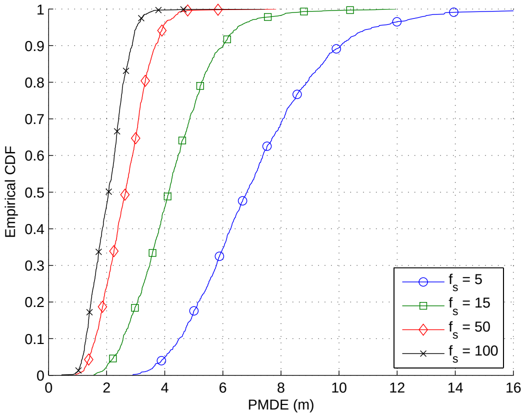


Figure 6.11: Empirical CDF of the PMDE metric for the straight trajectory.

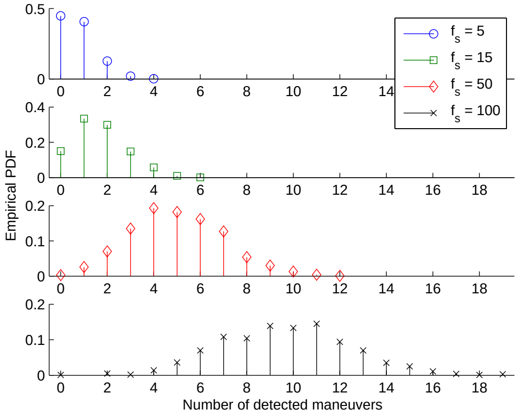


Figure 6.12: Empirical PDF of the number of detected maneuvers in the estimation of the straight trajectory.

several frequencies of measurements. Additionally in Fig. 6.12 we plot the empirical PDF of the number of detected maneuvers. While running the experiment for several different frequencies of measurements we can conclude that the higher the frequency is, the lower is the error, what is in fact expected. However, looking at Fig. 6.12, it may looks surprising at a first glance that the number of detected maneuvers also increases with the frequency. This can be explained by the fact that when the frequency increases, the number of measurements per covered unit of space is higher. Thus, due to the stochastic behavior of the innovation process, the chance of getting sequences of "large" innovations increases. Then, if any of these innovations gets higher value than the threshold of the maneuver detector, a maneuver is triggered. A solution for this can be to increase the forgetting factor of the exponential average (incorporated in the maneuver detector) when the frequency increases.

Until this point, the evaluation has been held in a scenario where no maneuvers actually happen. Thus, it is important to see the performance of the localization solution when maneuvers happen at unknown number and timestamps. In Fig. 6.13 we can see a single run for the cross trajectory.

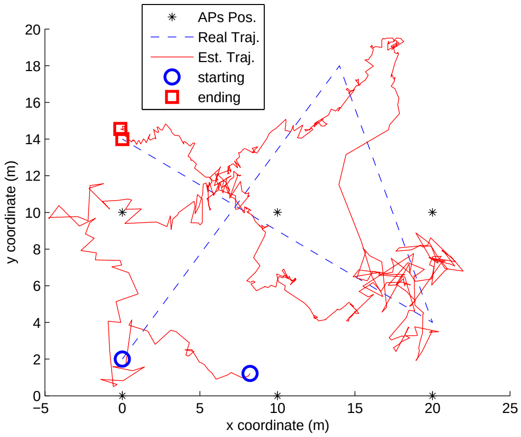


Figure 6.13: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

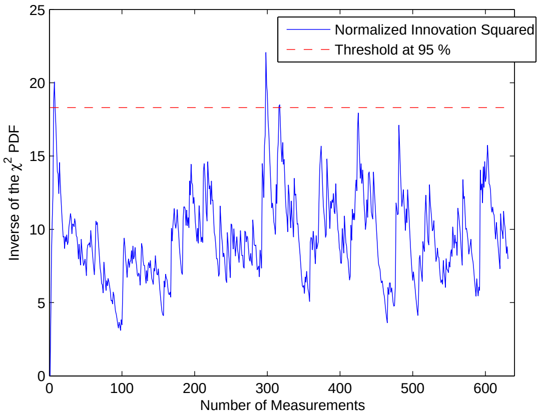


Figure 6.14: Maneuver detector for $f_s = 15Hz$.

In this trajectory, there happen 2 maneuvers which shall be detected. However, as we can see in Fig. 6.14, 3 maneuvers are actually detected. The first estimated maneuver at the very beginning of the estimation is basically the influence of a highly erroneous guess for the initial state of the user dynamics. This highly erroneous initial guess is observed as a sequence of "high" innovations, which subsequently triggers the maneuver detector. The other 2 maneuvers are detected too close to each other in comparison to what would be expected given that velocity is constant along the entire trajectory. The influence of this fact is observable in the final trajectory of Fig. 6.13, which shows, in the second segment, that consecutive positions are estimated further apart from each other than what happens in the other segments. An additional conclusion that we can draw from Fig. 6.13 is that, although detection of maneuvers allows the estimation of changes in detection, it also increases the noise in the final trajectory estimation. Looking at the performance of 1000 runs, we can reach similar conclusions as in the straight trajectory. Figure 6.15 shows the direct relation between frequency of measurements and performance. Comparing Fig. 6.15 and Fig. 6.11 we can see that there is a small improvement in performance for the cross trajectory. The reason is that since the trajectory is larger, the influence of the initial guess in the final trajectory estimation gets slightly lower, and subsequently performance higher.

As we have previously seen, the measurement noise in the present application, when translated into space coordinates, is considerably high when compared with the actual variations in our hidden process (i.e. the variation in positions). To show the influence of such situation in the performance

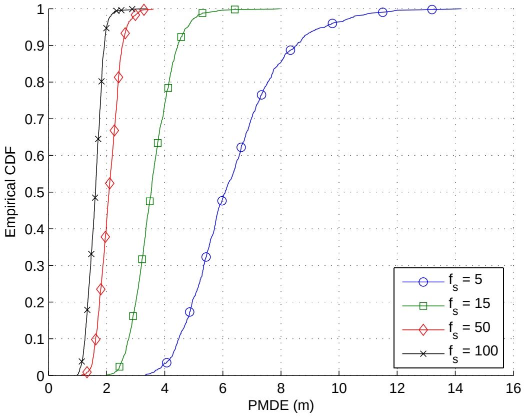


Figure 6.15: Empirical CDF of the PMDE metric for the cross trajectory.

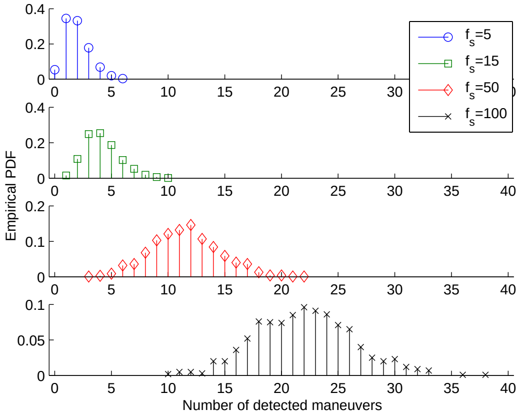


Figure 6.16: Empirical PDF of the number of detected maneuvers in the estimation of the cross trajectory.

of the all system, we have decided to define a trajectory (called staircase trajectory) with shorter segments than previous examples and test the system with different frequencies of measurements. In Fig. 6.17 we show the result of a single run for a frequency of $f_s = 15Hz$. As we can see, the

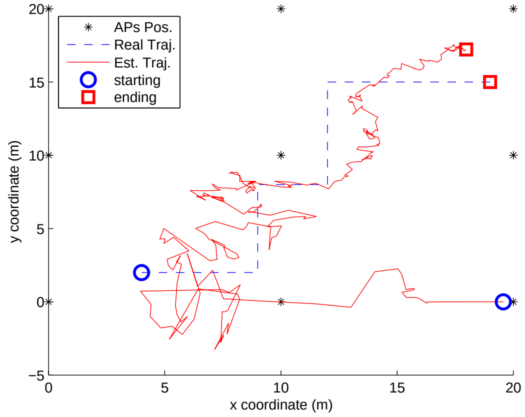


Figure 6.17: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

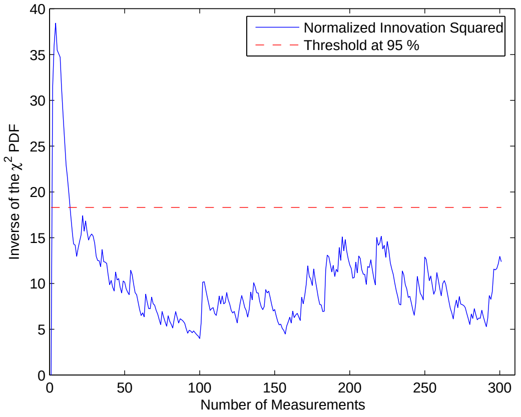


Figure 6.18: Maneuver detector for $f_s = 15Hz$.

segments are hardly distinguishable in the estimated trajectory. Moreover, looking at Fig. 6.18, we can see that in fact the maneuver detector has never been triggered by a potential maneuver. Note that the initial peak in Fig. 6.18 is due to the erroneously estimated initial position. Based on this results, it is fair to claim that, since maneuvers are not actually detected, it is possibly better to consider that whole movement as a single segment and completely disregards maneuvers that are not detectable. This would represent an advantage in the sense that the final estimation of the target trajectory would not be a noisy estimation as the one shown in Fig. 6.17. Additionally for completeness purposes we have decided to run the simulation in the same staircase trajectory, but with a measurement frequency of $f_s = 100Hz$. Figure 6.19 shows the estimated trajectory and

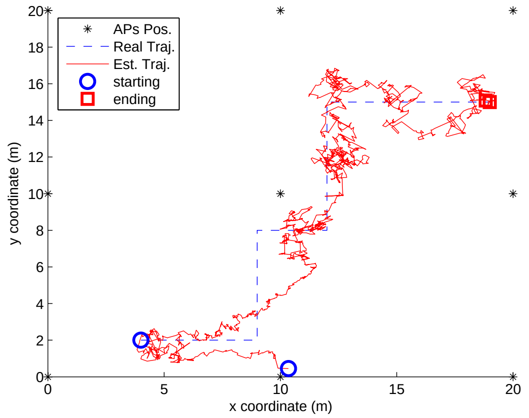


Figure 6.19: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 100Hz$.

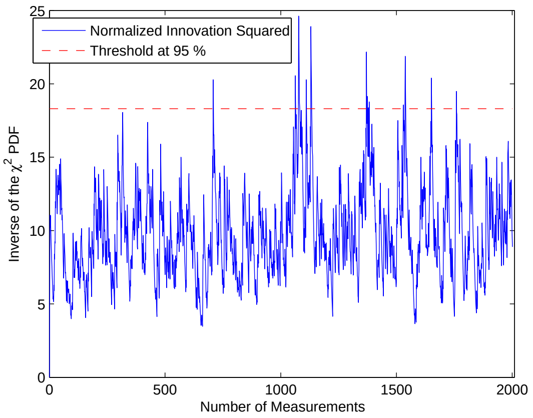


Figure 6.20: Maneuver detector for $f_s = 100Hz$.

Fig. 6.20 shows the signal for the maneuver detection decision. As we can see, for the current case, the estimated trajectory shows a trend more similar to the true movement than in the previous situation (in Fig. 6.17). From Fig. 6.20 it is possible to see that maneuvers are in fact detected, what explains the result of Fig. 6.19. These results motivate us to develop a model where noise is considered nonexistent within a segment. Instead, it can be considered as only existent in the beginning and end of each segment. In a conceptual point of view, we could imagine that one might have better performance by estimating segments as a whole instead of estimating time-wise position estimators.

6.3.3 The Drifting Points Model

As a starting point for a further comparison with the CV model, we have decided to run the DP model for the same data points used in Section 6.3.2. Starting by evaluating the result of a straight trajectory, we have plotted a single run of the simulation in Fig. 6.21 and Fig. 6.22. From Fig. 6.21, we can see that, in comparison with the previous CV model (Fig. 6.9), the DP model approximates the true trajectory with less error. Additionally it is also possible to see that the signal used for detecting maneuvers is considerably below the threshold. This was not so clear in the CV model, being the threshold even crossed during the beginning of the experiment. Evaluating the result for several independent runs, we plotted the CDF and the number of detected maneuvers from 1000 independent runs. The result shown in Fig. 6.23 clearly tells us that the DP models outperforms the CV model (Fig. 6.11) for this straight setup. Concerning the number of detected maneuvers, the PDFs of both models, Fig. 6.24 for the DP model and Fig. 6.12 for the CV model, show similar results.

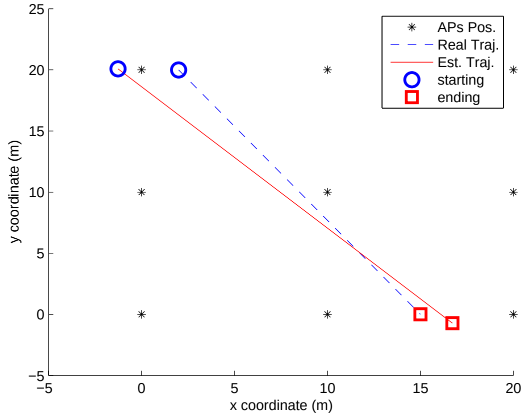


Figure 6.21: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

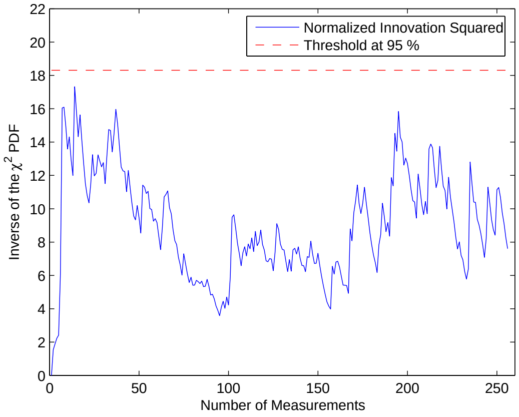


Figure 6.22: Maneuver detector for $f_s = 15Hz$.

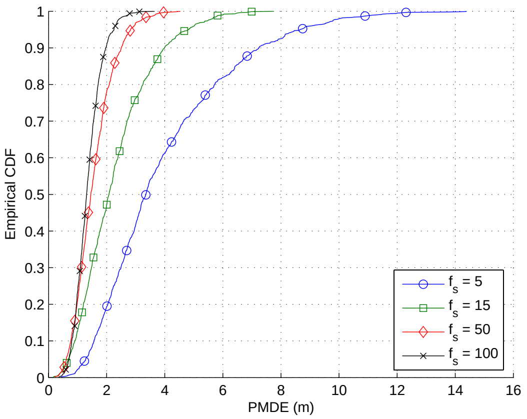


Figure 6.23: Empirical CDF of the PMDE metric for the straight trajectory.

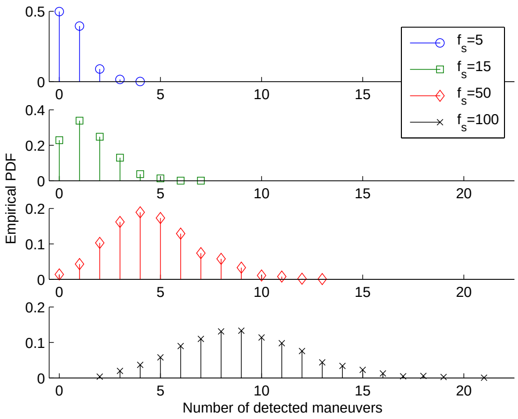


Figure 6.24: Empirical PDF of the number of detected maneuvers in the estimation of the straight trajectory.

When looking at the cross trajectory in Fig. 6.25, we can see that the system is still able to detect maneuvers. Due to the properties of the DP model, the final estimation of the trajectory is considerably less noisier than using the CV model (Fig. 6.13). Regarding the signal used for detecting maneuvers shown in Fig. 6.26, we can see interesting results in comparison with the same setup for the CV model. Regarding the actual detected maneuvers, the signal presents higher magnitude in the DP model case than in the case of the CV model (Fig. 6.14). Contrarily, the signals below the threshold, are of lower magnitude for the DP model. Another fact worth to mention lies in the initial measurements and the estimation of the initial state of the movement. Contrarily to the CV model, the DP model does not present such dependency with the errors in the estimation of the initial state, which are detected as a maneuver. In fact this behavior could

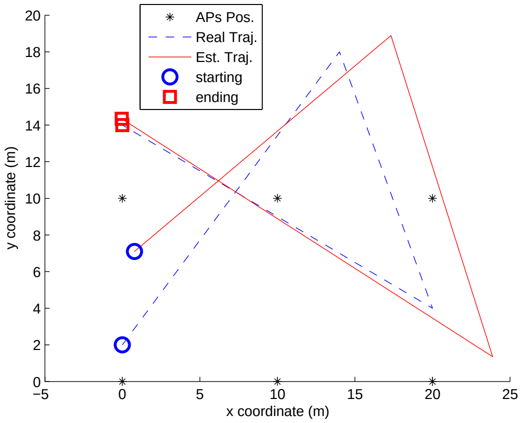


Figure 6.25: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

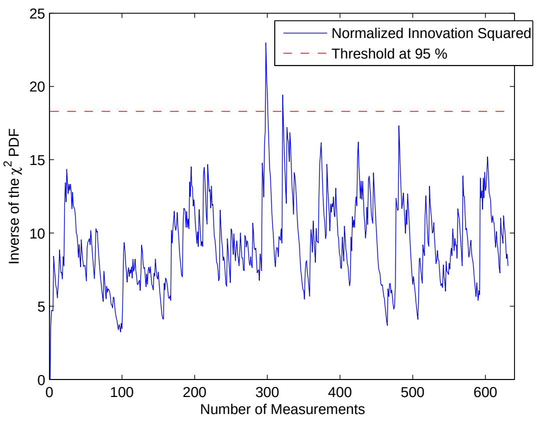


Figure 6.26: Maneuver detector for $f_s = 15Hz$.

be corrected in the CV model by simply increasing the process noise matrix that is active during a maneuver. The drawback of this approach is that the final estimation of the trajectory would result even noisier. For the DP model instead, increasing that parameter would not represent such a high impact since segments are estimated as a whole along time. This means that along time, instead of getting more confident only on the estimation of the position at time t_k, the DP model gives confidence in the estimation of the whole segment. Subsequently, the initial conditions have lower impact than for the standard model. The evaluation of the statistical behavior of the proposed model for the cross trajectory, was disregarded since the results are similar to those of the straight trajectory case. A closer comparison will be further done in Section 6.4.

The final test using the proposed model concerns the staircase trajectory. Once again, for the

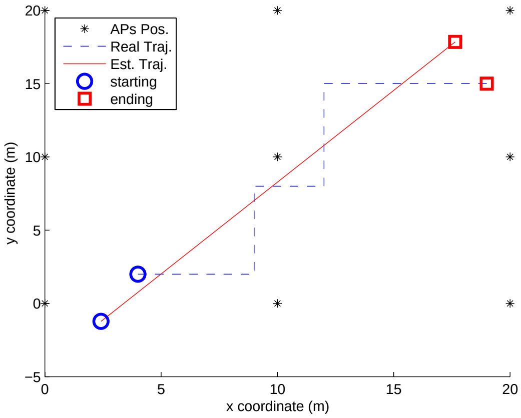


Figure 6.27: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 15Hz$.

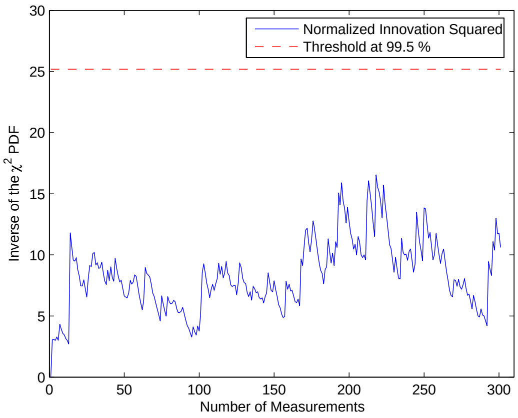


Figure 6.28: Maneuver detector for $f_s = 15Hz$.

frequency $f_s = 15Hz$ we can see in Fig. 6.28 that no maneuvers were detected. As a result, the final estimation of the trajectory is a straight line (see Fig.6.27). As a primary claim that we intend to further verify, we can say that in the case that no maneuvers are detected it is better to simply assume that they do not actually happen. This claim is well observable in the DP model of Fig. 6.27 since the final estimation totally disregards any change of direction. Considering now the case where more information is available, i.e. higher measurement frequency and subsequently more measurements per segment. This fact gives to the EKF a better chance to converge and subsequently the maneuver detector is more likely to detect maneuvers. As we can see in Fig. 6.30, the maneuver detector does in fact detect maneuvers. Although some of the maneuvers may be incorrectly detected, the final trajectory still approximates the real trajectory (see Fig. 6.29). As we can see, in comparison with the CV model results (in Fig. 6.19), the estimated trajectory is considerably less noisier for the case in Fig. 6.29.

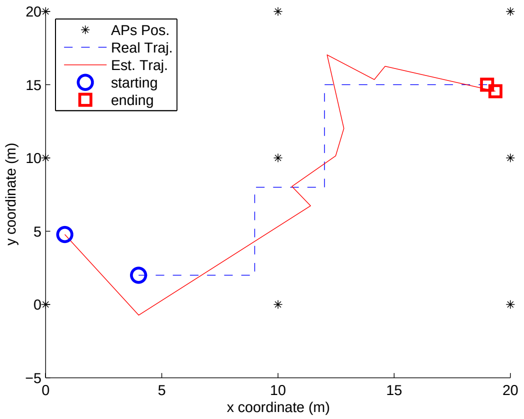


Figure 6.29: Example of the EKF operating with a maneuver detector for measurement frequency of $f_s = 100Hz$.

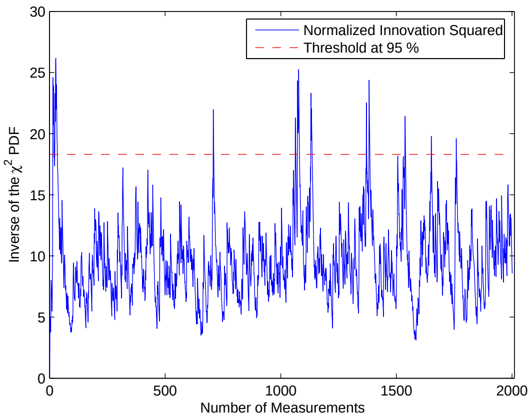


Figure 6.30: Maneuver detector for $f_s = 100Hz$.

6.4 Comparing the Drifting Points Model and the Constant Velocity Model

For comparison purposes we have decided to determine the gain of the DP model versus the CV model. In a set of 1000 independent runs of each algorithm, the gain was calculated according to the following expression:

$$G_{\mathfrak{a}} = \frac{\mathfrak{z}_{\mathfrak{a}}^{cv} - \mathfrak{z}_{\mathfrak{a}}^{dp}}{\mathfrak{z}_{\mathfrak{a}}^{cv}} \tag{6.12}$$

where $\mathfrak{z}_{\mathfrak{a}}$ is the inverse of the empirical CDF of the PMDE metric at $\mathfrak{a}\%$ of the 1000 independent runs, and the superscript identifies the model. As a consequence of the definition of $\mathfrak{z}_{\mathfrak{a}}$, the lower its value is, higher is the performance. As eq.(6.12) show, when the performance of the DP model is higher than the one of the CV model, $\mathfrak{z}_{\mathfrak{a}}^{cv} > \mathfrak{z}_{\mathfrak{a}}^{dp}$ and subsequently $G_{\mathfrak{a}} > 0$. Inversely, when the CV outperforms the DP, $\mathfrak{z}_{\mathfrak{a}}^{cv} < \mathfrak{z}_{\mathfrak{a}}^{dp}$ and thus $G_{\mathfrak{a}} < 0$.

Based on this metric we have evaluated the performance of both methods for different maneuver thresholds. The first evaluations regards the straight trajectory. As we can see in Fig. 6.31 the DP model clearly outperforms the CV model. As we can see, there is an increasing trend with the increase of the maneuver threshold. The reason is in the nature of the true trajectory. Since the trajectory is straight, increasing the threshold will imply that the probability of finding maneuvers decreases. Subsequently, less maneuvers tend to be estimated, and the probability that the DP model finds a single segment is higher. Contrarily, for the cross trajectory (see Fig. 6.32), this trend tends to lower gains when the threshold increases. The reason is the same as for the straight trajectory, but the consequences are different. Since in the cross trajectory there are in fact actual maneuvers, the underestimation of them drives the DP to lower performances. When appreciating

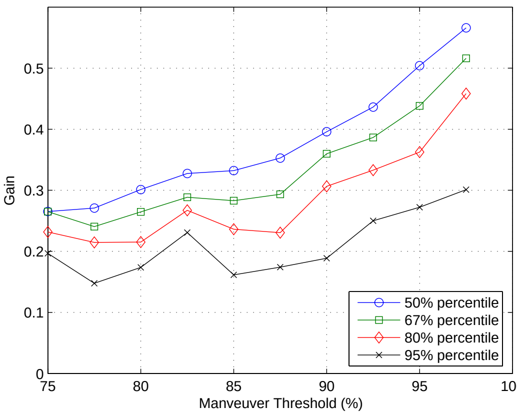


Figure 6.31: Gain vs. maneuver threshold for straight trajectory

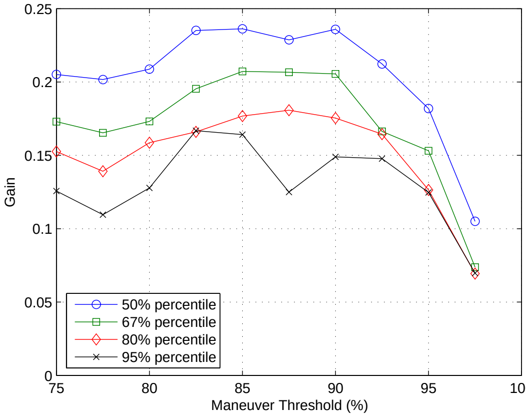


Figure 6.32: Gain vs. maneuver threshold for cross trajectory

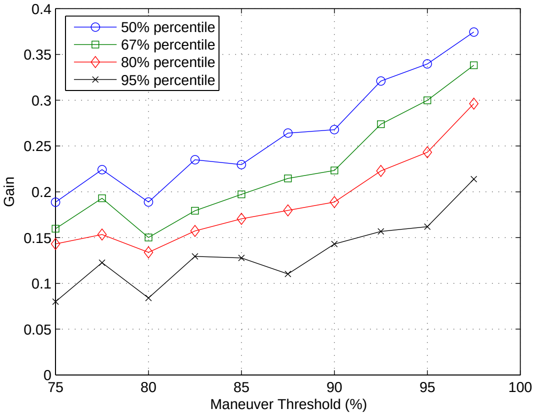


Figure 6.33: Gain vs. maneuver threshold for staircase trajectory

the results obtained for the staircase trajectory in Fig. 6.33, we can see a trend similar to the straight trajectory. The explanation for such fact lies in the claim done in the beginning of this chapter. Since for the present application the noise is very high when compared with the typical variation in the position, some of the smaller segments in the user trajectory may be impossible to estimate. Thus, as the results showed it is more adequate to neglect those undetectable segments and estimate only the most evident ones. The result is that the number of maneuvers may be underestimated, but nevertheless, performance is still enhanced. This is the behavior of the DP model.

One more conclusion that we can draw from these results is that the gain is consistently higher for lower values of $\mathfrak{a}$ (e.g. $\mathfrak{a} = 50\%$) given a constant maneuver threshold. This fact can be seen in any of the above plots.

6.5 Additional Batch Optimization for Off-line Processing

Until the moment the discussion has only considered solutions with linear complexity. However due to the properties of the proposed model, an additional optimization may be possible by an additional cost on complexity. This is possible due to the fact that for the DP model the entire trajectory can be uniquely defined based on a small subset of positions that define the entire trajectory, i.e. the drifting points.

6.5.1 Solution based on the Newton-Gauss Method

Since the number of drifting points is considerably smaller than the number of estimated positions in the CV model, a numerical minimization can be reasonably applied. The cost function for this optimization is defined as:

$$J = \sum_k \left(Z_k - \mathcal{H}(X_k^{dp}, 0) \right)^2 \tag{6.13}$$

where X_k^{dp} is the result of eq.(6.11), Z_k the measurement at time t_k and $\mathcal{H}()$ the observation model of eq.(3.21). The final estimation of the trajectory given by this additional optimization can be done by using the Newton-Gauss method:

$$\breve{Y}^{dp}_{1:\max(\ell)} = \arg \max_{Y^{dp}_{1:\max(\ell)}} J \tag{6.14}$$

where $Y^{dp}_{1:\max(\ell)}$ are the drifting points and $\breve{Y}^{dp}_{1:\max(\ell)}$ their optimized values.

6.5.1.1 Performance Analysis

For illustration purposes we have decided to plot the result obtained for the additional optimization. In Fig. 6.34 we have applied the additional optimization for the straight trajectory and in Fig. 6.35 for the cross trajectory. As we can see while comparing these results to the equivalent ones without

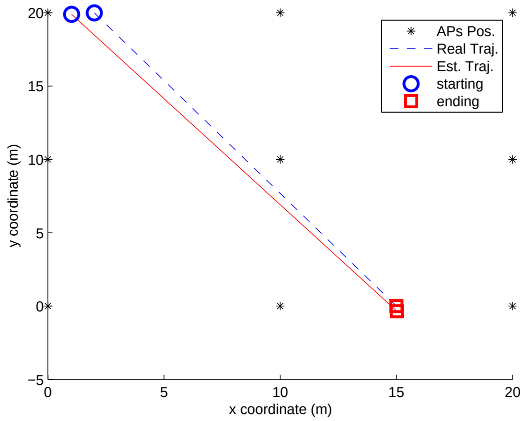


Figure 6.34: Example of the DP model with additional minimization for measurement frequency of $f_s = 15Hz$. Straight trajectory.

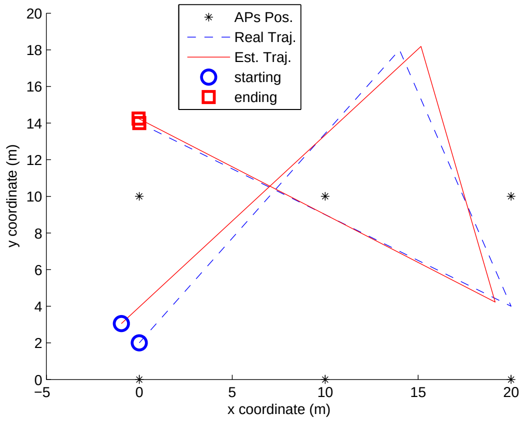


Figure 6.35: Example of the DP model with additional minimization for measurement frequency of $f_s = 15Hz$. Cross trajectory.

this minimization (in Fig. 6.21 and Fig. 6.25), the performance can be increased in the PMDE sense. However, looking at the statistical properties of 1000 runs of the same setups, we can see some unexpected results. Concerning the straight trajectory in Fig. 6.36, the results are considerably improved when this optimization is applied. However concerning the cross trajectory, the CDF in Fig. 6.37 shows a different behavior. The reason is that when the frequency is lower, the percentage of estimated trajectories with underestimated number of maneuvers is higher. Subsequently, this minimization solution does not give a considerable benefit on accuracy, because the number of linear segments is lower than the actual number.

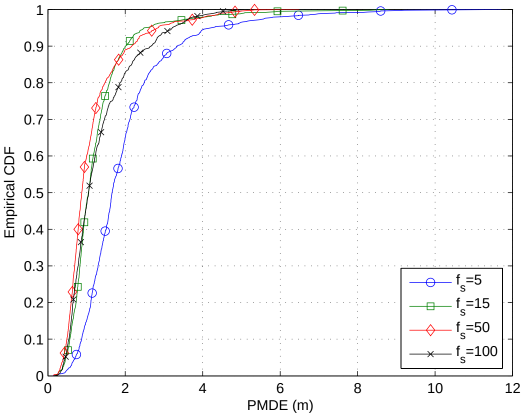


Figure 6.36: Empirical CDF of the PMDE metric for the straight trajectory after minimization.

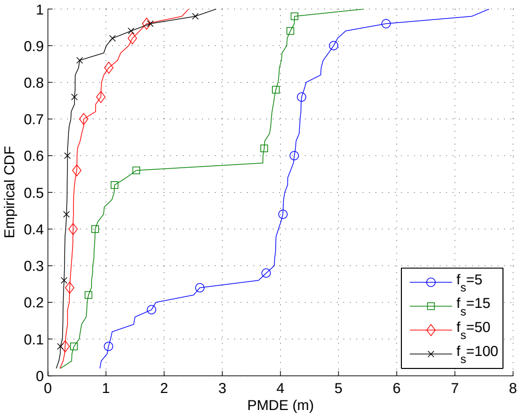


Figure 6.37: Empirical CDF of the PMDE metric for the cross trajectory after minimization.

6.6 Applying the Drifting Point Model in a Experimental Scenario

For analyzing the performance of the proposed model and comparing it with the standard solution we have decided to run the same experiments as of Chapter 5 for the squared trajectory. The trajectory as well as the RSS observations remain the same. Concerning the algorithms, they were implemented as stated throughout this chapter. However, as we have seen previously in this chapter the parameters of the maneuver detector shall be tuned. Thus in the following setup we have decided to use the parameters in Table 6.4.

As we can see in Table 6.4, both the forgetting factor and the maneuver threshold were increased.

Table 6.4: Parameters used in the χ^2 maneuver detector when applied in the experimental setup.

Parameter	Value
Forgetting Factor	0.975
Threshold (%)	99.5

While increasing the forgetting factor smoothes out the signals of the detector, the threshold decreases the chances of detecting maneuvers.

Similarly to what was done in Chapter 5, the first evaluation regards a single run. Figure 6.38 shows the result obtained while running both the CV and DP model in the squared trajectory. As we can see, the DP model clearly outperforms the CV model in terms of the visual estimation of the trajectory. The results o the DP model show that this model can be seen as a piecewise linear curve fitting to the CV model. In fact this represents a major advantage of the model since it is possible to clearly identify segments in the movement of the target. These segments can be more easily used for instance for user activity estimation, which can be considerably important in several applications where localization estimation itself is not the final goal, but just important information to reach the desired goal.

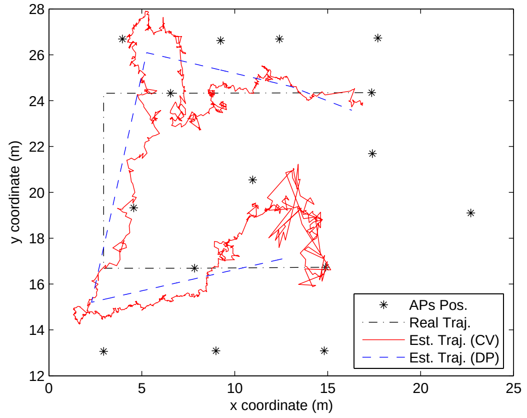


Figure 6.38: Comparison in a experimental setup between the CV and the DP model.

For a deeper insight on the performance of both methods, we have decided to plot the PMDE and the SRMSE for the squares trajectory in Fig. 6.39. Looking at the result for the squared trajectory we can see that for the PMDE metric the DP model outperforms the CV model. For the same trajectories, when the CDE error is compared, there are clear differences between the models. As we can see in Fig. 6.40, the DP model is clearly better than the CV model, i.e. the length of the estimated trajectory is closer to the true one.

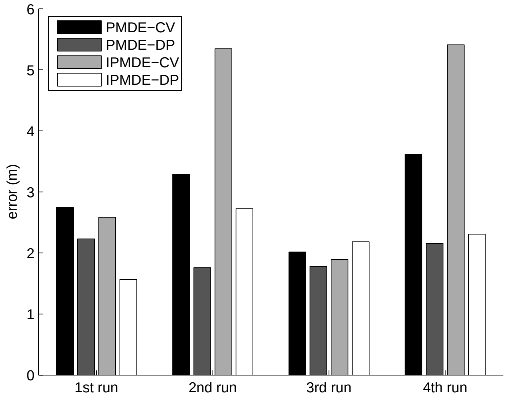


Figure 6.39: Performance comparison in a experimental setup between the CV and the DP model, in terms of the PMDE and SRMSE metrics.

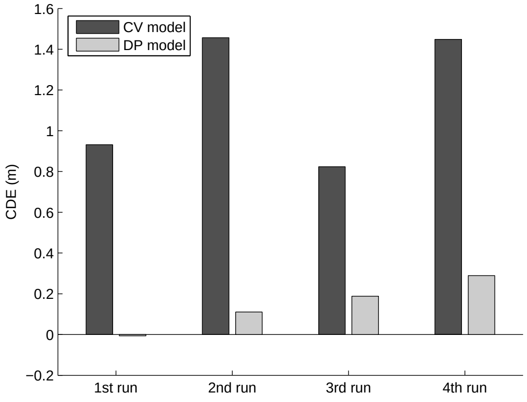


Figure 6.40: Performance comparison in a experimental setup between the CV and the DP model, in terms of the CDE metric.

6.7 Conclusions

In this section we have proposed an alternative model for the common CV model. The proposed Drifting Point model is based, similarly to the CV model, in a Bayesian filter. The key point of the proposed model is in the strategy used for estimating the trajectory. The model estimates the segments of linear movement instead of simply estimate point-wise kinetic states.

Together with the proposed model, the solution proposes a decoupling in the movement moments once maneuvers are detected. This idea goes against the Bayesian nature. However, due to the nature of the proposed model, the performance is still higher than in standard solutions. A possible future work subject could be to apply the proposed model while complying with the Bayesian recursive coupling of the estimators.

The very first conclusion obtained from the results is that the DP model estimates less noisy trajectories. This is due to the fact that the DP model does not present the pointwise noise characteristic of the CV model. This can be an important advantage for location base services, since trajectories are easily segmented.

Additionally we have seen that the DP model consistently shows better PMDE performance in a simulation framework than the CV model. In fact the gain of using the proposed solution can be as high as 20% or 30% depending on the scenario. We have see as well that the DP model consistently give better performance in terms of length of trajectory, i.e. the estimated trajectory length is closer to the true one in the DP model than it is in the CV model.

Chapter 7

Enhancing Localization Accuracy in Cellular Systems by Exploiting User-to-User ad-hoc links

Along the previous Chapter 5 and Chapter 6, localization techniques and enhancement methods were studied in short-range technologies supported by a networks of APs. Localization was performed using RSS readings obtained in the communication links between the WD and each one of the APs.

In the present chapter, localization is studied in a cooperative framework. Users in close proximity of each other are able to communicate with their peers and also communicate with an infrastructure as in Fig. 7.1. The communication between the users and the infrastructure is permitted by a long-range cellular network such as GSM or UMTS. The links among the users are ad-hoc short-range links such as Bluetooth of WLAN 802.11. As a cooperative scheme, the users are enforced to measure their long-range links and the links with the other users. This information is then shared in the network and a localization algorithm runs with all the information. The users are assumed to be all in range of each other and all communicate with all. In a scalable solution, these grouping of cooperative users would need to be addressed, though in the current chapter that is delimited. Additionally, the users are assumed to be static, i.e. no movement is experienced by any of the users.

Regarding the measurements used in the localization algorithms, it is assumed that the short-range links are a source of RSS measurements and the long-range links are a source of AoA and either ToA or TDOA. The type of measurements assumed in each type of link are chosen this way because they are those that carry more accurate information regarding the properties of the link (in a localization point of view). Based on the different types of measurements, three typical scenarios are assumed. The scenarios are typical scenarios for localization in cellular networks. The scenarios are identified by names previously proposed in the literature which do not consider cooperative schemes. Although the localization mechanisms used further in this chapter are not the same as in

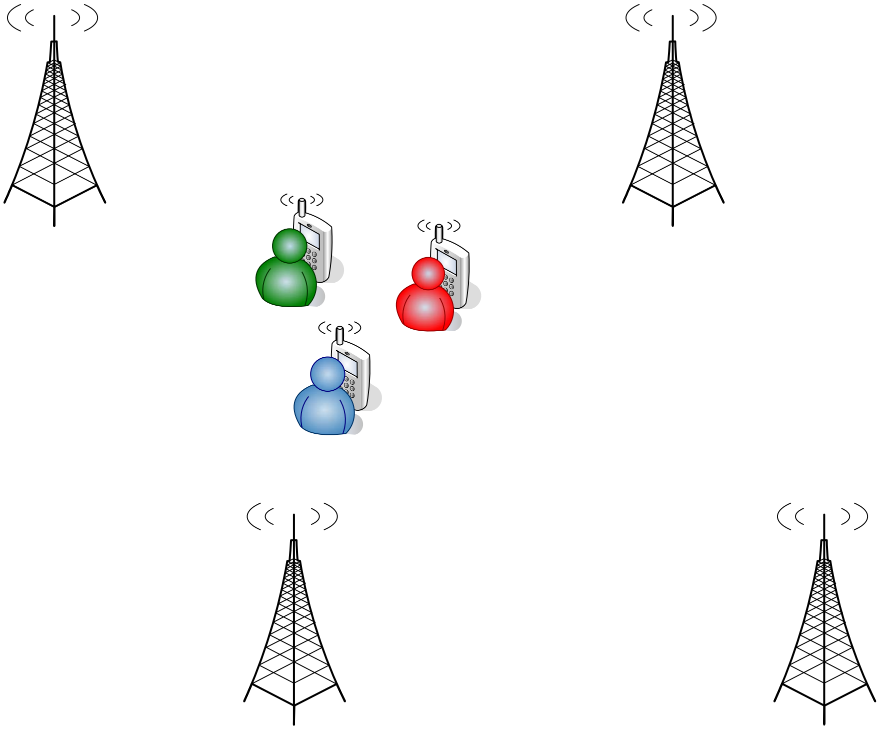


Figure 7.1: Mobile Cooperative Localization: Users in the close proximity able of communicating through short-range links cooperate in order to enhance the location accuracy of the overall cooperative users.

the literature, the same terminology is used to ease the task of referring to the combination of the different types of measurements involved in those methods.

- Hybrid TOA/AoA Positioning (HTAP) - Technique that uses a single BS able of capturing AoA and ToA measurements from the links with the WDs in its coverage range [DF00].
- Hybrid Line of Position (HLOP) - Technique that uses at least 3 BSs in order to to obtain measurements of ToA. Additionally it uses AoA only measured by the BS which holds the actual data communication with the WD [VCJ04].
- Hybrid Time Difference of Arrival (HTDOA) - This technique is similar to HLOP but instead of ToA measurements it uses TDOA measurements obtained from 4 different BSs [CH94]. Note that AoA measurements are also used the same way as in HLOP.

Note that this methods are presented in the literature as least squares approaches using non cooperative schemes. To study cooperative schemes, all the 3 scenarios are augmented with the RSS measurements obtained among the users. For both non-cooperative and cooperative schemes, the localization algorithms are not based in least squares as in the literature. Instead they are based in Bayesian frameworks, in particular Kalman family filters.

The evaluation of the proposed solutions is based on simulation tools that aim at simulate the propagation conditions in city scenarios. The metrics of performance are the mean of the RMSE of the localization estimators with respect to the true position of each cooperative user. This metric is often compared with the Federal Communications Commission (FCC) requirements [FCC99]. Additionally the performance of the cooperative schemes is compared with the performance of the non cooperative schemes.

The contributions of this chapter are two different algorithms for combining the several sources of measurements in a single framework for estimating the position of all the individual users. While in the first approach all the measurements are combined in a single estimation procedure, in the second approach, the short- and long-range are separately treated and combined in a subsequent step.

7.1 Statistical Models for the Propagation in Short- and Long-Range Channels

The present section explains the models used in Section 7.2 and Section 7.3. The models describe the propagation conditions and the statistical behavior of the measurements used in localization. In particular, the statistical models for the long-range measurements of ToA and AoA are taken from [DF00]. The differentiation between Line of Sight (LoS) and Non Line of Sight (NLoS) is based on the results concerning the LoS probability given by [WTN04]. From this same probability, it is also extracted the channel model used for the RSS measurements. Although the statistical models in [WTN04] are obtained from ray tracing simulation of the city of Bristol, the target technologies are the same used in the present work, i.e., the BS-WD and WD-WD links are studied respectively for the frequencies of 2.1 GHz (UMTS) and 5.2 GHz (802.11a).

7.1.1 Statistical Models for measurements of ToA and AoA

Fig. 7.2 shows the LOS probability as a function of the distance between the transmitter and the receiver, for both BS-WD and WD-WD links [WTN04]. It can be observed that when the distance between transmitter and receiver lies in the region of 10-20 m, which reasonably corresponds to the typical values of street width, the LOS probability is around 1. Additionally when the separation distance between transmitter and receiver increases, the LoS probability is higher for BS-WD links as compared to WD-WD links. This is because the transmitter, when it is a WD is placed at lower altitude than the BS, e.g. $1.5m$ instead of $15m$. Subsequently, the signal propagation paths are affected not only by the surrounding buildings in the vicinity of the transmitter but also affected by reflections in the terrain. As a consequence, the WD-WD links can also be NLoS even when no buildings lie in the propagation path.

For the generation of measurements, Fig. 7.2 is used in order to define the probabilities of being in LoS, given the distance between transmitter and receiver. Given this probability, random samples are generated out of the standard uniform distribution. If these samples are higher than the value dictated by Fig. 7.2, the connection is in LoS. Otherwise in NLoS.

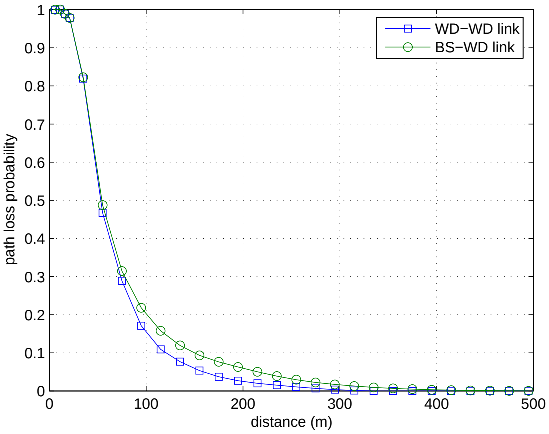


Figure 7.2: LOS probability vs. distance between transmitter and receiver.

LoS

Errors in the measurements are mainly due to equipment impairments. The traditional approach is to assume those errors normally distributed with zero mean and typical standard deviation values of $\sigma_t = \frac{30}{c}s$ (c is the speed of light) and $\sigma_\theta = 1$ degrees for ToA and AoA respectively [DF00].

NLoS

The estimation errors are large and primarily due to reflections or diffractions of the signal between a BS and a WD. As a consequence, the following exponential PDF can be used to model the excess delay, which holds information about the time dispersion that the channel induces [GEYC97]:

$$\mathrm{pr}(\tau) = \frac{1}{\tau_{rms}} \exp\Big(-\frac{\tau}{\tau_{rms}}\Big), \quad \tau_{rms} = \eta\Big\{d^{(i)[j]}\Big\}^{\varepsilon}\chi \tag{7.1}$$

where τ_{rms} is the root mean square (RMS) delay spread, η is the median value of the RMS delay spread in μs at 1 km, $d^{(i)[j]}$ is the distance between BS_j and WD_i in km, ε is an exponent with value between 0.5 and 1, and χ represents the log-normal shadow fading, so that $10\log\chi$ is a Gaussian random variable with zero mean $\mu_\chi = 0$ and standard deviation $\sigma_\chi = 4$ dB. The autocorrelation of the shadow fading among the WDs is modeled by:

$$\rho(\Delta X) = \exp\Big(-\frac{|\Delta X|}{d_{dec}} ln2\Big) \tag{7.2}$$

where ρ is the correlation coefficient, ΔX is the distance among the WDs, and d_{dec} is the decorrelation distance. In practice, if χ_1 is the log-normal component for the WD 1 at position X_1, the WD 2 at position X_2, which is ΔX away from X_1, is obtained by $\chi_2 = N(\mu_2, \sigma_2)$, where $N(\mu_2, \sigma_2)$ is a normal distribution with mean $\mu_2 = \rho(\Delta X)\chi_1$ dB and standard deviation $\sigma_2 = \sigma_\chi\sqrt{1 - \Big[\rho(\Delta X)\Big]^2}$ dB [S$^+$06]. The parameters η, ε and σ_χ are selected from four typical mobile channels [DF00] listed in table 7.1. The channel type that is used is the one describing the Urban scenario (B). The error in the angle is simulated by a zero mean Gaussian distribution with standard deviation given by:

Table 7.1: Parameters used in the generation of measurements of AoA and ToA for localization in cellular communications.

CHANNEL TYPE	η [μs]	ε	σ_χ [dB]
(A) Bad urban	1	0.5	4
(B) Urban	0.4	0.5	4
(C) Suburban	0.3	0.5	4
(D) Rural	0.1	0.5	4

$$\sigma_\theta = \frac{c\tau}{d^{(i)[j]}} \tag{7.3}$$

where τ is sampled from the excess delay expression in eq.(7.1)

7.1.2 Statistical Models for measurements of RSS

For determining whether a WD is in LoS with other or not, the strategy used for the long-range measurements in Section 7.1.1 is here applied as well.

Path loss

The following model is used for the mean outdoor path loss:

$$p = p_0 + 20\log(f) + 10\beta\log(d) \tag{7.4}$$

where d is the distance between transmitter and receiver in m, f is the operating frequency in MHz, and p_0 is the path loss in dBm at a short reference distance $d_0 = 1m$. Note that eq.(7.4) is equivalent to eq.(2.1). For WD-WD links, p can assume two models [WTN04]:

- in LoS it is assumed that $p_0 = -27.56$ dBm and $\beta = 2$ (free space path loss);
- in NLoS it is assumed that $p_0 = -51.22$ dBm and $\beta = 5.82$.

Shadowing

The shadowing process L_{shadow} is characterized by a lognormal distribution with a distance-dependent standard deviation σ_{shadow} given by:

$$\sigma_{shadow} = S\left[1 - \exp\left(-\frac{d-d_0}{D_S}\right)\right] \tag{7.5}$$

where S is the maximum standard deviation in dB, and D_S is the growth distance factor in meter. For WD-WD links, σ_{shadow} is modeled [WTN04]:

- in LOS with $S = 2$ dB, $D_S = 36m$, and $d_0 = 0m$;
- in NLOS with $S = 23.4$ dB, $D_S = 36m$, and $d_0 = 10m$.

Fast fading

The most popular models for fast fading in LOS and NLOS are respectively Rice and Rayleigh, where the ratio between the expected power of the dominant path ρ^2 and the power of the Rayleigh components $2\sigma^2$ is often expressed by the *Ricean K-factor*: $K_0 = \rho^2/2\sigma^2$. For WD-WD links, K_0 is modeled [WTN04]:

- in LOS by:

  $$K_0 = -n_{ff}d + L_{ff} \tag{7.6}$$

  with $L_{ff} = 23$ dB and $n_{ff} = 0.029$

- in NLOS by:

  $$K_0 = logN(\mu_{ff}, \sigma_{ff}) - 10 \tag{7.7}$$

  where $logN(\mu_{ff}, \sigma_{ff})$ is a lognormal distribution with mean $\mu_{ff} = 2.43$ dB and standard deviation $\sigma_{ff} = 0.45$ dB.

7.2 Localization of Static Devices Using the EKF for Data Association

In this section the cooperative EKF is presented. For comparison purposes the non-cooperative EKF can also be described as follow, given that only one WD is localized and subsequently RSS measurements are not used. Due to its straightforwardness, once the cooperative EKF is defined, the non-cooperative EKF is not explicitly presented.

7.2.1 Design of the Cooperative EKF

Before introducing the proposed EKF solution, it is necessary to define the mathematical models that describe the relation either (i) between the measurements and the position coordinates or (ii) the measurements and the distances between the nodes. Since several of the further expressions include the concept of distance between the nodes, the Euclidean distance between WD_i and BS_j is defined as:

$$\left\{\hat{d}^{(i)[j]}\right\}^2 = \left\{\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{[j]}\right\}^T \left\{\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{[j]}\right\} \tag{7.8}$$

and the Euclidean distance between WD_i and WD_j as:

$$\left\{\hat{d}^{(i)(j)}\right\}^2 = \left\{\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\right\}^T \left\{\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(j)}\right\} \tag{7.9}$$

Note that while the superscript (i) identifies the WD i, the superscript $[i]$ identifies the BS i. For both measurements, ToA and TDOA, it is used the law that describes the speed of light in vacuum. Thus, from eq.(2.3) we get ToA measurements given by:

$$\hat{t}^{(i)[j]} = \mathcal{H}_t\Big(\hat{d}^{(i)[j]}\Big) = \frac{1}{c}\hat{d}^{(i)[j]} + t_0 \tag{7.10}$$

and for eq.(2.4), TDOA given by:

$$\hat{T}^{(i)[j]} = \mathcal{H}_T\Big(\hat{d}^{(i)[j]} - \hat{d}^{(i)[1]}\Big) = \frac{1}{c}\Big\{\hat{d}^{(i)[j]} - \hat{d}^{(i)[1]}\Big\} \tag{7.11}$$

where $c = 3 \times 10^8$ m/s is the speed of light, t_0 is the time instant at which BS_j begins transmission. For the AoA, we use the trigonometric function in eq.(2.5) to relate angles and coordinates:

$$\hat{\theta}^{(i)} = \mathcal{H}_\theta\Big(\hat{\mathbf{x}}^{(i)}\Big) = \text{atan}\frac{\hat{y}^{(i)} - y^{[1]}}{\hat{x}^{(i)} - x^{[1]}} \tag{7.12}$$

For the RSS measurements, it is used the free space path loss as given by eq.(2.1):

$$\hat{p}^{(i)(j)} = \mathcal{H}_p\Big(\hat{d}^{(i)(j)}\Big) = -27.56 + 20\,log(f) + 10\beta\,log\Big[\hat{d}^{(i)(j)}\Big] \tag{7.13}$$

In order to model the filter, it is required to primarily define the hidden and observable vectors as shown in Chapter 3. Since it is necessary to ensure cooperation among all the devices, the state space to be estimated has to include the coordinates of all the devices:

$$\hat{X}_k = \left[\Big\{\hat{X}_k^{(1)}\Big\}^T \ldots \;\Big\{\hat{X}_k^{(n)}\Big\}^T\right]^T, \quad \hat{X}_k^{(i)} = \Big[\hat{x}_k^{(i)} \quad \hat{y}_k^{(i)}\Big]^T \tag{7.14}$$

From now on, it is assumed that the networks is composed by N BSs and n WDs.
Concerning the measurements, they are ToA (or TDOA), AoA and RSS. For the ToA, TDOA and RSS measurements, it is possible to define a vector of measurements for each WD i:

$$\mathbf{t}_k^{(i)} = \Big[t_k^{(i)[1]} \ldots \; t_k^{(i)[N]}\Big]^T, \tag{7.15}$$

$$\mathbf{T}_k^{(i)} = \Big[T_k^{(i)[2]} \ldots \; T_k^{(i)[N]}\Big]^T, \tag{7.16}$$

and

$$\mathbf{P}_k^{(i)} = \Big[p_k^{(i)(1)} \ldots \; p_k^{(i)(i-1)} \quad p_k^{(i)(i+1)} \ldots \; p_k^{(i)(n)}\Big]^T. \tag{7.17}$$

Note that, since for each WD there is only a single BS performing measurements of AoA, there is no need to specify a vector like done in the previous expressions. The full set of measurements is now obtained by integrating all the measurements of the same kind in the same vector, and later integrate all the several types of measurements in a single vector Z. Thus, respectively for ToA, TDOA, AoA and RSS:

$$\mathbf{t}_k = \left[\left\{\mathbf{t}_k^{(1)}\right\}^T \dots \left\{\mathbf{t}_k^{(n)}\right\}^T\right]^T, \tag{7.18}$$

$$\mathbf{T}_k = \left[\left\{\mathbf{T}_k^{(1)}\right\}^T \dots \left\{\mathbf{T}_k^{(n)}\right\}^T\right]^T, \tag{7.19}$$

$$\Theta_k = \left[\theta_k^{(1)[1]} \dots \theta_k^{(n)[1]}\right]^T, \tag{7.20}$$

and

$$\mathbf{P}_k = \left[\left\{\mathbf{P}_k^{(1)}\right\}^T \dots \left\{\mathbf{P}_k^{(n)}\right\}^T\right]^T. \tag{7.21}$$

Finally, the full measurements vector Z is defined according to the type of available measurements. For the case of RSS-aided hybrid TOA/AOA, Z is:

$$Z_k = \left[\left\{\mathbf{t}_k\right\}^T \quad \left\{\Theta_k\right\}^T \quad \left\{\mathbf{P}_k\right\}^T\right]^T \tag{7.22}$$

whereas for the case of RSS-aided hybrid TDOA/AOA, it is:

$$Z_k = \left[\left\{\mathbf{T}_k\right\}^T \quad \left\{\Theta_k\right\}^T \quad \left\{\mathbf{P}_k\right\}^T\right]^T \tag{7.23}$$

After defining the state space and the measurements space, it is necessary to specify the motion model and the perceptual model to be used by the EKF. By assuming that the WDs are static and that there is no external excitation, the new predicted state $\hat{X}_{k|k-1}$ in Eq. (3.20) is equal to the previous state:

$$\hat{X}_{k|k-1} = \mathcal{F}\left(\hat{X}_{k-1|k-1}, 0, 0\right) = \hat{X}_{k-1|k-1} \tag{7.24}$$

Concerning the perception model, it is simply necessary to obtain the vector $\hat{Z}_k$ based on the models defined by Eq. (7.10), Eq. (7.11), Eq. (7.12) and Eq. (7.13):

$$\hat{Z}_k = \mathcal{H}_\star\left(\hat{X}_{k|k-1}\right) \tag{7.25}$$

where the entries of $\hat{Z}_k$ are calculated using the model $\mathcal{H}_\star(\bullet)$, which is an entry-wise function that denotes for each entry of $\hat{Z}_k$ the appropriate function $\mathcal{H}_t(\bullet)$, $\mathcal{H}_T(\bullet)$, $\mathcal{H}_\theta(\bullet)$ or $\mathcal{H}_p(\bullet)$. Another important step in the design of the EKF is the determination of the process and measurement noise covariance matrices. By assuming that the WDs are static, it implies that no process noise exists and subsequently Q would be equal to the zero matrix. However, in order to allow a faster convergence of the filter, a process noise adaptation method (e.g. annealing) is used in order to allow faster convergence of the filter. The adaptation parameters of the method are adjusted by tuning. Concerning the measurement noise, we consider R as a diagonal matrix (no correlation between measurements):

$$R = \begin{bmatrix} \sigma^2_{TOA/TDOA} I & 0 & 0 \\ 0 & \sigma^2_{AOA} I & 0 \\ 0 & 0 & \sigma^2_{RSS} I \end{bmatrix} \tag{7.26}$$

where the variances $\sigma^2_{TOA}/\sigma^2_{TDOA}$, σ^2_{AOA} and σ^2_{RSS} are derived from monte carlo simulations of the models defined in Section 7.1. Two different approaches were used in order to set these values:

- The position of the WD was set as constant, and the standard deviations were determined after running the monte carlo simulation. This approach represents the case when a-priori information concerning the noise statistics exists.
- The position of the WD was changed for every new run of the monte carlo simulation. This case represents no a-priori information concerning the noise statistics.

In order to completely model the filter, it is still missing to define several of the matrices in Eq. (3.27) and Eq. (3.28). The matrices A_k and H_k represent respectively the Jacobian of Eq. (7.24) and Eq. (7.25) in order to X_k. Since the WDs are static, A_k is equal to the identity matrix.

7.2.2 Performance Analysis

For evaluating the performance of the simulated approaches, it was decided to use the parameters stated in Table 7.2. The cell radius as well as the cluster radius represent a typical scenario for respectively the cellular and the sort-range links. The distance of decorrelation was assumed 20m. This means that for distances higher than $20m$, there is a weak correlation ($< .5$) between the cellular measurements observed by the users. For the first evaluation of the cooperative localization solution presented in this chapter, it was decided to plot the CDF curves for the three types of methods. Each curve was obtained out of 5000 independent runs of the same setup and for each run 200 samples of each measurement were generated. As a reference we will look specifically at the 67% and 95% percentiles of the RMSE defined by the FCC requirements [FCC99]. The FCC mandate requires localization in cellular networks to be at least $150m$ and $300m$ for 67% and 95%

Table 7.2: Parameters used in the simulation of the ad-hoc cellular system

Parameter	Value
Cell Radius (m)	1000
Cluster Radius (m)	50
(x, y) (m)	$(500, 0)$
d_{dec} (m)	20
Numb. BS	$\{3, 5\}$
Numb. WD	$\{2, 4, 6, 8, 10\}$

of the cases, respectively. These references are commonly used as standard references in the literature. Figure 7.3 shows the CDF curve obtained for the HTAP method. Several curves were plotted

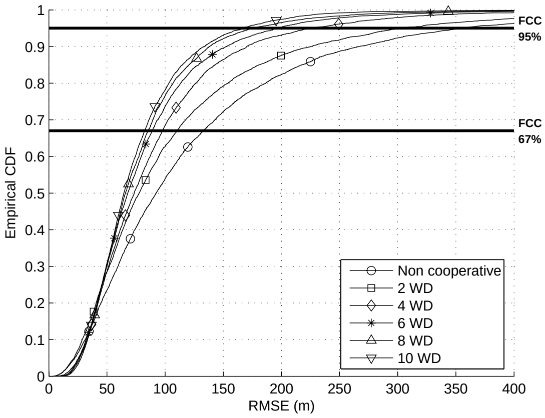


Figure 7.3: Localization results for the HTAP for 3 BSs

for different number of cooperative devices, as well as for the non cooperative case. As Figure 7.3 show, cooperation between users brings a considerable improvement in accuracy regarding the RMSE metric. Additionally, the higher the number of cooperative devices, lower is the increment in accuracy. It is also possible to see that cooperation helps on meeting the FCC requirements concerning position accuracy. For the case of the HLOP and HTDOA methods, similar results

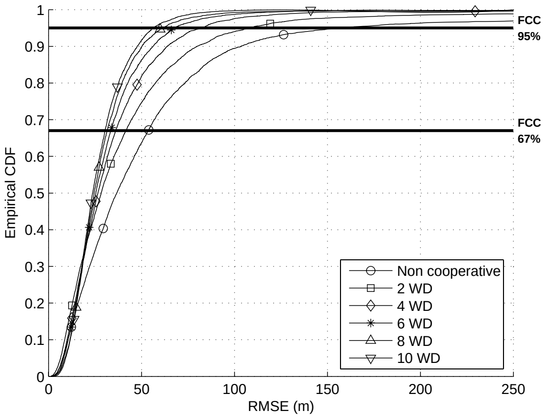


Figure 7.4: Localization results for the HLOP for 3 BSs

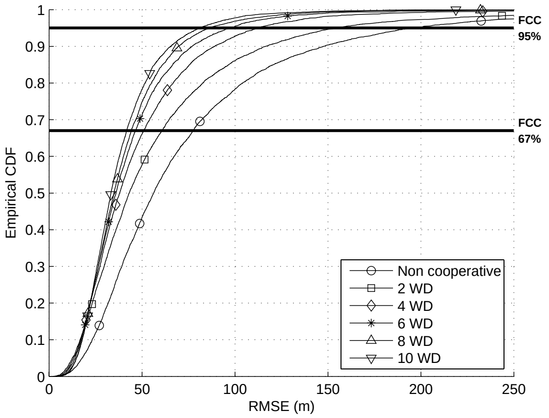


Figure 7.5: Localization results for the HTDOA for 3 BSs

are plotted in Fig. 7.4 and Fig. 7.5 respectively. According to both plots, the results concerning the effect of cooperation in the localization accuracy are equivalent. While comparing the three methods of localization, HLOP is the one with best performance concerning both of the 67% or 95% error percentile. On the other hand, HTAP is the one with the worst performance, because of the lower amount of information used for localization when compared with the other methods.

Although HTAP presents the worst accuracy, it is expected to outperform in terms of management, i.e. since the amount of information to be transferred among the different nodes is lower and the sources and destinations of information are less, the transactions of that information are simpler to execute. Contrarily, the HLOP method, besides giving the best performance, it is the most complex to implement, since BS and WD need to be synchronized (as it was explained in Chapter 2).

In order to quantify the accuracy of each one of the methods with respect to the percentiles defined by the FCC, the RMSE for each one of those percentiles is evaluated. In order to define the best and worst case scenario, we consider two approaches concerning the observation noise: (i) for the best case it is obtained from the empirical statistics of the measurements of each channel given the position of the WDs; (ii) for the worst case, the process noise is obtained from a monte carlo simulation independent on the position, i.e. the position was different for each run. Subsequently, in the second case, the noise statistics are constant. While the first case approximates the case when a-priori information on the scenario is available, the second case assumes no a-priori information. For the HTAP method, Fig. 7.6 shows the plot of the RMSE when there is no a-priori knowledge concerning the observation noise. In Fig. 7.7 it is shown the similar results but when there is no a-priori knowledge regarding the observation noise. As Fig. 7.6 and Fig. 7.7 show, when the observation process noise is known, cooperation can bring considerably higher improvement than in the case where no a-priori information is available. Regarding the RMSE metric for the HLOP method plotted in Fig. 7.8 and in Fig. 7.9, it is possible to see that the performance is considerably penalized when the noise statistics are unknown, i.e. when there is no a-priori information regarding the noise statistics. When looking at the results for both HTAP and HLOP methods, cooperation brings higher incremental performance (relative to the corresponding non cooperative scenario) for the HLOP method. Additionally, as shown in Fig. 7.8, the number of BSs used in the HLOP localization solution has an impact in accuracy, i.e. the higher the number of BSs, higher is the performance. For the HTDOA method (Fig. 7.10 and Fig. 7.11), the results are slightly different

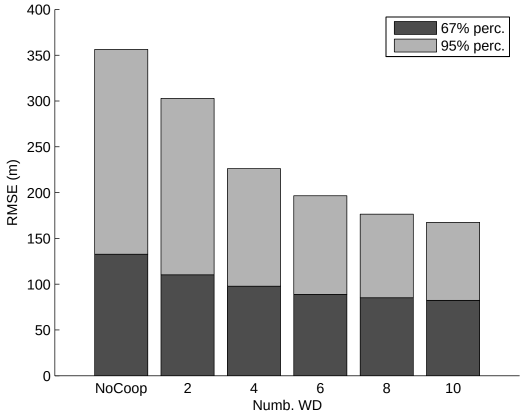


Figure 7.6: Performance for the HTAP when σ is known

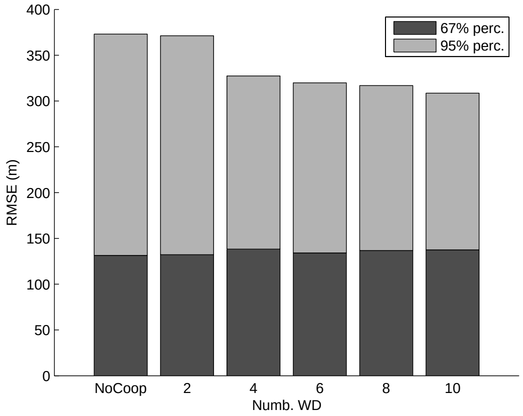


Figure 7.7: Performance for the HTAP when σ is constant.

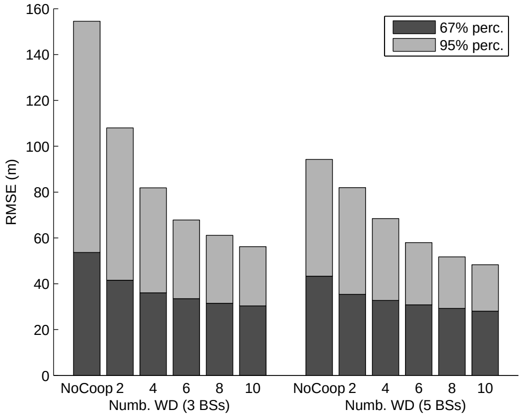


Figure 7.8: Performance for the HLOP when σ is known

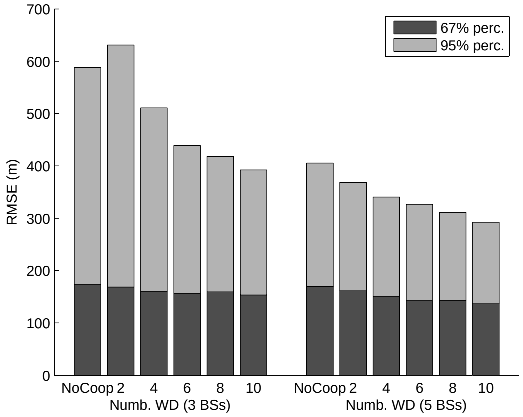


Figure 7.9: Performance for the HLOP when σ is constat.

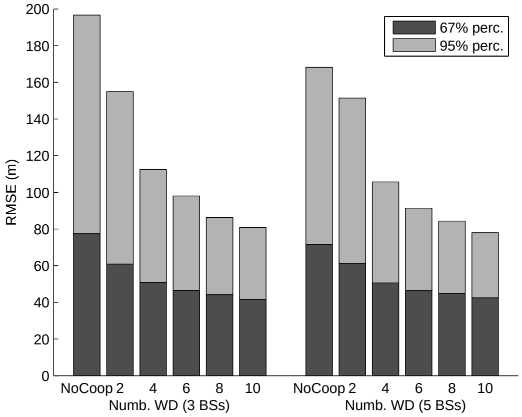


Figure 7.10: Performance for the HTDOA when σ is known

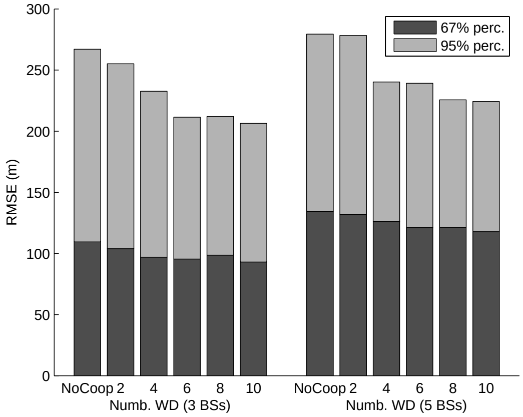


Figure 7.11: Performance for the HTDOA when σ is constant.

from the two previous cases. First, the pre acquired knowledge on the noise statistics has no such high impact in the performance as in the HLOP method, but still higher than for the HTAP case. Then, it is possible to see in Fig. 7.11 that the number of BSs does not have considerable impact in the performance when noise statistics are unknown.

To quantify the performance of the cooperative schemes in comparison to the standard non cooperative case, a metric of gain, similar to the one in eq.(6.12), is defined for each one of the percentiles considered by FCC. The gain for the $\mathfrak{a}\%$ percentile is defined as:

$$G_{\mathfrak{a}} = \frac{\mathfrak{z}_{\mathfrak{a}}^{nc} - \mathfrak{z}_{\mathfrak{a}}^{c}}{\mathfrak{z}_{\mathfrak{a}}^{nc}} \tag{7.27}$$

where $\mathfrak{z}_{\mathfrak{a}}^{c}$ and $\mathfrak{z}_{\mathfrak{a}}^{nc}$ are the inverse CDF for the percentile $\mathfrak{a}\%$ respectively in the cooperative and non-cooperative case. Again, for this evaluation both cases concerning the observation noise are presented. For the 67%, when the noise statistics are known, it is possible to see in Fig. 7.12 that the gain of cooperation is similar to all the 3 tested methods. Concerning the number of cooperative devices, the gain is higher for higher number of devices. Contrarily, when the noise statistics are not known (Fig. 7.13), the gain presents different behavior, where for some cases cooperation can even degrade performance. For instance, looking at the HTAP method, cooperation does not show improvement in accuracy for the 67% percentile. Furthermore, the relation with the number of cooperative users is not monotonous as in Fig. 7.12 for any of the tested methods. Regarding the equivalent results for the 95% percentile, Fig. 7.14 shows a higher gain than for the case evaluating the 67% percentile. Furthermore, contrarily to what Fig. 7.13 shows, in Fig. 7.15 it is possible to see that for the 95% percentile the gain is monotonous with respect to the number of cooperative users. When no a-priori knowledge exists concerning the noise statistics (Fig. 7.15), the gain values for the 95% percentile are generally better than those in Fig. 7.13.

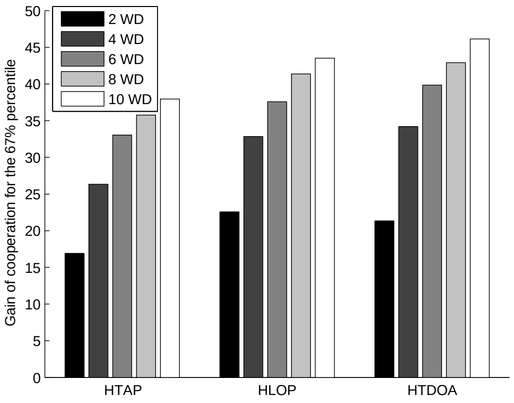


Figure 7.12: Gain of cooperation for the 67% percentile with σ known

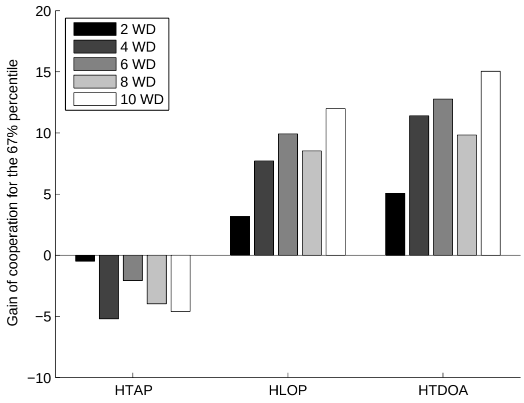


Figure 7.13: Gain of cooperation for the 67% percentile with σ unknown

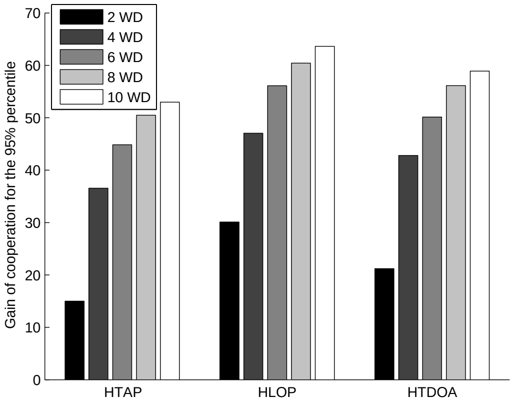


Figure 7.14: Gain of cooperation for the 95% percentile with σ known

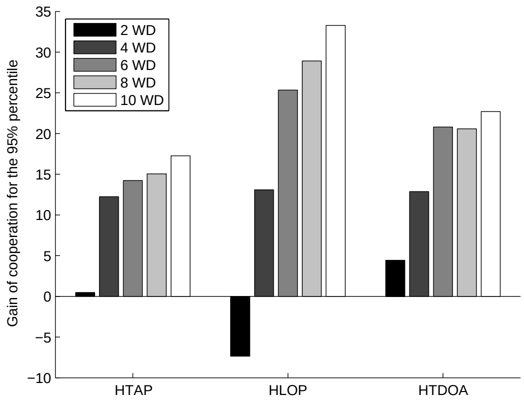


Figure 7.15: Gain of cooperation for the 95% percentile with σ unknown

As a conclusion, cooperation improves accuracy in cellular systems by typically values of 20% to 60% depending on the number of cooperative users and the method of localization used. Even for the case when no a-priori knowledge exists concerning the measurement noise statistics, cooperation generally improves performance. Additionally, as it is expected, when pre acquired information is available, not only the accuracy is higher, but also the gain by cooperation is higher. It is important to mention that no a-priori information is not the most common scenario, because there are nowadays several solutions for gathering information concerning the propagation conditions in a determined scenario.

7.3 The Two-Level Data Fusion for Cluster Mobility and Individuals Mobility

Due to the low erroneous information obtained in the short-range links when compared to the information obtained in the long range links, an idea of decoupling short- and long-range measurements is presented in this section.

The proposed solution consists in decoupling relative localization and absolute localization. The short-range ad-hoc links among the users are used to perform relative localization and the long-range cellular links are used to perform the absolute localization of the overall group. Figure 7.16 shows a schematic representation of the algorithm. As it is possible to see in Fig. 7.16, the relative localization is done among the several cooperative mobiles by estimating distances between them,

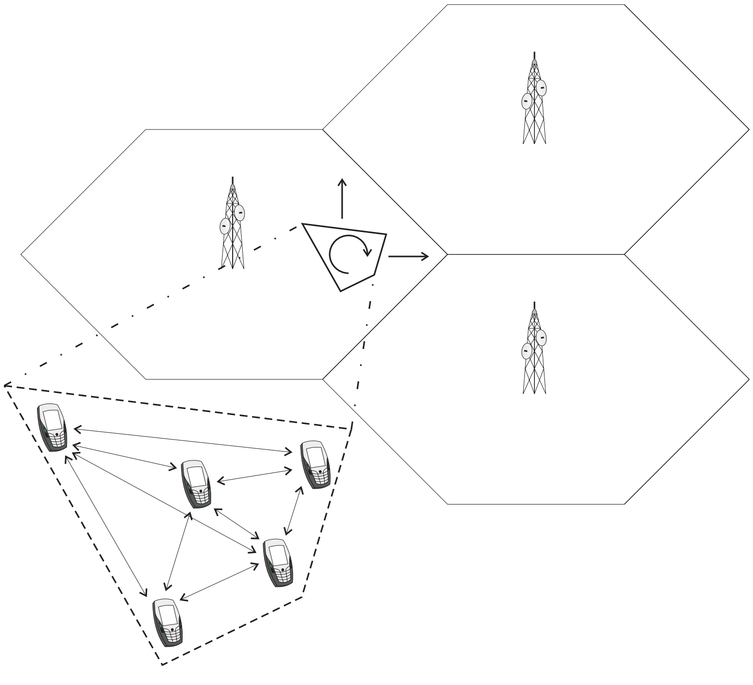


Figure 7.16: Schematic representation of the proposed algorithm. Note that the arrows represent quantities to be estimated.

while the absolute localization is done by estimating the coordinates and orientation of the group in the cellular network. The proposed algorithm, alike Kalman filter, runs in a cyclic way, where each iteration is performed when observations are available. The algorithm considers the following steps:

1. Decoupling the state space of the user estimated positions into (i) relative coordinates and (ii) center of mass coordinates, i.e. position of the group further defined as the position of the WD 1.
2. Single iteration of an extended Kalman filter for estimating relative location.
3. Single iteration of an unscented Kalman filter for estimating the coordinates of the center of mass.
4. Coupling estimations of relative and center of mass coordinates in order to have absolute estimators.

Although each iteration contains a certain number of different operations, some of these operations may be skipped depending on each kind of observations are available at each precise moment. For example, assuming that at a determined moment only short-range observations are available, the algorithm only runs the operations numbered as 1, 2 and 4. This feature allows short- and long-range systems to operate with different measurement rates.

Figure 7.17 shows an operational representation of the algorithm. Due to the decoupling and coupling operations, the algorithm is able to operate in the case when there is only one type

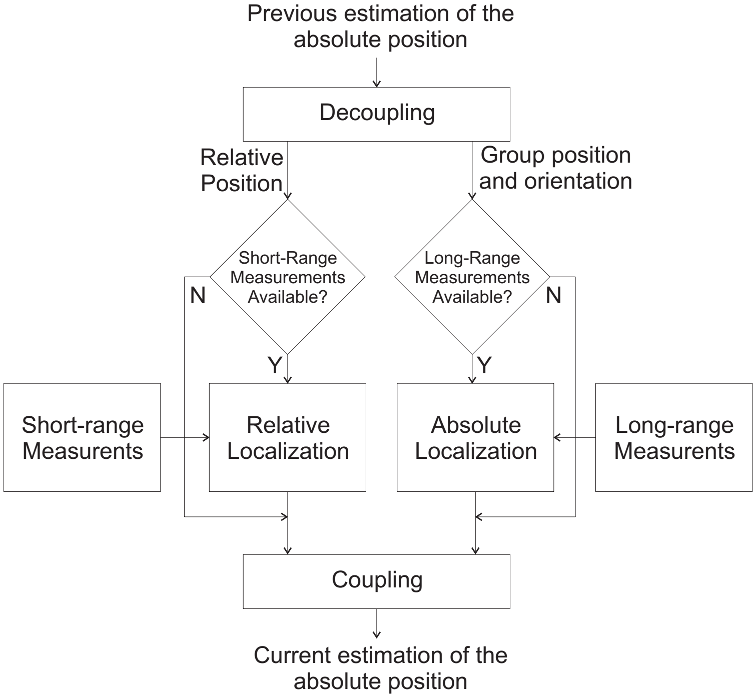

Previous estimation of the absolute position
Decoupling
Relative Position
Group position and orientation
Short-Range Measurements Available?
Long-Range Measurements Available?
N
N
Y
Y
Short-range Measurents
Relative Localization
Absolute Localization
Long-range Measurents
Coupling
Current estimation of the absolute position

Figure 7.17: Operational representation of the proposed algorithm.

of measurements (either short- or long-range measurements) and also in the case when both are available. In the following sections, the blocks in Fig. 7.17 will be explained in detail.

7.3.1 Decoupling and Coupling

Decoupling the absolute coordinates in relative coordinates and group position and orientation is easily done by applying a transformation of coordinates.

Transformation of coordinates

In order to transform the coordinates, it is necessary to obtain the Current Transformation Matrix (CTM) that corresponds to a translation followed by a rotation of the axis.

Let one assume that the absolute coordinates for WD i are defined by $X^{(i)abs}$ and that the relative coordinates $X^{(i)rel}$ (relative to the position of WD 1 and WD 2 as further described). Additionally, assuming T_{ctm} as the CTM for the transformation of coordinates, the following relation can be written:

$$\begin{bmatrix} X^{(i)rel} \\ 1 \end{bmatrix} = T_{ctm} \begin{bmatrix} X^{(i)abs} \\ 1 \end{bmatrix} \tag{7.28}$$

where the number 1 is included to allow consideration of constant components in the transformation. Considering a translation of the quantity (a, b) followed by a rotation of angle θ, the CTM matrix can be defined as:

$$T_{ctm} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \tag{7.29}$$

Knowing that the translation is done of the quantity equivalent to absolute position of WD 1 and that the rotation is equivalent to the angle of the segment between WD 2 and WD 1 with respect to the absolute coordinate system, one can define a, b and θ as:

$$\begin{bmatrix} a \\ b \\ \theta \end{bmatrix} = \begin{bmatrix} x^{(1)abs} \\ y^{(1)abs} \\ \arctan\left(\frac{d_y^{(1)(2)abs}}{d_x^{(1)(2)abs}}\right) \end{bmatrix} \tag{7.30}$$

where

$$\begin{aligned} d_x^{(1)(2)abs} &= x^{(2)abs} - x^{(1)abs} \\ d_y^{(1)(2)abs} &= y^{(2)abs} - y^{(1)abs} \end{aligned} \tag{7.31}$$

The eq.(7.30) defines the center of mass of the group of cooperative WDs. Using eq.(7.30) in eq.(7.29), we can rewrite the CTM matrix as:

$$T_{ctm} = \frac{1}{d^{(1)(2)}} \begin{bmatrix} d_x^{(1)(2)abs} & -d_y^{(1)(2)abs} & 0 \\ d_y^{(1)(2)abs} & d_x^{(1)(2)abs} & 0 \\ 0 & 0 & d^{(1)(2)} \end{bmatrix} \begin{bmatrix} 1 & 0 & -x^{(1)abs} \\ 0 & 1 & -y^{(1)abs} \\ 0 & 0 & 1 \end{bmatrix} \tag{7.32}$$

Note that this transformation is not unique, meaning that other transformations are possible.

Decoupling of coordinates

When performing a decoupling of coordinates, the relative coordinates of the WDs can be obtained by using eq.(7.28) with the CTM of eq.(7.32), while the coordinates and orientation of the group is obtained by eq.(7.30). Since eq.(7.28) gives the relative positions of the WDs, one can see that, for WD 1 because of the fact that it is the reference in the relative coordinate system, it will be $(0^{rel}, 0^{rel})$. Additionally, since WD 2 is assumed to be in the x axis of the relative coordinate system, it will assume position $(x^{(2)rel}, 0^{rel})$. Thus, since the coordinates of the WD 1 and the y coordinate of the WD 2 define the relative coordinate system and subsequently they cannot have other value besides 0, they can be omitted in the definition of the state space in the relative localization step (see Section 7.3.2).

Coupling of coordinates

When performing the coupling, the absolute coordinate of the WDs is obtained by including the coordinates and orientation of the group in eq.(7.29) and perform the inverse calculation of eq.(7.28). When using the transformation of coordinates, the points $(0^{rel}, 0^{rel})$ for WD 1 and $(x^{(2)rel}, 0^{rel})$ for WD2 shall be integrated with the state space of the relative coordinates (which does not have those 0 valued coordinates).

7.3.2 Relative Localization

To perform an estimation of the relative localization of the WDs, an EKF was used. The choice of the filter was based on the fact that the system has a nonlinear nature which is not very accentuated, i.e., the linearization by itself does not introduce considerable error when compared to the noise introduced by the processes involved.
In order to design the EKF, one has to primarily define the hidden process and the observable process. For the present case, the hidden process is the set of relative positions, while the observations are the measured power values between the WDs.

$$X = \left[x^{(2)rel} \quad \left\{ X^{(3)rel} \right\}^T \quad \dots \quad \left\{ X^{(n)rel} \right\}^T \right]^T \tag{7.33}$$

$$Z = \left[\begin{array}{cccc} p^{(1)(2)}, & p^{(1)(3)}, & \dots & p^{(1)(n)}, \\ & p^{(2)(3)}, & \dots & p^{(2)(n)}, \\ & & \ddots & \vdots \\ & & & p^{(n-1)(n)} \end{array} \right]^T \tag{7.34}$$

Note that eq.(7.33) do not consider the coordinates of WD 1 and the y coordinate of WD 2 due to the fact explained in Section 7.3.1 and that eq.(7.34) is a column vector. Furthermore, it is assumed in eq.(7.34) that for each pair of WDs i and j there is a single observation $p^{(i)(j)}$ where $i < j$. Instead, if measurements are assumed to be made downlink and uplink between the pairs of WDs, the eq.(7.34) would need to be extended to consider observations $p^{(i)(j)}$ where $i \neq j$.

The second step on the design of the filter is the definition of the evolution and the observation models for the present case. Since it is assumed that WDs are static, the predicted state is simply equal to the previous estimation, while the observation model is defined by a propagation model such as the path loss equation:

$$\hat{X}_{k|k-1} = \hat{X}_{k-1|k-1} \tag{7.35}$$

$$\hat{Z}_k = \alpha - 20\beta \log(d) \tag{7.36}$$

Note that eq.(7.36) is equivalent to eq(2.1).

The final design decision is the definition of the process noise and measurement noise. Also the several components of the process and observation noise are assumed to be independent, i.e. matrices Q and R are diagonal matrices. Concerning the measurement noise, the standard deviation of the measurements was used in the diagonal entries of R.

$$R = \sigma_p^2 I \tag{7.37}$$

Although σ_p^2 in eq.(7.37) is defined based on observations, its value could be obtained from either literature or computer simulation or a pre-calibration phase. The actual values used for σ_p^2 are derived from a monte carlo simulation of the propagation models introduced in Section 7.1, given the position of the transmitter and receiver. Concerning the process noise Q, it can be assumed zero, since WDs are static. However, the same annealing approach used in Section 7.2 is also used in the current relative localization context. The adaptation parameters of the method are adjusted by tuning.

7.3.3 Absolute Localization

In opposite to what happens in the Section 7.3.2, the absolute localization has strong nonlinear behavior due to the estimation of the orientation. Since the orientation is related with position coordinates by an *atan* function, the linearization introduces errors in the process estimation of similar magnitude to the process state itself. For this reason, an UKF is used for estimating the location of the group of WDs in the cellular system.

As in the EKF, in the UKF the sate space of the hidden process and the observable process has to be defined. For absolute localization, the states to be estimated are those shown in eq.(7.30).

$$X = \begin{bmatrix} a & b & \theta \end{bmatrix}^T \tag{7.38}$$

For the observable process, measurement are obtained from two sources: ToA (or equivalently TDOA) and AoA. Thus,

$$Z = \left[\mathbf{t}^T, \quad \mathbf{A}^T \right]^T \tag{7.39}$$

where,

$$\mathbf{t} = \left[\left\{ \mathbf{t}^{(1)} \right\}^T \quad \left\{ \mathbf{t}^{(n)} \right\}^T \right]^T \tag{7.40}$$

$$\mathbf{t}^{(i)} = \left[t^{(i)[1]} \quad \dots \quad t^{(i)[n]} \right]^T \tag{7.41}$$

$$\Theta = \left[\theta^{(1)[1]} \quad \dots \quad \theta^{(n)[1]} \right]^T \tag{7.42}$$

for the definition of the hidden and the observable models, it is recalled the assumption that WDs are assumed static, meaning that predicted state equals the previously estimated state.

$$\hat{X}_{k|k-1} = \hat{X}_{k-1|k-1} \tag{7.43}$$

The perception model is in fact a set of two expression, one for the AoA and another for the ToA.

$$t^{(i)[j]} = \frac{d^{(i)[j]}}{c} \tag{7.44}$$

$$\theta^{(i)[1]} = \arctan \left(\frac{\hat{y}^{(i)} - y^{[1]}}{\hat{x}^{(i)} - x^{[1]}} \right) \tag{7.45}$$

Concerning the noise covariance Q of the hidden process, it can be used an approach similar to the one in Section 7.2 and in Section 7.3.2, i.e. an initial value is set for the noise covariance and then the annealing method is applied in order to incrementally reduce that noise along time. On the other hand, for the noise covariance R of the observable process, it is used the following expression:

$$R = \left[\begin{array}{c|c} \sigma_t^2 I & 0 \\ \hline 0 & \sigma_\theta^2 I \end{array} \right] \tag{7.46}$$

where I is the identity matrix of proper size. The parameters σ_t^2 and σ_θ^2 are respectively the standard variation of the ToA and AoA measurements. The value is, similarly to Section 7.3.2 obtained from a monte carlo simulation of the propagation models given the distance between transmitter and receiver.

7.3.4 Performance Analysis

For analysing the performance of the just proposed solution, it was made an evaluation of the RMSE for the three considered localization techniques. Again, the number of cooperative WD is assumed $n = \{2, 4, 6, 8, 10\}$. For the HTAP technique, 5000 different runs of the same experiment for each value n were performed and 200 measurements of each type were generated for each run. The resulting CDF plotted in Fig. 7.18 shows a major difference concerning the cooperative case when n equals 2. For the case with $n = 2$ the RMSE is around $170m$ for 67% of the cases. In opposite, for the same percentile, the RMSE for $n = \{4, 6, 8, 10\}$, the results lie between $50m$ and $100m$. This discrepancy is due to the measurements of ToA. Since the whole cluster is estimated as a single point, high error on the measurements greatly impact the results. This effect looses importance when the number of cooperative nodes increases, since errors in measurements are likely to cancel each other out. Although for HTAP the number of WDs overcomes this effect, for HLOP (Fig. 7.19), the effect has a stronger impact, i.e. the errors are higher than for HTAP. On the other hand, when the technique is HTDOA this destructive effect is again only seen for $n = 2$

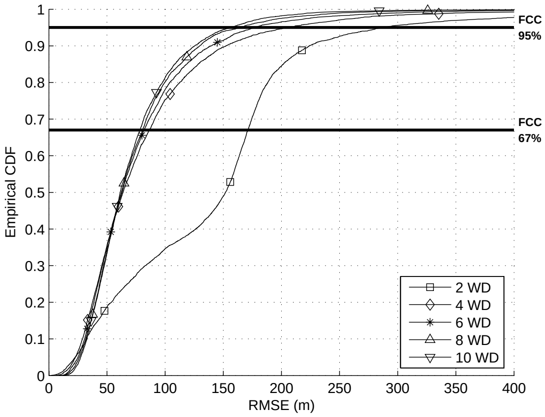


Figure 7.18: Empirical CDF of the RMSE metric for the HTAP localization technique, when the proposed 2LKF is used for data fusion.

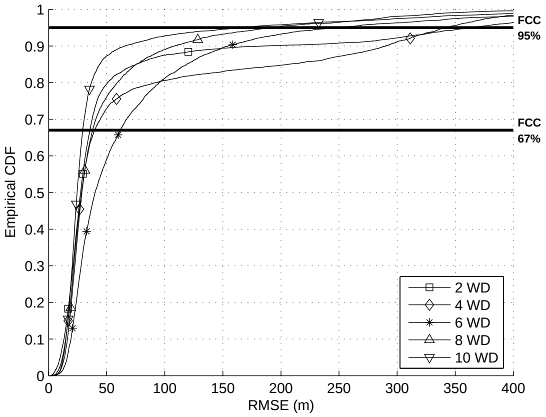


Figure 7.19: Empirical CDF of the RMSE metric for the HLOP localization technique, when the proposed 2LKF is used for data fusion.

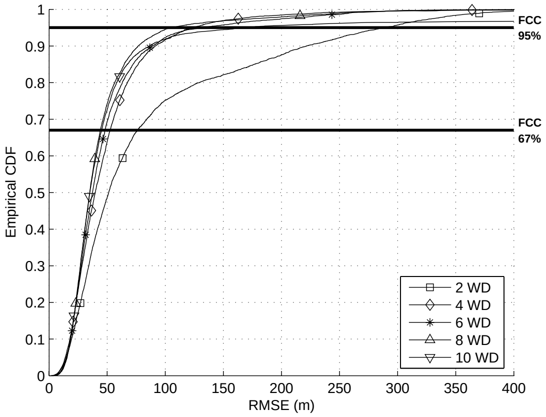


Figure 7.20: Empirical CDF of the RMSE metric for the HTDOA localization technique, when the proposed 2LKF is used for data fusion.

(Fig. 7.20). For the HTDOA case, the fact that the method uses TDOA measurements makes it more robust, i.e. since the measurements are differences of biased times of arrival, it is likely to happen that errors cancel each other out. For a more practical evaluation of the results and a comparison with the previously presented solutions, the gain of each technique was plotted versus the number of cooperative WDs. As Fig. 7.21 shows for the 67% percentile, the gain strongly depends on the number of cooperative WDs. For both HTAP and HTDOA the gain is at least 30% lower than for the cases with more than 2 WDs. Instead, for n higher than 2, the gain of th 2LKF is generally between 30% and 40%. On the other hand, for the HLOP method, the gain presents higher variations, being the worse case when $n = 6$. When comparing Fig. 7.21 and Fig. 7.12, the gain shows comparable magnitude for the EKF and the 2LKF, when any of the HTAP or HTDOA methods is used. Instead, for the HLOP method the result is poorer when the 2LKF algorithm is used. For the percentile of 95%, the results plotted in Fig. 7.22 comply with the comparison between EKF and 2LKF. The major difference concerns again the HLOP method. For the 2LKF, the HLOP method presents a negative gain for any of the n tested number of cooperative WD.

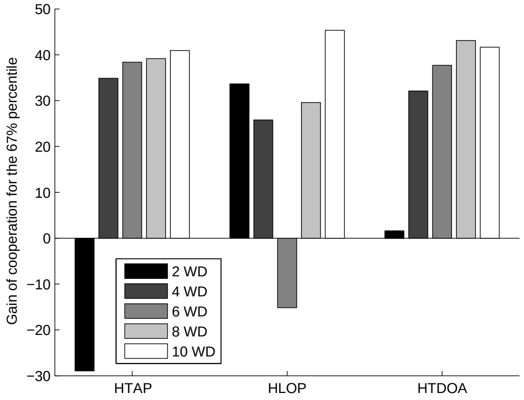


Figure 7.21: Gain of the 2LKF cooperative algorithm with respect to the non-cooperative method for 67% of all the tested cases

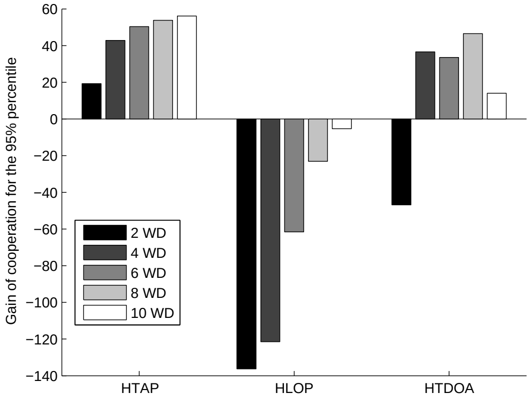


Figure 7.22: Gain of the 2LKF cooperative algorithm with respect to the non-cooperative method for 95% of all the tested cases

7.4 Conclusions

In this chapter, a common scenario where users with two technologies are able to communicate among each other through short-range links or with a wired network by cellular links. The purpose was to study cooperative localization in cellular networks envisaging a gain in accuracy by using cooperative schemes. The users, in a cooperative fashion, share information of their cellular and their ad-hoc links in order to reach an enhancement on position accuracy for all the cooperative WDs. The cellular and ad-hoc links were simulated in such a ways that measurements of ToA, TDOA, AoA and RSS are statistically generated. Analysis consisted in evaluating the available solutions for the cellular system while localizing the users as individuals and further comparison with cooperative schemes. An EKF solution and a proposed 2LKF where the two candidates. The results have shown that cooperation generally enhances localization accuracy of all the devices, meaning for certain cases a gain of around 40%. It was shown that, as expected, when a-priori knowledge concerning the statistics of the measurements exists, the localization accuracy presents considerably better performance according to the RMSE metric. It was also shown that when the EKF is used for implementing cooperation, the HLOP was the localization method with higher accuracy if there exists a-priori knowledge concerning the observation noise. Contrarily, when no a-priori knowledge exists, the same HLOP method was the one with worst performance.

Additionally, this chapter proposed a different approach for cooperative localization. The method named 2LKF consists in separating the long-range links and short-range links. While the short-range links allow relative localization in a cluster-wise fashion, the cellular links allow the localization of the whole cluster as a single entity. This solution presents the advantages that is more scalable in terms of the number of cooperative WDs, but also requires management of the clusters. This matter has not been considered in the dissertation. Additionally the 2LKF solution lows down the size of the matrix calculations, when compared with the EKF. Concerning RMSE accuracy, the proposed solution presents a performance comparable with the EKF for the HTAP and the HTDOA methods. Contrarily for the HLOP method, the EKF considerably outperforms the 2LKF with the latter presenting worse performance than the standard non cooperative case.

Concerning the applicability of the proposed cooperative schemes, the EKF-based solution is adequate when the link length in the cellular links are measured using ToA or TDOA. The 2LKF is more adequate for TDOA. Additionally, if the number of cooperative devices is high, the 2LKF solution shall be used since it is more scalable.

Part IV

Conclusions

Chapter 8

Summary and Conclusions

This dissertation has focused on methods for obtaining location information in wireless communications networks. The work was conducted in three major directions. First the latency time of obtaining location information in wireless networks was studied in a use case of short-range ad-hoc networks. Second, location accuracy was studied in short-range wireless networks. The problems were identified and solutions for those problems were proposed. Third, cooperative schemes of localization were proposed in short-range aided cellular networks.

8.1 Conclusions

Latency time is studied in short-range ad-hoc links. The mechanisms where location information can be obtained from, usually run during the establishment of new connections. The reason is that at that time, power control mechanisms are not yet active, but it is still possible to measure received signal from the wireless channel. Since the mechanisms to establish connection in wireless communications operate according to randomized link layer protocols, the latency time presents a stochastic behavior. As a use case of such short-range ad-hoc networks, Bluetooth technology was evaluated. A rigorous study of the Bluetooth specifications V2.0 was done and the latency time was evaluated during the inquiry procedure, i.e. the procedure in Bluetooth which initiates new connections. It was verified that in fact the latency time has a stochastic behavior, which strongly depends on the number of devices attempting to establish connection. It was observed that the procedures that run pseudo-arbitrary mechanisms to avoid collision between communication packets have a deep impact on the latency time, when a single devices is being requested by several nodes for a connection. Thus, the statistics of the latency time are expected to vary depending on the number of nodes in the networks, in particular, for localization applications using signal strength due to the fact that multiple sources of data are required.

After evaluating the latency time in short-range networks, the accuracy on the localization methods was studied. The evaluation was make in an experimental setup where two different trajectories and 4 runs of each trajectory were performed. As typical situation in short-range networks, the localization setup was performed in an indoor scenario. The technology was again Bluetooth and the measurements used for estimating position were signal strength measured during the inquiry procedure. The initial test aimed at comparing the performance of the EKF and EKS algorithms and pinpoint some of their problems. In general it was observed, as expected that the EKS outperforms the EKF, due to the fact that the EKS uses future measurements to correct present estimators. Regarding these algorithms, it was observed that they have strong dependency with several factors: the choice of the initial state for the user dynamics; the unknown parameters of the propagation path loss; and abrupt jumps in the user dynamics. The initial state problem was solved by using a annealing technique of optimization. To solve the problem concerning the unknown parameters, they were included in a framework for jointly estimating the user dynamics and those parameters. The solution, also based in a EKF turned out to be instable for some of the experimented trajectory runs. For estimating abrupt changes in the user dynamics, a multiple model framework, SEKF, was used and compared with the EKF. While in the EKF those jumps are estimated by increasing the noise process of the user dynamics, in the SEKF they are medelled by different filters running in parallel. The consequence is that, on one hand, for the EKF the overall trajectory estimation is noisier and, on the other hand, for the SEKF, the processing time enlarges due to the number of parallel filters. Generally, it was observed that the resultant performance of the SEKF was similar to the one presented by the EKF. The last evaluation attempted to study the influence of the noise statistics assumed in the signal strength measurements. Thus, a theoretical model for the noise in the propagation was defined and used in a PF. The EKF which assumes by definition gaussian noise, generally resulted in worse performance than the PF.

Besides the two solutions, the EKF with enlarged noise in the user dynamics and the SEKF, another solution was tested for estimating jumping behavior in the user dynamics: a EKF coupled with e maneuver detector. Based on this solutions a segment-wise linear model, the Drifting Point (DP) model, for the user dynamics was proposed. The DP model, in contrast to the standard CV, estimates trajectories as segments. Based on the idea that noise is uniquely introduced in the maneuvering points, having clean straight trajectories between those maneuvering point, all the measurements are used to estimate those maneuvering positions. Other non-maneuvering positions are simply obtained by interpolation of the estimated maneuvering points. The DP model, tested on simulation tools has outperformed the standard CV model. With the proposed model the accuracy of the localization solution was enhanced by the order to 10% to 50% depending on the setup.

As a potential usage of the typical below-ten-meter accuracy obtained by short-range communications, it was proposed an ad-hoc aided cellular network. The user in a cellular network, equipped with a second ad-hoc short-range technology is enforced to cooperate with other users in order to enhance the accuracy of all users. While the long-range links allow localization of the users based on the knowledge of the position of the BSs, the short-range links constrain those positioning estimator to a lower range of possibilities. This dissertation proposes two solutions for integrating the short-range and the long-range measurements. The first method is a design of an EKF which integrates all the information obtained from all the users. The second solution, named 2LKF relies on the decoupling of absolute and relative localization in order to permit the treatment of the short-range domain and the long-range domain separately. Thus, the relative localization method estimates the relative position of each user with respect to his peers, while the absolute localization estimates the absolute position of the users as a group in the cellular network. This 2LKF solution besides reducing the size of the involved matrices, modularizes the sources of measurements for logical treatment of information. Both EKF-based and the 2LKF were tested with several combinations of measurement. It was observed that the EKF-based solution enhances performance in comparison to non-cooperative schemes, for either ToA or TDOA measurements as source of information for estimating the link length in cellular networks. The 2LKF has resulted in a more scalable solution which is more adequate than the EKF-based solution when the number of cooperative WDs is high. The major drawback of the 2LKF solution is that it does not bring benefit on accuracy for ToA measurements.

Summarizing, it can be concluded that location information has two important aspects, the latency, i.e. the required time since location in requested until it is actually available, and accuracy, i.e. a measure of the error between the estimated position and the true one. Regarding the latency, it was observed that it has a strong dependency not only with the mechanism to measure the wireless channel (such as power control), but it also has additional dependencies with other mechanisms such as collision avoidance. Regarding the accuracy of positioning solutions in wireless networks, it was observed that it is complex to model the propagation effects in such a way that they can be used within statistical frameworks able of estimating position based on noisy measurements. The problem, mainly related with the variability of those effects, makes positioning solutions in wireless networks difficult to implement. Additionally, due to the user mobility, which is characterized by abrupt changes in its dynamics (which are themselves also unknown), the problem gets an incremental complexity. The result is that complex models (such as the multiple model frameworks) do not necessarily result in better performance, meaning that simpler models with more restrictive assumptions may perform equally good, but are simpler to implement, maintain and more stable. Finally, it was verified that more accurate solutions of localization may consider integration of multiple technologies and cooperation schemes between the users. By the price of some complexity in terms of managements, privacy and security, the accuracy of localization can be considerable enhanced.

8.2 Outlook

Looking at the big picture of all the presented work, we have learned that localization solutions in wireless networks are of high complexity when they aim at accurately localize the users in the networks. The major problems are the statistical properties of the noise in the channel reading and the high mobility of the users in the network. In this dissertation, we have proposed localization solutions for enhancing accuracy in wireless networks. Accuracy enhancements were proposed, on one hand, by considering the mobility models of the users, and, on the other hand, by considering cooperative schemes were multiple technologies coexist.

Although localization solution in wireless networks aims at localizing users without additional hardware, future solutions shall surely consider redesigning of the physical layer mechanisms if the main goal is to accurately localize the users. Additionally, with the advent of new technologies coexisting in current wireless devices, novel localization solutions will surely exploit this possibility and surely result in enhanced mechanisms of localization. For these reasons, accurate localization in wireless networks shall be exploited in the direction of optimizing physical layer protocols and exploit the coexistence of multiple technologies.

Part V

Appendixes

Appendix A

Propagation Effects in the Wireless Channel

In the scenarios where WDs are commonly used, propagation conditions are subject to severe problems and destructive effects. Even for the simplest scenario that one can imagine, propagation conditions strongly influence the communication channel. Obviously, nowadays the number of techniques to successfully bypass these problems is enormous. These techniques, rely on very robust algorithms that are able to deal with the undesired propagation effects and produce wireless communications with acceptable quality of service.
Some of the most common propagation effects in terrestrial wireless communication are described bellow.

Path loss: This effect is defined as a large scale effect due to the fact that it is only visible when the WD moves its position of more than hundreds of wavelengths. For this reason, this effect is the one that most positioning solutions rely on in order to estimate position. This effect is due to the inherent loss that the signals are subject to when the physical length of the communication link increases.

Additional losses: This effect, most commonly present in indoor scenarios, is due to obstruction in the propagation channel. Walls, floors and ceiling are commonly the responsible entities for such additional losses. This effect, commonly difficult to estimate is a major problem in indoor localization. Fingerprinting techniques [HB01a] are the most suited to build positioning systems not-so-dependent on this effect.

Shadowing: When connection is obstructed thought the shortest path, but clear along an indirect path, communication is still possible. However, the signal will be strongly influenced by the sharp corners of the objects that block the direct path. These sharp corners disperse the signal and subsequently worsen the quality of the signal at the receiver.

Multipath: This effect is basically the result of the sum of all the signals at the receiver from different trajectories. This effect result in a noisy reception of the signal. Multiple and superposition of several copies of the same signal difficult the intelligibility of the signal at the receiver. This problem can be solved by using stochastic filtering techniques.

Scattering: This effect concerns the deviations that the signal suffers along its travel. Humidity in the air can be one consequence of such scattering of the signal.

Diffraction: When signals travel along two different mediums which have a surface of contact, the signal will suffer a deviation in that surface.

Body Shadowing: This is an effect very common in handsets. Due to the difference of the propagation properties of the human body and the air, the signal is subject to severe dispersion and losses.

For other types of wireless communication systems, such as the high-range satellite communications or micro-range body communication networks, several other propagation effects exist. In satellite systems, the atmosphere itself introduces additional problems in the communication. For body networks, the propagation along the body introduces additional problems to be solved.

Appendix B

Bluetooth survey for location purposes

Bluetooth is a technology that is, nowadays, universally spread over any kind of devices. It represents a good candidate to support research in positioning solutions for short-range wireless technologies which were not originally designed for localization. This challenge force us to obtain a positioning solution supported on existent technology features.
In this chapter, we make an overview of the main characteristics of Bluetooth in Section B and later in Section B and Section B we describe the features existent in Bluetooth that allow us to design a position solution for such technology.

Bluetooth General Concepts

The Bluetooth technology started being developed in 1994 by Ericsson Mobile Communications, and later in 1998 expanded to a wider group, called Special Interest Group (SIG). The first specifications V1.0 were defined in 1999, and later reviewed for publications of new Bluetooth [Blu].

Bluetooth operates in the Industrial, Scientific and Medical (ISM) frequency band, between $f = 2.4GHz$ and $f = 2.4835GHz$. In this band, Bluetooth uses a technique of spread spectrum called Frequency Hopping Spread spectrum (FHSS), which makes use of a hop frequency of $1600hops/s$ among 79 different frequencies with $1\mu s \equiv 1MHz$ wide. Because the modulation used in Bluetooth is Gaussian Frequency Shift Keying (GFSK), the $1MHz$ frequency corresponds to a capacity of $1Mb/s$.

Besides the transmission power being dependent on the class of device, its value is considerable small if compared with other technologies operating in the same frequency band. For devices of class 1, the transmission power is defined as $100mW$ ($20dBm$), while for devices of class 2 that value is about $2.5mW$ ($4dBm$). These transmissions power values limit the communication range to about $100m$ for class 1 and $10m$ for class 2 devices.

Bluetooth nodes transmit data among them in a serial fashion, using asynchronous links, Asynchronous Connection Oriented (ACL), or synchronous links, Synchronous Connection Oriented (SCO).
The ACL link can support symmetrical or asymmetrical communication. For symmetrical communication, the maximum bit rate is $433.9Kbps$ in both directions, while for asymmetrical communication the maximum bit rate is $732.3Kbps$ in one direction and $57.6Kbps$ in the opposite direction. On the other hand, the SCO link only supports symmetrical connections, using for that 3 channels of 64Kbps each.

The Bluetooth Networks have a formation defined as scatternets, which structure is represented in Fig. B.1. As we can see, the network is formed by several cells, called piconets. In each piconet, Bluetooth is allowed to have up to 8 active nodes at the same time. Among these nodes, one is assigned to be the master during the inquiry procedure and is the one that rules all the connection inside the piconet. The remaining nodes play the role of slaves, being allowed to communicate only when requested by the master. In order to establish the connectivity among piconets there are some nodes called bridges that belong to more than one piconet and have the function of transmitting the information from one piconet to the other.

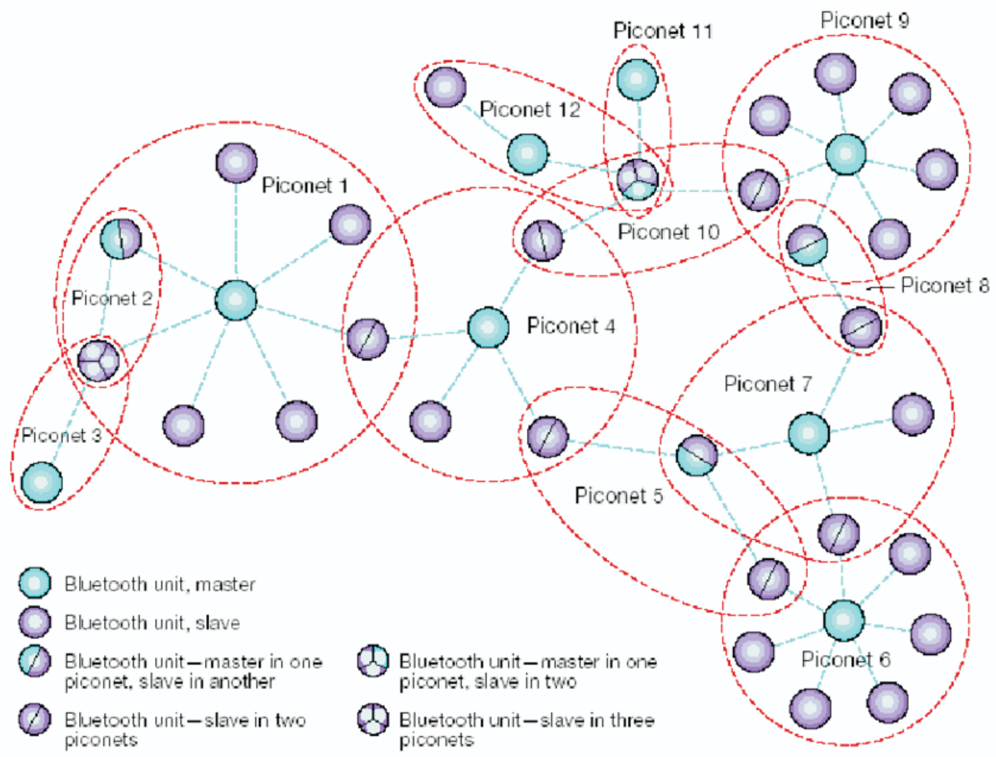

Piconet 11
Piconet 9
Piconet 12
Piconet 1
Piconet 10
Piconet 8
Piconet 2
Piconet 4
Piconet 7
Piconet 3
Piconet 5
Piconet 6
Bluetooth unit, master
Bluetooth unit, slave
Bluetooth unit—master in one piconet, slave in another
Bluetooth unit—master in one piconet, slave in two
Bluetooth unit—slave in two piconets
Bluetooth unit—slave in three piconets

Figure B.1: Bluetooth Scatternet

The Inquiry procedure

The *Inquiry Procedure* (Fig. B.2) is the mechanism existent in Bluetooth that allows nodes to establish a connection. This procedure is the first step, which is then followed by the *Page Procedure* in order to finalize the connection establishment.

In order to discover other devices in its range, a device, called master, shall enter periodically in the *Inquiry Substate* (Section B). In this substate, the device transmits inquiry messages in form of ID packets at different hop frequencies (Section B), as we can see in Fig. B.2. One of these packets shall be received by another device, called slave, which one must be running the *Inquiry Scan Substate* (Section B). In this substate, the slave node must be scanning in the same frequency in order to receive the ID packet sent by the master node (frequency f_i in Fig. B.2). When the packet is received, the slave may provide an answer to the master, what is done entering in the *Inquiry Response Substate* (Section B). However this substate may not be entered if the slave decides not to answer to the master. According to the Bluetooth specifications, the slave has the right to choose if answer or not.

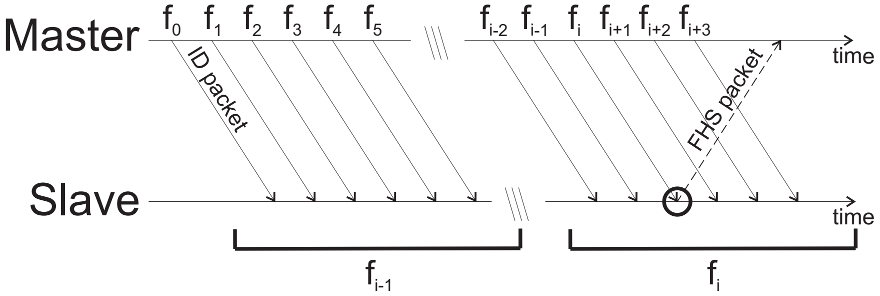

Master
f_0 f_1 f_2 f_3 f_4 f_5
f_{i-2} f_{i-1} f_i f_{i+1} f_{i+2} f_{i+3}
time
ID packet
FHS packet
Slave
time
f_{i-1}
f_i

Figure B.2: Inquiry Procedure in a macro point of view

Inquiry Substate

The *Inquiry Substate* is entered by the master when it wants to collect information about neighboring devices. This substate has a typical duration of $10.24s$ as advised by the Bluetooth specifications. During this period, the master sends continuously ID packets (Fig. B.3) at different hop frequencies dictated by a pseudo-random generator (Section B). The frequencies are divided in 2 sets, called inquiry trains, each of which has 16 frequencies. As we can see in Fig. B.3, every $1.28s$ there is a change of 1 frequency in the set of used frequencies, and every $2.56s$ there is a interchange of frequency sets used. Section B addresses this aspect in detail.

The master may either search for any device, inquiring for the General Inquiry Access Code (GIAC) or search for a certain type of devices, inquiring for the Dedicated Inquiry Access Code (DIAC). Despite the type of devices that the master is searching for, it always follows the same hop frequency sequence, which is determined based on the GIAC and the phase of the sequence determined by

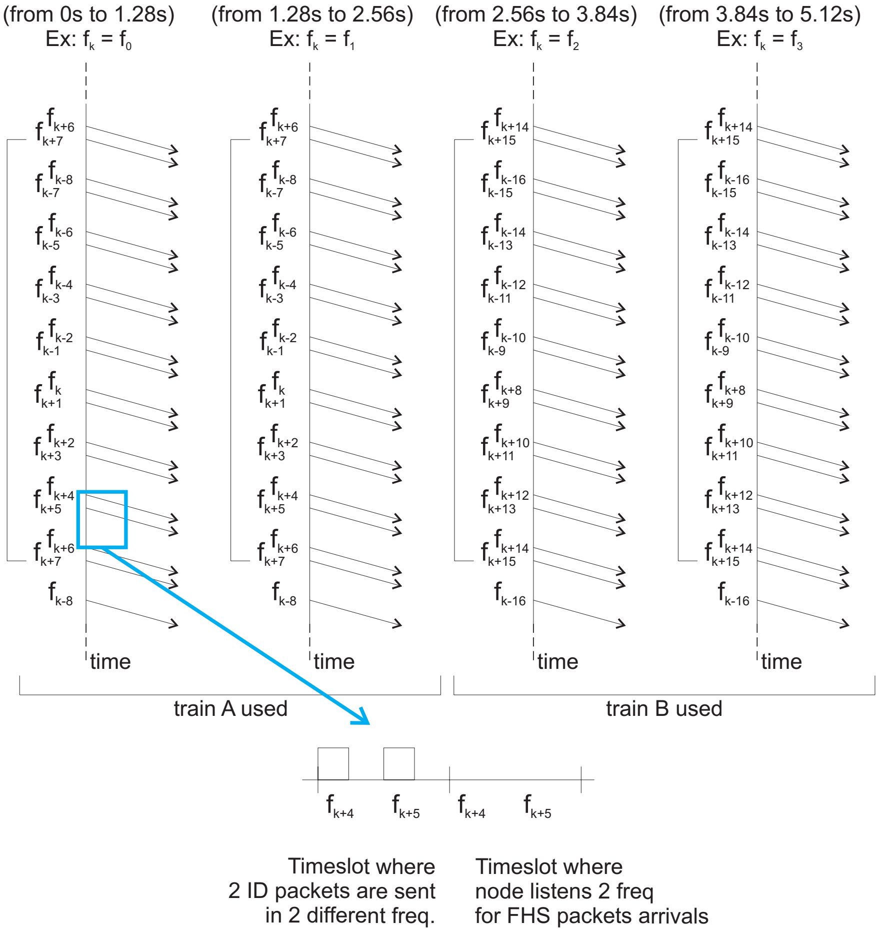


Figure B.3: The Inquiry Substate in detail

the Lower Part Address (LAP) of the Bluetooth address.

During the *Inquiry Substate*, the time is divided in between sending inquiry messages and listening answer messages. In even timeslots the master sends ID packets while in odd timeslots the master listens for FHS packets (Section B address characteristics of the packets). For each timeslot either 2 ID packets are sent, one in each half of the timeslot, or 2 frequencies are listened for FHS packages, one in each half of the timeslot (Fig. B.3). In two consecutive slots the device listens to the frequencies as they are dictated by the frequency generator.

Inquiry Scan Substate

Inquiry Scan Substate is entered by a device when it wants to check if there are any other devices in its range searching for it. This substate can be entered either from the standby or connection state. Entering from the standby state means that there was no packet traffic in the device, while entering from the connection state implies that there was some packet traffic already existent. The existence or not of additional packet traffic in the device influences some parameters in the Inquiry

Scan Substate.

When no packet traffic exists in the network, as advised in the Bluetooth specifications, the Inquiry Scan Substate has a duration of $11.25s$, called Scan Window, and an interval between scans of $1.28s$, called Scan Interval. The Scan Window equals the same duration of 18 timeslots, what was done with the purpose of covering lacks of synchronization between master and slave. During the Scan Window, the device searches for ID packets continuously in the same frequency, changing the scan frequency only after one Scan Interval. The process of determining the scanning frequency is explained in Section B. During the remaining period of the Inquiry Interval, the device returns to the standby or connection state, as it was before start scanning.

The device can be scanning for the GIAC and/or one or more DIACs. Thus, when the device is triggered with an ID packet, the device read the packet and decides, based on the inquiry access code, if it wants to go further with the connection or just discharge the packet. In case the ID packet includes the inquiry access code searched, the device may enter the Inquiry Response Substate (Section B).

Inquiry Response Substate

This substate is entered by the slave as a way of providing an answer to the master, when the slave is triggered with an ID packet.

After receiving the ID packet, still in the Inquiry Scan Substate, the device right after $625\mu s$, and already in Inquiry Response Substate, shall send an FHS packet in the dictated by the frequency generator. When sent the FHS packet, the device adds one unit in phase with step of $1.28s$ to a counter included in the frequency generator.
When the answer is provided, it can happen that the device is synchronized with other devices in such a way that there is a time period where they search the same frequency. In this case, the FHS packets would collide and the master wouldn't receive any answer. To avoid this problem, right after sending the FHS packet, the slave shall choose a random number between 0 and 1023 timeslots ($639.375ms$) uniformly distributed, and leave the Inquiry Procedure during that period. Because devices will wake up after different time periods, the synchronization, most probably, is lost. When the device wakes up, it returns to the Inquiry Scan Substate.

Packets used in Inquiry Procedure

As seen already in previous sections, there are two different packets used in the Inquiry Procedure:

- **ID packet** This packet has a fixed length of $68bits$. Since for Bluetooth specifications V1.2, the data rate is $1Mb/s$, the size of the packet represents, roughly, only 11% of the timeslot used to send the ID packet. This packet includes, for the case of the Inquiry Procedure, only the inquiry access code.

- **FHS packet** This packet is sent in the slave-master direction and it aims at synchronizing both devices. It encapsulates the Bluetooth Address and the clock of the sender. This packet has a size of $366bits$, meaning for the Bluetooth specifications a fraction of around 59% of the duration of one timeslot.

The Frequency Generator

The frequency generator existent in Bluetooth is relatively complex having several operations and several input parameters. In this section we explain conceptually how this generator works and which outputs it does, disregarding detailed explanation on the implementation.

The frequency generator is a unique piece of hardware that is responsible to generate every frequency hopping for every possible case in Bluetooth protocols. Inquiry Procedure is one out of all possible cases that use the frequency generator. In the inquiry Procedure, two input parameters are needed. The first is the GIAC, which defines the sequence used and the native Bluetooth clock, that determines the phase of hopping. However, depending on the substate of the Inquiry Procedure, the frequency hopping will behave slightly different.

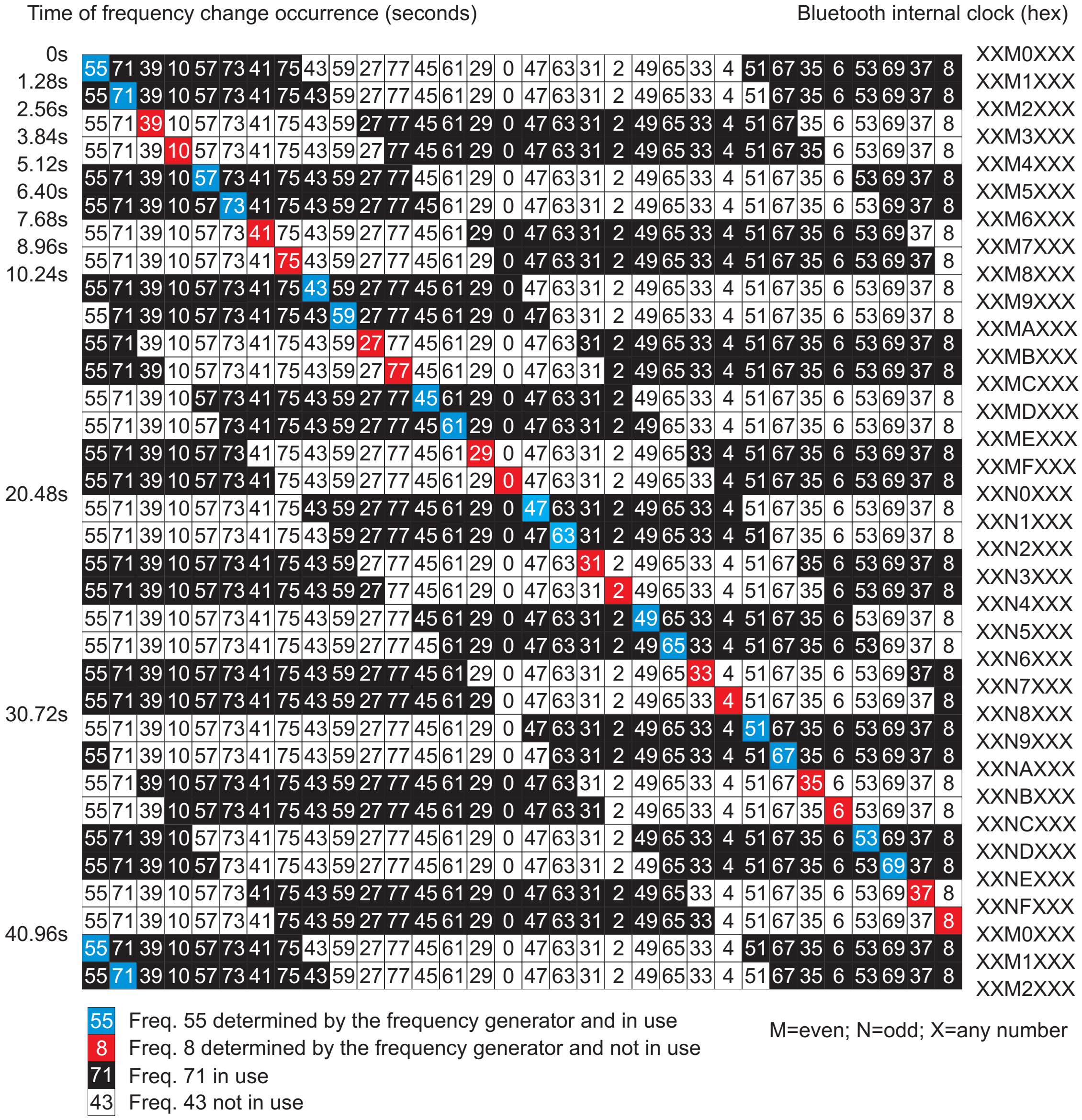
In general, for Inquiry Procedure, only 32 out of the 79 frequencies are used. From those 32 frequencies, one different frequency, f_k, is selected each $1.28s$ following the sequence in Table B.1. For example if at time t_0 is selected frequency $f_k = 57$, at time $t_0 + 1.28$ is selected frequency $f_k = 73$. For Inquiry Scan Substate, the selected frequency represents the frequency that devices listen to. For the Inquiry Substate, the frequency used to send ID packets is more complex.

Table B.1: Set of frequencies used in the Inquiry procedure: sorted in ascendant manner and in the way they appear in the sequence

Sequence	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0
	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8
Sorted	0	2	4	6	8	10	27	29	31	33	35	37	39	41	43	45
	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	77

In the Inquiry Substate, the selected frequency, is not the only used frequency. Although it happens a frequency selection every $1.28s$, the substate makes use of a set of 16 frequencies, called inquiry train. There are 2 inquiry trains: the inquiry train A that is defined as the set of frequencies from f_{k-8} to f_{k+7}, i.e., includes f_k and inquiry train B that is the complementary of inquiry train A, i.e., includes frequencies from f_{k-16} to f_{k-9} and from f_{k+8} to f_{k+15}. Looking at Fig. B.4 we can see the trains and that the frequency selection change at each $1.28s$ has a consequence of interchanging one frequency between both trains. Besides that, Bluetooth specifications say that every $2.56s$ a change of used train must occur. This fact is observable in Fig. B.4 at time instants multiples of $2.56s$.

As we can see from Fig. B.4 (in the right) the hopping phase is determined by the native clock,



Time of frequency change occurrence (seconds)																																	Bluetooth internal clock (hex)
0s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM0XXX
1.28s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM1XXX
2.56s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM2XXX
3.84s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM3XXX
5.12s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM4XXX
6.40s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM5XXX
7.68s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM6XXX
8.96s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM7XXX
10.24s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM8XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM9XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMAXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMBXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMCXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMDXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMEXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXMFXXX
20.48s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN0XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN1XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN2XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN3XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN4XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN5XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN6XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN7XXX
30.72s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN8XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXN9XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNAXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNBXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNCXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNDXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNEXXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXNFXXX
40.96s	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM0XXX
	55	71	39	10	57	73	41	75	43	59	27	77	45	61	29	0	47	63	31	2	49	65	33	4	51	67	35	6	53	69	37	8	XXM1XXX
																																	XXM2XXX

55 Freq. 55 determined by the frequency generator and in use
8 Freq. 8 determined by the frequency generator and not in use
71 Freq. 71 in use
43 Freq. 43 not in use

M=even; N=odd; X=any number

Figure B.4: Frequency shifting that occurs in the master node

being the clock bits from $12 - 16$ responsible to determine the actual selected frequency f_k.

Appendix C

Analytical Approximation of the Inquiry Procedure in Bluetooth

This section gives an analytical approximation for the Inquiry Time for scenarios with $N = \{1, 3\}$ APs. The analytical formulation approaches the Inquiry Time by considering only the most influencing factors. Thus in this section it assumed that:

- Since the time duration of the scanning windows is more than 100 orders of magnitude lower than the time duration of the scanning interval, the duration of the scanning window is assumed zero. This means that the WD provides an answer to the AP as long as the WD is scanning in a frequency present in inquiry train that the AP is using.

- No collision of packages happen.

The first formulation is obtained for the case of a single AP inquiring the WD. When the AP starts inquiring, the WD may be or not in the scan window, meaning that the AP may have to wait a period of time T_s that is uniformly distributed between $0s$ and $1.28s$ if is given that the scanning frequency is the train A.

$$\mathrm{pr}(T_s) = Unif(0, 1.28)[s] \tag{C.1}$$

Once the scan window starts or if the inquiry started during the scan window, an additional time must be considered. This time, defined as T_{sw} is the time spent since the scan window started until the frequency scanned by the slave is used by the master. However, by assumption this time is neglected. The previous description was made given that the scanning frequency belong to the train A. However, it can easily be extended to the general case because the probability of the frequency being either in train A or B is the same and equals 1/2. Thus, this distribution is a discrete function as described in eq.(C.2).

$$\mathrm{pr}(T_{tr}) = \frac{1}{2}\delta(t) + \frac{1}{2}\delta(t - 2.56)[s] \tag{C.2}$$

where the outcomes of T_{tr} are either $0s$ or $2.56s$ with equal probability.
With the two variables T_s and T_{tr} and the respective distribution in eq.(C.1) and eq.(C.2) we can define the inquiry time T_t as the sum of the two variables (eq.(C.3))

$$T_t = T_s + T_{tr} \tag{C.3}$$

The distribution of a variable like T_t in eq.(C.3) that is the sum of random variables equals the convolution of their distributions.

$$\mathrm{pr}(T_t) = \mathrm{pr}(T_s) * \mathrm{pr}(T_{tr}) \tag{C.4}$$

$$\mathrm{pr}(T_t) = \frac{1}{2}\Bigg[Unif(0, 1.28) + Unif(2.56, 3.84)\Bigg] \tag{C.5}$$

For comparing the analytical result of eq.(C.5) with the empirical distribution previously presented, the PDF and the CDF of both distributions are plotted respectively in Fig. C.1 and Fig. C.2.

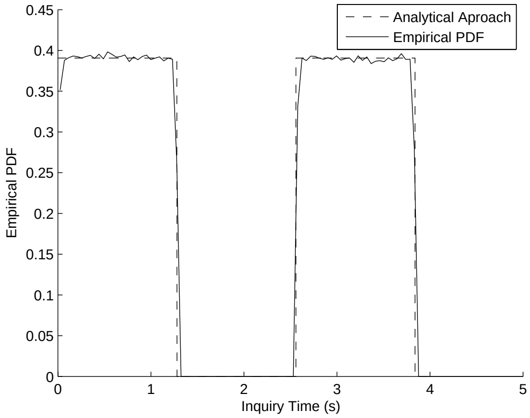


Figure C.1: PDF of the inquiry time for a scenario with 1 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

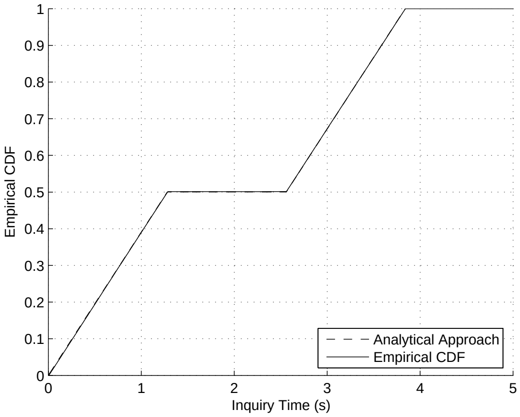


Figure C.2: CDF of the inquiry time for a scenario with 1 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

Similarly to the 2 AP scenario, it is possible, for the 3 AP scenario, to define a random variable describing the distribution of the inquiry trains which the scanning frequency belongs to. For 3 APs this random variable is still possible to define because of the properties of the worse case, i.e the case when all the APs have the scanning frequency in the same train. For this case, the longest inquiry time, corresponding the sum of the inquiry interval, and 2 maximum back-off times, is still lower than the time span before exchange of inquiry trains. Thus, the possible outcomes are:

- C_{3A} - The frequency used by the WD for scanning is for the 3 APs in their train A.
- C_{2A1B} - The scanning frequency is in train A for 2 AP and in train B for the other AP.

- C_{1A2B} - The scanning frequency is in train A for one AP and in train B for the other 2 APs.
- C_{3B} - The scanning frequency is in the train B of all the 3 APs.

The probability of each outcome is $\Pr(C_{3A}) = \Pr(C_{3B}) = 1/8$ and $\Pr(C_{2A1B}) = \Pr(C_{1A2B}) = 3/8$ and subsequently the distribution of the inquiry time can be defined by:

$$\mathrm{pr}(T) = \Pr(C_{3A})\mathrm{pr}(T|C_{3A})+\Pr(C_{2A1B})\mathrm{pr}(T|C_{2A1B})+\Pr(C_{1A2B})\mathrm{pr}(T|C_{1A2B})+\Pr(C_{3B})\mathrm{pr}(T|C_{3B}) \tag{C.6}$$

$$\mathrm{pr}(T) = \frac{1}{8}\mathrm{pr}(T|C_{3A}) + \frac{3}{8}\mathrm{pr}(T|C_{2A1B}) + \frac{3}{8}\mathrm{pr}(T|C_{1A2B}) + \frac{1}{8}\mathrm{pr}(T|C_{3B}) \tag{C.7}$$

Based on the sequence of events that happen for each different outcome of the random variable C, the probability densities of eq.(4.3) can be defined as:

$$\begin{cases} \mathrm{pr}(T|C_{3A}) = Unif(0, 1.28) * Unif(0, 0.639) * Unif(0, 0.639) \\ \mathrm{pr}(T|C_{2A1B}) = Unif(2.56, 3.84) \\ \mathrm{pr}(T|C_{1A2B}) = Unif(2.56, 3.84) * Unif(0, 0.639) \\ \mathrm{pr}(T|C_{3B}) = Unif(2.56, 3.84) * Unif(0, 0.639) * Unif(0, 0.639) \end{cases} \tag{C.8}$$

For the density $\mathrm{pr}(T|C_{1A2B})$ the reason that makes the AP with the scanning frequency in train A to not influence the other 2, is the same previously explained in the 2 AP scenario (Section 4.3.1). By applying eq.(4.4) in eq.(4.3), the final distribution is given by:

$$\begin{aligned} \mathrm{pr}(T) &= \tfrac{1}{8}\left\{\Big[Unif(0, 1.28) + Unif(2.56, 3.84)\Big] * Unif(0, 0.639) + 3Unif(2.56, 3.84)\right\} \\ &* \; Unif(0, 0.639) + \tfrac{3}{8}Unif(2.56, 3.84) \end{aligned} \tag{C.9}$$

where its PDF and CDF are plotted respectively in Fig. C.3 and Fig. C.4. As Fig. C.3 shows, the additional peak at time $5s$ to $6s$ has a higher density than for the 2 AP scenario. This means that the effect of packet collisions is higher for this case, what is actually expectable. Additionally, it is possible to see in Fig. C.4 that the analytical approximation is generally less accurate than in the 2 AP scenario.

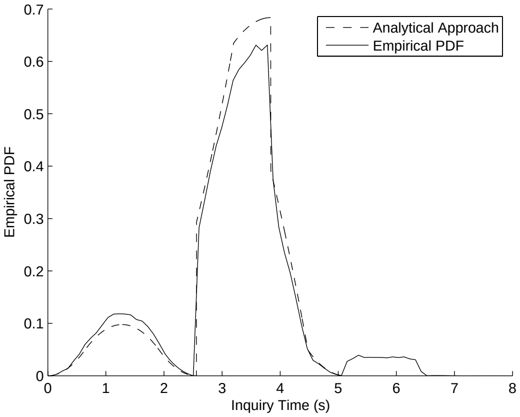


Figure C.3: PDF of the inquiry time for a scenario with 3 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

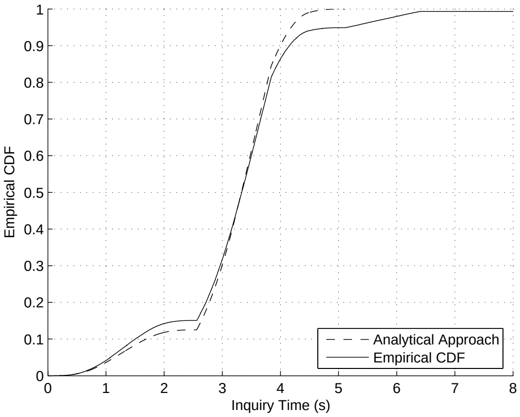


Figure C.4: CDF of the inquiry time for a scenario with 3 AP and 1 WD. Comparison between an analytical approach and an empirical simulation of the inquiry specifications.

Appendix D

Additional evaluation of the EKF and EKS

This section introduces additional results on the analysis performed in Section 5.5.

The second parameter, influencing the propagation path loss, interesting to study is the path loss exponent β. Since the experiment was done in an indoor scenario, the path loss exponent is not expected to be 2 as it is in the free space path loss. This fact can be observed in Fig. D.1 and Fig. D.2, where β gives maximum performance for values around 2.5.

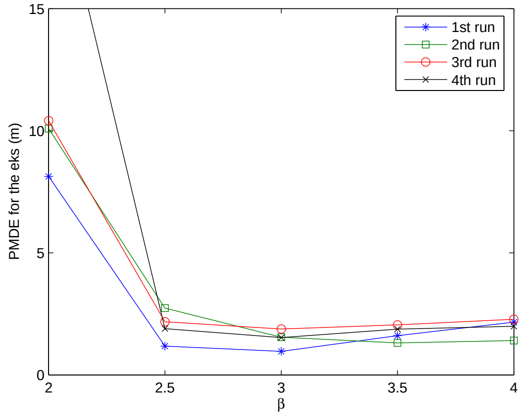


Figure D.1: PMDE performance of the EKS with respect to the β parameter for the straight trajectory.

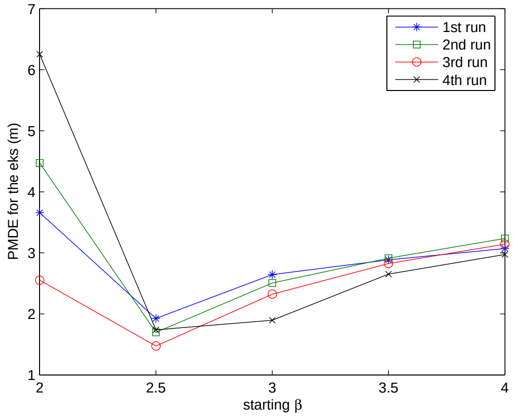


Figure D.2: PMDE performance of the EKS with respect to the β parameter for the squared trajectory.

As we can see in the background Section 3.1.1, an additional parameters influencing the performance of the EKS is its lag interval of smoothing. Different lag time values were used and the PMDE of the position estimation was plotted in Fig. D.3 and Fig. D.4. As we can see, the higher the lag time is, the better is the system performance, but at the same time, the lower is the gradual increment in performance. For a lag value higher than around 100 or 200 observations the gain in performance is

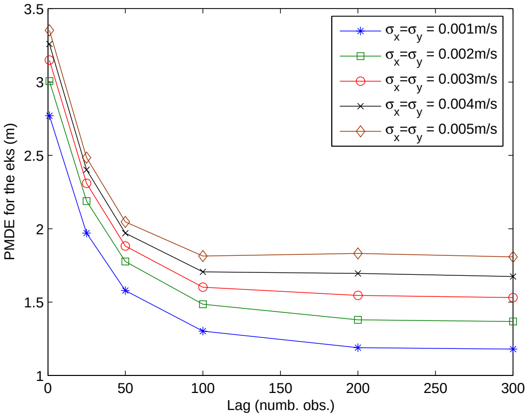


Figure D.3: The influence of the lag in the PMDE performance of the EKS. Results for the 1st run of the straight trajectory.

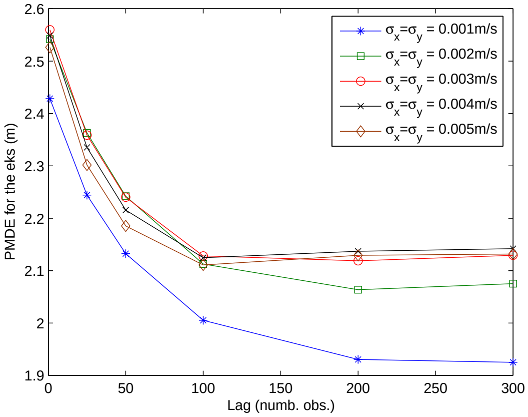


Figure D.4: The influence of the lag in the PMDE performance of the EKS. Results for the 1st run of the squared trajectory.

ignorable for the presented design parameters. Moreover, the higher the lag is, heavier is the computational effort for estimating the trajectory, which is due to the quadratic complexity of the EKS.

Another aspect important to study is the capacity of the system to track WDs moving at different velocities. The fact is that for different velocities, the effective amount of measurements is different, given the same trajectory length and the same measurement frequency. Thus, for higher velocities the amount of information is lower than for the case of lower velocities. In order to study this aspect, instead of running the same experiment over and over again, the measurements were processed such that different speeds are emulated. As a starting point it was defined the speed as the overall covered distance divided by the entire period of time. Based on this definition of speed, the set of measurements from each AP were reduced in order to include only every *i-th* obtained measurement and the time reference of the experiment was compressed by a factor of i. This mechanism implies: (i) the speed to be increased in a factor of i, given that the trajectory length and the measurement frequency remain the same; and still (ii) give an acceptable approximation for the Bluetooth link layer protocol of communication (see Chapter 4). The use of this mechanism implies that time references change and subsequently the acceleration process noise $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{y}}$ has to be redesigned. For this reason, we have decided to increase the velocity in a factor of 10 and 30 and manually tune the parameters $\sigma_{\ddot{x}}$ and $\sigma_{\ddot{x}}$ in order to get the best performance. Thus in Fig D.5 and Fig D.6 we show the tracking result for a compression factor of 10, what implies a velocity of $0.23m/s$ for the straight trajectory and $0.44m/s$ for the squared trajectory. Note that, as it was shown in Table 5.1, the two trajectories were walked at different velocities, meaning that when the velocities are increased by the same factor i, their value will be also different in each case. The process noise used was $\sigma_{\ddot{x}} = \sigma_{\ddot{y}} = 2.5m/s^2$. Equivalently, in Fig D.7 and Fig D.8 we can see the best obtained performed for a compression factor of 30 and $\sigma_x = 50m/s^2$. The real velocity in this setup is $0.70m/s$ for the straight trajectory and $1.34m/s$ for the squared trajectory.

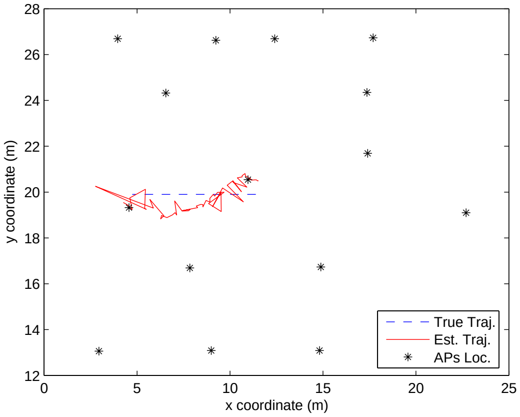


Figure D.5: Performance of the EKS when the velocity is emulated by a factor of 10 in the straight trajectory. The velocity is $0.23m/s$ and $\sigma_{\ddot{x}} = 2.5m/s^2$.

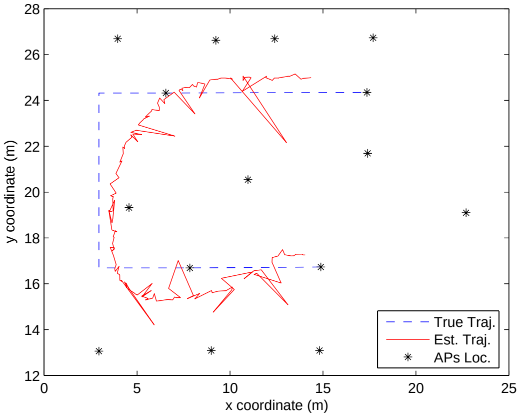


Figure D.6: Performance of the EKS when the velocity is emulated by a factor of 10 in the squared trajectory. The velocity is $0.44m/s$ and $\sigma_{\ddot{x}} = 2.5m/s^2$.

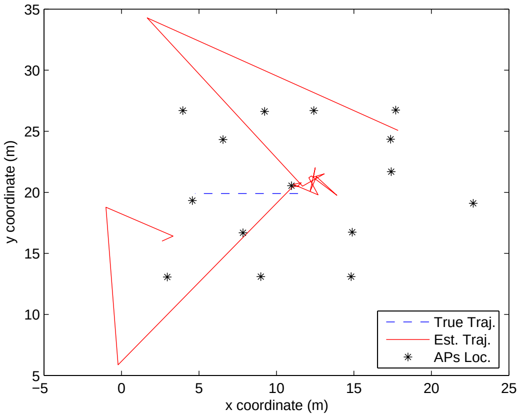


Figure D.7: Performance of the EKS when the velocity is emulated by a factor of 30 in the straight trajectory. The velocity is $0.70m/s$ and $\sigma_{\ddot{x}} = 50m/s^2$.

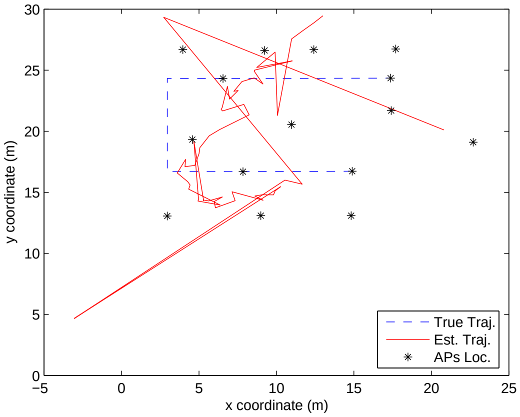


Figure D.8: Performance of the EKS when the velocity is emulated by a factor of 10 in the squared trajectory. The velocity is $1.35m/s$ and $\sigma_{\ddot{x}} = 50m/s^2$.

One important result to mention is that, when speed increases, there is a need of increasing the position covariance noise, what has a consequence of producing noisier estimators of position. It is important to underline at this point the fact that the EKF and EKS were optimized for the given trajectory and still the results are fairly poor. Furthermore, when this optimization would not take place, the obtained results would have been unacceptably inaccurate. This means that the filter is extremely sensitive when the target moves at higher speeds such as a human walk (Fig. D.8). The reason for this fact is twofold: first, the measurements are highly corrupted by noise; and second, the frequency of measurements is considerably low. These two effects when combined, as in the present application, result in impracticability to track targets moving above a determined velocity.

Appendix E

Localization in Wireless Networks Using Least Square Approach

When estimating positioning in an off-line localization procedure, one has available the entire set of data. Subsequently, the entire data can be used in a batch algorithm and estimate the entire trajectory at once. A simple solution for this problem is to use a LS solution to minimize the error in the estimated trajectory. According to any standard linear estimation reference, where here we cite for example [KAB00], the least squares problem can be defined as the problem of determining the value of the desired unknown variables that minimizes the sum of the squared residuals, i.e.

$$J = \sum_{i=1}^{n} \Big(y_i - f(x_i) \Big)^2 \tag{E.1}$$

where the cost function J is evaluated based on the observations y_i and the transformation f of the desired states x_i. In the case that the function f is linear, the algorithm can be simplified.

A Linear Least Square Solution for Straight Movement Estimation

In the above Section 5.4, the tracking problem was studied in terms of system observability. Inspired in the reference [SN03], the problem was derived in order to have a linear relation between the measurements and the state variables $[x, y, \dot{x}, \dot{y}]$. The resultant expression, in eq.(5.22), with terms defined by eq.(5.23) and eq.(5.24), is in fact a linear equation, where the least square algorithm can be easily applied. According to linear estimation literature [KAB00], the least squares problem can be defined as the problem of determining the value of the desired unknown variables that minimizes the sum of the squared residuals. This definition in a linear relation between the unknown variable and the observation can be defined as:

$$X = (H^T H)^{-1} H^T Z \tag{E.2}$$

where H and Z are respectively given by eq.(5.23) and eq.(5.24).
In order to test the performance of such solution, we have decided to plot the tracking results for a single run. The results plotted in Fig. E.1 and Fig. E.2, represent the result when σ_Z assumes respectively the values of $1dB$ and $6dB$. For each plot, several measurement sampling rates where used. As we can see in Fig. E.1 the higher the measurement sampling rate, closer is the estimated trajectory to the real one. However, it is clear that the performance is poor, what can be observed even for high values of measurement rates, such as $f_s = 100$ measurement a second. When observing Fig. E.2, where $\sigma_Z = 6dB$, which represents a more realistic value for the measurement noise, the results are even worse. We can see that not even for high values of the measurement sampling rates, the estimated trajectory shows acceptable accuracy.

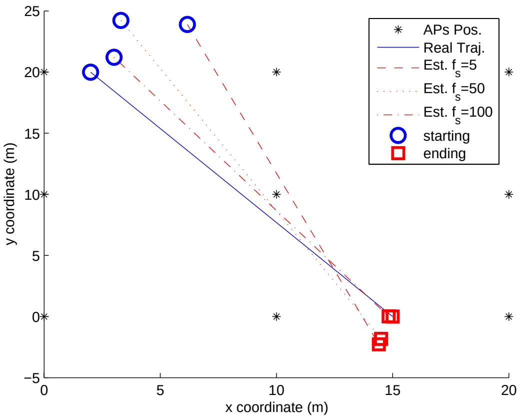


Figure E.1: Single run of a LS solution when $\sigma_Z = 1dB$.

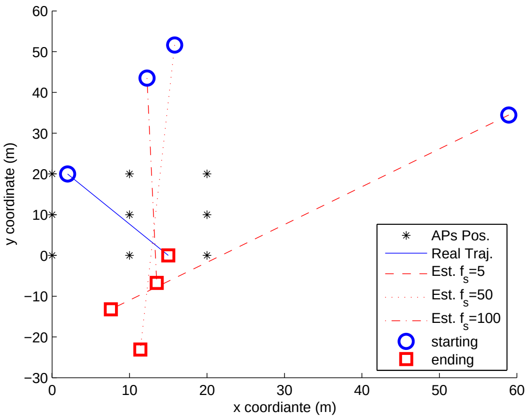


Figure E.2: Single run of a LS solution when $\sigma_Z = 6dB$.

To strengthen the just aforementioned performance of this LS solution we have decided to run the same simulation 1000 times and plot the empirical CDF of the PMDE metric. For each run, the generator of noise was initiated with a different seed. As we can see in any of the Figures E.3 or E.4, the conclusion is confirmed. Additionally we can conclude that in any case the performance gets better and better, the higher the measurement rate is, what follows the common sense. However, the performance is still very unacceptably poor in whatever case considered.
In order to find an explanation for such fact, we shall look into eq.(5.23). As we can see, the Z vector is dependent on the APs position and on the measurements. The problem is that the measurements are in linear scale, meaning that their dependence with the noise in a dB scale is exponential (see eq.(5.13)). For this reason, high Gaussian noise in a dB scale are translated into extremely high noise and non-Gaussian components when translated into distance. This introduces in the summation included in the vector Z a strong domination of one of the parameters, *viz.* m in order to the remaining terms. From the presented results we can conclude that the solution presented is not suitable for the case when the WD is moving.

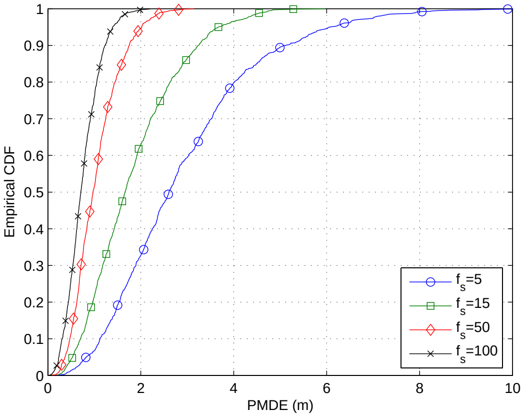


Figure E.3: Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 1dB$.

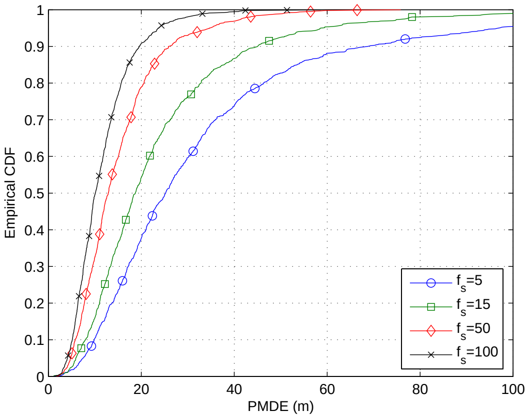


Figure E.4: Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 6dB$.

Using the Ordinary Least Square Algorithm for Straight Movement Estimation

As we have seen in the previous section, an attempt to obtain a Linear Least Squares (LLS) solution has resulted in bad performance. Thus, in this section we will attempt to use the Ordinary Least Square (OLS) method for estimating the target trajectory. Since eq.(E.1) generalizes the problem of the OLS, we will attempt to use that expression and minimize it according to a numerical method of optimization. The chosen method is the simplex search method implemented by the optimization toolbox of MATLAB. The method, developed in [LRWW98] is an unconstrained nonlinear optimization algorithm that minimizes a function of several variables. The variables defined for the present application will be the initial and the final position of the target trajectory. Let us assume that measurements of received power were obtained at discrete times $t = \{t_0, ..., t_k\}$ and that the target was at its initial position X_0 for t_0 and at its final position X_k at t_0. Since velocity is assumed constant along the entire movement, we can interpolate the position of the target at time t_i, $(i = 1, ..., k)$ by simply using the initial position, final position and the corresponding timestamps. Thus,

$$X_i = \text{interp}\Big([t_0 \quad t_k] \quad ; \quad [X_0 \quad X_k] \quad ; \quad t_i\Big) \tag{E.3}$$

Subsequently, if measurement P_i is available at time t_i the OLS can be defined as:

$$\{\hat{X}_0, \hat{X}_k\} = \arg \min_{\{X_0, X_k\}} \sum_{i=1}^{k} \left[P_i - \left(\alpha - 10\beta \log \Big(||X_i, \Lambda_i|| \Big) \right) \right]^2 \tag{E.4}$$

where the function argmin is performed by the optimization routine of MATLAB. However, the simplex method used in the minimization requires one to provide an initial guess for the variables which are being searched. Thus, initial guess for the initial position was estimated using a primary

OLS using the first 3 measurements from different APs, while the final position was estimated using the last 3 measurements of the entire set of measurement.

After implementing the aforementioned OLS algorithm, we have performed some tests of performance on the simulation software. Primarily we have chosen to obtain results for a single run using measurement frequencies of $f_s = 5Hz$, $f_s = 50Hz$ and $f_s = 100Hz$. The results, presented in Fig. E.5 undoubtfuly outperform the LLS solution presented in Fig. E.2. The reason is that for the OLS the measurements were not pre-manipulated as they were in the LLS, the problem of indirect loss of information does not happen. Additionally, looking at the graphic in Fig. E.6, we can see that the performance of the PMDE metric is lower than $2.5m$ for 95% when $f_s = 5Hz$. When the measurement frequency is $f_s = 50Hz$, we can see that for 100% of the cases the PMDE is smaller than a meter. We shall however underline that such results are based on the fact that the simulator assumes perfect gaussian distributions, meaning that for non-gaussian distributions, the results may vary. Moreover, it is not guarantied indeed that the method will converge to the absolute minimum, what is in fact a limitation of the simplex search method. This numerical minimization algorithm guaranties that the minimum is found given that the cost function is convex, what is not the present case.

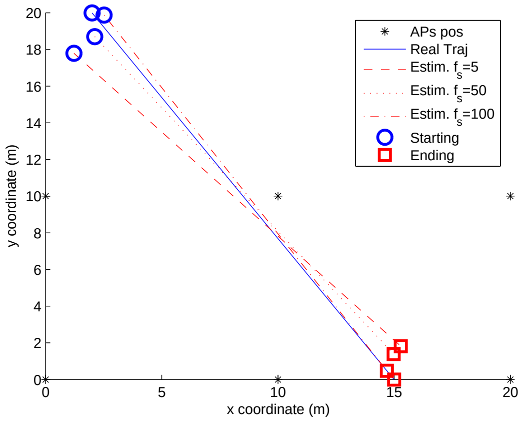


Figure E.5: Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 1dB$.

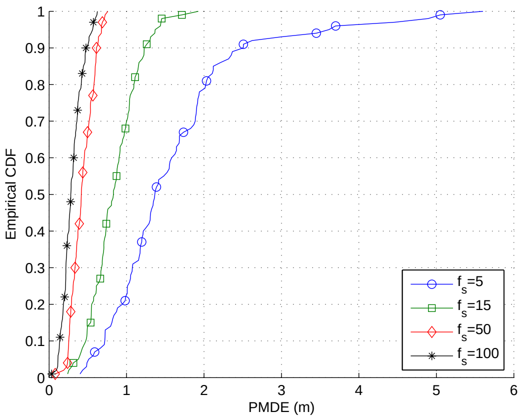


Figure E.6: Empirical CDF of the PMDE metric for the straight trajectory when $\sigma_Z = 6dB$.

The biggest disadvantage of this method is however the computational effort that the numerical minimization requires. It surely results in an unpractical solution for segment-wise trajectories, as it is the assumption in this chapter. The problem is that the higher the number of maneuvers, higher is the number of variables in the minimization and more costly is the algorithm.

Bibliography

[AK07] D.N. Aloi and O.V. Korniyenko. Comparative Performance Analysis of a Kalman Filter and a Modified Double Exponential Filter for GPS-Only Position Estimation of Automotive Platforms in an Urban-Canyon Environment. *IEEE Transactions on Vehicular Technology*, 56(5), 2007.

[AKK04] JN Al-Karaki and AE Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(6):6–28, 2004.

[AMG$^+$02] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S.A. Adelaide. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[BJ76] G.E.P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day Series in Time Series Analysis. Holden-Day, 1976.

[Blu] The Official Bluetooth Technology Info Site Bluetooth.com. http://www.bluetooth.com/.

[Blu04] SIG Bluetooth. *Specification of the Bluetooth System. Core Specification v2. 0+ EDR.* November, 2004.

[BOJ$^+$06] M. Bauer, R.L. Olsen, M. Jacobsen, L. Sanchez, M. Imine, and N. Prasad. Context Management Framework for MAGNET Beyond. *Workshop on Capturing Context and Context Aware Systems and Platforms, Proceedings of IST Mobile and Wireless Summit*, 2006.

[BP00] P. Bahl and VN Padmanabhan. RADAR: an in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2, 2000.

[BS88] Y. Bar-Shalom. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering.* Academic Press, Cambridge, Massachusetts, 1988.

[BSLK01] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* Wiley New York, 2001.

[BZ] Blip Systems Blip Zones. http://www.blipsystems.com/.

[CCKM01] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. *Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 18–34, 2001.

[CH94] YT Chan and KC Ho. A simple and efficient estimator for hyperbolic location. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 42(8):1905–1915, 1994.

[Che99] C.T. Chen. *Linear System Theory and Design*. Oxford University Press, New York Oxford, third edition, 1999.

[CS98] JJ Caffery and GL Stuber. Overview of radiolocation in CDMA cellular systems. *Communications Magazine, IEEE*, 36(4):38–45, 1998.

[CSC+06] M.Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical Metropolitan-Scale Positioning for GSM Phones. *Proc. 7th Int. Conf. Ubiquitous Computing (UbiComp)*, 2006.

[CST94] S.E. Cohn, NS Sivakumaran, and R. Todling. A Fixed-Lag Kalman Smoother for Retrospective Data Assimilation. *Monthly Weather Review*, 122(12):2838–2867, 1994.

[DA00] A.K. Dey and G.D. Abowd. Towards a Better Understanding of Context and Context-Awareness. *CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness*, 2000.

[DF00] P. Deng and PZ Fan. An AOA assisted TOA positioning system. *Communication Technology Proceedings, 2000. WCC-ICCT 2000. International Conference on*, 2, 2000.

[DKNP06] M. Duflot, M. Kwiatkowska, G. Norman, and D. Parker. A formal analysis of bluetooth device discovery. *International Journal on Software Tools for Technology Transfer (STTT)*, 8(6):621–632, 2006.

[Dod] Social Network Dodgeball. http://www.dodgeball.com/.

[EPS] Ekahau Ekahau Positioning System. http://www.ekahau.com/.

[FCC99] FCC. FCC Acts to Promote Competition and Public Safety in Enhanced Wireless 911 Services. *Federal Communications Commission*, september 1999.

[FF07] S. Frattasi and J. Figueiras. *Ad-Coop Positioning System : Using an Embedded Kalman Filter Data Fusion*, pages 120–131. CRC Press, 2007.

[FHL+03] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July-September 2003.

[FS06] J. Figueiras and H.P. Shwefel. Fast and Low-Complexity Simulations of the Inquiry Time in Bluetooth. In *In Proceedings of the 13th International Conference on Telecommunications (ICT 2006)*, may 2006.

[FSK05] J. Figueiras, H.P. Shwefel, and I. Kovacs. Accuracy and timing aspects of location information based on signal-strength measurements in Bluetooth. In *In Proceedings of the 16th International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC 2005)*, september 2005.

[GEYC97] LJ Greenstein, V. Erceg, YS Yeh, and MV Clark. A new path-gain/delay-spread propagation model for digital cellular channels. *Vehicular Technology, IEEE Transactions on*, 46(2):477–485, 1997.

[Gol05] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[GSS93] NJ Gordon, DJ Salmond, and AFM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

[HB01a] J. Hightower and G. Borriello. A survey and taxonomy of location sensing systems for ubiquitous computing. UW CSE 01-08-03, University of Washington, Department of Computer Science and Engineering, Seattle, WA, August 2001.

[HB01b] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.

[HBB02] J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pages 22–28, Callicoon, NY, June 2002. IEEE Computer Society Press.

[HFL+04] A. Haeberlen, E. Flannery, A.M. Ladd, A. Rudys, D.S. Wallach, and L.E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84, 2004.

[Hig04] J. Hightower. *The Location Stack*. PhD thesis, Department of Computer Science & Engineering, University of Washington, Seattle, WA, 2004.

[JU97] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 3, 1997.

[K+96] E.D. Kaplan et al. *Understanding GPS: Principles and Applications*. Artech House, 1996.

[KAB00] T. Kailath, Sayed A.H., and Hassibi B. *Linear Estimation.* Prentice Hall Information and Systems Science Series. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[Küp05] A. Küpper. *Location-based services.* Wiley, 2005.

[Lab] Place Lab. http://www.placelab.org/.

[LAK$^+$01] H. Laitinen, S. Ahonen, S. Kyriazakos, J. Lähteenmäki, R. Menolascino, and S. Parkkila. Cellular Location Technologies, Cellular Network Optimisation Based on Mobile Location. *Cello Consortium, Document ID: CELLO-WP2-VTT-D03-007-Int, November*, 2001.

[LaV03] J.J. LaViola. Double exponential smoothing: an alternative to Kalman filter-based predictive tracking. *Proceedings of the workshop on Virtual environments 2003*, pages 199–206, 2003.

[LBR$^+$02] AM Ladd, KE Bekris, A. Rudys, G. Marceau, LE Kavraki, and DS Wallach. Robotics-based location sensing using wireless ethernet. *Mobicom 2002, Atlanta, Georgia*, 2002.

[LCC$^+$05] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place Lab: Device Positioning Using Radio Beacons in the Wild. *Proceedings of Pervasive*, 2005.

[LFFS06] E. Lhomme, S. Frattasi, J. Figueiras, and H.P. Schwefel. Enhancement of Localization Accuracy in Cellular Networks via Cooperative AdHoc Links. In *In Proceedings of the 3rd IEE Mobility Conference*, october 2006.

[LRWW98] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[LS] Google Local Search. http://local.google.com/.

[mat04] Matlab 7.0, 2004. http://www.mathworks.com/.

[MG01] A.P. Andrews M.S. Grewal. *Kalman Filter: Theory and Practice Using MATLAB.* John Wiley & Sons, Inc., 2 edition, 2001.

[MMF$^+$05] G. Monghal, Y. Malidor, J. Figueiras, H.P. Schwefel, and Kovacs I. "Enhanced triangulation method for positioning of moving devices. In *In Proceedings of the 8th International Symposium on Wireless Personal Multimedia Communications (WPMC 2008)*, september 2005.

[MRW$^+$07] C. Mayorga, F Rosa, S. Wardana, G. Simone, M. Raynal, J. Figueiras, and Frattasi S. Cooperative Positioning Techniques for Mobile Localization in Hybrid WiMAX / Wi-Fi Networks. In *In Proceedings of the IEEE International Conference on Pervasive Services (ICPS 2007)*, july 2007.

[Mur88] K. Murphy. Switching kalman filters. Technical report, Dept. of Computer Science, University of California, Berkeley, Tech. Rep, 1988.

[Nav] Navizon. http://www.navizon.com/.

[NIS07] NIST/SEMATECH. *e-Handbook of Statistical Methods.* NIST, April 2007. http://www.itl.nist.gov/div898/handbook/.

[oDoT01] Department of Defense and Department of Transportation. *2001 Federal Radionavigation Systems.* Department of Defense and Department of Transportation, National Technical Information Service, Springfield, Virginia 22161, 2001.

[OVLdL05] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate GSM Indoor Localization. *the Seventh International Conference on Ubiquitous Computing (UbiComp 2005)*, 2005.

[Par00] J.D. Parsons. *The mobile radio propagation channel.* Wiley, 2000.

[PBK04] BS Peterson, RO Baldwin, and JP Kharoufeh. A specification-compatible bluetooth inquiry simplification. *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 307–315, 2004.

[PBK06] BS Peterson, RO Baldwin, and JP Kharoufeh. Bluetooth Inquiry Time Characterization and Selection. *Mobile Computing, IEEE Transactions on*, 5(9):1173–1187, 2006.

[PKC02] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis. On indoor position location with wireless LANs. *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, 2, 2002.

[Pos] True Position. http://www.trueposition.com/.

[RWM$^+$07] F. Rosa, S. Wardana, C. Mayorga, G. Simone, M. Raynal, J. Figueiras, and S. Frattasi. Experimental Activity on Cooperative Mobile Positioning in Indoor Environments. In *In Proceedings of the 2nd IEEE Workshop on Advanced Experimental Activities on Wireless Networks & Systems (EXPONWIRELESS)*, june 2007.

[S$^+$06] G. Senarath et al. Multi-hop Relay System Evaluation Methodology (Channel Model and Performance Metric). *IEEE 802.16s Relay Task Group*, October 2006.

[Sah06] Sinan; Sahinoglu, Zafer; Gezici. Ranging in the ieee 802.15.4a standard. In Wizard V. Oz and Mihalis Yannakakis, editors, *Proc. IEEE Annual Wireless and Microwave Technology Conference, 2006 (WAMICON '06)*, pages 1–5, December 2006.

[SIG03] Bluetooth SIG. *Bluetooth V1.1 core specifications.* Bluetooth SIG, http://www.bluetooth.com, 2003.

[SN03] A.H. Sayed and Yousef N.R. *Wireless Location.* Wiley Encyclopedia of Telecommunications. J. Proakis, John Wiley & Sons, New York, 2003.

[Soc03] IEEE Computer Society. Part 15.4: Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE 802.15.4*, 802, October 2003.

[Son99] T.L. Song. Observability of target tracking with range-only measurements. *Oceanic Engineering, IEEE Journal of*, 24(3):383–387, 1999.

[Sta01] EV Stansfield. Introduction to Kalman Filters. *IEE Signal Processing, Kalman Filter Tutorial, March*, 2001.

[TCS97] R. Todling, SE Cohn, and NS Sivakumaran. Suboptimal Schemes for Retrospective Data Assimilation Based on the Fixed-Lag Kalman Smoother. *Monthly Weather Review*, 126(8):2274–2286, 1997.

[TV04] E. Trevisani and A. Vitaletti. Cell-ID location technique, limits and benefits: an experimental study. *Proceedings of the 6th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2004), IEEE Computer Society Press*, 2004.

[Ubi] Ubisense. http://www.ubisense.net/.

[VCJ04] S. Venkatraman and J. Caffery Jr. Hybrid TOA/AOA techniques for mobile location in non-line-of-sight environments. *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, 1:21–25, 2004.

[VPP97] MC Vanderveen, CB Papadias, and A. Paulraj. Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array. *Communications Letters, IEEE*, 1(1):12–14, 1997.

[WFPS] Skyhook Wireless Wi-Fi Positioning System. http://www.skyhookwireless.com/.

[WTN04] Z. Wang, EK Tameh, and AR Nix. Statistical peer-to-peer channel models for outdoor urban environments at 2 GHz and 5 GHz. *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, 7, 2004.

[YASN02] M.A. Youssef, A. Agrawala, A.U. Shankar, and S.H. Noh. A Probabilistic Clustering-Based Indoor Location Determination System. 2002.

Vita

João Figueiras received his *Licenciatura* (five-year) degree in Electrical Engineering and Computer Science, from the Technical University of Lisbon (IST-UTL), Portugal, in 2004. During his *Licenciatura* degree, he was a guest student at Aalborg University, Denmark, where he wrote his final thesis. From November 2004 onwards, he has been employed by the Dept. of Electronic Systems, Networking and Security Group, Aalborg University as a PhD student under the project Wireless Access Network Devices & Applications (WANDA). During his PhD studies, he was a short-term research visitor in the Electrical Engineering Department of UCLA, CA, USA and in the Mobile Team at Google Inc, Mountain View, USA. Parallel to his PhD, he has been an active member of IEEE. His major activity concerns management and organization of the technical conference aiming at student, AISPC'08.

Permanent Address: Urbanizacao Monte da Vinha L8
2925 Azeitão
Portugal