Computer Systems and Robotics Laboratory
Electrical Engineering Department
Federal University of Minas Gerais (UFMG)
Av. Antônio Carlos 6627, CEP 31270-901,
Belo Horizonte, MG Brasil

CORO
Computação e Robótica-UFMG

State estimation of aerial vehicles flying near the ground

Antonio Carlos Bana Chiella

Doctoral dissertation submitted to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Advisor: Prof. Guilherme Augusto Silva Pereira, Dr.

Co-Advisor: Prof. Bruno Otávio Soares Teixeira, Dr.

Belo Horizonte, September, 2019

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

STATE ESTIMATION OF AERIAL VEHICLES FLYING NEAR THE GROUND

Antonio Carlos Bana Chiella

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Prof. Guilherme Augusto Silva Pereira

Belo Horizonte - MG

Setembro de 2019

C533s

Chiella, Antonio Carlos Bana.
State estimation of aerial vehicles flying near the ground [recurso
eletrônico] / Antonio Carlos Bana Chiella. - 2019.
1 recurso online (xxvii,134 f. : il., color.) : pdf.

Orientador: Guilherme Augusto Silva Pereira.

Tese (doutorado) - Universidade Federal de Minas Gerais,
Escola de Engenharia.

Anexos: f. 125-134.

Bibliografia: f. 113-123.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Kalman, Filtragem de - Teses.
3. Quatérnios - Teses. 4. Veículos autônomos - Teses. I. Pereira,
Guilherme Augusto Silva. II. Universidade Federal de Minas Gerais.
Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica: Biblioteca Profº Mário Werneck, Escola de Engenharia da UFMG

TESE DE DOUTORADO Nº 305

STATE ESTIMATION OF AERIAL VEHICLES FLYING NEAR THE GROUND

Antonio Carlos Bana Chiella

DATA DA DEFESA: 04/09/2019

"State Estimation of Aerial Vehicles Flying Near the Ground"

Antonio Carlos Bana Chiella

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 04 de setembro de 2019.

Por:

Prof. Dr. Guilherme Augusto Silva Pereira
DEE (UFMG) - Orientador

Teixeira.

Prof. Dr. Bruno Otávio Soares Teixeira
DELT (UFMG) - Coorientador

Prof. Dr. Leonardo Antônio Borges Tôrres
DELT (UFMG)

Prof. Dr. Guilherme Vianna Raffo
DELT (UFMG)

Prof. Dr. João Yoshiyuki Ishihara
Engenharia Elétrica (UNB)

Marco Henrique Terra

Prof. Dr. Marco Henrique Terra
Engenharia Elétrica (USP/SC)

To my father Carlos, for being an example of courage and determination.

To my mother Leonice, for your serenity and tenderness.

To my sister Thayse, for always encouraging me to follow my dreams.

Resumo

Veículos autônomos representam um grande avanço tecnológico. No entanto, o caminho para alcançar plataformas completamente autônomas ainda está sendo traçado, contendo inúmeros desafios que envolvem as áreas de controle, guiagem e estimação de estados. Entre os sistemas do veículo, os sistemas de estimação de estados são componentes vitais em um veículo auto-guiado, uma vez que esses provem a informação utilizada pelos sistemas de guiagem e controle. Contudo, estimar os estados de veículos móveis não é uma tarefa trivial, principalmente para aqueles que operam próximo ao solo, onde a degradação ou até mesmo a ausência do sinal de sistemas de navegação global, como o GPS, interferem diretamente na estimação. Além disso, para veículos que executam manobras agressivas e têm a possibilidade de realizar movimentos com 6 graus de liberdade, o problema é ainda mais desafiador. Neste contexto, esta tese aborda o problema de estimação de estados de sistemas móveis operando próximos ao solo, que apresentam as dificuldades mencionadas acima, mas também permitem o uso de diferentes tipos de sensores, tais como câmeras e LiDAR. Grande atenção é dada a veículos aéreos equipados com tais sensores, sendo a velocidade, posição e orientação os principais estados a serem estimados. Considerando a natureza não linear do sistema, novos algoritmos de fusão sensorial baseados no filtro de Kalman *unscented* são propostos para combinar dados de sensores heterogêneos de forma robusta e ainda manter a restrição de norma unitária imposta na representação baseada em quatérnios unitários da atitude do veículo. Sendo capazes de se adaptar a variações das incertezas dos dados sensoriais, os algoritmos propostos conseguem mitigar os efeitos de erros de medição desconhecidos de forma a apresentar uma boa estimativa dos estados. Vários resultados experimentais com veículos aéreos em situações de incerteza na medição variantes no tempo, incluindo um avião de corrida equipado com câmera realizando manobras acrobáticas e um quadrirrotor equipado com sensor LiDAR voando em uma floresta densa, são apresentados.

Abstract

Autonomous vehicles represent an enormous technological advance. Nevertheless, the road toward a completely autonomous vehicle platform is still being drawn, containing numerous challenges that involve control, guidance, and state estimation. Among the vehicle systems, the state estimation systems are vital components for self-guided vehicles, since they provide information used by guidance and control systems. However, estimating the vehicle states is not a trivial task, especially for those operating near the ground in the presence of vegetation, buildings, and mountains, where degradation or even the absence of the signal from global navigation systems, such as GPS, directly interferes in the estimation. In addition, for vehicles that perform aggressive maneuvers and have the ability to perform 6-degree of freedom movements, the problem is even more challenging. In this context, this thesis addresses the problem of state estimation of mobile systems operating near the ground, which presents the difficulties mentioned above, but also allows for the use of different sensors, such as cameras and LiDARs. Special attention is given to air vehicles equipped with such sensors, in which velocity, position, and orientation are the main states to be estimated. Taking into account the nonlinear nature of the system, new unscented Kalman filter-based sensor fusion algorithms are proposed to robustly merge heterogeneous sensor data and still maintain the constraint of unitary norm imposed on the unit quaternion representation of the vehicle attitude. Being able to adapt to uncertainty time variations in sensor data, the proposed algorithms can mitigate the effects of unknown measurement errors thus providing a good estimation of the states. Several experimental results with air vehicles in situations of time-varying measurement uncertainty, which include a race airplane equipped with camera performing acrobatic maneuvers and a quadrotor equipped with planar LiDAR flying in a dense forest, are presented.

Agradecimentos

Faço desse, um momento para expressar minha gratidão a pessoas queridas e de muita importância em minha vida, que contribuíram e me deram suporte para a conclusão dessa tese.

Agradeço a minha família, minha mãe Leonice, meu pai Carlos e minha irmã Thayse, por todo carinho, amor e incentivo ao longo dos anos de estudo e trabalho.

Agradeço a minha amiga e namorada Graziele, por todo carinho, apoio e compreensão.

Aos amigos que fiz na UFMG, Marcus, Daniel, Petrus, Wendy, Rodrigo, Anny, Juan José, Juan Camilo, Jaime, Stella e Ana Paula, obrigado por compartilharem comigo conhecimento, e os momentos vivenciados.

Não me esquecendo dos amigos de longa data, Bruna e Jhonatan, que sempre me ouviram e apoiaram.

Aos colegas do laboratório CORO, Henrique, Adriano e Victor, obrigado pelo ótimo ambiente de trabalho, pelo conhecimento compartilhado e pela ajuda nos experimentos. Aproveito para agradecer ao professor Paulo Iscold, que gentilmente disponibilizou dados para alguns dos experimentos.

Aos meus orientadores, Guilherme e Bruno, muito obrigado pela amizade, pela paciência, por me aconselharem e por compartilharem comigo suas ideias. A nossa convivência influenciará toda a minha vida.

Agradeço aos professores, funcionários e alunos do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE/UFMG) pela convivência e apoio, fundamentais para elaboração desta tese.

Aproveito para agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo auxílio financeiro concedido durante a realização deste projeto de pesquisa.

Por fim, estendo meus agradecimentos a todos aqueles com quem tive a oportunidade de conviver e trabalhar e que participaram direta ou indiretamente do processo de conclusão deste trabalho e que aqui não estão citados. Muito obrigado!

Acknowledgments

I make this a moment to show my gratitude to dear and very important people in my life who have contributed and supported me in completing this thesis.

I thank my family, my mother Leonice, my father Carlos and my sister Thayse, for all the affection, love and encouragement over the years of study and work.

I thank my friend and girlfriend Graziele, for all the affection, support, and understanding.

To the friends who I made at UFMG, Marcus, Daniel, Petrus, Wendy, Rodrigo, Anny, Juan José, Juan Camilo, Jaime, Stella, and Ana Paula, thank you for sharing the knowledge and the moments experienced.

I have not forgotten the longtime friends, Bruna and Jhonatan, who have always listened and supported me.

To the lab colleagues CORO, Henrique, Adriano, and Victor. Thank you for the great working environment, the shared knowledge and the help in the experiments. I would like to thank Professor Paulo Iscold, who kindly provided data for some of the experiments.

To my advisors, Guilherme and Bruno, thank you for your friendship, patience, for advising me and for sharing your ideas with me. Our coexistence will influence my whole life.

Thanks for teachers, staff, and students of the Graduate Program in Electrical Engineering (PPGEE / UFMG) coexistence and support, fundamental for the elaboration of this thesis.

I would like to thank the Coordination for the Coordination Improvement of Higher

Education Personnel (CAPES), the National Council for Scientific and Technological Development (CNPq), and Minas Gerais Research Foundation (FAPEMIG) for the financial support granted during the execution of this research project.

Finally, I extend my thanks to all those with whom I have the opportunity to live and work and who directly or indirectly makes the process of completing this work and are not mentioned here. Thank you!

Contents

Resumo **v**

Abstract **vii**

List of Figures **xvi**

List of Tables **xxi**

Acronyms **xxiii**

Symbols and Notation **xxv**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 3

1.3 Research Objectives 4

1.4 Contributions 5

1.5 Thesis Outline 6

2 Fundamental Principles and Literature Review **9**

2.1 Navigation Systems 9

2.1.1 Conventional Integrated Navigation Systems 11

2.1.2 Other Navigation Systems 12

2.2 State Estimation in Navigation Systems 14

2.3 Time Varying Uncertainty and Measurement Failures 16

2.4 Chapter Summary 19

3 Nonlinear State Estimation and new Robust/Adaptive Algorithms **21**

3.1 The State Estimation Problem 22

3.2 Unscented Kalman Filter 23

3.2.1 Unscented Transform 23

3.2.2 The Unscented Kalman Filter Algorithm 26

3.3 Adaptive Filtering Approaches 28

3.3.1 Covariance-Matching Approach 29

3.3.2 Interacting Multiple Model 30

3.4 Outlier Filtering 32

3.4.1 Outlier Filtering Based on the Chi-Squared Test 33

3.4.2 Outlier Filtering Based on Hampel Identifier 34

3.5 Unscented Rauch-Tung-Stribel Smoother 35

3.6 Robust Adaptive Unscented Kalman Filters: New Algorithms 37

3.7 Chapter Summary 38

4 Quaternion-based Kalman Filtering 39

4.1 Quaternion Operations 40

4.1.1 Fundamentals of Unit Quaternions Algebra 40

4.1.2 Euclidean Tangent Space and Rotation Vector Parameterization . . 41

4.1.3 Subtraction, Sum, and Weighted Mean Operations 43

4.1.4 Unit Quaternion Statistics 44

4.2 Quaternion-based State Estimation 45

4.2.1 Quaternion *Unscented* Transform 46

4.2.2 Quaternion-based UKF 48

4.3 Quaternion-based Robust Adaptive State Estimators: New Algorithms . . 49

4.3.1 Quaternion-based Robust Adaptive Unscented Kalman Filter 49

4.3.2 Quaternion-based Interacting Multiple Model Filter 50

4.3.3 Quaternion-based Unscented Rauch-Tung-Stribel Smoother 52

4.4 Chapter Summary 53

5 Case Study: Localization of Aerobatic Airplanes Equipped with Cameras 55

5.1 Introduction 55

5.2 Problem Statement 56

5.3 Mathematical Modeling 58

5.3.1 Process Model 58

5.3.2 Observation Model 61

5.4 Experimental Results 63

5.4.1 Filtering and Smoothing Application 64

5.4.2 Comparison Between Filters and Smoothers 65

5.4.3 Adaptability of Filters 67

5.4.4 Discussion 68

5.5 Concluding Remarks 70

6 Case Study: Attitude Estimation 71

6.1 Introduction 71

6.2 Mathematical Modeling 72

6.2.1 Kinematic Model of Attitude 72

6.2.2 Observation Model 73

6.3 Experimental Results and Discussion 74

6.3.1 Magnetic Field Distortion 77

6.3.2 Linear Acceleration Disturbance 79

6.3.3 Rotations Around the Origin 81

6.4 Conclusions 83

7 Case Study: Localization of Aerial Vehicles in Forest Environments 87

7.1 Introduction 87

7.2 Problem Statement 89

7.3 LiDAR-Based Motion Estimation in Forests 90

7.4 Mathematical Modeling 93

7.4.1 Process Model 93

7.4.2 Observation Model 95

7.5 Offline Experimental Results with Actual Data 96

7.6 Onboard Experimental Results 102

7.7 Conclusions 105

8 Conclusion and Future Work 107

8.1 Concluding Remarks 107

8.2 Future Work 110

References 123

A Gaussian Bayesian Filtering and Smoothing 125

A.1 Recursive Bayesian Approach 125

A.2 The Kalman Filter 129

A.3 The Raunch-Tung-Striebel Smoother Algorithm 131

A.4 Multiple Model Approach 132

List of Figures

1.1 Example of autonomous vehicles. 2

2.1 Essential subsystems of a navigation system: sensing and the state estimation subsystems. The sensing subsystem comprises a set of sensors whose output is used by the state estimation subsystem. The state estimation subsystem processes the received information to provide an estimate of the vehicle states. 10

2.2 An integrated navigation system, where dead reckoning and position-fixing are used to compute the vehicle position. Notice that, the navigation error is unbounded for dead reckoning, but its limit is updated by the position-fixing method. 11

3.1 Comparison between unscented transform and first order linearization. . . 26

3.2 Block diagram of the UKF. The filtering algorithm is performed in two steps, *forecast* and *data-assimilation* steps. In the first step, the mean $\hat{x}_{k|k-1}$ and covariance $P^{xx}_{k|k-1}$ are computed based on previous estimates $\hat{x}_{k-1}$ and P^{xx}_{k-1}. Then, based on measurements y_k, the estimates are updated. 28

3.3 Block diagram of the IMM algorithm. 31

3.4 Graphical illustration of the weight factor behavior for one dimension. . . . 34

3.5 Block diagram of the URTS smoother. The somoothing algorithm runs backward in time, from k_f to k_0, and is performed in two steps, named *forecast* and *smoothing.* In the first step, the mean $\hat{x}_{k+1|k}$ and covariance $P^{xx}_{k+1|k}$ are computed based on filter estimates $\hat{x}_k$ and P^{xx}_k. Then, the smoothed estimates $\hat{x}^s_k$ and covariance $P^{xx,s}_k$ are computed in the smoothing steps. The smoothed states its covariance are initialized as $\hat{x}^s_{k_f} = \hat{x}_{k_f}$ and $P^{xx,s}_{k_f} = P^{xx}_{k_f}$, respectively. 36

3.6 Block diagram of the RAUKF-CMl. The filtering algorithm is performed in three steps, called *forecast*, *robust noise estimation*, and *data-assimilation.* In the first step, the mean $\hat{x}_{k|k-1}$ and covariance $P^{xx}_{k|k-1}$ are computed based on previous estimates $\hat{x}_{k-1}$ and P^{xx}_{k-1}. Then, the output measurement noise covariance matrix $\hat{R}_k$ is robust estimated based on sliding sampling window $\{\nu_{k-N+1}, ..., \nu_{k-1}\}$ and measurement y_k. In the third step, the state estimates are updated. 38

4.1 Example of quaternion operations on a 2-*sphere*, $\mathcal{S}^2$. The unit quaternions e_a, e_b, e_c, and e_m can be seen as points along the surface of $\mathcal{S}^2$. Observe that, e_m is a result of weighted mean operation between e_a and e_b, choosing $W = [0.5]_{2\times1}$. The subtraction operation between e_b and e_c, or, e_a and e_c, yields r_{bc} or r_{ac}, in the tangent space, respectively. The unit quaternions e_a or e_b, can be recovered by sum operation between e_c and r_{ac}, or, e_c and r_{bc}, respectively. 44

5.1 Camera providing an image similar to the pilot view. A vision system detects the georeferenced landmarks (race gates), marked by green rectangles. 56

5.2 Architecture of the flight path estimator. Blocks in the left represent the set of sensors that provide measurement information. In the right, the gray block represent filtering algorithms. The dashed border white blocks represent the sub-steps of filtering algorithm. The UT block represents a preprocessing step, where the measured attitude in Euler angles and geodetic coordinates are converted to unit quaternion and Cartesian coordinate, respectively. Alternatively, a smoother can be used to yield $\hat{x}^s_k$ and $P^{xx,s}_k$. 57

5.3 2D flight path using QAUKF-CM/smoother algorithms. (a) *Dataset-1* and (b) *Dataset-2*. GNSS data (black dot), smoother algorithm (in solid green line), filter algorithm (in solid blue line), and visual *landmarks* (orange bars and orange dot). 66

5.4 Window of data corresponding to the position x_{N}, in thin solid line, and plus and minus three standard deviation $\pm3\sigma_{x_{\mathrm{N}}}$, in thick solid line, where $\sigma_i \triangleq \sqrt{P^{xx}_{k|k,ii}}$. The results were estimated by QAUKF-CM (green line) and QAUKF-CM-Smoother (blue line). 67

5.5 Estimated position and corresponding standard deviation, where $\sigma_i \triangleq \sqrt{P^{xx}_{k|k,ii}}$, using different algorithms. The gray-shaded regions illustrate some contaminated GNSS measurements or outages of the GNSS signal. . 69

6.1 Experimental setup using the MicroStrain 3DM-GX1® IMU and the Comau Smart Six robot. 74

6.2 Results for abrupt magnetic disturbances experiment, scenario (i). In the left column, linear acceleration a_{m} and magnetic field b_{m} measurements, in the right column, the attitude error. 78

6.3 QIMM mode of operation for abrupt magnetic disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of heading angle. . . 79

6.4 Results for slow varying magnetic disturbances experiment, scenario (i). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error. 80

6.5 Results for abrupt magnetic disturbances experiment, scenario (i). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error. 81

6.6 QIMM mode of operation for slow varying magnetic disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of heading angle. 81

6.7 Results for linear acceleration disturbance experiment, scenario (iii). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error. 82

6.8 Angular rate ω_x and ω_y, respectively, measured by the gyros for third experiment, scenario (ii). 83

6.9 QIMM mode of operation for linear acceleration disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of yaw angle. 83

6.10 Results for individual axis rotation about the origin, scenario (iv). Figure (a) shows actual orientation for individual axis movements; Figures (b), (c) and (d) show the estimation error for ϕ, θ and ψ angles, respectively. . . . 84

6.11 Results for simultaneous axes rotation about the origin, scenario (v). Figure (a) shows actual orientation for simultaneous axis movements; Figures (b), (c) and (d) show the estimation error for ϕ, θ and ψ angles, respectively. 85

7.1 The quadrotor Matrice 100 DJI used in the state estimation experiment. The vehicle is commercialized with built-in AHRS and GNSS, and was equipped with a Hokuyo UTM-30LX-EW planar LiDAR. 88

7.2 Block diagram of the proposed navigation solution. Data from the available sensors are combined using a robust and adaptive version of UKF (RAUKF) [Chiella et al., 2019c]. 90

7.3 Procedure to compute the radius of tree trunk. 92

7.4 a) Original laser scan in blue and the detected discontinuities in red. The blue dots represent the laser beams, and the tree trunks are represented by red circles. 92

7.5 Satellite view of the forest where our experiments were executed. Image provided by Google Maps. 97

7.6 Two dimensional path estimated by the proposed RAUKF-CMχ^2 algorithm, DJI's proprietary GNSS solution and laser-odometry (LO). 98

7.7 Velocity in East direction estimated by our algorithm and by DJI's GNSS solution. 98

7.8 Two dimensional path estimated by the RAUKF-CMχ^2 both, with normal and without abnormal measurements, and by the UKF with abnormal measurements. 99

7.9 Position in the East direction estimated by the RAUKF-CMχ^2 both, with normal and abnormal measurements, and by the UKF with abnormal measurements. 100

7.10 Three standard deviation $\sigma_\mathrm{E} = \sqrt{P^{xx}_{k,ii}}$ of position estimate in East direction.100

7.11 Innovation in the East direction regarding to the GNSS measurement, ν_E, and its uncertainty given by three standard deviation $\sigma_\mathrm{E} = \sqrt{P^{yy}_{ii}}$. The discontinuity in data is associated with GNSS data drop out. 101

7.12 Innovation in the East direction regarding to the GNSS measurement, ν_N, and its uncertainty given by three standard deviation $\sigma_\mathrm{N} = \sqrt{P^{yy}_{ii}}$. The discontinuity in data is associated with GNSS data drop out. 101

7.13 Block diagram of the proposed navigation solution. Data from the available sensors are combined using a robust and adaptive version of UKF (RAUKF-CMχ^2). The filter output feeds the motion control system, which consists of a path planner and velocity controller. 103

7.14 Satellite view of the forest where our onboard experiments were executed. Image provided by Google Maps. The path estimated by RAUKF-CMχ^2 is shown in red, when AHRS, LO, and GNSS were available and in yellow GNSS was unavailable. 104

7.15 Three-dimensional path estimated by RAUKF-CMχ^2 (red), for the target surveillance path represented by the black curve. Observe that the estimated path deviates from the target one due to the presence of trees, represented by brown cylinders. It is important to mention that, since the diameter and position of the trees are not known, those cylinders are coarse approximations of the actual forest, used for visualization only. 105

7.16 (a) and (b) Position and velocity in the East direction; (c) and (d) three standard deviation $\sigma_\mathrm{E} = \sqrt{P^{xx}_{k,ii}}$ of position and velocity estimates in the East direction. 106

A.1 Sequential Bayesian filtering. 128

List of Tables

6.1 Root Mean Square Error (RMSE) in degrees for disturbance scenarios (i) and (ii). The lowest RMSE results are highlighted in bold. 75

6.2 Root Mean Square Error (RMSE) in degrees for disturbance scenarios (iii), (iv), and (v). The lowest RMSE results are highlighted in bold. 75

Acronyms

AHRS	Attitude and Heading Reference System;
CM	Covariance-Matching;
DOF	Degree of Freedom;
EKF	Extended Kalman Filter;
FDI	Fault Detection and Isolation;
FPR	Flight Path Reconstruction;
GLONASS	*Globalnaya Navigationnaya Sputnikovaya Sistema*;
GNSS	Global Navigation Satellite Systems;
GPS	Global Positioning System;
ICP	Iterative Closest Point;
IMM	Interacting Multiple Model;
IMU	Inertial Measurement Unit;
INS	Inertial Navigation System;
KF	Kalman Filter;
LO	Laser Odometry.
MAV	Micro Aerial Vehicle;
ML	Maximum Likelihood;
MM	Multiple Model;
MMAE	Multiple Model Adaptive Estimation;
MMSE	Minimum Mean-Squared Error;
NED	North-East-Down;
ODE	Ordinary Differential Equation;
PDF	Probability Density Function;
QAUKF-CM	Quaternion-based Adaptive Unscented Kalman Filter by using Covariance Matching;
QAUKF-MCM	Quaternion-based Adaptive Unscented Kalman Filter by using Mean and Covariance Matching;
QIMM	Quaternion-based Interacting Multiple Model Filter;

QRAUKF-CMχ^2	Quaternion-based Robust Adaptive Unscented Kalman Filter by using Covariance Matching and χ^2-Test;
QRAUKF-CMH	Quaternion-based Robust Adaptive Unscented Kalman Filter by using Covariance Matching and Hampel Identifier;
QUKF	Quaternion-based Unscented Kalman Filter;
QURTS	Quaternion-based Unscented Rauch-Tung-Stribel Smoother;
RAUKF-CMχ^2	Robust Adaptive Unscented Kalman Filter by using Covariance Matching and χ^2-Test;
RAUKF-CMH	Robust Adaptive Unscented Kalman Filter by using Covariance Matching and Hampel Identifier;
RBAR	Red Bull Air Race.
ROS	Robot Operating System;
SP	Sigma Points;
UAV	Unmanned Aerial Vehicle;
UKF	Unscented Kalman Filter;
URTS	Unscented Rauch-Tung-Stribel Smoother;
UT	Unscented Transformation;
VO	Visual Odometry.

Symbols and Notation

$\mathbb{R}$	set of real numbers;
$\mathbb{N}$	set of nonnegative integers;
$\triangleq$	equals by definition;
$\mathcal{M}$	discrete set of events;
$\mathcal{S}^3$	unit sphere in the $\mathbb{R}^4$ with center at the origin;
$\mathbb{H}$	set of quaternions;
$\mathbb{H}_1$	set of unit quaternions;
$I_{n\times n}$	$n \times n$ identity matrix;
$[0]_{n\times m}$	$n \times m$ zero matrix;
$[A]_m$	$n \times m$ matrix whose columns are equal to $A \in \mathbb{R}^n$;
$(A)^{1/2}$	square-root of $(A) = A^{1/2}\left(A^{1/2}\right)^T$;
$(A)^{-1}$	inverse of (A);
$(A)^T$	transpose of (A);
$\mathrm{diag}(A_1, ..., A_n)$	block diagonal matrix;
$\max(A, B)$	return the largest element taken from A or B;
$\det(A)$	determinant of matrix A;
A_{ij}	scalar in the ith row and jth column of the matrix A;
$\exp(\cdot)$	exponential function
$\mathrm{vec}(e)$	given a quaternion $e \in \mathbb{H}$ return a 4-dimensional vector in $\mathbb{R}^4$;
$\times$	cross product;
$\otimes$	quaternion product;
$\oplus$	quaternion sum;
$\ominus$	quaternion subtraction;
$(\cdot)^*$	quaternion conjugate;
$\lVert \cdot \rVert$	Euclidean norm;
$\mathrm{E}[\cdot]$	mathematical expectation;
$\rho(x\vert y)$	conditional probability density function of x given y;

$\rho(x)$	probability density function of x;
$\mathcal{N}(\cdot)$	normal distribution;
$\Pr[\cdot]$	probability of an event
$\bar{x}$	true mean of the random variable x;
$\bar{P}^{xx}$	true covariance matrix for the random variable x;
t	continuous-time index;
k	discrete-time index;
T	sampling period;
f	process model in state space;
h	observation model;
$\mathrm{UT}(\cdot)$	unscented transform function;
$\mathrm{QUT}(\cdot)$	quaternion unscented transform function;
x_k	state vector;
u_{k-1}	input vector;
w_{k-1}	process noise vector;
y_k	output measurement vector;
v_k	measurement noise vector;
Q_{k-1}	process-noise covariance matrix;
R_k	measurement-noise covariance matrix;
$\hat{x}_{k\|k-1}$	forecast state estimate;
$\hat{\mathring{x}}_{k\|k-1}$	forecast of augmented state estimate;
$\hat{x}_k$	data-assimilation state estimate;
$\hat{\mathring{x}}_k$	data-assimilation of augmented state estimate;
ν_k	innovation;
$P^{xx}_{k\|k-1}$	forecast error-covariance matrix;
P^{xx}_{k}	data-assimilation error-covariance matrix;
$P^{yy}_{k\|k-1}$	innovation covariance matrix;
$P^{xy}_{k\|k-1}$	cross-covariance matrix;
K_k	Kalman gain at time k;
$\mathcal{X}_{j,k\|k}$	jth column of the sigma-point matrix $\mathcal{X}_{k\|k}$;
γ_j	jth weighting factor for calculating the state estimate and error covariance;
N	size of moving window;
r_k	unknown input;
$\hat{r}_{k\|k}$	unknown input estimate;
s_k	scale factor;

$\hat{x}^{f}_{k\|k}$, $\hat{x}^{b}_{k\|k}$	forward and backward state estimates;
$\hat{x}^{s}_{k}$	smoothed state estimate;
$P^{xx,f}_{k\|k}$, $P^{xx,b}_{k\|k}$	forward and backward error-covariance matrix;
$P^{xx,s}_{k}$	smoothed error-covariance matrix;
M^{j}_{k}	model j in effect at time k;
γ^{j}_{k}	jth model probability at time k;
Λ^{j}_{k}	likelihood associated to jth mode;
$\hat{x}^{i}_{k-1\|k-1}$	state estimates associated to the ith mode;
$P^{xx,i}_{k-1\|k-1}$	covariance associated to the ith mode;
$\hat{x}^{0,j}_{k-1\|k-1}$	mixed state estimates associated to the jth mode;
$P^{xx,0,j}_{k-1\|k-1}$	mixed covariance associated to the jth mode;

Chapter 1

Introduction

"Research is to see what everybody else has seen, and think what nobody has thought."

Arthur Schopenhauer

1.1 Motivation

The continuous advances in technology have made possible the use of several robotic platforms in a large variety of activities. Due to these, the scientific and industrial communities have given special attention to develop fully autonomous platforms. However, the road towards the fully autonomous navigation systems gives rise to challenging problems, which require knowledge and integration of perception, estimation, guidance, and control.

Estimation systems combine information from multiple uncertain sources, such as sensor data and mathematic models, to provide accurate estimates of vehicle states. The output of an estimation system, also referred to as navigation solution in the navigation area, is used to guide and control the vehicle. The fulfillment of the autonomous mission is directly affected by estimates, making state estimation a critical component of autonomous vehicles [Nonami et al., 2010, Kendoul, 2012].

Among the techniques to merge sensor data, probabilistic methods, mainly those based on Bayes' rule, have been applied with success in actual applications. The main Bayesian approaches applied are particle filters (PF) [Gordon et al., 1993] and some approximations based on the Kalman filter (KF) [Kalman, 1960], such as the extended Kalman filter

(a) Waymo's self-driving car. (b) Drone for delivering Amazon products.

FIGURE 1.1: Example of autonomous vehicles.

(EKF) [Jazwinski, 1970] and the unscented Kalman filter (UKF) [Julier and Uhlmann, 2004]. In fact, the essential tools for data fusion are reasonably well established, but the development and use of these tools in realistic world applications are still under development.

In the literature, authors propose Bayesian platforms that enable the fusion of heterogeneous sensors [Bachrach et al., 2010, Chambers et al., 2011, Tomic et al., 2012, Shen et al., 2014, Song et al., 2016]. However, these platforms focus on showing the feasibility of the filtering approach, and other issues, such as measurement failures, are often neglected or omitted. Missions that involve transition among different environments are only possible if the filtering approach is able to combine information from a wide variety of sensors and also is robust to the time-varying uncertainty of such data. Following this proposition, we are interested in improving existing estimation techniques in order to combine information from multiple sources, even in the presence of measurement failures.

The motivation for this work comes mainly from the applicability of autonomous platforms in the most varied tasks. Examples of applications include air and urban traffic supervision, management of natural hazards, intervention in hostile environments such as nuclear power plants, inspection and maintenance of infrastructure, precision agriculture, among others [Siciliano and Khatib, 2016]. In addition, impacting more directly the civilians lives, self-driving cars, such as the ones developed by Waymo[1], see Figure 1.1(a), and the use of aerial vehicles, such as the drones used by Amazon[2] for product delivery, see Figure 1.1(b), are still in development.

[1] https://waymo.com, September 4th, 2019

[2] https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011, September 4th, 2019

The use of global navigation satellite systems (GNSS), such as the global navigation system (GPS), is the standard approach for localization. If the vehicle operates in outdoor environments without satellite signal occlusion, such as the drone in Figure 1.1(b), which operates the majority of time at high altitude, the GNSS solution works satisfactorily. However, in missions near the ground, for instance, the autonomous car in Figure 1.1(a), the GNSS signal may be damaged or even interrupted due to the proximity of constructions and trees, or even because the mission is executed inside a building. In such cases, it is essential to use other sensors, such as laser scanners [Bachrach et al., 2010, Shen et al., 2011] and also vision camera [Weiss, 2012, Kottas et al., 2013, Schmid et al., 2013] to acquire information about the vehicle movement. Thus, by combining information from different sensors, sophisticated systems are able to estimate the location of the vehicle and even map the environment simultaneously. In that sense, the complexity of perception and state estimation systems significantly increases.

Each sensor performs differently depending on environmental conditions. For instance, visual-based algorithms are sensitive to strong sunlight and laser-based approaches need structured environments. To handle different types of uncertainty due to the operational condition of sensors is a challenging task, principally by two aspects. First, the type of information provided by the sensor may change the fusion scheme significantly. Second, standard sensor fusion algorithms consider that the measurement uncertainties are constant during the task accomplishment. In this work, the operational condition of sensors is modeled as random unknown uncertainties that vary in time.

1.2 Problem Statement

A general system can be represented by a nonlinear stochastic discrete-time dynamic system such as:

$$\begin{aligned} x_k &= f\left(x_{k-1},\, u_{k-1},\, q_{k-1},\, k-1\right), \qquad (1.1) \\ y_k &= h\left(x_k, r_k, k\right), \qquad (1.2) \end{aligned}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_q} \times \mathbb{N} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_r} \times \mathbb{N} \to \mathbb{R}^{n_y}$ are the process and observation models, respectively, $x_k \in \mathbb{R}^{n_x}$ is the state vector, $u_{k-1} \in \mathbb{R}^{n_u}$ is the input vector, $y_k \in \mathbb{R}^{n_y}$ is the measurement vector, $q_{k-1} \in \mathbb{R}^{n_q}$ and $r_k \in \mathbb{R}^{n_r}$ are, respectively, the process and measurement noise vectors, and k denotes discrete time. Assume that, for all $k \geq 1$, the known data are the measured outputs y_k, the inputs u_{k-1}, and the probability density function $\rho\left(x_0\right)$, where $x_0 \in \mathbb{R}^{n_x}$ is the initial state vector. The process and measurement disturbances are characterized by the probability density functions (PDFs) $\rho\left(q_{k-1}\right)$ and $\rho\left(r_k\right)$, which are assumed to be partially known.

Under the stated assumptions, given the dynamic and measurement models (1.1)-(1.2) and the sequence of past and present measured data $u_{1:k-1} = \{u_1, ..., u_{k-1}\}$ and $y_{1:k} = \{y_1, ..., y_k\}$, the problem consists in estimating the mean $\hat{x}_k$ and covariance P_k^{xx} that characterize the PDF $\rho(x_k|y_{1:k})$.

The state-estimation problem is challenging. First, for nonlinear systems, the PDF $\rho(x_k|y_{1:k})$ is not completely characterized by its mean and covariance [Särkkä, 2013]. Second, the uncertainties in the probabilistic model are partially unknown and can be time-varying, which means that it is a non-ergodic random process. Thus, in this thesis, we want to take advantage of the well established estimation tools for nonlinear systems, in order to find reliable estimates in the presence of time-varying uncertainty. Third, the standard tools for estimation are established to Euclidean state spaces. For localization problems, some state spaces are not Euclidean, such as the set of unit quaternions, that are constrained to the unit norm in order to represent orientation. In that case, modifications must be made in order to deal with these constrained states and the time-varying uncertainties related to them.

1.3 Research Objectives

Under the light of the previous statement, Sections 1.1 and 1.2, state estimation is an important component to achieve autonomous navigation. This system must be able to combine information from multiples heterogeneous sensors, in which the uncertainty of measurements can vary depending on the environment and dynamical conditions that the platform is exposed.

In this context, the main goal of this research is to develop state-estimation algorithms that can combine information from multiple heterogeneous sources. Actually, we do not intend to engineer the problem by changing the platform hardware, but we wish to develop general and reliable fusion algorithms that can handle time-varying uncertainty of measurements in autonomous navigation systems. In addition, we do not intend that the fusion algorithm is the only security level of estimation system, but an additional one, which means that additional systems such as diagnosis of hardware failure can be employed.

Although the developed algorithms are not limited to a specific type of system, we have a special interest in aerial vehicles. These systems are versatile, being able to work in indoor and outdoor environments, covering a large area. Thus, the main results shown in this work were obtained from experiments performed by aerial vehicles.

In line with the main objectives, we list the following specific ones:

1. Study the state-estimation problem for navigation systems. Due to inherent nonlinear nature of theses systems, we wish to investigate approximations based on the unscented Kalman filter.

2. Attitude estimation is important for navigation systems. Although unit quaternions have good properties, the unit norm constraint add challenges to the state estimation problem. Thus, to investigate estimation algorithms to handle this problem is one of the objectives of this thesis.

3. In addition to the state estimation problem, real-world applications have additional challenges regarding data-acquisition, parameter tuning, and also with respect to the implementation of the solution in an actual mobile platform. Thus, we also want to use some software tools, such as the Robot Operating System (ROS), that enable a straightforward way to handle sensor data in robotic platforms.

1.4 Contributions

The present work has ushered in three major contributions to the area of sensor fusion. Two of them are theoretical developments, whereas the third one refers to case studies, in which state-estimation algorithms are applied to solve specific problems. We summarize each theoretical contribution as follows:

1. In Chapter 3, we investigate the nonlinear filtering algorithm based on unscented Kalman filter. The standard algorithm is not able to handle time-varying uncertainty of measurements. Thus, based on adaptive approaches, we propose two new algorithms that are able to mitigate the influence of a time-varying uncertainty of output measurements. The performance of the proposed algorithms are partially illustrated in the publications:

   (a) the problem of estimating the flight path of a fixed wing aerial vehicle performing aggressive maneuvers is addressed in [Chiella et al., 2016], and;

   (b) the problem of state estimation of micro-aerial vehicles in forest environments is addressed in [Chiella et al., 2019d, Chiella et al., 2019a].

2. In Chapter 4, we investigate the problem of attitude estimation parameterized as unit quaternion. In such case, the unit norm constraint can not be ensured due to operations of sum and weighting. Based on exponential and logarithm maps, the filtering algorithms proposed in Chapter 3 are extended to deal with unit norm constraint of unit quaternions. In addition, a new interacting multiple model filter and a new Rauch-Tung-Stribel smoother for unit quaternions are proposed. The performance of the proposed algorithm is illustrated in the problem of attitude estimation using magnetic, angular rate, and gravity (MARG) sensors, whose is addressed in [Chiella et al., 2018, Chiella et al., 2019c, Chiella et al., 2019b].

   Experimental results regarding the application of the developed algorithms are presented in Chapters 5, 6, and 7.

1.5 Thesis Outline

This document is organized into eight chapters, as follows. In this chapter, we introduced the motivation to investigate state estimation for autonomous vehicles.

In Chapter 2, the state of the art related to state estimation for navigation is presented. First, we introduce the main motion estimation approaches. Then, the commonly used data fusion approaches are presented. Finally, we quickly review some works and the main concepts related to the time-varying uncertainty of measurements.

In Chapter 3, the state estimation problem for nonlinear systems in Euclidian spaces is presented. Related to the problem of time-varying uncertainty of measurements, adaptive and some basic robust approaches are reviewed. Based on these approaches, extensions to the standard UKF are proposed. Finally, considering that the state estimation procedure can be performed offline, a smoother algorithm is reviewed.

The attitude of aerial vehicles is usually represented by unit quaternions. In Chapter 4 an unscented Kalman filter version for unit quaternions is reviewed. Then, we propose four new robust adaptive unscented Kalman filters for quaternions. In addition, a new interacting multiple model filter and unscented Rauch-Tung-Striebel smoother for unit quaternions are proposed.

In Chapters 5, 6, and 7, the proposed algorithms are tested using experimental data. Three study cases are presented: i) localization of an aerial vehicle performing aggressive maneuvers; ii) localization of an aerial vehicle in a forest environment; and iii) attitude estimation using MARG sensors.

In Chapter 8, some characteristics of the state estimation algorithms investigated through this work are discussed. Conclusions with respect to the implemented approaches are formulated.

In Appendix A, the general equation for computing Bayesian filtering solutions for both linear- and nonlinear-Gaussian state-space models are presented. In addition, a formulation of the multiple model approach is presented.

Chapter 2

Fundamental Principles and Literature Review

"If I have seen further it is by standing on the shoulders of giants."

Isaac Newton

In this chapter, we review the main concepts related to integrated navigation systems and state estimation, separating the main topics in five sections. First, in Section 2.1 the terminologies and categorization of navigation systems in two different types, dead reckoning and fixed-point approaches, are presented. Then, in Section 2.1.1 and 2.1.2, we briefly introduce different types of integrated navigation solutions reported in the bibliography. State estimation techniques are the basis of navigation systems, being responsible principally for combining information from multiple uncertain data sources. In Section 2.2 the main algorithms reported in the navigation area are reviewed. Section 2.3 presents the main data fusion techniques and the related approaches to improve the estimates and mitigate the effect of time-varying uncertainties. Related works necessary to support more specific topics of the thesis will be surveyed wherever necessary.

2.1 Navigation Systems

Navigation, in the broad sense, is a method of determining and planning the movement of a vehicle from one place to another [Titterton and Weston, 2004, Kendoul, 2012, Groves, 2013]. In this work, except in specific cases, navigation only refers to a method of determining the vehicle states, such as position, orientation, and velocity with respect to a known reference frame. The output of the navigation system is called the navigation solution and can also include, in addition to vehicle states, the estimation errors.

Figure 2.1 illustrates a typical navigation system, which in turn has essentially two subsystems: sensing and state estimation subsystems. The sensing subsystem contains one or more sensors, providing information about the internal system or surrounding environment, also referred to as proprioceptive and exteroceptive sensors, respectively [Siegwart et al., 2011]. In some applications, the sensing subsystem provides preprocessed data, called pseudo-measurements. The state estimation systems are responsible for processing the sensor measurements, which are related to the vehicle states.

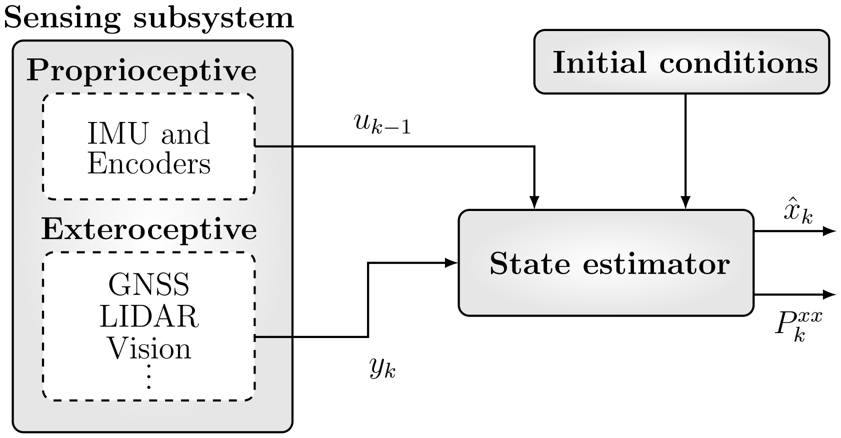


FIGURE 2.1: Essential subsystems of a navigation system: sensing and the state estimation subsystems. The sensing subsystem comprises a set of sensors whose output is used by the state estimation subsystem. The state estimation subsystem processes the received information to provide an estimate of the vehicle states.

Most navigation techniques are based on dead reckoning or/and position fixing techniques. In the first class of methods, given a starting position, the vehicle states are continually (except for hardware failure) estimated, by summing up the relative position measurements. Each of these incremental motions is subject to an error, and consequently, the error in the navigation solution can grow unbounded. In the second method, the vehicle states are computed relative to reference points, whose location is known. The navigation error is bounded in the position fixing method; however, due to possible loss of signal, the solution is not continuous.

The aforementioned techniques are usually complementary. Thus, it is common to combine the dead reckoning technique with one or more position fixing techniques in an integrated navigation system to get the benefits of both techniques. Figure 2.2 illustrates the combination of dead reckoning and position-fixing methods. During dead reckoning estimation, the estimation uncertainty, represented by ellipses, grows unbounded, then when position-fixing measurements are assimilated the uncertainty decreases.

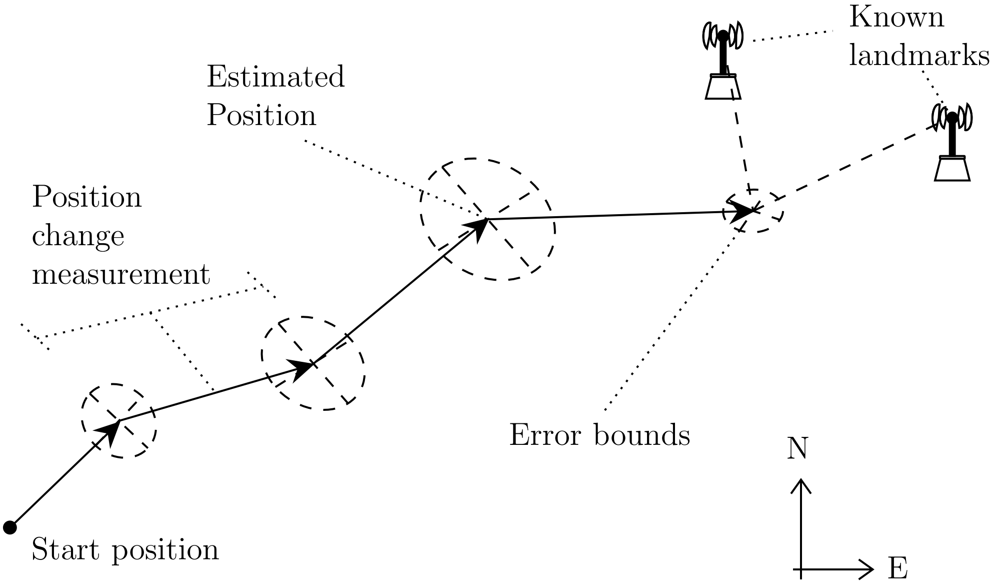


FIGURE 2.2: An integrated navigation system, where dead reckoning and position-fixing are used to compute the vehicle position. Notice that, the navigation error is unbounded for dead reckoning, but its limit is updated by the position-fixing method.

2.1.1 Conventional Integrated Navigation Systems

Conventional navigation solutions are usually performed by combining data from global navigation satellite systems (GNSS), such as Global Positioning System (GPS) and *Globalnaya Navigationnaya Sputnikovaya Sistema* (GLONASS), with an inertial navigation system (INS) and compass [Leutenegger and Siegwart, 2012]. An inertial navigation system is a complete three-dimensional dead reckoning navigation system, comprising a set of inertial sensors, accelerometers and gyrometers, known as inertial measurement unit (IMU). INS and compass provide high-frequency information, but the errors in an inertial navigation solution can grow with time as successive accelerometers and gyrometers errors are integrated. On the other hand, GNSS provides accurate absolute position and velocity at low frequency, which can be used to correct IMU-based information, forming a combination between dead-reckoning and position fixing techniques, see Figure 2.2.

The use of the GNSS solution (latitude, longitude, altitude, and velocity) in a sensor fusion algorithm is referred to as a loosely coupled integration. The main advantages of this approach in relation to techniques that use the GNSS raw data are simplicity and redundancy since other navigation systems can be easily included. A drawback is that in this approach one needs to know the covariance of the GNSS output error, which usually is not a realistic information, as it is normally based on satellite availability. The tightly coupled approach combines raw GNSS data, such as pseudo-range and pseudo-range-rate, with IMU data in the fusion algorithm. It usually performs better than a loosely-coupled approach, since the errors of raw data can be better described. However, the algorithm complexity increases, as more data must be treated by the integration system [Groves, 2013].

Commercial aerial vehicles for civil purposes normally have an embedded navigation solution based on GNSS and IMU. In such products, the raw GNSS measurements are normally not available, making the loosely-coupled approach preferable. The fusion algorithms developed in this work are tested in loosely-coupled approaches, see Sections 5 and 7, but, can be used in tightly-coupled approaches as well. The reader can find more information about tightly and loosely coupled INS/GNSS integration in [Upadhyay et al., 1993, Carvalho et al., 1997, Kendoul et al., 2010, Zhou et al., 2011, Hu et al., 2015] and [Mulder et al., 1999, De Mendonça et al., 2007, Teixeira et al., 2011, Li et al., 2016], respectively. In the aforementioned works, despite being a complete navigation solution, INS/GNSS is used together with other sensors, such as barometer and differential pressure sensor, in the localization problem.

2.1.2 Other Navigation Systems

As previously commented, localization by GNSS has become a standard approach in outdoor conditions. On the other hand, for vehicles working near the ground, such as land vehicles or aerial vehicles that flies at low altitude, the GNSS signal can be damaged by different variables, which make this navigation solution unsuitable [Kendoul et al., 2010]. Furthermore, some civilian GNSS have accuracy within several meters, which is unacceptable for localizing small mobile vehicles [Siegwart et al., 2011]. In this way, to achieve integrity and accuracy requirements, other types of integrated solutions should be provided, for instance, a vision-based and/or laser-based navigation solutions.

Camera images can yield rich information about the surrounding environment and can be used to estimate the vehicle movement. Visual odometry (VO) is a technique that uses a sequence of images to detect and track features of the environment, which estimates the vehicle movement via dead-reckoning. In [Bay et al., 2006, Zhang et al., 2014] a monocular camera is used to solve the six degrees of freedom (6DOF) estimation problem, but it may require additional information to recover the image scale. On the other hand, stereo-vision approaches, such as proposed in [Garcia and Solanas, 2004, Fraundorfer et al., 2012, Song et al., 2016] have the advantage of direct scale the observation. Another option is to use RGB-D sensors for VO [Shen et al., 2013a, Fang and Scherer, 2014, Tubman et al., 2016]. These sensors can measure the depth with high accuracy.

In some works other approaches are used to extract information from the camera. For instance, in [Kendoul et al., 2009, Weiss et al., 2012, Bin et al., 2014] optic flow is employed to estimate the vehicle velocity. In [Chiella et al., 2016], by using a priori map, vision information can be used to identify known landmarks, then estimate the vehicle pose relative to this marks. This approach provides a drift-free pose estimate since the landmark has its position known. In Chapter 5, the localization problem of an aerial vehicle, where the global position information comes from a vision system, is presented.

Similarly to INS/GNSS combination, INS/VO combination can be performed in loosely or tightly coupled approaches. Loosely coupled approaches treat the inertial and vision units as two separate modules running at different rates and exchanging information, while the tightly coupled approach combines them into a single optimal filter. Loosely coupled approaches are treated in some works such as [Armesto et al., 2004, Weiss, 2012, Shen et al., 2013b, Chambers et al., 2014], while the tightly-coupled approach is considered in [Baldwin et al., 2009, Jones and Soatto, 2011]. An advantage of loosely-coupled approaches is that the visual system can be treated as a black-box, abstracting the internal algorithms, which allows more flexibility.

Due to the low cost, low energy consumption and rich information, vision-based odometry is a good choice for providing additional measurements on small vehicles. However, vision-based approaches are strongly influenced by the change in the environment lighting.

In contrast, laser-based approaches are less influenced by the change in the environment lighting, with the price of an increase in payload and energy consumption. In [Shen et al., 2011, Tomic et al., 2012] a laser-based iterative closest point (ICP) approach is employed to solve the navigation problem. In these works, the ICP algorithm is used to estimate the position and heading of a UAV. However this approach is limited to indoor environments or tasks near the buildings that have vertical walls. In [Cui et al., 2014, Schultz et al., 2016, Chiella et al., 2019d] a similar algorithm is used to navigate a UAV in a forest. In this type of environment, due to dense tree canopy, the GNSS signal can be damaged, making the navigation a challenging task. We also address the navigation in forest environments using a laser-based approach in Chapter 7.

2.2 State Estimation in Navigation Systems

It is worthwhile to notice the localization problem can be seen as a particular case of state estimation that is limited to estimate the vehicle pose, which consists of position and orientation. The Kalman filter (KF) has been widely used for localization and path reconstruction. As the navigation equations are usually nonlinear, approximations, like the extended (EKF) and the unscented Kalman filters (UKF), are employed.

Traditionally, the navigation and inertial sensor errors are used as states, which is commonly referred as an indirect form of KF or simply by error-state-KF (ErKF) [Maybeck, 1982, Titterton and Weston, 2004, Groves, 2013]. This approach takes the same form of EKF, but instead of the system and measurement matrices being linearized about the prior state estimate, they are linearized about a predetermined state value [Groves, 2013]. A drawback is that the transformations are reliable only when the error propagation can be well described by first-order linearization approximations. For example, in aviation applications, the attitude is usually approximated by small angle errors. If an INS/GNSS integration is used, the heading error is weakly observable, requiring high dynamics of the platform during the application to properly estimate the error states, which is commonly called as in-motion alignment [Shin and El-Sheimy, 2004, Zhou et al., 2011]. Furthermore, when the heading error is large, its product with other states is not negligible. Because of this, first order approximations do not well describe the error evolution, requiring a system model that does not ignore higher order terms [Shin and El-Sheimy, 2004]. Alternatively, in a UKF navigation solution, the nonlinear propagation and observation models are directly employed in the filter, such that the nonlinearities can be better handled [Shin and El-Sheimy, 2004, Crassidis, 2006, Li et al., 2016]. One disadvantage of the UKF, if compared to the EKF, is its computational burden; however, nowadays this is not a serious constraint for most applications. For instance, even for micro aerial vehicles, where the computational resources are limited due to payload capabilities, the onboard operation of UKF-based solutions can be demonstrated [Chambers et al., 2014, Shen et al., 2014]. Furthermore, as presented in [Rosen and Medvedev, 2013] Kalman-based and particle filters can be efficiently implemented in parallel on multicore platforms.

Attitude estimation is usually performed by dedicated systems, such as attitude and heading reference systems (AHRS). It is common to represent the attitude by parameterizations such as Euler angles or unit quaternions. When the span of vehicle angles is large, what usually happens for aerial vehicles that make acrobatic maneuvers [Chiella et al., 2016], the unit quaternion parametrization is attractive due to the absence of singularities. However, KF-based algorithms pertain to the Euclidean spaces systematization, performing sum and multiplication by scalar operations. Unit quaternions are not closed under these operations, thus the unit norm constraint cannot be ensured [Choukroun et al., 2006].

In order to circumvent the unit norm constraint of unit quaternion, an indirect form of KF called multiplicative extended Kalman filter (MEKF) [Lefferts et al., 1982] was created. As previously discussed, this approach is valid only for small estimation errors, and outside of this region the error is not well represented by first-order approximations. In such case, algorithms based on the unscented UKF as the unscented quaternion estimator (USQUE) [Crassidis and Markley, 2003, Crassidis, 2006] yield better results, being less sensitive to large initialization errors, although not solving completely the problem. Since unit quaternions are constrained to a Riemannian manifold, using its *logarithm* and *exponential* maps, as in [Sipos, 2008], can better preserve its properties, such as the Riemannian metric [Menegaz et al., 2019]. A more general formulation of UKF for unit quaternions, encompassing different unit quaternion parameterizations, can be found in [Menegaz and Ishihara, 2018].

In contrast to KF approaches, which adopt a probabilistic determination of the modeled state, complementary filters (CF) are based on frequency analysis, being simplistic and usually computationally more efficient. In [Euston et al., 2008], the authors proposed an explicit complementary filter (ECF) for the orientation estimation of UAVs. Such a filter utilizes a proportional-integral (PI) controller to estimate the bias of the gyro. In [Madgwick et al., 2011], the authors present a computationally efficient gradient descent algorithm given measurements from a magnetic, rate and gravity (MARG) sensor. The proposed algorithm has low computational cost and is able to reduce the effect of the magnetic disturbance. A similar algorithm is presented in [Madgwick, 2010], which also estimates gyros' bias. The problem of this algorithm is that the orientation estimated using accelerometers suffer the influence of magnetic disturbances due to the coupling in the gradient descent algorithm used. In [Valenti et al., 2015], quaternion measurement is computed as the composition of two algebraic quaternion, which mitigates the influence of magnetic distortion.

2.3 Time Varying Uncertainty and Measurement Failures

The standard GNSS-INS solution usually works well whenever the navigation is performed in unobstructed environment, such as aerial vehicles flying at high altitudes in a leveled flight. However, if the navigation takes place close to the ground, in the presence of buildings and mountains, or if the vehicle performs acrobatic maneuvers, the GNSS satellite signal may be blocked, causing the data to be unreliable or even unavailable [Jin et al., 2014]. As aforementioned in Section 2.1.2, in these situations, other types of information sources must be used. However, all these types of navigation solutions are susceptible to errors, whose characteristics are not well known and can be time-varying. The accuracy of a navigation solution or sensor can degrade due to malfunctioning or even by a change in the operational condition. Thus, the quality of measurements is affected by different noise and/or bias amplitudes. When a measurement has an abnormal behavior, which means that it deviates from an expected behavior, referred to as normal condition, we assume we have a measurement fault [Gustafsson, 2008]. Computing this time-varying characteristics remains an open problem to robust state estimation [Cadena et al., 2016].

Fault detection methods are based on a comparison of measurements to a reference. In practice, this redundant information is derived in two different ways [Isermann, 2005, Gustafsson, 2008, Zug et al., 2012]. First, the current practice in the industry, e.g. aircraft industry, is hardware redundancy, which uses multiple parallel sensors fulfilling the same function. In that case, an increased effort in power supply and installation space is needed. An alternative to component redundancy is analytical redundancy based on a mathematical model, also referred to as *model-based* redundancy. Instead of comparing several signals for the same variable such as in hardware redundancy, model-based methods compare system output with analytically generated signals. A common way to provide fault detection is to construct residuals from measurements, which are designed to be zero in the fault-free case and nonzero otherwise.

Filtering algorithms are straightforward tools to provide analytical redundancy and consequently generate residuals. The chi-square test is the conventional and commonly used failure detection methodology to monitor dynamical systems and estimation algorithms based on the KF [Da, 1994]. The chi-square test belongs to the class of statistical hypothesis-testing mechanisms and is commonly used to examine whether the assumed mean and covariance matrix match the actual settings or not. In [Chambers et al., 2014] a chi-square test based on a single sampling time is used to reject outliers yielded by VO and GPS. This approach can lead to false alarms, rejecting measurements with good quality. To avoid false alarms, the test based on single sampling time can be replaced by a test based on a moving average [Bar-Shalom et al., 2004]. Median-based approaches can also be use to mitigate the outlier influence [Moghaddamjoo and Kirlin, 1989]. Another common statistics tests are the Sequential Probability Ratio Test (SPRT) and Cumulative Sum (CUMSUM), which is an especial case of the first one [Gustafsson, 2008]. In [Sundvall and Jensfelt, 2006] the CUMSUM test was used to detect faults in the measured pose provided by a laser-based odometry and a wheel-based odometry.

The performance of estimation algorithms based on KF are highly sensitive to parameters tuning [Tandeo et al., 2018, Shao et al., 2019], such as the process and measurement covariance matrices, named in this work by Q_{k-1} and R_k, respectively. Some adaptive approaches use the information of innovation to estimate the unknown inputs error covariance matrices. These matrices can be interpreted as a measurement of reliability of the input data. For instance, the covariance R_k of the measurement y_k describes how reliable the measurement information is. One extreme case is $R_k = 0$ that says that measurement is exact. In contrast, we have $R_k \to \infty$ which means that the measurement is useless and should be discarded. To improve its performance, one possible approach consists in modifying their tuning parameters online, which is known as adaptive Kalman filtering.

Different schemes are employed in the literature to improve the performance of KF estimating its parameters online. Covariance-matching-based (CM) approaches were employed in [Hide et al., 2004, Jwo et al., 2009, Chauhan and Gao, 2017]. These algorithms improve the filter consistency by estimating the noise covariance matrices using a N-sized window of innovation sequence. Alternatively, in [Hu et al., 2003, Hide et al., 2004, Hajiyev and Soken, 2013, Hajiyev and Soken, 2016], the approach called covariance scaling (CS) is used. In such a method, scaling factors are computed from the innovation sequence to adjust the noise covariance matrices. Algorithms based on multiple model adaptive estimation (MMAE) and interacting multiple model (IMM) were investigated in [Hide et al., 2004, Abuhashim et al., 2010, Lu et al., 2015, Jwo and Tseng, 2011] and [Jwo et al., 2013], respectively. These algorithms run multiple filters in parallel, each one with a different internal model, and the innovation behavior is used as an indicator of how each filter represents the most likely operational mode. Among these methods, the CM approaches yield improved results in the estimation of the covariance matrix for Gaussian distribution, if compared to the CS approaches. CM also presents a greater simplicity if compared to approaches based on multiple models. However, in the presence of outliers, its performance can be damaged. In a more recent approach, [Sarkka and Nummenmaa, 2009, Särkkä and Hartikainen, 2013] propose to use the variational Bayesian approach to approximate the posterior distribution and then estimate the measurement uncertainty. Numerical simulations showed performance similar to the IMM approach, but with a lower computational burden. In addition, it is important to point out that, like the KF, these adaptive approaches also belong to Euclidean systematization, thus requiring modifications to be used with unitary quaternions.

2.4 Chapter Summary

In this chapter, we review the main concepts related to navigation systems, such as the principal integrated navigation approaches and the state estimation techniques reported in the literature. For the navigation solution, Kalman filter-based approaches are by far the main used state estimation techniques. These algorithms are very sensitive to abnormal measurements, needing additional modifications, such as those based on robust and adaptive techniques. All these algorithms presented in the literature are reported with better results than the traditional KF-based algorithms. However, no extensions to unit quaternions are explored in the aforementioned papers. For instance, the algorithms based on covariance matching perform operations of sum and weighting with innovation sequence, and the multiple model approaches reconstruct its estimates from the individual results of each filter, which also involves these operations that are not closed in the set of unit quaternions.

In the next chapter, robust and adaptive techniques are combined in order to improve the performance of the UKF. Then, to handle properly the unit norm constraint of unit quaternions, we present for these algorithms some modifications based on logarithm and exponential maps. The performance of the proposed algorithms are shown by experiments with actual data.

Chapter 3

Nonlinear State Estimation and new Robust/Adaptive Algorithms

> *"The discovery of the Kalman filter came about through a single, gigantic, persistent mathematical exercise. Just as Newton was lucky having timed his birth so as to have Kepler's laws ready and waiting for him, I was lucky, too. The pieces of the slumbering Newtonian revolution, which I needed for my monster exercise, were available, scattered all around, partially forgotten but ready to be picked up and reassembled again."*
>
> Rudolph E. Kalman

This chapter presents a review of nonlinear filtering and smoothing estimators used in this thesis (sections 3.2 and 3.5, respectively). Special emphasis is put on the unscented Kalman filter and the unscented Rauch-Tung-Striebel smoother, since they allow straightforward handling of nonlinear equations through the unscented transform (UT). The chapter is also intended to introduce and explain notations and some terminologies used in the rest of the text, thus facilitating its understanding. Despite this work being focused on the multisensor fusion problem of mobile vehicles, the reviewed algorithms are presented in a general form, thus not restricting its application to this particular problem.

As introduced in the last chapters, in practical applications, the statistical properties of measurement noise are unknown or partially known. In addition, the measurement noise can be contaminated by spurious errors, such as outliers, that change its probability distribution characteristics. These errors are prejudicial to state estimation, yielding inaccurate estimates. Sections 3.3 and 3.4 are dedicated to adaptive and robust techniques. We pay special attention to the observation model, thus, the presented adaptive and robust methods are addressed to characterize the measurement noise statistics and to mitigate the effect of damaged data. Based on the reviewed methods, by combining adaptive and outlier detection approaches, we present the first theoretical results of this thesis in Section 3.6, a new family of nonlinear filtering algorithms.

3.1 The State Estimation Problem

Given the nonlinear stochastic discrete-time dynamic system (1.1)-(1.2). Assume that, for all $k \geq 1$, the known data are the measured outputs y_k, the inputs u_{k-1}, the initial estimate $\hat{x}_0 \triangleq E[x_0]$ and the initial error-covariance $P_0^{xx} \triangleq E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$, where $x_0 \in \mathbb{R}^{n_x}$ is the initial state vector and $E[\cdot]$ denotes the expected value. Furthermore, the process noise has unknown mean $E[q_k] = \hat{q}_k$ and known covariance matrix $E[q_k q_k^T] = Q_k$, the measurement noise has unknown mean $E[r_k] = \hat{r}_k$ and covariance matrix $E[v_k v_k^T] = R_k$. The process and measurement noises are also assumed to be mutually independent. The process and measurement noises are usually assumed to have zero mean, $\hat{q}_k = 0$ and $\hat{r}_k = 0$, respectively. However, in this work, we assume a more general case in order to treat the adaptive filtering.

Under the stated assumptions, given the dynamic and measurement models (1.1)-(1.2) and the sequence of past and present measured data $u_{0:k-1}$ and $y_{1:k}$, the state-estimation problem is to find the estimate $\hat{x}_k = E[x_k]$ and covariance $P_k^{xx} = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$ that characterize the *a posteriori* conditional probability density function $\rho(x_k|y_{1:k})$.

Let $\bar{x}_k$ and $\bar{P}_k^{xx}$, respectively, be given by

$$\bar{x}_k = \mathrm{E}\left[x_k|y_{1:k}\right], \tag{3.1}$$

$$\bar{P}_k^{xx} = \mathrm{E}\left[\left(x_k - \bar{x}_k\right)\left(x_k - \bar{x}_k\right)^T|y_{1:k}\right], \tag{3.2}$$

denote the true mean and covariance of $\rho(x_k|y_{1:k})$. In general, for nonlinear systems, the solution to this problem can be quite complex, due to the fact that *posterior* PDF could not be completely characterized by its mean $\bar{x}_k$ and covariance $\bar{P}_k^{xx}$. Moreover, the estimate $\hat{x}_k$ and its covariance P_k^{xx} are not necessarily equal to $\bar{x}_k$ and $\bar{P}_k^{xx}$, respectively. Due to the nonlinear characteristics of the system (1.1)-(1.2), our proposition is to use as basis to our approach the unscented Kalman filter [Julier and Uhlmann, 2004], presented in Section 3.2.

In addition to nonlinearity, more complexity is added to this problem since the uncertainty, represented by the covariance matrices Q_k and R_k, may be partially or totally unknown. Actually, in practical applications, the input u_k and output y_k measurements can suffer abnormal behaviors, changing online their statistic properties. The time-varying uncertainty of measurement drastically damage the state estimates, and thus, must be treated properly.

3.2 Unscented Kalman Filter

The unscented Kalman filter is a suboptimal solution for the stochastic filtering problem of a discrete-time dynamic system described by (1.1)-(1.2). Instead of analytically or numerically linearizing the dynamic system and using the Kalman filter equations as is done in the EKF, the unscented Kalman filter employs the unscented transform (UT) [Julier and Uhlmann, 2004].

3.2.1 Unscented Transform

The unscented transform is the main core of UKF. It approximates the posterior mean $\hat{z} \in \mathbb{R}^{n_z}$ and the covariance matrix $P^{zz} \in \mathbb{R}^{n_z \times n_z}$ of a random variable z obtained from the nonlinear transformation

$$z = g(x_1, x_2, c), \tag{3.3}$$

where $g : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_c} \to \mathbb{R}^{n_z}$ is the nonlinear transformation, $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ are *a priori* random variables with known mean $\bar{x}_1$ and $\bar{x}_2$ and covariance matrices $\bar{P}^{x_1x_1} \in \mathbb{R}^{n_1 \times n_1}$ and $\bar{P}^{x_2x_2} \in \mathbb{R}^{n_2 \times n_2}$, respectively, and c is a deterministic variable assumed to be known.

We define the augmented state vector $\bar{x} \in \mathbb{R}^n$ as

$$\bar{x} \triangleq \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \tag{3.4}$$

where $n = n_1 + n_2$, as well the augmented covariance matrix $\bar{P}^{xx} \in \mathbb{R}^{n \times n}$

$$\bar{P}^{xx} \triangleq \begin{bmatrix} \bar{P}^{x_1x_1} & [0]_{n_1 \times n_2} \\ [0]_{n_2 \times n_1} & \bar{P}^{x_2x_2} \end{bmatrix}. \tag{3.5}$$

The UT is based on a set of deterministically chosen vectors, known as *sigma points* (SP). The SP $\mathcal{X}_j \in \mathbb{R}^n$ and the associated weights w_j, $j = 1, ..., 2n$ can be chosen as [Arasaratnam and Haykin, 2009]

$$\mathcal{X} \triangleq [\bar{x}]_{2n} + \sqrt{n} \left[\left(\bar{P}^{xx}\right)^{\frac{1}{2}} \quad -\left(\bar{P}^{xx}\right)^{\frac{1}{2}} \right], \tag{3.6}$$

$$w_j \triangleq \frac{1}{2n} \tag{3.7}$$

satisfying

$$\sum_{j=1}^{2n} w_j \mathcal{X}_j = \bar{x}, \tag{3.8}$$

$$\sum_{j=1}^{2n} w_j \left(\mathcal{X}_j - \bar{x}\right)\left(\mathcal{X}_j - \bar{x}\right)^T = \bar{P}^{xx}, \tag{3.9}$$

$$\sum_{j=1}^{2n} w_j = 1, \tag{3.10}$$

where $\mathcal{X}_j$ is the jth column of the matrix $\mathcal{X} \in \mathbb{R}^{n \times 2n}$, and $(\cdot)^{\frac{1}{2}}$ is the Cholesky square root. The SP (3.6) can be partitioned as

$$\begin{bmatrix} \mathcal{X}^{x_1} \\ \mathcal{X}^{x_2} \end{bmatrix} \triangleq \mathcal{X}, \tag{3.11}$$

where $\mathcal{X}^{x_1} \in \mathbb{R}^{n_1 \times 2n}$ and $\mathcal{X}^{x_2} \in \mathbb{R}^{n_2 \times 2n}$. Then, each SP $\mathcal{X}_j$ is propagated through a transformation g yielding

$$\mathcal{Z}_j = g\left(\mathcal{X}_j^{x_1}, \mathcal{X}_j^{x_2}, c\right), \tag{3.12}$$

where $\mathcal{Z}_j \in \mathbb{R}^{n_z}$ is the jth column of the matrix $\mathcal{Z} \in \mathbb{R}^{n_z \times 2n}$.

From (3.12), we obtain the mean $\hat{z}_k$, the covariance P_k^{zz}, and the cross-covariance $P_k^{x_1 z}$, which are given by

$$\hat{z} = \sum_{j=1}^{2n} w_j \mathcal{Z}_j, \tag{3.13}$$

$$P^{zz} = \sum_{j=1}^{2n} w_j \left(\mathcal{Z}_j - \hat{z}\right)\left(\mathcal{Z}_j - \hat{z}_k\right)^T. \tag{3.14}$$

$$P^{x_1 z} = \sum_{j=1}^{2n} w_j \left(\mathcal{X}_j^{x_1} - \bar{x}_1\right)\left(\mathcal{Z}_j - \hat{y}_k\right)^T. \tag{3.15}$$

In this work, for notational simplicity, we define the unscented transform as the function UT comprising the set of equations (3.6)-(3.7) and (3.12)-(3.15), that is,

$$\left\{\hat{z}, P^{zz}, P^{x_1 z}\right\} = \mathrm{UT}\left(\bar{x}, \bar{P}^{xx}, c, g\right), \tag{3.16}$$

where $\bar{x}$ and $\bar{P}^{xx}$ are given by (3.4) and (3.5). Alternative schemes for choosing sigma points are reviewed in [Menegaz et al., 2015]. As point out by [Menegaz et al., 2015], depending on the UT chosen, a variant of UKF can be generated, which can differ from each other depending on how the transformed statistics are calculated and by the choice of sigma representation. In the case of the process noise being non-additive, the system state needs to be augmented with the noise random vectors [Haykin et al., 2001]. In the special case where the process and measurement noise are purely additive, the state vector do not need to be augmented and the covariances of noise source are incorporated into the state covariance by an additive procedure [Särkkä, 2013]. In the Section 3.2.1 a UT for non-additive process is presented. Example 3.2.1 illustrates the performance of UT and first-order linearization approaches.

Example 3.2.1. Adapted from [Simon, 2006]. Let $x = \begin{bmatrix} r \\ \theta \end{bmatrix}$ be a random vector, with

$r \sim \mathcal{N}(\bar{r}, \sigma_r^2)$ and $\theta \sim \mathcal{N}(\bar{\theta}, \sigma_\theta^2)$ undergoing the nonlinear transformation $z = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}$. We aim to obtain the mean and covariance of z. Figure 3.1 compare the results of three methods, first order linearization approach, unscented transformation and Monte Carlo sampling, which is considered to be the ground truth. We use $\bar{r} = 3$, $\sigma_r = 1$, $\bar{\theta} = 0$, and $\sigma_\theta = 1$. Compared to the Monte Carlo result, UT yields better results (regarding both mean and covariance) than the analytical linearization approach. In general, UT and first order linearization approach give similar results; however, the linearization approach is influenced by the evaluated point, yielding sometimes worse results than UT. □

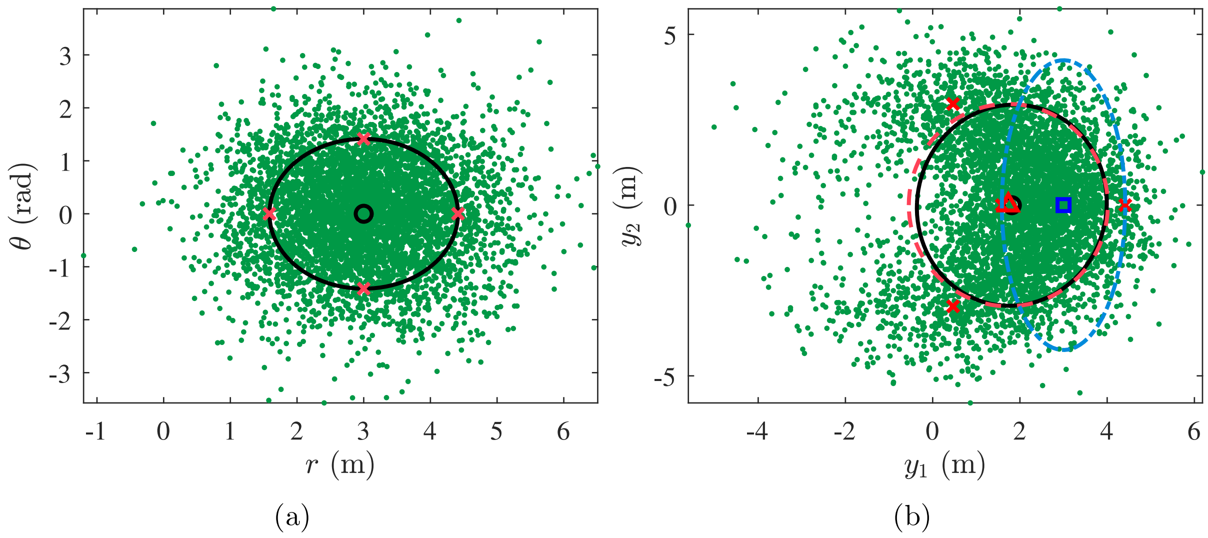


FIGURE 3.1: (a) 5000 samples (green dots) of $x \in \mathbb{R}^2$ whose mean $\bar{x} = [3\ 0]^T$ and covariance $\bar{P}^{xx} = I_{2\times2}$ are assumed to be known. The sigma points are presented by (red ×). (b) Comparison of the approximated mean $\hat{z} \in \mathbb{R}^2$ and covariance $\hat{P}^{zz} \in \mathbb{R}^{2\times2}$ of z using (i) Monte Carlo sampling (black circle and black solid ellipsoid), (ii) analytical linearization approach (blue square and blue dot-dashed ellipsoid), and (iii) unscented transform (red triangle and red dashed ellipsoid).

3.2.2 The Unscented Kalman Filter Algorithm

Figure 3.2 illustrates the UKF algorithm. Henceforth, the notation $\hat{x}_{k|k-1}$ indicates an estimate of x_k at time k based on information available up to and including time $k-1$. Likewise, $\hat{x}_k$ indicates an estimate of x_k at time k based on information available up to and including time k. Let the process noise vector be partitioned as $q_{k-1} \triangleq \left[q_{1,k-1}^T \;\; q_{2,k-1}^T\right]^T \in \mathbb{R}^{n_q}$ with covariance matrix $Q_{k-1} \triangleq \operatorname{diag}(Q_{1,k-1},\, Q_{2,k-1}) \in \mathbb{R}^{n_q\times n_q}$, where $q_{1,k-1} \in$

$\mathbb{R}^{n_q-n_x}$ is the non-additive noise related to the state vector and $q_{2,k-1} \in \mathbb{R}^{n_x}$ is the additive partition of noise. To improve the numerical stability of the filter [Xiong et al., 2006], "small" additive noise is considered for all states.

Thus, the UKF *forecast* step, usually named *prediction* step in navigation literature, is given by

$$\left(\hat{x}_{k|k-1}, \tilde{P}^{xx}_{k|k-1}, \emptyset\right) = \mathrm{UT}\left(\hat{\breve{x}}_{k-1}, P^{\breve{x}\breve{x}}_{k-1}, u_{k-1}, f\right), \tag{3.17}$$

$$P^{xx}_{k|k-1} = \tilde{P}^{xx}_{k|k-1} + Q_{2,k-1}. \tag{3.18}$$

The augmented state vector $\hat{\breve{x}}_{k-1} \in \mathbb{R}^{\breve{n}}$ and the corresponding covariance matrix $P^{\breve{x}\breve{x}}_{k-1} \in \mathbb{R}^{\breve{n}\times\breve{n}}$ are respectively given by

$$\hat{\breve{x}}_{k-1} \triangleq \begin{bmatrix} \hat{x}_{k-1} \\ \hat{q}_{1,k-1} \end{bmatrix},$$

$$P^{\breve{x}\breve{x}}_{k-1} \triangleq \begin{bmatrix} P^{xx}_{k-1} & [0]_{n_x\times(n_q-n_x)} \\ [0]_{(n_q-n_x)\times(n_x)} & Q_{1,k-1} \end{bmatrix},$$

with $\breve{n} = n_x + n_{q_1}$.

The state estimate and error covariance matrix is updated at time k using information from y_k in the *data-assimilation* step, also called *update* step, according to

$$\left(\hat{y}_{k|k-1}, \tilde{P}^{yy}_{k|k-1}, P^{xy}_{k|k-1}\right) = \mathrm{UT}\left(\hat{x}_{k|k-1}, P^{xx}_{k|k-1}, \emptyset, h\right), \tag{3.19}$$

$$P^{yy}_{k|k-1} = \tilde{P}^{yy}_{k|k-1} + R_k, \tag{3.20}$$

$$\nu_k = y_k - \hat{y}_{k|k-1}, \tag{3.21}$$

$$K_k = P^{xy}_{k|k-1}\left(P^{yy}_{k|k-1}\right)^{-1}, \tag{3.22}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k\nu_k, \tag{3.23}$$

$$P^{xx}_k = P^{xx}_{k|k-1} - K_k P^{yy}_{k|k-1} K^T_k, \tag{3.24}$$

where $\nu_k \in \mathbb{R}^{n_y}$ is the innovation, $P^{yy}_{k|k-1} \in \mathbb{R}^{n_y\times n_y}$ is the covariance of innovation, and $K_k \in \mathbb{R}^{n_x\times n_y}$ is the Kalman gain. It is important to observe that, in actual applications the *forecast* and *data-assimilation* steps do not happen in the same frequency, Thus, equations (3.19)-(3.24) need to be computed only when a new measurement is available.

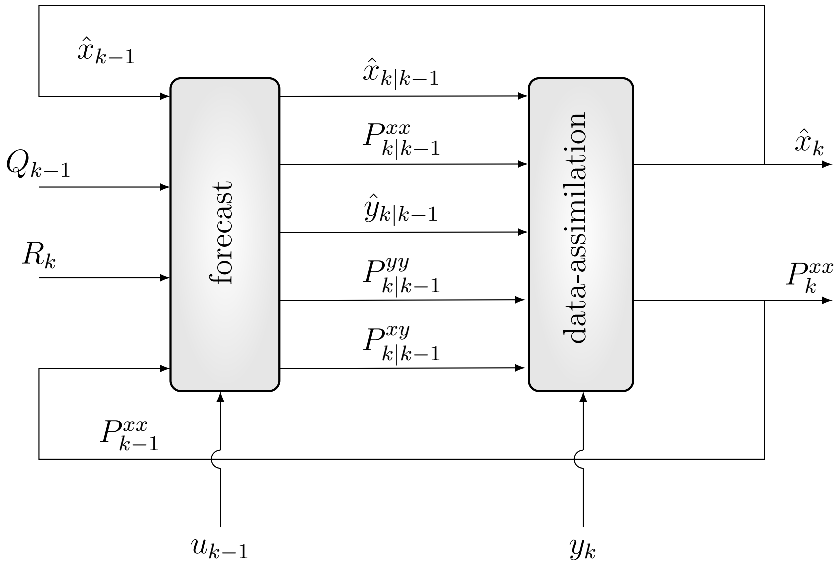


FIGURE 3.2: Block diagram of the UKF. The filtering algorithm is performed in two steps, *forecast* and *data-assimilation* steps. In the first step, the mean $\hat{x}_{k|k-1}$ and covariance $P^{xx}_{k|k-1}$ are computed based on previous estimates $\hat{x}_{k-1}$ and P^{xx}_{k-1}. Then, based on measurements y_k, the estimates are updated.

3.3 Adaptive Filtering Approaches

The original formulation of the Kalman filter, see appendix A.2, and its nonlinear approximations, such as the UKF presented in subsection 3.2.2, assume complete *a priori* knowledge of the process and measurement noise statistics, namely mean and covariance. However, in practical scenarios, the stochastic properties of the system vary depending on factors like environmental and operational conditions. In such cases, the process Q_k and observation R_k covariance matrices, which reflect the uncertainties of the system, might vary too. In this way, to improve the performance of the sensor fusion algorithms, a possible approach is to modify online the parameters related to the stochastic inputs, which is known as adaptive filtering.

In this section, the covariance matching (CM) and the interacting multiple models (IMM) approaches are reviewed. The CM approaches use samples of the innovation sequence ν_k to empirically estimate the covariance matrices. In contrast, IMM approaches employ a bank of filters running in parallel, where each filter can model a specific environmental condition or a disturbance. Then, an estimate is computed by combining the result of each filter.

3.3.1 Covariance-Matching Approach

Based on the assumption that the observation covariance matrix R_k is constant during a sliding sampling window with finite length N, the basic idea of CM is to make the innovation ν_k consistent with its covariance $P^{yy}_{k|k-1} \triangleq E[\nu_k \nu_k^T]$, computed by the filtering algorithm. Thus, if the filter-based estimated innovation covariance is significantly larger, or smaller, than the sample-based innovation covariance, the estimate of noise covariance is increased or decreased, respectively.

The algorithm presented in [Mehra, 1972] considers that the actual covariance of ν_k can be approximated by its sample covariance of the sequence of the N last innovation values, that is

$$P^{yy}_{k|k-1} \approx \frac{1}{N} \sum_{j=k-N+1}^{k} \nu_j \nu_j^T. \tag{3.25}$$

Recall that, by definition, the UKF presented in section 3.2.2, approximates the innovation covariance by

$$P^{yy}_{k|k-1} \triangleq \tilde{P}^{yy}_{k|k-1} + R_k, \tag{3.26}$$

where $\tilde{P}^{yy}_{k|k-1}$ is given by

$$\tilde{P}^{yy}_{k|k-1} \triangleq \sum_{j=1}^{2n} w_j \left(\mathcal{Y}_{j,k|k-1} - \hat{y}_{k|k-1}\right) \left(\mathcal{Y}_{j,k|k-1} - \hat{y}_{k|k-1}\right)^T. \tag{3.27}$$

Hence, an estimator for R_k is given by

$$\hat{R}_k = \frac{1}{N} \sum_{j=k-N+1}^{k} \nu_j \nu_j^T - \tilde{P}^{yy}_{k|k-1}. \tag{3.28}$$

It is important to point out that the choice of the window size N has direct influence on the estimation of R_k. In general, the tuning of N must take into account the system dynamics. If R_k is not expected to change fast over time, then a larger window size N could be chosen. Otherwise, the value of N should be tuned smaller.

3.3.2 Interacting Multiple Model

The interacting multiple model (IMM) algorithm [Blom and Bar-Shalom, 1988] assumes that the system is modeled from a set of M candidate switching models denoted by $\mathcal{M}$. As in [Li and Bar-Shalom, 1994], we assume that the observation model is described by a set of models, where each model has a different noise level characterized by R_k^i, $\forall i \in \mathcal{M}$.

Figure 3.3 illustrates the IMM algorithm. A mathematical derivation of this filtering algorithm can be seen in appendix A.4. At each iteration, for $i, j = 1...M$, the mixing probabilities $w_{k-1}^{i|j}$, are used to mix the *prior* state estimates $\hat{x}_{k-1}^i$ and the *prior* covariance matrices $P_{k-1}^{xx,i}$. The mixed states $\hat{x}_{k-1}^{0,j}$ and covariance $P_{k-1}^{xx,0,j}$ are used as inputs for a filter bank, where each filter matches one of the M candidate models. The output of each filter is the model-conditioned state estimate $\hat{x}_k^j$, the associated covariance $P_k^{xx,j}$, the model-conditioned innovation ν_k^j and its covariance $P_{k|k-1}^{yy,j}$. The model-conditioned innovation and its covariance are used to compute the likelihood function $\Lambda_k^j = \rho_j(\nu_k^j)$ of each model, which in turn, is used to update the model probabilities $\gamma_{k|k-1}^j$. Finally, the updated model probabilities are used to combine the model-conditioned estimates and covariances.

The IMM algorithm starts by initializing the filter bank with the initial estimates $\hat{x}_0^i$, the covariance matrix $P_0^{xx,i}$ and the model probabilities γ_0^i, $\forall\, i \in \mathcal{M}$. In practice, there is no prior knowledge about the actual operating mode, thus the model probabilities γ_0^i are set with the same probability.

The IMM algorithm can be performed in five steps. The first step is the computation of the *mixing probabilities*, that are given by

$$w_{k-1}^{i|j} = \frac{1}{\bar{c}_j} \Pi_{ij} \gamma_{k-1}^i, \tag{3.29}$$

$\forall\, i, j \in \mathcal{M}$, where

$$\bar{c}_j = \sum_{i=1}^{M} \Pi_{ij} \gamma_{k-1}^i \tag{3.30}$$

is a normalization factor, and $\Pi \in \mathbb{R}^{M \times M}$ is Markov model transition probabilities. In the literature, the diagonal values of Π usually vary from 0.9 to 0.98. However, as commented in [Bar-Shalom et al., 2004, p. 471], the algorithm does not seem to be very sensitive to

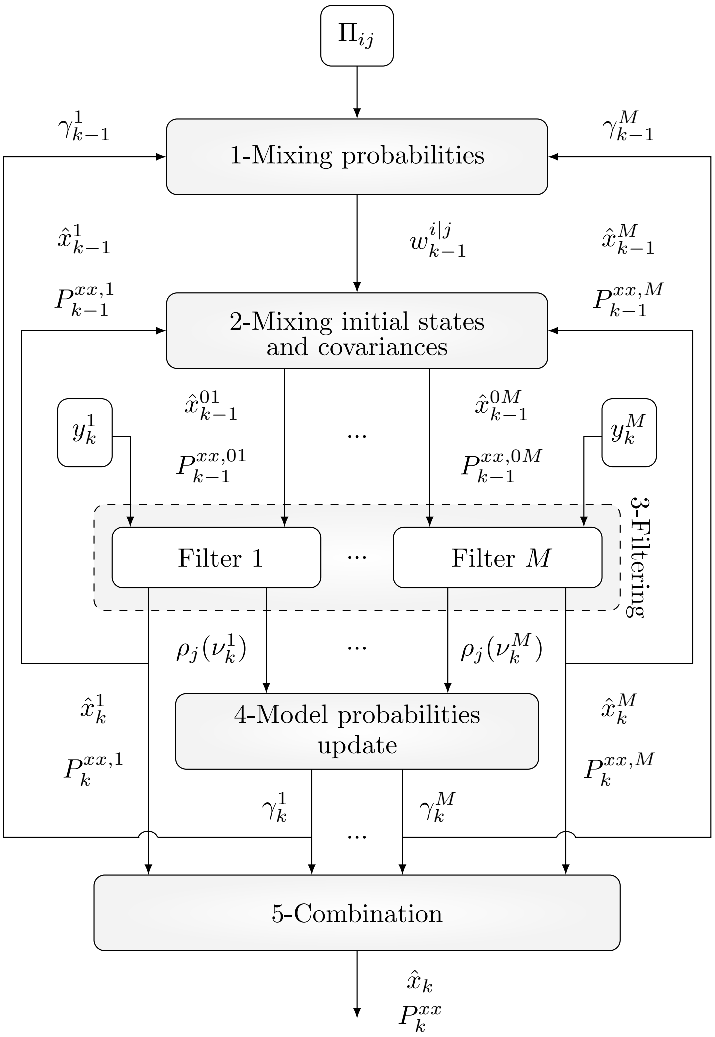


FIGURE 3.3: Block diagram of the IMM algorithm. The IMM estimator consist of five steps. In the first step, the mixing probabilities $w^{i|j}_{k-1}$ are computed; in the second step, the *prior* state estimates $\hat{x}^i_{k-1}$ and covariance matrix $P^{xx,i}_{k-1}$ are mixed; in the third step, the mixed states $\hat{x}^{0,j}_{k-1}$ and covariances $P^{xx,0,j}_{k-1}$ are used as inputs to the model-conditioned filter; in the fourth step, the probability of each model $\gamma^j_{k|k-1}$ is updated; in the last step, called *combination*, the state estimates $\hat{x}^j_k$ and covariances $P^{xx,j}_k$ are combined (adapted from [Eras-Herrera et al., 2019]).

the tuning of such parameter.

In the second step, the inputs $\hat{x}^{0j}_{k-1}$ and $P^{xx,0j}_{k-1}$, for each filter, are computed by mixing the state estimates produced by all filters from a previous time step as

$$\hat{x}^{0j}_{k-1} = \sum_{i=1}^{M} w^{i|j}_{k-1} \hat{x}^i_k \tag{3.31}$$

$$P^{xx,0j}_{k-1} = \sum_{i=1}^{M} w^{i|j}_{k-1} \left\{ P^{xx,i}_{k-1} + \left(\hat{x}^i_{k-1} - \hat{x}^{0j}_{k-1}\right)\left(\hat{x}^i_{k-1} - \hat{x}^{0j}_{k-1}\right)^T \right\}, \tag{3.32}$$

$\forall\, i, j \in \mathcal{M}$, where $\hat{x}^i_{k-1}$ and $P^{xx,i}_{k-1}$ are the updated mean and covariance for model i at time step $k-1$. In the third step, the estimates $\hat{x}^{0j}_{k-1}$ and covariances $P^{xx,0j}_{k-1}$ are used as inputs to the filter matched to model M^j_k. Therefore, the UKF presented in Section 3.2 is used for each model, forming a bank of filters [Teixeira et al., 2014]. In addition, the likelihood associated to each one of the M filters is computed as:

$$\rho_j(\nu^j_k) = C_j \exp\left(-\frac{1}{2}(\nu^j_k)^T (P^{yy,j}_{k|k-1})^{-1} \nu^j_k\right), \; j = 1...M, \tag{3.33}$$

where

$$C_j = \frac{1}{\sqrt{(2\pi)^{n_y} \det(P^{yy,j}_{k|k-1})}},$$

n_y is the measurement vector dimension, and ν^j_k is the innovation of jth model with covariance $P^{yy,j}_{k|k-1}$. Then, the probability of each model γ^j_k at time step k is computed as

$$\gamma^j_k = \frac{1}{c}\rho_j(\nu^j_k)\bar{c}_j, \tag{3.34}$$

$$c = \sum_{j=1}^{M} \rho_j(\nu^j_k)\bar{c}_j, \tag{3.35}$$

where c is a normalizing factor and $\bar{c}_j$ is given by (3.30). The last step of the algorithm is to combine the estimates and covariances, which are computed as

$$\hat{x}_k = \sum_{j=1}^{M} \gamma^j_k \hat{x}^j_k, \tag{3.36}$$

$$P^{xx}_k = \sum_{j=1}^{M} \gamma^j_k \left\{ P^{xx,j}_k + \left(\hat{x}^j_k - \hat{x}_k\right)\left(\hat{x}^j_k - \hat{x}_k\right)^T \right\}. \tag{3.37}$$

3.4 Outlier Filtering

The UKF-based approaches do not perform adequately when the error statistics follow a non-Gaussian distribution, particularly those with thicker tails and high probability of large errors compared with the Gaussian distribution. In this sense, filtering approaches that are insensitive to deviations in the assumed underlying probability distribution from which the data is sampled, are called *robust*. Formal definitions of robust estimation can be found in [Hampel, 1971]. In this work, these errors are treated as outliers and are assumed to have low probability to occur.

Outliers are spurious data that contaminate the statistical distribution, leading to erroneous analysis. The contaminated measurements deviate significantly from the normal observations, which directly reflect in the innovation value ν_k, and, consequently, in the covariance estimated by CM. In contrast, IMM approach can handle non-Gaussian distribution by using the Gaussian sum methodology [Bar-Shalom et al., 2004].

We present now two *ad-hoc* techniques for dealing with observation data that do not follow the assumed Gaussian distribution. These techniques are based on the chi-squared test and the Hampel identifier. Instead of completely discarding a possible outlier, the innovation data is processed by a statistical procedure, and the influence of the contaminated measurement is mitigated by some scale factor. Although simple, both techniques can be effective to handle outliers.

3.4.1 Outlier Filtering Based on the Chi-Squared Test

To minimize the influence of outliers, a particular method is to judge each element of the innovation [Gustafsson, 2008] with a χ^2-test. Thus, for the ith element of the innovation vector, at a sample time j, the *normalized innovation squared* can be computed as

$$\epsilon^{\nu}_{j,i} = \frac{\nu^2_{j,i}}{P^{yy}_{j-1|j-2,ii}}. \tag{3.38}$$

Notice that it is used the covariance at sample time $j-1|j-2$ instead of $j|j-1$. In this way, an abnormal measurement does not influence the estimation of the covariance matrix at time j. Under the linear Gaussian assumption, the PDF of $\epsilon^{\nu}_{j,i} \sim \chi^2_1$ is a chi-square distribution with one degree of freedom. Then, for a significance level $\alpha \in [0,1]$, the probability of a "normal" measurement is $\rho\left(\epsilon^{\nu}_{j,i} \leq \zeta\right) = 1 - \alpha$, where $\zeta \in \mathbb{R}^+$ is a value taken from the chi-square cumulative distribution function from the choice of α. If the threshold ζ is exceed, then the innovation $\nu_{j,i}$ is scaled down according to

$$\hat{\nu}_j = \lambda_j \nu_j. \tag{3.39}$$

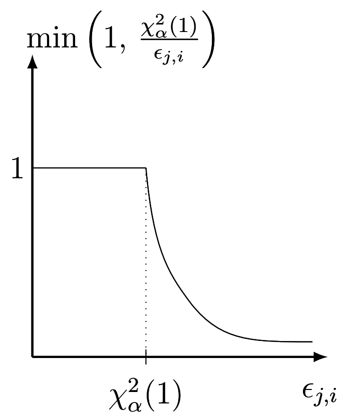


FIGURE 3.4: Graphical illustration of the weight factor behavior for one dimension.

The weight factor $\lambda_j \in \mathbb{R}^{n_r \times n_r}$ is a diagonal matrix, being each element of main diagonal computed as

$$\lambda_{j,ii} = \min\left(1, \frac{\zeta}{\epsilon^{\nu}_{j,i}}\right) . \tag{3.40}$$

Figure 3.4 illustrates the behavior of the innovation weight factor, in which for $\epsilon^{\nu}_{j,i} \leq \zeta$ the function returns unity value.

3.4.2 Outlier Filtering Based on Hampel Identifier

The Hampel identifier [Davies and Gather, 1993] is an outlier identification method reported as extremely effective in practice [Pearson, 2002]. This method is a variation of the three-sigma rule of statistics, in which the mean and standard deviation are replaced by the median and the median absolute deviation (MAD).

Based on Hampel identifier, and following the same approach of Section 3.4.1, the gain $\lambda_j \in \mathbb{R}^{n_r \times n_r}$ is defined as

$$\lambda_{j,ii} \triangleq \min\left(1, \frac{n_\sigma s_i}{|\nu_{j,i} - \mathrm{med}\{\nu_{j,i}\}_N|}\right) , \tag{3.41}$$

where $s_i = 1.4826\,\mathrm{med}\{|\nu_{j,i} - \mathrm{med}\{\nu_{j,i}\}_N|\}$ is the scaled MAD, n_σ is the number of standard deviations (confidence region) by which the innovation sample must differ from the local median, med is the median operator, $\{\cdot\}_N$ is a moving window with size N,

$j \triangleq k - N + 1 \ldots k$ is an index for each element of the moving window, and i is the index of each element of the innovation vector. Observe that gain λ_j is calculated for each element of the moving window based on the current moving window. As before, the weight factor mitigates the outlier influence by filtering the innovation, using (3.39).

3.5 Unscented Rauch-Tung-Stribel Smoother

So far we have only considered filtering algorithms. However, whenever all data are available offline, it may be of interest to estimate the states conditioned to all obtained measurements, thus solving the problem of trajectory reconstruction. Next, we present the unscented Rauch-Tung-Stribel smoother (URTSS) [Särkkä, 2008].

The smoothing algorithm has two parts. First, a filtering algorithm (UKF of Section 3.2.2, for instance) is run, yielding the filtering estimates $\hat{x}_k$ and P_k^{xx} for the time interval $k = k_0 \ldots k_f$, being k_0 and k_f the first and last time sample, respectively. Next, the following algorithm runs backwards from $k_f - 1$ to k_0. This is the second part, whose *forecast* step is given by

$$\left(\hat{x}_{k+1|k}, \tilde{P}^{xx}_{k+1|k}, P^{xy}_{k+1|k}\right) = \mathrm{UT}\left(\hat{\breve{x}}_k, P_k^{\breve{x}\breve{x}}, u_k, f\right), \tag{3.42}$$

$$P^{xx}_{k+1|k} = \tilde{P}^{xx}_{k+1|k} + Q_{2,k}, \tag{3.43}$$

where the augmented state vector $\hat{\breve{x}}_k \in \mathbb{R}^{\breve{n}}$ and the corresponding covariance matrix $P_k^{\breve{x}\breve{x}} \in \mathbb{R}^{\breve{n}\times\breve{n}}$ are respectively given by

$$\hat{\breve{x}}_k \triangleq \begin{bmatrix} \hat{x}_k \\ \hat{q}_{1,k-1} \end{bmatrix},$$

$$P_k^{\breve{x}\breve{x}} \triangleq \begin{bmatrix} P_k^{xx} & [0]_{n_x \times (n_q - n_x)} \\ [0]_{(n_q - n_x)\times n_x} & Q_{1,k} \end{bmatrix},$$

with $\breve{n} = n_x + n_{q_1}$, and whose *smoothing* step is given by

$$\mu_k = \hat{x}^s_{k+1} - \hat{x}_{k+1|k}, \tag{3.44}$$

$$K^s_k = P^{xy}_{k+1|k} \left(P^{xx}_{k+1|k}\right)^{-1}, \tag{3.45}$$

$$\hat{x}^s_k = \hat{x}_k + K^s_k \mu_k, \tag{3.46}$$

$$P^{xx,s}_k = P^{xx}_k + K^s_k \left(P^{xx,s}_{k+1} - P^{xx}_{k+1|k}\right) \left(K^s_k\right)^T, \tag{3.47}$$

with $x^s_{k_f} = \hat{x}_{k_f}$ and $P^{xx,s}_{k_f} = P^{xx}_{k_f}$.

Observe that the terms $\hat{x}_k$ and P^{xx}_k are the UKF mean and covariance estimates. It would also possible to store the values of $\hat{x}_{k+1|k}$, $P^{xx}_{k+1|k}$, and K^s_k during the filtering process, avoiding recomputing them. Figure 3.5 illustrates the URTS smoother algorithm.

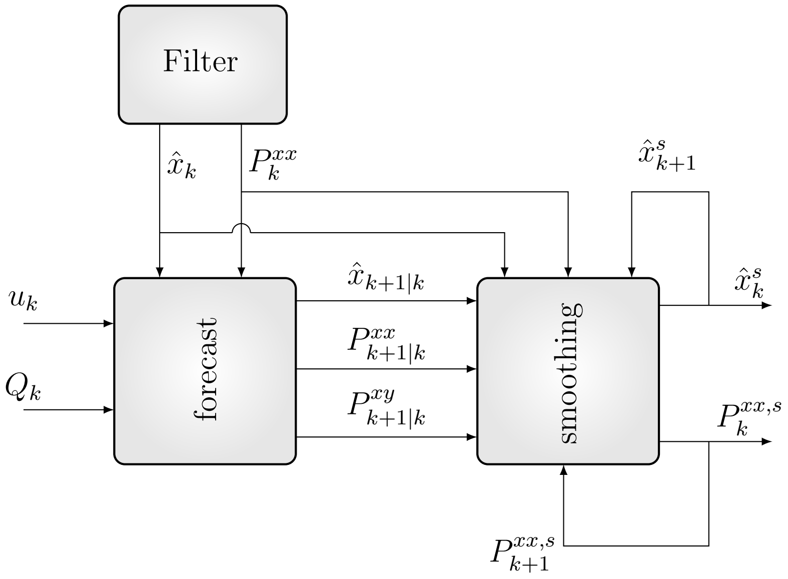


FIGURE 3.5: Block diagram of the URTS smoother. The somoothing algorithm runs backward in time, from k_f to k_0, and is performed in two steps, named *forecast* and *smoothing*. In the first step, the mean $\hat{x}_{k+1|k}$ and covariance $P^{xx}_{k+1|k}$ are computed based on filter estimates $\hat{x}_k$ and P^{xx}_k. Then, the smoothed estimates $\hat{x}^s_k$ and covariance $P^{xx,s}_k$ are computed in the smoothing steps. The smoothed states its covariance are initialized as $\hat{x}^s_{k_f} = \hat{x}_{k_f}$ and $P^{xx,s}_{k_f} = P^{xx}_{k_f}$, respectively.

3.6 Robust Adaptive Unscented Kalman Filters: New Algorithms

As the first contribution of this thesis we propose new robust adaptive estimation algorithms. By combining the adaptive approaches in Section 3.3.1 and the outlier filtering approaches presented in Section 3.4, we propose two new filtering approaches: i) robust adaptive unscented Kalman filter by using covariance matching and χ^2-test (RAUKF-CMχ^2); and ii) robust adaptive unscented Kalman filter by using covariance matching and Hampel identifier (RAUKF-CMH).

The new filtering algorithms basically differ due to the outlier filtering approaches. In this way, for the sake of conciseness, we present here only the equations of RAUKF-CMl, where $l \in \{\chi^2, \text{H}\}$. Figure 3.6 illustrates the RAUKF-CMl algorithm.

The RAUKF-CMl *forecast* step is given by equations (3.17)-(3.18). The second step, called here *robust noise estimation*, is given by equations (3.19), (3.21), and

$$\hat{R}_k = \max\left(\frac{1}{N}\sum_{j=k-N+1}^{k} \lambda_j \nu_j \left(\lambda_j \nu_j\right)^T - \tilde{P}^{yy}_{k|k-1}, R_0\right), \tag{3.48}$$

$$P^{yy}_{k|k-1} = \tilde{P}^{yy}_{k|k-1} + \hat{R}_k, \tag{3.49}$$

where R_0 is a lower threshold, given by the nominal measurement-noise covariance. The *data-assimilation* step is given by (3.22), (3.24), and

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \lambda_k \nu_k, \tag{3.50}$$

where the gain λ_k is given by (3.40), if $l = \chi^2$, or (3.41), if $l = H$.

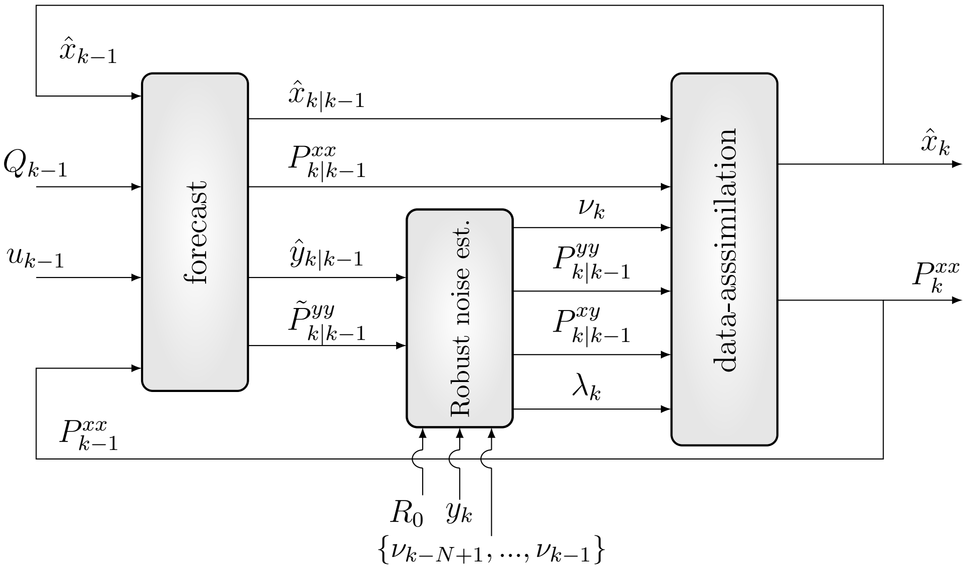


FIGURE 3.6: Block diagram of the RAUKF-CMl. The filtering algorithm is performed in three steps, called *forecast*, *robust noise estimation*, and *data-assimilation*. In the first step, the mean $\hat{x}_{k|k-1}$ and covariance $P^{xx}_{k|k-1}$ are computed based on previous estimates $\hat{x}_{k-1}$ and P^{xx}_{k-1}. Then, the output measurement noise covariance matrix $\hat{R}_k$ is robust estimated based on sliding sampling window $\{\nu_{k-N+1}, ..., \nu_{k-1}\}$ and measurement y_k. In the third step, the state estimates are updated.

3.7 Chapter Summary

In this chapter, we review nonlinear extensions of the Kalman state estimators, called unscented Kalman filter and unscented Rauch-Tung-Stribel smoother. We also review two adaptive approaches, namely covariance matching and interacting multiple model approach. To minimize the effect of spurious data that contaminate the statistical distribution, we introduce two gain factors that identify measurement gross errors and mitigate their effects.

Based on covariance matching and innovation filtering approaches, two new robust adaptive algorithms were proposed. The new algorithms are able to adapt the measurement covariance matrix and also mitigate the effect of measurement outliers online.

Chapter 4

Quaternion-based Kalman Filtering

"Simplicity is the ultimate sophistication."

Leonardo Da Vinci's

As previously commented, the localization problem consists in estimating the vehicle position and attitude. Generally, attitude is parameterized as Euler angles or unit quaternions. Although intuitive, Euler angles exhibit singularities in the kinematic description, which is known as gimbal lock [Hanson, 2006]. Due to the absence of singularities, the unit quaternion parameterization is preferred and discussed in this work.

We have presented in the last chapter state estimation algorithms for Euclidean state spaces, which means that the estimate pertains to the Euclidean vector space. The unit quaternion representation pertains to a Riemannian space. Then, the usual vector operations, such as weighted sum operations, common to most sensor fusion methodologies, may violate the unit norm constraint of the quaternion [Menegaz, 2016].

In this chapter, the Euclidean state estimators presented in the last chapter are revised in order to handle properly unit quaternion states. First, the unit quaternion algebra and some concepts of random unit quaternions are revisited in Sections 4.1 and 4.1.4, respectively. In Section 4.2, we review a modified version of unscented transform for unit quaternion, the bases of the quaternion-based unscented Kalman filter, presented in the same section. Based on quaternion operations defined in this chapter, new extensions to four adaptive algorithms, reviewed in Section 4.3.1, are proposed. In addition, we also propose new extensions to IMM and URTSS algorithms for the unit quaternion representation, in Sections 4.3.2 and 4.3.3, respectively.

4.1 Quaternion Operations

Quaternions are hypercomplex numbers developed by Sir William Rowan Hamilton (1805–1865) [Hanson, 2006], in an attempt to generalize the concept of complex numbers from 2D to 3D space. Quaternions, represented by the set $\mathbb{H}$, form a 4-dimensional algebra over the real numbers and can be used to parameterize the rotation group SO(3).

As a rotation, quaternions are constrained to unit norm, being called unit quaternions, which set is denoted by $\mathbb{H}_1$. The unit quaternion group is topologically a 3-sphere, denoted by $\mathcal{S}^3$, embedded in the $\mathbb{R}^4$. This group is closed under multiplication, but not under sum and weighting operations [Menegaz and Ishihara, 2018].

Next, we review some fundamentals of unit quaternion algebra and also construct some basic tools for its statistic analysis. The following results are important in the construction of UKF for unit quaternions and for adaptive approaches based on the sample mean and covariance.

4.1.1 Fundamentals of Unit Quaternions Algebra

The unit quaternion $e \in \mathbb{H}_1$ can be represented as

$$e = e_0 + [i \;\; j \;\; k] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}, \tag{4.1}$$

or in compact form as,

$$e = (v, \; n), \tag{4.2}$$

where $v = e_0 \in \mathbb{R}$ and $n = [e_1 \;\; e_2 \;\; e_3]^T \in \mathbb{R}^3$ are the real and imaginary parts, in analogy with complex numbers. It is also common to represent quaternions in a 4-dimensional vector, which is performed by the vec$(\cdot)$ operator defined as

$$\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} \triangleq \text{vec}(e), \tag{4.3}$$

which can also be represented in a compact form by $\text{vec}(e) = [v \;\; n]^T \in \mathbb{R}^4$.

The unit quaternion inverse operation is equal to its conjugate, given by $e^{-1} = e^* = (v, \; -n)$. The product $\otimes$ between quaternions, preserves membership in $\mathcal{S}^3$, being defined as

$$e_a \otimes e_b \triangleq \left(v_a v_b - n_a^T n_b, \; v_a n_b + v_b n_a + n_a \times n_b \right), \tag{4.4}$$

where $e_a = (v_a, n_a) \in \mathbb{H}_1$, $e_b = (v_b, n_b) \in \mathbb{H}_1$, and $\times$ denotes the cross-product. We can observe that, due to the cross-product, the multiplication operation is not commutative, in general.

For e and $e^* \in \mathbb{H}_1$,

$$e \otimes e^* = e^* \otimes e = \left(1, [0]_{3\times1}\right), \tag{4.5}$$

which is the identity element [Hanson, 2006].

A vector $x_a \in \mathbb{R}^3$ can be rotated by a unit quaternion e, as

$$(0, \; x_b) = e \otimes (0, \; x_a) \otimes e^*, \tag{4.6}$$

where $x_b \in \mathbb{R}^3$ is the rotated vector [Hanson, 2006].

4.1.2 Euclidean Tangent Space and Rotation Vector Parameterization

The group $\mathcal{S}^3$ is a Riemannian manifold, whose elements can be mapped to the 3-dimensional Euclidean *tangent space* $\text{T}_e\mathcal{S}^3$. Many operations are defined in the Euclidean tangent space, such as the sample mean and covariance. These basic statistics to the Riemannian manifold are the bases of the filtering algorithms developed here. Furthermore, there are direct and inverse mappings between the manifold and its *tangent space*, $\mathcal{S}^3 \longleftrightarrow \mathrm{T}_e\mathcal{S}^3$, with *exponential* and *logarithm* functions, respectively. This behavior can also be seen as the *exponential map* between a Lie-Group and its Lie-Algebra [Sipos, 2008]. More precisely, for $e = (v, n) \in \mathbb{H}_1$ and $r \in \mathbb{R}^3$ the dual relations are

$$e = \exp\left((0, r)\right) \Longleftrightarrow (0, r) = \log\left(e\right). \tag{4.7}$$

Given a rotation $\theta \in \mathbb{R}$ and the unit vector $w \in \mathbb{R}^3$, the corresponding unit quaternion is $e = \left(\cos\left(\frac{\theta}{2}\right), \, \sin\left(\frac{\theta}{2}\right) w\right)$. Defined in this way, the unit quaternion covers the SO(3) twice, which means that $e = -e$, called antipodal equivalence [Sipos, 2008]. Now, let $r = \theta w$ be a *rotation vector* representing a rotation θ about the unit axis w. The unit quaternion to *rotation vector* mapping, called *logarithm* mapping, is given by [Chirikjian and Kyatkin, 2001]:

$$r = \begin{cases} 2\arccos\left(v\right)\frac{n}{\|n\|}, \text{ if } \|n\| \neq 0 \text{ and } v \geq 0\,, \\ -2\arccos\left(-v\right)\frac{n}{\|n\|}, \text{ if } \|n\| \neq 0 \text{ and } v < 0\,, \\ [0]_{3\times 1}\,, \text{ if } \|n\| = 0\,, \end{cases} \tag{4.8}$$

where $v = \cos\left(\frac{\theta}{2}\right)$, $n = \sin\left(\frac{\theta}{2}\right) w$. Observe that, the quaternion antipodal ambiguity $e = -e$ is treated by checking the signal of v [Kim et al., 2017, Chiella et al., 2019c].

The inverse mapping, called *exponential* mapping, is [Kraft, 2003]:

$$e = \begin{cases} \left(\cos\left(\frac{\|r\|}{2}\right), \, \sin\left(\frac{\|r\|}{2}\right)\frac{r}{\|r\|}\right), \text{ if } \|r\| \neq 0\,, \\ \left(1, \; [0]_{3\times 1}\right), \text{ if } \|r\| = 0\,. \end{cases} \tag{4.9}$$

For brevity, *logarithm* (4.8) and *exponential* (4.9) mappings are written as $e = \mathrm{R2Q}(r)$ and $r = \mathrm{Q2R}(e)$ operations, respectively.

4.1.3 Subtraction, Sum, and Weighted Mean Operations

The difference between e_a and $e_b \in \mathbb{H}_1$ is defined as

$$e_a \ominus e_b \triangleq \mathrm{Q2R}\left(e_a \otimes e_b^*\right) . \tag{4.10}$$

The operation (4.10) is analogous in the Euclidean space, where given two vectors ξ_a and $\xi_b \in \mathbb{R}^n$, the difference between ξ_a and ξ_b is giving by $\xi_a - \xi_b$.

The sum of a unit quaternion $e_a \in \mathbb{H}_1$ and a rotation vector $r \in \mathbb{R}^3$, is defined as

$$e_a \oplus r \triangleq \mathrm{R2Q}\left(r\right) \otimes e_a . \tag{4.11}$$

Similarly, in the Euclidean space, the sum operation between ξ_a and ξ_b is giving by $\xi_a + \xi_b$.

Lastly, the weighted mean operation for a set of unit quaternions $E = \{e_i\}_{n_w}$, $i \triangleq 1 \dots n_w$, is given by

$$\hat{e} = \mathrm{WM}\left(E, W\right) , \tag{4.12}$$

where $W = \{w_i\}_{n_w}$ is a set of corresponding weights. The quaternion mean $\hat{e} \in \mathbb{H}_1$ can be computed in a closed form by using the relation

$$M \triangleq \sum_{i=1}^{n_w} w_i \mathrm{vec}(e_i) \mathrm{vec}(e_i)^T, \tag{4.13}$$

where $M \in \mathbb{R}^{4\times4}$, and the quaternion mean is the eigenvector of M corresponding to the maximum eigenvalue [Markley et al., 2007]. Iterative algorithms can also be used, as the gradient descent algorithm, proposed in [Pennec, 2006], and modified to unit quaternion in [Kraft, 2003, Sipos, 2008]. In the Euclidean space, the weighted mean is computed by $\hat{\xi} = \sum_{i=1}^{n_w} w_i \xi_i$.

Figure 4.1 illustrates graphically the operations of sum, subtraction, and weighted mean for unit quaternions. From now on, we use the $\otimes$, $\ominus$, $\oplus$, and WM operations for Riemannian and Euclidean spaces. Therefore, for Euclidean state spaces, these operations are seen as the usual multiplication, subtraction, sum, and weighted mean operations. In addition, for a vector $\xi \in \mathbb{R}^{n_\xi}$, $\mathrm{vec}(\xi) = \xi$.

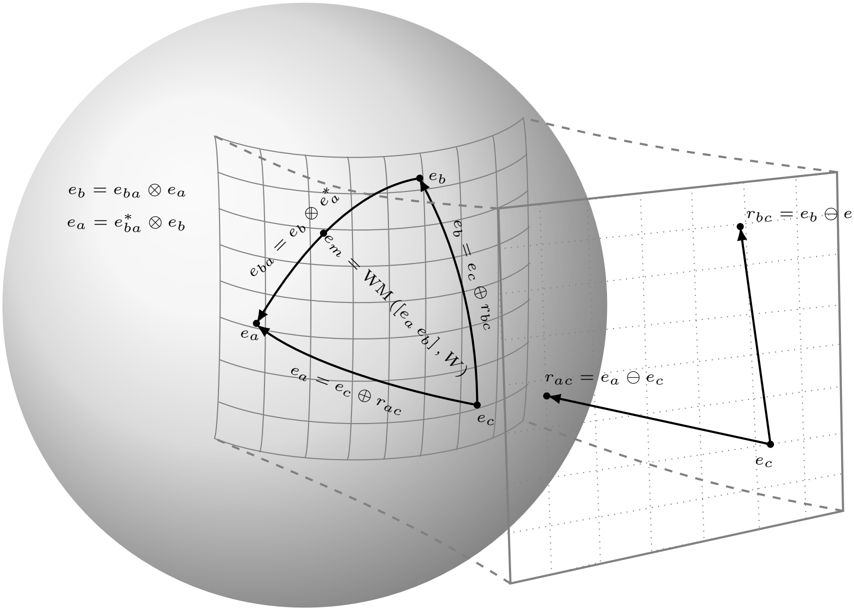

$e_b = e_{ba} \otimes e_a$
$e_a = e_{ba}^* \otimes e_b$
e_b
$e_{ba} = e_b \oplus e_a^*$
$e_m = \mathrm{WM}([e_a\ e_b], W)$
$e_b = e_c \oplus r_{bc}$
e_a
$e_a = e_c \oplus r_{ac}$
e_c
$r_{bc} = e_b \ominus e_c$
$r_{ac} = e_a \ominus e_c$
e_c

FIGURE 4.1: Example of quaternion operations on a 2-*sphere*, $\mathcal{S}^2$. The unit quaternions e_a, e_b, e_c, and e_m can be seen as points along the surface of $\mathcal{S}^2$. Observe that, e_m is a result of weighted mean operation between e_a and e_b, choosing $W = [0.5]_{2\times1}$. The subtraction operation between e_b and e_c, or, e_a and e_c, yields r_{bc} or r_{ac}, in the tangent space, respectively. The unit quaternions e_a or e_b, can be recovered by sum operation between e_c and r_{ac}, or, e_c and r_{bc}, respectively.

4.1.4 Unit Quaternion Statistics

We have seen that a rotation can be described in terms of a unit quaternion or a rotation vector, each having some advantages and disadvantages. Unit quaternions are free of singularities but have constraints, thus belongs to the group $\mathcal{S}^3$. In contrast, rotation vectors are free of constraints and can be treated as elements in a vector space, where many useful statistic operations, such as sample mean and covariance, are defined, but there are singularities.

Based on the unit quaternions tools presented in Section 4.1.1, we introduce our definitions of random variables and probability density functions for unit quaternions. The random unit quaternion is defined as

$$e \triangleq \bar{e} \oplus \delta, \tag{4.14}$$

where $\bar{e} \in \mathbb{H}_1$ is assumed be the noisy free component of our random unit quaternion, which means that it remains on $\mathcal{S}^3$, and $\delta \sim \mathcal{N}\left([0]_{3\times1}, R_{3\times3}\right) \in \mathbb{R}^3$ is the noisy component. This means that the PDF for δ will induce a PDF on $\mathcal{S}^3$. Thus,

$$\delta \triangleq e \ominus \bar{e}. \tag{4.15}$$

We can observe that δ can be seen as the difference between e and $\bar{e}$, where the *distance* is given by the magnitude of the angle of rotation, that is $\|\delta\| = \|e \ominus \bar{e}\|$. This distance can be viewed as a Riemannian metric and is preserved in the unit quaternion operations [Menegaz, 2016].

Now, let $E = \{e_i\}_{n_w}$ and $W = \{w_i\}_{n_w}$, $i \triangleq 1 \dots n_w$, be a set of random unit quaternions and weights, respectively, thus $e_i \in \mathcal{S}^3$, and $w_i \in \mathbb{R}$. The sample mean of E is defined as

$$\mathrm{E}\left[e\right] \triangleq \arg\min_{\bar{e}\in\mathcal{S}^3} \sum_{i=1}^{n_w} w_i \left(e_i \ominus \bar{e}\right)\left(e_i \ominus \bar{e}\right)^T. \tag{4.16}$$

The solution to (4.16) is given by (4.12). If there exists $\bar{e} = \mathrm{E}\left[e\right]$, then the sample covariance $P^{ee} \in \mathbb{R}^{3\times3}$ of E respective to $\bar{e}$ is defined as [Hauberg et al., 2013, Menegaz et al., 2019]

$$P^{ee} \triangleq \sum_{i=1}^{n_w} w_i \left(e_i \ominus \bar{e}\right)\left(e_i \ominus \bar{e}\right)^T. \tag{4.17}$$

4.2 Quaternion-based State Estimation

Without loss of generality, let us assume that our dynamic system is modeled by the nonlinear equations

$$x_k = f(k_{k-1}, u_{k-1}, q_{k-1}, k-1) \tag{4.18}$$

$$y_k = h(x_k, k) \oplus r_k, \tag{4.19}$$

in which, $\forall k \geq 1$, $\text{vec}\,(x_k) = [\text{vec}\,(x_{\text{H},k}) \; x_{\text{E},k}] \in \mathbb{R}^{n_x}$ is the system state vector, where $x_{\text{H},k} \in \mathbb{H}_1$ and $x_{\text{E},k} \in \mathbb{R}^{n_x-4}$, the known data are the measured output $y_k \in \mathbb{H}_1$ and input $u_{k-1} \in \mathbb{R}^{n_u}$. It is also assumed that process noise $q_{k-1} \in \mathbb{R}^{n_q}$ and measurement noise $r_k \in \mathbb{R}^{n_r}$ are mutually independent with covariance matrices $Q_{k-1} \in \mathbb{R}^{n_q \times n_q}$ and $R_k \in \mathbb{R}^{n_r \times n_r}$, respectively. The state estimation problem aims at providing approximations for the mean $\hat{x}_k = \text{E}[x_k] \in \mathbb{R}^{n_x}$ and covariance $P_k^{xx} = \text{E}[(x_k \ominus \hat{x}_k)(x_k \ominus \hat{x}_k)^T] \in \mathbb{R}^{(n_x-1)\times(n_x-1)}$ that characterize the *a posteriori* PDF $\rho(x_k|y_{1:k})$. Notice that the covariance matrix P_k^{xx} has dimension $(n_x - 1) \times (n_x - 1)$. This is because the unit quaternion is represented by a 4-dimensional vector, and the uncertainty is represented by a rotation vector, that is a 3-dimensional vector.

4.2.1 Quaternion ***Unscented*** Transform

The UT approximates the mean $\hat{z} \in \mathbb{R}^{n_z}$ and its covariance $P^{zz} \in \mathbb{R}^{(n_z-1)\times(n_z-1)}$ of a random variable z obtained from the nonlinear transformation $z = g(x_1, x_2, c)$, where $\text{vec}\,(x_1) \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ are random variables with mean $\bar{x}_1$ e $\bar{x}_2$ and covariance matrices $\bar{P}^{x_1x_1} \in \mathbb{R}^{(n_1-1)\times(n_1-1)}$ and $\bar{P}^{x_2x_2} \in \mathbb{R}^{n_2\times n_2}$, respectively, and c a known deterministic variable. In addition, assumes that the random variable x_1 is composed by a unit quaternion part $x_{1,\text{H}}$ and a unconstrained Euclidean part $x_{1,\text{E}}$; thus $\text{vec}\,(x_1) \triangleq \begin{bmatrix}\text{vec}\,(x_{1,\text{H}})^T & x_{1,\text{E}}^T\end{bmatrix}^T$.

Now, we define the augmented state vector $\text{vec}\,(\bar{x}) \in \mathbb{R}^n$ as

$$\bar{x} \triangleq \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \tag{4.20}$$

where $n = n_1 + n_2$, as well as the augmented covariance matrix $\bar{P}^{xx} \in \mathbb{R}^{(n-1)\times(n-1)}$

$$\bar{P}^{xx} = \begin{bmatrix} \bar{P}^{x_1x_1} & [0]_{(n_1-1)\times n_2} \\ [0]_{n_2\times(n_1-1)} & \bar{P}^{x_2x_2} \end{bmatrix}. \tag{4.21}$$

The sigma points $\text{vec}\,(\mathcal{X}_j) \in \mathbb{R}^n$ and the associated weights w_j, $j = 1, \ldots, 2(n-1)$ can be chosen as

$$\mathcal{X} = [\bar{x}]_{2(n-1)} \oplus \sqrt{n-1} \left[\left(\bar{P}^{xx}\right)^{\frac{1}{2}} \quad -\left(\bar{P}^{xx}\right)^{\frac{1}{2}} \right], \tag{4.22}$$

$$w_j = \frac{1}{2(n-1)}, \tag{4.23}$$

where $\mathcal{X}_j$ is the jth column of matrix $\mathrm{vec}\left(\mathcal{X}\right) \in \mathbb{R}^{n \times 2(n-1)}$, and $\mathrm{vec}\left([\bar{x}]_{2(n-1)}\right) \in \mathbb{R}^{n \times 2(n-1)}$ is a matrix whose columns are equal to $\bar{x}$. Notice that, the columns of the covariance matrix $\bar{P}^{xx}$ can be seen as a perturbation variable, where the unit quaternion part is parameterized as a *rotation vector*, which means that the covariance matrix is defined in the *tangent space*, hence the $n-1$ dimension. The SP (4.22) can be partitioned as

$$\begin{bmatrix} \mathcal{X}^{x_1} \\ \mathcal{X}^{x_2} \end{bmatrix} \triangleq \mathcal{X}, \tag{4.24}$$

where $\mathrm{vec}\left(\mathcal{X}^{x_1}\right) \in \mathbb{R}^{n_1 \times 2(n-1)}$ and $\mathcal{X}^{x_2} \in \mathbb{R}^{n_2 \times 2(n-1)}$.

Then, each sigma point $\mathcal{X}_j$ is propagated through g:

$$\mathcal{Z}_j = g\left(\mathcal{X}_j^{x_1}, \mathcal{X}_j^{x_2}, c\right), \tag{4.25}$$

where $\mathrm{vec}\left(\mathcal{Z}_j\right) = \left[\mathrm{vec}\left(\mathcal{Z}_{j,\mathrm{H}}\right)^T \quad \mathcal{Z}_{j,\mathrm{E}}^T\right]^T \in \mathbb{R}^{n_z}$ is the jth column of the matrix $\mathrm{vec}\left(\mathcal{Z}\right) \in \mathbb{R}^{n_z \times 2(n-1)}$.

From (4.25), we obtain $\hat{z}$, P^{zz} and $P^{x_1 z}$ as

$$\hat{z} = \mathrm{WM}\left(\mathcal{Z}, W\right), \tag{4.26}$$

$$P^{zz} = \sum_{j=1}^{2(n-1)} w_j \left(\mathcal{Z}_j \ominus \hat{z}\right) \left(\mathcal{Z}_j \ominus \hat{z}\right)^T, \tag{4.27}$$

$$P^{x_1 z} = \sum_{j=1}^{2(n-1)} w_j \left(\mathcal{X}_j^{x_1} \ominus x_1\right) \left(\mathcal{Z}_j \ominus \hat{z}\right)^T, \tag{4.28}$$

where $W = \{w_j\}_{2(n-1)}$.

As before, for notation simplicity, we define the quaternion unscented transform as the function $\mathrm{QUT}(\cdot)$ comprising the set of equations (4.22)-(4.28).

4.2.2 Quaternion-based UKF

The quaternion-based unscented Kalman filter (QUKF) algorithm presented in this section is based on the ones shown in [Sipos, 2008, Menegaz and Ishihara, 2018], which are slightly modified to encompass direct unit quaternion measurements and multiplicative noise in the process. Henceforth, we follow the same notation defined before, where $\hat{x}_{k|k-1}$ indicates an estimate of x_k at time k based on information available up to and including time $k-1$. Likewise, $\hat{x}_k$ indicates an estimate of x_k at time k based on information available up to and including time k. Let the process noise be partitioned as $q_{k-1} \triangleq \begin{bmatrix} q_{1,k-1}^T & q_{2,k-1}^T \end{bmatrix}^T \in \mathbb{R}^{n_q}$ with covariance matrix $Q_{k-1} \triangleq \operatorname{diag}\left(Q_{1,k-1},\ Q_{2,k-1}\right) \in \mathbb{R}^{n_q \times n_q}$, where $q_{1,k-1} \in \mathbb{R}^{n_q - n_x + 1}$ is the multiplicative noise related to the state vector and $q_{2,k-1} \in \mathbb{R}^{n_x - 1}$ is the additive noise. Notice that, the dimension of process covariance matrix is decreased by one.

Given these definitions, the modified UKF *forecast* step is given by

$$\left(\hat{x}_{k|k-1}, \tilde{P}^{xx}_{k|k-1}, \emptyset\right) = \mathrm{QUT}\left(\hat{\breve{x}}_{k-1}, P^{\breve{x}\breve{x}}_{k-1}, u_{k-1}, f\right), \tag{4.29}$$

$$P^{xx}_{k|k-1} = \tilde{P}^{xx}_{k|k-1} + Q_{2,k-1}. \tag{4.30}$$

The augmented state vector $\operatorname{vec}\left(\hat{\breve{x}}_{k-1}\right) \in \mathbb{R}^{\breve{n}}$ and the corresponding covariance matrix $P^{\breve{x}\breve{x}}_{k-1} \in \mathbb{R}^{(\breve{n}-1)\times(\breve{n}-1)}$ are respectively given by

$$\hat{\breve{x}}_{k-1} \triangleq \begin{bmatrix} \hat{x}_{k-1} \\ \hat{q}_{1,k-1} \end{bmatrix},$$

$$P^{\breve{x}\breve{x}}_{k-1} \triangleq \begin{bmatrix} P^{xx}_{k-1} & [0]_{(n_x-1)\times(n_q-n_x+1)} \\ [0]_{(n_q-n_x+1)\times(n_x-1)} & Q_{1,k-1} \end{bmatrix},$$

with $\breve{n} = n_q + 1$.

The state estimate and error covariance matrix are updated using information from

y_k in the *data-assimilation* step, given by

$$\left(\hat{y}_{k|k-1}, \tilde{P}^{yy}_{k|k-1}, P^{xy}_{k|k-1}\right) = \mathrm{QUT}\left(\hat{x}_{k|k-1}, P^{xx}_{k|k-1}, \emptyset, h\right), \tag{4.31}$$

$$P^{yy}_{k|k-1} = \tilde{P}^{yy}_{k|k-1} + R_k, \tag{4.32}$$

$$\nu_k = y_k \ominus \hat{y}_{k|k-1}, \tag{4.33}$$

$$K_k = P^{xy}_{k|k-1}\left(P^{yy}_{k|k-1}\right)^{-1}, \tag{4.34}$$

$$\hat{x}_k = \hat{x}_{k|k-1} \oplus K\nu_k, \tag{4.35}$$

$$P^{xx}_k = P^{xx}_{k|k-1} - K_k P^{yy}_{k|k-1} K_k^T, \tag{4.36}$$

where ν_k is the innovation.

4.3 Quaternion-based Robust Adaptive State Estimators: New Algorithms

In order to properly handle the time-varying measurement uncertainty and unit quaternion norm constraints, we propose now extensions to the RAUKF, IMM, and RTS smoother, presented in the Sections 3.3.2, 3.5, and 3.6, respectively.

4.3.1 Quaternion-based Robust Adaptive Unscented Kalman Filter

Based on QUKF, we extend the algorithms proposed in Section 3.6 to handle unit quaternion. The two new algorithms are: i) quaternion-based robust adaptive unscented Kalman filter by using covariance matching and χ^2-test (QRAUKF-CMχ^2); and ii) quaternion-based robust adaptive unscented Kalman filter by using covariance matching and Hampel identifier (QRAUKF-CMH). As in Section 3.6, the new filtering algorithms basically differ from outlier filtering approaches. Then we present the next QRAUKF-CMl, where $l \in \{\chi^2, \mathrm{H}\}$.

Quaternion-based Robust Adaptive Unscented Kalman by Covariance Matching - l

The QRAUKF-CMl *forecast* step is given by equations (4.29)-(4.30). The second step, called here *robust noise estimation* is given by equation (4.31), (4.33),

$$\hat{R}_k = \max\left(\frac{1}{N}\sum_{j=k-N+1}^{k} \lambda_j \nu_j \left(\lambda_j \nu_j\right)^T - \tilde{P}^{yy}_{k|k-1}, R_0\right), \tag{4.37}$$

$$P^{yy}_{k|k-1} = \tilde{P}^{yy}_{k|k-1} + \hat{R}_k, \tag{4.38}$$

where R_0 is a lower threshold, given by the nominal measurement-noise covariance. The *data-assimilation* step is given by (4.34), (4.36), and

$$\hat{x}_k = \hat{x}_{k|k-1} \oplus K_k \lambda_k \nu_k, \tag{4.39}$$

where the gain λ is given by (3.40) or (3.41).

4.3.2 Quaternion-based Interacting Multiple Model Filter

The mixing and combination steps of the classical IMM algorithm contain weighting and displacement operations, which do not preserve the unit quaternion norm. Thus, based on unit quaternion statistics, Section 4.1.4, we present now a new version of IMM that can properly handle the unit quaternion constraints, called by quaternion-based interacting multiple model filter (QIMM).

The QIMM algorithm starts by initializing the filter bank with the initial estimates $\hat{x}^i_0$, the covariance matrix $P^{xx,i}_0$ and the model probabilities γ^i_0, $\forall\, i \in \mathcal{M}$. Thus, the new IMM algorithm can be performed in five steps, as follows. The first step is the computation of the *mixing probabilities*, that are given by

$$w^{i|j}_{k-1} = \frac{1}{\bar{c}_j}\Pi_{ij}\gamma^i_{k-1}, \tag{4.40}$$

$\forall\, i,j\ \in\ \mathcal{M}$, where

$$\bar{c}_j = \sum_{i=1}^{n} \Pi_{ij}\gamma_{k-1}^{i} \tag{4.41}$$

is a normalization factor, and $\Pi \in \mathbb{R}^{M\times M}$ is Markov model transition probabilities. In the second step, the inputs $\hat{x}_{k-1}^{0j}$ and $P_{k-1}^{xx,0j}$, for each filter, are computed by mixing the state estimates produced by all filters from a previous time step as

$$\hat{x}_{k-1}^{0j} = \mathrm{WM}\left(\{\hat{x}_k^i\}_M, \{w_{k-1}^{i|j}\}_M\right), \tag{4.42}$$

$$P_{k-1}^{xx,0j} = \sum_{i=1}^{n} w_{k-1}^{i|j}\left\{P_{k-1}^{xx,i} + \left(\hat{x}_{k-1}^{i} \ominus \hat{x}_{k-1}^{0j}\right)\left(\hat{x}_{k-1}^{i} \ominus \hat{x}_{k-1}^{0j}\right)^T\right\}, \tag{4.43}$$

$\forall\, i,j\ \in\ \mathcal{M}$, where $\hat{x}_{k-1}^{i}$ and $P_{k-1}^{xx,i}$ are the updated mean and covariance for model i at time step $k-1$. In the third step, the estimates $\hat{x}_{k-1}^{0j}$ and covariances $P_{k-1}^{xx,0j}$ are used as inputs to the filter matched to model M_k^j. Therefore, the UKF presented in Section 4.3.1 is used for each model, forming a bank of filters. In addition, the likelihood associated to each one of the M filters is computed as

$$\rho_j(\nu_k^j) = C_j \exp\left(-\frac{1}{2}(\nu_k^j)^T(P_{k|k-1}^{yy,j})^{-1}\nu_k^j\right), \tag{4.44}$$

$j = 1 \ldots M$, where

$$C_j = \frac{1}{\sqrt{(2\pi)^{(n_y-1)}\det(P_{k|k-1}^{yy,j})}},$$

n_y is the measurement vector dimension, ν_k^j with covariance $P_{k|k-1}^{yy,j}$. Then, the probability of each model at time step k is computed as

$$\gamma_k^j = \frac{1}{c}\rho_j(\nu_k^j)\bar{c}_j, \tag{4.45}$$

$$c = \sum_{j=1}^{M} \rho_j(\nu_k^j)\bar{c}_j, \tag{4.46}$$

where c is a normalizing factor and $\bar{c}_j$ is given by (4.41). The last step of the algorithm is to combine the estimates and covariances, which are computed as

$$\hat{x}_k = \mathrm{WM}\left(\{\hat{x}^j_k\}_M, \{\gamma^j_k\}_M\right), \tag{4.47}$$

$$P^{xx}_k = \sum_{j=1}^{n} \gamma^j_k \left\{ P^{xx,j}_k + \left(\hat{x}^j_k \ominus \hat{x}_k\right)\left(\hat{x}^j_k \ominus \hat{x}_k\right)^T \right\}. \tag{4.48}$$

Notice that, the standard equations of IMM are modified to take into account of unit quaternion states. In this way, the mixing and combination steps are performed by the operations of subtraction (4.10) and weighted mean (4.12).

4.3.3 Quaternion-based Unscented Rauch-Tung-Stribel Smoother

We now can use the same quaternion tools to modify the URTS smoother to handle the unit norm constraints. As presented in Section 3.5, the smoothing algorithm has two parts. First, a filtering algorithm (QUKF of Section 4.2.2, for instance) is executed, yielding the filtering estimates $\hat{x}_k$ and P^{xx}_k for the time interval $k = k_0 \ldots k_f$. Next, the quaternion-based unscented Rauch-Tung-Stribel smoother (QURTS) algorithm runs backwards from $k_f - 1$ to k_0. This is the second part, whose *forecast* step is given by

$$\left(\hat{x}_{k+1|k}, \tilde{P}^{xx}_{k+1|k}, P^{xy}_{k+1|k}\right) = \mathrm{QUT}\left(\hat{\breve{x}}_k, P^{\breve{x}\breve{x}}_k, u_k, f\right), \tag{4.49}$$

$$P^{xx}_{k+1|k} = \tilde{P}^{xx}_{k+1|k} + Q_{2,k}, \tag{4.50}$$

where the augmented state vector $\mathrm{vec}\left(\hat{\breve{x}}_k\right) \in \mathbb{R}^{\breve{n}}$ and the corresponding covariance matrix $P^{\breve{x}\breve{x}}_k \in \mathbb{R}^{(\breve{n}-1)\times(\breve{n}-1)}$ are respectively given by

$$\begin{aligned}
\mu_k &= x^s_{k+1} \ominus \hat{x}_{k+1|k}, \\
\hat{\breve{x}}_k &\triangleq \begin{bmatrix} \hat{x}_k \\ \hat{q}_{1,k-1} \end{bmatrix}, \\
P^{\breve{x}\breve{x}}_k &\triangleq \begin{bmatrix} P^{xx}_k & [0]_{(n_x-1)\times(n_q-n_x+1)} \\ [0]_{(n_q-n_x+1)\times(n_x-1)} & Q_{1,k} \end{bmatrix},
\end{aligned}$$

with $\breve{n} = n_q + 1$; and whose *smoothing* step is given by

$$K_k^s = P_{k+1|k}^{xy} \left(P_{k+1|k}^{xx} \right)^{-1}, \tag{4.51}$$

$$\hat{x}_k^s = \hat{x}_k \oplus K_k^s \mu_k, \tag{4.52}$$

$$P_k^{xx,s} = P_k^{xx} + K_k^s \left(P_{k+1}^{xx,s} - P_{k+1|k}^{xx} \right) \left(K_k^s \right)^T, \tag{4.53}$$

with $x_{k_f}^s = x_{k_f}$ and $P_{k_f}^{xx,s} = P_{k_f}^{xx}$.

4.4 Chapter Summary

In this chapter, we reviewed an extension of the unscented Kalman filter for attitude represented by unit quaternion. Based on defined quaternion operations, four new state estimator algorithms for unit quaternion were proposed, namely QRAUKF-CMχ^2, QRAUKF-CMH, QIMM, and QURTS smoother, respectively. In the next chapters, experimental results with actual data are presented.

Chapter 5

Case Study: Localization of Aerobatic Airplanes Equipped with Cameras

5.1 Introduction

The Kalman filter has been widely used for localization and path reconstruction of aerial vehicles. As previously discussed in Chapter 2, localization consists in estimating the aircraft position and attitude, which is usually performed by combining data from global navigation satellite systems, inertial navigation system and compass [Leutenegger and Siegwart, 2012]. INS and compass provide high frequency information, but INS accuracy deteriorates with time due to sensor biases and noise. On the other hand, GNSS provides accurate absolute position and velocity at low frequency, which can be used to correct inertial measurement unit based information.

The technological motivation for this case study comes from aerial races and aerobatics competitions (see Figure 5.1) when, in addition to aggressive aerobatic maneuvers, the presence of strong television signals may also jam the GNSS signal. On the other hand, images from the same television system that causes such a problem could be a source of information that may help on the localization effort once, some known landmarks may be visible, since the vehicle flies at low altitude. It is worth to notice that localization is very important in such competitions since a good flight path reconstruction is worthy information for the pilots who want to improve their performance. In addition, the estimated flight path can be used in TV transmissions, improving the illustration of the race.

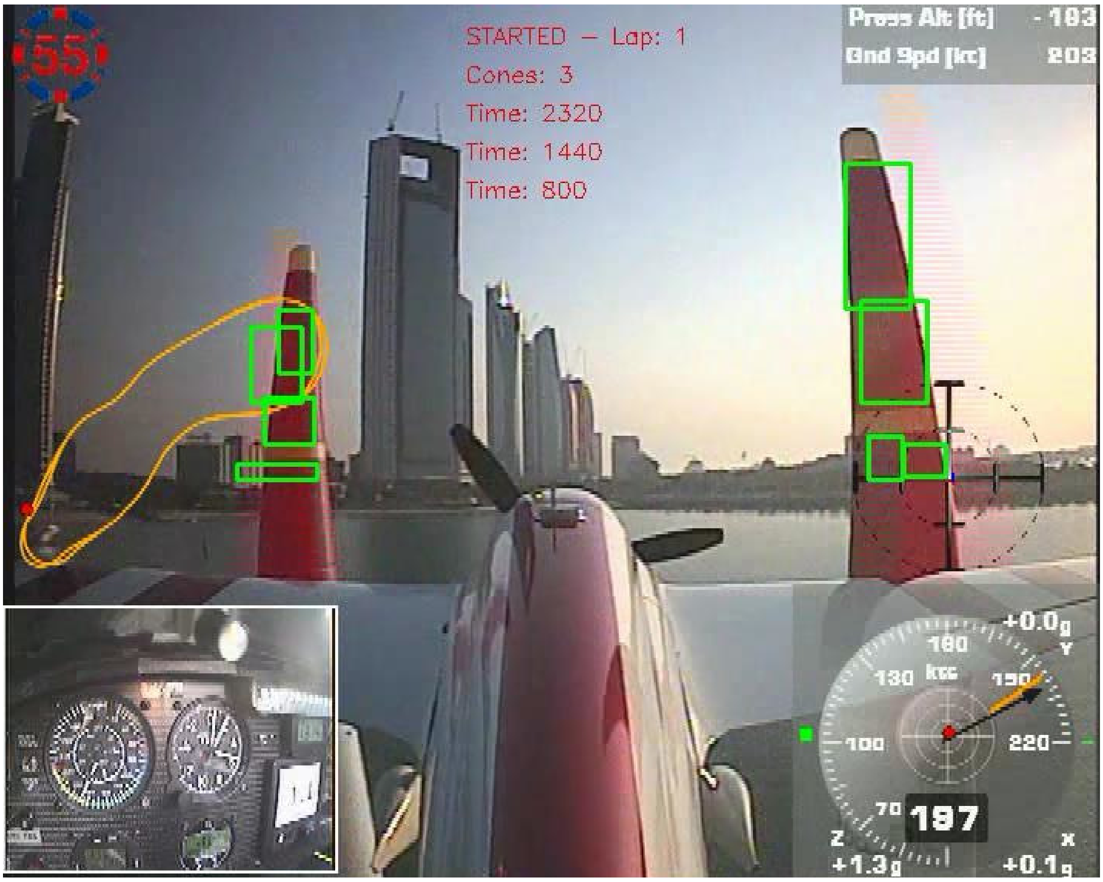


FIGURE 5.1: Camera providing an image similar to the pilot view. A vision system detects the georeferenced landmarks (race gates), marked by green rectangles.

5.2 Problem Statement

We address the problem of flight path estimation of an aerial vehicle performing aggressive and acrobatic maneuvers near the ground. This problem can be cast as a classical state estimation problem, which can be solved by filtering or smoothing algorithms. The filter approach is concerned with the estimation of the vehicle states, where the calculation of state vector is based on measurements up to the present time. In contrast, smoothers use all available measurements of a complete flight to calculate the state vector, which is normally performed offline, in a post-flight data analysis.

The estimated states $\hat{x}$ are compounded by the vehicle position $p \in \mathbb{R}^3$ and the orientation $e \in \mathbb{H}_1$ (represented by unit quaternion) relative to the north-east-down (NED) reference frame and the vehicle velocity $v \in \mathbb{R}^3$ relative to the body reference frame. Thus, by properly combining the aircraft kinematic model f with a set of sensor data measuring the input u, and output y signals, the state x, and, implicitly, the flight path

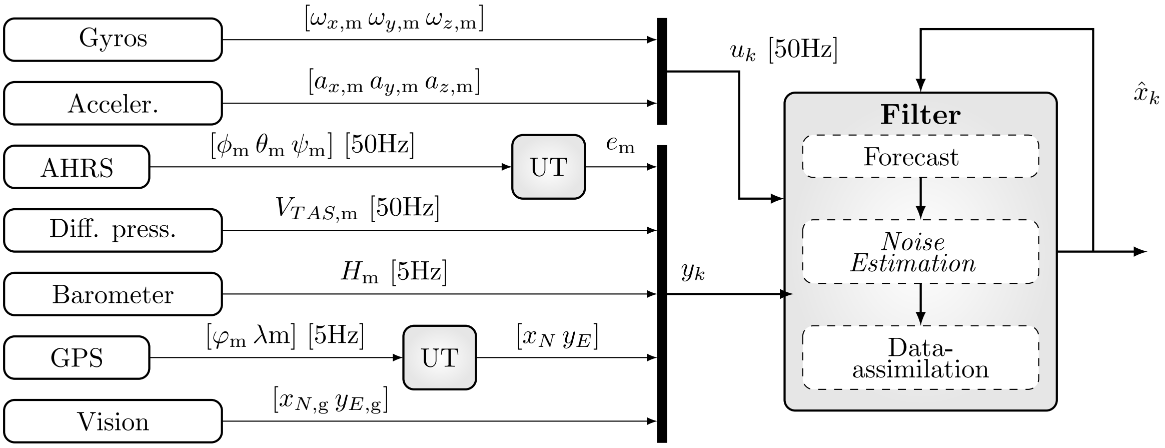

Gyros
Acceler.
AHRS
Diff. press.
Barometer
GPS
Vision
$[\omega_{x,\mathrm{m}}\,\omega_{y,\mathrm{m}}\,\omega_{z,\mathrm{m}}]$
$[a_{x,\mathrm{m}}\,a_{y,\mathrm{m}}\,a_{z,\mathrm{m}}]$
$[\phi_\mathrm{m}\,\theta_\mathrm{m}\,\psi_\mathrm{m}]$ [50Hz]
UT
e_m
$V_{TAS,\mathrm{m}}$ [50Hz]
H_m [5Hz]
$[\varphi_\mathrm{m}\,\lambda\mathrm{m}]$ [5Hz]
UT
$[x_N\,y_E]$
$[x_{N,\mathrm{g}}\,y_{E,\mathrm{g}}]$
u_k [50Hz]
y_k
Filter
Forecast
Noise Estimation
Data-assimilation
$\hat{x}_k$

Figure 5.2: Architecture of the flight path estimator. Blocks in the left represent the set of sensors that provide measurement information. In the right, the gray block represent filtering algorithms. The dashed border white blocks represent the sub-steps of filtering algorithm. The UT block represents a preprocessing step, where the measured attitude in Euler angles and geodetic coordinates are converted to unit quaternion and Cartesian coordinate, respectively. Alternatively, a smoother can be used to yield $\hat{x}_k^s$ and $P_k^{xx,s}$

.

of the vehicle can be estimated, yielding the state estimate $\hat{x}$ with covariance P^{xx}.

Figure 5.2 illustrates the architecture of the state estimator based on the available sensors. In this work, we assume that data are measured from: 3-axis accelerometer measuring linear accelerations $a_{x,\mathrm{m}}$, $a_{y,\mathrm{m}}$ and $a_{z,\mathrm{m}}$; 3-axis gyroscope measuring the angular velocity components $\omega_{x,\mathrm{m}}$, $\omega_{y,\mathrm{m}}$ and $\omega_{z,\mathrm{m}}$; barometer measuring the absolute pressure and, consequently, the altitude H_m; differential pressure sensor measuring the airspeed $V_{\mathrm{TAS,m}}$; attitude and heading reference system (AHRS) providing the orientation angles ϕ_m, θ_m and ψ_m, which are converted to unit quaternion e_m using the unscented transform; GNSS measuring the latitude φ and longitude λ, which are converted to Cartesian position in north $x_{N,\mathrm{m}}$ and east $y_{E,\mathrm{m}}$ directions using the UT; and camera providing images similar to the pilot view. From the camera data, absolute position in north and east directions, $x_{N,\mathrm{g}}$ and $y_{E,\mathrm{g}}$, are recovered by detecting the instant in which the airplane passes nearby georeferenced landmarks, as illustrated in Figure 5.1. In this figure, the camera is used to detect the gates of an air race.

The UT is employed to transform the indirect measurements to direct ones. In this way, we avoid augmenting the state vector to handle the nonadditive noise and consequently decrease the computational burden of the filtering process.

GNSS measurements are strongly affected by aggressive flight maneuvers and by television broadcast signals. Then, adaptive approaches are employed to address the time-varying uncertainty of the corresponding measurements. It is important to point out that the GNSS receptor can also provide altitude measurements. However, in our experiments, we observed more stable measurements from the barometer and decided to use them.

5.3 Mathematical Modeling

In this section the stochastic models used in the state estimators are presented. First, we present the process model f, which is based on the kinematic equations of the aircraft. Then, we show the observation model h, relating the measured output data y to the states x.

5.3.1 Process Model

The temporal evolution of a six-degree-of-freedom vehicle is described by three sets of nonlinear first-order ordinary differential equations (ODE) relative to a local NED reference frame, in which is considered a non-rotating and flat earth. These are plausible assumptions, since the vehicle flies in a small region of the planet, and therefore for short distances. For these equations, both the translational accelerations $a = [a_x \; a_y \; a_z]^T \in \mathbb{R}^3$ and body rotation rates $\omega = [\omega_x \; \omega_y \; \omega_z]^T \in \mathbb{R}^3$ are the process model's inputs u [Mulder et al., 1999, Teixeira et al., 2011].

The first set of equations describes how the translational velocity components along the rigid body, given by $v = [v_x \; v_y \; v_z]^T \in \mathbb{R}^3$, evolve with time (argument t is omitted on the right-hand side for brevity)

$$\dot{v}(t) = v \times \omega + \left(\mathcal{R}_{\mathrm{b}}^{\mathrm{NED}}\right)^T g + a \,, \tag{5.1}$$

where $g = [0 \; 0 \; g_z]^T \in \mathbb{R}^3$ is the gravity acceleration vector with $g_z = 9.81\mathrm{m/s}^2$, and $\mathcal{R}_{\mathrm{b}}^{\mathrm{NED}}$ is the orthogonal rotation matrix that represents the rotation of the body reference frame in relation to the NED reference frame as follows [Beard and McLain, 2012, p. 256],

$$\mathcal{R}_{\mathrm{b}}^{\mathrm{NED}} = \begin{bmatrix} e_0^2+e_1^2-e_2^2-e_3^2 & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & e_0^2-e_1^2+e_2^2-e_3^2 & 2(e_2e_3-e_1e_0) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & e_0^2-e_1^2-e_2^2+e_3^2 \end{bmatrix}, \tag{5.2}$$

where e_0, e_1, e_2, and e_3 are components of a unit quaternion.

The set of equations that relates the center of gravity position, $p = [x_N \; y_E \; z_D]^T \in \mathbb{R}^3$, relative to the NED reference frame is given by

$$\dot{p}(t) = \mathcal{R}_{\mathrm{b}}^{\mathrm{NED}} v \,. \tag{5.3}$$

Aiming at discrete-time state estimators, the continuous-time dynamic equations (5.1) and (5.3) are discretized by integrating over time interval $[(k-1)T, \;\; kT]$, where $t = kT$ relates continuous time to the discrete index and $T > 0$ is the sampling period. In this case, only the right-hand end-point of $[(k-1)T, \;\; kT]$ given by $x_k \triangleq x(kT)$ is used. In this case study, the 4th order Runge Kutta integration with fixed step is used for discretization; [Simon, 2006, p. 26].

Finally, the attitude is represented by a unit quaternion, which forms a four-dimensional algebra, denoted by $\mathbb{H}_1$, over the real numbers. Thus, the attitude process function is given in its discretized form as [Crassidis and Markley, 2003]

$$\mathrm{vec}\,(e_k) = A_{k-1}\mathrm{vec}\,(e_{k-1})\,, \tag{5.4}$$

where

$$A_{k-1} \triangleq \cos\left(\frac{T}{2}\|\omega\|\right) \mathrm{I}_{4\times 4} + \frac{1}{\|\omega\|}\sin\left(\frac{T}{2}\|\omega\|\right)\Omega(\omega)\,,$$

$$\Omega(\omega) \triangleq \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}.$$

All inertial instruments exhibit practical errors, such as random noise, bias, scale factor, and cross-coupling. In most cases, the bias is the dominant term in the overall errors [Groves, 2013]. Then, it is considered that the measured input vector $u_k = \left[a_k^T \;\; \omega_k^T\right]^T \in \mathbb{R}^6$ is corrupted by bias and random noise and is modeled as

$$u_{\mathrm{m},k} = u_k + \beta_k + q_{u,k} \,, \tag{5.5}$$

where "m" denotes the onboard measurements, $u_{\mathrm{m},k} = \left[a_{x_\mathrm{m}} \;\; a_{y_\mathrm{m}} \;\; a_{z_\mathrm{m}} \;\; \omega_{x_\mathrm{m}} \;\; \omega_{y_\mathrm{m}} \;\; \omega_{z_\mathrm{m}}\right]^T \in \mathbb{R}^6$ are the accelerations and angular rates measured by the accelerometers and gyroscopes, respectively, $\beta_k = \left[\beta_{a_x} \;\; \beta_{a_y} \;\; \beta_{a_z} \;\; \beta_{\omega_x} \;\; \beta_{\omega_y} \;\; \beta_{\omega_z}\right]^T \in \mathbb{R}^6$ are the bias terms, and $q_{u,k} \sim \mathcal{N}\left([0]_{6\times1},\; Q_u\right) \in \mathbb{R}^6$ is the input noise vector. In (5.1)-(5.4), in order to use the onboard measurements, we set $u_k = u_{\mathrm{m},k} - \beta_k - q_{u,k}$.

Accelerometer and gyroscope drifts, given by the parameter vector β_k, are modeled as the random-walk process,

$$\beta_k = \beta_{k-1} + q_{\beta,k-1} \,, \tag{5.6}$$

where $q_\beta \sim \mathcal{N}\left([0]_{6\times1},\; Q_\beta\right) \in \mathbb{R}^6$. The bias components can be jointly estimated with vehicle states, yielding the *joint state vector* $\mathrm{vec}\left(x_k\right) \in \mathbb{R}^{16}$ defined as

$$x_k \triangleq \left[v_k^T \;\; p_k^T \;\; e_k \;\; \beta_k^T\right]^T \,. \tag{5.7}$$

Equations (5.1), (5.3), (5.4), and (5.6) compose the *process model* of the vehicle, which can be compactly recast as

$$x_k = f\left(x_{k-1},\, u_{k-1},\, q_{k-1},\, k-1\right) \,, \tag{5.8}$$

where f denotes a nonlinear function of state vector x_k, input vector u_{k-1}, and process noise vector $q_{k-1} \triangleq \left[q_{u,k-1}^T \;\; q_{\beta,k-1}^T\right]^T \in \mathbb{R}^{12}$.

5.3.2 Observation Model

The observation model relates the components of the state vector x_k to the measured output variables $\operatorname{vec}(y_k) \in \mathbb{R}^{10}$ given by

$$y_k \triangleq [e_{\mathrm{m}} \;\; V_{TAS,\mathrm{m}} \;\; x_{E,\mathrm{m}} \;\; y_{E,\mathrm{m}} \;\; H_{\mathrm{m}} \;\; x_{N,\mathrm{g}} \;\; y_{E,\mathrm{g}}]^T \,. \tag{5.9}$$

An AHRS provides the orientation estimates in Euler angles, which can be transformed to quaternion and used directly as a measurement of attitude. The nonlinear transformation from Euler angles to unit quaternions is given by [Beard and McLain, 2012, p. 259]

$$\operatorname{vec}(e_{\mathrm{m}}) = \begin{bmatrix} \cos\left(\frac{\psi_{\mathrm{m}}}{2}\right)\cos\left(\frac{\theta_{\mathrm{m}}}{2}\right)\cos\left(\frac{\phi_{\mathrm{m}}}{2}\right) + \sin\left(\frac{\psi_{\mathrm{m}}}{2}\right)\sin\left(\frac{\theta_{\mathrm{m}}}{2}\right)\sin\left(\frac{\phi_{\mathrm{m}}}{2}\right) \\ \cos\left(\frac{\psi_{\mathrm{m}}}{2}\right)\cos\left(\frac{\theta_{\mathrm{m}}}{2}\right)\sin\left(\frac{\phi_{\mathrm{m}}}{2}\right) - \sin\left(\frac{\psi_{\mathrm{m}}}{2}\right)\sin\left(\frac{\theta_{\mathrm{m}}}{2}\right)\cos\left(\frac{\phi_{\mathrm{m}}}{2}\right) \\ \cos\left(\frac{\psi_{\mathrm{m}}}{2}\right)\sin\left(\frac{\theta_{\mathrm{m}}}{2}\right)\cos\left(\frac{\phi_{\mathrm{m}}}{2}\right) + \sin\left(\frac{\psi_{\mathrm{m}}}{2}\right)\cos\left(\frac{\theta_{\mathrm{m}}}{2}\right)\sin\left(\frac{\phi_{\mathrm{m}}}{2}\right) \\ \sin\left(\frac{\psi_{\mathrm{m}}}{2}\right)\cos\left(\frac{\theta_{\mathrm{m}}}{2}\right)\cos\left(\frac{\phi_{\mathrm{m}}}{2}\right) - \cos\left(\frac{\psi_{\mathrm{m}}}{2}\right)\sin\left(\frac{\theta_{\mathrm{m}}}{2}\right)\sin\left(\frac{\phi_{\mathrm{m}}}{2}\right) \end{bmatrix} . \tag{5.10}$$

Since the Euler angles estimates are uncertain, we perform the propagation through the nonlinear functions by means of the unscented transform (reviewed in the Section 3.2.1), using a preprocessing similar to the one used in [Terra et al., 2014]. As in [Kraft, 2003, Vartiainen et al., 2014], we assume that the measured unit quaternion $\operatorname{vec}(e_{\mathrm{m},k}) = [e_{0,\mathrm{m}} \;\; e_{1,\mathrm{m}} \;\; e_{2,\mathrm{m}} \;\; e_{3,\mathrm{m}}] \in \mathbb{R}^4$ is a random variable, in which the random noise is incorporated in the model by means of quaternion multiplication, preserving the unit norm constraint. Then, the attitude measurement equation is given by

$$e_{\mathrm{m}} = e \oplus r_e \,, \tag{5.11}$$

where $r_e \sim \mathcal{N}\left([0]_{3\times 1}, R_e\right) \in \mathbb{R}^3$ is the measured quaternion noise parameterized as *rotation vector*. The transformation between unit quaternion and *rotation vector* was presented in Section 4.1.2.

The true air speed $V_{\mathrm{TAS,m}}$ is measured from a differential pressure sensor. The resultant air velocity components along the axes of the rigid body is given by

$$V_{\text{TAS,m}} = \sqrt{v_x^2 + v_y^2 + v_z^2} + r_{V_{\text{TAS}}} \, , \tag{5.12}$$

where $r_{V_{\text{TAS}}} \sim \mathcal{N}\left(0, R_{V_{\text{TAS}}}\right) \in \mathbb{R}$ is the measurement noise, $V_{\text{TAS,m}} = \sqrt{2P_{dif,\text{m}}\rho}$ is the indirect measured airspeed, $\rho = 1.225\text{kg/m}^3$ is the air density at vehicle location, and $P_{\text{dif,m}}$ is the measured differential pressure.

The GNSS sensor provides absolute position in terms of latitude and longitude. As the presented mathematical models rely on a Cartesian system, where the earth surface is assumed to be flat and non-rotate, this geodetic information must be transformed to a local Cartesian coordinate system. In this thesis, the Universal Transverse Mercator coordinate system (UTM) [Snyder, 1987] is used to transform from geodetic information into east and north directions. As before, the UT is used to perform this transformation. In doing so, we obtain the linear measurement model

$$\begin{bmatrix} x_{\text{N,m}} \\ y_{\text{E,m}} \end{bmatrix} = \begin{bmatrix} x_{\text{N}} \\ y_{\text{E}} \end{bmatrix} + \begin{bmatrix} r_{x_N} \\ r_{y_E} \end{bmatrix} , \tag{5.13}$$

where $\left[r_{x_{\text{N}}} \;\; r_{y_{\text{E}}}\right]^T \sim \mathcal{N}\left(\left[\bar{x}_{\text{N}} \;\; \bar{y}_{\text{E}}\right]^T, R_{GNSS}\right) \in \mathbb{R}^2$ is the measurement noise, in which $\left[\bar{x}_{\text{N}} \;\; \bar{y}_{\text{E}}\right] \in \mathbb{R}^2$ represents an unknown offset in the measurement. We assume that the time-varying errors in the GNSS measurements can be taken into account by properly estimating the unknown offset and the corresponding covariance.

Altitude is inferred from the atmospheric pressure provided by barometer, yielding

$$H_{\text{m}} = -z_{\text{D}} + r_H \, , \tag{5.14}$$

where $r_H \sim \mathcal{N}\left(0, \; R_H\right) \in \mathbb{R}$, $H_{\text{m}} = P_{\text{m}}/\rho g_z$ is the indirect measured altitude, P_{m} is the absolute pressure, and ρ is the air density at vehicle location.

Finally, the vision system yields absolute position information, which is obtained by identifying the instant in which the vehicle passes through the georeferenced landmarks.

The corresponding model is given by

$$\begin{bmatrix} x_{\mathrm{N,g}} \\ y_{\mathrm{E,g}} \end{bmatrix} = \begin{bmatrix} x_{\mathrm{N}} \\ y_{\mathrm{E}} \end{bmatrix} + \begin{bmatrix} r_{x_{\mathrm{N,g}}} \\ r_{y_{\mathrm{E,g}}} \end{bmatrix} , \tag{5.15}$$

where $\left[r_{x_{\mathrm{N,g}}} \; r_{y_{\mathrm{E,g}}}\right]^T \sim \mathcal{N}\left([0]_{2\times 1}^T , R_{\mathrm{g}}\right) \in \mathbb{R}^2$, and the subscript "g" denotes vision information.

Equations (5.11)-(5.15) form the *observation model*, which can be written in its compact form as

$$y_k = h\left(x_k,\, r_k,\, k\right) \;, \tag{5.16}$$

where h denotes a nonlinear function of state x_k and measurement noise $r_k \sim \mathcal{N}\left(\bar{r}_k, R_k\right) \in \mathbb{R}^{(n_y-1)}$, n_y is the dimension of the measurement vector. Notice that, the measurement frequency vary depending on each sensor, yielding a different measurement vector for each time instant.

5.4 Experimental Results

We now present results of both localization and path reconstruction problems from two sets of experimental flight data collected during different sessions of the Red Bull Air Race (RBAR). The RBAR World Championship is an international series of races with the participation of at least eight pilots at each race. The objective is to navigate in an aerial racetrack, delimited by air-filled pylons, called air gates, in the fastest possible way. The engines and propellers are standard for all teams. Thus, the pilots and their team dedicate their efforts to perfect airframe aerodynamics, improving the pilot skills and planing the best trajectory to follow during the race.

In this work, filtering and smoothing algorithms are used to estimate the aircraft flight path. At this point, the importance of the adaptive approach and the inclusion of vision information must be clarified. During the air race, the GNSS-information is damaged or even interrupted due to aggressive aerobatic maneuvers. In such cases, the adaptive filtering methods must take into account the variation in the reliability of the GNSS, updating its corresponding noise covariance matrix. If the measurement uncertainty increases, more weight is put into the prediction. Thus, the global information yielded by the vision system plays an important role to improve the accuracy and convergence of the state estimation.

The two datasets used in this section were collected during the races at Abu Dhabi and Croatia in 2014, referred here by *dataset-1* and *dataset-2*, respectively. Both datasets contain information from takeoff to landing. The duration of the flight tests are 907s for *dataset-1* and 847s for *dataset-2*.

5.4.1 Filtering and Smoothing Application

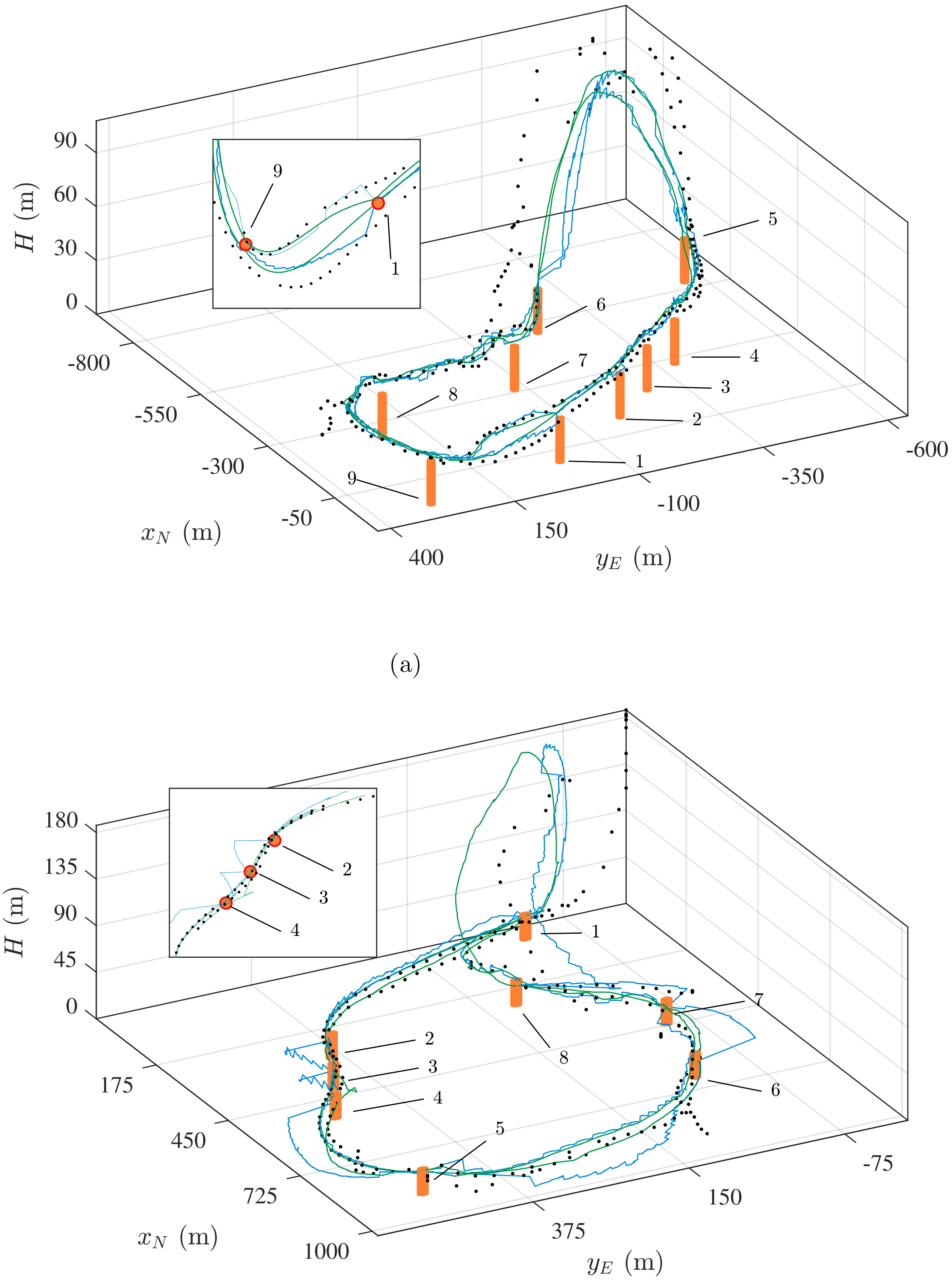
We applied the state estimators presented in Chapter 4 to localize and reconstruct the path of the aerial vehicle described in Section 5.2. The robust part of the filtering algorithms presented in Section 4.3.1 are not employed. Thus, we call this algorithm by quaternion-based adaptive unscented Kalman filter using covariance matching (QAUKF-CM). In addition, we test another algorithm based on mean and covariance matching, called by quaternion-based adaptive unscented Kalman filter using mean and covariance matching (QAUKF-MCM). All the algorithms were implemented from scratch using MATLAB. In this work, the adaptive algorithms are only used to deal with the variable uncertainty of the GNSS measurements. At this point, it is important to clarify how each one of the methods previously discussed tackle the variable uncertainty of the GNSS measurements. For the algorithms based on covariance scaling, only the entries of the noise covariance matrix related to the GNSS are adapted online. The covariance matrix adaptation is performed whenever a new GNSS measurement is available. In contrast, the IMM based algorithm runs two filters in parallel and combines their estimates. Likewise, the mode probability, γ_k, is updated whenever a new GNSS measurement is available. The smoothing algorithm uses the states estimated by the four filtering algorithms and then work backwards from the final sample to the first.

All algorithms are parameterized based on experiments performed with *dataset-1*. Matrix $Q_{1,k-1} \in \mathbb{R}^{3\times3}$ is defined as a diagonal matrix, whose elements are related to the angular rate noise of gyrometers and are given by $\sigma_\omega = [0.05,\ 0.02,\ 0.015]^T$rad/s. The diagonal matrix $Q_{2,k-1} \in \mathbb{R}^{15\times15}$ is the covariance matrix of the process additive noise, whose standard deviations were empirically adjusted as $\sigma_{\mathrm{v}} = [0.5,\ 0.5,\ 0.5]^T\mathrm{m/s}$, $\sigma_{\mathrm{p}} = [0.7071,\ 0.7071,\ 0.7071]^T\mathrm{m}$, $\sigma_{\mathrm{q}} = [0.008,\ 0.008,\ 0.008]^T\mathrm{rad}$, $\sigma_{\beta_a} = 10^{-6}\,[0.4,\ 0.4,\ 0.4]^T\mathrm{m/s^2}$, and $\sigma_{\beta_\omega} = 10^{-10}\,[1,\ 1,\ 1]^T\mathrm{rad/s}$. The covariance matrix R_k is a diagonal matrix whose entries are set according to the information provided by the instrumentation as follows. The estimated standard deviation of the AHRS, differential pressure, GNSS, barometer, and vision system measurements are $\sigma_{\mathrm{AHRS}} = [0.008,\ 0.008,\ 0.008]^T\mathrm{rad}$, $\sigma_{V_{\mathrm{TAS}}} = 0.3\,\mathrm{m/s}$, $\sigma_{\mathrm{GNSS}} = [2,\ 2]^T\mathrm{m}$, $\sigma_{\mathrm{bar}} = 0.5\,\mathrm{m}$, and $\sigma_{\mathrm{gate}} = [2,\ 2]^T\mathrm{m}$, respectively. The AHRS data, used as output measurement, have errors correlated with the process input measurements. It happens because the commercial AHRS use the same set of sensors both to compute the attitude and to provide acceleration and angular velocities. Thus, in this work, the estimated uncertainty of Euler angles σ_{AHRS} is slightly enlarged. This simplistic approach reduces the Kalman gain and then mitigate the effects of correlation. The GNSS standard deviation was chosen based on error information available on a known data sheet of another equipment. However, it is only a lower bound value, since the adaptive algorithm may increase it. The initial conditions $\hat{x}_0 \triangleq \left[\hat{x}_0^T,\ \hat{\beta}_0^T\right]^T$ are defined according to the the first measurements of the datasets with assumed zero bias. For the adaptive algorithms based on covariance matching we assume that the GNSS measurement uncertainty is approximately constant for 2 seconds, then the moving window is set to $N = 10$. For the QIMM, two models are used, one to represent the standard condition of measurement and another to represent an unreliable measurement condition. The difference between the two models is in the covariance matrix of the GNSS measurements, which is multiplied by the factor 400 in the second model to represent the case in which GNSS is unreliable. Finally, we assume no prior knowledge on the actual operating mode $\gamma_0 = [0.5,\ 0.5]^T$ and low transition probability between operating modes are set as

$$\Pi = \begin{bmatrix} 0.96 & 0.04 \\ 0.04 & 0.96 \end{bmatrix}.$$

5.4.2 Comparison Between Filters and Smoothers

Figure 5.3 shows the aircraft flight path during the race estimated by the quaternion-based adaptive unscente Kalman filter by covariance matching QAUKF-CM and QAUKF-CM-Smoother algorithms for *dataset-1* and *dataset-2*. Since both vision system and GNSS



(a)

(b)

FIGURE 5.3: 2D flight path using QAUKF-CM/smoother algorithms. (a) *Dataset-1* and (b) *Dataset-2*. GNSS data (black dot), smoother algorithm (in solid green line), filter algorithm (in solid blue line), and visual *landmarks* (orange bars and orange dot).

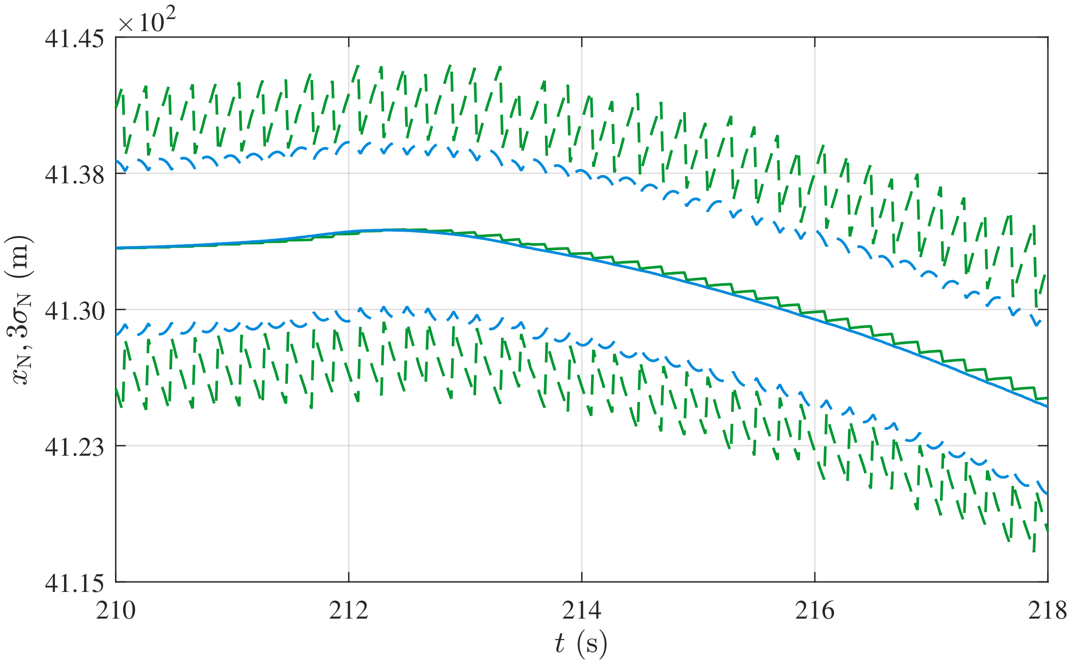


FIGURE 5.4: Window of data corresponding to the position $x_{\rm N}$, in thin solid line, and plus and minus three standard deviation $\pm 3\sigma_{x_{\rm N}}$, in thick solid line, where $\sigma_i \triangleq \sqrt{P^{xx}_{k|k,ii}}$. The results were estimated by QAUKF-CM (green line) and QAUKF-CM-Smoother (blue line).

provides information in low rate, the estimates yielded by the filtering algorithms suffer sudden corrections. As can be seen in gates 6 and 7 in Figure 5.3(a) and almost all gates in Figure 5.3(b), the inclusion of vision information yields large corrections in the position estimates. Figure 5.4 shows a window of data corresponding to position x_N and plus and minus three standard deviations. Note that, unlike the cases in which filters are used, the smoother algorithm mitigates the abrupt changes on the position estimates and also improves the estimate uncertainty.

5.4.3 Adaptability of Filters

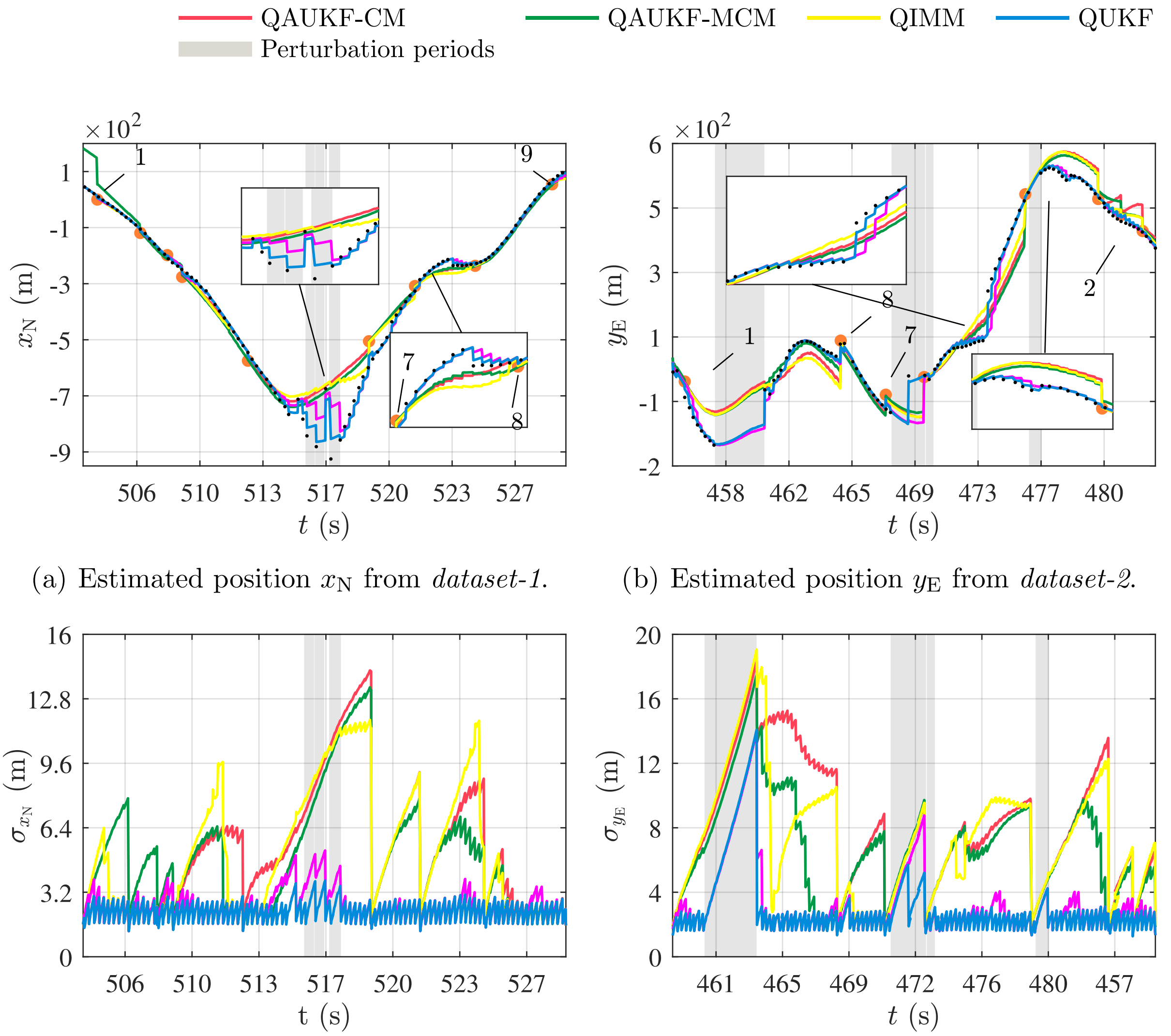
Figures 5.5(a) and 5.5(c) show the position x_N and its standard deviation for the first lap of the race, estimated from *dataset-1*. Figures 5.5(b) and 5.5(d) show the position y_E and its standard deviation for the second lap of the race, estimated from *dataset-2*. These selected intervals illustrate some erroneous GNSS measurements and also outages of GNSS signal (gray-shaded regions). During the GNSS outages, the uncertainty with respect to localization grows for all filtering algorithms due to the fact that the estimations are based only on the forecast step. Notice that, for adaptive algorithms, the uncertainty for position estimates grows between the gates and decreases whenever vision information is assimilated, while, for the QUKF, it remains almost constant. This behavior shows that the estimates are more uncertain during the race and that the adaptive algorithms are more sensitive to this variability. The adaptive filters were able to detect sudden jumps in GNSS measurements; see Figure 5.5(a) approximately from 515s to 517s. These filters were also able to detect the gradual failures of the GNSS measurements, such as in Figure 5.5(b) approximately from 455s to 458s and from 470s to 474s. Notice that, the QUKF does not adapt to the time-varying uncertainty of the GNSS measurements. Thus, if the measurement is contaminated, then the estimates may be affected in data-assimilation step, which can be observed by the jumps in the position estimates yielded by the QUKF.

Conversely, as shown in Figure 5.5(b) from 473s to 485s, the adaptive algorithms consider some GNSS measurements, that are not erroneous, with high uncertainty. This is due to previous discrepancies in localization, which is propagated ahead. It is also noted that the algorithm QAUKF-MCM yielded biased estimates, which can be seen due to a discrepancy between the estimated position at the starting of the race. In all of these cases, information provided by the vision system plays an important role, providing additional information.

5.4.4 Discussion

All adaptive algorithms were sensitive to changes in the GNSS measurement reliability, which happened during aggressive and aerobatic maneuvers of the vehicle. On the other hand, in some cases due to previous errors in the position estimates, these algorithms assigned low reliability to possibly good GNSS measurements. In such cases, vision information played an important role, yielding redundant information to correct the position estimates.

Despite the adaptive algorithms yielded similar results during the race, QAUKF-CM seems to be the preferable choice. During the experiments, QAUKF-MCM yielded biased estimates, which was noted by a discrepancy between the estimated position during the start of the race. In such case, the biased estimate was improved by the vision information



(a) Estimated position x_N from *dataset-1*.

(b) Estimated position y_E from *dataset-2*.

(c) Estimated standard deviation σ_{x_N} in x_N direction from *dataset-1*.

(d) Estimated standard deviation σ_{y_E} in x_N direction from *dataset-2*.

FIGURE 5.5: Estimated position and corresponding standard deviation, where $\sigma_i \triangleq \sqrt{P^{xx}_{k|k,ii}}$, using different algorithms. The gray-shaded regions illustrate some contaminated GNSS measurements or outages of the GNSS signal.

during the race. For QIMM, modeling different operating modes seems to be more difficult and require more experience than parameterizing the data window for the covariance matching approach.

Finally, the modified version of the QRTS smoother was employed. The smoothing algorithm improves the flight path estimated by filtering algorithms and mitigates the sudden corrections caused by low GNSS and vision rates.

5.5 Concluding Remarks

In this case study, we addressed the problem of state estimation for an aerial vehicle flying near the ground and performing aerobatic maneuvers. Since the attitude is modeled using unit quaternions, the algorithms developed in Chapter 4 are used, however the outilier rejection approaches are not employed. We also investigate the inclusion of vision information obtained from the vehicle's onboard camera to improve the localization, especially when GNSS information is unreliable or unavailable and for the case in which aggressive maneuvers are performed.

Lastly, this chapter illustrates the use of the developed adaptive algorithms to deal with erroneous GNSS measurements. In addition, some characteristics of such algorithms are investigated. However, the lack of a true flight path limits the performance comparison among the algorithms. Based on our experiments, the QAUKF-CM algorithm yielded better results and seems to be preferable in relation to the others. The setup of this algorithm seems to be less difficult than the QIMM algorithm and did not yield biased estimates as the QAUKF-MCM algorithm.

Chapter 6

Case Study: Attitude Estimation

6.1 Introduction

Attitude estimation is a crucial task for a variety of applications, such as human motion tracking [Vartiainen et al., 2014], augmented reality [Michel et al., 2017], satellite control [Gui and de Ruiter, 2017] and navigation and control of aerial [Pereira et al., 2008], and sub-aquatic vehicles [Costanzi et al., 2016]. In these applications, orientation information is usually provided by an attitude and heading reference system (AHRS).

With the raising of small flying vehicles, also known as drones, small and inexpensive AHRS have populated the market. For such systems, it is common to estimate the orientation by combining information from a magnetic, rate and gravity (MARG) sensor, also known as inertial measurement unit (IMU), usually composed of micromechanical (MEMS) three-axis gyroscope and accelerometer and a three-axis magnetometer [Jang and Liccardo, 2007]. The standard approach for attitude estimation is to compute the three components of inertial orientation by integrating the gyroscope measurements, and use the gravity projection and heading angle estimated by the accelerometers and magnetometers to correct the angles estimated with the gyro. Although theoretically simple, naive implementations of this approach may not be precise because magnetometer measurements are easily influenced by ferrous material in its vicinity, and accelerometers measure not only the gravitational direction but also linear acceleration. In these cases, it is difficult to dissociate magnetic field perturbation and linear acceleration from both the magnetic field of the earth and gravity to compute the attitude accurately, which can lead to poor estimates [Groves, 2013]. To present an alternative solution to these problems is one of the main objectives of this case study.

In this chapter, the algorithms proposed in Chapter 4 for attitude estimation are evaluated. The performance of the proposed algorithms are confronted against the non-adaptive QUKF, the open source algorithm based on complementary filter proposed in [Madgwick, 2010] and the commercial algorithm embedded in the MARG device used in our experiments, which were executed using a manipulator robot for validation purposes. The results presented in this chapter were partially published in [Chiella et al., 2019c, Chiella et al., 2019b].

6.2 Mathematical Modeling

6.2.1 Kinematic Model of Attitude

Assuming that angular rates $\omega_k \in \mathbb{R}^3$, measured by a 3-axis gyros form the input vector u_k of the dynamic system, the discrete-time attitude model is given by expression (5.4). We assume that $u_k = \omega_k \in \mathbb{R}^3$ is corrupted by random noise and bias terms, modeled as $u_{\mathrm{m},k} = u_k + \beta_k + q_{\mathrm{u},k}$, in which “m” denotes a measured variable, $u_{\mathrm{m},k} = [\omega_{x_\mathrm{m}} \ \omega_{\mathrm{y_m}} \ \omega_{\mathrm{z_m}}]^T \in \mathbb{R}^3$ are angular rates measured by a 3-axis gyro, $\beta_k = [\beta_{\omega_\mathrm{x}} \ \beta_{\omega_\mathrm{y}} \ \beta_{\omega_\mathrm{z}}]^T \in \mathbb{R}^3$ are bias terms, and $q_{\mathrm{u},k} \sim \mathcal{N}\left([0]_{3\times1},\ Q_\mathrm{u}\right) \in \mathbb{R}^3$ is the input random noise. To directly use the measured inputs in (5.4), bias terms and random noise are estimated and subtracted from the measurement. Then, $u_k = u_{\mathrm{m},k} - \beta_k - q_{\mathrm{u},k}$.

As in Equation (5.6), bias terms β_k are modeled as a random walk process, where $q_\beta \sim \mathcal{N}\left([0]_{3\times1},\ Q_\beta\right) \in \mathbb{R}^3$ and are jointly estimated with the other system states, yielding a *joint state vector* $\mathrm{vec}\left(x_k\right) \triangleq \left[\mathrm{vec}\left(e_k\right)^T \quad \beta_k^T\right]^T \in \mathbb{R}^7$.

Equations (5.4) and (5.6) compose the *process model*, which can be compactly presented as

$$x_k = f\left(x_{k-1},\, q_{k-1},\, u_{k-1},\, k-1\right), \tag{6.1}$$

where f denotes a nonlinear function of previous state x_{k-1}, with input u_{k-1}, and process noise $q_{k-1} \triangleq \left[q_{\mathrm{u},k-1}^T \ \ q_{\beta,k-1}^T\right]^T$.

6.2.2 Observation Model

The observation model relates the components of state vector x_k with the output measurement vector $y_k \in \mathbb{H}_1$, defined as $y_k \triangleq e_{\mathrm{m}}$. Measurements are corrupted by random errors and modeled as $e_{\mathrm{m}} = e_k \oplus r_k$, where $r_k \sim \mathcal{N}\left([0]_{3\times1}, R_k\right) \in \mathbb{R}^3$ is the measurement noise parameterized as a *rotation vector*. Therefore, the *observation model* may be written as

$$y_k = h\left(x_k,\, r_k,\, k\right) \,. \tag{6.2}$$

In this chapter, the measured acceleration $a_{\mathrm{m},k} = \left[a_{\mathrm{x}}\; a_{\mathrm{y}}\; a_{\mathrm{z}}\right]^T \in \mathbb{R}^3$ and magnetic field $b_{\mathrm{m},k} = \left[b_{\mathrm{x}}\; b_{\mathrm{y}}\; b_{\mathrm{z}}\right]^T \in \mathbb{R}^3$ are used to compute the unit quaternion $e_{\mathrm{m}} \in \mathbb{H}_1$. Assuming normalized measurements such that $\|a_{\mathrm{m},k}\| = 1$ and $\|b_{\mathrm{m},k}\| = 1$, the unit quaternion representing the body attitude can be computed as [Valenti et al., 2015, Valenti et al., 2016]:

$$e_{\mathrm{m}}^* = e_{\mathrm{acc}} \otimes e_{\mathrm{mag}} \,, \tag{6.3}$$

$$e_{\mathrm{acc}} = \begin{cases} \left(\lambda_1, \left[-\frac{a_{\mathrm{y}}}{2\lambda_1}\; \frac{a_{\mathrm{x}}}{2\lambda_1}\; 0\right]^T\right), & a_{\mathrm{z}} \geq 0 \\ \left(-\frac{a_{\mathrm{y}}}{2\lambda_2}, \left[\lambda_2\; 0\; \frac{a_{\mathrm{x}}}{2\lambda_2}\right]^T\right), & a_{\mathrm{z}} < 0, \end{cases} \tag{6.4}$$

$$e_{\mathrm{mag}} = \begin{cases} \left(\frac{\lambda_3}{\sqrt{2\Gamma}}, \left[0\; 0\; \frac{\lambda_3}{\sqrt{2\Gamma}}\right]^T\right), & l_{\mathrm{x}} \geq 0 \\ \left(\frac{l_{\mathrm{y}}}{\sqrt{2}\lambda_4}, \left[0\; 0\; \frac{\lambda_4}{\sqrt{2\Gamma}}\right]^T\right), & l_{\mathrm{x}} < 0, \end{cases} \tag{6.5}$$

where $\lambda_1 = \sqrt{(a_z+1)/2}$, $\lambda_2 = \sqrt{(1-a_z)/2}$, $\Gamma = l_{\mathrm{x}}^2 + l_{\mathrm{y}}^2$, $\lambda_3 = \sqrt{\Gamma + l_{\mathrm{x}}\sqrt{\Gamma}}$, $\lambda_4 = \sqrt{\Gamma - l_{\mathrm{x}}\sqrt{\Gamma}}$ and $l_{\mathrm{m},k} = \left[l_{\mathrm{x}}\; l_{\mathrm{y}}\; l_{\mathrm{z}}\right]^T$ such that$(0, l_{\mathrm{m},k}) = e_{acc}^*\left(0, b_{\mathrm{m},k}\right) e_{acc}$.

Because these equations are nonlinear, the unscented transform, presented in section (4.2.1), is used to propagate the measured acceleration $a_{\mathrm{m},k}$ and magnetic field $b_{\mathrm{m},k}$ errors through equations (6.3)–(6.5). These errors are modeled as zero mean random

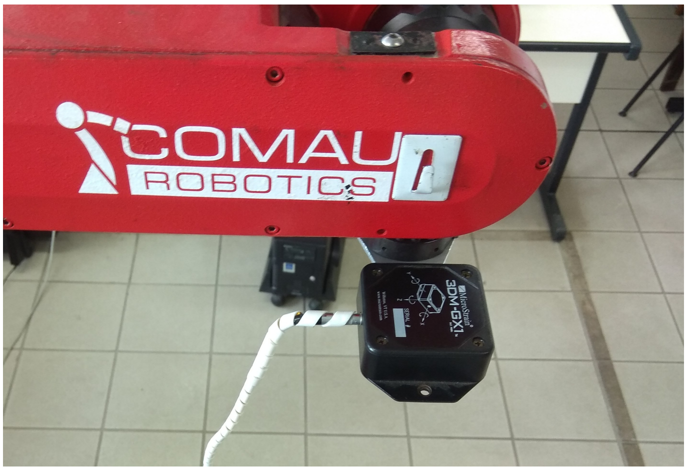


FIGURE 6.1: Experimental setup using the MicroStrain 3DM-GX1® IMU and the Comau Smart Six robot.

errors. In so doing, we obtain e_{m} used in (6.2).

6.3 Experimental Results and Discussion

In this section we compare the performance of the proposed QRAUKF-CMl, and QIMM algorithms with the classical QUKF algorithm for quaternions, the complementary filter (CF) proposed in [Madgwick, 2010], and the commercial algorithm embedded in the MicroStrain 3DM-GX1® IMU. We implemented RAUKF and QIMM using Matlab. Our code is available at `https://bitbucket.org/coroufmg/raukf_cm`. Five disturbance scenarios were evaluated: (i) abrupt and (ii) slow varying magnetic disturbances; (iii) linear accelerations; iv) individual axis rotation about the origin; and v) simultaneous axes rotations about the origin. The last two experiment, scenarios (iv)-(v), suffer linear accelerations due to the lever arm between the robot end effector and IMU[1]. Actual data was collected at 40Hz from the IMU, which was mounted on the end effector of a Comau Smart Six manipulator, used to perform controlled movements and to provide accurate orientation information. Figure 6.1 illustrates our setup.

To set QUKF and QRAUKF-CMl, we have assumed that the covariance matrix $Q_{1,k-1} \in \mathbb{R}^{3\times3}$ is diagonal with elements related to the angular rates measured by the gyros. This matrix was estimated as $\sigma_\omega = [0.4584\ 0.3724\ 0.4927]^T$ deg/s. The process additive noise was represented by the diagonal matrix $Q_{2,k-1} \in \mathbb{R}^{6\times6}$. This ma-

[1] Videos showing the experiments are found at: `https://goo.gl/mtFSqG`

Table 6.1: Root Mean Square Error (RMSE) in degrees for disturbance scenarios (i) and (ii). The lowest RMSE results are highlighted in bold.

	Abrupt magnetic			Slow magnetic		
Algorithm	$\tilde{\phi}$	$\tilde{\theta}$	$\tilde{\psi}$	$\tilde{\phi}$	$\tilde{\theta}$	$\tilde{\psi}$
QRAUKF-CMχ^2	**0.3**	**0.9**	0.53	0.07	**0.09**	1.85
QRAUKF-CMH	**0.3**	**0.9**	0.48	0.07	**0.09**	**1.84**
QIMM	**0.3**	**0.9**	**0.06**	**0.06**	**0.09**	11.33
QUKF	**0.3**	**0.9**	0.62	0.99	0.51	13.03
CF	**0.3**	**0.9**	0.6	9.22	11	28.9
3DM-GX1	0.34	1	0.54	0.13	**0.09**	11.28

Table 6.2: Root Mean Square Error (RMSE) in degrees for disturbance scenarios (iii), (iv), and (v). The lowest RMSE results are highlighted in bold.

	Linear acceleration			Individual rotations			Simultaneous rotations		
Algorithm	$\tilde{\phi}$	$\tilde{\theta}$	$\tilde{\psi}$	$\tilde{\phi}$	$\tilde{\theta}$	$\tilde{\psi}$	$\tilde{\phi}$	$\tilde{\theta}$	$\tilde{\psi}$
QRAUKF-CMχ^2	0.29	0.9	0.17	**1.1**	1.32	0.97	2.9	1.99	1.39
QRAUKF-CMH	0.89	**0.17**	0.16	1.08	**1.31**	0.91	2.88	**1.94**	**1.37**
QIMM	**0.2**	0.9	**0.15**	**1.07**	1.34	**0.85**	2.89	1.95	1.39
QUKF	4.0	3.78	2.37	1.97	1.73	2.23	**2.76**	2.08	4.20
CF	1.87	1.60	0.53	1.17	1.61	1.17	2.97	2.54	2.03
3DM-GX1	0.38	1	0.4	1.08	1.34	1.37	2.88	2.17	2.06

trix is related to the attitude, parametrized as a rotation vector, and the bias terms of the gyros. The standard deviations were empirically set as $\sigma_{\mathrm{v}} = [57.3 \times 10^{-20}]_{3\times1}$ deg and $\sigma_{\beta} = [57.3 \times 10^{-9}]_{3\times1}$ deg/s, for attitude and bias terms, respectively. The covariance matrix of measurements R_k is computed by the UT. The measured acceleration and magnetic field are propagated through the nonlinear function represented by equations (6.3)-(6.5). Standard deviations of accelerometer and magnetometer are $\sigma_{\mathrm{a}} = [0.0361 \;\; 0.0455 \;\; 0.0330]^T$ m/s^2 and $\sigma_{\mathrm{m}} = [11 \;\; 9.8 \;\; 98]^T \times 10^{-4}$Gauss [G], respectively, which were estimated from a temporal window of collected data with a steady-state behavior. The standard deviations σ_{ω}, σ_{a}, and σ_{m} were estimated from experimental data. For this, we have used a window of approximately 20 s. During this calibration process, the MARG sensor was kept stationary (steady-state behavior). The tuning parameters were estimated before performing the main state estimation experiments, during which they remain unaltered. Observe that the measurement covariance matrix R_k is updated online, in contrast with the process covariance matrices $Q_{1,k-1}$ and $Q_{2,k-1}$, that also remain unaltered. Although we used a simplistic parameter estimation approach, the parameterization seems to be appropriate, even for different experiments.

The sliding window size of QRAUKF-CMl, $l \in \{\chi^2, \mathrm{H}\}$, was empirically set to be $N = 20$ samples, which represents a period of 0.5 s during which the noise covariance is assumed to be constant, the confidence region $n_\sigma = 3$ standard deviations for RAUKF-CMH, and $\zeta = 7.8$ for RAUKF-CMχ^2.

For the QIMM, three models are used: i) standard condition of measurement; ii) unreliable measurement condition of all angles; iii) and unreliable heading measurement. The difference between the three models are in the covariance matrices of the attitude measurements, which are multiplied by the factor $G_i \in \mathbb{R}^{3\times3}$, $i \triangleq 1 \ldots 3$, set as

$$G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; G_2 = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}, \; G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}.$$

The measurement standard deviation is around 0.005 rad, which means that we are considering that the standard deviation during abnormal measurements is around 0.5 rad or approximately 28.5 degrees. It is important to observe that, if the measurements are independent, then we will have eight possible models. This is not the case, once the roll and pitch angles have an influence on heading angle measurement. During our experiments, we try to use eight models, but the operating mode always converged to one of the three above.

We assume no prior knowledge on the actual operating mode $\gamma_0 = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}^T$ and low transition probability between operating modes are set as the transition probability

$$\Pi = \begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.02 & 0.96 & 0.02 \\ 0.02 & 0.02 & 0.96 \end{bmatrix}.$$

CF has two parameters, the gain that quantifies the gyro measurement noise, set as

$\beta_{CF} = 0.007$, and the gain that quantifies the bias terms, set as $\zeta_{CF} = 0.01$. These values follows the authors recommendations [Madgwick, 2010].

The accuracy of state estimate $\hat{x}_k$ over a time interval $k = k_0 ... k_f$ is quantified by the root mean square error (RMSE) index

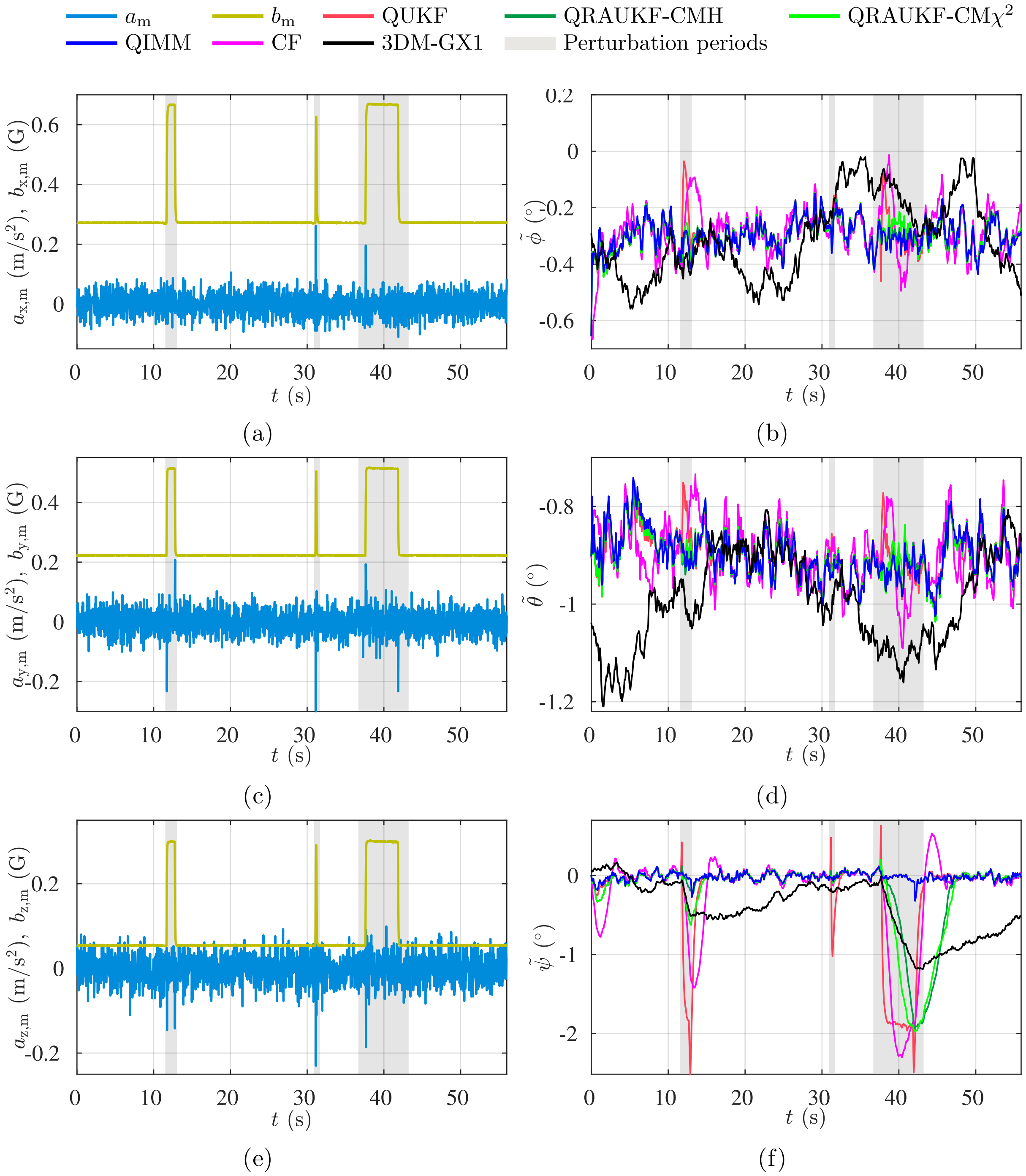
$$RMSE \triangleq \sqrt{\frac{1}{k_f - k_0 + 1} \sum_{i=k_0}^{k_f} (x_k - \hat{x}_k)^2}, \tag{6.6}$$

where x_k is the true value of state.

6.3.1 Magnetic Field Distortion

In our first experiment, the magnetic brakes of the manipulator robot are turned on and off a few times, thus causing an abrupt variation in the magnetic field that is perceived by the magnetometers. The left column of Figure 6.2 shows the linear acceleration and the magnetic field in the xyz axes. Observe that jumps in the magnetic field were generated, influencing the measurement of heading angle ψ, see Figure 6.2(f). Due to the shaking caused by the release of the brakes, some spikes of acceleration also appear. The right column of Figure 6.2 shows the estimation error for each algorithm. Notice that, QUKF and CF algorithms are more sensitive to the perturbations, converging quickly to measurements even with abnormal behavior. In contrast, QRAUKF-CMl, QIMM, and 3DM-GX1 estimates converge slowly to measurements in the presence of abnormal behavior. However, QRAUKF-CMl and QIMM are less sensitive to short duration perturbations and converge faster than other algorithms to the measurement as the disturbance is over. Figure 6.3 shows the QIMM operation mode. Observe that, the operation mode adequately converges to the expected mode of operation, γ_1 in normal operation, γ_2 for abnormal measurements in all angles, which is due to linear acceleration, and γ_3 for abnormal measurement in heading angle, which is due to magnetic disturbance.

In a second experiment, shown in Figure 6.4, the magnetic field was artificially and slowly disturbed with a magnetic material. This kind of perturbation is usually difficult to detect and can damage the estimation. Notice that QRAUKF-CMl is the less sensitive to the slow varying abnormal measurement. CF, yields the worst results as shown by



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 6.2: Results for abrupt magnetic disturbances experiment, scenario (i). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error.

RMSE in Table 6.1. The abnormal behavior of magnetic field affects the bias estimates of angular rate in the z-direction for QUKF and in all directions for CF. The experiment also shows that the QIMM algorithm is sensitive to slow varying disturbance. In fact, the mode operating γ_i converges to an unexpected mode at the beginning of the experiment, as shown in Figure 6.6.

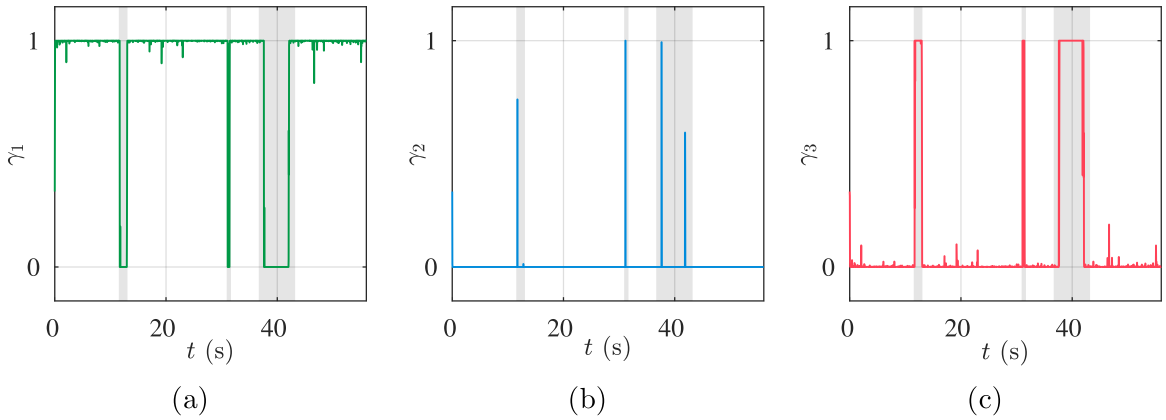


FIGURE 6.3: QIMM mode of operation for abrupt magnetic disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of heading angle.

6.3.2 Linear Acceleration Disturbance

Roll (ϕ) and pitch (θ) angles are computed by the projection of the gravity vector, which is measured by the accelerometer. However, the accelerometer measures the linear body acceleration together with the gravity vector, which masks the gravity vector observation disturbed by linear acceleration. Thus, the linear acceleration disturbs the observation of ϕ, θ, and consequently the heading angle ψ.

To test the behavior of the algorithms against the perturbation of linear accelerations, the manipulator executed independent translational movements in each axis. The left column of Figure 6.7 shows the linear acceleration and the magnetic field in the xyz axes. We observed that, even the movements being executed separately in each axis, linear accelerations appear in all axis. This is probably due to a small angle in the link joining the IMU and the robot end effector. The right column of Figure 6.7 shows the estimation error for each algorithm. Table 6.2 shows the values of RMSE for this and other experiments. Notice that, the QUKF provides the worst results. In contrast, QRAUKF-CMHl and QIMM have the best results. Figure 6.9 shows the QIMM operating mode. Observe that, during the majority of the experiment, the mode changes from γ_1 to γ_2, which represents a normal measurement condition and abnormal measurements in all angles, respectively.

Figure 6.8 shows angular rate measurements in x and y directions. It is possible to see that the measurement noise vary during the linear movement of robot end effector. This

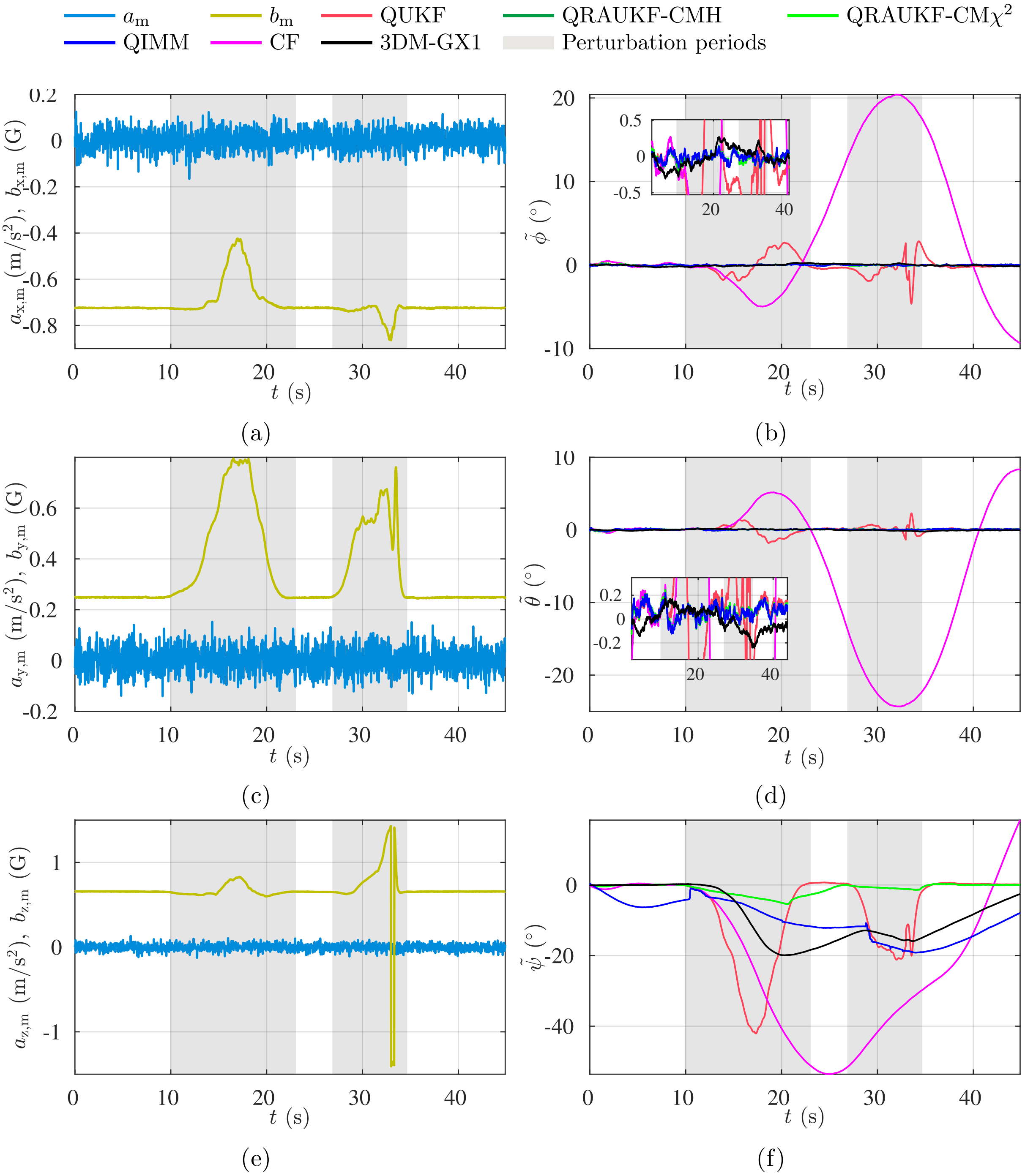
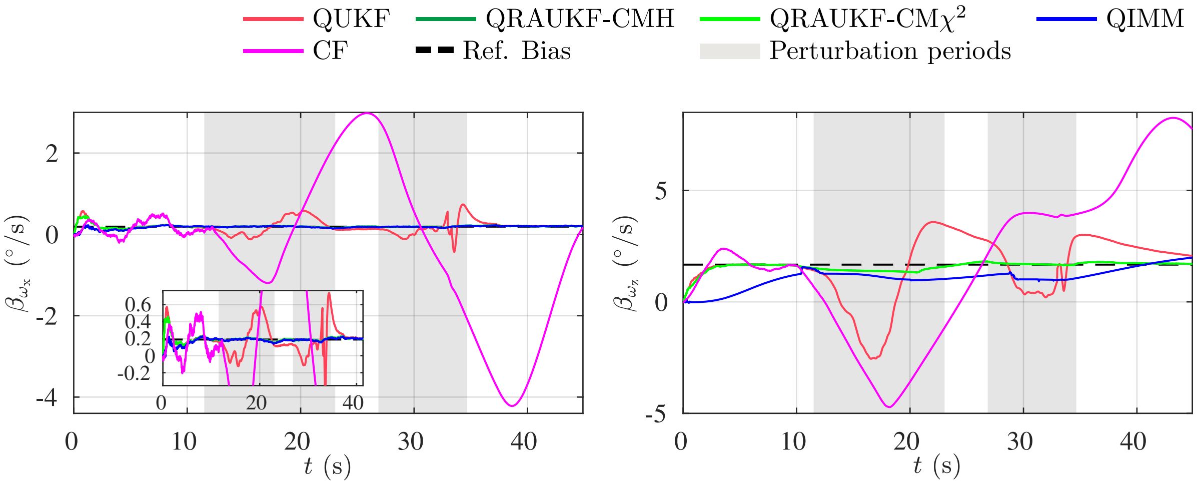


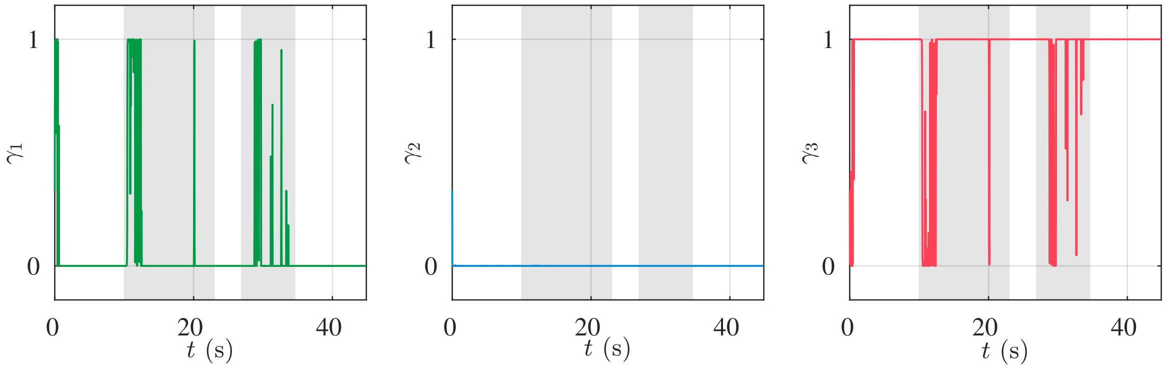
FIGURE 6.4: Results for slow varying magnetic disturbances experiment, scenario (i). In the left column, linear acceleration $a_{\rm m}$ and magnetic field $b_{\rm m}$ measurements, in the right column, the attitude error.

type of errors are not handled by the algorithms and directly damage the estimation.



(a) (b)

FIGURE 6.5: Results for abrupt magnetic disturbances experiment, scenario (i). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error.

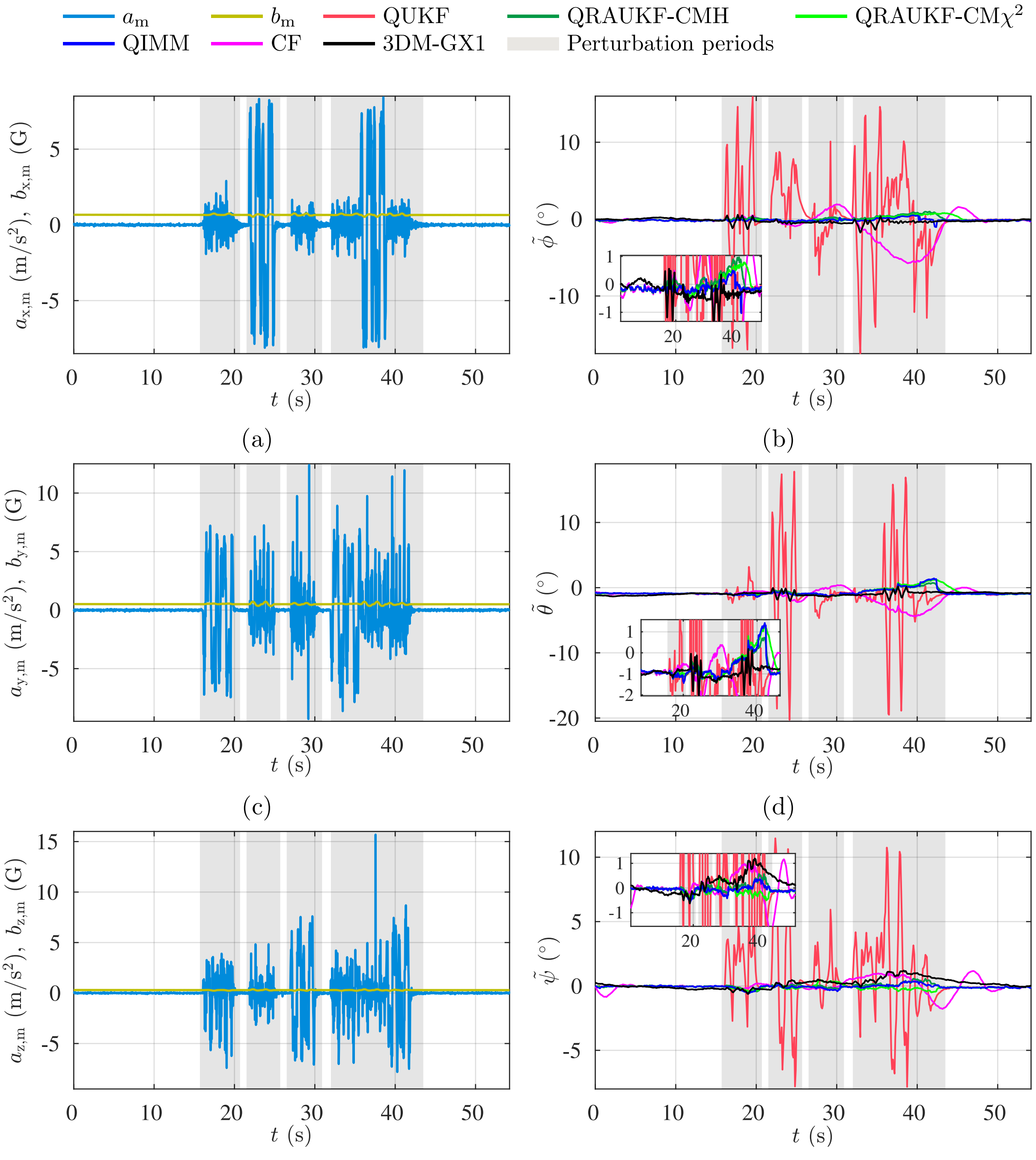


(a) (b) (c)

FIGURE 6.6: QIMM mode of operation for slow varying magnetic disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of heading angle.

6.3.3 Rotations Around the Origin

In our last two experiments, presented in Figures 6.10 and 6.11, rotations about the origin in each axis separately and simultaneously were performed. In these cases, estimates are influenced by linear accelerations that appear due to a lever arm between the IMU and the robot end effector. Figures 6.10a and 6.11a show the actual movement performed by the manipulator. Again, the proposed algorithms yield better results, as shown by Table 6.2. Notice by Figure 6.10 that QRAUKF-CMl, QIMM, and 3DM-GX1 algorithm have similar errors, however, QRUKF-CMl and QIMM converge faster to measurement after the perturbation finishes. The poor performance of QUKF and CF is due to bias



(a) (b)

(c) (d)

(e) (f)

FIGURE 6.7: Results for linear acceleration disturbance experiment, scenario (iii). In the left column, linear acceleration a_m and magnetic field b_m measurements, in the right column, the attitude error.

estimates that are influenced by linear acceleration.

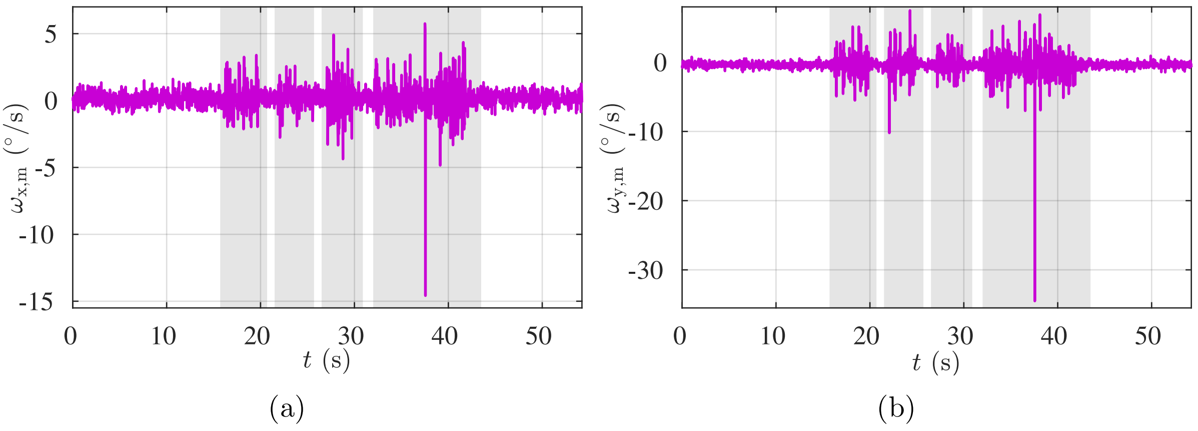


FIGURE 6.8: Angular rate ω_{x} and ω_{y}, respectively, measured by the gyros for third experiment, scenario (ii).

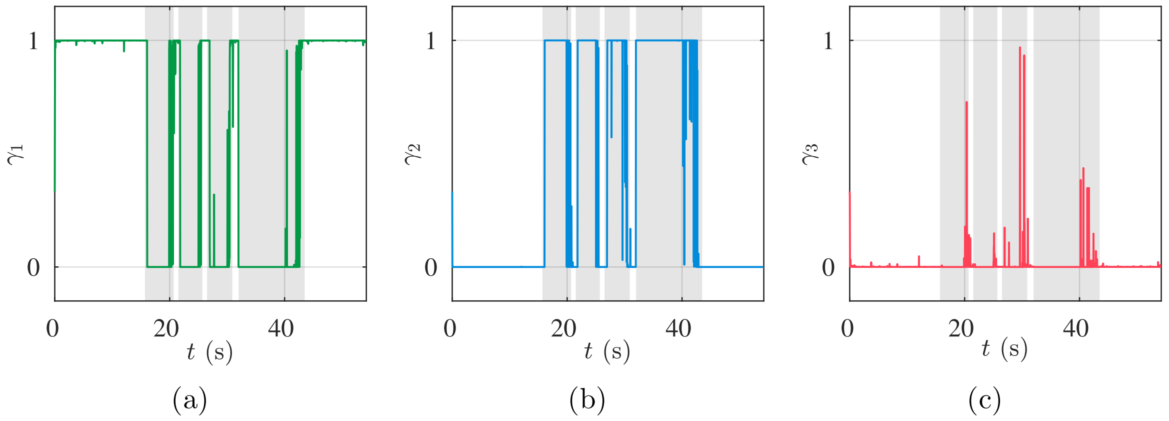
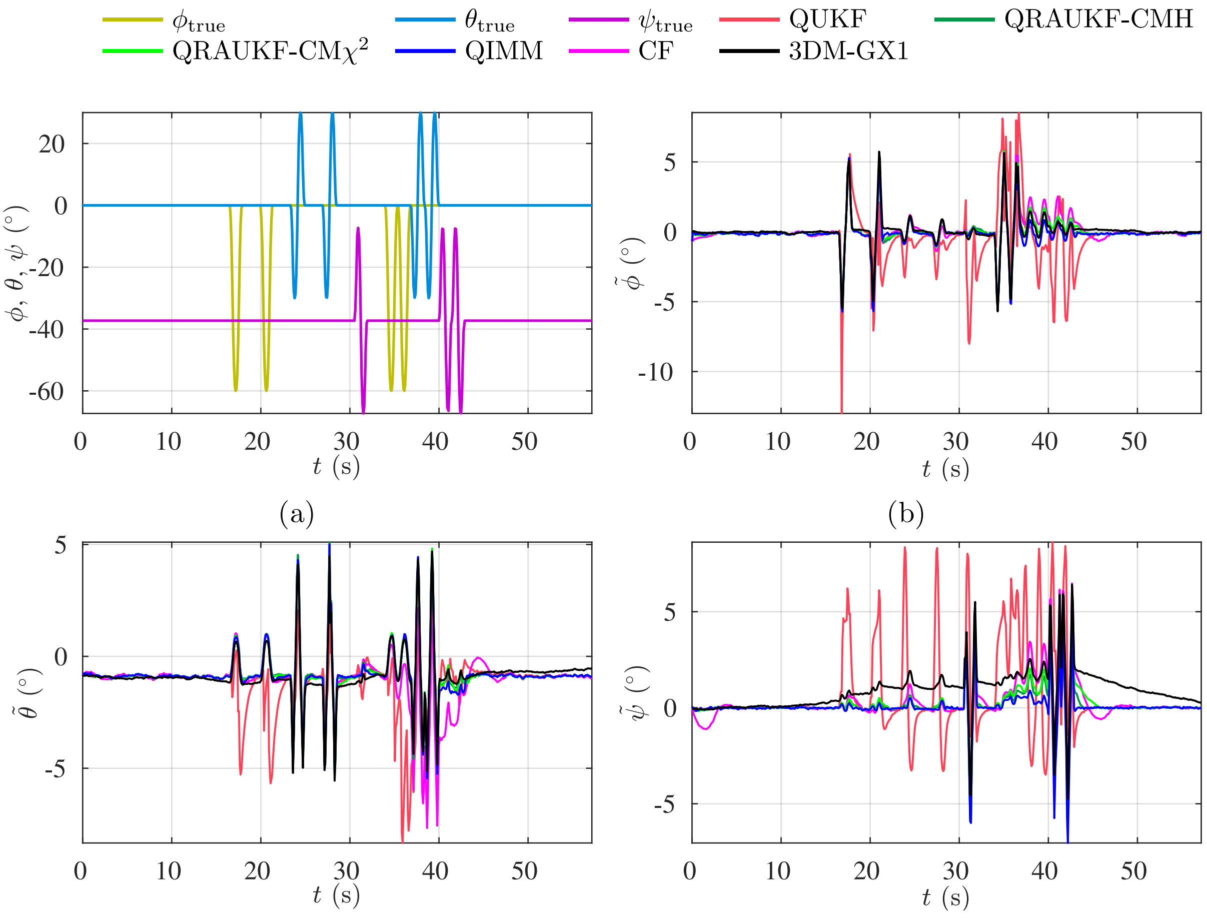


FIGURE 6.9: QIMM mode of operation for linear acceleration disturbances experiment. γ_1 represents a normal measurement behavior, γ_2 represents abnormal measurement of all angles, and γ_3 abnormal measurement of yaw angle.

6.4 Conclusions

The proposed algorithms in Chapter 4 were compared to a nonadaptive version of QUKF, a complementary filter, and commercial algorithm embedded in the IMU. Some experiments were performed to verify the performance of the algorithms in situations were distorted magnetic field and linear accelerations exist. The proposed algorithms have better RMSE results in all situations tested.



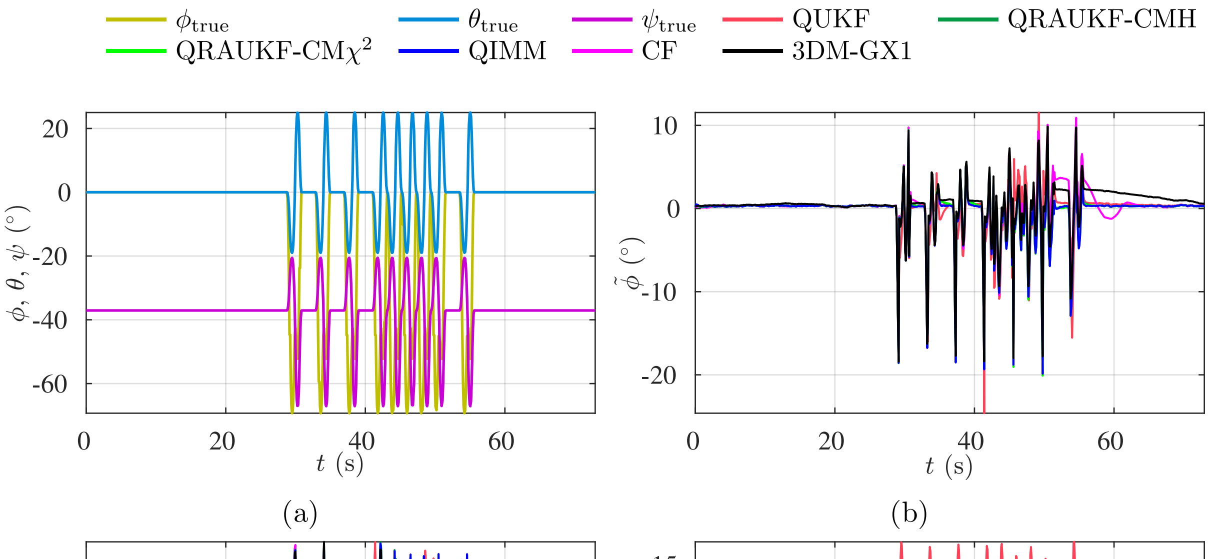
(a)

(b)

(c)

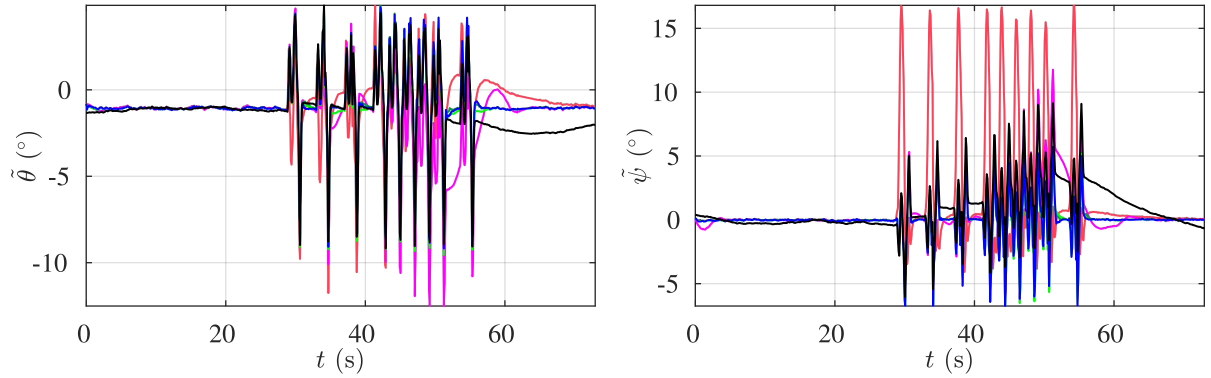
(d)

FIGURE 6.10: Results for individual axis rotation about the origin, scenario (iv). Figure (a) shows actual orientation for individual axis movements; Figures (b), (c) and (d) show the estimation error for ϕ, θ and ψ angles, respectively.



(a)

(b)



(c)

(d)

FIGURE 6.11: Results for simultaneous axes rotation about the origin, scenario (v). Figure (a) shows actual orientation for simultaneous axis movements; Figures (b), (c) and (d) show the estimation error for ϕ, θ and ψ angles, respectively.

Chapter 7

Case Study: Localization of Aerial Vehicles in Forest Environments

7.1 Introduction

Robot navigation in forests is a big challenge, mainly due to the several obstacles existent in such environments, such as tree trunks, bushes and uneven or swamped terrains. Ground robots that are able to navigate in forests are usually expensive due to their complex and adaptive locomotion systems [Freitas et al., 2010, Morita et al., 2018]. In this scenario, autonomous micro air vehicles (MAVs), such as electric drones, thus appear as viable and cost-effective alternative for such robots, since they are able to fly below the canopies of the trees and execute several missions such as exploration [Belbachir et al., 2015], search and rescue [Karma et al., 2015], ecological research [Arroyo-Mora et al., 2019] and forest management [Krisanski et al., 2018]. Although the terrain of the forest has low or no influence in the motion of the MAVs, forest environments may be GNSS denied environments, which prevents the use of standard and commercial localization systems.

A prerequisite for autonomous navigation is the accurate estimation of the vehicle states, enabling position and velocity control. In this thesis, we consider state estimation of an aerial vehicle, such as the one in Figure 7.1, flying in a forest environment. In such environments, the solutions based on the global navigation satellite systems (GNSS) are unreliable or unavailable.

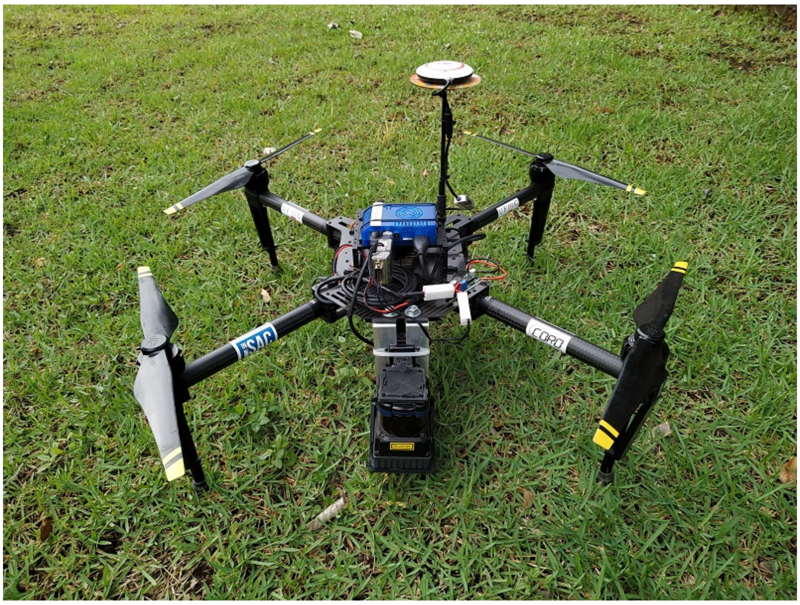


Figure 7.1: The quadrotor Matrice 100 DJI used in the state estimation experiment. The vehicle is commercialized with built-in AHRS and GNSS, and was equipped with a Hokuyo UTM-30LX-EW planar LiDAR.

Due to their complementary characteristics, the combination of inertial and GNSS measurements is the standard approach for most navigation solutions for drones and other small flying vehicles. However, for long drop-out periods of the GNSS, the lower grade attitude and heading reference systems (AHRS), which are normally embedded in these aerial vehicles, are not sufficient to estimate the position and velocity. Therefore, it is necessary to combine information from other exteroceptive sensors such as cameras and laser detection and ranging (LiDAR) sensors [Shen et al., 2014, Chambers et al., 2014].

Localization in forests was studied in [Jutila et al., 2007], where a procedure to extract landmarks is proposed. In the same work, a ground vehicle is used to collect data and process a SLAM algorithm offline. In [Cui et al., 2013, Cui et al., 2014] the navigation of an aerial vehicle in a forest is considered. The authors propose a laser-odometry (LO) algorithm and use the estimated velocity as a measurement in the update step of the KF. No treatment to the time-varying uncertainty of the LO measurements is proposed.

Combination of inertial and visual/LiDAR odometry for state estimation is commonly divided into loosely and tightly coupled approaches. The first one jointly estimates the vehicle states and visual landmarks position [Strelow and Singh, 2004]. The second approach computes the vehicle motion by comparing sequential image/scans [Shen et al., 2011, Tang et al., 2015], and, then, the estimated motion is used in the fusion algorithm.

In the present work, loosely-coupled LO is used due to its smaller computational burden.

However, in addtion two challenges arise in this strategy: the first is how LO and GNSS errors are modeled in the standard Kalman filter-based fusion algorithms, since they are usually corrupted with time varying noise, such as outliers and slow drift, which are not easily modeled by Gaussian variables. This problem can be handled by one of adaptive algorithms proposed in this thesis; the second is how to combine relative information from LO with absolute information from GNSS system into a fusion architecture;

For the problem of combining relative and absolute measurements, there are typically three approaches in the literature: using LO measurements as pseudo-global information [Shen et al., 2011]; numerical differentiating the relative motion to compute velocity [Cui et al., 2014]; and applying the so-called stochastic cloning approach [Roumeliotis et al., 1998]. In practice, pseudo-global position drifts with time, and the numerically computed velocity is a poor approximation of the actual velocity. Thus, some authors [Shen et al., 2014, Chambers et al., 2014, Song et al., 2016] advise that one should rather consider LiDAR-odometry as a relative measurement and use stochastic cloning. This is basically a state augmentation technique, where two instances of the same states, at different time instants, are concatenated in the state vector. These two instances are then used to define a measurement model that explicitly considers relative information given by LO. Our strategy uses the stochastic cloning approach with a UKF.

7.2 Problem Statement

In this chapter we address the problem of localization of an aerial vehicle in forest environments. We assume that the platform is equipped with an attitude and heading reference system, a global navigation satellite system, which may fail when the vehicle is flying inside a forest, and a LiDAR based odometry. From these measurements systems, our solution uses the following information whenever they are available: i) AHRS, providing the attitude of the vehicle represented as the unit quaternion $\mathrm{vec}\,(e_{\mathrm{m}}) = [e_{0,\mathrm{m}}\; e_{1,\mathrm{m}}\; e_{2,\mathrm{m}}\; e_{3,\mathrm{m}}]^T \in \mathbb{R}^4$, with respect to the north-east-down (NED) coordinate frame, and acceleration $a_{\mathrm{m}} = [a_{\mathrm{x,m}}\; a_{\mathrm{y,m}}\; a_{\mathrm{z,m}}]^T$ with respect to the vehicle's body reference frame; ii) GNSS, providing global position $p_{\mathrm{m}} = [p_{\mathrm{N,m}}\; p_{\mathrm{E,m}}\; p_{\mathrm{D,m}}]^T$ and velocity

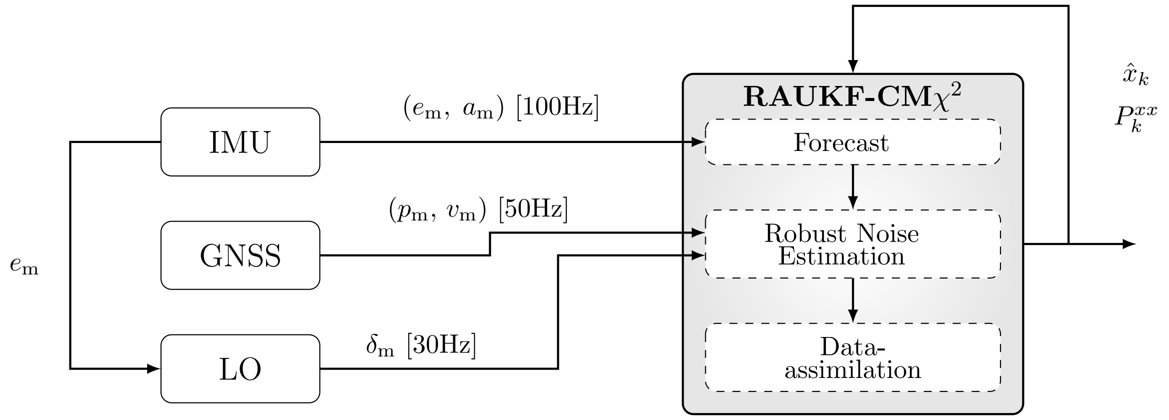


Figure 7.2: Block diagram of the proposed navigation solution. Data from the available sensors are combined using a robust and adaptive version of UKF (RAUKF) [Chiella et al., 2019c].

$v_\mathrm{m} = [v_{\mathrm{N,m}}\ v_{\mathrm{E,m}}\ v_{\mathrm{D,m}}]^T$ with respect to NED; and iii) LO, providing relative position measurements $\delta_\mathrm{m} = [\delta_{\mathrm{N,m}}\ \delta_{\mathrm{E,m}}]^T$ with respect to NED. In our notation "m" denotes onboard measurements. Figure 7.2 shows a block diagram of the proposed solution. Basically, the information provided by sensors is combined by an UKF-based fusion algorithm. Fortunately, most of the drones found in the market today are equipped with AHRS, GNSS or some combination of both that will deliver the data required by our approach. In the sequence, the LiDAR-based odometry, which is still not a popular drone accessory, will be described.

7.3 LiDAR-Based Motion Estimation in Forests

LiDAR-based odometry is a motion estimation technique that uses the matching between consecutive laser scan data to estimate the incremental motion of the vehicle. In a forest, raw laser measurements do not have much information, once most of the measurement beams do not hit any obstacle. In such a way, more information can be extracted from measurements by detecting environmental features. If done efficiently, the detection of features can decrease considerably the amount of data to be processed for motion estimation, thus reducing the computational burden of the entire system. The proposed feature-based laser-odometer algorithm is mainly composed of two steps: i) feature extraction; and ii) incremental motion estimation.

Feature extraction is the first step towards accurate motion estimation. Considering that the operating environment is a forest, the centers of tree trunks seem to be natural choices for features. To detect the trunks in the LiDAR data we have three steps. First, the range measurements are constrained to minimum and maximum values. This is necessary to reduce the influence of noise in the measurements, which increases with distance, and to eliminate beams that hit parts of the vehicle. In the second step, laser scans are segmented using edge points, detected as discontinuities in the scan:

$$\Delta_i = \frac{r_{i+1} - r_{i-1}}{2}\,, \tag{7.1}$$

where r_i is the ith range beam, for $i = 2, \ldots, n_r - 1$ where n_r maximum number of beams. Figure 7.4(a) shows the original scan, represented by a sequence of range values r_i and the detected discontinuities Δ_i. A tree trunk is probably found between the peaks down and up of the signal.

The third step of the feature extraction algorithm assumes that all trunks are cylindrical and estimates the radius r_c of the tree trunk. Figure 7.3 illustrates the radius estimation method used in our work. This method follows the procedure proposed in [Jutila et al., 2007], where the radius is computed as

$$r_c = \frac{r_m \sin(\psi_c)}{1 - \sin(\psi_c)}, \tag{7.2}$$

where $\psi_c = (\psi_b - \psi_a)/2$ is the angle of the beam that hits the center of the tree, and ψ_a and ψ_b are the angles of edges. Figure 7.4(b) shows some trees (red circles) estimated using this method. To eliminate possible wrong features, such as bushes, we only consider tree trunks with radius larger than 0.1 m and smaller than 1.5 m.

Before the estimation of the circles that model the trees, range information from the LiDAR was transformed from the body coordinate frame to the NED coordinate frame. Thus, the positions of the centers of the circles are represented in NED. These centers are then considered as features and used in an ICP algorithm [Rusinkiewicz and Levoy, 2001]. This algorithm is used to establish the correspondence between the features just found with the ones found in a previous instant of time. Thus, for a set of features $\mathcal{D} = \{d_1, d_2, \ldots, d_{n_d}\}$ at time step k and $\mathcal{M} = \{m_1, m_2, \ldots, m_{n_m}\}$ at time step $l < k$, where d_i, $m_j \in \mathbb{R}^2$ are centers of the fitted circles and n_d and n_m are the numbers of

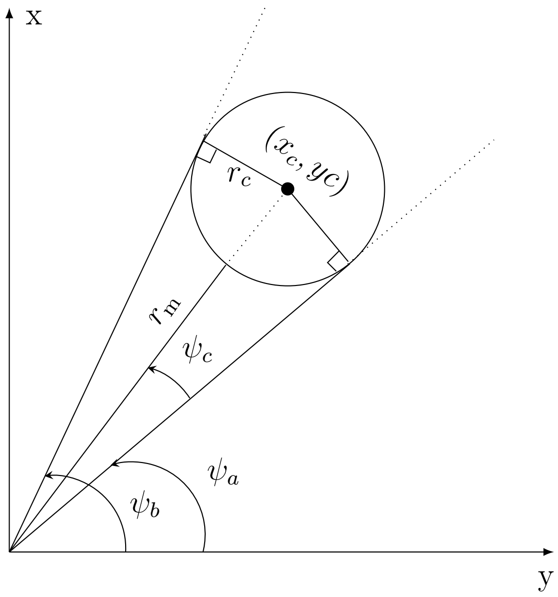
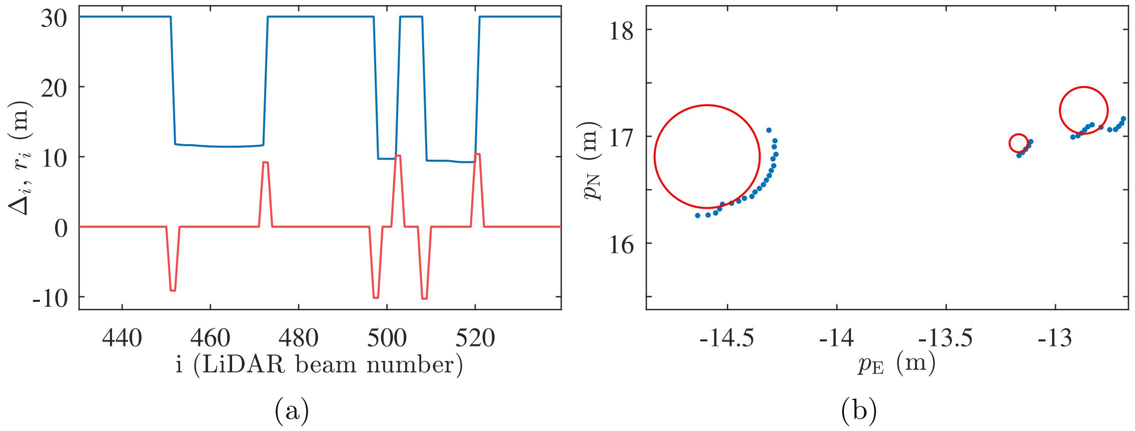


FIGURE 7.3: Procedure to compute the radius of tree trunk.



(a)

(b)

FIGURE 7.4: a) Original laser scan in blue and the detected discontinuities in red. The blue dots represent the laser beams, and the tree trunks are represented by red circles.

features extracted, the problem is to find an alignment, rotation $\mathcal{R}$ and translation ρ, that minimizes the distance between the two sets of points. This can be formulated as:

$$\epsilon(\mathcal{R},\rho) = \sum_{i=1}^{n_c} \left\| \mathcal{R} d_i + \rho - m_i \right\|^2 , \tag{7.3}$$

$$(\mathcal{R}^*, \rho^*) = \arg\min_{\mathcal{R},\rho} \epsilon(\mathcal{R},\rho) , \tag{7.4}$$

where $\epsilon(\mathcal{R},\rho)$ is called ICP metric fit error, and n_c is the number of common features between the current feature set $\mathcal{D}$ and the past feature set $\mathcal{M}$. The closed-form solution for the least-squares problem of (7.4) in 2D is given by [Lu and Milios, 1997]:

$$\mathcal{R}^* = \mathcal{R}(\psi)\,,$$
$$\rho^* = \bar{m} - \mathcal{R}^*\,\bar{d}\,,$$

where $\mathcal{R}(\psi)$ is a 2D rotation matrix that represents the vehicle rotation of angle ψ about the vertical axis, $\bar{m} \triangleq \frac{1}{n_c}\sum_i^{n_c} m_i$, $\bar{d} \triangleq \frac{1}{n_c}\sum_i^{n_c} d_i$, and

$$\psi = \arctan\left(\frac{S_{12} - S_{21}}{S_{11} + S_{22}}\right)$$

with S_{ij} being the element ij of the covariance matrix $S \triangleq \frac{1}{n_c}\sum_i^{n_c}(d_i - \bar{d})(m_i - \bar{m})^T$.

7.4 Mathematical Modeling

7.4.1 Process Model

The temporal evolution of the vehicle dynamics is described by two sets of nonlinear first-order ordinary differential equations relative to a local NED coordinate frame. The set of equations that relate the center of gravity position, $p = [p_\text{N}\; p_\text{E}\; p_\text{D}]^T \in \mathbb{R}^3$, with respect to NED coordinate frame is given by

$$\dot{p}(t) = v(t). \tag{7.5}$$

The set of equations that describes how the linear velocity components along the rigid body, given by $v = [v_\text{N}\; v_\text{E}\; v_\text{D}]^T \in \mathbb{R}^3$, evolve with time is given by

$$\dot{v}(t) = \mathcal{R}_\text{b}^\text{NED}(e)a(t) + g, \tag{7.6}$$

where $g = [0\; 0\; g_z]^T \in \mathbb{R}^3$ is the gravity acceleration vector with $g_z = 9.81\text{m/s}^2$, and $\mathcal{R}_\text{b}^\text{NED}$ is the orthogonal rotation matrix that represents the rotation of the body coordinate frame with respect to the NED coordinate frame, equation (5.2). Similar to the previous chapter, this rotation matrix is computed using the attitude provided by the AHRS. In systems for which this information is not provided, such attitude information needs to be estimated together with the other vehicle states. Also, it is important to mention that, for the case of high velocities, a term corresponding to the Coriolis effect may be added in equation (7.6). In this work, we assume the vehicle achieves small enough velocities. As before, the continuous-time dynamic equations (7.5) and (7.6) are discretized by integrating over time interval $[(k-1)T, kT]$.

It is also considered that the measured input vector $u_k = \left[a_k^T \;\; \mathrm{vec}\left(e_k\right)^T\right]^T \in \mathbb{R}^7$ is corrupted by bias $\beta_{a,k}$ and random noise q_k, and are modeled as

$$a_{\mathrm{m},k} = a_k + \beta_{a,k} + q_{a,k}, \tag{7.7}$$

$$e_{\mathrm{m},k} = e_k \oplus q_{e,k}, \tag{7.8}$$

where $q_a \sim \mathcal{N}\left([0]_{3\times 1},\; Q_a\right) \in \mathbb{R}^3$ and $q_e \sim \mathcal{N}\left([0]_{3\times 1},\; Q_e\right) \in \mathbb{R}^3$.

The accelerometer bias $\beta_{a,k}$ is modeled as a random-walk process:

$$\beta_{a,k} = \beta_{a,k-1} + q_{\beta,k-1}\,, \tag{7.9}$$

where $q_\beta \sim \mathcal{N}\left([0]_{3\times 1},\; Q_\beta\right) \in \mathbb{R}^3$. The bias components are jointly estimated with vehicle states, yielding the *joint state vector* $\check{x}_k \in \mathbb{R}^9$ defined as

$$\check{x}_k \triangleq \left[p_k^T \;\; v_k^T \;\; \beta_{a,k}^T\right]^T\,, \tag{7.10}$$

In our work, LiDAR-based odometry (LO) yields relative measurements, which means that it depends on past states. Therefore, the state vector (7.10) is augmented with a "clone", $\grave{p}_l^c = \left[p_\mathrm{N}^c \; p_\mathrm{E}^c\right]^T \in \mathbb{R}^2$, of the position states projected in the xy-plane as estimated in time step $l < k$, $\grave{p}_l = \left[p_\mathrm{N} \; p_\mathrm{E}\right]^T \in \mathbb{R}^2$. Here the term clone (represented by the superscript "c") is used to define a simple and exact copy of a past state, as is done by [Roumeliotis and Burdick, 2002]. After a new LO measurement is obtained and used to correct the system estimates (see Section 3.2.2), the cloned states are updated with the newest estimate of $\grave{p}_k$. The equation that describes the evolution of the cloned states with respect to time is given by

$$\grave{p}_k^c = \grave{p}_{k-1}^c. \tag{7.11}$$

Notice that there is no noise in this model, indicating that the cloned states remain the same until they are replaced by a new clone. We define the augmented state vector $x_k \in \mathbb{R}^{11}$ as

$$x_k \triangleq \begin{bmatrix} \check{x}_k^T & (\grave{p}_k^c)^T \end{bmatrix}^T . \tag{7.12}$$

The discretized version of (7.5) and (7.6) together with (7.9) and (7.11) compose the *process model* of the vehicle, which can be compactly recast as

$$x_k = f\left(x_{k-1},\, u_{k-1},\, q_{k-1},\, k-1\right) . \tag{7.13}$$

7.4.2 Observation Model

The observation model relates the components of the state vector x_k with the measured output variables $y_k \in \mathbb{R}^8$ given by

$$y_k \triangleq \left[p_{\mathrm{m},k}\ v_{\mathrm{m},k}\ \delta_{\mathrm{m},k}\right]^T . \tag{7.14}$$

Global position and velocity are given by the GNSS system and are modeled as

$$p_{\mathrm{m},k} = p_k + r_{p,k} \,, \tag{7.15}$$

$$v_{\mathrm{m},k} = v_k + r_{v,k} \,, \tag{7.16}$$

where $r_{p,k} \sim \mathcal{N}\left([0]_{3\times1},\, R_p\right) \in \mathbb{R}^3$ and $r_{v,k} \sim \mathcal{N}\left([0]_{3\times1},\, R_v\right) \in \mathbb{R}^3$ are the position and velocity noises, respectively.

LiDAR-odometry gives incremental displacement $\delta_{\mathrm{m},k} \in \mathbb{R}^2$ in the xy-plane, which means that the measurement depends both on the current $\grave{p}_k$ and the past states, stored as clone $\grave{p}_k^c$ (see Equation (7.12)). The augmentation of the state vector with a copy (clone) of the past state is the approach known as stochastic cloning [Roumeliotis and Burdick, 2002]. Then, assuming that the state vector is augmented with the position states $\grave{p}_k^c$, the relative measurement model is given by

$$\delta_{\mathrm{m},k} = \grave{p}_k - \grave{p}_k^c + r_{\delta,k}, \tag{7.17}$$

where $r_{\delta,k} \sim \mathcal{N}\left([0]_{2\times 1},\ R_{\delta p}\right) \in \mathbb{R}^2$.

The complete *observation model* may be written as

$$y_k = h\left(x_k,\, r_k,\, k\right), \tag{7.18}$$

where h is a function of random noise r_k and the current x_k states given by (7.15), (7.16) and (7.17).

During the filtering process, after measurement update, the estimate of cloned states $\hat{\grave{p}}_k^c$ are replaced with a new copy of current state estimate $\hat{\grave{p}}_k$ and a new covariance matrix P_k^{xx} is computed, as [Chambers et al., 2014]

$$\begin{aligned} P_k^{xx} &= C\left(P_k^{\check{x}\check{x}}\right) C^T, \\ C &= \begin{bmatrix} I_{3\times 3} & [0]_{3\times 3}\,[0]_{3\times 3} \\ [0]_{3\times 3} & I_{3\times 3}\ [0]_{3\times 3} \\ [0]_{3\times 3} & [0]_{3\times 3}\ I_{3\times 3} \\ \begin{bmatrix} I_{2\times 2}\ [0]_{2\times 1} \end{bmatrix} & [0]_{2\times 3}\,[0]_{2\times 3} \end{bmatrix}. \end{aligned} \tag{7.19}$$

Notice that, the operations with the cloned states are performed only when new relative measurement is available.

7.5 Offline Experimental Results with Actual Data

This section presents an experiment with the customized commercial aerial vehicle shown in Figure 7.1. Our DJI Matrice 100 quadrotor, which is commercialized with built-in AHRS and GNSS, was equipped with a Hokuyo UTM-30LX-EW planar LiDAR. AHRS and GNSS are, in fact, a DJI's proprietary navigation solution that runs at the low-level hardware and delivers attitude, global position and velocity information in a fairly high frequency, that is 100Hz for AHRS and 50Hz for GNSS. Our experiment was performed in a forest environment in the main campus of the Universidade Federal de Minas Gerais. Figure 7.5 shows a satellite view of the environment and, in blue, the path followed by the robot.

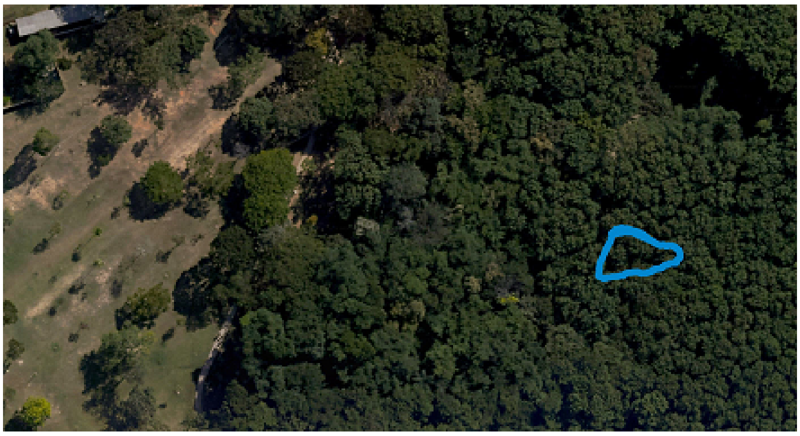


FIGURE 7.5: Satellite view of the forest where our experiments were executed. Image provided by Google Maps.

During the experiment, the vehicle flew in a manual mode and the data was recorded in a *bag* file[1]. The entire algorithm was developed in C++ using the robot operating system (ROS) as the middleware. In the sequence, we present the results of our experiment.

Due to tree canopies, while the vehicle was flying inside the forest, the drone's GNSS signal was damaged. Signal blockages however, did not last long enough to influence the GNSS position estimation but caused a velocity estimate of 0 m/s. Figure 7.6 shows a 2D image illustrating the path executed by the vehicle as given by the DJI's GNSS solution, by our laser-odometry solution and by the fusion of both, as proposed in sections 7.3 and 3.6, respectively. Figure 7.7 shows a comparison between the velocity estimates from the proprietary GNSS and our fusion algorithm. Notice that, even with the abnormal GNSS velocity measurements, the fusion algorithm was able to constantly estimate the velocity.

To access the robustness of our RAUKF-CMχ^2 strategy when it faces abnormal sensors measurements, GNSS and laser-odometry estimates were artificially contaminated with two different types of errors that are likely to happen in practice. The first one was

[1] File format to store ROS messages, `http://wiki.ros.org/Bags`

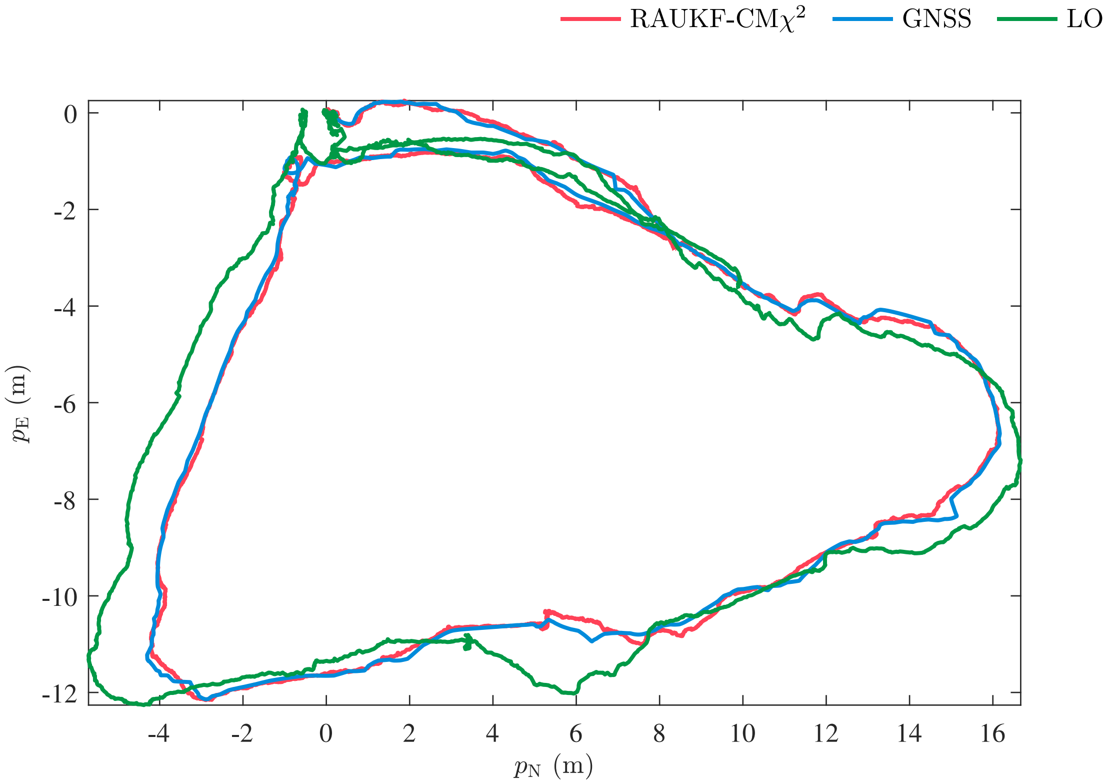


FIGURE 7.6: Two dimensional path estimated by the proposed RAUKF-CMχ^2 algorithm, DJI's proprietary GNSS solution and laser-odometry (LO).

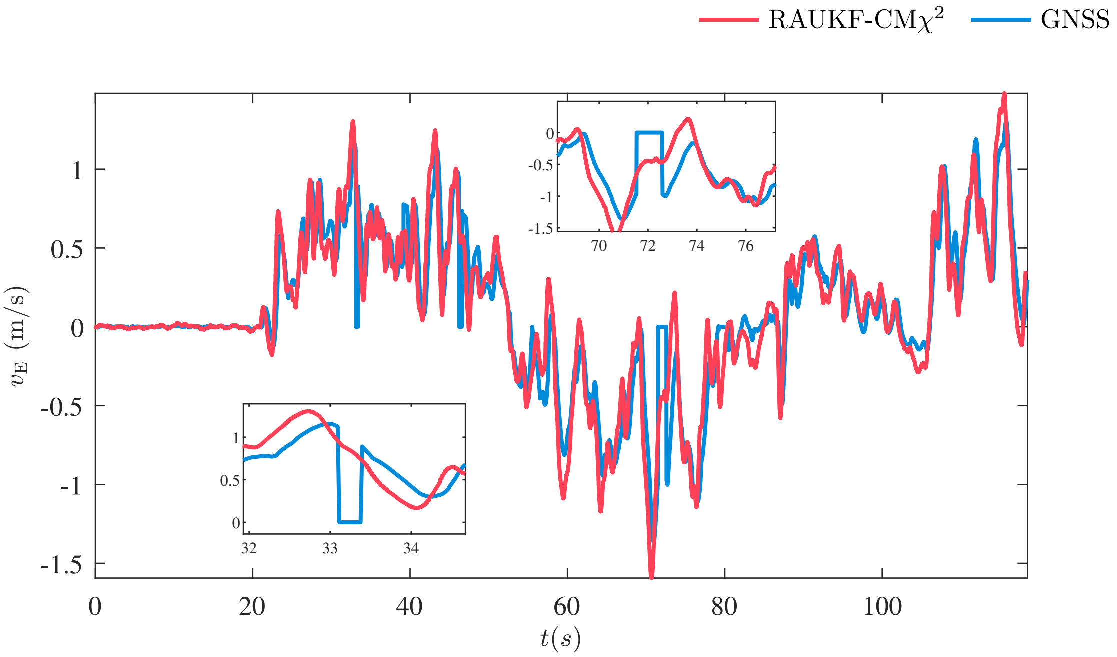


FIGURE 7.7: Velocity in East direction estimated by our algorithm and by DJI's GNSS solution.

the addition of relative position outliers, which may be caused in practice by a wrong data association and a consequent error in the computed transformation. The second error simulated satellite signal blocking. We have observed experimentally that, before a complete signal blocking, DJI's GNSS position estimate drifts slowly and the velocity estimate goes to zero immediately.

Thus, four abnormal measurements are artificially generated: i) during the 10–17 s period, laser-odometer estimates were contaminated with outliers of amplitudes 25cm, 50cm, and 1m; ii) between 20 and 40 s, the GNSS position and velocity were removed; iii) during the period of 50–60 s, GNSS position was contaminated with a drift of 0.5 m/s slope while the GNSS velocity remained in zero, and; iv) during time 90–95s the GNSS and LO measurements were removed.

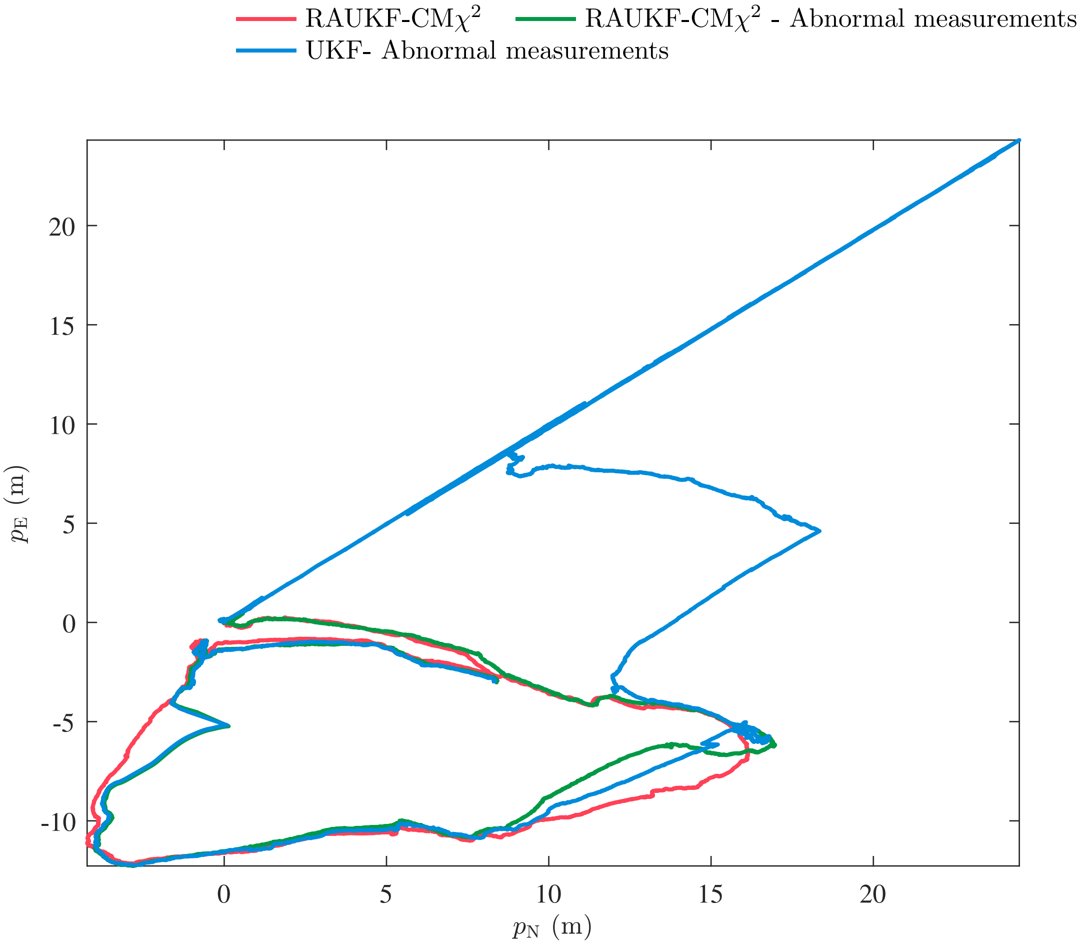


FIGURE 7.8: Two dimensional path estimated by the RAUKF-CMχ^2 both, with normal and without abnormal measurements, and by the UKF with abnormal measurements.

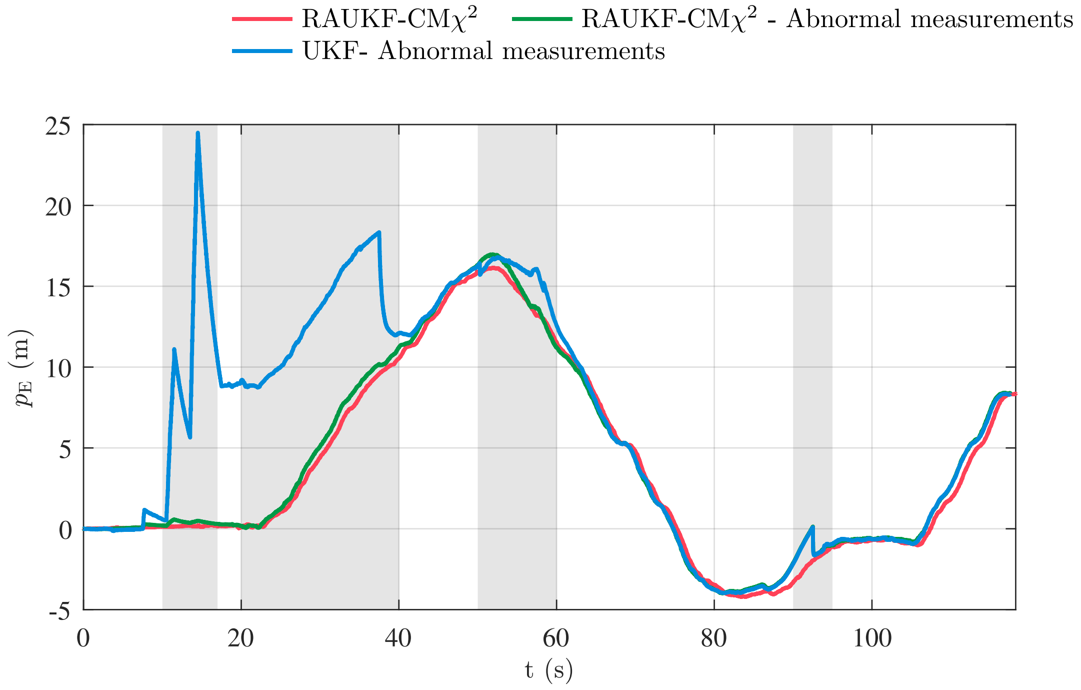


FIGURE 7.9: Position in the East direction estimated by the RAUKF-CMχ^2 both, with normal and abnormal measurements, and by the UKF with abnormal measurements.

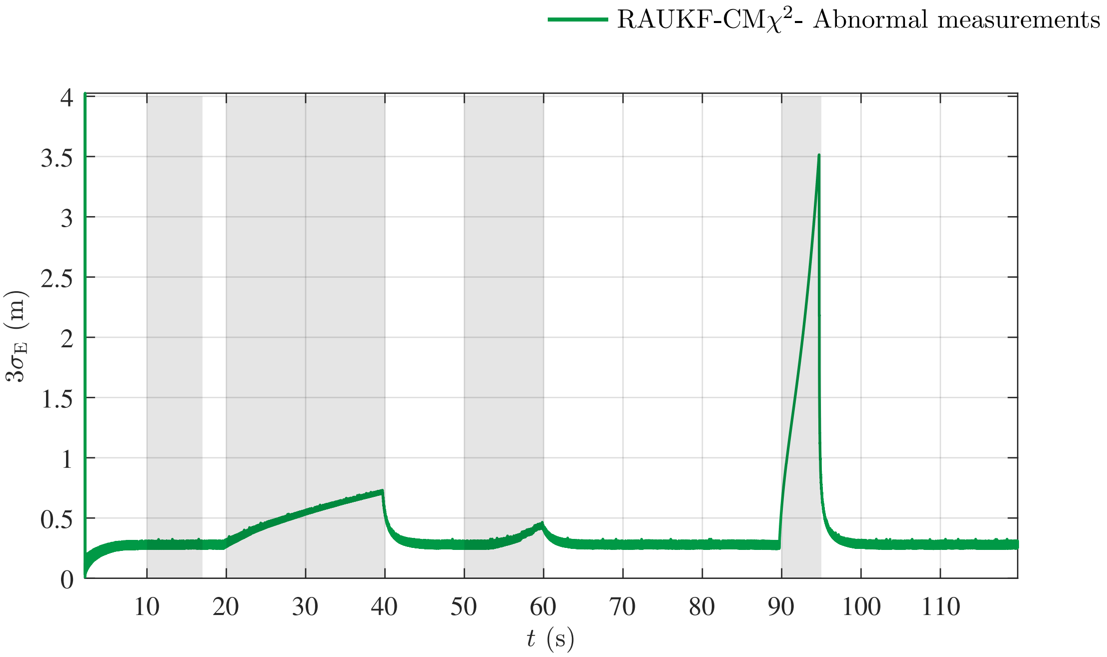


FIGURE 7.10: Three standard deviation $\sigma_{\mathrm{E}} = \sqrt{P^{xx}_{k,ii}}$ of position estimate in East direction.

Figure 7.9 shows the estimated north direction position for the proposed RAUKF-CMχ^2 algorithm and the standard UKF. RAUKF-CMχ^2 estimates without the artificial errors in GNSS and LO is also showed. Notice that, RAUKF-CMχ^2 is robust to outliers

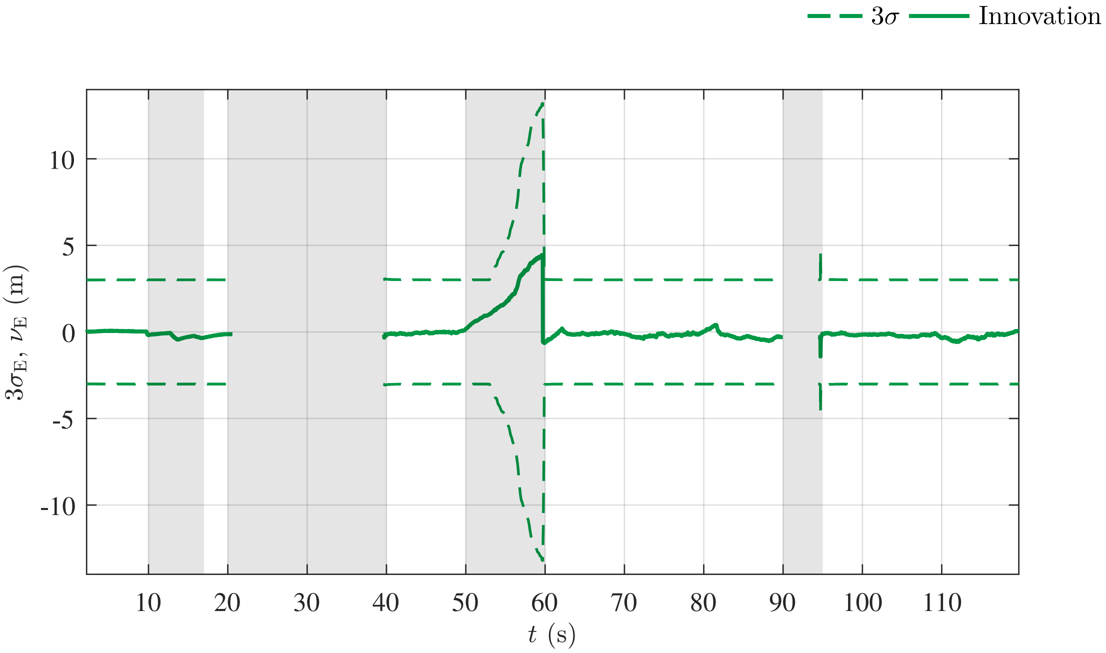


FIGURE 7.11: Innovation in the East direction regarding to the GNSS measurement, ν_E, and its uncertainty given by three standard deviation $\sigma_E = \sqrt{P_{ii}^{yy}}$. The discontinuity in data is associated with GNSS data drop out.

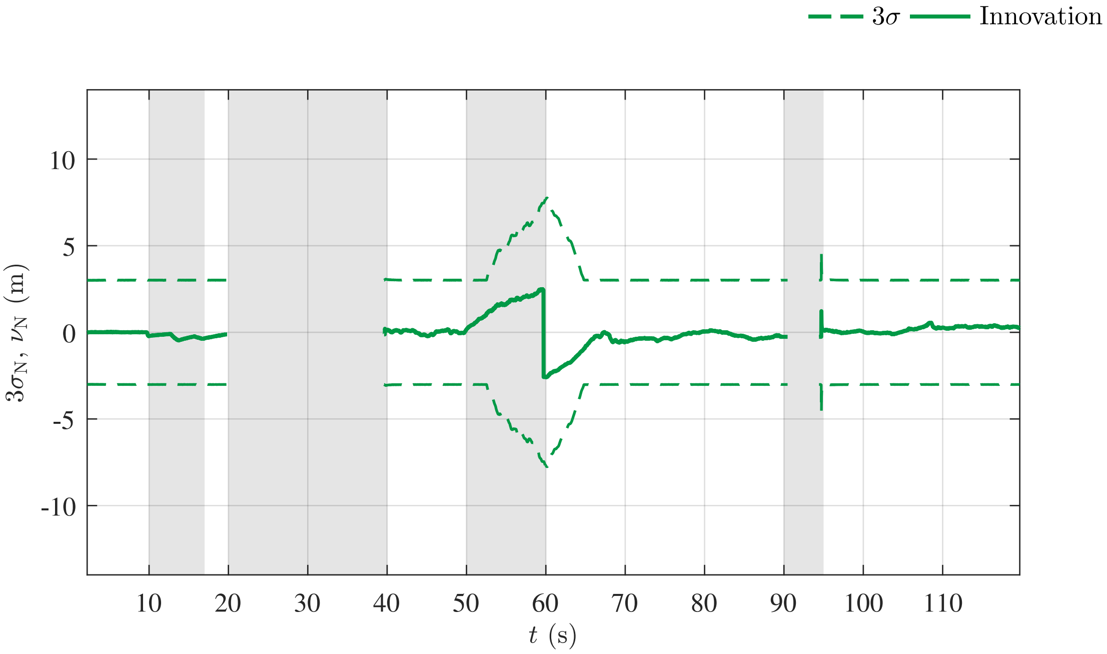


FIGURE 7.12: Innovation in the East direction regarding to the GNSS measurement, ν_N, and its uncertainty given by three standard deviation $\sigma_N = \sqrt{P_{ii}^{yy}}$. The discontinuity in data is associated with GNSS data drop out.

in the LO measurements. It is possible to see that, on the other hand, during the periods of 20–40s, the standard UKF is highly influenced by those outliers, due to the absence of GNSS measurement during this period. It is also possible to notice that the RAUKF-

CMχ^2 is less influenced by the slow drift in the GNSS position estimate from 50 to 60 s. This is due of the adaptive part of the algorithm, which is sensitive to a slow-growing error in the measurement as shown in Figure 7.11 and 7.12. When the drift error is interrupted, the GNSS measurement is not in fault anymore, but the measured absolute position is distant from the estimated position. Thus, the filter continues to adapt the covariance of GNSS measurement, see Figure 7.12 from 50 to 60 s.

Due to noise and abnormal behavior of sensor measurements, it is expected that the state estimates present errors. Thus, the state covariance matrix P_k^{xx} must capture the uncertainty growth during these periods. Figure 7.10 shows the RAUKF-CMχ^2 uncertainty estimation during the periods of 20–40 s and 90–95 s when the GNSS measurements are removed. Notice that the uncertainty in the estimate grows less in the first period due to the relative position estimates given by LO. In the second period, the LO estimates were also removed, which explains a large growth in the covariance. This behavior is consistent with the assumption that relative measurements accumulate error with time. It is also possible to see that, the covariance matrix capture the abnormal behavior of the GNSS measurement the 50–60 s time interval.

7.6 Onboard Experimental Results

In this experiment, the quadrotor flies autonomously between trees. The forest environment has sparse trees, which means that the distance between each of the trees is sufficient to allow our quadrotor navigates. Figure 7.13 shows a block diagram of our complete navigation solution. In our navigation system, the filter output at 100 Hz is used to feedback the motion control system, which consists of a path planner and a velocity controller. The path planner computes the vehicle path and uses the information from LiDAR to construct a local map for collision avoidance. Then, the velocity controller drives the vehicle through the planned path. The complete system runs on the onboard computer Odroid XU4 with an octa-core ARM processor, 2GB of RAM, running Ubuntu Mate 16.04. The entire navigation system was developed in C++ using ROS as middleware.

The experiment was performed in a forest environment in the main campus of the Universidade Federal de Minas Gerais. Figure 7.14 shows a satellite view of the environment

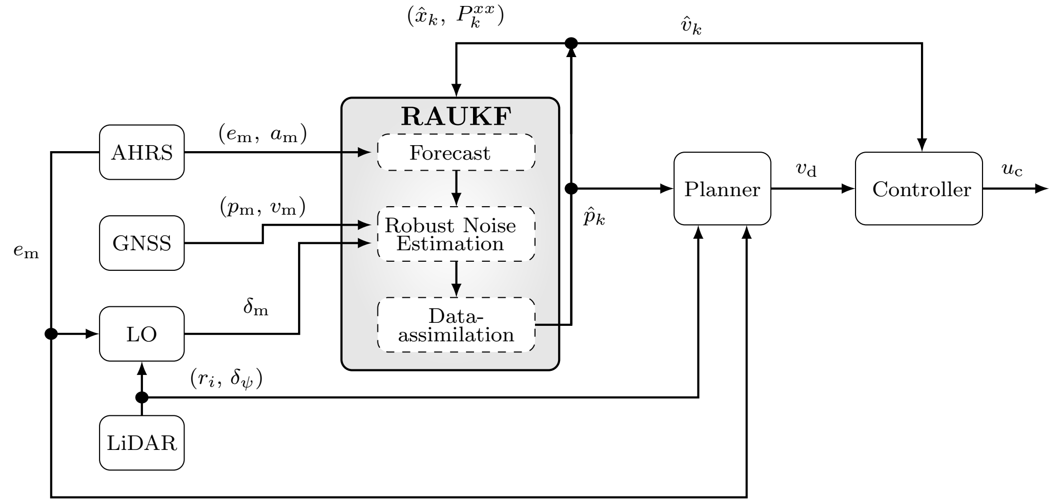


Figure 7.13: Block diagram of the proposed navigation solution. Data from the available sensors are combined using a robust and adaptive version of UKF (RAUKF-CMχ^2). The filter output feeds the motion control system, which consists of a path planner and velocity controller.

and the vehicle path as estimated by RAUKF-CMχ^2. In this figure, GNSS was not available in the yellow parts of the path. In what follows we discuss the main characteristics and behavior of the proposed navigation system during this experiment.

Due to tree canopies, while the vehicle is flying inside the forest, the drone's GNSS signal can be damaged. In our experiment, it did not happen. Therefore, to assess the robustness of our navigation system during GNSS failures, we artificially blocked the GNSS signal in the following periods, from 40-50 s, from 70-80 s, and from 100-120 s, which was done via software while the vehicle flew autonomously. Figure 7.15 shows the desired task (black curve) and RAUKF-CMχ^2 estimates (red curve) in 3D.

From Figure 7.15 we observe that our system was able to combine GNSS and relative LO measurements adequately. In addition, even in the absence of global measurements, RAUKF-CMχ^2 estimates position and velocity, allowing the vehicle motion control. Notice that the desired curve was not followed accurately, which is expected, once the initial plan did not considered obstacles in the environment.

Figures 7.16(a) and (b) show the position and velocity in the East direction. The blockage periods of GNSS signal are delimited by gray-shaded regions. We observe that RAUKF-CMχ^2 position and velocity estimates tend to converge to GNSS position and velocity measurements. This behavior is expected, once GNSS is an absolute measurement,

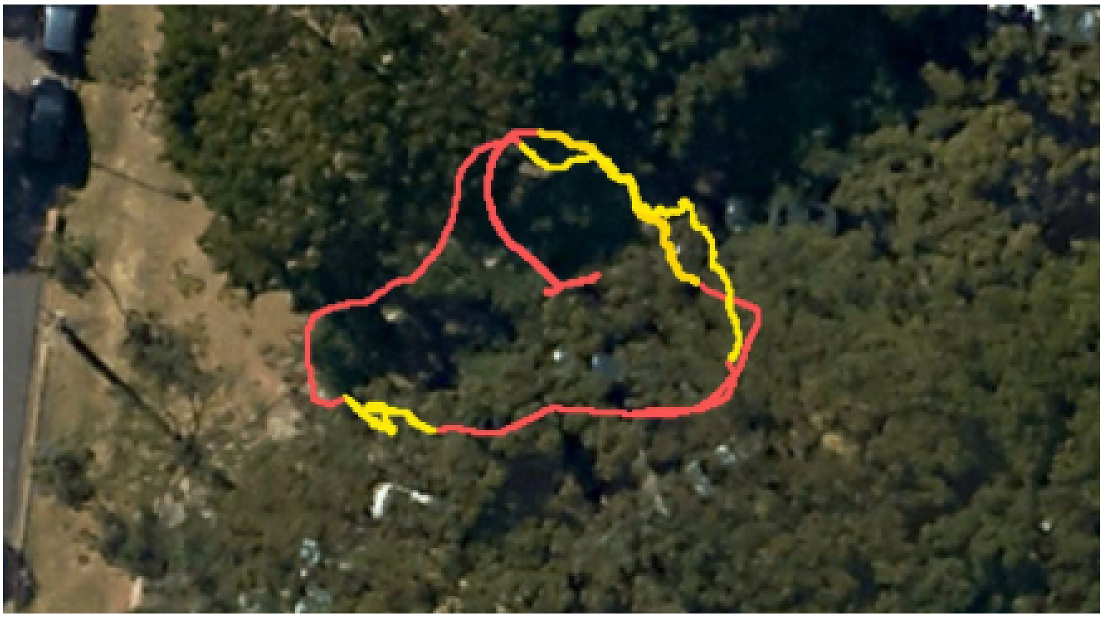


FIGURE 7.14: Satellite view of the forest where our onboard experiments were executed. Image provided by Google Maps. The path estimated by RAUKF-CMχ^2 is shown in red, when AHRS, LO, and GNSS were available and in yellow GNSS was unavailable.

which is not the case for the LO.

Figures 7.16(c) and (d) show the three standard deviations of position and velocity estimates in the East direction, respectively. Notice that, when only the LO relative measurements are used in the data-assimilation step, the uncertainty grows unbounded, reflecting the error integration effect and unobservability of global position. On the other hand, the availability of relative measurements, make the estimated velocity uncertainty bounded, which means that the velocity is observable.

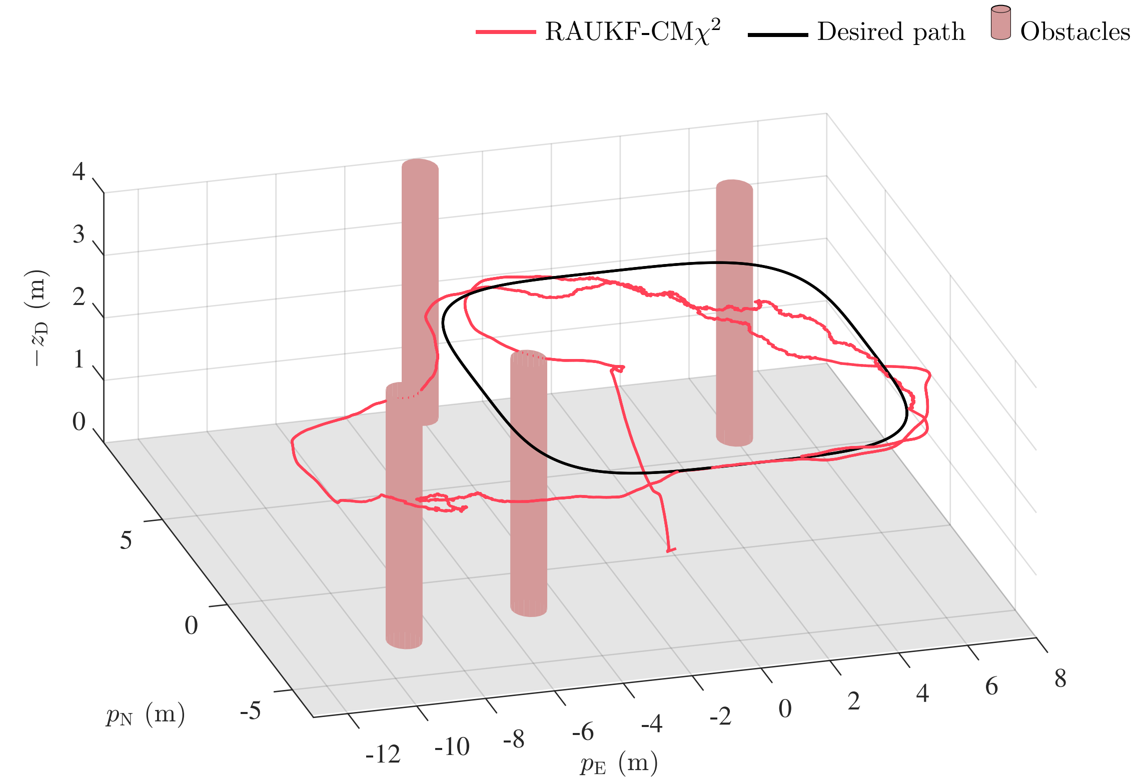
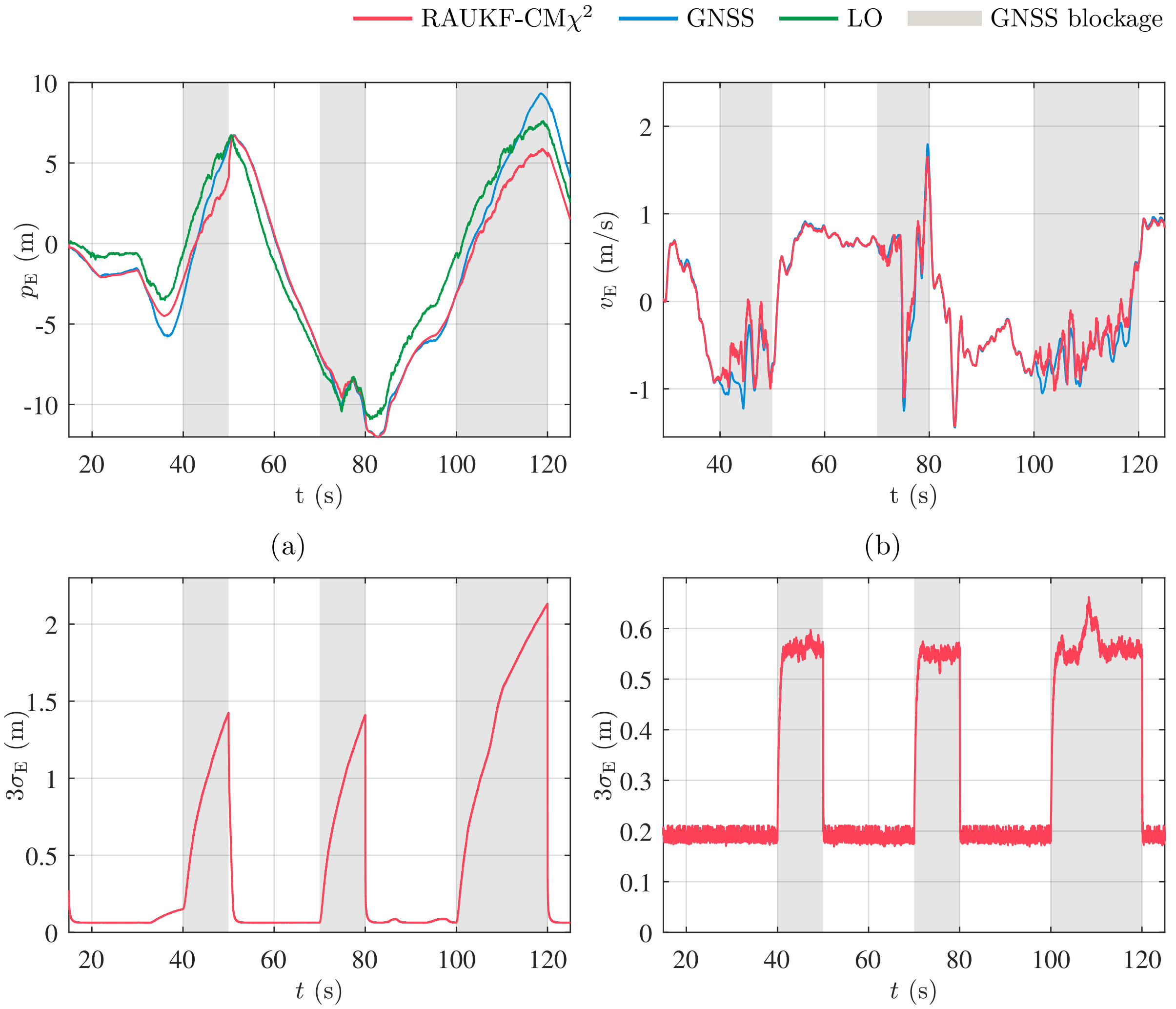


FIGURE 7.15: Three-dimensional path estimated by RAUKF-CMχ^2 (red), for the target surveillance path represented by the black curve. Observe that the estimated path deviates from the target one due to the presence of trees, represented by brown cylinders. It is important to mention that, since the diameter and position of the trees are not known, those cylinders are coarse approximations of the actual forest, used for visualization only.

7.7 Conclusions

This chapter presented a localization solution for aerial vehicles navigating inside a forest. The proposed solution is based on combining LiDAR-based odometry, GNSS and AHRS information using a robust adaptive sensor fusion algorithm based on UKF. The LiDAR-based odometry relies on the fact that trees are easy identified in a laser scan. Tree detection highly increases the efficiency of the method, allowing it to run in simple onboard hardware. The performance of the proposed algorithm was shown through two experiments using actual data.

In the first experiment, data was collected during a flight of the aerial vehicle in manual mode. Then, offline tests with artificially abnormal measurements were performed on a personal computer. The proposed RAUKF-CMχ^2 is able to deal with some abnormal



(a)

(b)

(c)

(d)

FIGURE 7.16: (a) and (b) Position and velocity in the East direction; (c) and (d) three standard deviation $\sigma_\mathrm{E} = \sqrt{P^{xx}_{k,ii}}$ of position and velocity estimates in the East direction.

information, both in relative and global measurements.

In the second experiment, the proposed fusion algorithm runs online on an onboard computer embedded in the vehicle. The navigation solution was used to feed the path planner and controller in a fully closed-loop system. The experiment was performed in a sparse forest, where the GNSS signal did not suffer substantial blockages. Thus, we artificially dropout the GNSS signal in order to test the behavior of the system. The experiment performed in a sparse forest environment shows that the fusion algorithm is able to adequately combine global and relative measurements in real-world problems.

Chapter 8

Conclusion and Future Work

8.1 Concluding Remarks

State estimators are fundamental to the navigation of autonomous vehicles. Among the state estimation techniques, Kalman-based state estimators are normally the standard solution adopted. Although the estimation theory is well established, standard algorithms, such as the UKF, are not able to adequately handle real-world issues, such as time-varying uncertainty of measurements. The state estimation problem is even more challenging in the presence of constrained states, such as unit quaternions.

In this thesis, we address the problem of state estimation of vehicles performing 6-degrees of freedom maneuvers, in which attitude is represented by unit quaternions and the available measurements have time-varying uncertainty. We propose state estimators that are able to mitigate the influence of measurement failures by using robust adaptive filtering techniques. In our approach, the uncertainty in the measurements is modeled by random noise with unknown statistic parameters, namely, mean and covariance. The main concern of this work was the state estimation problem for vehicles operating in environments with damaged GNSS measurements. However, the proposed algorithms can also be used for other types of systems. In addition, the proposed algorithms were tested in real-world problems.

In Chapter 3, the unscented Kalman filter algorithm is revised. To handle the time-varying uncertainty, adaptive approaches based on covariance matching and interacting multiple model approach are investigated. The first adaptive approach, although simple, can estimate online the measurement error covariance. Two methods to detect and mitigate the influence of outliers were also reviewed. One is based on the chi-square test, and a second is based on the Hampel identifier. In addition, the unscented Rauch-Tung-Striebel smoother was also presented. Based on the reviewed approaches, two new robust adaptive unscented Kalman filters were proposed, namely robust adaptive unscented Kalman filter by using covariance matching and Hampel identifier - RAUKF-CMH and robust adaptive unscented Kalman filter by using covariance matching and χ^2-test - RAUKF-CMχ^2. These algorithms represent the first theoretical contribution of this thesis (Section 3.6).

The localization of a vehicle is a particular state estimation problem. It is usual to parameterize vehicle attitude by unit quaternions. This parameterization brings a particular challenge to the state estimation problem, once the unit norm of quaternions must be maintained. In Chapter 4, probabilistic tools for unit quaternions are reviewed. In order to handle the time-varying uncertainty and also ensure the unit norm constraint, RAUKF-CMl, IMM, and URTS smoother, investigated in Chapter 3, are extended. The new class of estimation algorithms are called quaternion-based robust adaptive unscented Kalman filter by using covariance matching and Hampel identifier-QRAUKF-CMH, quaternion-based robust adaptive unscented Kalman filter by using covariance matching and χ^2-test-QRAUKF-CMχ^2, quaternion-based interacting multiple model filter-QIMM, and quaternion-based unscented Rauch-Tung-Stribel smoother-QURTS. This set of new algorithms represent the second theoretical contribution of this thesis (Section 4.3).

In Chapter 5, the localization and flight path reconstruction problem of aerial vehicles performing aggressive maneuvers is addressed. The motivation for this case study comes from aerial races, in which the aerobatic maneuvers and television signals may jam the GNSS signal. In this work, experimental flight data collected during different sessions of the Red Bull Air Race are used to test the performance of adaptive algorithms, without the robust part, developed in Section 4.3. The results show that the adaptive algorithms are able to detect and mitigate time-varying uncertainty in the GNSS measurements. In addition, the QRTS smoother improves the flight path estimation, mitigating the abrupt changes in the position estimates and also decreasing the uncertainty of estimates.

In Chapter 6, we investigate the attitude estimation with MARG sensors. In this study case, five abnormal measurement scenarios are tested with actual data collected from a commercial MARG sensor. The results show a superior performance of the proposed algorithms faced with linear acceleration disturbs, and abrupt and slow varying changes in the magnetic field.

In Chapter 7, the problem of state estimation of a micro aerial vehicle flying in a forest environment is presented. In this case, GNSS and laser-based odometer measurements with time-varying uncertainty are investigated. This study case presents two experiments. First, collected data is used to test the performance of RAUKF-CMχ^2 faced with two types of abnormal measurements, outliers in the laser-based odometer and slow drift in GNSS measurements. Then, in the second experiment, the proposed estimator algorithm is used in a complete navigation solution, where its estimates are used to feed a path planner and velocity control. The results show that the proposed algorithm is able to adequately merge heterogeneous data.

Portions of this work have been published in national and international conferences and a journal:

1. CHIELLA, A. C. B.; MACHADO, H. N.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. .*GNSS/LiDAR-based navigation of an aerial robot in sparse forests.* Sensors 2019, 19, 4061.

2. CHIELLA, A. C. B.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. Quaternion-based robust attitude estimation using an adaptive unscented Kalman filter. Sensors 2019, 19, 2372.

3. CHIELLA, A. C. B.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. *State estimation for aerial vehicles in forest environments.* The 2019 International Conference on Unmanned Aircraft Systems (ICUAS). Atlanta. p. 882-890.

4. CHIELLA, A. C. B.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. *Robust attitude estimation using an adaptive unscented Kalman filter.* The International Conference on Robotics and Automation (ICRA) de 2019. Montreal, Canada. p. 7748-7754

5. CHIELLA, A. C. B.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. . *Filtro de Kalman unscented adaptativo para estimação de atitude representada por quatérnios.* Congresso Brasileiro de Automática, 2018, João Pessoa. Anais do XXII Congresso Brasileiro de Automática (CBA'18). p. 1-8.

6. CHIELLA, A. C. B.; TEIXEIRA, B. O. S. ; PEREIRA, G. A. S. . *Localization of aerial vehicles performing aggressive maneuvers in GPS-unreliable environments using vision information and adaptive state estimators.* Congresso Brasileiro de Automática, 2016, Vitória. Anais do XXI Congresso Brasileiro de Automática (CBA'16), 2016. p. 1-6.

8.2 Future Work

Important theoretical investigations need to be addressed in future work. In this work, *ad-hoc* techniques were proposed to modify the UKF in order to handle failures in measurement data. The proposed solutions do not explicitly consider the non-Gaussian distribution for filtering approach. More recent approaches, such as [Sarkka and Nummenmaa, 2009], use the variational Bayesian approach to approximate the posterior distribution. These algorithms also consider Euclidean state spaces and thus need to be extended in order to estimate states pertaining to non-Euclidean spaces.

As illustrated in the attitude estimation case study in Section 6, the adaptive algorithms directly influence the bias estimates. In addition, in our work, only the measurement covariance estimation is addressed in the adaptive estimation methods. A question that remains is about the observability of states and parameters that are estimated. This needs to be addressed in order to investigate the influence of adaptive approaches in the state estimate and the possibility to simultaneously estimate both the process and observation covariance matrices. In addition, it is also expected that the inclusion of more sensors leads to better observability. However, the effect of individual sensors in the observability properties remains unexplored. The information about observability properties could be beneficial to a motion planning system, in order to perform risk assessment of a task.

In practical implementations of filtering algorithms, measurements from different sensors can arrive out-of-sequence due to different sensor processing times and transmission latencies. This means that an incoming measurement can be older than the current state. A classical form to try to overcome this problem is by saving a buffer with all necessary variable, and use the measurement in its respective time instant, then to reprocess the filter. This procedure is sometimes very time-consuming. Thus, investigations, in order to combine adaptive algorithms with techniques to treat out of sequence measurements need to be addressed.

As the reliability of autonomous systems improves, deployment of multiple vehicles in more complex missions, in a collaborative way, becomes feasible. In this regard, heterogeneous systems, such as ground and aerial vehicles, could share information, improving the navigation performance. This is also a challenging problem that needs to be addressed in future investigations.

References

[Abuhashim et al., 2010] Abuhashim, T. S., Abdel-Hafez, M. F., and Al-Jarrah, M. A. (2010). Building a robust integrity monitoring algorithm for a low cost GPS-aided-INS system. *International Journal of Control, Automation and Systems*, 8(5):1108–1122. 18

[Ackerson and Fu, 1970] Ackerson, G. and Fu, K. (1970). On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1):10–17. 133

[Arasaratnam and Haykin, 2009] Arasaratnam, I. and Haykin, S. (2009). Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269. 24

[Armesto et al., 2004] Armesto, L., Chroust, S., Vincze, M., and Tornero, J. (2004). Multi-rate fusion with vision and inertial sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 193–199. 13

[Arroyo-Mora et al., 2019] Arroyo-Mora, J. P., Kalacska, M., Inamdar, D., Soffer, R., Lucanus, O., Gorman, J., Naprstek, T., Schaaf, E. S., Ifimov, G., Elmer, K., et al. (2019). Implementation of a uav–hyperspectral pushbroom imager for ecological monitoring. *Drones*, 3(1):12. 87

[Bachrach et al., 2010] Bachrach, A., Winter, A. D., He, R., Hemann, G., Prentice, S., and Roy, N. (2010). Range - robust autonomous navigation in GPS-denied environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1096–1097. 2, 3

[Baldwin et al., 2009] Baldwin, G., Mahony, R., and Trumpf, J. (2009). A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2237–2242. 13

[Bar-Shalom et al., 2004] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software.* John Wiley & Sons. 17, 30, 33, 125, 132, 133

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. 13

[Beard and McLain, 2012] Beard, R. W. and McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice.* Princeton University Press. 59, 61

[Belbachir et al., 2015] Belbachir, A., Escareno, J., Rubio, E., and Sossa, H. (2015). Preliminary results on uav-based forest fire localization based on decisional navigation. In *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 377–382. IEEE. 87

[Bin et al., 2014] Bin, X., Yang, L., Xu, Z., Meihui, C., and Fu, W. (2014). Hovering control of a nano quadrotor unmanned aerial vehicle using optical flow. In *33rd Chinese Control Conference (CCC)*, pages 8259–8264. 13

[Blom and Bar-Shalom, 1988] Blom, H. A. P. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783. 30, 133

[Cadena et al., 2016] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332. 17

[Candy, 2011] Candy, J. V. (2011). *Bayesian signal processing: Classical, modern and particle filtering methods*, volume 54. John Wiley & Sons. 125, 127

[Carvalho et al., 1997] Carvalho, H., Moral, P. D., Monin, A., and Salut, G. (1997). Optimal nonlinear filtering in GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):835–850. 12

[Chambers et al., 2011] Chambers, A., Achar, S., Nuske, S., Rehder, J., Kitt, B., Chamberlain, L., Haines, J., Scherer, S., and Singh, S. (2011). Perception for a river mapping robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 227–234. 2

[Chambers et al., 2014] Chambers, A., Scherer, S., Yoder, L., Jain, S., Nuske, S., and Singh, S. (2014). Robust multi-sensor fusion for micro aerial vehicle navigation in GPS-degraded/denied environments. In *American Control Conference (ACC), 2014*, pages 1892–1899. 13, 15, 17, 88, 89, 96

[Chauhan and Gao, 2017] Chauhan, S. V. S. and Gao, G. X. (2017). Joint GPS and vision estimation using an adaptive filter. In *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION)*, pages 808–812. 18

[Chiella et al., 2019a] Chiella, A. C. B., Machado, H. N., Teixeira, B. O. S., and Pereira, G. A. S. (2019a). GNSS/LiDAR-based navigation of an aerial robot in sparse forests. *Sensors*, 19(19). 6

[Chiella et al., 2016] Chiella, A. C. B., Teixeira, B. O. S., and Pereira, G. A. S. (2016). Localization of Aerial Vehicles Performing Agressive Maneuvers in GPS-unreliable Environments Using Vision information and Adaptive State Estimators. In *Anais do XXI Congresso Brasileiro de Automática, Vitória, ES*. 6, 13, 15

[Chiella et al., 2018] Chiella, A. C. B., Teixeira, B. O. S., and Pereira, G. A. S. (2018). Filtro de Kalman unscented adaptativo para estimação de atitude representada por quatérnios. In *Congresso Brasileiro de Automática, João Pessoa*. 6

[Chiella et al., 2019b] Chiella, A. C. B., Teixeira, B. O. S., and Pereira, G. A. S. (2019b). Quaternion-based robust attitude estimation using an adaptive unscented Kalman filter. *Sensors*, 19(10). 6, 72

[Chiella et al., 2019c] Chiella, A. C. B., Teixeira, B. O. S., and Pereira, G. A. S. (2019c). Robust attitude estimation using an adaptive unscented Kalman filter. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7748–7754. xix, 6, 42, 72, 90

[Chiella et al., 2019d] Chiella, A. C. B., Teixeira, B. O. S., and Pereira, G. A. S. (2019d). State estimation for aerial vehicles in forest environments. In *2019 International Conference on Unmaned Aircraft Systems*, pages 882–890. 6, 14

[Chirikjian and Kyatkin, 2001] Chirikjian, G. S. and Kyatkin, A. B. (2001). *Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups*. CRC press. 42

[Choukroun et al., 2006] Choukroun, D., Bar-Itzhack, I. Y., and Oshman, Y. (2006). Novel quaternion Kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):174–190. 15

[Costanzi et al., 2016] Costanzi, R., Fanelli, F., Monni, N., Ridolfi, A., and Allotta, B. (2016). An attitude estimation algorithm for mobile robots under unknown magnetic disturbances. *IEEE/ASME Transactions on Mechatronics*, 21(4):1900–1911. 71

[Crassidis, 2006] Crassidis, J. L. (2006). Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756. 15

[Crassidis and Markley, 2003] Crassidis, J. L. and Markley, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of Guidance Control and Dynamics*, 26(4):536–542. 15, 59

[Cui et al., 2013] Cui, J., Wang, F., Dong, X., Yao, K. A. Z., Chen, B. M., and Lee, T. H. (2013). Landmark extraction and state estimation for UAV operation in forest. In *Proceedings of the 32nd Chinese Control Conference*, pages 5210–5215. 88

[Cui et al., 2014] Cui, J. Q., Lai, S., Dong, X., Liu, P., Chen, B. M., and Lee, T. H. (2014). Autonomous navigation of UAV in forest. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 726–733. 14, 88, 89

[Da, 1994] Da, R. (1994). Failure detection of dynamical systems with the state chi-square test. *Journal of guidance, control, and dynamics*, 17(2):271–277. 17

[Davies and Gather, 1993] Davies, L. and Gather, U. (1993). The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792. 34

[De Mendonça et al., 2007] De Mendonça, C. B., Hemerly, E. M., and Sandoval Góes, L. C. (2007). Adaptive stochastic filtering for online aircraft flight path reconstruction. *Journal of aircraft*, 44(5):1546–1558. 12

[Eras-Herrera et al., 2019] Eras-Herrera, W. Y., Mesquita, A. R., and Teixeira, B. O. S. (2019). Equality-constrained state estimation for hybrid systems. *IET Control Theory Applications*, 13(13):2018–2028. 31

[Euston et al., 2008] Euston, M., Coote, P., Mahony, R., Kim, J., and Hamel, T. (2008). A complementary filter for attitude estimation of a fixed-wing UAV. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 340–345. 16

[Fang and Scherer, 2014] Fang, Z. and Scherer, S. (2014). Experimental study of odometry estimation methods using RGB-D cameras. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 680–687. 13

[Fraundorfer et al., 2012] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., and Pollefeys, M. (2012). Vision-based autonomous mapping and exploration using a quadrotor MAV. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4557–4564. 13

[Freitas et al., 2010] Freitas, G., Gleizer, G., Lizarralde, F., Hsu, L., and dos Reis, N. R. S. (2010). Kinematic reconfigurability control for an environmental mobile robot operating in the amazon rain forest. *Journal of Field Robotics*, 27(2):197–216. 87

[Garcia and Solanas, 2004] Garcia, M. A. and Solanas, A. (2004). 3D simultaneous localization and modeling from stereo vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 847–853 Vol.1. 13

[Gordon et al., 1993] Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings, Pt. F*, volume 140, pages 107–113. 1

[Groves, 2013] Groves, P. D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House. 9, 12, 14, 60, 71

[Gui and de Ruiter, 2017] Gui, H. and de Ruiter, A. H. (2017). Quaternion invariant extended Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 41(4):863–878. 71

[Gustafsson, 2008] Gustafsson, F. (2008). *Adaptive filtering and change detection*, volume 1. Wiley New York. 16, 17, 33

[Hajiyev and Soken, 2013] Hajiyev, C. and Soken, H. E. (2013). Robust adaptive Kalman filter for estimation of UAV dynamics in the presence of sensor/actuator faults. *Aerospace Science and Technology*, 28(1):376 – 383. 18

[Hajiyev and Soken, 2016] Hajiyev, C. and Soken, H. E. (2016). Fault tolerant estimation of UAV dynamics via robust adaptive Kalman filter. In *Complex Systems*, pages 369–394. Springer. 18

[Hampel, 1971] Hampel, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, pages 1887–1896. 33

[Hanson, 2006] Hanson, A. J. (2006). *Visualizing quaternions*. Morgan Kaufmann Publishers. 39, 40, 41

[Hauberg et al., 2013] Hauberg, S., Lauze, F., and Pedersen, K. S. (2013). Unscented Kalman filtering on Riemannian manifolds. *Journal of mathematical imaging and vision*, 46(1):103–120. 45

[Haykin et al., 2001] Haykin, S. S. et al. (2001). *Kalman filtering and neural networks*. Wiley Online Library. 25

[Hide et al., 2004] Hide, C., Moore, T., and Smith, M. (2004). Adaptive Kalman filtering algorithms for integrating GPS and low cost INS. In *Position Location and Navigation Symposium*, pages 227–233. 18

[Hu et al., 2003] Hu, C., Chen, W., Chen, Y., and Liu, D. (2003). Adaptive Kalman filtering for vehicle navigation. *Journal of Global Positioning Systems*, 2(01):42–47. 18

[Hu et al., 2015] Hu, G., Gao, S., and Zhong, Y. (2015). A derivative UKF for tightly coupled INS/GPS integrated navigation. *ISATransactions*, 56:135 – 144. 12

[Hwang et al., 2006] Hwang, I., Balakrishnan, H., and Tomlin, C. (2006). State estimation for hybrid systems: applications to aircraft tracking. *IEE Proceedings - Control Theory and Applications*, 153(5):556–566. 132

[Isermann, 2005] Isermann, R. (2005). Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29(1):71 – 85. 17

[Jang and Liccardo, 2007] Jang, J. S. and Liccardo, D. (2007). Small UAV automation using MEMS. *IEEE Aerospace and Electronic Systems Magazine*, 22(5):30–34. 71

[Jazwinski, 1970] Jazwinski, A. H. (1970). *Stochastic processes and filtering theory*. Inc. reprinted by Dover, 2007. 2

[Jin et al., 2014] Jin, S., Cardellach, E., and Xie, F. (2014). *GNSS remote sensing*. Springer. 16

[Jones and Soatto, 2011] Jones, E. S. and Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430. 13

[Julier and Uhlmann, 2004] Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422. 2, 23

[Jutila et al., 2007] Jutila, J., Kannas, K., and Visala, A. (2007). Tree measurement in forest by 2D laser scanning. In *2007 International Symposium on Computational Intelligence in Robotics and Automation*, pages 491–496. 88, 91

[Jwo et al., 2013] Jwo, D., Chung, F., and Yu, K. (2013). GPS/INS Integration Accuracy enhancement using the Interacting Multiple Model nonlinear filters. *Journal of applied research and technology*, 11(4):496–509. 18

[Jwo and Tseng, 2011] Jwo, D. J. and Tseng, C. H. (2011). Interacting multiple model nonlinear filter design for ultra-tightly coupled integrated navigation. In *8th Asian Control Conference (ASCC)*, pages 1294–1299. 18

[Jwo et al., 2009] Jwo, D. J., Tseng, C. H., Chen, M. Y., and Cho, T. S. (2009). *Adaptive and nonlinear Kalman filtering for GPS navigation processing.* INTECH Open Access Publisher. 18

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45. 1

[Karma et al., 2015] Karma, S., Zorba, E., Pallis, G., Statheropoulos, G., Balta, I., Mikedi, K., Vamvakari, J., Pappa, A., Chalaris, M., Xanthopoulos, G., et al. (2015). Use of unmanned vehicles in search and rescue operations in forest fires: Advantages and limitations observed in a field trial. *International journal of disaster risk reduction*, 13:307–312. 87

[Kendoul, 2012] Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378. 1, 9

[Kendoul et al., 2009] Kendoul, F., Fantoni, I., and Nonami, K. (2009). Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robotics and Autonomous Systems*, 57(6):591–602. 13

[Kendoul et al., 2010] Kendoul, F., Yu, Z., and Nonami, K. (2010). Guidance and non-linear control system for autonomous flight of minirotorcraft unmanned aerial vehicles. *Journal of Field Robotics*, 27(3):311–334. 12

[Kim et al., 2017] Kim, S., Haschke, R., and Ritter, H. (2017). Gaussian mixture model for 3-Dof orientations. *Robotics and Autonomous Systems*, 87:28–37. 42

[Kottas et al., 2013] Kottas, D. G., Hesch, J. A., Bowman, S. L., and Roumeliotis, S. I. (2013). On the consistency of vision-aided inertial navigation. In *Experimental Robotics*, pages 303–317. 3

[Kraft, 2003] Kraft, E. (2003). A quaternion-based unscented Kalman filter for orientation tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 1, pages 47–54. 42, 43, 61

[Krisanski et al., 2018] Krisanski, S., Del Perugia, B., Taskhiri, M. S., and Turner, P. (2018). Below-canopy UAS photogrammetry for stem measurement in radiata pine plantation. In *Remote Sensing for Agriculture, Ecosystems, and Hydrology XX*, volume 10783, page 1078309. International Society for Optics and Photonics. 87

[Lefferts et al., 1982] Lefferts, E. J., Markley, F. L., and Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429. 15

[Leutenegger and Siegwart, 2012] Leutenegger, S. and Siegwart, R. (2012). A low-cost and fail-safe inertial navigation system for airplanes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 612–618. 11, 55

[Li et al., 2016] Li, K., Hu, B., Chang, L., and Li, Y. (2016). Comparison of direct navigation mode and indirect navigation mode for integrated SINS/GPS. *Transactions of the Institute of Measurement and Control*, 38(1):3–13. 12, 15

[Li and Bar-Shalom, 1994] Li, X. R. and Bar-Shalom, Y. (1994). A recursive multiple model approach to noise identification. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):671–684. 30, 133

[Lu and Milios, 1997] Lu, F. and Milios, E. (1997). Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic systems*, 18(3):249–275. 93

[Lu et al., 2015] Lu, P., Van Eykeren, L., and van Kampen, E. (2015). Selective-reinitialization multiple-model adaptive estimation for fault detection and diagnosis. *Journal of Guidance, Control, and Dynamics*, 38(8). 18

[Madgwick, 2010] Madgwick, S. O. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Technical report, University of Bristol (UK). 16, 72, 74, 77

[Madgwick et al., 2011] Madgwick, S. O. H., Harrison, A. J. L., and Vaidyanathan, R. (2011). Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE Intl. Conf. on Rehabilitation Robotics*, pages 1–7. 16

[Mallick et al., 2012] Mallick, M., Krishnamurthy, V., and Vo, B.-N. (2012). *Integrated tracking, classification, and sensor management: theory and applications*. John Wiley & Sons. 133

[Markley et al., 2007] Markley, F. L., Cheng, Y., Crassidis, J. L., and Oshman, Y. (2007). Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197. 43

[Maybeck, 1982] Maybeck, P. S. (1982). *Stochastic models, estimation, and control*, volume 3. Academic press. 14

[Mehra, 1972] Mehra, R. (1972). Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 17(5):693–698. 29

[Menegaz, 2016] Menegaz, H. M. T. (2016). *Unscented Kalman filtering on Euclidean and Riemannian manifolds*. PhD thesis, University of Brasília. 39, 45

[Menegaz et al., 2015] Menegaz, H. M. T., Ishihara, J. Y., Borges, G. A., and Vargas, A. N. (2015). A systematization of the unscented Kalman filter theory. *IEEE Transactions on Automatic Control*, 60(10):2583–2598. 25

[Menegaz et al., 2019] Menegaz, H. M. T., Ishihara, J. Y., and Kussaba, H. T. M. (2019). Unscented Kalman filters for Riemannian state-space systems. *IEEE Transactions on Automatic Control*, 64(4):1487–1502. 15, 45

[Menegaz and Ishihara, 2018] Menegaz, H. T. and Ishihara, J. Y. (2018). Unscented and square-root unscented Kalman filters for quaternionic systems. *International Journal of Robust and Nonlinear Control*, pages 1–28. 15, 40, 48

[Michel et al., 2017] Michel, T., Genevès, P., Fourati, H., and Layaïda, N. (2017). On attitude estimation with smartphones. In *IEEE Intl. Conference on Pervasive Computing and Communications*, pages 267–275. 71

[Moghaddamjoo and Kirlin, 1989] Moghaddamjoo, A. and Kirlin, R. L. (1989). Robust adaptive Kalman filtering with unknown inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8):1166–1175. 17

[Morita et al., 2018] Morita, M., Nishida, T., Arita, Y., Shige-eda, M., di Maria, E., Gallone, R., and Giannoccaro, N. I. (2018). Development of robot for 3D measurement of forest environment. *Journal of Robotics and Mechatronics*, 30(1):145–154. 87

[Mulder et al., 1999] Mulder, J., Chu, Q., Sridhar, J., Breeman, J., and Laban, M. (1999). Non-linear aircraft flight path reconstruction review and new advances. *Progress in Aerospace Sciences*, 35(7):673–726. 12, 58

[Nonami et al., 2010] Nonami, K., Kendoul, F., Suzuki, S., Wang, W., and Nakazawa, D. (2010). *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer. 1

[Pearson, 2002] Pearson, R. K. (2002). Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 10(1):55–63. 34

[Pennec, 2006] Pennec, X. (2006). Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127. 43

[Pereira et al., 2008] Pereira, G. A. S., Iscold, P., and Torres, L. A. B. (2008). Airplane attitude estimation using computer vision: simple method and actual experiments. *Electronics Letters*, 44(22):1303–1304. 71

[Rosen and Medvedev, 2013] Rosen, O. and Medvedev, A. (2013). Efficient parallel implementation of state estimation algorithms on multicore platforms. *IEEE Transactions on Control Systems Technology*, 21(1):107–120. 15

[Roumeliotis and Burdick, 2002] Roumeliotis, S. I. and Burdick, J. W. (2002). Stochastic cloning: A generalized framework for processing relative state measurements. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1788–1795. 94, 96

[Roumeliotis et al., 1998] Roumeliotis, S. I., Sukhatme, G. S., and Bekey, G. A. (1998). Sensor fault detection and identification in a mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1383–1388. 89

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. 91

[Särkkä, 2008] Särkkä, S. (2008). Unscented Rauch–Tung–Striebel smoother. *IEEE Transactions on Automatic Control*, 53(3):845–849. 35

[Särkkä, 2013] Särkkä, S. (2013). *Bayesian filtering and smoothing*, volume 3. Cambridge University Press. 4, 25, 125

[Särkkä and Hartikainen, 2013] Särkkä, S. and Hartikainen, J. (2013). Non-linear noise adaptive Kalman filtering via variational Bayes. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE. 18

[Sarkka and Nummenmaa, 2009] Sarkka, S. and Nummenmaa, A. (2009). Recursive noise adaptive Kalman filtering by variational Bayesian approximations. *IEEE Transactions on Automatic Control*, 54(3):596–600. 18, 110

[Schmid et al., 2013] Schmid, K., Tomic, T., Ruess, F., Hirschmuller, H., and Suppa, M. (2013). Stereo vision based indoor/outdoor navigation for flying robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3955–3962. 3

[Schultz et al., 2016] Schultz, A., Gilabert, R., Bharadwaj, A., de Haag, M. U., and Zhu, Z. (2016). A navigation and mapping method for UAS during under-the-canopy forest operations. In *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 739–746. 14

[Shao et al., 2019] Shao, T., Duan, Z., Ge, Q., and Liu, H. (2019). Recursive performance ranking of Kalman filter with mismatched noise covariances. *IET Control Theory Applications*, 13(4):459–466. 17

[Shen et al., 2011] Shen, S., Michael, N., and Kumar, V. (2011). Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 20–25. 3, 14, 88, 89

[Shen et al., 2013a] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. (2013a). Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *Robotics: Science and Systems*, volume 1. Citeseer. 13

[Shen et al., 2013b] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. (2013b). Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1758–1764. 13

[Shen et al., 2014] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. (2014). Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981. 2, 15, 88, 89

[Shin and El-Sheimy, 2004] Shin, E.-H. and El-Sheimy, N. (2004). An unscented Kalman filter for in-motion alignment of low-cost imus. In *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*, pages 273–279. 14, 15

[Siciliano and Khatib, 2016] Siciliano, B. and Khatib, O. (2016). *Springer Handbook of Robotics*. Springer Science & Business Media. 2

[Siegwart et al., 2011] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press. 10, 12

[Simon, 2006] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons. 25, 59, 132

[Sipos, 2008] Sipos, B. J. (2008). Application of the manifold-constrained unscented Kalman filter. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 30–43. 15, 42, 43, 48

[Snyder, 1987] Snyder, J. P. (1987). *Map projections–A working manual*, volume 1395. US Government Printing Office. 62

[Song et al., 2016] Song, Y., Nuske, S., and Scherer, S. (2016). A multi-sensor fusion MAV state estimation from long-range stereo, IMU, GPS and barometric sensors. *Sensors*, 17(1):11. 2, 13, 89

[Strelow and Singh, 2004] Strelow, D. and Singh, S. (2004). Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195. 88

[Sundvall and Jensfelt, 2006] Sundvall, P. and Jensfelt, P. (2006). Fault detection for mobile robots using redundant positioning systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3781–3786. 17

[Tandeo et al., 2018] Tandeo, P., Ailliot, P., Bocquet, M., Carrassi, A., Miyoshi, T., Pulido, M., and Zhen, Y. (2018). Joint estimation of model and observation error covariance matrices in data assimilation: a review. *arXiv:1807.11221*. 17

[Tang et al., 2015] Tang, J., Chen, Y., Niu, X., Wang, L., Chen, L., Liu, J., Shi, C., and Hyyppä, J. (2015). Lidar scan matching aided inertial navigation system in GNSS-denied environments. *Sensors*, 15(7):16710–16728. 88

[Teixeira et al., 2014] Teixeira, B. O. S., Castro, W. S., Teixeira, A. F., and Aguirre, L. A. (2014). Data-driven soft sensor of downhole pressure for a gas-lift oil well. *Control Engineering Practice*, 22:34–43. 32

[Teixeira et al., 2011] Teixeira, B. O. S., Tôrres, L. a. B., Iscold, P., and Aguirre, L. a. (2011). Flight path reconstruction - A comparison of nonlinear Kalman filter and smoother algorithms. *Aerospace Science and Technology*, 15(1):60–71. 12, 58

[Terra et al., 2014] Terra, G. S., Tôrres, L. A., and Teixeira, B. O. (2014). Attitude estimation using a two-step unscented approach with gyro bias estimation and acceleration correction. *Anais do XX Congresso Brasileiro de Automática*. 61

[Titterton and Weston, 2004] Titterton, D. and Weston, J. L. (2004). *Strapdown inertial navigation technology*, volume 17. IET. 9, 14

[Tomic et al., 2012] Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & Automation Magazine*, 19(3):46–56. 2, 14

[Tubman et al., 2016] Tubman, R., Potgieter, J., and Arif, K. M. (2016). Efficient robotic SLAM by fusion of RatSLAM and RGBD-SLAM. In *2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 1–6. 13

[Tugnait, 1982] Tugnait, J. K. (1982). Detection and estimation for abruptly changing systems. *Automatica*, 18(5):607 – 615. 133

[Upadhyay et al., 1993] Upadhyay, T. N., Cotterill, S., and Deaton, A. W. (1993). Autonomous GPS/INS navigation experiment for space transfer vehicle. *IEEE Transactions on Aerospace and Electronic Systems*, 29(3):772–785. 12

[Valenti et al., 2015] Valenti, R. G., Dryanovski, I., and Xiao, J. (2015). Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs. *Sensors*, 15(8):19302–19330. 16, 73

[Valenti et al., 2016] Valenti, R. G., Dryanovski, I., and Xiao, J. (2016). A linear Kalman filter for MARG orientation estimation using the algebraic quaternion algorithm. *IEEE Transactions on Instrumentation and Measurement*, 65(2):467–481. 73

[Vartiainen et al., 2014] Vartiainen, P., Bragge, T., Arokoski, J. P., and Karjalainen, P. A. (2014). Nonlinear state-space modeling of human motion using 2-D marker observations. *IEEE Transactions on Biomedical Engineering*, 61(7):2167–2178. 61, 71

[Weiss et al., 2012] Weiss, S., Achtelik, M. W., Lynen, S., Chli, M., and Siegwart, R. (2012). Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 957–964. 13

[Weiss, 2012] Weiss, S. M. (2012). *Vision Based Navigation for Micro Helicopters*. PhD thesis, Eidgenössische Technische Hochschule (ETH). 3, 13

[Xiong et al., 2006] Xiong, K., Zhang, H., and Chan, C. (2006). Performance evaluation of UKF-based nonlinear filtering. *Automatica*, 42(2):261–270. 27

[Zhang et al., 2014] Zhang, B., Jiao, Y., Ma, Z., Li, Y., and Zhu, J. (2014). An efficient image matching method using speed up robust features. In *2014 IEEE International Conference on Mechatronics and Automation*, pages 553–558. 13

[Zhou et al., 2011] Zhou, J., Yang, Y., Zhang, J., Edwan, E., Loffeld, O., and Knedlik, S. (2011). Tightly-coupled INS/GPS using quaternion-based unscented Kalman filter. In *AIAA Guidance, Navigation, and Control Conference*, page 6488. 12, 14

[Zug et al., 2012] Zug, S., Dietrich, A., and Kaiser, J. (2012). Fault-handling in networked sensor systems. *Fault Diagnosis in Robotic and Industrial Systems*. 17

Appendix A

Gaussian Bayesian Filtering and Smoothing

In this chapter, we derive the Bayesian filtering equations, which are the general equation for computing filtering solutions for both linear and nonlinear state-space models. We also present the Kalman filter and smoother equations which give the solution to the linear Gaussian Bayesian filtering and smoothing problems. The formulation of the multiple model approach, which assumes hybrid systems, is also presented. The contents of this appendix are mainly based on the works of [Candy, 2011], [Särkkä, 2013], and [Bar-Shalom et al., 2004].

A.1 Recursive Bayesian Approach

The recursive Bayesian processes consider that the state space model, defined in Section 3.1, is a probabilistic state space model, which consists of a sequence of conditional probability distributions:

$$x_k \sim \rho\left(x_k | x_{k-1}, u_{k-1}\right), \tag{A.1}$$

$$y_k \sim \rho\left(y_k | x_k\right), \tag{A.2}$$

where $k \in \mathbb{N}$ denotes the discrete time, $x_k \in \mathbb{R}^{n_x}$ is the system state, $u_{k-1} \in \mathbb{R}^{n_u}$ and $y_k \in \mathbb{R}^{n_y}$ are the known input and output measurements, respectively, $\rho\left(x_k | x_{k-1}, u_{k-1}\right)$ is the dynamic model describing the evolution of state conditioned to the previous state x_{k-1} and input u_{k-1}, and $\rho\left(y_k | x_k\right)$ is the measurement model, which is the distribution of measurements conditioned to the state.

It is assume that the probabilistic state space model(A.1)-(A.2) is constrained to the following Markov properties:

Property A.1.1. Markov property of states

The state sequence $x_{0:k_f} = \{x_0, ..., x_{k_f}\}$ form a Markov sequence, where $k_f \in \mathbb{N}$. The state x_k and the whole future states $x_{k+1:k_f} = \{x_{k+1}, ..., x_{k_f}\}$, given the state x_{k-1} and the input u_{k-1}, are independent of anything that has happened before the time step $k-1$. Also, all the past states $x_{k-1:0}$ are independent of the future states $x_{k+1:k_f}$ and inputs $u_{k+1:k_f}$, given the present state x_k and input u_k.

Property A.1.2. Conditional independence of measurements

Given the current state x_k, the current measurement y_k is conditionally independent of the whole measurement $y_{1:k-1}$ and state $x_{0:k-1}$ histories.

Due the property of conditional independence of measurements and the Markov property of states we have that $\rho(y_k|y_{1:k-1}, x_{1:k}) = \rho(y_k|x_k)$ and $\rho(x_k|x_{1:k-1}, u_{0:k-1}, y_{1:k-1}) = \rho(x_k|x_{k-1}, u_{k-1})$. Then, for a given instant $N \in \daleth$, the *joint prior distribution of states and the joint likelihood of the measurements are, respectively*

$$\rho\left(x_{0:k_f}|u_{0:k_f-1}\right) = \rho\left(x_0\right)\prod_{k=1}^{k_f}\rho\left(x_k|x_{k-1}, u_{k-1}\right), \tag{A.3}$$

$$\rho\left(y_{1:k_f}|x_{0:k_f}\right) = \prod_{k=1}^{k_f}\rho\left(y_k|x_k\right). \tag{A.4}$$

By using the Bayes theorem, the *a posteriori* PDF can be written as

$$\rho(x_{0:k_f}|y_{1:k_f}, u_{0:k_f-1}) = \frac{\rho(y_{1:k_f}|x_{0:k_f}, u_{0:k_f-1})\rho(x_{0:k_f}|u_{0:k_f-1})}{\rho(y_{1:k_f}|u_{0:k_f-1})} \tag{A.5}$$

$$= \frac{\rho(y_{1:k_f}|x_{0:k_f})\rho(x_{0:k_f}|u_{0:k_f-1})}{\rho(y_{1:k_f})}. \tag{A.6}$$

As new observations arrived, the number of computations of the Equation (A.6) increases, which is not feasible for real-time applications. Then, the *a posteriori* PDF is marginalized in order to have algorithms with constant number of computations per time step. Two marginal distributions are considered here: the marginal distribution of the current state x_k given the current and previous output measurements $y_{1:k}$, $\rho\left(x_k|u_{0:k-1}, y_{1:k}\right)$, referred to as Bayesian filter, and; the marginal distribution of state x_k given a certain interval $k_f > k$ of the measurements $y_{1:k_f}$, $\rho\left(x_k|y_{1:k_f}\right)$, referred to as Bayesian smoother. For simplicity, from now on, we disregard the input measurements $u_{0:k-1}$.

The Bayes Filter

The purpose of filtering is to compute the *marginal a posteriori* PDF of state $\rho\left(x_k|y_{1:k}\right)$. By applying the Bayes theorem to the problem defined in Section 3.1, one tries to obtain recursive equations for the *a posteriori* probability density of state x_k, based on available observations $y_{1:k}$ [1]

$$\begin{aligned}\rho(x_k|y_{1:k}) &= \frac{\rho(y_{1:k}|x_k)\rho(x_k)}{\rho(y_{1:k})}\\ &= \frac{\rho(y_k,y_{1:k-1}|x_k)\rho(x_k)}{\rho(y_k,y_{1:k-1})}\\ &= \frac{\rho(y_k|y_{1:k-1},x_k)\rho(y_{1:k-1}|x_k)\rho(x_k)}{\rho(y_k|y_{1:k-1})\rho(y_{1:k-1})} &\text{(A.7)}\\ &= \frac{\rho(y_k|x_k)\rho(x_k|y_{1:k-1})}{\rho(y_k|y_{1:k-1})}. &\text{(A.8)}\end{aligned}$$

The recursive estimation process, commonly known as Bayes filter comprises two essential steps [Candy, 2011]: *prediction* or *forecast* and *update* or *data-assimilation*. Figure A.1 illustrates the two steps sequential Bayesian filter, wherein the *a posteriori* density function is given by

$$\rho(x_k,|y_{1:k}) = \mathcal{W}_{k|k-1}\rho(x_k|y_{1:k-1}), \tag{A.9}$$

where the *weight* $\mathcal{W}$ is defined by

$$\mathcal{W}_{k|k-1} = \frac{\rho(y_k|x_k)}{\rho(y_k|y_{1:k-1})}. \tag{A.10}$$

The *forecast* step of the Bayes filter consists in the calculation of the prior distribution $\rho(x_k|y_{1:k-1})$. Thus, given the posterior distribution $\rho(x_{k-1}|y_{1:k-1})$ from the previous time $k-1$, the prior distribution can be obtained by solving the Chapman-Kolmogorov equation

[1] The following conditional probability rules are applied: $\rho(A,B|C) = \rho(A|B,C)\rho(B|C)$ and $\rho(A,B) = \rho(A|B)\rho(B)$.

given by

$$\rho(x_k|y_{1:k-1}) = \int \rho(x_k|x_{k-1})\rho(x_{k-1}|y_{1:k-1})\mathrm{d}x_{k-1}, \tag{A.11}$$

where $\rho(x_k|x_{k-1})$ is given by the state transition equation (1.1) and the know statistics of the process noise.

The *data-assimilation* step consists in calculating the posterior PDF $\rho(x_k, |y_{1:k})$. In this step, the most recent measurement y_k can be incorporated, reducing the uncertainty of the estimate $\hat{x}_k$. Therefore, the term $\rho(y_k|x_k)$ denotes the measurement probability, which is also commonly known as *likelihood* function. Finally, the *evidence* (normalizing factor) can be calculated by the law of total probability, given by

$$\rho(y_k|y_{1:k-1}) = \int \rho(y_k|x_k)\rho(x_k|y_{1:k-1})\mathrm{d}x_k, \tag{A.12}$$

which is a function of measurements $y_{1:k}$. Note that the current measurement y_k is assimilated into the prior PDF $\rho(x_k|y_{1:k-1})$ obtained during the *forecast* step.

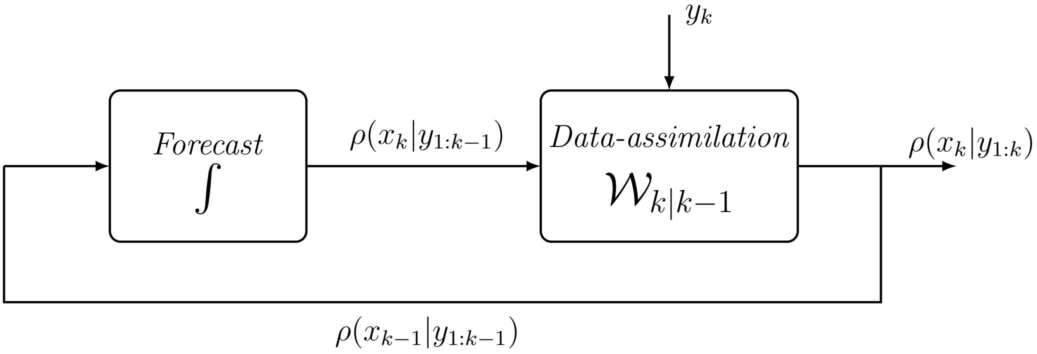


FIGURE A.1: Sequential Bayesian filtering.

For most applications, the analytic solutions of (A.11) and (A.12) are intractable. Nevertheless, by admitting some restrictions on the utilized PDFs, e.g. if they can be represented by a finite number of parameters such as mean and covariance, the so-called *parametric filters* or *Gaussian filters* implementations of Bayes filter can be performed. Kalman-based filters are example of practical Bayesian algorithms (Appendix A.2).

A.2 The Kalman Filter

Let the probabilistic state space model be a linear Gaussian model, then

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}q_{k-1} \tag{A.13}$$

$$y_k = C_k x_k + r_k, \tag{A.14}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state, $y_k \in \mathbb{R}^{n_y}$ is the output measurement, $q_{k-1} \in \mathbb{R}^{n_q}$ is the process noise, and $r_{k-1} \in \mathbb{R}^{n_y}$ is the output measurement noise. Assume that, for all $k \geq 1$ the matrices $A_{k-1} \in \mathbb{R}^{n_x \times n_x}$, $B_{k-1} \in \mathbb{R}^{n_x \times n_u}$, $G_{k-1} \in \mathbb{R}^{n_x \times n_q}$, $C_k \in \mathbb{R}^{n_y \times n_x}$, the inputs u_{k-1}, and the outputs y_k are known. Also, the process noise q_{k-1} and the output measurement noise r_k are assumed to be Gaussian, zero-mean, and mutually independent, with covariance matrices $Q_{k-1} \in \mathbb{R}^{n_q \times n_q}$ and $R_k \in \mathbb{R}^{n_y \times n_y}$, respectively, that is

$$\rho\left(q_{k-1}\right) = \mathcal{N}\left([0]_{n_q \times 1}, Q_{k-1}\right) \tag{A.15}$$

$$\rho\left(r_k\right) = \mathcal{N}\left([0]_{n_y \times 1}, R_k\right). \tag{A.16}$$

The initial state vector $x_0 \in \mathbb{R}^{n_x}$ is assumed to be Gaussian with initial estimate $\hat{x}_0 \triangleq \mathrm{E}\left[x_0\right]$ and and error-covariance $P_0^{xx} \triangleq \mathrm{E}\left[\left(x_0 - \hat{x}_0\right)\left(x_0 - \hat{x}_0\right)^T\right]$, that is,

$$\rho\left(x_0\right) = \mathcal{N}\left(\hat{x}_0, P_0^{xx}\right). \tag{A.17}$$

Under the linear-Gaussian assumption, the prior distribution, the probabilistic process and measurement models are:

$$\rho\left(x_{k-1}|y_{1:k-1}\right) = \mathcal{N}\left(\hat{x}_{k-1}, P_{k-1}^{xx}\right), \tag{A.18}$$

$$\rho\left(x_k \,|\, x_{k-1}\right) = \mathcal{N}\left(A_{k-1}x_{k-1} + B_{k-1}u_{k-1}, Q_{k-1}\right), \tag{A.19}$$

$$\rho\left(y_k \,|\, x_k\right) = \mathcal{N}\left(C_k x_k, R_k\right), \tag{A.20}$$

where $\hat{x}_{k-1}$ is the *prior* estimate with *prior* covariance matrix P_{k-1}^{xx}, From (A.11), the *forecast* PDF is given by:

$$\rho\left(x_k|y_{1:k-1}\right) = \mathcal{N}\left(\hat{x}_{k|k-1}, P^{xx}_{k|k-1}\right), \tag{A.21}$$

$$\rho\left(y_k \mid y_{1:k-1}\right) = \mathcal{N}\left(\hat{y}_{k|k-1}, P^{yy}_{k|k-1}\right), \tag{A.22}$$

where $\hat{x}_{k|k-1}$ is the *forecast* estimate, $\hat{y}_{k|k-1}$ is the *forecast* output estimate, and the *forecast* error covariance $P^{xx}_{k|k-1}$, innovation covariance $P^{yy}_{k|k-1}$, and cross covariance $P^{xy}_{k|k-1}$ are defined by:

$$P^{xx}_{k|k-1} \triangleq \mathrm{E}\left[\left(x_k - \hat{x}_{k|k-1}\right)\left(x_k - \hat{x}_{k|k-1}\right)^T\right], \tag{A.23}$$

$$P^{yy}_{k|k-1} \triangleq \mathrm{E}\left[\left(y_k - \hat{y}_{k|k-1}\right)\left(y_k - \hat{y}_{k|k-1}\right)^T\right], \tag{A.24}$$

$$P^{xy}_{k|k-1} \triangleq \mathrm{E}\left[\left(x_k - \hat{x}_{k|k-1}\right)\left(y_k - \hat{y}_{k|k-1}\right)^T\right]. \tag{A.25}$$

From (A.9), the solution to the state estimate problem for linear systems is given in the *data-assimilation* step by

$$\rho(x_k, |y_{1:k}) = \mathcal{N}\left(\hat{x}_k, P^{xx}_k\right), \tag{A.26}$$

where $\hat{x}_k$ is the state estimate and P^{xx}_k is the error covariance matrix of estimate, defined as

$$P^{xx}_k \triangleq \mathrm{E}\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right]. \tag{A.27}$$

The Kalman Filter Algorithm

Thus, for the linear system (A.13)-(A.14), the *forecast* step of Kalman filter is given by:

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1}, \tag{A.28}$$

$$P^{xx}_{k|k-1} = A_{k-1}P^{xx}_{k-1}A^T_{k-1} + G_{k-1}Q_{k-1}G^T_{k-1}. \tag{A.29}$$

The *data-assimilation* step is given by:

$$\hat{y}_{k|k-1} = C_k \hat{x}_{k|k-1}, \tag{A.30}$$

$$P^{yy}_{k|k-1} = C_k P^{xx}_{k|k-1} C_k^T + R_k, \tag{A.31}$$

$$P^{xy}_{k|k-1} = P^{xx}_{k|k-1} C_k^T \tag{A.32}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \left(y_k - \hat{y}_{k|k-1} \right), \tag{A.33}$$

$$P^{xx}_k = P^{xx}_{k|k-1} - K_k P^{yy}_{k|k-1} K_k^T, \tag{A.34}$$

where the Kalman gain $K_k \in \mathbb{R}^{n_x \times n_y}$ is defined as

$$K_k \triangleq P^{xy}_{k|k-1} \left(P^{yy}_{k|k-1} \right)^{-1}. \tag{A.35}$$

A.3 The Raunch-Tung-Striebel Smoother Algorithm

First, a filtering algorithm (Appendix A.2) is run, yielding the filtering estimates $\hat{x}_k$ and P^{xx}_k for the time interval $k = k_0 \dots k_f$. Then, the two steps smoother algorithm runs backwards from $k_f - 1$ to k_0. The first step, also called *forecast* step given by:

$$\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k, \tag{A.36}$$

$$P^{xx}_{k+1k} = A_k P^{xx}_k A_k^T + G_k Q_k G_k^T. \tag{A.37}$$

The second step, called *smoothing* step is given by:

$$K^s_k = P^{xx}_k A_k^T \left(P^{xx}_{k+1|k} \right)^{-1}, \tag{A.38}$$

$$\hat{x}^s_k = \hat{x}_k + K^s_k \left(x^s_{k+1} - \hat{x}_{k+1|k} \right), \tag{A.39}$$

$$P^{xx,s}_k = P^{xx}_k + K^s_k \left(P^{xx,s}_{k+1} - P^{xx}_{k+1|k} \right) \left(K^s_k \right)^T, \tag{A.40}$$

with $x^s_{k_f} = x_{k_f}$ and $P^{xx,s}_{k_f} = P^{xx}_{k_f}$.

A.4 Multiple Model Approach

The multiple model (MM) approach assumes that the system can be suitably described by a finite number of models. Such systems are called hybrid, since they have both analog states whose dynamics are modeled by difference stochastic equations and digital states (modes) which are usually modeled by a finite Markov chain [Hwang et al., 2006].

The correct model selection depends on the type of output data available. The innovation of different filters for each model are used as functions that reflect the likelihood that estimates of each of different models is the correct one.

The Bayesian framework is used to obtain the corresponding *a posteriori* probabilities [Bar-Shalom et al., 2004, Simon, 2006]. Thus, assume that the measurement equation (1.2) is rewritten with model-dependent measurement noise[2],

$$y_k = h(x_k, k) + v_k^j, \;\; \forall\, j \in \mathcal{M}, \tag{A.41}$$

where j denote a discrete set of r models, denoted by $\mathcal{M} = \{M^1, ..., M^r\}$. Note that the difference between (1.2) and (A.41) is the mode-dependent noise, so the stochastic hybrid system has different measurement noise statistics from model to model.

Thus, let the sth sequence of models up to and including time $k-1$ be denoted by

$$M_{k-1,s} = \{M_{1,s}^i, ..., M_{k-1,s}^i\}, \;\; s = 1, ..., r^{k-1} \text{ and } \forall\, i \in \mathcal{M}, \tag{A.42}$$

where the model i is in effect at time $k-1$. Hence, the lth sequence, with model j in effect at time k is denoted by

$$M_{k,l} = \{M_{k-1,s}, \, M_k^j\}, \;\; l = 1, ..., r^k \text{ and } \forall\, j \in \mathcal{M}. \tag{A.43}$$

It is assumed that the switchings are governed by a Markov process with known model transition probabilities

$$\Pr\{M_k^j | M_{k-1}^i\} \triangleq \Pi_{ij}, \;\; \forall\, i, j \in \mathcal{M}, \tag{A.44}$$

[2] In a general formulation of MM approach, multiple models can represent the process and observation equations.

where Π_{ij} is the transition probability from model i to model j, and $\Pr\{\cdot\}$ denotes the probability of an event.

Using the total probability theorem with respect to set (A.43), the conditional PDF of state x_k is obtained by a Gaussian mixture with exponential increasing number of terms

$$\rho(x_k|y_{1:k}) = \sum_{l=1}^{r^k} \rho(x_k|M_{k,l}, y_{1:k})\Pr\{M_{k,l}|y_{1:k}\}, \tag{A.45}$$

where $\rho(x_k|M_{k,l}, y_{1:k})$ is obtained by constructing an estimator as in Section 3.2 based upon (1.1) and (1.2), whose system parameters are matched to the Markov chain sequence (A.43) [Tugnait, 1982]. The *a posteriori* probability of lth sequence given the measurement data up to k, is obtained by using the Bayes' rule as

$$\begin{aligned}
\gamma_{k,l} &\triangleq \Pr\{M_{k,l}|y_{1:k}\} \\
&= \Pr\{M_{k,l}|y_k, y_{1:k-1}\} \\
&= \frac{\rho(y_k|M_{k,l}, y_{1:k-1})\Pr\{M_{k,l}|y_{1:k-1}\}}{\rho(y_k|y_{1:k-1})} \\
&= \frac{1}{c}\rho(y_k|M_{k,l}, y_{1:k-1})\Pr\{M_{k,l}|M_{k-1,s}, y_{1:k-1}\}\Pr\{M_{k-1,s}|y_{1:k-1}\} \\
&= \frac{1}{c}\rho(y_k|M_{k,l}, y_{1:k-1})\Pi_{ij}\gamma_{k-1,s},
\end{aligned} \tag{A.46}$$

where c is a normalization constant, i is the last model of sequence s and j is the last model of sequence l.

Notice that the computational requirements grow exponentially with time, $\mathcal{O}(r^k)$. Unfortunately, the optimal estimator has no practical feasibility. Thus, suboptimal estimation schemes with fixed computational cost are proposed. A simple way to obtain a suboptimal algorithm is to keep a fixed number of models sequences, discard the rest and normalize the probabilities to sum up to unity [Bar-Shalom et al., 2004]. Among the major suboptimal algorithms, are the Generalized Pseudo-Bayesian (GPB) [Ackerson and Fu, 1970] and the Interacting Multiple Model (IMM) [Blom and Bar-Shalom, 1988, Li and Bar-Shalom, 1994]. The last one, has become one of the most popular maneuvering target tracking approaches [Mallick et al., 2012].

The IMM approach considers all the possible models in the last two sampling time, i.e $k-1$ and k. Therefore, (A.45) is rewritten as

$$\rho(x_k|y_{1:k}) = \sum_{j=1}^{r} \rho(x_k|M_k^j, y_{1:k})\Pr\{M_k^j|y_{1:k}\}. \tag{A.47}$$

The mode-conditioned *a posteriori* PDF of the state x_k is given by

$$\begin{aligned}\rho(x_k|M_k^j, y_{1:k}) &= \rho(x_k|M_k^j, y_k, y_{1:k-1}) \\ &= \frac{\rho(y_k|M_k^j, x_k)}{\rho(y_k|M_k^j, y_{1:k-1})}\rho(x_k|M_k^j, y_{1:k-1}).\end{aligned} \tag{A.48}$$

Then, it applies the total probability theorem in the mode-conditioned *a priori*

$$\begin{aligned}\rho(x_k|M_k^j, y_{1:k-1}) &= \sum_{i=1}^{r} \rho(x_k|M_k^j, M_{k-1}^i, y_{1:k-1})\Pr\{M_{k-1}^i|M_k^j, y_{1:k-1}\} \\ &\approx \sum_{i=1}^{r} \rho(x_k|M_k^j, M_{k-1}^i, \hat{x}_{k-1|k-1}^i, P_{k-1|k-1}^{xx,i})w_{k-1|k-1}^{i|j},\end{aligned} \tag{A.49}$$

where $w_{k-1|k-1}^{i|j} = \Pr\{M_{k-1}^i|M_k^j, y_{1:k-1}\}$ are the mixing probabilities. Equation (A.49) is a mixture with a different weight $w_{k-1|k-1}^{i|j}$ for each model M_k^j. This equation is the core of the IMM algorithm and reflects the fact that the past can be summarized by r mode-conditioned means and covariances. The mixture is assumed to be a mixture of Gaussian PDFs and then approximated by moment matching.

The model probability is defined as

$$\begin{aligned}\gamma_k^j &\triangleq Pr\{M_k^j|y_{1:k}\} \\ &= Pr\{M_k^j|y_k, y_{1:k-1}\} \\ &= \frac{1}{c}\rho(y_k|M_k^j, y_{1:k-1})Pr\{M_k^j|y_{1:k-1}\} && \text{(A.50)} \\ &= \frac{1}{c}\Lambda_k^j \sum_{i=1}^{r} Pr\{M_k^j|M_{k-1}^i, y_{1:k-1}\}\gamma_{k-1}^i && \text{(A.51)}\end{aligned}$$

where $\Lambda_k^j = \rho(y_k|M_k^j, y_{1:k-1})$, $\gamma_{k-1}^i = Pr\{M_{k-1}^i|y_{1:k-1}\}$ and c is a normalization constant, given by

$$c \triangleq \sum_{j=1}^{r} \Lambda_k^j \sum_{i=1}^{r} \Pi_{ij}\gamma_{k-1}^i. \tag{A.52}$$