

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Signal, Image, Acoustique et Optimisation

Présentée et soutenue par :

M. AMANI BEN AFIA

le vendredi 20 octobre 2017

Titre :

Développement d'algorithmes GNSS/INS/SLAM pour la navigation en milieux contraints

---

École doctorale :

Mathématiques, informatique et télécommunications (MITT)

Unité de recherche :

École Nationale de l'Aviation Civile (ENAC)

Directeurs de Thèse :

M. CHRISTOPHE MACABIAU

MME ANNE-CHRISTINE ESCHER

Rapporteurs :

M. JOHN RAQUET, AIR FORCE INSTITUTE OF TECHNOLOGY

M. SILVERE BONNABEL, MinesParisTech

Membres du jury :

M. JOHN RAQUET, AIR FORCE INSTITUTE OF TECHNOLOGY, Président

M. CHRISTOPHE MACABIAU, ÉCOLE NATIONALE DE L'AVIATION CIVILE, Membre

MME ANNE-CHRISTINE ESCHER, ÉCOLE NATIONALE DE L'AVIATION CIVILE, Membre

M. STEPHAN WEISS, ALPEN-ADRIA-UNIVERSITAT, Membre

M. MARC POLLINA, M3 Systems, Membre

A tout ceux qui étaient là pour moi pendant ma thèse

Acknowledgments

I would like to deeply acknowledge all the people who helped me both directly and behind the scenes to complete my Ph.D.

Tout d'abord, je remercie mon directeur de thèse Dr. Christophe Macabiau pour m'avoir donné la chance d'effectuer cette thèse au sein du groupe SigNav et pour avoir pu me trouver du temps, avec toute la charge de travail qu'il a. Ses idées inspirantes, sa confiance en moi, sa rigueur, ainsi que son sens de l'humour ont rendu chaque discussion avec lui passionnante et instructive.

Je remercie également ma co-encadrante de thèse Dr. Anne-Christine Escher pour son support constant, aussi bien sur le plan professionnel qu'humain. Ses conseils et ses encadrements ont été d'une grande valeur pour l'aboutissement de ce travail.

Je tiens à remercier également Dr. Antoine Drouin pour son aide précieuse, son soutien et sa patience. Sa contribution a été très importante pour l'aboutissement de cette thèse.

I gratefully thank Prof. John Raquet for reviewing my research and for being the president of this thesis defense jury. I would also like to thank him for his valuable comments and his encouragements. Although it took me 2 weeks to investigate on who mentioned "Dr." on my office nameplate after the thesis defense, I was so honored to know that "the wanted man" was Prof. Raquet. I would also like to thank Prof. Silvère Bonnabel for finding the time to review my thesis, despite his heavy workload. I also gratefully thank Prof. Stephan Weiss for accepting to be a member of this thesis defense jury and for the interesting discussions we have on the multi-sensor fusion topics. His works on this field have been very inspiring and instructive for me. Many thanks to Mr. Marc Pollina for accepting to be a member of the thesis defense.

Je remercie également tous les membres de Signav pour leur amitié et pour toutes les discussions intéressantes qu'on a eu. Je remercie Myriam (pour son amitié et son support sur tous les plans), Ikhlas et Sara (pour la bonne ambiance à la fin de ma thèse et pour leur soutien pendant les moments difficiles), Philippe (avec qui j'ai partagé mon bureau pendant 3 ans et avec qui je pouvais passer des heures à discuter du filtre de Kalman ou de l'hybridation), Jérémy et Paul (qui étaient toujours là pour répondre à mes questions), Enik (mon coéquipier des heures tardives de fin thèse), Giuseppe (mon co-bureau à la fin de sa thèse), Jean-Baptiste (pongiste à ses heures perdues), Alizé (ma compagne du voyage à Hawaï), Leslie et Lina (pour m'avoir bien accueilli au labo au début de ma thèse), Daniel (pour m'avoir encadré tout au au début), Hélène (pour sa bonne humeur contagieuse), Quentin, Jade, Johan, Capucine, Roberto, Auryn, Xiao et Anne-Marie (à qui je souhaite un bon courage pour finir leurs thèses). Je remercie également tous les enseignants chercheurs des groupes SigNav et EMA pour leur disponibilité à chaque fois que je les sollicite. Je remercie aussi Coco, Cathy, Hasna et Soumia pour leur gentillesse infinie et leur encouragement.

Un grand merci à mes amis et proches qui étaient toujours là pour m'apporter leur soutien inconditionnel et leur encouragement. Les années de thèse avec vous sont beaucoup plus agréables.

Finalement, je tiens à remercier toute ma famille, sans qui, je n'aurais jamais été ce que je suis maintenant. En particulier j'offre cette thèse à mes parents qui, même loin de moi, étaient toujours là pour me soutenir et me pousser vers l'avant, comme ils l'ont toujours fait depuis mon plus jeune âge. Mon frère et sa petite famille qui sont toujours là à mes côtés malgré la distance. Et pour finir, je remercie Daly, pour tout ce qu'il a fait pour moi, surtout pendant la période de rédaction.

Abstract

The requirements in terms of navigation accuracy, integrity, continuity and availability are increasing for land vehicles especially with the development of automated and autonomous vehicle applications. This type of applications requires a navigation system not only capable of providing continuously an accurate and reliable navigation solution but also having a reasonable cost.

In the last decades, the *Global Navigation Satellite System* (GNSS) has been the most widely used navigation system especially with the receivers decreasing cost over the years. However, despite of its capability to provide absolute navigation information with long time accuracy, this system suffers from problems related to signal propagation especially in urban environments where buildings, trees and other structures hinder the reception of GNSS signals and degrade their quality. This can result in significant positioning error exceeding in some cases a kilometer. Many techniques are proposed in the literature to mitigate these problems and improve the GNSS accuracy. Unfortunately, all these techniques have limitations.

A possible way to overcome these problems is to fuse "good" GNSS measurements with other sensors having complementary advantages. In fact, by exploiting the complementarity of sensors, hybridization algorithms can improve the navigation solution compared to solutions provided by each stand-alone sensor. Generally, the most widely implemented hybridization algorithms for land vehicles fuse GNSS measurements with inertial and/or odometric data. Thereby, these *Dead-Reckoning* (DR) sensors ensure the system continuity when GNSS information is unavailable and improve the system performance when GNSS signals are corrupted, and in turn, the GNSS limits the drift of the DR solution if it is available. However, the performance achieved by this hybridization depends thoroughly on the quality of the DR sensor used especially when GNSS signals are degraded or unavailable.

Since the targeted application is cost-sensitive and low-cost sensors should be used, the common solution of fusing GNSS with inertial and/or odometric data mentioned previously cannot provide the high performance required by the application. For this reason, the use of vision-based navigation techniques to provide additional information is proposed in this thesis to improve the navigation performance. In fact, cameras have become, recently, an attractive positioning sensor with the development of *Visual Odometry* (VO) and *Simultaneous Localization and Mapping* (SLAM) techniques, capable of providing accurate navigation solution while having reasonable cost. In addition, visual navigation solutions have a good quality in textured environments where GNSS is likely to have bad performance. This interesting aspect makes visual techniques promising to achieve good performance in difficult GNSS environments.

Therefore, this Ph.D. thesis focuses on developing a multi-sensor fusion architecture integrating visual information and particularly studies the contribution of fusing this information with the previously mentioned sensors to improve the vision-free navigation system performance.

The dissertation begins with a description of the context and challenges of this work, an identification of the navigation requirements and a presentation of the thesis objective and contribution.

Then a description of the commonly used land navigation sensors is performed. The system principle and the measurement models of each sensor are provided. The state of the art of the integrated systems is addressed and the integration techniques and strategies are explained.

The third chapter reviews the existing vision-based navigation techniques and highlights their challenges especially in outdoor large-scale environments. A particular interest is brought to ORB-SLAM open-source software that has been used in this study to provide the visual information fed to the developed system.

The fourth chapter provides a detailed explanation of the proposed multi-sensor navigation system. The choices made to design this system are justified and the contribution of each sensor is described mathematically. A special attention is given to the observability analysis of the parameters the developed system should be able to estimate.

The fifth chapter addresses the integrity monitoring problem and discusses the integrity challenges faced, on one hand, with the visual information and its processing with the SLAM technique, and on the other hand with the hybridized structure whose integrity study is not a trivial task.

The sixth chapter provides the results and the performance analysis of the proposed solution. The presented results are based on real datasets collected in different types of environments. The contribution of visual information is particularly highlighted.

Finally, the last chapter concludes on the works done during the Ph.D. and proposes some perspectives that could be considered for future works.

Résumé

Les exigences en termes de précision, intégrité, continuité et disponibilité de la navigation terrestre sont de plus en plus strictes, surtout depuis le développement des véhicules autonomes. Ce type d'applications nécessite un système de navigation non seulement capable de fournir une solution de navigation précise et fiable, mais aussi ayant un coût raisonnable.

Durant les dernières décennies, les systèmes de navigation par satellites (GNSS) ont été les plus utilisés pour la navigation, surtout avec la baisse continue des coûts des récepteurs. Cependant, malgré sa capacité à fournir des informations de navigation absolue avec une bonne précision dans des milieux dégagés, ce système souffre de problèmes liés à la propagation du signal dans des milieux contraints tels que le milieu urbain où des bâtiments, des arbres et d'autres structures empêchent la bonne réception des signaux GNSS et dégradent leur qualité. Ceci peut entraîner une erreur de position importante qui peut dépasser le kilomètre dans certains cas. Beaucoup de techniques ont été proposées dans la littérature pour remédier à ces problèmes et améliorer les performances GNSS. Toutefois, ces techniques présentent des limitations.

Une façon possible de surmonter ces problèmes est de fusionner les "bonnes" mesures GNSS avec les mesures d'autres capteurs ayant des caractéristiques complémentaires. En effet, en exploitant la complémentarité des capteurs, les algorithmes d'hybridation peuvent améliorer la solution de navigation par rapport aux solutions fournies par chacun des capteurs considéré individuellement. Les algorithmes d'hybridation les plus largement mis en œuvre pour les véhicules terrestres fusionnent les mesures GNSS avec des données inertielles et / ou odométriques. Ainsi, ces capteurs de navigation à l'estime (NE) assurent la continuité du système lorsque les informations GNSS ne sont pas disponibles et améliorent les performances du système lorsque les signaux GNSS sont altérés et, de son côté, le GNSS limite la dérive de la solution de NE s'il est disponible. Cependant, les performances obtenues par cette hybridation dépendent énormément de la qualité du capteur de NE utilisé, surtout lorsque les signaux GNSS sont dégradés ou indisponibles.

Étant donné que l'application visée doit respecter le critère d'un coût raisonnable, les solutions classiques de fusion du GNSS avec les données inertielles et / ou odométriques bas coût ne peuvent pas fournir le niveau de performance requis par l'application. C'est pourquoi l'utilisation de techniques de navigation basées sur la vision pour fournir des informations supplémentaires est proposée dans cette thèse pour améliorer les performances de navigation. En effet, les caméras deviennent un capteur de positionnement de plus en plus attrayant avec le développement de techniques d'Odométrie Visuelle et de Localisation et Cartographie simultanées (SLAM), capable de fournir une solution de navigation précise tout en ayant des coûts raisonnables. En outre, les solutions de navigation visuelle ont une bonne qualité dans les environnements texturés où le GNSS risque d'avoir de mauvaises performances.

Par conséquent, cette thèse se concentre sur le développement d'une architecture de fusion multi-capteurs bas coût intégrant l'information visuelle tout en étant facilement embarquable.

Une attention particulière est accordée à l'étude de la contribution l'information visuelle pour améliorer les performances du système de navigation sans vision.

La thèse commence par une description du contexte de ce travail, une identification des exigences de navigation et une présentation de l'objectif et de la contribution de cette thèse.

Ensuite, une description des systèmes de navigation terrestre couramment utilisés est effectuée. Le principe du système et les modèles de mesures de chaque capteur sont fournis. Un état de l'art des systèmes d'hybridation est présenté et les techniques et stratégies d'hybridation sont expliquées.

Le troisième chapitre passe en revue les techniques existantes de navigation basée sur la vision et souligne leurs défis, en particulier dans les environnements extérieurs à grande échelle. Un intérêt particulier est apporté au logiciel open source ORB-SLAM qui a été utilisé dans cette étude pour fournir l'information visuelle en entrée de la solution développée.

Le quatrième chapitre fournit une explication détaillée de l'architecture de navigation multi-capteurs proposé. Les choix réalisés pour concevoir cette architecture sont justifiés et la contribution de chaque capteur est décrite mathématiquement. Une attention particulière est accordée à l'analyse d'observabilité des paramètres que le système développé devrait pouvoir estimer.

Le cinquième chapitre traite du problème de contrôle de l'intégrité des signaux utilisés et analyse les moyens de sélectionner les mesures des capteurs afin de conserver celles intègres et exclure celles susceptibles d'être affectés par de grandes erreurs.

Le sixième chapitre présente les résultats et l'analyse des performances de la solution proposée. Les résultats présentés sont obtenus suite à la réalisation de collectes de données réelles dans différents types d'environnements. La contribution de l'information visuelle est particulièrement soulignée.

Enfin, le dernier chapitre conclut sur les travaux réalisés au cours de la thèse et propose des perspectives qui pourraient être envisagées pour des travaux futurs.

Table of Contents

Acknowledgments ..... **iv**
Abstract ..... **vi**
Résumé ..... **viii**
Table of Contents ..... **x**
List of Figures ..... **xiv**
List of Tables ..... **xviii**
List of Acronyms ..... **xx**
Notations ..... **xxiv**
1 Introduction **1**
1.1 Motivation and challenges 1
1.2 Thesis objectives and contribution 4
1.3 Thesis outline 4
2 Navigation Systems for Ground Vehicles **7**
2.1 GNSS 8
2.1.1 System principle 8
2.1.2 GNSS Measurement models 10
2.1.3 GNSS Navigation solution 14
2.2 Inertial navigation 16
2.2.1 System principle 16
2.2.2 Inertial measurement errors and models 17
2.2.3 Inertial mechanization 19
2.3 Odometry 22
2.3.1 Wheel Speed Sensors 22
2.3.2 Steering angle sensors 22
2.3.3 Ackermann steering model 23
2.3.4 Odometric measurement models 25
2.4 Sensor complementarity and integrated navigation 26
2.4.1 Integration techniques and strategies 27
2.4.2 State of the art of existing ground vehicle integrated navigation based on INS, GNSS and/or odometry and limits of the existing solutions 29

2.5 Conclusion 30
3 Vision-based Navigation for Ground Vehicle Navigation in Outdoor Environments 31
3.1 State of the art of vision-based navigation techniques 32
3.1.1 Classification of vision-based navigation techniques 32
3.1.2 Vision-based Navigation fundamentals 34
3.2 ORB SLAM 46
3.2.1 Selection of ORB-SLAM 46
3.2.2 ORB-SLAM characteristics 46
3.3 Challenges of vision-based navigation in outdoor environments 48
3.4 VSLAM measurement model 49
3.5 Conclusion 51
4 Proposed Hybridization Architecture for the Improvement of Ground Vehicle Navigation in Urban Environments 53
4.1 System architecture 54
4.2 Extended Kalman Filter 57
4.3 State vector 59
4.4 State transition model 62
4.4.1 Continuous-time model 62
4.4.2 Discrete-time model 65
4.5 Observation model 65
4.5.1 GNSS observation model 65
4.5.2 SLAM observation model 67
4.5.3 WSS and NHC observation model 68
4.5.4 ZARU observation model 69
4.6 Observability analysis 69
4.6.1 Observability analysis using Lie derivatives 70
4.6.2 Nonlinear system formulation 71
4.6.3 Measurement formulation 72
4.6.4 Observability analysis 73
4.7 Conclusion 78
5 Integrity Monitoring Challenges 79
5.1 Integrity concept 80
5.1.1 Integrity definition and requirements 80

5.1.2 Fault-free mode and failure modes ... 82
5.1.3 Sensor fault-free and failure mode identification ... 82
5.1.4 Integrity risk allocation ... 87
5.1.5 Failure mode characterization ... 89
5.2 Integrity monitoring function ... 90
5.2.1 Fault detection and exclusion ... 90
5.2.2 State of the art of sequential FDE ... 91
5.3 Case study of integrity challenges ... 93
5.3.1 Integrity requirements challenges ... 93
5.3.2 Fusion architecture integrity challenges ... 94
5.4 Conclusion ... 97
6 Results and Performance Analysis ... 99
6.1 Experimental procedure ... 100
6.1.1 Equipment ... 100
6.1.2 Description of the trajectory ... 102
6.2 Sensors and filter performance analysis ... 105
6.2.1 GPS solution ... 105
6.2.2 SLAM solution ... 110
6.2.3 Hybridized navigation solution ... 118
6.3 Conclusion ... 129
7 Conclusions and Proposals on Further Research ... 131
7.1 Thesis conclusions ... 131
7.2 Perspective and future works ... 134
Bibliography ... 137
A Coordinate frames and Transformations ... 149
A.1 Coordinate frames ... 149
A.2 Frame Transformations ... 152
B Rotation mathematics ... 155
B.1 Quaternions ... 155
B.2 Rotation matrix ... 158
B.3 Euler angles ... 159
B.4 Relationship between the different representations ... 162
B.5 Rotation error model ... 166

C Calculation of the Kalman Filter observation matrices 169

C.1 Perturbation model 169

C.2 Observation matrices 171

D Proofs of observability analysis 177

D.1 Step 1: Observability analysis of the WSS scale factor kw 177

D.2 Step 2: Observability analysis of the attitude $qb2l$ and the accelerometers scale factors kf 178

D.3 Step 3: Observability analysis of the accelerometers biases bf 181

D.4 Step 4: Observability analysis of the gyroscopes biases $b\omega$ 181

D.5 Step 5: Observability analysis of the gyroscopes cales factors $k\omega$ 183

D.6 Note on the observability using only position information 187

List of Figures

Figure 2-1 GNSS error sources 9
Figure 2-2 Allan variance deviation of the IMU gyroscopes 19
Figure 2-3 Allan variance deviation of the IMU accelerometers 19
Figure 2-4 INS mechanization in the navigation frame (adapted from [20]) 21
Figure 2-5 Principle of the magnetic WSS 22
Figure 2-6 2D Ackermann steering model 23
Figure 2-7 Comparison of the estimated heading rate using rear wheels and steering wheel . 25
Figure 2-8 Comparison of the estimated heading rate error using rear wheels and steering wheel 25
Figure 2-9 Velocity estimation errors for each wheel as a function of true velocity 26
Figure 3-1 Classification of vision-based techniques 33
Figure 3-2 Trajectory estimation 34
Figure 3-3 Pinhole camera projection 36
Figure 3-4 Visual odometry diagram 38
Figure 3-5 Optical flow matching 39
Figure 3-6 Feature matching 40
Figure 3-7 Epipolar geometry constraint 41
Figure 3-8 VSLAM triangulation process 43
Figure 3-9 Keyframe selection 44
Figure 3-10 ORB-SLAM trajectory before loop closure 48
Figure 3-11 ORB-SLAM trajectory after loop closure 48
Figure 4-1 Multi-sensor fusion architecture 55
Figure 5-1 Loss of integrity 81
Figure 5-2 Integrity risk tree 88
Figure 5-3 GPS fault tree 88
Figure 5-4 VSLAM fault tree 89
Figure 5-5 HAL computation from a typical traffic example 94
Figure 5-6 multi-sensor architecture integrity monitoring 95
Figure 6-1 Measuremet campaign equipment 100
Figure 6-2 Reference stations distance with respect to the car (urban test) 101
Figure 6-3 Reference position accuracy (standard deviation) 101
Figure 6-4 Sensor configuration for data collection 102
Figure 6-5 Urban and suburban trajectory 103
Figure 6-6 Suburban environment description 103
Figure 6-7 Urban environment description 104
Figure 6-8 Francazal airport trajectory 104
Figure 6-9 $C/N0$ of GPS measurements in suburban-urban environment 106
Figure 6-10 Number of GPS measurements before and after exclusion through comparison to $C/N0 = 30$ dB-Hz in suburban-urban environment 106

Figure 6-11 Impact of the low $C/N0$ exclusion on HDOP, horizontal position error and horizontal velocity error in suburban-urban environment 107
Figure 6-12 GPS L1 horizontal position error and its associated 3σ bound without low $C/N0$ exclusion in suburban-urban environment 108
Figure 6-13 GPS L1 horizontal velocity error and its associated 3σ bound without low $C/N0$ exclusion in suburban-urban environment 108
Figure 6-14 $C/N0$ of GPS measurements in airport environment 109
Figure 6-15 Number of GPS measurements before and after exclusion through comparison to $C/N0 = 30$ dB-Hz in airport environment 109
Figure 6-16 GPS L1 horizontal position error and its associated 3σ bound with low $C/N0$ exclusion in airport environment 110
Figure 6-17 GPS L1 horizontal velocity error and its associated 3σ bound with low $C/N0$ exclusion in airport environment 110
Figure 6-18 ORB-SLAM trajectory in vision frame (Test 1) 111
Figure 6-19 ORB-SLAM image processing (Test 1) 111
Figure 6-20 ORB-SLAM image processing (Test 2) 112
Figure 6-21 ORB-SLAM trajectory in vision frame (Test 2) 112
Figure 6-22 Comparison between ORB-SLAM and reference trajectories (Test 1) 113
Figure 6-23 Reference and ORB-SLAM trajectories after alignment (Test 1) 113
Figure 6-24: ORB-SLAM transformation estimation using a sliding window (Test 1) 114
Figure 6-25 ORB-SLAM ATE after alignment (Test 1) 115
Figure 6-26 Comparison between ORB-SLAM and reference trajectories (Test 2) 116
Figure 6-27 Reference and ORB-SLAM trajectories after alignment (Test 2) 116
Figure 6-28 ORB-SLAM transformation estimation (Test 2) 117
Figure 6-29 Horizontal position error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1) 118
Figure 6-30 Horizontal velocity error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1) 118
Figure 6-31 Heading error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1) 118
Figure 6-32 IMU accelerometer and gyroscope biases and their associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1) 120
Figure 6-33 IMU accelerometer and gyroscope scale factors and their associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1) 121
Figure 6-34 Comparison of the horizontal position error and the horizontal velocity error before and after innovation test using GPS/INS/WSS/NHC filter (Test 1) 122
Figure 6-35 Number of GPS measurements before and after innovation test exclusion 122
Figure 6-36 Horizontal position and velocity errors and their associated 3σ bound of the GPS/INS/WSS/NHC/IT/VSLAM filter (Test 1) 124
Figure 6-37 ORB-SLAM transformation parameters estimated by the GPS/INS/WSS/NHC/IT/VSLAM filter (Test 1) 125
Figure 6-38 Horizontal position error and its associated 3σ bound of the GPS/INS/WSS/NHC/IT/VSLAM filter with 300 s GPS outage (Test 1) 126

Figure 6-39 Comparison of the different configuration horizontal position error without GPS outage (Test 1) 126
Figure 6-40 Comparison of the different configuration horizontal position error with 300 s GPS outage (Test 1) 126
Figure 6-41 Comparison of the different configuration horizontal position error without GPS outage (Test 2) 128
Figure 6-42 Comparison of the different configuration horizontal position error with 60 s GPS outage (Test 2) 128
Figure A-1 ECI, ECEF and ENU frames 150
Figure A-2 Body frame 150
Figure A-3 Image, camera and vision frames 151
Figure B-1 Euler angles 159
Figure B-2 Transformation from (l) to (v) 160

List of Tables

Table 1.1 Summary of the existing positioning technologies and of their performance 2
Table 2.1 Multipath model parameters 13
Table 2.2 Standard deviation values of the pseudorange rate 14
Table 2.3 IMU specifications 18
Table 2.4 Comparison of theoretical and experimental IMU errors 19
Table 2.5 WSS velocity scale factor 26
Table 4.1 Sensor outputs and associated frames 60
Table 4.2 Expression of the $\mathcal{O}$ matrix 74
Table 4.3 Summary of state observability of the defined architecture 78
Table 5.1 Classification of VSLAM errors into fault-free mode and failure modes 86
Table 5.2 Integrity risk and HAL for steering in the airport Gate 94
Table 5.3 Computation of the maximum allowable SLAM uncertainty 96
Table 6.1 Satellite availability before and after exclusion through comparison to $C/N0 = 30$ dB-Hz in suburban-urban environment 107
Table 6.2 Comparison of the navigation solution accuracy before and after low $C/N0$ exclusion in suburban-urban environment 108
Table 6.3 Comparison of the navigation solution accuracy before and after low $C/N0$ exclusion in airport environment 110
Table 6.4 ORB-SLAM transformation estimation (Test 1) 113
Table 6.5 ORB-SLAM accuracy after alignment (Test 1) 115
Table 6.6 ORB-SLAM transformation estimation (Test 2) 116
Table 6.7 ORB-SLAM accuracy after alignment (Test 2) 117
Table 6.8 Horizontal accuracy performance before and after innovation test using GPS/INS/WSS/NHC filter (Test 1) 123
Table 6.9 Comparison of the different configuration horizontal position accuracy with no GPS outage (Test 1) 127
Table 6.10 Comparison of the different configuration horizontal position drift after 300s of GPS outage (Test 1) 127
Table 6.11 Comparison of the different configuration horizontal velocity accuracy with no GPS outage (Test 1) 127
Table 6.12 Comparison of the different configuration horizontal velocity drift after 300s of GPS outage (Test 1) 127
Table 6.13 Comparison of the different configuration heading accuracy with no GPS outage (Test 1) 127
Table 6.14 Comparison of the different configuration heading drift after 300s of GPS outage (Test 1) 127
Table 6.15 Comparison of the different configuration horizontal position accuracy with no GPS outage (Test 2) 128
Table 6.16 Comparison of the different configuration horizontal position drift after 60s of GPS outage (Test 2) 128

Table 6.17 Comparison of the different configuration horizontal velocity accuracy with no GPS outage (Test 2) 129
Table 6.18 Comparison of the different configuration horizontal velocity drift after 60s of GPS outage (Test 2) 129
Table 6.19 Comparison of the different configuration heading accuracy with no GPS outage (Test 2) 129
Table 6.20 Comparison of the different configuration heading drift after 60s of GPS outage (Test 2) 129
Table D.1 Expression of the $\mathcal{O}$ matrix 177

List of Acronyms

ABAS: Aircraft Based Augmentation System

ABS: Anti-lock Braking System

AIME: Autonomous Integrity Monitoring Extrapolation

AL: Alert Limit

ARW: Angle Random Walk

A-SMGCS: Advanced Surface Movement Guidance and Control System

ATE: Absolute Trajectory Error

BA: Bundle Adjustment

CAN: Controller Area Network

CDF: Cumulative Distribution Function

C/A: Coarse Acquisition

$\mathbf{C/N_0}$: Carrier-to-Noise Ratio

DCM: Direction Cosine Matrix

DME: Distance Measurement Equipment

DoF: Degrees of Freedom

DR: Dead Reckoning

ECEF: Earth Centered Earth Fixed

ECI: Earth-Centered-Inertial

EKF: Extended Kalman Filter

ENU: East North Up

FAST: Features from Accelerated Segment Test

FD: Fault Detection

FDE: Fault Detection and Exclusion

FT: Feature Tracking

GBAS: Ground Based Augmentation Systems

GLR: Generalized Likelihood Ratio

GNSS: Global Navigation Satellite System

GPS: Global Positioning System

HAL: Horizontal Alert Limit

HDOP: Horizontal Dilution Of Precision

HMI: Hazardous Misleading Information

HPE: Horizontal Position Error

HPL: Horizontal Protection Level

IBA: Bundle Adjustment with inequality constraint

ICAO: International Civil Aviation Organization

ICR: Instantaneous Center of Rotation

IMU: Inertial Measurement Unit

INS: Inertial Navigation System

IR: Integrity Risk

IT: Innovation Test

KF: Kalman Filter

LIDAR: Light Radar

LKF: Linearized Kalman Filter

LOS: Line-Of-Sight

LSD: Large Scale Direct

MEMS: Micro- Electro-Mechanical Systems

MD: Missed Detection

MSF: Major Service Failure

MSS: Maximum Solution Separation

NHC: Non-Holonomic-Constraints

NLOS: Non-Line-Of-Sight

NTP: Network Time Protocol

ORB: Oriented FAST and Rotated BRIEF

PE: Position Error

PF: Particle Filter

PnP: Perspective from n Points

PPS: Pulse Per Second

PTAM: Parallel Tracking And Mapping

RAIM: Receiver Autonomous Integrity Monitoring

RANSAC: RANdom SAmple Consensus

RF: Radio Frequency

RGB-D: Red-Green-Blue-Depth

RNP: Required Navigation Performance

ROS: Robot Operating System

RPE: Relative Pose Error

RTK: Real Time Kinematic

SBAS: Satellite Based Augmentation System

SF: Scale Factor

SIFT: Scale Invariant Feature Transform

SLAM: Simultaneous Localization And Mapping

SNR: Signal-to-Noise Ratio

SURF: Speeded Up Robust Feature

SV: Satellite in View

TTA: Time To Alert

UKF: Unscented Kalman Filter

URA: User Range Accuracy

VAL: Vertical Alert Limit

VO: Visual Odometry

VPL: Vertical Protection Level

VRW: Velocity Random Walk

VSLAM: Visual Simultaneous Localization And Mapping

WSS: Wheel Speed Sensor

ZARU: Zero Angular Rate Update

ZVU: Zero Velocity Update

Notations

Conventions

x	Scalar
$\boldsymbol{x}$	Vector or Matrix
$\tilde{x}$	Measured quantity
$\hat{x}$	Estimated quantity
δx	Error on the quantity estimation
x_k	Quantity at epoch k

Variables

- **Roman letters**

$\boldsymbol{b}_m$	Bias affecting the measurement m
$\boldsymbol{b_m}$	Bias affecting the measurement vector $\boldsymbol{m}$
b_G^{clk}	GNSS receiver clock bias
$\dot{b}_G^{clk}$	GNSS receiver clock drift
c	Speed of light
$\boldsymbol{C}_{x2y}$	Rotation matrix from x-frame to y-frame
$\boldsymbol{c_0}$	Camera principal point
d^i	Geometric distance between the GNSS receiver and the i^{th} satellite
$\boldsymbol{E}_{b2l}$	Skew-symmetric matrix of the vehicle attitude error
$\boldsymbol{E}^{ATE}$	Absolute Trajectory Error
$\boldsymbol{E}^{RPE}$	Relative pose Error
$\boldsymbol{E}^c$	Camera essential matrix
$\boldsymbol{F}$	System transition matrix (Kalman filter)
$\boldsymbol{F}^c$	Camera fundamental matrix
$\boldsymbol{f}_{ib}^b$	Accelerometer specific force expressed in the body frame
$\boldsymbol{F}_{ib}^b$	Diagonal matrix of $\boldsymbol{f}_{ib}^b$
F_{iono}	Mapping function used after Klobuchar ionospheric error corrections

f_{L1}	GPS L1 C/A signal frequency
f_{L2C}	GPS L2C signal frequency
f_0	GNSS signal frequency
f	Camera focal length
F_{tropo}	Mapping function used after UNB3 tropospheric error corrections
$\boldsymbol{g}^l$	Gravity vector expressed in the navigation frame
$\boldsymbol{G}^{pose}$	Ground truth vehicle pose
h	Vehicle ellipsoidal height
$\boldsymbol{H}_m$	Observation matrix associated to the measurement m (Kalman filter)
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{K}_c$	Camera calibration matrix
k_m	Scale factor affecting the measurement m
$\boldsymbol{k_m}$	Scale factor affecting the measurement vector $\boldsymbol{m}$
k_{ff}	Multiplicative coefficient computed from the integrity risk allocated to the fault-free mode
L_v	Vehicle wheelbase
N_{teeth}	Number of vehicle wheel gear teeth
N_{sat}	Number of satellites in view
$\boldsymbol{\mathcal{O}}$	System observability matrix
$\boldsymbol{p}_x^y$	Position of point $\boldsymbol{x}$ in the y-frame
$\boldsymbol{p}_h^y$	Position in homogeneous coordinates in the y-frame
P_{FA}	False alarm probability
$\mathrm{P}_{fm_i}^S$	probability of occurrence of the sensor S failure mode fm_i
$\mathrm{P}_{HMI\|fm_i}^S$	HMI probability given that the sensor S failure mode fm_i is present and not detected
$\mathrm{P}_{IR,ff}^S$	Integrity risk allocated to the sensor S fault-free mode
P_{IR,fm_i}^S	Integrity risk allocated to the sensor S failure mode fm_i
P_{IR}	Integrity risk
$\boldsymbol{P}$	State covariance matrix (Kalman filter)
P_{MD}	Miss-detection probability
$\boldsymbol{q}_{x2y}$	Quaternion defining the rotation from x-frame to y-frame
$\boldsymbol{Q}_k$	spectral density matrix (Kalman filter)

$\boldsymbol{q}_{\eta x}$ Spectral density of the process x whose error is η

$\boldsymbol{R}$ Observation covariance matrix (Kalman filter)

R_M Earth radius of curvature in the meridian

R_N Earth radius of curvature in the prime vertical

R_{wheel} Vehicle wheel radius

r_s Vehicle steering ratio

$\boldsymbol{S}^{pose}$ VSLAM output pose

s_γ Camera skew factor

T_s Sampling period

T_{teeth} Elapsed time between two wheel gear teeth detection

u Horizontal pixel coordinate in the image frame

v Vertical pixel coordinate in the image frame

$\boldsymbol{v}_x^y$ Velocity of point $\boldsymbol{x}$ in the y-frame

v_{FL} Vehicle front left wheel velocity

v_{FR} Vehicle front right wheel velocity

v_R Vehicle rear axle center velocity

v_{RL} Vehicle rear left wheel velocity

v_{RR} Vehicle rear right wheel velocity

$\boldsymbol{W}_{ib}^b$ Diagonal matrix of $\boldsymbol{\omega}_{ib}^b$

$\boldsymbol{X}_S$ State vector associated to the sensor S (Kalman filter)

$\boldsymbol{X}$ Total system state vector (Kalman filter)

$\boldsymbol{Z}$ Observation vector (Kalman filter)

$\boldsymbol{Z}_S$ Observation vector associated to the sensor S (Kalman filter)

- **Greek letters**

γ_g Vertical component of the gravity in the local frame

Δf_G^i Doppler measurement between the GNSS receiver and the i^{th} satellite

δ_L Vehicle front left wheel steering angle

$\Delta \boldsymbol{p}_x^y$ Lever arm between the points $\boldsymbol{x}$ and $\boldsymbol{y}$

δ_R Vehicle front left wheel steering angle

Δt^r	Offset between the GNSS receiver clock and GPS-Time
Δt^i	Offset between the i^{th} satellite clock and GPS-Time
$\Delta \dot{t}^r$	Drift of the GNSS receiver clock with respect to GPS-Time
$\Delta \dot{t}^i$	Drift of the i^{th} satellite clock with respect to GPS-Time
$\Delta \alpha_s$	steering wheel turn
δ_s	Vehicle steering angle
$\Delta \delta_s$	Vehicle steering turn
$\boldsymbol{\varepsilon}_{x2y}$	x-frame to y-frame rotation error
ε_m	measurement noise
$\boldsymbol{\varepsilon_m}$	Measurement vector noise
η_m	Measurement process noise (Kalman filter)
$\boldsymbol{\eta_m}$	Measurement vector process noise (Kalman filter)
θ	Vehicle pitch
θ^i	GNSS i^{th} satellite elevation angle
$\mathbf{\Pi}$	Local frame to image frame projection matrix
ρ_G^i	GNSS i^{th} satellite pseudorange
$\boldsymbol{\rho}_G$	GNSS pseudorange vector
$\dot{\rho}_G^i$	GNSS i^{th} satellite pseudorange rate
$\dot{\boldsymbol{\rho}}_G$	GNSS pseudorange rate vector
σ_m	Measurement error standard deviation
τ_m	Correlation time of the Gauss-Markov process m
φ	Vehicle roll
ϕ	vehicle geodetic latitude
$\mathbf{\Phi}$	Discertized system transition matrix (Kalman filter)
ϕ_m	User's geomagnetic latitude
ψ	Vehicle heading
ω_e	Earth rotation/angular rate
$\boldsymbol{\omega}_{xy}^z$	Rotation/angular rate of y-frame with respect to x-frame expressed in z-frame
ω_o	2D rotation/angular rate of an object o with respect to the local 2D frame

Functions

h	Observation function (Kalman filter)
f	System function (Kalman filter)
$I(u, v, k)$	(u, v) pixel intensity function at epoch k
Q_{dist}	Probability density function of the distribution $dist$

Mathematical operations

- **Operations on variables**

diag $(\boldsymbol{x})$	Diagoanl matrix of vector $\boldsymbol{x}$
$\lvert x \rvert$	Module of scalar x
$\lVert \boldsymbol{x} \rVert_2$	Euclidean norm of vector $\boldsymbol{x}$
$\dot{x}$	Time derivative of scalar x
$\dot{\boldsymbol{x}}$	Time derivative of vector $\boldsymbol{x}$
$\boldsymbol{x}^T$	Transpose of vector/matrix $\boldsymbol{x}$
$\boldsymbol{x}^{-1}$	Inverse of vector/matrix $\boldsymbol{x}$
e^x	exponential of scalar x
$\boldsymbol{x} \times \boldsymbol{y}$	Cross product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$
$(\boldsymbol{x}) \times$	Skew-symmetric matrix of vector $\boldsymbol{x}$
$\boldsymbol{x} \otimes \boldsymbol{y}$	Quaternion multiplication of quaternions $\boldsymbol{x}$ and $\boldsymbol{y}$

- **Operations on functions**

$\underset{\boldsymbol{x}}{\operatorname{argmin}} f(\boldsymbol{x})$	The value of vector $\boldsymbol{x}$ that minimizes the function $f(\boldsymbol{x})$
$\frac{\partial f(x, y, \cdots)}{\partial x}$	Partial derivative of $f(x, y, \cdots)$ with respect to the variable x
∇f	Gradient of the f function
f^{-1}	Inverse of the f function

Other notations

$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and standard deviation σ
$\binom{n}{p}$	Mathematical combination (n choose p)
$\mathrm{P}(e)$	Probability that an event e occurs

Chapter 1

Introduction

1.1 Motivation and challenges

Automated and autonomous driving applications are nowadays closer than ever to being part of our everyday life thanks to the ongoing research advances on this topic (Okuda, 2014). This promising field is facing at present several challenges. Technological aspects of autonomous/automated driving such as navigation (positioning, path planning and decision making, control), perception, obstacle avoidance, etc. are among the most challenging tasks. In particular, positioning is an important task that needs to be addressed carefully. In fact, many other issues involving safety of life, such as vehicle control or path planning and decision making, rely on positioning and require not only having continuously an accurate navigation solution but also to be confident in this information.

However, this is a complex task because of many challenges. First, road applications require stringent level of positioning accuracy and integrity in order to prevent human harm/material damage in environments where people, vehicles and other structures are likely to be close to each other. At present, there are no positioning performance requirements, but works are going on to handle this problem (GSA, 2015). Second, although GNSS is the most widely used system to perform positioning, this system is not free from drawbacks since it suffers from problems related to signal propagation especially in urban environments where buildings, trees and other structures hinder the reception of GNSS signals and degrade their quality. This can result in significant positioning error exceeding in some cases a kilometer. Many techniques are proposed in the literature to mitigate these problems and improve the GNSS accuracy. Unfortunately, all these techniques have limitations (Jiang, 2012). Finally, the third challenge is the sensors cost which is a very restricting constraint. In fact, a possible way to overcome the GNSS problem is to fuse its measurements with other sensors having complementary characteristics like being independent from the surrounding environment. Generally, the most widely implemented hybridization algorithms for land vehicles fuse GNSS measurements with inertial and/or odometric data. Thereby, these DR sensors ensure the system continuity when GNSS information is unavailable and improve the system performance when GNSS signals are corrupted, and in turn, the GNSS limits the drift of the DR solution if it is available. However, the performance achieved by this hybridization depends thoroughly on the quality of the DR sensor used especially when GNSS signals are degraded or unavailable. It is well known that the high grade *Inertial Measurement Units* (IMU) are very expensive, hence not adapted to ground vehicle applications. The only usable IMUs are the Micro-Electro-Mechanical Systems MEMS IMUs having an affordable cost at the expense of quality, despite the progress that has been done in this field.

The state-of-the-art of performance and cost constraints is summarized in the following table (BGLE-MIMOSA, 2012). It shows that each sensor and configuration of the existing solutions has their advantages and drawbacks. GNSS is not expensive, but it is not accurate, not robust and does not provide a continuous navigation solution in difficult environments. If it is used in precise positioning mode (e.g. RTK), then the accuracy is improved but its cost increases considerably. In addition, the dependence on the environmental conditions degrades the continuity and the robustness in difficult GNSS environment. The combination of GNSS and IMU improves the continuity of the navigation solution. However, the accuracy of this solution is only possible when using high grade sensors. Finally, if GNSS is fused with other low-cost sensors, then the cost decreases but the accuracy level is of few meters.

Technology	Accuracy	Continuity	Robustness	Cost	Examples of industrial solutions
GNSS	Few meters	Low	Low	Tens to hundreds of euros	Ublox, Septentrio, NovAtel, CSR
GNSS + RTK	Decimeter	Low	Low	Tens of thousands of euros	Ublox, Septentrio, NovAtel, CSR
GNSS+ IMU	Few meters if GNSS+MEMS IMU	High	Medium to high	Hundreds of euros	Novatel (SPAN), Ublox (Automotive Dead Reckoning)
	Sub-meter to meter if GNSS+higher grade IMU			Tens to hundreds of thousands of euros	
	Decimeter if GNSS RTK+higher grade IMU				
GNSS+IMU+ other sensors	Few meters if GNSS	High	High	Hundreds of euros	STMicroelectronics /CSR (LMS333D + SiRFusion)

Table 1.1 Summary of the existing positioning technologies and of their performance

Therefore, other sensors have been considered to find a compromise between accuracy and cost. Many sensors have been proposed and tested in the literature. Among these sensors we can find the sensors providing range information such as the Light Radar (LIDAR) (Gao, 2015), the laser rangefinders (Hiremath, 2014) and the ultra-sound sensors (Nordh, 2007). The LIDAR has very good performances, but its cost is relatively high for car applications. Ultra-sound sensors are also not adapted to car applications in large scale environments because of the lack in their range. The drawback of all these sensors is that they only provide ranging information.

Recently, it has been proven that vision could be a promising navigation sensor that provides accurate navigation solution (Nistér, 2006). Indeed, cameras have the advantage of providing an extensive amount of information while having low weight, limited power consumption, small size and reasonable cost (Engel, 2012). Vision-based navigation is inspired from human navigation: the observation of the world and objects around us allows us to deduce our relative position with respect to our environment. This important information makes vision-based navigation a promising solution able to improve navigation performance. The first navigation algorithms called *Visual Odometry* were developed for NASA planetary rovers (Moravec, 1980) exploring Mars to be able to measure their 6-degree-of-freedom (DoF) in rough terrains where the classical wheel odometry experience slippage. Then, VO has progressed over the years. In (Nistér, 2006), the assessment of a stereo pair of cameras performance in realistic ground-vehicle scenarios over a few hundred meters shows a few meters accuracy in positioning and sub-degree accuracy in heading angle recovery. In the last years, successful results have been obtained using a more sophisticated vision-based technique consisting not only in motion measuring but also in mapping the surrounded environment. This technique called *Visual Simultaneous Localization And Mapping* (VSLAM) (Dissanayake, 2001), has proven to have better performance than VO. A state of the art review of vision-based positioning techniques and the solutions proposed to benefit from this information in positioning have been done in this thesis, and a paper addressing this review (Ben-Afia, 2014) has been published.

Taking into account all these constraints, challenges and current advances, a French project called MIMOSA has been launched in 2013 by industrials and researches in order to develop a navigation equipment for road professional market, having good positioning performance and taking advantage of progresses made in sensor fusion and vision-based positioning fields. Two applications are targeted in this project: ground vehicles in airport surfaces and garbage trucks moving in suburban to urban environments.

The idea in this project is not to develop a new vision-based positioning technique, but to exploit the progress made in this field in order to fuse visual information with other positioning sensors to achieve a robust navigation system capable of providing continuously sub-meter level accuracy. This means that, using a visual processing module, visual information should be converted in motion or positioning information before its integration in the main navigation system consisting of multiple sensors. This approach of considering the visual processing module as a black box has the advantage of having a multi-sensor system that is independent from the technique used by the visual module. This allows the modularity of the system by giving the possibility to decide whether this information should be used or not, especially because vision depends on light conditions.

This Ph.D. thesis is part of MIMOSA project and aims at exploiting the visual information in a multi-sensor fusion framework, while taking into account the requirements and technological constraints set by the MIMOSA project and mentioned previously. More details about the thesis objectives are given in Section 1.2

1.2 Thesis objectives and contribution

The global objective of this Ph.D. thesis is to define a multi-sensor navigation that takes advantage of low-cost sensors, especially vision sensors, and to provide continuously a robust navigation solution meeting the sub-meter level accuracy. This overall objective has been divided in several sub-objectives and realized in the following steps:

- The review of the state of the art of low-cost sensors and techniques used currently in positioning is performed. A special attention is given to vision-based positioning to understand its principles and its techniques. A classification of vision-based positioning techniques is proposed and the key elements of this particular type of navigation are investigated. In addition, the different techniques and strategies to fuse sensor information are studied and compared.
- The sensors that will be integrated in the proposed multi-sensor navigation solution in addition to camera are selected. This selection is mainly based on the cost/performance trade off.
- The identified sensors are studied, and their measurements are characterized. In addition, the different ways to use these sensors in positioning are investigated.
- A multi-sensor hybridization architecture integrating the identified sensors is proposed. The choices made to define this architecture are justified and a detailed description of the mathematical aspect of hybridization is provided.
- The review of the integrity challenges of the proposed solution is addressed. In particular the vision-based solution integrity is addressed, and its possible failure modes are listed. The challenges of the overall proposed hybridization architecture are also discussed.

The performances achieved with the proposed architecture are assessed, based on real data collected in the environments targeted by the MIMOSA project, namely urban and airport environments. The contribution of each sensor to the improvement or degradation of the navigation solution is highlighted. In particular, the visual information contribution is studied.

1.3 Thesis outline

In order to synthesize the work done during this Ph.D. thesis, the document is organized as follows:

Chapter 2 is an overview of sensors selected to be used in the multi-sensor architecture based on their performance/cost tradeoff. First, an overview of these sensors, mainly composed of a GNSS receiver for global navigation, an IMU for inertial navigation and 4 *Wheel Speed Sensors* (WSS) for odometry, is performed. Then, the principle of their associated navigation modules is explained, and their measurement errors are characterized. Finally, a state of the art review of the sensor fusion strategies and solutions is addressed.

Chapter 3 reviews the existing vision-based navigation techniques and highlights their challenges especially in outdoor large-scale environments. First, a classification of the different manners these techniques have been used in the literature is proposed. Then the review of the

technical aspects of vision-based positioning is performed. A particular interest is given to ORB-SLAM (Mur-Artal, 2015) open-source software, used in this study to provide the visual information. Then, the challenges of these techniques are emphasized. Finally, the measurement model of the visual information is given.

Chapter 4 provides a detailed explanation of the proposed multi-sensor navigation system. The choices made to design this system are justified and the contribution of each sensor is described mathematically. A special attention is given to the observability analysis of the parameters the developed system should be able to estimate.

Chapter 5 addresses the integrity monitoring problem and discusses the integrity challenges faced, on one hand, with the visual information and its processing with the SLAM technique, and on the other hand with the hybridized structure whose integrity study is not a trivial task.

Chapter 6 provides the results and the performance analysis of the proposed solution. The presented results are based on real datasets collected in urban and airport environments. The contribution of visual information is particularly highlighted.

Finally, the last chapter concludes on the works done during the Ph.D. and proposes some perspectives that could be considered for future works.

Chapter 2

Navigation Systems for Ground Vehicles

This chapter is an overview of sensors selected to be used in the multi-sensor architecture based on their performance/cost tradeoff. First, an overview of these sensors composed of a GNSS receiver for global navigation, an IMU for inertial navigation and 4 Wheel Speed Sensors (WSS) for odometry, is performed. Then, the principle of their associated navigation modules is explained and their measurement errors are characterized. Finally, a state of the art review and comparison of the sensor fusion strategies and solutions is addressed.

Contents

2.1 GNSS 8
2.1.1 System principle 8
2.1.2 GNSS Measurement models....10
2.1.3 GNSS Navigation solution 14
2.2 Inertial navigation 16
2.2.1 System principle 16
2.2.2 Inertial measurement errors and models....17
2.2.3 Inertial mechanization 19
2.3 Odometry 22
2.3.1 Wheel Speed Sensors 22
2.3.2 Steering angle sensors 22
2.3.3 Ackermann steering model....23
2.3.4 Odometric measurement models 25
2.4 Sensor complementarity and integrated navigation....26
2.4.1 Integration techniques and strategies....27
2.4.2 State of the art of existing ground vehicle integrated navigation based on INS, GNSS and/or odometry and limits of the existing solutions....29
2.5 Conclusion 30

2.1 GNSS

2.1.1 System principle

Global navigation Satellite Systems are positioning systems able to provide a user, equipped with a GNSS receiver, with a tridimensional position, velocity, and time. The concept of GNSS positioning is based on measuring the time delay it takes for a signal transmitted by the satellite to reach a user receiver. This signal propagation time is then multiplied by the speed of light to obtain an estimate of the satellite-to-receiver range. By measuring the propagation time of the signals broadcast from multiple satellites with known positions, thanks to the information broadcast in the navigation message, the receiver may determine the position of its antenna.

The described GNSS principle requires synchronized satellite and receiver clocks to compute the signal propagation time between its transmission and reception. However, these clocks are actually not synchronized, resulting in errors in the measured distances. Consequently, the computed distance between the satellite and the receiver includes a time component associated to the offset between the satellite and the receiver clocks, in addition to the true range. For this reason, this computed distance is named pseudorange. The offset between the satellite and the receiver clocks is named satellite-receiver clock offset.

In order to monitor timing, a reference time, the GNSS Time, is defined. The satellite-receiver clock offset may be expressed as the difference between the GNSS-Time-satellite clock offset and the receiver-GNSS Time clock offset. The satellite clock drift is monitored by the ground segment and, parameters for the receiver to estimate the satellite-GNSS Time clock offset are transmitted in the navigation message, thus allowing the synchronization of the satellite time with GNSS Time. The GNSS-Time-receiver clock offset remains unknown and must be estimated in the navigation solution computation.

The mathematical model of the pseudorange provided by a satellite i at time epoch k is therefore given by:

$$\rho_{G,k}^{i} = d_k^i + c\left(\Delta t_k^r - \Delta t_k^i\right) + \varepsilon_k^i \tag{2.1}$$

where

- $d_k^i = \sqrt{\left(x_k - x_k^i\right)^2 + \left(y_k - y_k^i\right)^2 + \left(z_k - z_k^i\right)^2}$ is the geometric distance between the receiver and the satellite i
- $\boldsymbol{X}_{G,k} = (x_k\,, y_k\,, z_k)$ are the receiver coordinates in a reference frame
- $\boldsymbol{X}_{s,k}^i = \left(x_k^i\,, y_k^i\,, z_k^i\right)$ are the satellite i coordinates in the same reference frame
- c is the speed of light
- Δt_k^i is the GNSS-Time-satellite clock offset. This offset is broadcast in the navigation message and is corrected by the receiver using the correction models detailed in (Farrell, 1999).
- Δt_k^r is the receiver-GNSS-Time clock offset at the reception time.

- ε_k^i is the error affecting the pseudorange. This term reflects the uncertainty in the estimation of the signal propagation time and is due to many sources of error. These errors corrupting the GNSS pseudorange measurements are mainly:
  - the uncertainties on the GNSS-Time-satellite clock offset correction model and Ephemeris
  - Error due to ionosphere delay
  - Error due to troposphere delay
  - Error due to Multipath and *Non-Line-Of-Sight* (NLOS) signals
  - Receiver noise

These errors are presented in Figure 2-1. When a GNSS signal propagates from satellite to the GNSS antenna, it goes through the atmosphere and undergoes ionosphere and troposphere delays. Then, it may be reflected by buildings and structures surrounding the antenna resulting in receiving the reflected signal in addition to the direct signal (multipath), or even receiving only the indirect signal (NLOS).

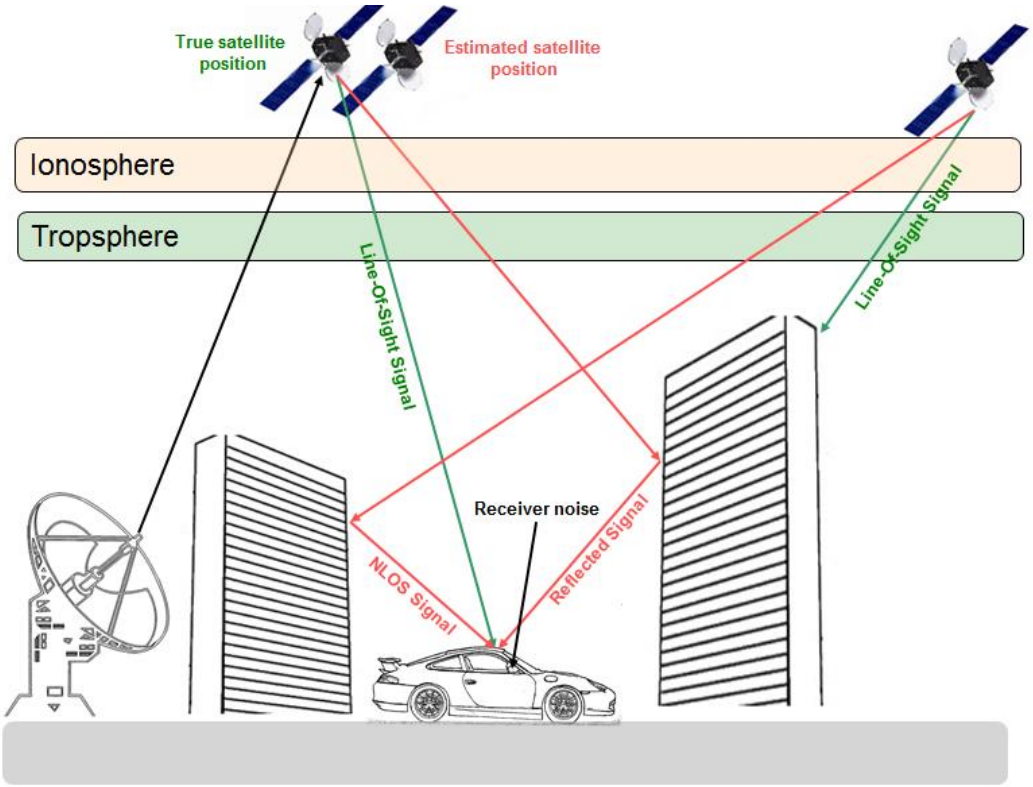


Figure 2-1 GNSS error sources

Based on equation (2.1), 4 unknowns should be actually estimated: the three-dimensional user position as well as the receiver-GNSS-Time clock offset. This explains the need to have at least 4 GNSS pseudorange measurements to solve the 4-unknown system of equations.

A GNSS receiver can also compute the velocity of the user by exploiting the pseudorange rate measurement deduced from the Doppler measurements using the following formula:

$$\dot{\rho}_{G,k}^{i} = \Delta f_G^i \cdot \frac{c}{f_0} \tag{2.2}$$

where

- Δf_G^i is the Doppler measurement provided by the GNSS receiver. It represents the frequency shift caused by the relative motion between the receiver and the satellite i.
- c is the speed of light
- f_0 is the transmitted signal frequency ($f_0 = 1575.42$ MHz for L1 signals)

The mathematical model of the pseudorange rate provided by a satellite i at time epoch k is therefore given by:

$$\dot{\rho}_{G,k}^{i} = \dot{d}_k^i + c\left(\Delta \dot{t}_k^r - \Delta \dot{t}_k^i\right) + \dot{\varepsilon}_k^i \tag{2.3}$$

where

- $\dot{d}_k^i$ is the receiver-satellite distance rate
- c is the speed of light
- $\Delta \dot{t}_k^r$ is the receiver clock drift
- $\Delta \dot{t}_k^i$ is the satellite clock drift
- $\dot{\varepsilon}_k^i$ is the error drift

The accuracy of the velocity computed using the Doppler measurements is in the order of few centimeters per second (Hoffmann-Wellenhof, 1992).

2.1.2 GNSS Measurement models

To evaluate GNSS positioning performance, error sources have to be taken into account and corresponding measurement errors have to be modeled as precisely as possible.

2.1.2.1 Pseudorange measurement model

Two types of pseudorange error models are considered:

- The nominal error model: it characterizes the pseudorange measurement errors that are present when all GNSS segments are working according to their specifications and the magnitudes of other external error sources have typical values. These errors are modelled as zero-mean independent Gaussian distributions. Therefore, the mathematical model of nominal error of pseudorange i at time epoch k is given by:

$$\varepsilon_k^i \sim \mathcal{N}\left(0, \left(\sigma_{PR,k}^i\right)^2\right) \tag{2.4}$$

  where $\sigma_{PR,k}^i$ is the standard deviation of the pseudorange error affecting the satellite i measurement.
- The faulty error model: it characterizes unusual errors that may cause a positioning failure. The origins of these errors may be:
  - A satellite clock malfunction

- A large multipath/NLOS
- A large and punctual ionospheric delay
- etc.

These errors are modelled as a bias added to the nominal error. Therefore, the mathematical model of an error on pseudorange measurement i at time epoch k is given by:

$$\varepsilon_k^i \sim \mathcal{N}\left(b_k^i, \left(\sigma_{PR,k}^i\right)^2\right) \tag{2.5}$$

where b_k^i is the additive bias.

The mathematical pseudorange model given in equation (2.4) is the result of convolution of the errors listed in Section 2.1.1, which are assumed independent. Each of these errors is assumed to be zero-mean and Gaussian. This assumption is conservative since real errors are not exactly described by this distribution. The zero-mean Gaussian model is obtained using overbounding techniques described in (DeCleene, 2000). The model of each individual error is described in the following sections.

2.1.2.1.1 Satellite clock error and ephemeris errors

The satellite clock error is the offset between the time maintained by the atomic clocks on the satellite and the GNSS Time. The parameters to correct the satellite clock error are computed and broadcast to the users. After correction, a residual error remains.

The Ephemeris error is the difference between the estimated and the actual satellite position. The estimated satellite position is uploaded to the satellite and is broadcast to the users in the navigation message.

The satellite clock and ephemeris errors are overbounded by a zero-mean Gaussian distribution with a standard deviation equal to the *User Range Accuracy* (URA). The value of URA for modernized GPS is given in (Neri, 2011):

$$\varepsilon_{clk\&eph,k}^i \sim \mathcal{N}\left(0, \sigma_{URA,k}^2\right) \tag{2.6}$$

with $\sigma_{URA,k} = 0.85\ m$

2.1.2.1.2 Ionosphere propagation error

The ionospheric error is the measurement error caused by the propagation of the GNSS signals in the ionosphere. In this atmosphere region, the signal travels with a speed different from the speed of light, delaying consequently the pseudorange. The ionosphere is a dispersive medium, i.e. the propagation speed depends on the carrier frequency, so the ionospheric delay can be eliminated using dual frequency receivers. In this work only single L1 C/A frequency solutions are considered and in this case the ionospheric error can be reduced using suitable models, the most common being the Klobuchar model (Klobuchar, 1987) which removes about 50% of ionospheric error on average and whose coefficients are broadcast in the navigation message. The ionospheric residual error after correction with the Klobuchar model is given by (Salos, 2012):

$$\varepsilon_{iono,k}^i \sim \mathcal{N}\left(0, \left(\sigma_{ionoL1,k}^i\right)^2\right) \tag{2.7}$$

with

$$\sigma_{ionoL1,k}^{i} = \begin{cases} F_{iono} \cdot 9 & 0 \leq |\phi_m| \leq 20° \\ F_{iono} \cdot 4.5 & 20° < |\phi_m| \leq 55° \\ F_{iono} \cdot 6 & 55° < |\phi_m| \leq 90° \end{cases} \quad (2.8)$$

where

- F_{iono} is the mapping function used in klobuchar corrections:

$$F_{iono} = 1 + 16\left(0.53 - \frac{\theta^i}{180}\right)^3 \quad (2.9)$$

- θ^i is the satellite i elevation angle
- ϕ_m is the user's geomagnetic latitude

2.1.2.1.3 Troposphere propagation error

The troposphere error is the measurement error resulting from the propagation of the GNSS signals in the troposphere. In this region of atmosphere, the signal also travels with a speed different from the speed of light, adding consequently a delay to the pseudorange. The troposphere is a non-dispersive medium for frequencies up to 15 GHZ and dual-frequency cannot be used to eliminate its error. Several models can be used to reduce the influence of the tropospheric error, bringing sub-meter accuracy (Spilker, 1996). In this study, the UNB3 (Collins, 1999) model is used. After correction by the UNB3 model, the tropospheric residual error is given by:

$$\varepsilon_{tropo,k}^{i} \sim \mathcal{N}\left(0, \left(\sigma_{tropo,k}^{i}\right)^2\right) \quad (2.10)$$

with

$$\sigma_{tropo,k}^{i} = F_{tropo} \cdot \sigma_{tropo,v} \quad (2.11)$$

where

- $\sigma_{tropo,v} = 12cm$
- $F_{tropo} = \begin{cases} \frac{1.001}{\sqrt{0.002001 + \sin^2 \theta^i}} & for \theta^i \geq 4° \\ \frac{1.001}{\sqrt{0.002001 + \sin^2 \theta^i}}\left(1 + 0.015 \cdot \left(4° - \theta^i\right)^2\right) & for\ 2° \leq \theta^i < 4° \end{cases}$
- θ^i is the satellite i elevation angle.

2.1.2.1.4 Receiver noise

The receiver error is caused mainly by the thermal noise produced at the tracking process level. Errors are of few decimeters for code measurement (Conley, 2006). The mathematical model of this error is given by:

$$\varepsilon_{rcv,k}^{i} \sim \mathcal{N}\left(0, \left(\sigma_{rcv,k}^{i}\right)^2\right) \quad (2.12)$$

with (Julien, 2011):

$$\sigma_{rcv,k}^{i} = 1.5\ m$$

2.1.2.1.5 Multipath error

The multipath error is due to the reflection of the propagated signal on the surroundings of the GNSS antenna. In general, the multipath error is the dominant error in dense scenarios like urban environments. The impact of this error on the pseudorange can vary from few meters in benign environments to over one hundred meters in strongly reflective scenarios (Parkinson, 1996) and can be much greater in situations where only the NLOS are received by the receiver (Mezentsev, 2005).

The study presented in (EU-US, 2010) shows that the Jahn multipath model (Jahn, 1996) is a good model in urban and suburban environments, while in an open sky environment the multipath error is better modeled with the Mats Brenner multipath model (Brenner, 1998).

The multipath error model is given by:

$$\varepsilon_{mpth,k}^{i} \sim \mathcal{N}\left(0, \left(\sigma_{mpth,k}^{i}\right)^2\right) \tag{2.13}$$

with

$$\sigma_{mpth,k}^{i} = \begin{cases} \max\left\{a + b.e^{c\,\theta^i}\ , e\right\} & \text{in open sky and suburban environments} \\ \max\left\{a + b.\tan^{-1}\left(c\left(\theta^i - d\right)\right), e\right\} & \text{in urban environments} \end{cases} \tag{2.14}$$

where

- $e = 10^{-4}$
- θ^i is the elevation angle in (°).
- a,b,c and d are given for the L1 C/A in Table 2.1:

	Open sky	**Suburban**	**Urban**
a	0.22176	0.55349	6.3784
b	2.2128	30.257	-3.5782
c	-0.057807	-0.23566	0.1725
d	--	--	29.075

Table 2.1 Multipath model parameters

2.1.2.1.6 Total pseudorange error

The total pseudorange error is the convolution of all the independent errors described above. It is modeled consequently as Gaussian distribution with a variance equal to the sum of the variances of each error component:

$$\left(\sigma_{PR,k}^{i}\right)^2 = \sigma_{URA,k}^2 + \left(\sigma_{ionoL1,k}^{i}\right)^2 + \left(\sigma_{tropo,k}^{i}\right)^2 + \left(\sigma_{rcv,k}^{i}\right)^2 + \left(\sigma_{mpth,k}^{i}\right)^2 \tag{2.15}$$

2.1.2.2 Pseudorange rate measurement model

We only consider the nominal pseudorange rate model in this study and we assume that all the error sources mentioned above contribute to a unique error $\dot{\varepsilon}_k^i$ which follows a zero-mean Gaussian distribution:

$$\dot{\varepsilon}_k^i \sim \mathcal{N}\left(0, \left(\sigma_{PRR,k}^i\right)^2\right) \tag{2.16}$$

where $\sigma_{PRR,k}^i$ is the pseudorange rate error standard deviation. Tests carried out in urban environment in (Carcanague, 2013) have shown that the value of $\sigma_{PRR,k}^i$ depends on the vehicle dynamics and the GNSS carrier to noise ratio C/N_0 . Table 2.2 gives the values of $\sigma_{PRR,k}^i$:

	C/N_0 (db.Hz)						
Vehicle speed (m/s)	**30-33**	**33-36**	**36-39**	**39-42**	**42-45**	**45-48**	**48-51**
0-3	0.53	0.43	0.28	0.21	0.13	0.09	0.07
3-6	2.04	1.61	1.08	0.71	0.55	0.32	0.24
6-9	2.37	2.00	1.43	0.86	0.53	0.29	0.18
9-12	3.08	2.33	1.60	1.06	0.99	0.28	0.22
12-15	2.54	1.87	1.25	0.83	0.54	0.27	0.17
15-18	1.26	1.36	1.48	0.80	0.53	0.27	0.15
18-21	1.37	1.29	0.87	0.55	0.44	0.25	0.16
21-24	1.44	1.25	1.09	0.73	0.37	0.26	0.16
24-27	1.33	0.90	0.58	0.47	0.31	0.26	0.11
27-30	1.01	0.92	0.61	0.53	0.31	0.31	0.14

Table 2.2 Standard deviation values of the pseudorange rate

2.1.3 GNSS Navigation solution

2.1.3.1 Position computation using the pseudorange measurement

As described in Section 2.1.1, the GNSS receiver computes the GNSS antenna position using at least four pseudorange measurements. After applying the corrections described in Section 2.1.2, the corrected pseudorange measurement is given by:

$$\rho_{G,k}^i = d_k^i + b_k^{clk} + \varepsilon_k^i \tag{2.17}$$

We remind that:

- $d_k^i = \sqrt{\left(x_k - x_k^i\right)^2 + \left(y_k - y_k^i\right)^2 + \left(z_k - z_k^i\right)^2}$ is the geometric distance between the receiver and the satellite i
- $\boldsymbol{X}_{G,k} = (x_k\,, y_k\,, z_k)$ is the GNSS antenna position in a reference frame and should be determined by the receiver.
- $\boldsymbol{X}_{s,k}^i = \left(x_k^i\,, y_k^i\,, z_k^i\right)$ is the satellite i position in the same reference frame.
- b_k^{clk} is the receiver clock bias and is determined by the receiver. It is therefore included in $\boldsymbol{X}_{G,k}$ that becomes from now on $\boldsymbol{X}_{G,k} = \left(x_k\,, y_k\,, z_k, b_k^{clk}\right)$

The pseudorange measurements are linearized using a Taylor series expansion around an approximate point $\widehat{\boldsymbol{X}}_{G,k} = \left(\hat{x}_k\,, \hat{y}_k\,, \hat{z}_k, \hat{b}_k^{clk}\right)$. The expansion is truncated at the first order giving the following linearized pseudorange expression:

$$\begin{aligned}\rho_{G,k}^{i} = \hat{\rho}_{G,k}^{i} &+ \left.\frac{\partial \rho_{G,k}^{i}}{\partial x_k}\right|_{\widehat{\boldsymbol{X}}_{G,k}} (x_k - \hat{x}_k) + \left.\frac{\partial \rho_{G,k}^{i}}{\partial y_k}\right|_{\widehat{\boldsymbol{X}}_{G,k}} (y_k - \hat{y}_k) + \left.\frac{\partial \rho_{G,k}^{i}}{\partial z_k}\right|_{\widehat{\boldsymbol{X}}_{G,k}} (z_k - \hat{z}_k) \\ &+ \left.\frac{\partial \rho_{G,k}^{i}}{\partial b_k^{clk}}\right|_{\widehat{\boldsymbol{X}}_{G,k}} \left(b_k^{clk} - \hat{b}_k^{clk}\right) + \varepsilon_k^{i}\end{aligned} \tag{2.18}$$

where

- $\hat{\rho}_{G,k}^{i}$ is a prediction of the pseudorange measurement, evaluated in the point $\widehat{\boldsymbol{X}}_{G,k}$
- $\left.\frac{\partial \rho_k^{i}}{\partial x_k}\right|_{\widehat{\boldsymbol{X}}_{G,k}}$ is the partial derivative of the pseudorange measurement with respect to the variable x_k and evaluated in the point $\widehat{\boldsymbol{X}}_{G,k}$.

The development of equation (2.18) for N_{sat} satellites in view yields:

$$\boldsymbol{\rho}_{G,k} - \widehat{\boldsymbol{\rho}}_{G,k} = \boldsymbol{H}_{G,k}\, \delta \boldsymbol{X}_k + \boldsymbol{\varepsilon}_k \tag{2.19}$$

where

- $\boldsymbol{\rho}_{G,k} = \left[\rho_{G,k}^{1}\,, \cdots, \rho_{G,k}^{N_{sat}}\right]^T$ is the vector of the measured pseudoranges
- $\widehat{\boldsymbol{\rho}}_{G,k} = \left[\hat{\rho}_{G,k}^{1}\,, \cdots, \hat{\rho}_{G,k}^{N_{sat}}\right]^T$ is the vector of the predicted pseudoranges evaluated in $\widehat{\boldsymbol{X}}_{G,k}$:
- $\boldsymbol{H}_{G,k}$ is the Jacobian matrix of $\boldsymbol{\rho_{G,k}}$ evaluated in $\widehat{\boldsymbol{X}}_{G,k}$
- $\delta \boldsymbol{X}_k = \left[(x_k - \hat{x}_k)\ \,, (y_k - \hat{y}_k)\ \,, (z_k - \hat{z}_k)\ \,, \left(b_k^{clk} - \hat{b}_k^{clk}\right)\ \right]^T$ is the unknown vector which represents the prediction error vector
- $\boldsymbol{\varepsilon}_k = \left[\varepsilon_k^{1}\,, \cdots, \varepsilon_k^{N_{sat}}\right]^T$ is the vector of residual errors

Using equation (2.19), $\delta \boldsymbol{X}_k$ is estimated using the Iterative Least Squares method. The antenna coordinates and receiver clock bias are then deduced using:

$$\boldsymbol{X}_k = \delta \boldsymbol{X}_k + \widehat{\boldsymbol{X}}_k \tag{2.20}$$

2.1.3.2 Velocity computation using the pseudorange rate measurement

As explained in Section 2.1.1, the GNSS receiver provides also the Doppler measurements from which we can compute the pseudorange rate using equation (2.2). After applying the corrections to equation (2.3), the corrected pseudorange rate measurement model is given by:

$$\dot{\rho}_{G,k}^{i} = \dot{d}_k^{i} + \dot{b}_k^{clk} + \dot{\varepsilon}_k^{i} \tag{2.21}$$

with

$$\dot{d}_k^i = \boldsymbol{V}_{G,k}{}^T \frac{\left(\boldsymbol{X}_{G,k} - \boldsymbol{X}_{s,k}^i\right)}{d_k^i} - \boldsymbol{V}_{s,k}^i{}^T \frac{\left(\boldsymbol{X}_{G,k} - \boldsymbol{X}_{s,k}^i\right)}{d_k^i} \tag{2.22}$$

where

- $\boldsymbol{V}_{G,k} = \left[V_{x_k}, V_{y_k}, V_{z_k}\right]^T$ is the receiver velocity vector
- $\boldsymbol{V}_{s,k}^i = \left[V_{x_k}^{\ i}, V_{y_k}^{\ i}, V_{z_k}^{\ i}\right]^T$ is the satellite i velocity vector
- $\dot{b}_k^{clk}$ is the receiver clock drift
- $\dot{\varepsilon}_k^i$ is the pseudorange rate error

Therefore, if the position is known, we can write:

$$\dot{\rho}_{G,k}^i + \boldsymbol{V}_{s,k}^i{}^T \frac{\left(\boldsymbol{X_k} - \boldsymbol{X_k^i}\right)}{d_k^i} = \frac{\left(\boldsymbol{X_k} - \boldsymbol{X_k^i}\right)^T}{d_k^i} \boldsymbol{V}_{G,k} + \dot{b}_k^{clk} + \dot{\varepsilon}_k^i \tag{2.23}$$

As it can be noticed, equation (2.23) is linear for the receiver velocity components and clock drift, which are the unknowns of the problem. So, the linearization process is unnecessary. The velocity components are therefore estimated using the Least Squares Algorithm.

From now on until the end of this thesis, although the vectors and components are always time-dependent, this dependency on time will not be expressed for sake of simplicity unless it is necessary to do it.

2.2 Inertial navigation

2.2.1 System principle

An *Inertial Navigation System* (INS) is a combination of sensors able to determine all navigation parameters of a moving object, i.e. position, velocity and attitude. The ensemble of sensors is an IMU and consists of three accelerometers, three gyroscopes and eventually three magnetometers mounted on an orthogonal triad. In this study, the magnetometers are not considered since the Earth's magnetic field can be altered by electrical currents, buildings, vehicles or any type of structure using ferrous materials (Groves, 2008).

Accelerometers measure the specific force also called non-gravitational acceleration. It is what people and instruments sense. For example, under zero acceleration, the reaction to gravitation is sensed, and the specific force is equal and opposite to the acceleration due to gravitation (Groves, 2008).

Inertial Navigation is based on the DR principle, which is the process of estimating the current position of the vehicle from a previously determined position knowing the vehicle's motion. In fact, to obtain the velocity at the current time, the specific force corrected of the gravitational term is integrated and added to the previous velocity. The current position is then obtained by integrating the velocity and adding the obtained displacement to the previous position.

Nearly all IMUs fall into one of the two following categories: Stable Platform Systems and Strapdown Systems (Woodman, 2007). Stable platform systems are maintained aligned with the local navigation frame (l) (See Appendix A) in which the position and the velocity are expressed. The gyroscopes detect rotations and this information is fed back to actuators which rotate the platform to keep it aligned with (l). The orientation angles of the vehicle can be read directly. The specific force integration is directly performed in (l).

In Strapdown platforms, the IMU is mounted rigidly onto the vehicle. The angular rates provided by the gyroscopes are integrated to obtain the vehicle's current orientation with respect to (l), knowing its previous orientation. The accelerometers provide the specific force in the vehicle frame also called body frame (b) if the IMU platform is perfectly aligned with (b) (See Appendix A). To obtain the vehicle position and velocity in (l), the specific force is turned first into (l) before integration, using the orientation of (b) with respect to (l).This platform illustrated and detailed in Section 2.2.3.

Strapdown platforms are currently the most widely used configurations (Woodman, 2007) and are used in this work.

2.2.2 Inertial measurement errors and models

The measurements provided by an IMU are affected by errors which degrade their accuracy. In this thesis, the IMU used is an XSENS MTi developed with *Micro-Electro-Mechanical Systems* (MEMS) technology. Details about the errors that arise in a MEMS IMU are given (Woodman, 2007). These errors are summarized by the following:

- Turn-on bias: it is the average of each IMU measurement, obtained during a specific period whilst the sensor is not undergoing any motion. It is generally constant and deterministic.
- In-run bias: it is due to flicker noise in the electronics and in other components susceptible to random flickering (Woodman, 2007). It is in general modeled as a Gauss-Markov process (Angrisano, 2010):

$$\dot{b}_m = -\frac{1}{\tau_{b_m}} b_m + \eta_{b_m} \tag{2.24}$$

  where

  - τ_{b_m} is the correlation time of the process. Its value is set to 100 s which is a typical value for MEMS sensors (Woodman, 2007).
  - η_{b_m} is the process driving noise with a standard deviation σ_{b_m}

- Angle/Velocity Random Walk (ARW/VRW): it is the thermo-mechanical perturbation that affects each measurement of the IMU. It is modeled as a zero-mean white noise (Woodman, 2007). When integrated, this error affecting the IMU specific force and angular rate becomes a zero-mean random walk affecting the velocity and the angle respectively. The ARW/VRW values are provided in the sensor specification datasheet

by the sensor manufacturers. The random walk standard deviations σ_ω and σ_f are derived from the ARW/VRW values using:

$$\begin{aligned}\sigma_\omega(°/s) &= \frac{1}{\sqrt{T_s}} ARW\left(°/s/\sqrt{Hz}\right) \\ \sigma_f(m/s^2) &= \frac{1}{\sqrt{T_s}} VRW\left(m/s^2/\sqrt{Hz}\right)\end{aligned} \tag{2.25}$$

where T_s is the IMU sampling time.

- Scale factor: it is the ratio between the change in the output signal of the sensor and the change in the physical quantity to measure. In ideal conditions the scale factor should be unity however for MEMS IMU, this quantity is never unity and is modeled as Gauss-Markov process (Angrisano, 2010):

$$\dot{k}_m = -\frac{1}{\tau_{k_m}} k_m + \eta_{k_m} \tag{2.26}$$

where

  - τ_{k_m} is the correlation time of the process. The scale factor changes very slowly in time, therefore its correlation time is set to a large value (5 hours for example (Angrisano, 2010))
  - η_{k_m} is the process driving noise with a standard deviation σ_{k_m}

Taking into account these error sources, the IMU measurements can be modeled by the following expression:

$$\tilde{m} = (1 + k_m)m + b_m + \eta_m \tag{2.27}$$

where

- $\tilde{m}$ is one of the measurements outputted by the IMU
- m is the true value of the IMU measurement (if the sensor was ideal without errors)
- k_m is the scale factor affecting the measurement m
- b_m is the bias affecting the measurement m
- η_m is the white random noise affecting the measurement m. Its standard deviation is σ_m given in equation (2.25)

The specifications of the IMU used in this thesis are summarized in the following table:

Error	**Gyroscopes**	**Accelerometers**
Angle/Velocity Random Walk	$0.46\ °/s/\sqrt{Hz}$	$0.007\ m/s^2/\sqrt{Hz}$
Scale Factor	0.05%	

Table 2.3 IMU specifications

Based on Table 2.3 as well as a 10 hours long static data collection, the IMU error parameters are given in Table 2.4. The experimental results are based on the Allan standard deviation analysis (El-Sheimy, 2008) illustrated in Figure 2-2 and Figure 2-3.

	Gyroscopes			Accelerometers		
	Parameter	**Theoretical**	**Experimental**	**Parameter**	**Theoretical**	**experimental**
Bias	$\sigma_{b_{\omega x}}$	N/A	$8.3 \times 10^{-3\circ}/s$	$\sigma_{b_{fx}}$	N/A	$2.6 \times 10^{-4}\ m/s^2$
	$\sigma_{b_{\omega y}}$	N/A	$9.5 \times 10^{-3\circ}/s$	$\sigma_{b_{fy}}$	N/A	$2.1 \times 10^{-4}\ m/s^2$
	$\sigma_{b_{\omega z}}$	N/A	$9.5 \times 10^{-3\circ}/s$	$\sigma_{b_{fz}}$	N/A	$2.6 \times 10^{-4}\ m/s^2$
	$\tau_{b_{\omega x}}$	100 s	N/A	$\tau_{b_{fx}}$	100 s	N/A
	$\tau_{b_{\omega y}}$	100 s	N/A	$\tau_{b_{fy}}$	100 s	N/A
	$\tau_{b_{\omega z}}$	100 s	N/A	$\tau_{b_{fz}}$	100 s	N/A
Scale factor	σ_{k_ω}	5.10^{-4}	N/A	σ_{k_f}	5.10^{-4}	N/A
	τ_{k_ω}	18000s	N/A	τ_{k_f}	18000s	N/A
noise	$\sigma_{\omega x}$	$0.046\ °/s$	$0.054\ °/s$	σ_{fx}	$7.10^{-4}\ m/s^2$	$7.8 \times 10^{-4}\ m/s^2$
	$\sigma_{\omega y}$	$0.046\ °/s$	$0.05\ °/s$	σ_{fy}	$7.10^{-4}\ m/s^2$	$8.1 \times 10^{-4}\ m/s^2$
	$\sigma_{\omega z}$	$0.046\ °/s$	$0.056\ °/s$	σ_{fz}	$7.10^{-4}\ m/s^2$	$9.7 \times 10^{-4}\ m/s^2$

Table 2.4 Comparison of theoretical and experimental IMU errors

Figure 2-2 and Figure 2-3 show that the IMU error is mainly a combination of noise (slope of -1/2), bias (slope of 0) and rate random walk for the accelerometers (slope of +1/2). The first 57 seconds are dominated by the noise which means that the error for the first 57 seconds is due to this source of error. The noise standard deviation corresponds to $\tau = 1\ sec$. From 57 seconds, the bias is the main source of error. Its standard deviation corresponds to the value of σ corresponding to the flat part. The scale factor parameters cannot be determined experimentally because in static conditions, this quantity is not observable. It is the reason why we rely on theoretical values to characterize this type of error.

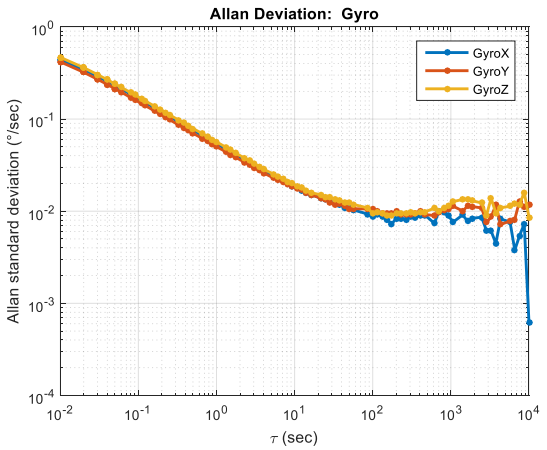


Figure 2-2 Allan variance deviation of the IMU gyroscopes

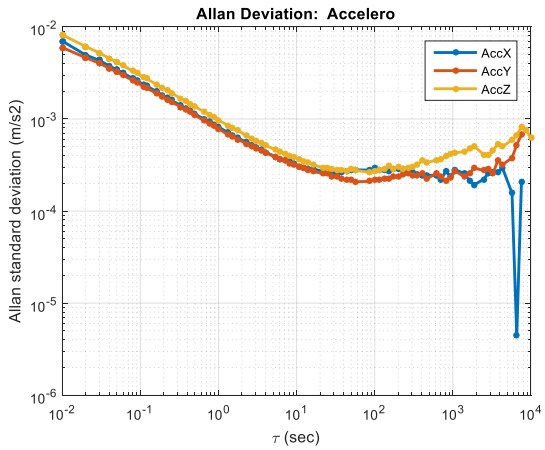


Figure 2-3 Allan variance deviation of the IMU accelerometers

2.2.3 Inertial mechanization

The process of computing the navigation parameters from the inertial measurements provided by the inertial sensors is called the INS Mechanization. In this work, the local navigation frame

considered to implement the mechanization is the East North Up (ENU) Local Tangent Plane (l) defined in Appendix A.

The mathematical development of the mechanization equations is detailed in (Farrell, 1999) and (Shin, 2001). The mechanization equations are given by:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_b^l \\ \dot{\boldsymbol{v}}_b^l \\ \dot{\boldsymbol{C}}_{b2l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_b^l \\ \boldsymbol{C}_{b2l}\boldsymbol{f}_{ib}^b - \left(2\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{el}^l\right) \times \boldsymbol{v}_b^l + \boldsymbol{g}^l \\ \boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{il}^b\right) \times \end{bmatrix} \tag{2.28}$$

where

- $(\cdot) \times$ denotes the skew-symmetric matrix
- $\boldsymbol{p}_b^l = [p_b^E\ , p_b^N\ , p_b^U]^T$ is the position of the vehicle in the navigation frame (l)
- $\boldsymbol{v}_b^l = [v_b^E\ , v_b^N\ , v_b^U]^T$ is the velocity of the vehicle in the navigation frame (l)
- $\boldsymbol{C}_{b2l}$ is the attitude of the vehicle, i.e. the (3×3) matrix expressing the orientation of the vehicle with respect to the navigation frame (l). It is also called the Direction Cosine Matrix (DCM). Its expression as a function of the Euler angles (roll (ϕ), pitch (θ) and heading (ψ),) is given by:

$$\boldsymbol{C}_{b2l} = \begin{bmatrix} \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \cos\theta\sin\psi & \sin\phi\cos\psi - \cos\phi\sin\theta\sin\psi \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\theta\cos\psi & -\sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi \\ -\sin\phi\cos\theta & \sin\theta & \cos\phi\cos\theta \end{bmatrix} \tag{2.29}$$

- $\boldsymbol{f}_{ib}^b = \left[f_{ib}^{bx}\ , f_{ib}^{by}\ , f_{ib}^{bz}\right]^T$ is the specific force vector in the body frame (b) measured by the accelerometers
- $\boldsymbol{\omega}_{ie}^l = [0\ , \omega_e\cos(\varphi)\ , \omega_e\sin(\varphi)]^T$ is the Earth rotation rate with respect to the Earth-Centered-Inertial (ECI) frame (i) (see Appendix A) expressed in the navigation frame (l), where φ is the latitude of the vehicle and $\omega_e = 7{,}2921158.10^{-5}\ rad/s$ is the constant earth rotation rate.
- $\boldsymbol{\omega}_{el}^l = \left[\frac{-v_b^N}{R_M+h}\ , \frac{v_b^E}{R_N+h}\ , \frac{-v_b^E\tan(\varphi)}{R_N+h}\right]^T$ is the rotation rate of the navigation frame (l) relative to the Earth-Centered-Earth-Fixed (ECEF) frame (e), where R_N and R_M are the Earth radii of curvature and h is the vehicle altitude with respect to the reference ellipsoid (WGS84).
- $\boldsymbol{g}^l = \left[0\ , 0\ , \gamma_g\right]^T$ is the gravity vector expressed in the navigation frame (l). γ_g depends on the vehicle's position (latitude and altitude) and can be computed using different gravity models (Farrell, 1999).
- $\boldsymbol{\omega}_{ib}^b$ is the angular rate of the body frame (b) with respect to the inertial frame (i), expressed in the body frame (b). It is the gyros measurements.
- $\boldsymbol{\omega}_{il}^b$ is the rotation rate of the navigation frame (l) with respect to the inertial frame (i), expressed in the body frame (b). It is given by:

$$\boldsymbol{\omega}_{il}^b = \boldsymbol{C}_{b2l}^T\left(\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{el}^l\right) \tag{2.30}$$

The INS mechanization described above is illustrated in Figure 2-4:

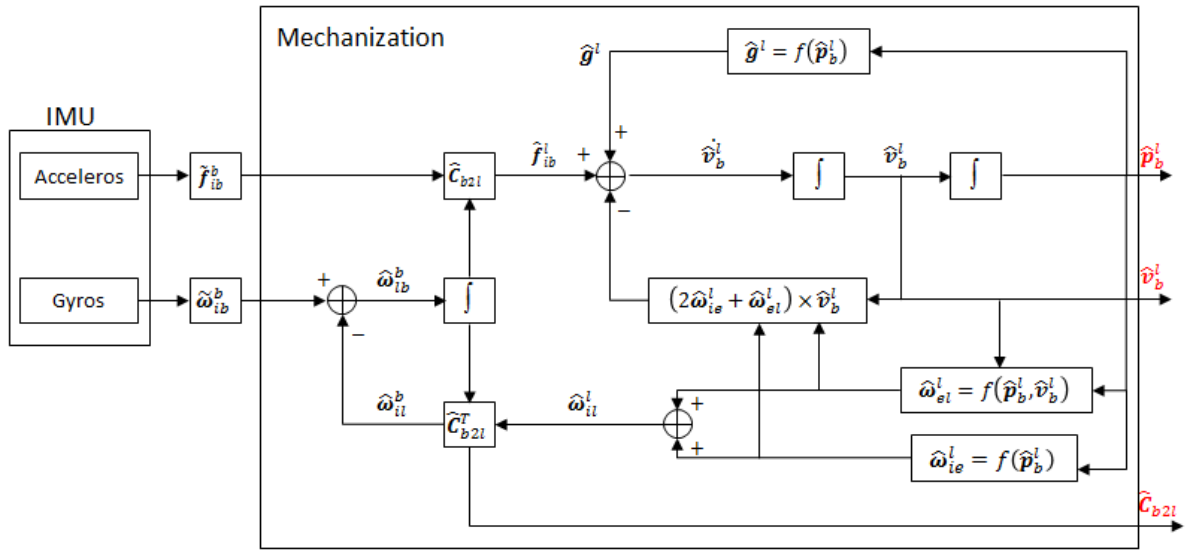


Figure 2-4 INS mechanization in the navigation frame (adapted from [20])

In strapdown INS mechanization, the first step is the use of the vehicle angular rate with respect to the inertial frame $\boldsymbol{\omega}_{ib}^{b}$, provided by the gyros rigidly attached to the vehicle. In order to compute the orientation of the vehicle with respect to the navigation frame $\boldsymbol{C}_{b2l}$, the angular rate of the vehicle with respect to the navigation frame, $\boldsymbol{\omega}_{lb}^{b}$, must be integrated. Therefore, the angular rate component corresponding to the rotation of the navigation frame with respect to the inertial frame $\boldsymbol{\omega}_{il}^{b}$ must be removed from the gyro measurement before integration. The integration is given in details in (Shin, 2001).

Once the vehicle orientation with respect to the navigation frame is determined, the specific force $\boldsymbol{f}_{ib}^{b}$ provided by the accelerometer in the body frame is rotated to the navigation frame and is corrected for the Coriolis and transport rate terms $\left(\left(2\boldsymbol{\omega}_{ie}^{l}+\boldsymbol{\omega}_{el}^{l}\right)\times\boldsymbol{v}_{b}^{l}\right)$ and for the gravity in the local frame $\boldsymbol{g}^{l}$. The expression of $\boldsymbol{g}^{l}$ is given by $\boldsymbol{g}^{l}=[0 \quad 0 \quad \gamma_g]^{T}$ where (Shin, 2001):

$$\gamma_g = a_1(1 + a_2 \sin^2\varphi + a_3 \sin^4\varphi) + (a_4 + a_5 \sin^2\varphi)h + a_6 h^2 \tag{2.31}$$

where

- φ and h are respectively the vehicle geodetic latitude and ellipsoidal height
- $a_1 = 9.7803267715$ $\quad a_4 = -0.0000030876910891$
  $a_2 = 0.0052790414$ $\quad a_5 = 0.0000000043977311$
  $a_3 = 0.0000232718$ $\quad a_6 = 0.0000000000007211$

The obtained acceleration is integrated to compute the vehicle velocity, which is then integrated to get the vehicle position.

2.3 Odometry

2.3.1 Wheel Speed Sensors

Most vehicles are nowadays equipped with Wheel Speed Sensor (WSS) used by the anti-lock braking system (ABS) and traction control system. The ABS is a standard feature in most of the vehicles, providing easily accessible odometric data at no additional cost. WSS provides a measurement of the vehicle velocity. It consists of a magnetic (or optical) pickup and a toothed steel gear (or optically encoded shaft). The pickup counts the rate at which the teeth or marks pass by. To better understand the working principle of WSS, the principle of the magnetic WSS is explained in Figure 2-5.

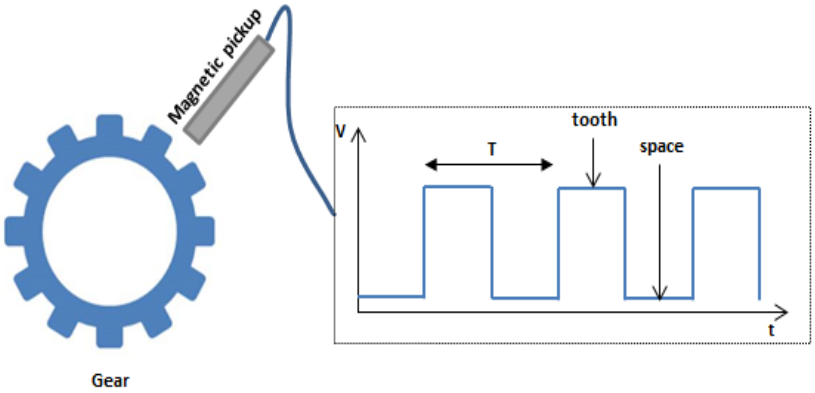


Figure 2-5 Principle of the magnetic WSS

The magnetic pickup placed near the rotating gear detects the change in the surrounding magnetic field. This creates a square wave having a high voltage when the tooth is close to the pickup and a low voltage when the space between two teeth is close to the pickup. The wheel velocity is then calculated using:

$$v_w = \frac{2\pi . R_{wheel}}{N_{teeth} . T_{teeth}} \tag{2.32}$$

where

- N_{teeth} is the number of gear teeth
- R_{wheel} is the wheel radius
- T_{teeth} is the elapsed time between two teeth detection.

2.3.2 Steering angle sensors

The steering angle of the vehicle is the angle between the front of the vehicle and the steered (front) wheels direction. This angle is determined using the measurement of the steering wheel turn that can be recovered from the vehicle Controller Area Network (CAN) bus. The relationship between the steering wheel turn and the steering angle is determined by the steering ratio using:

$$\Delta\delta_s = r_s . \Delta\alpha_s \tag{2.33}$$

where

- $\Delta\delta_s$ is the steering turn
- r_s is the steering ratio. Its value is included in general between 12 and 20 for passengers cars.
- $\Delta\alpha_s$ is the steering wheel turn

2.3.3 Ackermann steering model

A four-wheeled vehicle rolling motion can be described by the 2D Ackermann steering model. This model states that the roll axes of the four vehicle wheels intersect at a point called the Instantaneous Center of Rotation (ICR) (Siegwart, 2011) as shown in Figure 2-6. This point is the center of the circles described by each wheel when the vehicle moves.

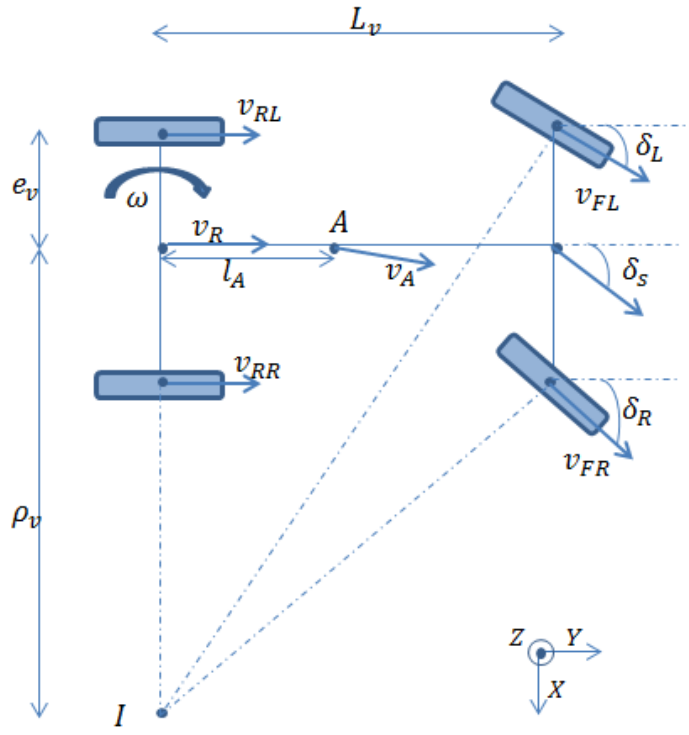


Figure 2-6 2D Ackermann steering model

Let us consider the following definition of the parameters:

- v_{RR} is the rear right wheel velocity of the vehicle
- v_{RL} is the rear left wheel velocity of the vehicle
- v_{FR} is the front right wheel velocity of the vehicle
- v_{FL} is the front left wheel velocity of the vehicle
- v_R is the rear axle center velocity
- δ_s is the steering angle of the vehicle
- ω_v is the heading rate of the vehicle
- L_v is the vehicle wheelbase

- e_v is the distance between one of the rear wheels and the rear axle center
- ρ_v is the distance between the ICR and the rear axle center

Using the Ackermann steering model and its geometric properties, the following relationships between the different motion quantities are derived:

- Velocity of the rear axle center in the body frame:

$$v_R = \begin{bmatrix} 0 & \frac{v_{RR} + v_{RL}}{2} & 0 \end{bmatrix}^T \tag{2.34}$$

or

$$v_R = \begin{bmatrix} 0 & \frac{\Delta_{FL} \,.\cos(\delta_L) + \Delta_{FR} \,.\cos(\delta_R)}{2} & 0 \end{bmatrix}^T$$

where

$$\begin{aligned} \delta_L &= \text{atan}\left(\frac{\tan(\delta_s) \,. L_v}{L_v - e_v.\tan(\delta_s)}\right) \\ \delta_R &= \text{atan}\left(\frac{\tan(\delta_s) \,. L_v}{L_v + e_v.\tan(\delta_s)}\right) \end{aligned} \tag{2.35}$$

- Velocity of any point A of the longitudinal axis of the vehicle (for example the gravity point)

$$\boldsymbol{v}_A = \boldsymbol{v}_R - \boldsymbol{\omega} \times \boldsymbol{l}_A \tag{2.36}$$

where $\boldsymbol{l}_A$ is the lever arm between the rear axle center and the point A

- Angular rate computed using the rear velocities:

$$\omega_v = \frac{v_{RR} - v_{RL}}{2e_v} \tag{2.37}$$

- Angular rate computed using front velocities

$$\omega_v = \frac{v_{FR}\cos(\delta_R) - v_{FL}\cos(\delta_L)}{2e_v} \tag{2.38}$$

- Relationship between the steering angle, the velocity and the angular rate

$$\tan(\delta_s) = L_v \,. \frac{\omega_v}{v_R} \tag{2.39}$$

The advantage of using this four-wheel model is the fact that it describes better the true motion of the vehicle. In addition, compared to the simple differential model, the four-wheeled model takes advantage from the redundancy of the measurements to obtain a more accurate estimate of the vehicle velocity as well as its angular velocity.

2.3.4 Odometric measurement models

In order to extract the odometric measurement models, an analysis of the odometric measurements collected during 55 min using the CAN bus of passengers car, namely the velocity of each wheel and the steering angle, is conducted. As illustrated in Figure 2-7 and Figure 2-8, the heading rate error given by the steering wheel is less noisy than the heading rate provided by the differential odometry.

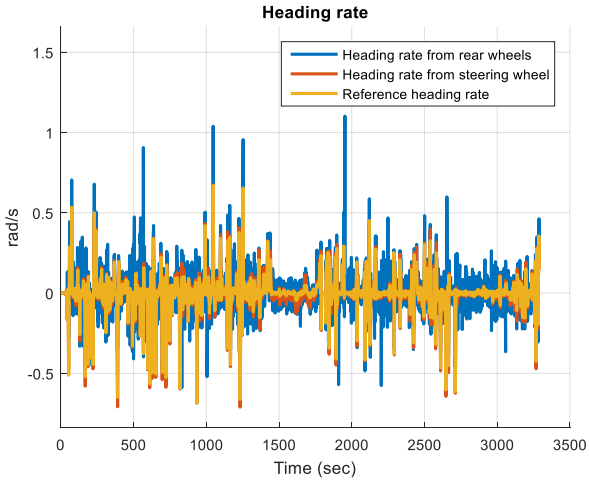


Figure 2-7 Comparison of the estimated heading rate using rear wheels and steering wheel

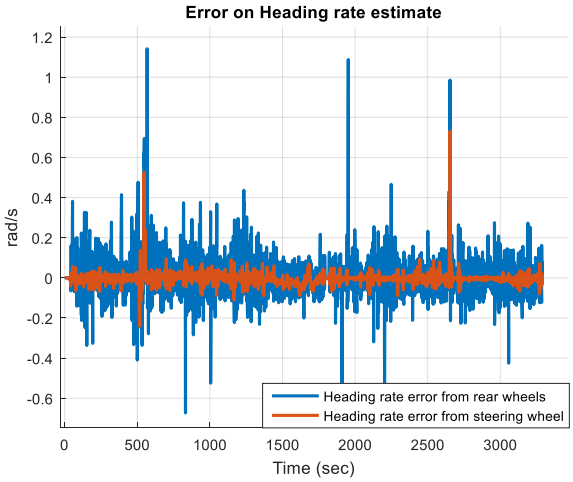


Figure 2-8 Comparison of the estimated heading rate error using rear wheels and steering wheel

Figure 2-8 shows that the steering wheel heading rate can be modeled by the following expression:

$$\omega_s = \omega_{true} + \varepsilon_s \tag{2.40}$$

where

- ω_s is the steering wheel heading rate
- ω_{true} is the true heading rate
- ε_s is zero-mean Gaussian noise with a standard deviation of 0.017 rad/s

As for the velocity provided by the WSSs, the measurements are affected by a scale factor and a noise as shown in Figure 2-9, illustrating, for each wheel, the velocity estimation error as a function of true velocity.

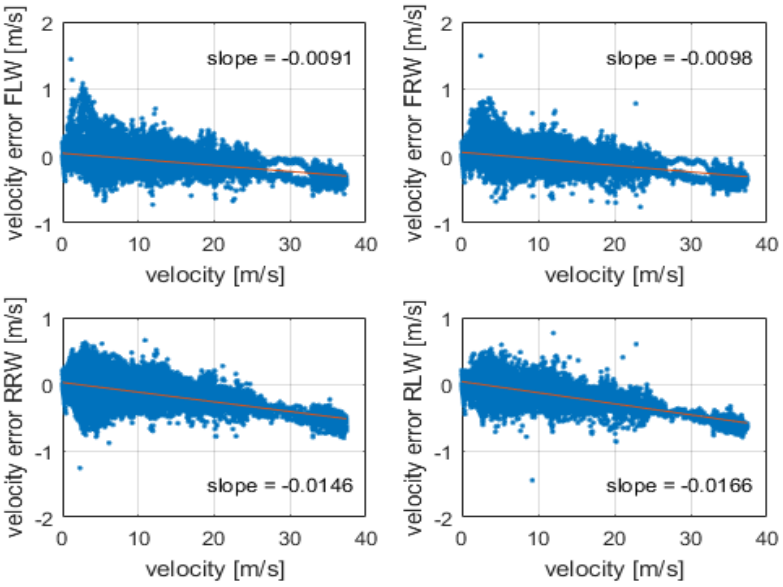


Figure 2-9 Velocity estimation errors for each wheel as a function of true velocity

The value of the scale factor for each wheel velocity is summarized in Table 2.5.

	FLW	FRW	RLW	RRW
Scale factor	-0.0091	-0.0098	-0.0146	-0.0166

Table 2.5 WSS velocity scale factor

The velocity measurement model of wheel speed sensors is consequently deduced:

$$v_w = (1 + k_w)v_{true} + \varepsilon_w \tag{2.41}$$

where

- v_w is the velocity provided by the WSS
- v_{true} is the true velocity
- k_w is a scale factor
- ε_w is a zero-mean Gaussian noise with a standard deviation of 0.5 m/s

After reviewing the principle and the measurement model of each identified sensor to be used in the multi-sensor fusion architecture, the next section will highlight the interest in fusing these sensors and will investigate the different strategies and techniques used for sensor fusion. The state of the art of proposed multi-sensor fusion systems involving GNSS, INS and WSSs is reviewed.

2.4 Sensor complementarity and integrated navigation

The purpose of this thesis is to use low-cost sensors to form a system capable of achieving high navigation performances in constrained environments. The idea is therefore to benefit from each sensor's advantages and to avoid each sensor's weakness by fusing the data provided by

the available sensors. The combination of sensors chosen in this thesis is a trade-off between cost and complementarity.

For example, inertial navigation has the advantage of operating autonomously and continuously, providing a high-bandwidth output and exhibiting low short-term noise. The fact of being autonomous makes this navigation technique invulnerable to external disturbances. However, inertial navigation is based on a DR principle requiring initialization and implying the navigation solution to drift due to the error accumulation, especially in our case of study where low-cost IMU should be used.

On the other hand, GNSS provides high long-term navigation solution accuracy with small errors in open-sky environments and relatively large errors in constrained environments such as urban environments. This dependence on the environment is due to the signal propagation which is prone to interference, reflection and even blockage. In fact, GNSS measurements suffer in urban environments from blockage reducing the number of satellites in view and resulting either in poor geometry degrading the navigation performance, or even in the unavailability of GNSS if the number of satellites in view is less than 4. The problem of unavailability should be reduced in the near future with the development of new constellations such as GALILEO, BEIDOU, etc. The principal issue in urban environments is signal reflections which result in significant positioning error. When the Line-Of-Sight (LOS) signal is reflected and received with the reflected signal, this is known as multipath. If the LOS signal is blocked and only its reflection arrives to the receiver antenna, this is known as Non-Line-Of-Sight (NLOS) signal. The latter error results in a positive pseudo-range measurement error equal to the additional path delay and its value may exceed in some cases a kilometer (Jiang, 2012). Many techniques are proposed in the literature in order to mitigate these problems and improve the GNSS accuracy (Jiang, 2012). Unfortunately, all these techniques have limitations as highlighted in (Jiang, 2012).For all these reasons, GNSS cannot be relied upon to provide a continuous navigation solution.

As for odometry, it has the advantage of benefiting from information that already exists in almost all the new vehicles and consequently it does not induce an additional cost. However, the odometric solution suffers from the dead-reckoning drawbacks mentioned earlier for the INS.

A possible way to overcome these problems of each sensor taken individually and to benefit from their complementary advantages is to fuse their information together. In fact, by exploiting the complementarity of sensors, hybridization algorithms can improve the navigation solution compared to solutions provided by each stand-alone sensor.

2.4.1 Integration techniques and strategies

The integration of the complementary data mentioned previously commonly uses filters. The most widely used filter is the *Extended Kalman Filter* (EKF) (Welch, 2006), (Kalman, 1960). However other filters, such as the *Unscented Kalman Filter* (UKF) (Wan, 2000), (Julier, 1997) and the *Particle Filter* (PF) (Gustafsson, 2002), are also used.

The UKF is a nonlinear adaptation of the KF not requiring the linearization process by Jacobian computation. Instead, the UKF approximates the state distribution by capturing its mean and covariance using a minimal set of carefully chosen sample points called sigma points. The filter propagates each point through the nonlinear process and measurement models, and then computes the weighted averages of the transformed points to determine the posterior state mean and state covariance. Despite its capacity to provide good results (Julier, 1995), the UKF is slightly more computationally expensive than the EKF because of the need to compute sigma points.

The PF is a type of sequential Monte Carlo estimation algorithm. The state estimates are represented as a set of discrete state vectors, known as particles, which are spread throughout their joint probability distribution (Groves, 2008). The main advantage of the PF compared to the KF is the possibility to overcome the constraint of Gaussianity generally needed for KFs to provide an optimal estimation. In addition, the PF is robust to non-linearities. However, this filter has the disadvantage of the high computational cost resulting from the high number of particles required to reach a good accuracy.

The most widely used filter in the literature is the EKF which uses the linearization required to model non-linear systems in the KF. The linearization process can degrade the filter performance in high dynamics. However, in constrained environments such as urban environments, the vehicle dynamics are in general low and the hazardous effect of the linearization process is negligible. In addition, this filter is efficient in terms of computational cost, especially for applications where there are a large number of states and observations compared to the other filters (Hide, 2007).

We generally consider three different ways to combine sensor measurements to estimate the navigation solution, which vary depending on the information exchanged by the sensors and the integration engine: loose-coupling, tight-coupling and deep-coupling. With loose-coupling, each sensor provides a positioning solution. The hybridized solution is the combination of different solutions provided by each sensor individually. With tight and deep coupling, fusion of sensor information takes place before positioning solution estimation. Tight architectures combine the sensors information to provide a unique solution, while in deep architectures, the sensors work as a unique system and fusion is performed inside one of the sensors processes. Detailed descriptions of these various architectures are found in (Kaplan, 2005), (Groves, 2008), (Skog, 2009).

In all the cases, two possible forms of integration can be considered. The direct form (also referred as the total state form) and the indirect form (also referred as the error state form). As the name indicates, in the total state formulation, the variables estimated by the fusion filter are the navigation parameters, and the measurements presented to the filter are the GNSS measurements. However, the error state form considers the errors of the navigation parameters. As highlighted in (Roumeliotis, 1999), the error state formulation is preferred, because the dynamics upon which this formulation is based, are low frequency, and consequently very adequately represented as linear. However, with the total state formulation, the vehicle dynamics include a high frequency component and can only be described by non-linear model.

In the case of indirect formulation, the integration filter can be designed to be feedforward or feedback as detailed in (Roumeliotis, 1999). The basic difference between the two configurations is mainly the way they handle the updated error estimate. In the first case, the updated error estimate is fed forward to correct the current state estimate without updating the dead-reckoning system. This system is not at all aware of the existence of an aiding system. In the feedback configuration, the correction is actually fed back to the dead-reckoning system to correct its "new" starting point as well as its "raw measurements" if they exist. The feedforward configuration is equivalent to the Linearized Kalman Filter (LKF) while the feedback configuration is equivalent to the EKF, as explained in (Roumeliotis, 1999). Indeed, in the EKF, the state propagation starts from the corrected state right after a measurement while in the LKF the propagation continues at the state that the propagation has reached when the measurement appeared, thus ignoring the correction just computed. The LKF and the feedforward configuration are thus free to drift with unbounded errors, and the assumption of small errors used in the linearization process of the filter can be violated, especially in the case of low-cost sensor.

2.4.2 State of the art of existing ground vehicle integrated navigation based on INS, GNSS and/or odometry and limits of the existing solutions

Generally, the most widely implemented hybridization algorithms for land vehicles fuse GNSS measurements with inertial and/or odometric data. Thereby, these dead-reckoning sensors ensure the system continuity when GNSS information is unavailable and improve the system performance when GNSS signals are severely corrupted, and in turn, the GNSS limits the drift of the dead-reckoning solution if it is available. In general, the fusion is done so that the dead-reckoning system errors are estimated and corrected by the GNSS. Varied sensor configurations can be utilized in the dead-reckoning system. Many studies in the literature focus on fusing INS with GNSS (Farrell, 1999), (Angrisano, 2010), (Shin, 2001). In general, the performance achieved by this hybridization depends thoroughly on the quality of the INS used especially when GNSS signals are degraded or unavailable.

Reduced inertial systems have been introduced in order to adapt this sensor to terrestrial applications and to reduce the cost of INS (Bhatti, 2008), (Li, 2010). This consists of using a reduced number of accelerometers and/or gyroscopes instead of a six-degrees of freedom IMU. An analysis performed in (Syed, 2007) compares the performance of five different configurations ranging from six sensors to only two sensors. Full sensors configuration gave the best results with minimum average drift. In addition, low-cost MEMS are no longer expensive, and the reduction of the number of sensors degrades further the performance especially with high levels of nonlinearity of MEMS sensors.

One way to tackle this issue is to integrate more sensors such as wheel speed, steering encoders, etc. as performed for instance in (Gao, 2007), (Spangenberg, 2007). However, the performance of these techniques is limited by the quality of GNSS and its availability.

2.5 Conclusion

This chapter has presented an overview of GNSS, INS and wheel odometry navigation systems. The technical fundamentals and the principle of each navigation technique are presented, and the error models are provided and analyzed. Then, each sensor advantages and limits are detailed in order to highlight the complementary characteristics of these sensors and emphasize the need of fusing them in a multi-sensor solution. A review of the strategies and techniques used to fuse information is performed. The advantages and drawbacks of each technique are discussed.

Chapter 3

Vision-based Navigation for Ground Vehicle Navigation in Outdoor Environments

In Chapter 2, the limits of the existing navigation solutions in terms of accuracy and robustness are detailed especially in challenging environments. The state of the art solutions are not sufficient to guarantee the required level of performance defined in Chapter 1. Therefore, other sensors have been considered to find a compromise between accuracy and cost, even in challenging environments.

Many sensors have been proposed and tested in the literature. Among them we can find the sensors providing range information such as the *Light Radar* (LIDAR) (Gao, 2015), the laser rangefinders (Hiremath, 2014) and the ultra-sound sensors (Nordh, 2007). The LIDAR has very good performances but its cost is relatively high for car applications. Ultra-sound sensors are also not adapted to car applications in large scale environments because of their short range. The laser rangefinders are more adapted to car application. All these sensors have the drawback of only providing range information with respect to surrounding objects.

Recently, it has been proven that vision could be a promising navigation sensor that provides accurate navigation solution (Nistér, 2006). This chapter reviews the existing vision-based navigation techniques and highlights their challenges especially in outdoor large-scale environments. First, a classification of the different manners these techniques have been used in the literature is proposed. Then the review of the technical aspects of vision-based positioning is performed. A particular interest is given to ORB-SLAM (Mur-Artal, 2015) open-source software, used in this study to provide the visual information. Then, the challenges of these techniques are emphasized. Finally, the measurement model of the visual information is given.

Contents

3.1 State of the art of vision-based navigation techniques 32
3.1.1 Classification of vision-based navigation techniques 32
3.1.2 Vision-based Navigation fundamentals 34
3.2 ORB SLAM 46
3.2.1 Selection of ORB-SLAM 46
3.2.2 ORB-SLAM characteristics 46
3.3 Challenges of vision-based navigation in outdoor environments 48
3.4 VSLAM measurement model 49
3.5 Conclusion 51

3.1 State of the art of vision-based navigation techniques

There has been extensive research in the field of vision-based navigation in the last decades that yielded to workable systems in various navigation applications such as land vehicle navigation, drone navigation, robotics, etc. Different vision-based techniques have been proposed in the literature and the review of the state of the art of these techniques is important to understand their fundamentals and issues, and to be able to identify the advantages and limits of each technique. A classification of these vision-based techniques comes out of this state of the art review.

3.1.1 Classification of vision-based navigation techniques

Figure 3-1 summarizes the vision-based navigation techniques found in the state of the art. It shows that vision-based techniques can be divided into two categories: Navigation in a known environment and navigation in an unknown environment.

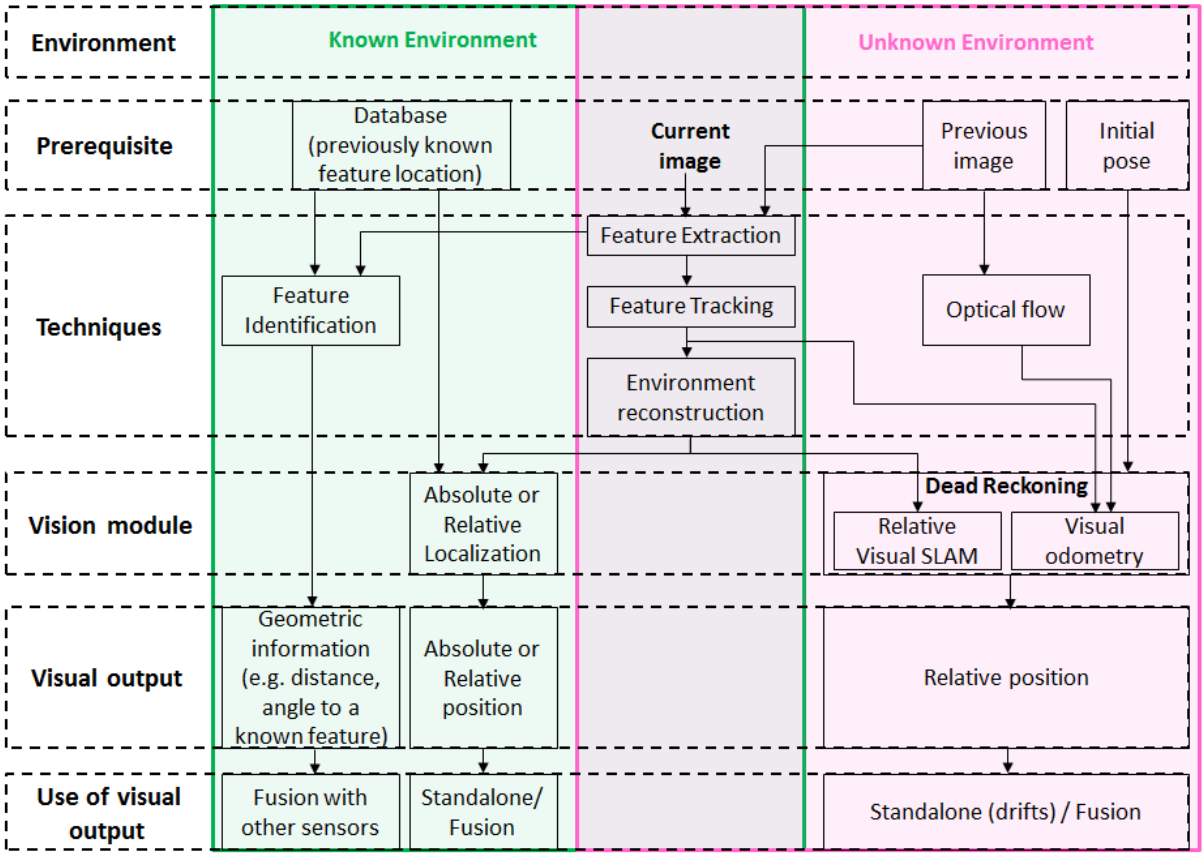


Figure 3-1 Classification of vision-based techniques

3.1.1.1 Navigation in a known environment

The first category of vision-based navigation techniques is performed in a known environment which means that a database of features [1] (also called map of features) describing the environment in which the vehicle moves has been previously established, before the navigation starts. In this case, two ways of using visual information are possible: the first consists in capturing the current image, extracting features and associating the extracted features with the database ones. This provides geometric information such as the distance of the camera on the vehicle to the feature, or the bearing angle to the feature. This information is then used as aiding information in a multi-sensor system as proposed in (Raquet, 2003).

The second way of navigating in a known environment consists in performing localization by extracting features from the current images and tracking them among the established database. These features are then used to deduce the position of the camera, based on the knowledge of the tracked features and the database characteristics. If the database is given in an absolute coordinate system, then the navigation parameters provided by the vision module are absolute and could be used alone or integrated in a fused architecture. However, if the database is given in a relative coordinate system, then the information provided by the vision module is relative and should be fused with other information to provide an absolute navigation solution.

[1] A feature also called a landmark is a salient zone in the image such as a corner or an edge.

This type of navigation has the advantage of relying on previously collected information that could be controlled (for example with the knowledge of the features position) and optimized. However this requires the previous establishment of database which can be heavy and complex.

3.1.1.2 Navigation in an unknown environment

If the environment is unknown, vision-based navigation is based on the DR principle illustrated in Figure 3-2. The main techniques used to perform navigation in unknown environments are VO and relative SLAM that we will be detailed in Section 3.1.2.3. Using these techniques, camera motion is computed based on the comparison of a sequence of successive images I_k. Pose estimation is performed with respect to the vision frame (v) (see Appendix A) and requires the knowledge of the initial pose[2].

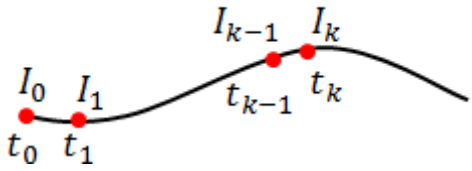


Figure 3-2 Trajectory estimation

The navigation outputs of the vision module based on the DR principle drifts in general. Therefore, it cannot be used as an accurate standalone navigation module. Integration with other sensors is therefore necessary.

The advantage of this approach, compared to navigating in known environment, is its independence of a priori information about the environment, but at the cost of accuracy.

All the previously mentioned techniques are based on vision-based navigation fundamentals that will be explained in the following section.

3.1.2 Vision-based Navigation fundamentals

3.1.2.1 The vision-based navigation problem formulation

The goal of a vision navigation module is to determine the motion of the vehicle to which a camera is rigidly attached. Before formulating the motion problem, the type of camera and the way it is mounted on the vehicle should be determined first.

The review of the state of the art of vision-based navigation shows that many types of cameras are used. The most widely used types of cameras are monocular cameras[3], stereo cameras[4] and omnidirectional cameras[5].

In general, vision systems reach accurate results when using stereovision. However, the main drawback of such a configuration compared to a single camera is that in large-scale

[2] The pose is defined by the combination of the position and attitude information
[3] A monocular camera is a single camera
[4] A stereo camera is the set of two cameras with overlapping fields of view
[5] An Omnidirectional camera is a camera with a 360° field of view

environments such as streets, the images captured by the cameras might contain objects that are placed too far. Processing these images does not allow recovering the depth values unless the stereo camera baseline is of few meters (Hong, 2012). In addition to this compactness issue, a calibration issue arises when a multi-camera system is used and the calibration of a single camera is much easier (Nützi, 2010). Therefore, a single camera will be considered in this study.

The camera can be mounted on the vehicle in four possible ways: looking forwards such in (Nistér, 2006), (Mur-Artal, 2015), looking downwards (Nourani-Vatani, 2011), looking upwards (Hwang, 2011) and tilting downwards (Hong, 2012) (Campbell, 2005). The forward-looking camera provides a high number of features thus allowing good motion estimation accuracy, if sufficient number of features are close to the camera. This is possible in general in highly textured environments such as indoors environment or outdoor urban environments. However, in large-scale environments, the depth information is badly recovered. Additionally, in poor illumination, this method is not feasible. These problems are handled using a downward facing camera looking at the ground texture, with a light source next to the camera, using the optical flow technique. However, downward-facing camera performance is degraded as soon as the vehicle moves rapidly. The tilted camera provides a larger field of view than the downward-facing camera and captures the ground texture, but captures also the vehicle shadow which looks stationary and provides erroneous information about the vehicle motion. Upward-looking cameras are mainly used in indoor environments and exploit the characteristics of this environment such as corners, lamps and door features, but are not adapted to outdoor environments because of the lack of features. Therefore, the most adapted technique to our case of study (a ground vehicle in outdoor environments) is the forward-looking camera.

3.1.2.2 Camera modeling and calibration

Before tackling the fundamentals of vision-based navigation, three frames should be defined: the image frame (I) defining the 2-dimensions image captured by the camera, the camera frame (c) attached to the camera, and the vision frame (v) (also called world frame by the computer vision community) that defines the environment in which the camera moves. More details about these frames are given in Appendix A.

To estimate the 3D camera motion from 2D images, vision techniques are usually based on narrow field-of-view cameras modeled using the *Pinhole Projection model* described in Figure 3-3. If the camera has large field-of-view, then a different projection model called the omnidirectional model is used. This model, addressed in (Scaramuzza, 2006) is used for cameras having a field-of-view within a range of 180°-360°. In this study, only narrow field-of-view cameras are considered and the pinhole model is used.

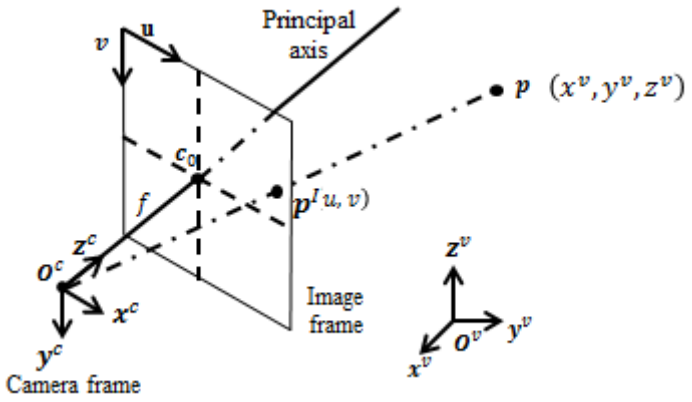


Figure 3-3 Pinhole camera projection

The relationship between the coordinates of a physical 3D point $\boldsymbol{p}^v = [x^v \quad y^v \quad z^v]^T$ expressed in the vision frame (v) and its projection in the image plane $\boldsymbol{p}^I = [u \quad v]^T$ expressed in the image frame (I) is given by:

$$k_c \, \boldsymbol{p}_h^I = \boldsymbol{\Pi} \, \boldsymbol{p}_h^v \tag{3.1}$$

Where k_c is a scale factor resulting from unknowing the depth of the scene using a single camera, $\boldsymbol{p}_h^I = [u \quad v \quad 1]^T$ and $\boldsymbol{p}_h^v = [x^v \quad y^v \quad z^v \quad 1]^T$ are the homogeneous coordinates of $\boldsymbol{p}^I$ and $\boldsymbol{p}^v$, and $\boldsymbol{\Pi}$ is a (3×4) projection matrix. The homogeneous coordinates are used in order to express the projection as a linear transformation. The projection matrix depends both on camera intrinsic and extrinsic parameters. Intrinsic parameters do not depend on the camera location but rather on the internal camera parameters such as the focal length f, the number of pixels per distance unit in $\boldsymbol{u}$ and $\boldsymbol{v}$ directions k_u and k_v, the skew factor s_γ which equals zero if and only if the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions are perfectly orthogonal, and the image pixel coordinates of the intersection between the optical axis and the image plane called the principal point $\boldsymbol{c_0} = (u_0, v_0)$. These intrinsic parameters define the *Calibration Matrix* of the camera $\boldsymbol{K}_c$, which expresses the transformation between the camera frame and the image frame. The expression of the calibration matrix is given by:

$$\boldsymbol{K}_c = \begin{pmatrix} k_u f & \mathrm{s}_\gamma & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

The camera calibration process is based on the estimation of $\boldsymbol{K}_c$, and is generally performed offline. To compute the calibration parameters, several pictures of a planar checkerboard are taken at distinct positions and orientations. The intrinsic and extrinsic parameters are then estimated using a least square minimization method. The camera extrinsic parameters depend on the camera location in the vision frame and correspond to the Euclidean relationship between this frame and the camera frame. This relationship is defined by a (3×3) rotation matrix $\boldsymbol{C}_{c2v}$ and a (3×1) position vector $\boldsymbol{p}_c^v$ expressed in the vision frame, where $\boldsymbol{C}_{c2v}$ and $\boldsymbol{p}_c^v$ define the camera pose. This means that given the coordinates $\boldsymbol{p}^c$ and $\boldsymbol{p}^v$ of a 3D point $\boldsymbol{p}$ in the camera and vision coordinates frames respectively, then:

$$\boldsymbol{p}^{v} = \boldsymbol{C}_{c2v}\,\boldsymbol{p}^{c} + \boldsymbol{p}_{c}^{v} \tag{3.3}$$

Since $\boldsymbol{C}_{c2v}^{-1} = \boldsymbol{C}_{c2v}^{T}$, then:

$$\boldsymbol{p}^{c} = \boldsymbol{C}_{c2v}^{T}(\boldsymbol{p}^{v} - \boldsymbol{p}_{c}^{v}) \tag{3.4}$$

Therefore, the projection matrix of a world point in the image plane using the pinhole model is given by:

$$\boldsymbol{\Pi} = \boldsymbol{K}_{c}\,[\boldsymbol{C}_{c2v}^{T} \quad -\boldsymbol{C}_{c2v}^{T}\boldsymbol{p}_{c}^{v}] \tag{3.5}$$

As explained in (Scaramuzza, 2011), many camera calibration toolboxes are available. The most popular ones are the Camera Calibrator application provided by the Computer Vision toolbox of Matlab (Bouguet, 2015) and the open-source Camera Calibration implementation of OpenCV (OpenCV, 2014).

3.1.2.3 Vision as a standalone navigation means

The use of vision as a standalone navigation tool has proven to be promising (Nistér, 2006). It has also been demonstrated that, compared to wheel-odometry navigation, vision-based navigation provides more accurate trajectory estimates, with relative position error ranging from 0.1% to 2% (Scaramuzza, 2011). However, visual navigation works effectively only if three conditions are fulfilled. Firstly, there should be sufficient illumination in the environment surrounding the camera unless the camera is infrared. However, infrared camera has the disadvantage of being too sensitive to sun light. Secondly, there should be enough static features in the scene filmed by the camera to deduce the camera motion with respect to these features. Finally, consecutive frames should be captured such as they sufficiently overlap to recognize the static features already seen and deduce the camera motion from it. This may be achieved using a high camera frame rate but requires a powerful processor if real-time applications are targeted.

As explained in Section 3.1.1, vision navigation can be performed using these following techniques:

- Visual Odometry (VO) which is the process of incrementally estimating a device pose by examining the changes that motion induces on images taken by its on-board camera(s) (Scaramuzza, 2011), without reconstructing the 3D scene surrounding the camera
- Visual Simultaneous Localization and Mapping (VSLAM[6]) which is the process of estimating the map of the environment filmed by the camera and at the same time deducing the camera pose in this environment.
- Absolute or relative localization if the environment has been already explored and a feature database has been already constructed. This task is performed thanks to the localization processing part of VSLAM and can be included in VSLAM.

[6] In the remaining parts of this thesis, VSLAM and SLAM refer both to visual SLAM

VO can be considered as a building block of a complete VSLAM as stated in (Scaramuzza, 2011). It consists of the pose estimation task of VSLAM. VSLAM includes, in addition to VO, the mapping task consisting of keeping track of the environment map and optimizing simultaneously this map and the camera pose.

The following sections give an overview on VO and VSLAM.

3.1.2.3.1 Visual odometry

Based on a sequence of images taken by a camera, VO generally relies on 3 main steps in order to estimate the trajectory of the camera by estimating its displacement: establishing matches between two successive images, removing outliers and estimating the motion that occurs between two image captures. These steps are illustrated in Figure 3-4 and described in the following steps:

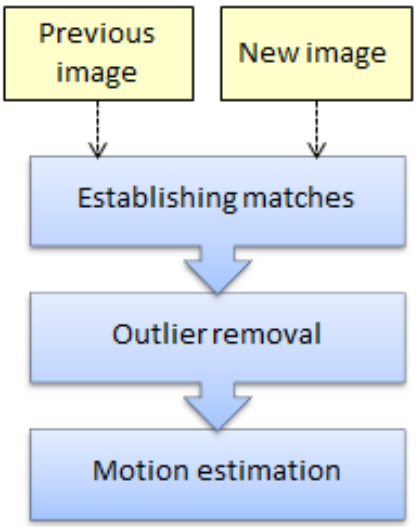


Figure 3-4 Visual odometry diagram

- **Establishing matches**

In this step, a comparison between two successive images is performed by matching their similar characteristics. This is performed either by using the *Optical Flow* (OF) technique or the *Feature Tracking* (FT) one.

- **Optical flow**

It is also called direct method or dense method. OF exploits the information available at each pixel in the image to estimate the parameters that fully describe the camera motion as explained in Figure 3-5. The OF between two consecutive frames is represented by a set of vectors, one for each pixel, where their norm depends on the motion speed and their directions represent the movement of the corresponding pixel in consecutive images. To estimate the OF at all pixels, the *Intensity Constancy* assumption is used. This assumption consists of considering that the intensity of a small image region remains constant with time, and is formulated by:

$$I(u, v, k-1) = I(u + \delta u, v + \delta v, k) \quad (3.6)$$

where $I(u, v, t-1)$ is the (u, v) point intensity at time epoch $(k-1)$ and $(\delta u, \delta v)$ is the displacement in the image plane of this point between time epochs $(k-1)$ and k. Many approaches have been proposed in the literature to solve this equation and find the displacement$(\delta u, \delta v)$. These approaches are reviewed and assessed in (Barron, 1992).

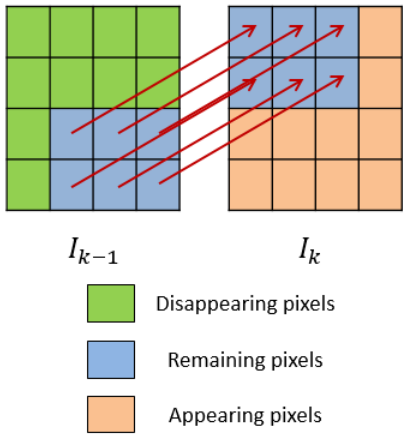


Figure 3-5 Optical flow matching

- **Feature tracking**

It is also called indirect method or sparse method. FT only exploits the information of only salient image locations such as corners or edges, which are called features or landmarks. The selection of these features is performed using feature detectors. These detectors must satisfy a tradeoff between feature quality (invariance to viewpoint and illumination changes and robustness to blur and noise) on one hand, and computational speed on the other hand. The most widely used detectors are the Harris detector (Harris, 1988), the Shi and Thomasi detector (Shi, 1994), and the FAST detector (Rosten, 2006).

Once detected, these features have to be characterized in a unique way. This second step in feature tracking is performed through feature description. A feature descriptor includes all feature characteristics. A very large number of feature descriptors have been developed. The most widely used algorithms are Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Feature (SURF) (Bay, 2008). Evaluations and comparison of feature detectors and descriptors can be found in (Gauglitz, 2011), (Hartmann, 2013) and (Hietanen, 2016).

The final step is Feature Matching consisting is associating features describing the same physical point in the successive images. This step can be performed in two ways: either by extracting features from both previous and current images independently, then performing matching, or by extracting features from the previous image, predicting the regions where the features could be found in the current image and then performing matching. The first approach is more suitable for large-scale environments while the second is generally used in small-scale environments. This phase consists of computing a matching score that indicates the likelihood that two features correspond to the same physical feature. The features that have the highest

scores are matched. The matching process can also be performed considering the computation of the distance between the descriptors of the features. In this case, the features having the smallest distances are matched. Figure 3-6 illustrates feature detection, description and matching.

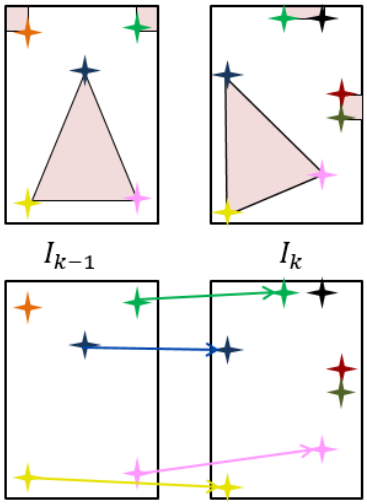


Figure 3-6 Feature matching

- **Comparison of Optical Flow and Feature Tracking**

As stated in (Younes, 2017), each of the two methods has its advantages and drawbacks. Optical flow is more robust than feature tracking since it exploits all the information available in the image even if some image regions are poorly textured. Nevertheless, optical flow is susceptible to failure if the intensity constancy assumption is violated. A second disadvantage is that applying the optical flow method for every pixel is computationally intensive; therefore, real-time applications were not feasible until recently where (Engel, 2014) and (Concha, 2015) proposed algorithms based on the direct method. Methods combining direct and indirect methods to benefit from their advantages have been recently proposed in the literature (Forster, 2014).

- **Outlier removal**

The second step of VO is a process applied to exclude wrong matches usually called outliers. These wrong matches can cause significant error in estimating camera motion, and removing them is necessary to obtain accurate motion estimation. To exclude wrong matches, the algorithm usually used is RANdom SAmple Consensus (RANSAC) (Choi, 2009), based on the Epipolar geometry constraint described in Figure 3-7.

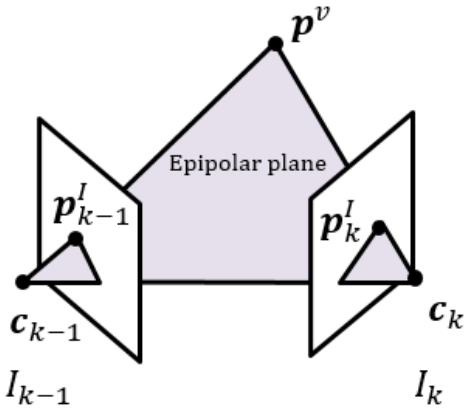


Figure 3-7 Epipolar geometry constraint

Epipolar geometry defines the geometrical constraints between two different views of the same 3D point. Denoting $\boldsymbol{p}^I_{k-1}$ and $\boldsymbol{p}^I_k$ the image points corresponding to the same 3D point $\boldsymbol{p}^v$ viewed from different locations $\boldsymbol{c}_{k-1}$ and $\boldsymbol{c}_k$ (since the camera moves), then $\boldsymbol{p}^I_{k-1}$, $\boldsymbol{p}^I_k$, $\boldsymbol{p}^v$, $\boldsymbol{c}_{k-1}$ and $\boldsymbol{c}_k$ should all lie in the same plane (Hartley, 2004). Based on this principle, all corresponding image points satisfy the coplanarity equation:

$$\boldsymbol{p}^{I\,T}_k \boldsymbol{F}^c_k \boldsymbol{p}^I_{k-1} = \boldsymbol{0} \tag{3.7}$$

where $\boldsymbol{F}^c_k$ is the *Fundamental Matrix* at time k. $\boldsymbol{F}^c_k$ expresses the camera motion from $\boldsymbol{c}_{k-1}$ to $\boldsymbol{c}_k$ and the internal camera calibration. Each pair of points $\boldsymbol{p}^{\mathrm{I}}_{k-1}$ and $\boldsymbol{p}^I_k$ adds a constraint on the fundamental matrix estimation. The normalized 8-point algorithm described in (Hartley, 2004) is used to estimate $\boldsymbol{F}^c_k$ using RANSAC algorithm for robustness against outliers. This means that to estimate $\boldsymbol{F}^c_k$, at least 8 pairs of points must be correctly matched. Points not satisfying the Epipolar geometry constraints defined by equation (3.7) are considered as outliers and are excluded.

If the camera is calibrated, i.e. the calibration matrix $\boldsymbol{K}_c$ is known, then the coplanarity equation is given by:

$$\overline{\boldsymbol{p}}^{I\,T}_k \boldsymbol{E}^c_k \; \overline{\boldsymbol{p}}^I_{k-1} = \boldsymbol{0} \tag{3.8}$$

Where $\overline{\boldsymbol{p}}^I_{k-1} = \boldsymbol{K}_c{}^{-1}\boldsymbol{p}^I_{k-1}$, $\overline{\boldsymbol{p}}^I_k = \boldsymbol{p}^I_k\left(\boldsymbol{K}_c{}^T\right)^{-1}$ and $\boldsymbol{E}^c_k$ is the *Essential Matrix* at time epoch k. The essential matrix is the specialization of the fundamental matrix to the case of known calibration parameters. At least 5 pairs of points must be correctly matched to estimate $\boldsymbol{E}^c_k$, using the 5-point algorithm addressed in (Nistér, 2004). This algorithm requires at least 5 good matches to estimate $\boldsymbol{E}^c_k$. If more good matches are available, then the system described by equation (3.8) is overdetermined and is solved in the least squares sense, providing consequently more robustness to noise.

- **Motion estimation**

Once outliers are determined and excluded using Epipolar geometry constraints and the fundamental/essential matrix computed, the camera motion between the current image and the

previous image can be estimated based on the matches established in the previous steps. If the intrinsic parameters are unknown, then only the fundamental matrix is estimated. In this case, the camera displacement is recovered up to a projective transformation, and it is not possible to recover the distance ratios and the angles. That is why, in most of VO applications, the camera is calibrated. In this case, the essential matrix is computed using equation (3.8). It has been shown in (Hartley, 2004) that the essential matrix can be decomposed into a rotation and a translation vector such as:

$$\boldsymbol{E}_k^c = \left(\left(\boldsymbol{p}_{k-1,k}^{\boldsymbol{c}_k} \right) \times \right) \boldsymbol{C}_{k-1,k} \tag{3.9}$$

where

- $\boldsymbol{p}_{k-1,k}^{\boldsymbol{c}_k}$ is the displacement vector between time epochs $(k-1)$ and k given up to a scale factor (Chu, 2012), expressed in the frame of the camera $\boldsymbol{c}_k$
- $\left(\boldsymbol{p}_{k-1,k}^{\boldsymbol{c}_k} \right) \times$ is the skew-symmetric matrix of $\boldsymbol{p}_{k-1,k}^{\boldsymbol{c}_k}$
- $\boldsymbol{C}_{k-1,k}$ is the rotation between time epochs $(k-1)$ and k.

The trajectory of the camera is recovered by concatenating the successive estimated translations and rotations using the principle described previously in Figure 3-2 and expressed mathematically by the following equation:

$$\begin{aligned} \boldsymbol{C}_{c_k 2v} &= \boldsymbol{C}_{c_{k-1} 2v} \ \boldsymbol{C}_{k-1,k}^T \\ \boldsymbol{p}_{c_k}^{\boldsymbol{v}} &= \boldsymbol{p}_{c_{k-1}}^{\boldsymbol{v}} + \boldsymbol{C}_{c_k 2v} \, \boldsymbol{p}_{k-1,k}^{\boldsymbol{c}_k} \end{aligned} \tag{3.10}$$

3.1.2.3.2 Visual Simultaneous Localization and Mapping

Contrary to VO which aims to estimate only the camera pose based on the displacement measured between two successive frames, VSLAM accounts for the correlations that exist between the camera pose and the 3D position of the observed features. In fact, VSLAM differs from VO in its aim to localize and build a map (determine the 3D feature positions) simultaneously, instead of only localizing. This improves dramatically the pose estimation accuracy since it takes into account the correlation between the observed features and the camera pose, but at the cost of an additional computational burden. The accuracy improvement is especially notable when a loop closure is detected. Indeed, by seeing again features that have been already seen, VSLAM is capable of integrating this constraint into the map, reducing consequently the pose drift, compared to VO. For these reasons, the choice between VSLAM and VO depends on the trade-off between accuracy and simplicity in implementation. But recently, with the computation processing advances, VSLAM becomes the preferred solution for vision-based navigation.

The SLAM map reconstruction is done through the triangulation process. This process, described in Figure 3-8, consists of estimating the 3D position of a feature, denoted $\widehat{\boldsymbol{p}}^{\boldsymbol{v}}$, from its 2D corresponding image points $\widetilde{\boldsymbol{p}}_{k-1}^{\mathrm{I}}$ and $\widetilde{\boldsymbol{p}}_k^{\boldsymbol{I}}$, detected in the image plane of the moving camera at two successive times $(k-1)$ and k. In perfect conditions, the reconstructed 3D

feature point is the intersection between the two rays going from the camera centers $\boldsymbol{c}_{k-1}$ and $\boldsymbol{c}_k$ to the detected features $\tilde{\boldsymbol{p}}^{\mathrm{I}}_{k-1}$ and $\tilde{\boldsymbol{p}}^{I}_{k}$. However, because of the image noise, the camera model, the calibration errors and the processing errors, these two rays almost never intersect. A solution is to choose the point at a minimal distance in the least squares sense from the intersecting rays as an estimate of the 3D feature position (Scaramuzza, 2011).

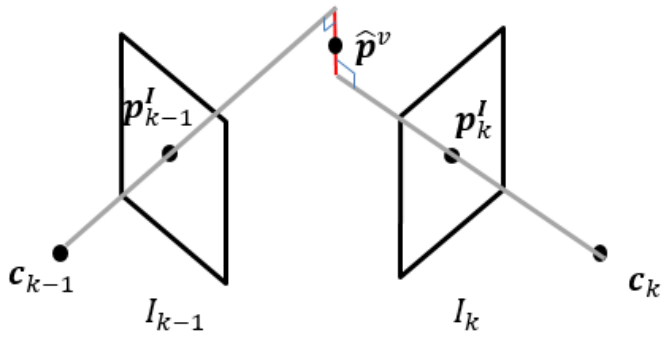


Figure 3-8 VSLAM triangulation process

- **VSLAM methodologies**

In literature, two different methodologies of VSLAM are generally implemented: the keyframe-based VSLAM (Mur-Artal, 2015) and the filtering-based SLAM (Strasdat, 2012). These two different implementations are presented below.

- **Filtering-based visual SLAM**

This approach is based on the construction of a probabilistic 3D feature map, describing the current pose estimation of the camera and the reconstructed 3D features that define the map, in the vision frame. In addition, this approach provides the estimate uncertainties. The map is initialized in the beginning of the process and is updated, as the camera moves, by Bayesian Filters such as the Kalman Filter or the Particle Filter. The probabilistic map is mathematically described through a state vector and a covariance matrix. The state vector comprises the concatenation of the estimations of the camera pose and the 3D features. The probability density function of the map parameters is usually approximated by a multiple variable Gaussian distribution. This technique is detailed in (Durrant-Whyte, 2006) and (Bailey, 2006).

- **Keyframe-based VSLAM**

The principle of Keyframe-based SLAM is to select some specific frames to reconstruct the 3D map of the environment based on the minimum number of correspondences that should exist between two frames. By doing this selection, the reconstruction process is done only when there is a significant change in the scene filmed by the camera. This selection has two major benefits: firstly, it allows the reduction of the VSLAM computational cost by only processing a small number of frames for the map reconstruction step and secondly it ensures a sufficient baseline between the two frames for the triangulation process. For example, if the camera moves straightforward, then there is no notable change in the scene seen by the camera and therefore,

the time between two keyframes is long. However, when the camera turns, the scene changes quickly, implying a rapid change of the keyframes in order to take into account the new features seen by the camera. Figure 3-9 explains the keyframe selection process. In this figure, Kf_n, is the n^{th} selected keyframe and I_j is the j^{th} captured image with $j \in \{k,\ k+1, \cdots\}$. For each newly captured I_j after the selection of Kf_n as a keyframe, the number of correspondences between Kf_n and I_j is computed. If this number is higher than a threshold, then I_j is not selected as a keyframe, otherwise, the test considers that the scene has sufficiently changed to select I_j as the $(n+1)^{th}$ keyframe. In this case, the selected $(n+1)^{th}$ keyframe is I_{k+4}.

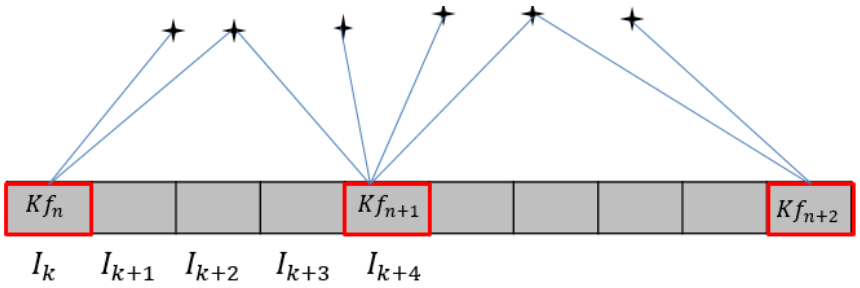


Figure 3-9 Keyframe selection

Although the map is reconstructed using only keyframes, the camera pose is computed for all the frames given the position of the already reconstructed 3D features. A refinement process, called *Bundle Adjustment* (BA) (Engles, 2006) and consisting of jointly optimizing the 3D reconstructed map and the camera pose, is generally performed. BA is a nonlinear least squares refinement that aims to minimize the reprojection error of a 3D feature point. This error is defined as the sum of squared residuals between the detected image features and the reprojection of their corresponding 3D features, using the projection model defined in Section 3.1.2.2, of their corresponding reconstructed 3D features. The mathematical formulation of BA is given is the following equation:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{i} \left\| (\tilde{\boldsymbol{p}}^I)^i - \left(\boldsymbol{p}^I_{\perp}(\boldsymbol{x})\right)^i \right\|_2^2 \tag{3.11}$$

where

- $\hat{\boldsymbol{x}}$ is the vector of the estimated camera pose and the reconstructed 3D feature positions
- $(\tilde{\boldsymbol{p}}^I)^i$ is the 2D vector of the measured image position of the i^{th} detected feature
- $\left(\boldsymbol{p}^I_{\perp}(\boldsymbol{x})\right)^i$ is the vector of the 2D positions obtained by projecting the reconstructed 3D i^{th} feature on the image plane

A review of BA algorithms is performed in (Triggs, 1999). As explained earlier, this optimization process is very important to improve the accuracy performance especially when a loop closure is detected. This means that when the scene includes again features that have been seen before, the drifts of the estimated map and pose due to the error accumulation can be removed using the past knowledge of feature positions.

In this case, instead of establishing matches between features detected in successive images, which is called (2D/2D correspondence), visual SLAM establishes matches between detected features in the current image and already reconstructed features. This process described in (Scaramuzza, 2011) is called '3D/2D correspondence'. The motion estimation is not deduced from the decomposition of the essential matrix such as in VO, but rather from solving the *Perspective-Three-Point Problem*. The definition and the solutions of this problem are detailed in (Moreno-Noguer, 2007).

- **Optimal approach for VSLAM**

It is difficult to compare the keyframe-based SLAM and the filtering-based SLAM and to conclude which approach is better. In fact, each approach has its advantages and drawbacks, depending on the intended application. Filtering-based visual SLAM has the advantage of handling easily measurement covariances, but is less accurate than keyframe SLAM in large-scale environments (Michot, 2010). An analysis of the performance of keyframe-based and filtering-based approaches is performed in (Strasdat, 2012). It revealed that the keyframe-based approach with BA is more efficient than the filtering-based approach in terms of accuracy and computational cost. Indeed, in filtering-based VSLAMs, the camera pose as well as all the map features already seen are included in the state vector and are updated at every processed frame. However, in keyframe-based VSLAMs, the camera pose is estimated using a subset of the entire map, without the need to update the map's features at every processed frame. This has the advantage of reducing significantly the processing time and hence making the real-time processing possible. In addition, the linearization process used in filtering-based VSLAMs is avoided in keyframe-based VSLAMs, which reduces the accumulation of errors. For these reasons, most new releases of VSLAM systems are keyframe-based. Therefore, the following study will only focus on keyframe-based VSLAMs.

- **Feature matching in Keyframe-based VSLAM**

Unlike for VO where the matching process is only applied to the 2D detected features (called 2D-2D correspondence), the matching process for the VSLAM includes the 3D reconstructed features. In the VSLAM-related literature, in addition to 2D-2D correspondence, 2 types of matching are generally applied: the 3D-2D and the 3D-3D correspondences.

The 3D-2D correspondence method consists of detecting features in the current image, associating them with the map already reconstructed based on the previous keyframes and estimating the camera pose $\boldsymbol{x}$ that minimizes the reprojection error between detected features in the current image and the reprojection of the corresponding previously reconstructed features on the current image plane. This process is formulated using equation (3.11), which is the same equation as for BA. The problem of solving this equation is known as *Perspective from n Points* (PnP) and many solutions are proposed in the literature (Moreno-Noguer, 2007).

The 3D-3D correspondence method consists of detecting features in the current image, reconstructing them in 3D, associating them with the map already reconstructed based on the previous keyframes and finally estimating the camera pose $\boldsymbol{x}$ that minimizes the distance between the 3D reconstructed features and the previously established map using:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{i} \left\| \left(\hat{\boldsymbol{p}}^{v}(\boldsymbol{x})\right)^{i} - \left(\boldsymbol{p}_{map}^{v}\right)^{i} \right\|_{2}^{2} \tag{3.12}$$

where

- $\left(\hat{\boldsymbol{p}}^{v}(\boldsymbol{x})\right)^{i}$ is the 3D position of the i^{th} feature reconstructed using the camera pose $\boldsymbol{x}$
- $\left(\boldsymbol{p}_{map}^{v}\right)^{i}$ is the 3D position of the i^{th} feature existing in the previously established map

The solutions that have been proposed to solve this problem are detailed in (Scaramuzza, 2011).

The 3D-2D correspondence method is the most popular since it has been shown in (Nistér, 2004) that it is more accurate than the 3D-3D correspondence one.

3.2 ORB SLAM

3.2.1 Selection of ORB-SLAM

A review of the state of the art of keyframe-based VSLAM during this thesis had led to identifying several solutions proposed in the literature. Recently, a survey on keyframe-based VSLAM has been proposed by (Younes, 2017), in which the authors identify 19 keyframe-based VSLAM proposed solutions. Among these solutions, only few Keyframe-based are available in open-access. The review of these solutions has led to identify 3 candidates: ORB-SLAM (Mur-Artal, 2015), Parallel Tracking and Mapping (PTAM) (Klein, 2007), and Large Scale Direct SLAM (LSD-SLAM) (Engel, 2014). In (Mur-Artal, 2015) , it has been shown that ORB-SLAM outperforms other types of VSLAM, in terms of accuracy and robustness. PTAM has been tested in the framework of this thesis. However, frequent losses of map track have occurred in outdoor environments, proving that this solution is more adapted to small scale operations. For all these reasons, ORB-SLAM is selected as the VSLAM algorithm that is used in this study.

3.2.2 ORB-SLAM characteristics

ORB-SLAM is based on 3 main tasks:

- tracking: localizing the camera every time a frame is captured and deciding where to insert a new keyframe for map building
- mapping: reconstructing the 3D map from keyframes and perform local BA to achieve an optimal reconstruction in the surroundings of the camera
- loop closing: when features already reconstructed are seen again, a loop is detected. The drift accumulated during the loop can therefore be removed.

This algorithm is based on ORB features (Rublee, 2011) which are invariant to viewpoint and illumination changes and extremely fast to compute and match, unlike the popular SURF or SIFT that require much more time to be extracted. ORB SLAM discretizes and stores the feature

descriptors into bags of words, known as visual vocabulary (Mur-Artal, 2014) which is used to speed up image and feature matching.

Pose estimation in ORB SLAM is established through a constant velocity motion model to predict the camera pose and the position of the features already reconstructed. This model assumes a smooth camera motion. If the motion model is violated (abrupt motions), ORB SLAM detects this problem through the number of matches found and uses a wider search of features around their position in the last frame. If tracking is lost, ORB SLAM uses a Global relocalization to handle this problem. The pose is estimated using the PnP algorithm that determines the camera pose from the 3D-2D correspondence explained in Section 3.1.2.3.2. The prediction of camera and feature location helps feature matching process to search in a small region instead of across the entire image. This reduces the computational expense of the algorithm. In addition, it serves as a starting point for the minimizing procedure of the reprojection error, which provides a better estimate of the camera pose.

To adapt ORB SLAM to large environments, a local map is defined by all the features present in the set of keyframes that share features with the current frame, as well as all neighbors of this set of keyframes. This neighborhood between keyframes is determined using the covisiblity graph. This graph is defined as a weighted graph having nodes and edges. Each node is a keyframe and an edge between two keyframes exists if they share observations of the same features (at least 15) (Mur-Artal, 2015). The weight of the edge is the number of common map features between two keyframes. Among the previously defined features, only some selected features are kept in the local map. This selection is performed such that the matching process is most likely to be successful. The remaining features are then matched in the current frame before a final camera pose refinement step takes place.

ORB SLAM's local mapping process is performed with the following steps: keyframe insertion, map point creation and culling, local BA and local keyframe culling. In the keyframe insertion process, the covisibility graph is updated with the new node (keyframe) and its corresponding edges, as well as the bag of words representing the new keyframe. The map point creation module is based on the triangulation of ORB features appearing in at least two nodes of the covisibility graph, as explained in Section 3.1.2.3.2. Before triangulation, the feature association and the outlier rejection are based on the Epipolar geometry described in Section 3.1.2.3.1. Triangulated features are tested for positive depth, reprojection error, and scale consistency to accommodate them into the map. To be retained in the map, map points should fulfill two conditions to ensure good map quality (very few outliers). These conditions consist of finding the point in more than 25% of the frames in which it is predicted to be visible, and seeing the point from at least 3 keyframes. Otherwise the map point is removed. After creating the map points and selecting the best ones, a local BA process is applied to the current keyframe, to all the keyframes connected to it in the covisibility graph, to all the map features seen by these keyframes and to all keyframes that see those map points but are not connected to the current keyframe. Finally, a keyframe culling process is applied. In this step, the keyframes that have 90% of their matched features observed in 3 other keyframes are deemed redundant and are removed. This allows reducing the computational complexity resulting from having a high number of keyframes in the map.

The third task performed in ORB-SLAM is loop closing. First, the loop is detected using a similarity measure between the current keyframe bag of word and the covisibility graph information. Then, a similarity transformation is computed from the current keyframe to the loop keyframe to provide information about the drift accumulated in the loop. Once this similarity validated with enough inliers, an optimization is performed and the loop closure is accepted. The loop fusion is then performed. It consists of merging duplicate map points in both keyframes and inserting a new edge in the covisibility graph that closes the loop by correcting the pose of the current keyframe and all its neighbors using the similarity transform. Then, all map points involved in the loop closure process are fused and all keyframes involved in this process update their edges in the covisibility graph to take into account this loop closure. Finally, the Essential graph, that consists of a covisibilty graph whose edges are reduced, is optimized to correct the drift. After optimization, the map points corresponding to the optimized keyframes are transformed to fit with the new keyframe poses.

Figure 3-10 and Figure 3-11 illustrate a comparison between the trajectories of a car on which we have mounted a camera and run ORB-SLAM (data collection is explained in Chapter 6). The trajectory is illustrated with the blue rectangles and the map with the red (active) and black (non-active) points. The car returns to its initial point after having traveled around a building. We can see in Figure 3-10 that the trajectory drifts before detecting the loop closure. However, once the loop closure is detected, the trajectory and map points are jointly adjusted to remove the drift as illustrated in Figure 3-11.

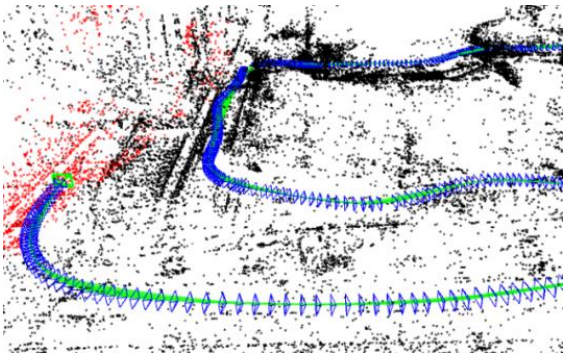


Figure 3-10 ORB-SLAM trajectory before loop closure

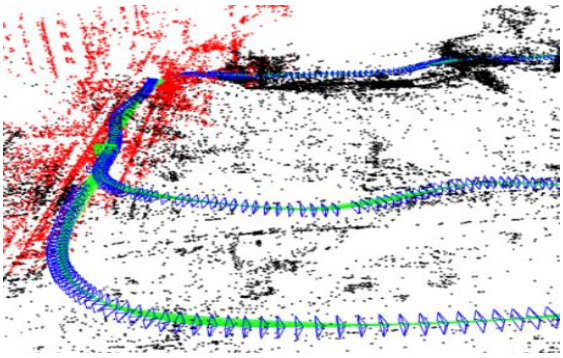


Figure 3-11 ORB-SLAM trajectory after loop closure

3.3 Challenges of vision-based navigation in outdoor environments

Although vision has shown to be promising in navigation, several challenges arise when using visual information provided by one of the techniques described previously. These challenges are summarized in the following points:

- Vision-based navigation techniques suffer from the accumulated error resulting from the use of the DR navigation principle, if they are used in unknown environments. Compared to VSLAM, VO suffers from a higher drift rate because VO techniques are based only on the DR principle, whereas VSLAM techniques combine DR and BA which improves the accuracy of visual solution, but at the cost of an increased computational burden for the optimization process. Furthermore, the possibility of loop

closure with VSLAM is very promising because it allows removing the drift that accumulates during navigation.

- Vision-based techniques have the drawback of providing their navigation outputs in an unknown frame constructed at the initialization step (vision frame). This issue makes the use of this information challenging, unless a previously established geo-referenced map is used to perform localization. This case of study will not be addressed in the scope of this thesis, and the problem of the unknown vision frame must be handled, to be able to use visual information in navigation.
- The choice of monocular vision introduces a scale factor to the position information. This scale should be determined to make the visual information useful.

For these reasons, the integration of vision-based information with other sensor measurements is necessary to improve the localization performances. Before tackling fusion, it is important first to address the VSLAM measurement model, which is a necessary step in multi-sensor fusion. This measurement model must take into account the lack of knowledge of the vision frame and of the scale factor affecting the VSLAM position.

3.4 VSLAM measurement model

As explained in the previous section, VSLAM provides the camera 3D position in undefined scale, the camera attitude and the reconstructed map with respect to its own arbitrary frame (v). To be able to fuse VSLAM with other sensors, its outputs must be expressed in the same frame as the other sensors. Considering the local frame (l) as the frame in which all sensors outputs are expressed, the relationship between the camera position estimated by the SLAM in (v) and the corresponding position in (l) is the following:

$$\tilde{\boldsymbol{p}}_c^v = k_v \boldsymbol{C}_{l2v} \boldsymbol{p}_c^l + \boldsymbol{p}_{lv}^v \tag{3.13}$$

where

- $\tilde{\boldsymbol{p}}_c^v$ is the camera 3D position in (v) estimated by the VSLAM
- k_v is the scale factor that affects the VSLAM 3D position measurement
- $\boldsymbol{C}_{l2v}$ is the rotation matrix from (l) to (v)
- $\boldsymbol{p}_c^l$ is the camera 3D position in (l)
- $\boldsymbol{p}_{lv}^v$ is the translational offset between (l) and (v)

As for the camera attitude estimated by the visual SLAM, the relationship with the camera attitude in (l) is given in terms of quaternions[7] by:

$$\tilde{\boldsymbol{q}}_{v2c} = \boldsymbol{q}_{l2c} \otimes \boldsymbol{q}_{v2l} \tag{3.14}$$

where

- $\otimes$ is the quaternion multiplication operator explained in Appendix B

[7] see Appendix B for quaternion definiition

- $\widetilde{\boldsymbol{q}}_{v2c}$ is the quaternion defining the attitude of the camera with respect to the (v) frame. This quaternion is the output of visual SLAM
- $\boldsymbol{q}_{v2l}$ is the quaternion defining the rotation from (v) to (l). Its associated rotation matrix is $\boldsymbol{C}_{v2l} = \boldsymbol{C}_{l2v}^{T}$. The relationship between a rotation matrix and its corresponding quaternion is also detailed in Appendix B
- $\boldsymbol{q}_{l2c}$ is the quaternion defining the misalignment between the camera and (l).

The main issue of this model is that the (v) frame is unknown because it is constructed during the SLAM initialization process and depends on the features detected by the SLAM during initialization. Therefore, the transformation between (v) and (l) which involves $k_v, \boldsymbol{C}_{l2v}$ (or equivalently $\boldsymbol{q}_{l2v}$) and $\boldsymbol{p}_{lv}^{v}$ is unknown and should be estimated to be able to exploit the SLAM outputs. The estimation of this transformation will be described later in this section, for SLAM performance evaluation purpose, and will be performed in Chapter 4 differently to be able to use the SLAM in a multi-sensor framework.

The evaluation of SLAM performance can be done by evaluating the reconstructed map quality. However, accurate ground truth maps are difficult to obtain (Sturm, 2012). Therefore, SLAM performance is evaluated in this study by computing the outputted SLAM poses and comparing them to the ground truth. To compute this error, the pose should be expressed in the SE(3) Euclidean group (Blanco, 2010). This means that the SLAM and ground truth poses can be expressed, at time k, with an invertible (4×4) matrix as follows:

$$\begin{aligned} \tilde{\boldsymbol{S}}_k^{pose} &= \begin{bmatrix} \widetilde{\boldsymbol{C}}_{v2c} & \widetilde{\boldsymbol{p}}_c^{v} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}_k \\ \boldsymbol{G}_k^{pose} &= \begin{bmatrix} \boldsymbol{C}_{l2c} & \boldsymbol{p}_c^{l} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}_k \end{aligned} \tag{3.15}$$

where

- $\mathrm{SE}(3) = \left\{ \boldsymbol{A} | \boldsymbol{A} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{p} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix};\ \boldsymbol{C} \text{ is a } (3\times 3) \text{ rotation matrix}, \boldsymbol{p} \text{ is a } (3\times 1) \text{ vector} \right\}$
- $\tilde{\boldsymbol{S}}_k^{pose}$ and $\boldsymbol{G}_k^{pose}$ are respectively the SLAM and the ground truth poses.
- $\widetilde{\boldsymbol{C}}_{v2c}$ and $\boldsymbol{C}_{l2c}$ are the (3×3) rotation matrices associated respectively to the quaternions $\widetilde{\boldsymbol{q}}_{v2c}$ and $\boldsymbol{q}_{l2c}$

A possible way to evaluate the SLAM performance is to apply the approach proposed in (Sturm, 2012). This approach consists of calculating two types of errors at each time epoch k:

- the Relative Pose Error (RPE) which measures the local accuracy of the trajectory over a fixed time interval Δ. It characterizes the drift of the trajectory when no loop closure is detected. It is given by:

$$\boldsymbol{E}_k^{RPE} = \left(\tilde{\boldsymbol{S}}_k^{pose^{-1}} \tilde{\boldsymbol{S}}_{k+\Delta}^{pose} \right)^{-1} \left(\boldsymbol{G}_k^{pose^{-1}} \boldsymbol{G}_{k+\Delta}^{pose} \right) \tag{3.16}$$

- the Absolute Trajectory Error (ATE) which measures the global consistency of the trajectory by comparing the absolute distances between the estimated and the ground truth trajectories:

$$\boldsymbol{E}_k^{ATE} = \boldsymbol{G}_k^{pose^{-1}} \; \boldsymbol{T}_k \; \tilde{\boldsymbol{S}}_k^{pose} \tag{3.17}$$

where $\boldsymbol{T}_k$ is the transformation in $SE(3)$ that maps the SLAM pose $\tilde{\boldsymbol{S}}_k^{pose}$ onto the corresponding ground truth pose $\boldsymbol{G}_k^{pose}$. This transformation is necessary because the SLAM pose is expressed in (v) while the ground truth pose is expressed in (l). Therefore, based on equation (3.13), the expression of $\boldsymbol{T}_k$ is given by:

$$\boldsymbol{T}_k = \frac{1}{k_v}\begin{bmatrix}\boldsymbol{C}_{l2v} & \boldsymbol{p}_{lv}^{v} \\ \boldsymbol{0}_{1x3} & 1\end{bmatrix}_k^{-1} \tag{3.18}$$

This transformation is computed based on Horn method whose steps are detailed in (Horn, 1987).

The SLAM performance analysis as well as the $\boldsymbol{T}_k$ computation are done in Chapter 6 where the data collections are described, and the results are presented.

3.5 Conclusion

In this chapter, we have focused on the study of vision-based navigation to identify the techniques and the type of measurement that will be used in the multi-sensor fusion architecture, and to understand the characteristics and challenges of this type of information.

First, a review of the state of the art of standalone vision-based navigation techniques had led to establishing a classification of these techniques depending on the environment in which the camera moves. This classification has led to the identification of two vision-based techniques: VO and VSLAM. These techniques have been addressed in detail and compared. VSLAM has been identified as the best technique capable of providing navigation information with the best accuracy performance. Indeed, unlike VO based on simple DR to provide camera motion, SLAM relies on several optimization processes improving jointly the accuracy of the navigation solution and of the map. For this reason, this information has been identified as the visual information to be used in the multi-sensor architecture. The review of open-source SLAMs and the comparison between these proposed solutions has led to the identification of ORB-SLAM as the solution that will be used to assess the performance of the hybridization architecture that will be defined in Chapter 4. This choice is mainly based on the possibility for ORB-SLAM to cope with outdoor large-scale environment, unlike most of the other SLAM solutions designed for indoor navigation.

Then, the challenges of these choices are discussed. In particular, issues related to the unknown transformation between the vision frame and the local frame are highlighted. This transformation includes an unknown scale factor resulting from monocular vision, an unknown rotation and an offset between the vision frame and the local frame. This transformation has been used to model the SLAM outputted pose. This modeling step is necessary for studying

SLAM error characteristics and to link the SLAM output to the other sensor outputs to design the multi-sensor fusion architecture.

Chapter 4

Proposed Hybridization Architecture for the Improvement of Ground Vehicle Navigation in Urban Environments

In Chapter 2 and Chapter 3, the advantages and limits of the identified navigation sensors in terms of accuracy and robustness are presented especially in constrained environments. The current chapter deals with the description of the proposed hybridization solution integrating the identified sensor measurements. The first part of the chapter is an overall description of the proposed solution architecture defining the IMU as the reference sensor and the other measurements provided by GPS, SLAM, odometry and ground vehicle motion constraints as aiding information to correct the IMU, in an error-state EKF closed-loop architecture. The second part is a presentation of the theoretical model of the KF in general and of the EKF in particular. The third, fourth and fifth parts detail respectively the Kalman error state vector used in our model, its transition model and the observation model relating the aiding measurements to the state vector. Finally, a special attention is given to the observability analysis of the defined state vector. The detailed filter equations and calculations are provided in Appendix B and the observability analysis proofs are given in Appendix C.

Contents

4.1 System architecture 54
4.2 Extended Kalman Filter 57
4.3 State vector 59
4.4 State transition model 62
4.4.1 Continuous-time model 62
4.4.2 Discrete-time model 65
4.5 Observation model 65
4.5.1 GNSS observation model 65
4.5.2 SLAM observation model 67
4.5.3 WSS and NHC observation model 68
4.5.4 ZARU observation model 69
4.6 Observability analysis 69
4.6.1 Observability analysis using Lie derivatives 70
4.6.2 Nonlinear system formulation 71
4.6.3 Measurement formulation 72
4.6.4 Observability analysis 73
4.7 Conclusion 78

4.1 System architecture

The proposed architecture consists of a single frequency GNSS receiver, a low-cost IMU, a vision module processing the images of a single camera using the VSLAM technique, odometry module providing the vehicle velocity and constraints describing the motion of a ground vehicle.

Figure 4-1 depicts the overall architecture of the proposed hybridization. The IMU is selected as the reference sensor since it is the only sensor providing, continuously and autonomously, a complete navigation solution (vehicle position, velocity and attitude) at the highest rate ($100\,Hz$). The IMU provides the accelerometer and gyroscope measurements. These measurements are processed using the INS mechanization described in Section 2.2.3 to provide the navigation solution in the local frame (l). All the filter computations are done in this frame. In particular, the position is expressed in meter unit in this frame instead of being expressed in radians in the geodetic frame (latitude, longitude, height). This allows avoiding numeric instability inside the filter resulting from the very small values of the latitude and longitude errors, as highlighted in (Angrisano, 2010) and (Shin, 2001).

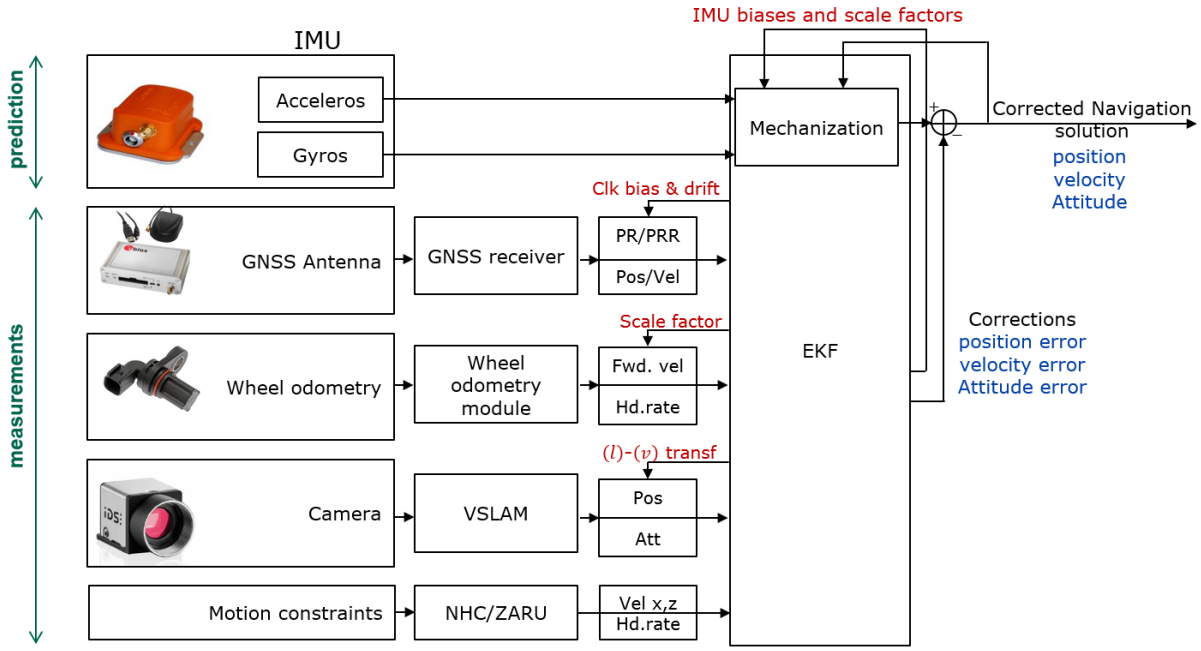


Figure 4-1 Multi-sensor fusion architecture

The INS mechanization errors are corrected by the measurements of the other sensors. An error-state EKF is used in order to estimate the corrections that should be applied to the inertial navigation solution. The choice of an error-state EKF is justified by the analysis done in Section 2.4.1.

The estimated IMU measurement errors (biases and scale factors) are fed back to the mechanization since we have to deal with a low-cost IMU. Otherwise the mechanization can experience unbounded error growth, and the assumption of small errors used in the linearization process of the filter can be violated (Shin, 2001). Indeed, in the feedforward configuration, the reference sensor which is the IMU in our case is unaware of the existence of the other sensors and the correction is only applied to the mechanization outputs. This makes the INS quantities free to drift with unbounded errors, and the assumption of small errors used in the linearization process of the filter can be violated in the case of a low-cost IMU. However, with the feedback configuration, the IMU errors as well as the linearization points are fed back to reduce the drift of the INS and the assumption of small errors is respected.

The aiding sensors provide either raw measurements or measurements processed by a navigation module. For GNSS, the raw measurements are used in difficult environments such as urban canyons to avoid the problem of GNSS navigation solution unavailability in case of satellite masking, and to be able to discard wrong measurements in case of significant degradation. The raw GNSS are code pseudorange $\widetilde{\boldsymbol{\rho}}_G$ and Doppler measurements used to compute the pseudorange rate $\widetilde{\dot{\boldsymbol{\rho}}}_G$ using equation (2.2). Two important decisions have been taken regarding GNSS.

First, if the GNSS raw measurements are used, then the code measurements are preferred to the carrier phase measurements, although carrier phase measurements can provide better accuracy. In fact, in urban environments, multipath makes difficult the estimation of the carrier phase

measurement ambiguities. In addition, in this environment, frequent cycle slips and even a complete loss of lock may occur. This results in a low availability of GNSS in this environment (Carcanague, 2011). Unlike carrier phase measurements, the Doppler measurements are provided by the GNSS receiver continuously (Bahrami, 2010) and are not prone to cycle slips. For this reason, Doppler measurements will be used in this study. In favorable environments where a sufficient number of satellites are in view, the processed GNSS measurements, namely the position and the velocity, are directly used.

Then, the single frequency GNSS technique is used. In fact, despite its capacity to remove the ionospheric delay, the GNSS dual frequency technique combining the GPS L1 C/A signal ($f_{L1} = 1575.42$ MHz) and the civilian GPS L2C signal ($f_{L2C} = 1227.60$ MHz) is not used in this work for two main reasons. The first reason is the cost of the single frequency receiver that is lower than the dual frequency receiver. The second reason is that, in 2013, the year of the beginning of this Ph.D. thesis, the number of GPS satellites emitting L2C signals was limited to 11 satellites (Martin, 2013), therefore requiring the civilian dual frequency receivers to also process codeless tracking of GPS L2P(Y), rendering the tracking very noisy and not robust.

As for wheel odometry, the vehicle velocity and angular rate are estimated using the WSS velocity of each wheel and the steering wheel angle measurement, as described in Section 2.3. Then, the vehicle velocity and angular rate are input to the EKF. Based on Section 2.3, the velocity measurement is affected by an unknown scale factor due to the tire radius change. This scale factor should be estimated in the Kalman filter and the WSS velocity should be corrected in order to obtain a good velocity measurement used to update the Kalman filter. Since the WSS provides only the forward velocity, the Non-Holonomic-Constraints (NHC) (Sukkarieh, 2000) complete the three-dimensional velocity by assuming that a ground vehicle moves only straightforward, and its lateral and vertical velocities should be equal to zero if there is no slip or jump. This assumption of no side slip of the NHC could be violated in practical situations. For this reason, NHC is modelled as:

$$\begin{aligned} v_{NHC}^{bx} &= \varepsilon_{NHC}^{bx} \\ v_{NHC}^{bz} &= \varepsilon_{NHC}^{bz} \end{aligned} \tag{4.1}$$

where ε_{NHC}^{bx} and ε_{NHC}^{bz} are Gaussian white noise sources with zero mean and a relatively high standard deviation equal to 0.5 m/s (Angrisano, 2010).

When a stationarity is detected, the trust that we have in a zero velocity, on the lateral and vertical axes, increases and the standard deviation associated to the NHC can be decreased improving the solution accuracy. We set this standard deviation to 0.01 m/s. This detection is based on the comparison of the WSS velocity mean with a threshold over a suitable time window. The stationarity of the vehicle can be advantageous also in calibrating the IMU gyroscopes, by estimating their measurement biases. This can be done using the *Zero Angular Rate Update* (ZARU), which states that at stationarities, the angular rate must be zero. This method can also be applied in linear motion. The linear motion is detected using the heading angular rate provided by the steering wheel angle and computed over a suitable time window. In addition to improving the accuracy, these methods used in stationarity allow reducing the

uncertainty associated to the filter solution. This uncertainty reduction is a key step towards defining an integrity monitoring algorithm and evaluating its performance, as it will be detailed in Chapter 5.

The visual information is provided by VSLAM that processes the images captured by a camera rigidly mounted on the vehicle and outputs the scaled camera pose. As highlighted in Chapter 3, this pose is given in the unknown vision frame (v) making this information unusable if the transformation between (v) and (l) is not determined. For this reason, this transformation is estimated in the Kalman filter and is fed back to transform the pose in the local frame as illustrated in Figure 4-1. To avoid errors due to misalignment between the camera and the IMU, this misalignment is also estimated by the Kalman filter. We remind that this configuration is chosen in order to have a modular system that does not rely mandatorily on the vision module and gives the possibility to remove the vision-based module in bad visibility conditions or loss of the map track in case of abrupt vehicle motion. Furthermore, this approach allows us to rely on any vision-based module found in the literature if advances are done in this field. For these reasons, the option of considering the visual information removable is preferred in this work.

The aiding measurements provided by visual, odometric, NHC, and ZARU in order to correct the inertial sensor form a DR system running without interruption. Each sensor provides its measurement at its own rate and the update of inertial measurements is performed each time a measurement arrives. This system is capable of remarkably reducing the drift compared to a stand-alone running inertial system. If available, GNSS measurements are used to calibrate the DR system. Otherwise, the DR system keeps on running to ensure the navigation continuity and to reduce the INS drift.

4.2 Extended Kalman Filter

The EKF is chosen in this work for the reasons given in Section 2.4.1. Before tackling the EKF, an overview of the standard KF is given. The KF is a recursive algorithm that estimates the states of a dynamic system based on noisy observations. It is a Bayesian estimation technique that is based on the a priori knowledge of the system properties and that aims to minimize the Mean Square Error. The KF may operate continuously in time, however the discrete KF is considered in this work because sensor fusion requires digital computation.

The discrete KF is based on the following core elements:

- The state vector $\boldsymbol{X}_k$ which is the set of system parameters that the KF estimates at the time epoch k. To this vector is associated an error covariance matrix $\boldsymbol{P}_k$ describing the uncertainties in the KF's estimated parameters and the degree of correlation between the errors of these parameters.
- The system process model which describes the state vector evolution with time:

$$\boldsymbol{X}_{k+1} = \boldsymbol{F}_k \boldsymbol{X}_k + \boldsymbol{\eta}_k \tag{4.2}$$

where $\boldsymbol{F}_k$ is the system transition matrix at time epoch k and $\boldsymbol{\eta}_k$ is the process driving noise whose covariance matrix is denoted $\boldsymbol{Q}_k$

- The measurement vector $\widetilde{\boldsymbol{Z}}_k$ which consists of a set of simultaneous noisy measurements that are expressed as a function of the state vector components. To this vector is associated a measurement noise covariance matrix $\boldsymbol{R}_k$ describing the noise statistics of the measurements. The relationship between the measurement vector and the state vector components is given by the measurement model:

$$\widetilde{\boldsymbol{Z}}_k = \boldsymbol{H}_k \boldsymbol{X}_k + \boldsymbol{\varepsilon}_k \tag{4.3}$$

where $\boldsymbol{H}_k$ is the observation matrix and $\boldsymbol{\varepsilon}_k$ is the measurement noise whose covariance matrix is $\boldsymbol{R}_k$.

The KF algorithm consists of two phases, system propagation and measurement update. The system propagation consists of predicting the state vector estimate and the associated error covariance matrix using the system process model. In the measurement update phase, the state vector estimate and its error covariance are updated using the new measurements. The first step of this phase consists of computing the Kalman gain matrix. This quantity is used for weighting the corrections applied to the state vector according to the uncertainty of the current state estimates and how noisy the measurements are. Then, the state vector is updated based on the measurement data weighted with the Kalman gain.

In the standard KF, both the system and the measurement models are assumed to be linear, which is not the case in the considered application. A nonlinear version of the KF is the EKF, considered in this study. In an EKF, the nonlinear system is approximated by a linear system. The following system process and measurement models are considered in the EKF:

$$\begin{aligned} \boldsymbol{X}_{k+1} &= f(\boldsymbol{X}_k) + \boldsymbol{\eta}_k \\ \widetilde{\boldsymbol{Z}}_k &= h(\boldsymbol{X}_k) + \boldsymbol{\varepsilon}_k \end{aligned} \tag{4.4}$$

where f and h are nonlinear functions. Therefore, the prediction phase of the EKF is done as follows:

State prediction

$$\widehat{\boldsymbol{X}}_{k+1|k} = f\left(\widehat{\boldsymbol{X}}_{k|k}\right) \tag{4.5}$$

State error covariance prediction

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k \boldsymbol{P}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q}_k \tag{4.6}$$

where $\boldsymbol{F}_k$ is the Jacobian matrix of f computed in $\widehat{\boldsymbol{X}}_{k|k}$

The EKF update phase is based on the following steps:

Kalman gain computation

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T \left(\boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^T + \boldsymbol{R}_{k+1}\right)^{-1} \tag{4.7}$$

where $\boldsymbol{H}_{k+1}$ is the Jacobian matrix of h computed in $\widehat{\boldsymbol{X}}_{k+1|k}$

State update

$$\widehat{\boldsymbol{X}}_{k+1|k+1} = \widehat{\boldsymbol{X}}_{k+1|k} + \boldsymbol{K}_{k+1}\left(\widetilde{\boldsymbol{Z}}_{k+1} - h\left(\widehat{\boldsymbol{X}}_{k+1|k}\right)\right) \tag{4.8}$$

State error covariance update

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1} \tag{4.9}$$

Although the EKF handles the linearity problem, it has the disadvantage of being based on a first order Taylor series linearization. This can lead to important problems in the stability of the filter as highlighted in (Spangenberg, 2007). To reduce the effect of this limitation, the error state form is considered because the dynamics of the system error are low frequency and could be represented as linear. Furthermore, the linearization point must be very close the true point, to avoid the divergence of the EKF. This is the reason why the closed loop is considered in this work.

4.3 State vector

This work is based on the error state formulation of an EKF where the state prediction is driven by the inertial system. The IMU is defined as the reference sensor in the architecture described above. Therefore, the designed filter aims to estimate the position, velocity and attitude errors of the IMU. However, due to errors in the inertial measurements as modeled in equation (2.27), the state vector is augmented by the inertial errors which are the biases and the scale factors of the accelerometers and gyroscopes measurements. Therefore, the core state vector is given by:

$$\delta\boldsymbol{X}_{IMU} = \left[\delta\boldsymbol{p}_b^{l\,T} \quad \delta\boldsymbol{v}_b^{l\,T} \quad \boldsymbol{\varepsilon}_{b2l}{}^T \quad \delta\boldsymbol{b}_f{}^T \quad \delta\boldsymbol{b}_\omega{}^T \quad \delta\boldsymbol{k}_f{}^T \quad \delta\boldsymbol{k}_\omega{}^T\right]^T \tag{4.10}$$

where

- $\delta\boldsymbol{p}_b^l = [\delta p_b^E \quad \delta p_b^N \quad \delta p_b^U]^T$ is the position error vector in (l)
- $\delta\boldsymbol{v}_b^l = [\delta v_b^E \quad \delta v_b^N \quad \delta v_b^U]^T$ is the velocity error vector in (l)
- $\boldsymbol{\varepsilon}_{b2l} = [\varepsilon^E \quad \varepsilon^N \quad \varepsilon^U]^T$ is the vehicle attitude error vector in (l)
- $\delta\boldsymbol{b}_f = [\delta b_{fx} \quad \delta b_{fy} \quad \delta b_{fz}]^T$ is the accelerometer biases error in (b)
- $\delta\boldsymbol{b}_\omega = [\delta b_{\omega x} \quad \delta b_{\omega y} \quad \delta b_{\omega z}]^T$ is the gyroscope biases error in (b)
- $\delta\boldsymbol{k}_f = [\delta k_{fx} \quad \delta k_{fy} \quad \delta k_{fz}]^T$ is the accelerometer scale factors error in (b)
- $\delta\boldsymbol{k}_\omega = [\delta k_{\omega x} \quad \delta k_{\omega y} \quad \delta k_{\omega z}]^T$ is the gyroscope scale factors error in (b)

The aiding sensors can then be added modularly to this core system. Table 4.1 summarizes each sensor outputs and the frame in which the outputs are expressed. The INS provides the vehicle position, velocity and attitude computed in the IMU location point. GNSS processed measurements are calculated in (l). For GNSS raw data, no frame is associated to these measurements because pseudoranges are distance scalar between the receiver and each satellite in view and can be expressed in any orthonormal frame. For Doppler measurements, they are

equivalent to a frequency shift scalar and are also frame-independent. For the wheel odometry, the velocity is computed either in the vehicle rear axle center point or in its front axle center point, depending on the wheels we use, and is then expressed in (b). Note that the attitude is given in terms of a rotation matrix $\boldsymbol{C}$ or equivalently a quaternion $\boldsymbol{q}$. More details about these two mathematical rotation representations are given in Appendix B.

Nav. module	Output		Frame
INS	Vehicle position	$\boldsymbol{p}_b^l$	(l)
	Vehicle velocity	$\boldsymbol{v}_b^l$	
	Vehicle attitude	$\boldsymbol{C}_{b2l}/\boldsymbol{q}_{b2l}$	
GNSS processed data	Antenna position	$\widetilde{\boldsymbol{p}}_G^l$	
	Antenna velocity	$\widetilde{\boldsymbol{v}}_G^l$	
GNSS raw data	Antenna Pseudoranges	$\widetilde{\boldsymbol{\rho}}_G$	No frame
	Antenna Doppler meas.	$\Delta\widetilde{\boldsymbol{f}}_G$	
VSLAM	Camera position	$\widetilde{\boldsymbol{p}}_c^v$	(v)
	Camera attitude	$\widetilde{\boldsymbol{C}}_{v2c}/\widetilde{\boldsymbol{q}}_{v2c}$	
Wheel odometry	Rear or front velocity	$\tilde{v}_w^{by}$	(b)
	Angular rate	$\widetilde{\boldsymbol{\omega}}_s$	

Table 4.1 Sensor outputs and associated frames

Based on the measurement models given by equations (2.17) and (2.21) for GNSS, equation (2.41) for the WSS, and equations (3.13) and (3.14) for SLAM, we notice that every sensor adds a number of unknown parameters which relate the provided measurements to the core state. To reduce the number of unknowns, it is assumed that the lever arms between the aiding sensors and the IMU are measured before the navigation starts and are perfectly known. More details about the configuration of sensors during the data collection are given in Chapter 6. The full system state is assembled from the core state $\delta\boldsymbol{X}_{IMU}$ and a series of additional states. These additional states are detailed in the following paragraphs.

The WSS measurement model given in equation (2.41) is affected by an unknown scale factor k_w and this scale factor error forms the sate vector associated to the WSS:

$$\delta\boldsymbol{X}_w = [\delta k_w] \tag{4.11}$$

For GNSS, if the processed position and velocity are input to the filter, then no additional states are introduced. However, if the raw GNSS pseudorange and Doppler measurements are used, then two additional states should be accounted for, which are the receiver clock bias b_G^{clk} and clock drift $\dot{b}_G^{clk}$ errors:

$$\delta \boldsymbol{X}_{GNSS} = [\delta b_G^{clk} \quad \delta \dot{b}_G^{clk}]^T \tag{4.12}$$

As for the SLAM outputs, the measurement models given in equation (3.13) and equation (3.14) relate the camera outputs expressed in (v) to their corresponding measurements expressed in (l). Since the designed filter takes the IMU as the reference sensor, we should express first the SLAM outputted pose as a function of the IMU pose by taking into account the lever arm $\Delta \boldsymbol{p}_c^b$ between the camera and the IMU as well as the alignment of the camera with respect to the IMU $\boldsymbol{C}_{b2c}$ (or equivalently $\boldsymbol{q}_{b2c}$). Based on equation (3.13) and equation (3.14), we have (Weiss, 2012):

$$\widetilde{\boldsymbol{p}}_c^v = k_v \, \boldsymbol{C}_{l2v} \left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l} \, \Delta \boldsymbol{p}_c^b \right) + \boldsymbol{p}_{lv}^v + \boldsymbol{\varepsilon}_{v,p} \tag{4.13}$$

$$\widetilde{\boldsymbol{q}}_{v2c} = (\boldsymbol{q}_{b2c} \otimes \boldsymbol{q}_{l2b} \otimes \boldsymbol{q}_{v2l}) + \boldsymbol{\varepsilon}_{v,q} \tag{4.14}$$

where

- $\widetilde{\boldsymbol{p}}_c^v$ and $\widetilde{\boldsymbol{q}}_{c2v}$ are the SLAM pose measurements
- $\boldsymbol{p}_b^l$ and $\boldsymbol{C}_{b2l}$ are the IMU pose
- k_v , $\boldsymbol{C}_{l2v}$ and $\boldsymbol{p}_{lv}^v$ are the scale factor, rotation and translational offset defining the transformation between (v) and (l)
- $\boldsymbol{\varepsilon}_{v,p}$ is an additive noise representing the SLAM position measurement error
- $\boldsymbol{\varepsilon}_{v,q}$ is an additive noise representing the SLAM attitude measurement error.

Therefore, 4 parameters are unknown: the 3 parameters describing the transformation between (v) and (l) as well as the alignment of the camera with respect to the IMU. In terms of error states, we can write:

$$\delta \boldsymbol{X}_{SLAM} = \left[\delta k_v \quad {\boldsymbol{\varepsilon}_{l2v}}^T \quad \delta {\boldsymbol{p}_{lv}^v}^T \quad {\boldsymbol{\varepsilon}_{b2c}}^T\right]^T \tag{4.15}$$

where

- $\boldsymbol{\varepsilon}_{l2v} = [\varepsilon_{l2v}^1 \quad \varepsilon_{l2v}^2 \quad \varepsilon_{l2v}^3]^T$ is the (v)-(l) frame orientation error
- $\delta \boldsymbol{p}_{lv}^v = \left[\delta p_{lv}^{vx} \quad \delta p_{lv}^{vy} \quad \delta p_{lv}^{vz}\right]^T$ is the offset error between (v) and (l)
- $\boldsymbol{\varepsilon}_{b2c} = [\varepsilon_{c2b}^1 \quad \varepsilon_{c2b}^2 \quad \varepsilon_{c2b}^3]^T$ is the orientation error of the camera frame (c) with respect to (b)

Finally, the full error state vector $\delta \boldsymbol{X}$ is assembled from the core state corresponding to the INS mechanization and the additional states corresponding to the GNSS receiver, the SLAM and the WSS algorithms:

$$\delta \boldsymbol{X} = \begin{bmatrix} \delta \boldsymbol{X}_{IMU} \\ \delta \boldsymbol{X}_{SLAM} \\ \delta \boldsymbol{X}_w \\ \{\delta \boldsymbol{X}_{GNSS}\} \end{bmatrix} \tag{4.16}$$

where $\delta \boldsymbol{X}_{GNSS}$ is put between braces because it is removed in case of GNSS loose coupling and kept in case of tight coupling.

4.4 State transition model

4.4.1 Continuous-time model

The process is modeled by the temporal behavior of the state vector components. It is defined by two matrices: the $\boldsymbol{F}$ matrix which describes the system dynamic evolution with time after linearization, and the $\boldsymbol{Q}$ matrix which describes the system noise.

The behavior of the navigation parameters, i.e. the position, velocity and attitude errors, are obtained by applying the following perturbation model to the INS mechanization given in equation (2.28) and neglecting terms of order higher than first:

$$\begin{aligned} \widehat{\boldsymbol{p}}_b^l &= \boldsymbol{p}_b^l + \delta \boldsymbol{p}_b^l \\ \widehat{\boldsymbol{v}}_b^l &= \boldsymbol{v}_b^l + \delta \boldsymbol{v}_b^l \\ \widehat{\boldsymbol{C}}_{b2l} &= (\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\, \boldsymbol{C}_{b2l} \end{aligned} \tag{4.17}$$

where

- $\widehat{\ }$ and δ denote estimated values and errors, respectively
- $\boldsymbol{E}_{b2l} = (\boldsymbol{\varepsilon}_{b2l})\times$ is the skew-symmetric matrix of the vehicle attitude

The following system is obtained:

$$\begin{bmatrix} \delta \dot{\boldsymbol{p}}_b^l \\ \delta \dot{\boldsymbol{v}}_b^l \\ \dot{\boldsymbol{\varepsilon}}_{b2l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{F}_{vp} & \boldsymbol{F}_{vv} & (\boldsymbol{C}_{b2l}\boldsymbol{f}_{ib}^b)\times \\ \boldsymbol{F}_{ep} & \boldsymbol{F}_{ev} & -(\boldsymbol{\omega}_{il}^l)\times \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{p}_b^l \\ \delta \boldsymbol{v}_b^l \\ \boldsymbol{\varepsilon}_{b2l} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{C}_{b2l} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & -\boldsymbol{C}_{b2l} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{f}_{ib}^b \\ \delta \boldsymbol{\omega}_{ib}^b \end{bmatrix} \tag{4.18}$$

where

- $\boldsymbol{F}_{vp} = \begin{bmatrix} 0 & \frac{2\omega_e(v_b^N cos(\varphi)+v_b^U sin(\varphi))}{R_m+h} + \frac{v_b^E v_b^N}{(R_n+h)(R_m+h)cos^2(\varphi)} & \frac{v_b^E v_b^U - v_b^E v_b^N tan(\varphi)}{(R_n+h)^2} \\ 0 & \frac{-2\omega_e v_b^E cos(\varphi)}{R_m+h} - \frac{{v_b^E}^2}{(R_n+h)(R_m+h)cos^2(\varphi)} & \frac{v_b^N v_b^U}{(R_m+h)^2} + \frac{{v_b^E}^2 \tan(\varphi)}{(R_n+h)^2} \\ 0 & -\frac{2\omega_e v_b^E sin(\varphi)}{R_m+h} & \frac{-{v_b^E}^2}{(R_n+h)^2} - \frac{{v_b^N}^2}{(R_m+h)^2} + \frac{2g^l}{R+h} \end{bmatrix}$
- $F_{vv} = \begin{bmatrix} \frac{v_b^N \tan(\varphi) - v_b^U}{R_n+h} & \frac{v_b^E \tan(\varphi)}{R_n+h} + 2\omega_e \sin(\varphi) & -2\omega_e \cos(\varphi) - \frac{v_b^E}{R_n+h} \\ \frac{-2v_b^E \tan(\varphi)}{R_n+h} - 2\omega_e \sin(\varphi) & \frac{-v_b^U}{R_m+h} & \frac{-v_b^N}{R_m+h} \\ \frac{2v_b^E}{R_n+h} + 2\omega_e \cos(\varphi) & \frac{2v_b^N}{R_m+h} & 0 \end{bmatrix}$

- $\boldsymbol{F}_{ep} = \begin{bmatrix} 0 & 0 & \frac{v_b^N}{(R_m+h)^2} \\ 0 & -\frac{\omega_e \sin(\varphi)}{R_m+h} & \frac{-v_b^E}{(R_n+h)^2} \\ 0 & \frac{\omega_e \cos(\varphi)}{R_m+h} + \frac{v_b^E}{(R_n+h)(R_m+h)\cos^2(\varphi)} & \frac{-v_b^E \tan(\varphi)}{(R_n+h)^2} \end{bmatrix}$
- $F_{ev} = \begin{bmatrix} 0 & \frac{-1}{R_m+h} & 0 \\ \frac{1}{R_n+h} & 0 & 0 \\ \frac{\tan(\varphi)}{R_n+h} & 0 & 0 \end{bmatrix}$
- $\delta \boldsymbol{f}_{ib}^b = \tilde{\boldsymbol{f}}_{ib}^b - \boldsymbol{f}_{ib}^b$ is the accelerometer error vector derived from equation (2.27):

$$\delta \boldsymbol{f}_{ib}^b = \boldsymbol{\gamma}(\boldsymbol{k}_f)\boldsymbol{f}_{ib}^b + \boldsymbol{b}_f + \boldsymbol{\eta}_f \tag{4.19}$$

where

$\boldsymbol{\gamma}(\boldsymbol{k}_f) = \text{diag}\left(\frac{1}{1+k_{fx}}, \frac{1}{1+k_{fy}}, \frac{1}{1+k_{fz}}\right)$

- $\delta \boldsymbol{\omega}_{ib}^b = \tilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{\omega}_{ib}^b$ is the gyroscope error vector derived from equation (2.27)

$$\delta \boldsymbol{\omega}_{ib}^b = \boldsymbol{\gamma}(\boldsymbol{k}_\omega)\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_\omega + \boldsymbol{\eta}_\omega \tag{4.20}$$

where

$\boldsymbol{\gamma}(\boldsymbol{k}_\omega) = \text{diag}\left(\frac{1}{1+k_{\omega x}}, \frac{1}{1+k_{\omega y}}, \frac{1}{1+k_{\omega z}}\right)$

- The IMU biases and scale factors are modeled as Gauss-Markov processes as discussed in Section 2.2.2.
- The state vector components related to the WSS and the SLAM are modeled as random walks

Therefore, the resulting continuous system process model is given by:

<u>If the GNSS is used in a loose configuration, then no additional states are related to GNSS</u>

$$\begin{bmatrix} \delta\dot{\boldsymbol{p}}_b^l \\ \delta\dot{\boldsymbol{v}}_b^l \\ \dot{\boldsymbol{\varepsilon}}_{b2l} \\ \delta\dot{\boldsymbol{b}}_f \\ \delta\dot{\boldsymbol{b}}_\omega \\ \delta\dot{\boldsymbol{k}}_f \\ \delta\dot{\boldsymbol{k}}_\omega \\ \delta\dot{k}_w \\ \dot{k}_v \\ \dot{\boldsymbol{\varepsilon}}_{l2v} \\ \delta\dot{\boldsymbol{p}}_{lv}^v \\ \dot{\boldsymbol{\varepsilon}}_{c2b} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times11} \\ \boldsymbol{F}_{vp} & \boldsymbol{F}_{vv} & (\boldsymbol{C}_{b2l}\boldsymbol{f}_{ib}^b)\times & \boldsymbol{C}_{b2l} & \mathbf{0}_3 & \boldsymbol{C}_{b2l}\boldsymbol{F}^b & \mathbf{0}_3 & \mathbf{0}_{3\times11} \\ \boldsymbol{F}_{ep} & \boldsymbol{F}_{ev} & -(\boldsymbol{\omega}_{il}^l)\times & \mathbf{0}_3 & -\boldsymbol{C}_{b2l} & \mathbf{0}_3 & -\boldsymbol{C}_{b2l}\boldsymbol{W}^b & \mathbf{0}_{3\times11} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\beta}_{b_f} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times11} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\beta}_{b_\omega} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times11} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\beta}_{k_f} & \mathbf{0}_3 & \mathbf{0}_{3\times11} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{\beta}_{k_\omega} & \mathbf{0}_{3\times11} \\ \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} & \mathbf{0}_{11\times3} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{p}_b^l \\ \delta\boldsymbol{v}_b^l \\ \boldsymbol{\varepsilon}_{b2l} \\ \delta\boldsymbol{b}_f \\ \delta\boldsymbol{b}_\omega \\ \delta\boldsymbol{k}_f \\ \delta\boldsymbol{k}_\omega \\ \delta k_w \\ k_v \\ \boldsymbol{\varepsilon}_{l2v} \\ \delta\boldsymbol{p}_{lv}^v \\ \boldsymbol{\varepsilon}_{c2b} \end{bmatrix} \tag{4.21}$$

$$+ \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times23} \\ \boldsymbol{C}_{b2l} & \mathbf{0}_3 & \mathbf{0}_{3\times23} \\ \mathbf{0}_3 & -\boldsymbol{C}_{b2l} & \mathbf{0}_{3\times23} \\ \mathbf{0}_{23\times3} & \mathbf{0}_{23\times3} & \boldsymbol{I}_{23} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_\omega \\ \boldsymbol{\eta}_{b_f} \\ \boldsymbol{\eta}_{b_\omega} \\ \boldsymbol{\eta}_{k_f} \\ \boldsymbol{\eta}_{k_\omega} \\ \boldsymbol{\eta}_{k_w} \\ \boldsymbol{\eta}_{k_v} \\ \boldsymbol{\eta}_{\varepsilon_{l2v}} \\ \boldsymbol{\eta}_{p_{lv}} \\ \boldsymbol{\eta}_{\varepsilon_{c2b}} \end{bmatrix} = \boldsymbol{F}\delta\boldsymbol{X} + \boldsymbol{G}\boldsymbol{\eta}$$

where

- $\boldsymbol{F}^b = \text{diag}\left(\boldsymbol{f}_{ib}^b\right)$ and $\boldsymbol{W}^b = \text{diag}\left(\boldsymbol{\omega}_{ib}^b\right)$
- $\boldsymbol{\beta}_{b_f} = \text{diag}\left(-1/\tau_{bfx}, -1/\tau_{bfy}, -1/\tau_{bfz}\right)$
- $\boldsymbol{\beta}_{\omega_f} = \text{diag}\left(-1/\tau_{b\omega x}, -1/\tau_{b\omega y}, -1/\tau_{b\omega z}\right)$
- $\boldsymbol{\beta}_{k_f} = \text{diag}\left(-1/\tau_{kfx}, -1/\tau_{kfy}, -1/\tau_{kfz}\right)$
- $\boldsymbol{\beta}_{k_\omega} = \text{diag}\left(-1/\tau_{k\omega x}, -1/\tau_{k\omega y}, -1/\tau_{k\omega z}\right)$

where τ_u is the correlation time of the Gauss-Markov process u.

If the GPS is used in a tight configuration, then the two states corresponding to the receiver clock bias b_G^{clk} and drift $\dot{b}_G^{clk}$ are added to the previous system. The two states are:

$$\begin{bmatrix} \delta\dot{b}_G^{clk} \\ \delta\ddot{b}_G^{clk} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta b_G^{clk} \\ \delta\dot{b}_G^{clk} \end{bmatrix} + \begin{bmatrix} \eta_{b_G} \\ \eta_{\dot{b}_G} \end{bmatrix} \tag{4.22}$$

where η_{b_G} and $\eta_{\dot{b}_G}$ are the clock erros whose spectral densities are given by:

$$\begin{aligned} q_{b_G} &= c^2 . \frac{h_0}{2} \\ q_{\dot{b}_G} &= c^2 . 2\pi^2 . h_{-2} \end{aligned} \tag{4.23}$$

where h_0 and h_{-2} are allan variance parameters describing the clock errors, typical values for compensated crystal clock are $2 . 10^{-19}$ and $2 . 10^{-20}$ respectively (Brown, 1997).

4.4.2 Discrete-time model

Since all the sensors are implemented with sampled data, the previous continuous time model must be discretized. The discretization steps are described in (Shin, 2001). Since the sampling time interval $\Delta t = t_{k+1} - t_k$ is very small, then the following approximation of the state transition matrix is considered:

$$\boldsymbol{\Phi}_k \approx \boldsymbol{I} + \boldsymbol{F}_k \Delta t \tag{4.24}$$

The discretization of the spectral density matrix $\boldsymbol{Q}$ is given by:

$$\boldsymbol{Q}_k = E[\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T] \approx \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T \Delta t \tag{4.25}$$

where

$$\boldsymbol{Q} = \text{diag}\left(\boldsymbol{q}_{\eta f}, \boldsymbol{q}_{\eta\omega}, \boldsymbol{q}_{\eta b_f}, \boldsymbol{q}_{\eta b_\omega}, \boldsymbol{q}_{\eta k_f}, \boldsymbol{q}_{\eta k_\omega}, q_{\eta k_w}, q_{\eta k_v}, \boldsymbol{q}_{\eta \varepsilon_{l2v}}, \boldsymbol{q}_{\eta p_{lv}}, \boldsymbol{q}_{\eta \varepsilon_{c2b}}\right)$$

with

- $q_{\eta u} = \sigma_u^2$ if u is a white Gaussian process with a variance equal to σ_u^2
- $q_{\eta u} = 2 * \sigma_u^2 / \tau_u$ if u is a Gauss-Markov process with a correlation time equal to τ_u and a noise variance equal to σ_u^2

4.5 Observation model

The update phase of the EKF is performed when acquiring the aiding sensors measurements. These measurements at time k are related to the state vector $\delta\boldsymbol{X}$ by the following linear equation:

$$\delta\boldsymbol{Z}_k = h(\delta\boldsymbol{X}_k) + \boldsymbol{\varepsilon}_k \tag{4.26}$$

where $\delta\boldsymbol{Z}_k$ is the difference between the measurement prediction $\widehat{\boldsymbol{Z}}$ computed with the state already propagated at the prediction phase, and the measurement observation $\widetilde{\boldsymbol{Z}}$ provided by the sensor. $\delta\boldsymbol{Z}_k$ is also called the measurement innovation.

The expressions of the nonlinear function h and its Jacobian $\boldsymbol{H}_k$ will be given for each sensor in the following sections.

4.5.1 GNSS observation model

In the case of loose coupling, the GNSS receiver provides the processed antenna position $\widetilde{\boldsymbol{p}}_G^l$ and velocity $\widetilde{\boldsymbol{v}}_G^l$. Taking into account the lever arm $\Delta\boldsymbol{p}_G^b$ between the GNSS antenna and the IMU, the GNSS measurements are given by:

$$\widetilde{\boldsymbol{p}}_G^l = \boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\ \Delta\boldsymbol{p}_G^b + \boldsymbol{\varepsilon}_{G,p} \tag{4.27}$$

$$\widetilde{\boldsymbol{v}}_G^l = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\left(\boldsymbol{\omega}_{lb}^b\right) \times \Delta\boldsymbol{p}_G^b\right) + \boldsymbol{\varepsilon}_{G,v} \tag{4.28}$$

where $\boldsymbol{\varepsilon}_{G,p}$ and $\boldsymbol{\varepsilon}_{G,v}$ are additive zero-mean, white and Gaussian noises whose variances are provided by the GNSS navigation module.

The GNSS measurement innovation is given by:

$$\delta\boldsymbol{Z}_G = \begin{bmatrix} \widehat{\boldsymbol{p}}_b^l - \widetilde{\boldsymbol{p}}_G^l + \widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_G^b \\ \widehat{\boldsymbol{v}}_b^l - \widetilde{\boldsymbol{v}}_G^l + \widehat{\boldsymbol{C}}_{b2l}\left(\boldsymbol{\omega}_{lb}^b\right) \times \Delta\boldsymbol{p}_G^b \end{bmatrix} \tag{4.29}$$

The Jacobian matrix associated to equation (4.27) and equation (4.28) is given by:

$$\boldsymbol{H}_G = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \left(\widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_G^b\right)\times & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_{3\times11} \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \left(\widehat{\boldsymbol{C}}_{b2l}\left(\omega_{ib}^b \times \Delta\boldsymbol{p}_G^b\right)\right)\times & \boldsymbol{0}_3 & -\widehat{\boldsymbol{C}}_{b2l}(\Delta\boldsymbol{p}_G^b)\times & \boldsymbol{0}_3 & -\widehat{\boldsymbol{C}}_{b2l}(\Delta\boldsymbol{p}_G^b)\times\boldsymbol{W} & \boldsymbol{0}_{3\times11} \end{bmatrix} \tag{4.30}$$

In the case of tight coupling, the GNSS receiver provides the pseudoranges $\boldsymbol{\rho}_G$ and the Doppler measurements $\Delta\tilde{\boldsymbol{f}}_G$. The Doppler measurements are converted into pseudorange rates $\dot{\boldsymbol{\rho}}_G$ using equation (2.2). Taking into account the lever arm $\Delta\boldsymbol{p}_G^b$ between the GNSS antenna and the IMU, the GNSS raw measurements are given by:

$$\widetilde{\boldsymbol{\rho}}_G = \begin{bmatrix} \tilde{\rho}_G^1 \\ \vdots \\ \tilde{\rho}_G^{N_{sat}} \end{bmatrix} = \begin{bmatrix} \sqrt{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{s1}^l\right)^T\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{s1}^l\right)} + b_G^{clk} + \varepsilon_{G,PR}^1 \\ \vdots \\ \sqrt{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{sN_{sat}}^l\right)^T\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{sN_{sat}}^l\right)} + b_G^{clk} + \varepsilon_{G,PR}^{N_{sat}} \end{bmatrix} \tag{4.31}$$

$$\widetilde{\dot{\boldsymbol{\rho}}}_G = \begin{bmatrix} \tilde{\dot{\rho}}_G^1 \\ \vdots \\ \tilde{\dot{\rho}}_G^{N_{sat}} \end{bmatrix} = \begin{bmatrix} \dfrac{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{s1}^l\right)^T\left(\boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{lb}^b\right)\times\Delta\boldsymbol{p}_G^b - \boldsymbol{v}_{s1}^l\right)}{d^1} + \dot{b}_G^{clk} + \varepsilon_{G,PRR}^1 \\ \vdots \\ \dfrac{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{sN_{sat}}^l\right)^T\left(\boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{lb}^b\right)\times\Delta\boldsymbol{p}_G^b - \boldsymbol{v}_{sN_{sat}}^l\right)}{d^{N_{sat}}} + \dot{b}_G^{clk} + \varepsilon_{G,PRR}^{N_{sat}} \end{bmatrix} \tag{4.32}$$

where

- N_{sat} is the number of satellites in view
- $\boldsymbol{p}_{si}^l$ is the position of the satellite i in (l)
- $\boldsymbol{v}_{si}^l$ is the velocity of the satellite i in (l)
- $d^i = \sqrt{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)}$ is the geometric range between the satellite i and the GNSS antenna
- $\varepsilon_{G,PR}^i$ and $\varepsilon_{G,PRR}^i$ are additive zero-mean, white and Gaussian noises whose variances are given in Section 2.1.

The GNSS raw measurements are linearized by expansion in Taylor series as explained in equation (2.18). The lever arm is compensated in the prediction of the measurements using equation (4.31) and equation (4.32). However, the measurement dependency on the attitude

error $\boldsymbol{\varepsilon}_{b2l}$ introduced by the lever arm will be assumed negligible, considering the small contribution of the lever arm to the direction cosine angles defining the observation matrix.

Therefore, the GNSS measurement innovation is given by:

$$\delta \boldsymbol{Z}_G = \begin{bmatrix} \tilde{\rho}_G^1 - \hat{\rho}_G^1 \\ \vdots \\ \tilde{\rho}_G^{N_{sat}} - \hat{\rho}_G^{N_{sat}} \\ \tilde{\dot{\rho}}_G^1 - \hat{\dot{\rho}}_G^1 \\ \vdots \\ \tilde{\dot{\rho}}_G^{N_{sat}} - \hat{\dot{\rho}}_G^{N_{sat}} \end{bmatrix} \tag{4.33}$$

The Jacobian matrix associated to equation (4.31) and equation (4.32) is given by:

$$\boldsymbol{H}_G = \begin{bmatrix} \boldsymbol{H}_{G,pPR_1} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times26} & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{G,pPR_{N_{sat}}} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times26} & -1 & 0 \\ \boldsymbol{H}_{G,pPRR_1} & \boldsymbol{H}_{G,vPRR_1} & \boldsymbol{0}_{1\times26} & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{G,pPRR_{N_{sat}}} & \boldsymbol{H}_{G,vPRR_N_{sat}} & \boldsymbol{0}_{1\times26} & 0 & -1 \end{bmatrix} \tag{4.34}$$

where

- $\boldsymbol{H}_{G,pPR_i} = \boldsymbol{H}_{G,vPRR_i} = \frac{-\left(\hat{\boldsymbol{p}}_b^l + \hat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T}{\hat{d}^i}$
- $\boldsymbol{H}_{G,pPRR_i} = \frac{-\left(\hat{\boldsymbol{v}}_b^l + \hat{\boldsymbol{C}}_{b2l}\left(\hat{\boldsymbol{\omega}}_{lb}^b\right)\times\Delta\boldsymbol{p}_G^b - \boldsymbol{v}_{si}^l\right)^T}{\hat{d}^i} - \frac{\left(\hat{\boldsymbol{p}}_b^l + \hat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \hat{r}_{si}}{\left(\hat{d}^i\right)^2}$

The computation details of matrix $\boldsymbol{H}_G$ are given in Appendix C.

4.5.2 SLAM observation model

As explained in Section 3.4, the SLAM provides the filter with the camera position and attitude in (v) and its outputs are expressed as a function of the filter states in equation (4.13) and equation (4.14). Therefore, the SLAM measurement innovation is given by:

$$\delta \boldsymbol{Z}_v = \begin{bmatrix} \hat{k}_v\,\hat{\boldsymbol{C}}_{l2v}\left(\hat{\boldsymbol{p}}_b^l + \hat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right) + \hat{\boldsymbol{p}}_{lv}^v - \tilde{\boldsymbol{p}}_c^v \\ \hat{\boldsymbol{q}}_{b2c} \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \hat{\boldsymbol{q}}_{v2l} - \tilde{\boldsymbol{q}}_{v2c} \end{bmatrix} \tag{4.35}$$

The Jacobian matrix associated to equation (4.13) and equation (4.14) is given by:

$$\boldsymbol{H}_v = \begin{bmatrix} \boldsymbol{H}_{v,pp} & \boldsymbol{0}_3 & \boldsymbol{H}_{v,p\varepsilon_{bl}} & \boldsymbol{0}_{3\times13} & \boldsymbol{H}_{v,pk_v} & \boldsymbol{H}_{v,p\varepsilon_{lv}} & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \{\boldsymbol{0}_{3\times2}\} \\ \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{H}_{v,q\varepsilon_{bl}} & \boldsymbol{0}_{4\times13} & \boldsymbol{0}_{4\times1} & \boldsymbol{H}_{v,q\varepsilon_{lv}} & \boldsymbol{0}_{4\times3} & \boldsymbol{H}_{v,q\varepsilon_{cb}} & \{\boldsymbol{0}_{4\times2}\} \end{bmatrix} \tag{4.36}$$

where

- $\boldsymbol{H}_{v,pp} = \hat{k}_v\,\hat{\boldsymbol{C}}_{l2v}$
- $\boldsymbol{H}_{v,p\varepsilon_{bl}} = \hat{k}_v\,\hat{\boldsymbol{C}}_{l2v}\left(\hat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)\times$

- $\boldsymbol{H}_{v,pk_v} = \widehat{\boldsymbol{C}}_{l2v}\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)$
- $\boldsymbol{H}_{v,p\varepsilon_{lv}} = \hat{k}_v\left(\left(\widehat{\boldsymbol{C}}_{l2v}\widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)\times + \left(\widehat{\boldsymbol{C}}_{l2v}\widehat{\boldsymbol{p}}_b^l\right)\times\right)$
- $\boldsymbol{H}_{v,q\varepsilon_{bl}} = -\frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{l2b})$
- $\boldsymbol{H}_{v,q\varepsilon_{lv}} = -\frac{1}{2}\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{l2b})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{v2l})$
- $\boldsymbol{H}_{v,q\varepsilon_{cb}} = \frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{l2b})\mathbb{E}(\widehat{\boldsymbol{q}}_{b2c})$

The expressions of $\boldsymbol{\mathcal{R}}$, $\boldsymbol{\mathcal{L}}$, $\mathbb{E}$ and $\boldsymbol{\Psi}$ as well as the computation of the matrix $\boldsymbol{H}_v$ are given in Appendix B and Appendix C respectively.

4.5.3 WSS and NHC observation model

The WSS provides the forward vehicle velocity as and the NHC complete the three-dimensional velocity by assuming that a ground vehicle moves only in straightforward direction and its lateral and vertical velocities should be equal to zero if there is no slip or jump. Therefore, the combined WSS and NHC velocity model is given by:

$$\tilde{\boldsymbol{v}}_w^b = \begin{bmatrix}0 & \tilde{v}_w^{by} & 0\end{bmatrix}^T \tag{4.37}$$

The expression of the WSS velocity expressed in (b) as a function of the IMU velocity $\boldsymbol{v}_b^l$ expressed in (l) takes into account three parameters. First, the WSS scale factor k_w should be considered. Then, the lever arm $\Delta\boldsymbol{p}_w^b$ between the point where the WSS velocity is calculated (vehicle rear or front wheel axle center) and the IMU should be taken into account. And finally, the rotation between frames (b) and (l) should be considered. Therefore, the WSS velocity is written as:

$$\tilde{\boldsymbol{v}}_w^b = \boldsymbol{D}\left(\boldsymbol{C}_{l2b}\boldsymbol{v}_b^l + \left(\boldsymbol{\omega}_{lb}^b\right)\times\Delta\boldsymbol{p}_w^b\right) + \boldsymbol{\varepsilon}_w \tag{4.38}$$

where

$$\boldsymbol{D} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1+k_w & 0\\ 0 & 0 & 1\end{bmatrix} \tag{4.39}$$

The measurement innovation is given by:

$$\delta\boldsymbol{Z}_{wss} = \widehat{\boldsymbol{D}}\left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\boldsymbol{\omega}_{lb}^b\right)\times\Delta\boldsymbol{p}_w^b\right) - \tilde{\boldsymbol{v}}_w^b \tag{4.40}$$

The Jacobian matrix associated to equation (4.38) is given by:

$$\boldsymbol{H}_w = [\boldsymbol{0}_3 \quad \boldsymbol{H}_{w,v} \quad \boldsymbol{H}_{w,\varepsilon_{b2l}} \quad \boldsymbol{0}_3 \quad \boldsymbol{H}_{w,b_\omega} \quad \boldsymbol{0}_3 \quad \boldsymbol{H}_{w,k_\omega} \quad \boldsymbol{H}_{w,k_w} \quad \boldsymbol{0}_{3\times10} \quad \{\boldsymbol{0}_{3\times2}\}] \tag{4.41}$$

where

- $\boldsymbol{H}_{w,v} = \widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{C}}_{l2b}$
- $\boldsymbol{H}_{w,\varepsilon_{b2l}} = -\widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{C}}_{l2b}\left(\widehat{\boldsymbol{v}}_b^l\right)\times$
- $\boldsymbol{H}_{w,b_\omega} = -\widehat{\boldsymbol{D}}\,(\Delta\boldsymbol{p}_w^b)\times$

- $\boldsymbol{H}_{w,k_\omega} = -\widehat{\boldsymbol{D}}\,(\Delta \boldsymbol{p}_w^b) \times \boldsymbol{W}^b$
- $\boldsymbol{H}_{w,k_w} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\boldsymbol{\omega}_{lb}^b\right) \times \Delta \boldsymbol{p}_w^b\right)$

The computation details of the matrix $\boldsymbol{H}_w$ are given in Appendix C.

4.5.4 ZARU observation model

ZARU consists in considering that the angular rate of the vehicle is equal to zero when a stationarity or a linear motion is detected. In this case, the measurement innovation is given by the IMU gyroscope measurements:

$$\delta \boldsymbol{Z}_{zaru} = \widetilde{\boldsymbol{\omega}}_{lb}^b \tag{4.42}$$

Based on this equation, the Jacobin matrix $\boldsymbol{H}_{zaru}$ matrix is easily derived:

$$\boldsymbol{H}_{zaru} = [\boldsymbol{0}_{3\times12} \quad \boldsymbol{I}_3 \quad \boldsymbol{0}_{3\times17} \quad \{\boldsymbol{0}_{3\times2}\}] \tag{4.43}$$

The noise covariance corresponding to ZARU represents the variance of the nominally-zero angular rate due to vibration and disturbances. Moreover, the heading axis is less affected by disturbances than the other two axes and this should be reflected in the assumed measurement noise (Groves, 2008).

4.6 Observability analysis

The use of many sensors in the proposed architecture introduces numerous additional states to be estimated by the Kalman filter. Therefore, a question raises in this case. Is it possible to achieve successful estimation of all the state vector components, given the available measurements? To answer this question, an observability analysis of the proposed filter states is conducted.

Observability can be defined as the ability of recovering the state values from the available measurements, the known control inputs, and a finite number of their time derivatives (Conte, 2006). Observability is a necessary condition for any filtering algorithm to converge to an unbiased estimate of the true system state (Lee, 1982). For an unobservable system, we cannot achieve successful estimation, even though the measurements are accurate enough.

For linear time-invariant systems, many observability studies are proposed in the literature. The most widely known studies are the linear observability rank test (Maybeck, 1979) or the eigenvector test (Verhaegen, 2007). However, the considered system in this thesis is time-varying and time-invariant approaches are consequently not applicable.

For linear time-varying systems, the observability analysis involves evaluating the observability Gramian matrix, which must usually be done numerically (Kelly, 2010). An alternative approach proposed in (Goshen-Meskin, 1992) approximates linear time-varying systems by a piecewise constant model and use the stripped observability matrix. Since the proposed system is not linear, this approach can be applied on the linearized system. However, the linearization

implies that the observability depends on the linearization point and not only on the system itself. In addition, linearization can introduce additional observable directions that are unobservable with the non-linear system. Therefore, analyzing the observability of the nonlinear system is the best approach to know whether the system, as it is defined above, is fully observable or not, regardless of the linearization point.

Hence, the nonlinear system observability analysis we conduct is based on the method proposed by (Hermann, 1977) and detailed in (Kelly, 2010), based on the differential geometry and the Lie derivatives. To conduct this analysis, we consider the full state variables and not the error state since the definition of the error state is an approximation where second and higher order terms are discarded under the assumption of small error state (Weiss, 2012).

The considered approach determines whether a nonlinear system is locally weakly observable, i.e., if for each point $\boldsymbol{x}_0 \in M \subseteq \mathbb{R}^m$, there exists an open neighborhood $U \subseteq M$ such that for every open neighborhood V of $\boldsymbol{x}_0$, $V \subseteq M$, there is no other state $\boldsymbol{x}_1 \in V$ that is indistinguishable from $\boldsymbol{x}_0$ (Hermann, 1977). Note that the local observability concept is stronger than the global concept because in global observability, it might be necessary to travel a considerable distance or for a long time to distinguish between the states and their neighbors, while local concept deals with the ability to distinguish states from their neighbors in a small-time interval or instantaneously.

4.6.1 Observability analysis using Lie derivatives

Let us consider the system:

$$\begin{cases} \dot{\boldsymbol{x}} = f_0(\boldsymbol{x}) + \sum_{i=1}^{p} f_i(\boldsymbol{x}) u_i \\ z = h(\boldsymbol{x}) \end{cases} \tag{4.44}$$

The Lie derivative of h with respect to a function f at $\boldsymbol{x}$ is given by:

$$L_f h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \tag{4.45}$$

The recursive definition of Lie derivative is given by:

$$L_g L_f h(\boldsymbol{x}) = \frac{\partial L_f h(\boldsymbol{x})}{\partial \boldsymbol{x}} g(\boldsymbol{x}) \tag{4.46}$$

Now, consider the system above, and let $\boldsymbol{\mathcal{O}}$ be its observability matrix, whose rows are formed by the gradients of the Lie derivatives of $h(\boldsymbol{x})$:

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \nabla L^0 \boldsymbol{h}_1 \\ \nabla L^0 \boldsymbol{h}_2 \\ \nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 \\ \nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_2 \\ \nabla L^1_{\boldsymbol{f}_1} \boldsymbol{h}_1 \\ \vdots \\ \nabla L_{\boldsymbol{f}_i} L^1_{\boldsymbol{f}_j} \boldsymbol{h}_i \\ \vdots \end{bmatrix} \tag{4.47}$$

$\boldsymbol{\mathcal{O}}$ has an infinite number of rows and the same number of columns as the state vector component number.

The system is said to be locally weakly observable at $\boldsymbol{x}_0$ if $\boldsymbol{\mathcal{O}}$ has full column rank at $\boldsymbol{x}_0$ (Kelly, 2010), i.e. the rank of $\boldsymbol{\mathcal{O}}$ is equal to the state vector length. To prove the full column rank of $\boldsymbol{\mathcal{O}}$, it suffices to show that a submatrix of $\boldsymbol{\mathcal{O}}$ comprising a subset of its rows such as the submatrix is square and invertible.

4.6.2 Nonlinear system formulation

Let us develop the nonlinear system derived from the INS mechanization in equation (2.28):

$$\begin{bmatrix} \dot{\boldsymbol{p}}^l_b \\ \dot{\boldsymbol{v}}^l_b \\ \dot{\boldsymbol{q}}_{b2l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^l_b \\ \boldsymbol{C}_{b2l} \boldsymbol{f}^b_{ib} - \left(2\boldsymbol{\omega}^l_{ie} + \boldsymbol{\omega}^l_{el}\right) \times \boldsymbol{v}^l_b + \boldsymbol{g}^l \\ -\frac{1}{2}\, \boldsymbol{q}_{b2l} \otimes \boldsymbol{\omega}^b_{bl} \end{bmatrix} \tag{4.48}$$

Using a low-cost IMU, the earth rotation is buried in sensor errors, and cannot be detected by the sensor. Thus, the Coriolis term is not considered. Moreover, for short distance applications, the transport rate is negligible. These justified assumptions are considered for the observability analysis and do not have any effect on it. Thus, the nonlinear system becomes:

$$\begin{bmatrix} \dot{\boldsymbol{p}}^l_b \\ \dot{\boldsymbol{v}}^l_b \\ \dot{\boldsymbol{q}}_{b2l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}^l_b \\ \boldsymbol{C}_{b2l} \boldsymbol{f}^b_{ib} + \boldsymbol{g}^l \\ \frac{1}{2}\, \boldsymbol{q}_{b2l} \otimes \boldsymbol{\omega}^b_{bl} \end{bmatrix} \tag{4.49}$$

The quaternion can also be written as:

$$\dot{\boldsymbol{q}}_{b2l} = \frac{1}{2} \boldsymbol{\Psi}(\, \boldsymbol{q}_{b2l}) \boldsymbol{\omega}^b_{ib} \tag{4.50}$$

where $\boldsymbol{\Psi}$ is given in Appendix B.

Considering this system and the IMU measurement models given in equations (4.19) and equation (4.20), the nonlinear system propagation equation is given by:

$$\dot{\boldsymbol{X}}_{INS} = \begin{bmatrix} \dot{\boldsymbol{p}}_b^l \\ \dot{\boldsymbol{v}}_b^l \\ \dot{\boldsymbol{q}}_{b2l} \\ \dot{\boldsymbol{b}}_f \\ \dot{\boldsymbol{b}}_\omega \\ \dot{\boldsymbol{k}}_f \\ \dot{\boldsymbol{k}}_\omega \\ k_w \\ k_v \\ \dot{\boldsymbol{q}}_{l2v} \\ \dot{\boldsymbol{p}}_{lv}^v \\ \dot{\boldsymbol{q}}_{c2b} \\ \{\dot{b}_G^{clk}\} \\ \{\ddot{b}_G^{clk}\} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{v}_b^l \\ \boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f)\boldsymbol{b}_f \\ \frac{-1}{2}\boldsymbol{\Psi}(\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\boldsymbol{b}_\omega \\ \boldsymbol{\beta}_{b_f}\boldsymbol{b}_f \\ \boldsymbol{\beta}_{b_\omega}\boldsymbol{b}_\omega \\ \boldsymbol{\beta}_{k_f}\boldsymbol{k}_f \\ \boldsymbol{\beta}_{k_\omega}\boldsymbol{k}_\omega \\ 0 \\ 0 \\ \boldsymbol{0}_{4\times1} \\ \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{4\times1} \\ \dot{b}_G^{clk} \\ 0 \end{bmatrix}}_{f_0(\boldsymbol{X})} + \underbrace{\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \frac{1}{2}\boldsymbol{\Psi}(\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega) \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} \end{bmatrix}}_{f_1(\boldsymbol{X})} \tilde{\omega}_{ib}^b + \underbrace{\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f) \\ \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_{4\times3} \\ \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} \end{bmatrix}}_{f_2(\boldsymbol{X})} \tilde{f}_{ib}^b \qquad (4.51)$$

where $\boldsymbol{C}_{b2l}$ is a function of the components of the quaternion $\boldsymbol{q}_{b2l}$ as explained in Appendix B.

4.6.3 **Measurement formulation**

For each aiding sensor, we can write the measurement functions as:

GNSS position:

$$\boldsymbol{h}_1 = \boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\ \Delta\boldsymbol{p}_G^b \qquad (4.52)$$

GNSS velocity:

$$\begin{aligned} \boldsymbol{h}_2 &= \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{ib}^b\right) \times \Delta\boldsymbol{p}_G^b \\ &= \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\tilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta\boldsymbol{p}_G^b \end{aligned} \qquad (4.53)$$

WSS and NHC velocity:

$$\boldsymbol{h}_3 = \boldsymbol{D}\left(\boldsymbol{C}_{l2b}\boldsymbol{v}_b^l + \left(\gamma(\boldsymbol{k}_\omega)\left(\tilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta\boldsymbol{p}_w^b\right) \qquad (4.54)$$

SLAM position:

$$\boldsymbol{h}_4 = k_v \, \boldsymbol{C}_{l2v} \left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l} \, \Delta \boldsymbol{p}_c^b \right) + \boldsymbol{p}_{lv}^v \tag{4.55}$$

SLAM attitude:

$$\boldsymbol{h}_5 = (\boldsymbol{q}_{b2c} \otimes \boldsymbol{q}_{l2b} \otimes \boldsymbol{q}_{v2l}) \tag{4.56}$$

Quaternion constraints:

$$h_6 = \boldsymbol{q}_{b2l}^T . \boldsymbol{q}_{b2l} = 1 \tag{4.57}$$

$$h_7 = \boldsymbol{q}_{l2v}^T . \boldsymbol{q}_{l2v} = 1 \tag{4.58}$$

$$h_8 = \boldsymbol{q}_{c2v}^T . \boldsymbol{q}_{c2v} = 1 \tag{4.59}$$

The Lie derivatives of the measurement functions are computed as follows:

$$\begin{gathered} L^0 \boldsymbol{h}_i = \boldsymbol{h}_i \\ L^1_{\boldsymbol{f}_j} \boldsymbol{h}_i = \nabla L^0 \boldsymbol{h}_i . \boldsymbol{f}_j \\ L^n_{\boldsymbol{f}_j} \boldsymbol{h}_i = \nabla L^{n-1}_{\boldsymbol{f}_j} \boldsymbol{h}_i . \boldsymbol{f}_j \\ L_{\boldsymbol{f}_j} L^n_{\boldsymbol{f}_k} \boldsymbol{h}_i = \nabla L^{n-1}_{\boldsymbol{f}_k} \boldsymbol{h}_i . \boldsymbol{f}_j \end{gathered} \tag{4.60}$$

where

- The operators ∇ and . express the Jacobian and the dot product respectively.
- $L^n_{\boldsymbol{f}_j} \boldsymbol{h}_i$ is the n^{th} Lie derivative of $\boldsymbol{h}_i$ along the vector field $\boldsymbol{f}_j$ (if $\boldsymbol{h}_i$ is differentiable n times along $\boldsymbol{f}_j$)

4.6.4 **Observability analysis**

To simplify the observability analysis, the results of previous observability studies are used. Thus the observability analysis of the proposed hybridization architecture is divided into two parts: the observability analysis that has been already done in previous publications and the new contribution of this thesis in the observability analysis.

The observability study of the INS/SLAM system using the Lie derivatives has been done by (Weiss, 2012) , (Kelly, 2010) and (Martinelli, 2011). These studies have shown that with the aid of an absolute system providing the 3D position, i.e. GNSS, all states associated to INS and SLAM are locally weakly observable. Nonetheless, these studies model the IMU measurements with only a bias and a white Gaussian Noise and do not take into account the IMU scale factors which have to be accounted for because of the measurement bad quality of a low-cost IMU. The following paragraph will consider the study of the observability of the nonlinear INS/GNSS/WSS/NHC system including the IMU and WSS measurement scale factors. To our knowledge, this study has not been addressed yet.

To show that all the system states are locally weakly observable, it suffices to show that all the states related to the INS, GNSS and WSS are locally weakly observable, since the NHC does not introduce unknown parameters to be estimated by the filter. All the INS associated state observability will be studied for this architecture, even though it has been done for INS/SLAM system is (Kelly, 2010). This choice is made because we need to be sure that this system is observable regardless of the SLAM, if we had to discard the SLAM information because of a potential failure (loss of features, bad visibility conditions, bad measurements, etc.)

The observability matrix of the INS/GNSS/WSS/NHC system is the following:

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \nabla L^0 \boldsymbol{h}_1 \\ \nabla L^0 \boldsymbol{h}_2 \\ \nabla L^0 h_{32} \\ \nabla L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \\ \nabla L^0 h_6 \\ \nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_2 \\ \nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \\ \nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \end{bmatrix} \tag{4.61}$$

To better understand the expression of $\boldsymbol{\mathcal{O}}$, the expressions of the Lie derivative gradients as functions of the state vector components are written in the Table 4.2. The highlighted boxes (in green and pink) are the boxes that are used to prove that the $\boldsymbol{\mathcal{O}}$ matrix has full column rank, and the corresponding $\boldsymbol{g}_i$ functions are computed. However, the white boxes are not used and therefore the corresponding $\boldsymbol{g}_i$ are not expanded.

	$\boldsymbol{p}_b^l$	$\boldsymbol{v}_b^l$	$\boldsymbol{q}_{b2l}$	$\boldsymbol{b}_f$	$\boldsymbol{b}_\omega$	$\boldsymbol{k}_f$	$\boldsymbol{k}_\omega$	k_w
$\nabla L^0 \boldsymbol{h}_1$	$\boldsymbol{I}_3$	$\boldsymbol{0}_3$	$\boldsymbol{g}_1(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_{3\times1}$
$\nabla L^0 \boldsymbol{h}_2$	$\boldsymbol{0}_3$	$\boldsymbol{I}_3$	$\boldsymbol{g}_2(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{g}_3(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{g}_4(\boldsymbol{X})$	$\boldsymbol{0}_{3\times1}$
$\nabla L^0 h_{32}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_5(\boldsymbol{X})$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_6(\boldsymbol{X})$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_7(\boldsymbol{X})$	$g_8(\boldsymbol{X})$
$\nabla L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_9(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{10}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times1}$
$\nabla L^0 h_6$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$2\boldsymbol{q}_{b2l}^T$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	0
$\nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_2$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{g}_{11}(\boldsymbol{X})$	$\boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f)$	$\boldsymbol{g}_{12}(\boldsymbol{X})$	$\boldsymbol{g}_{13}(\boldsymbol{X})$	$\boldsymbol{g}_{14}(\boldsymbol{X})$	0
$\nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{15}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{16}(\boldsymbol{X})$	$\boldsymbol{g}_{17}(\boldsymbol{X})$	$\boldsymbol{g}_{18}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times1}$
$\nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{19}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{20}(\boldsymbol{X})$	$\boldsymbol{g}_{21}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times1}$

Table 4.2 Expression of the $\boldsymbol{\mathcal{O}}$ matrix

The matrix $\boldsymbol{\mathcal{O}}$ is a (38×23) matrix. To prove that the rank of $\boldsymbol{\mathcal{O}}$ is 23, we use the block Gaussian elimination method explained in (Kelly, 2010). We assume in this study that the roll

of the vehicle is equal to zero and that the alignment of the IMU and the vehicle is perfect so that the gravity does not have an acceleration component on the vehicle lateral axis.

The detailed rank computation is divided in 5 steps and is detailed in Appendix D. In each step, the observability of a state using a sensor measurement is proven.

- **Step 1**: **Observability analysis of the WSS velocity scale factor k_w using h_{32}**

The WSS measurement expression is given by:

$$h_{32} = (1 + k_w)\left(\boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l + l_{vy}\right) \tag{4.62}$$

Where l_{vy} is the y-component of $\boldsymbol{l}_v = \left(\boldsymbol{\omega}_{ib}^b\right)_{true} \times \Delta\boldsymbol{p}_w^b$ representing the lever arm between the WSS velocity computation point and the IMU.

Therefore;

$$\nabla L^0 h_{32} = [\mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_5(\boldsymbol{X}) \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_6(\boldsymbol{X}) \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_7(\boldsymbol{X}) \quad \boldsymbol{g}_8(\boldsymbol{X})] \tag{4.63}$$

where

$$\boldsymbol{g}_8(\boldsymbol{X}) = \boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l + \boldsymbol{l}_{vy} \tag{4.64}$$

We prove in Appendix D that the scalar $g_8(\boldsymbol{X})$ has full column rank (equal to 1) if the vehicle moves, i.e. if the vehicle velocity is not null. This corresponds to intuition since the scale factor of the WSS appears only when the real vehicle velocity is not equal to zero.

- **Step 2: Observability analysis of the vehicle attitude $\boldsymbol{q}_{l2b}$ and the accelerometers scale factors $\boldsymbol{k}_f$ using $\boldsymbol{h}_2$ and $\boldsymbol{h}_6$**

The GNSS velocity measurement expression is given by:

$$\boldsymbol{h}_2 = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta\boldsymbol{p}_G^b \tag{4.65}$$

Therefore;

$$\nabla L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 = [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_9(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{10}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times1}] \tag{4.66}$$

where

$$\boldsymbol{g}_9(\boldsymbol{X}) = \frac{\partial}{\partial \boldsymbol{q}_{b2l}}\left(\boldsymbol{C}_{b2l}\boldsymbol{\gamma}(\boldsymbol{k}_f)\right) \tag{4.67}$$

$$\boldsymbol{g}_{10}(\boldsymbol{X}) = \frac{\partial}{\partial \boldsymbol{k}_f}\left(\boldsymbol{C}_{b2l}\boldsymbol{\gamma}(\boldsymbol{k}_f)\right) \tag{4.68}$$

We study the observability of the attitude and the accelerometers scale factors jointly using the matrix $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{g}_9(\boldsymbol{X}) & \boldsymbol{g}_{10}(\boldsymbol{X}) \\ 2\boldsymbol{q}_{b2l}^T & \boldsymbol{0}_{1\times 3} \end{bmatrix}$.

We prove in Appendix D that the attitude and the accelerometers scale factors are locally weakly observable if the three acceleration axes are excited. In our case of vehicular application, the vehicle lateral axis is only excited during turns, while the forward and downward axes undergo a nonzero acceleration because of the vehicle motion and the gravity effect, respectively. This implies that the acceleration scale factor of the lateral accelerometer axis is only locally weakly observable in turns, while the other axes accelerometer scale factors are locally weakly observable. We prove also that the attitude error is fully observable if at least two IMU axes (the forward and the vertical axes) undergo a nonzero acceleration. When the vehicle is stationary or moves at a constant speed, i.e. the forward acceleration is zero, the only existing acceleration is due to the gravity effect. If the vehicle pitch is nonzero, then the scale factors of the forward and vertical accelerometer axes are observable. This implies also that the attitude is fully observable. However, if the pitch is equal to zero, then only the vertical accelerometer scale factor axis is observable. This implies that there exists an attitude unobservable state because the rank of $\boldsymbol{A}$ is 1 column deficient.

- **Step 3**: **Observability analysis of the accelerometers biases $\boldsymbol{b_f}$ using $\boldsymbol{h_2}$**

Using again equation (4.65), the expression of $\nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_2$ is given by:

$$\nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_2 = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{g}_{11}(\boldsymbol{X}) & \boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f) & \boldsymbol{g}_{12}(\boldsymbol{X}) & \boldsymbol{g}_{13}(\boldsymbol{X}) & \boldsymbol{g}_{14}(\boldsymbol{X}) & 0 \end{bmatrix} \tag{4.69}$$

We prove in Appendix D that the matrix $\boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f)$ has always full column rank (equal to 3). We conclude that the three accelerometer biases are locally weakly observable if the GNSS provides a velocity information

- **Step 4**: **Observability analysis of the gyroscopes biases $\boldsymbol{b_\omega}$ using $\boldsymbol{h_2}$**

Using again equation (4.65), the expression of $\nabla L_{\boldsymbol{f}_0} L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2$ is given by:

$$\nabla L_{\boldsymbol{f}_0} L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 = [\boldsymbol{0}_{9\times 3} \quad \boldsymbol{0}_{9\times 3} \quad \boldsymbol{g}_{15}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times 3} \quad \boldsymbol{g}_{16}(\boldsymbol{X}) \quad \boldsymbol{g}_{17}(\boldsymbol{X}) \quad \boldsymbol{g}_{18}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times 1}] \tag{4.70}$$

where

$$\boldsymbol{g}_{16}(\boldsymbol{X}) = -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{b}_\omega}(\boldsymbol{g}_9(\boldsymbol{X})\boldsymbol{\Psi}(\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\boldsymbol{b}_\omega) \tag{4.71}$$

We prove in Appendix D that the matrix $\boldsymbol{g}_{16}(\boldsymbol{X})$ has full column rank (equal to 3) if at least two accelerometer axes are excited. Therefore, the local weak observability of the three gyroscopes biases is achieved if at least two accelerometer axes are excited. The discussion in step 2 applies also for this step.

- **Step 5**: **Observability analysis of the gyroscopes scale factors $\boldsymbol{k_\omega}$ using $\boldsymbol{h_2}$**

Using again equation (4.65), the expression of $\nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$ is given by:

$$\nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 = [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{19}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{20}(\boldsymbol{X}) \quad \boldsymbol{g}_{21}(\boldsymbol{X}) \quad \mathbf{0}_{9\times1}] \tag{4.72}$$

where

$$\boldsymbol{g}_{21}(\boldsymbol{X}) = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{k}_\omega}\left(\boldsymbol{g}_9(\boldsymbol{X})\boldsymbol{\Psi}(\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\right) \tag{4.73}$$

We prove in Appendix D, with the analysis of matrix $\boldsymbol{g}_{21}(\boldsymbol{X})$, that if the rotation about the forward axis is neglected (which is the case of a land vehicle), then the scale factor of the gyroscope measurement about this axis is not locally weakly observable. However, the other gyroscope scale factors are locally weakly observable if at least two accelerometer axes and the corresponding gyroscope axis are excited.

- **Note on the observability using only position information**

The observability of all the states has used the velocity information provided by GNSS. We have proven in Appendix D that the 3D GNSS position information is sufficient to observe all the states of the INS/GNSS/WSS/NHC system. The velocity information improves the observability of the system without adding new observables, thus making the filter convergence faster. This can be seen with the Lie derivative gradient order of the observability matrix. In fact, considering that the observability matrix rows are ordered such that the derivation order is increasing, then the sooner we proof that the parameter is observable, the higher is its observability degree. The comparison between the Lie derivative gradients associated to position and velocity measurement, whose expressions are given in Appendix , shows that we have lower derivative orders with the position and velocity information than with the position only.

- **Summary of the state observability**

Table 4.3 summarizes the different state observability as a function the sensor measurements and the vehicle motion. The (✓) is used for states that are observable using the measurement and the (X) is used in case the measurement is not useful to make the state observable.

	Global position (GNSS)	Velocity (GNSS/WSS)	Local position (VSLAM)	Local attitude (VSLAM)	Required motion
$\boldsymbol{p}_b^l$	✓	X	✓	X	None
$\boldsymbol{v}_b^l$	✓	✓	✓	X	None
$\boldsymbol{q}_{l2b}$	✓	✓	✓	X	> 2 excited acceleros
$\boldsymbol{b}_f$	✓	✓	✓	X	None
$\boldsymbol{b}_\omega$	✓	✓	✓	✓	> 2 excited acceleros
$\boldsymbol{k}_f$	✓	✓	✓	X	Corresponding accelero axis excited
$\boldsymbol{k}_\omega$	✓	✓	✓	X	> 2 excited acceleros + Corresponding gyro axis excited
k_w	X	✓	X	X	Excited WSS
k_v	X	X	✓	X	None
$\boldsymbol{\varepsilon}_{l2v}$	✓	X	X	X	None
$\delta\boldsymbol{p}_{lv}^v$	✓	X	X	X	None
$\boldsymbol{\varepsilon}_{c2b}$	X	X	X	✓	None

Table 4.3 Summary of state observability of the defined architecture

The validation of these results will be done in Chapter 6.

4.7 Conclusion

In this chapter, the architecture of the multi-sensor fusion system, based on an error-state EKF, has been proposed. The IMU is selected as the reference sensor since it is the only sensor providing continuously a complete navigation solution at the highest rate, independently of the surrounding environment. The other sensors, as well as motion constraints of a ground vehicle, are used to correct the INS solution. This closed loop solution is necessary to calibrate the bad quality IMU.

Based on this architecture, the states to be estimated by the filter have been identified for each sensor and gathered in the state vector. The mathematical models defining the Kalman propagation process and measurement update are detailed for all the sensors. An observability study has been carried out given the high number of states to be estimated by the filter. This study has confirmed the observability of all the states using the proposed architecture if some motion conditions are respected.

Chapter 5

Integrity Monitoring Challenges

The fusion of multiple sensors is an efficient way to improve the accuracy and the continuity of the navigation solution. However, qualifying the performance of a navigation system using only these two performance parameters is not sufficient for the positioning task, especially with the current development of automated and autonomous vehicles, demanding a minimum quality of sensor information to perform reliable positioning.

The reliability performance is addressed through integrity monitoring: it is a primary requirement of the positioning function to guarantee the safety of the users. This task consists primarily of providing a measure of trust of the estimated position, and if necessary of detecting "faulty" measurements to raise an alarm when the estimated navigation solution is claimed to be unreliable. Upon detection, an internal task of exclusion can be tentatively performed to exclude the fault source, thereby allowing good chances for navigation to return to safer performance without an interruption of service and enhancing the continuity.

The concept of positioning integrity is particularly mature in civil aviation community, where, for instance, the GNSS integrity requirements are well addressed and defined. For land vehicles applications, this problem is still not well addressed. Indeed, unlike some civil aviation operations (ICAO, 2006) and recently train control (Filip, 2006) and maritime navigation, no integrity requirements concerning the navigation information are publicly standardized for road vehicle navigation. Additionally, in our case, the proposed architecture in Chapter 4 includes SLAM outputs which rely on the integrity of the visual information. The application of the integrity concept to this type of environment-dependent information is a huge challenge that has not been explored yet.

This chapter consists of a first analysis on integrity monitoring challenges for airport surface and road applications using a multi-sensor navigation solution. Firstly, we recall the integrity requirement definitions as they are defined by the civil aviation community. Secondly, the sensor fault-free and failure modes are identified and an integrity risk tree is defined to identify the sensor failure modes. In particular, the challenges of defining an integrity system using visual information are addressed. Then, the integrity monitoring solutions that would be adapted to our proposed multi-sensor architecture are presented. These solutions take into account the fact that a Kalman filter is used in the proposed architecture. Finally, the integrity requirements for airport surface movement and road vehicles are given since these are the two cases of study targeted by the thesis project. The airport requirements are derived from the A-SMGCS manual (ICAO, 2004) and other research papers addressing this topic (Guilloton, 2011) (Schuster, 2011). The road requirements are a proposal that is based on a study of road vehicle traffic and the criticality of the road application. We emphasize that this chapter consists only of a discussion on integrity monitoring challenges and not a complete performance assessment of

an integrity solution for land vehicle application. The difficulties of performing this complete integrity study are highlighted.

Contents

5.1 Integrity concept 80
5.1.1 Integrity definition and requirements 80
5.1.2 Fault-free mode and failure modes 82
5.1.3 Sensor fault-free and failure mode identification 82
5.1.4 Integrity risk allocation 87
5.1.5 Failure mode characterization 89
5.2 Integrity monitoring techniques 90
5.2.1 Fault detection and exclusion 90
5.2.2 State of the art of sequential FDE 91
5.3 Case of study integrity challenges 93
5.3.1 Integrity requirements challenges 93
5.3.2 Fusion architecture integrity challenges 94
5.4 Conclusion 97

5.1 Integrity concept

5.1.1 Integrity definition and requirements

As defined by civil aviation and also the DoD for GPS, integrity is the measure of trust that can be placed in the correctness of the information supplied to the total system. Integrity includes the ability of a system to provide timely valid warnings to the user (alerts) when the system must not be used (Martineau, 2008).

In civil aviation applications and for GNSS Signal in Space, defined as the aggregate of GNSS signals expressing the required navigation performance of the aircraft for each phase of flight in terms of accuracy, integrity, availability and continuity of service, three parameters quantify the GNSS signal integrity requirements for a specific operation (e.g. en-route operation in civil aviation), assuming a fault-free GNSS receiver (ICAO, 2006):

- the Integrity Risk (P_{IR}): it is the probability of using a Hazardous Misleading Information (HMI) without detection within the time to alert. A HMI is defined when a positioning failure occurs while the system is declared available. Further details about the positioning failure and the system availability are given later in this chapter

- the Alert Limit (AL): it represents the largest position error (PE) allowable for a safe operation. In civil aviation, depending on the operation, two alert limits are defined: the Horizontal Alert Limit (HAL) and the Vertical Alert Limit (VAL). In the case of terrestrial navigation, we can usually avoid considering requirements on the quality of vertical position information. Therefore, typically only the HAL must be considered, although other types of error bounds could be considered such as the lateral AL or the longitudinal AL. The HAL is defined as the radius of a circle in the horizontal plane (East-North plane of the local frame (l)), with its center being at the true position, that describes the region that is required to contain the indicated horizontal position with the required probability for a particular navigation mode

- the time to alert (TTA): it is the maximum allowable elapsed time from the onset of a positioning failure until the equipment annunciates the alert.

A positioning failure occurs whenever the positioning error exceeds the AL (ICAO, 2006). Figure 5-1 illustrates a horizontal positioning failure: the estimated horizontal position is outside the HAL circle. If the positioning failure is not annunciated within the TTA, then a loss of integrity event occurs.

For the remaining of this chapter, because of the considered applications, we only focus on integrity monitoring in the horizontal plane.

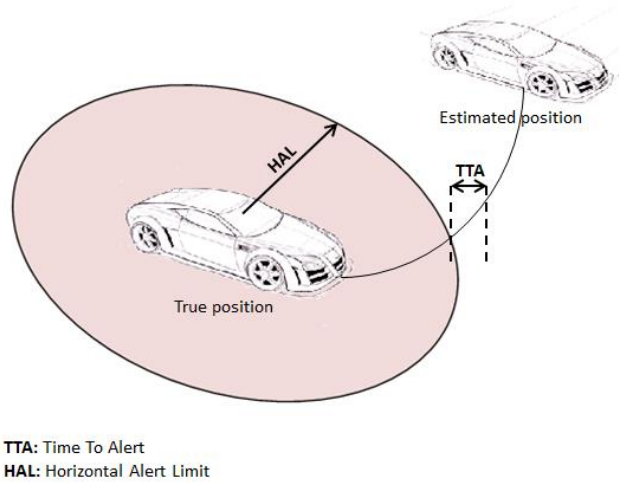


Figure 5-1 Loss of integrity

The probability of a loss of integrity at time k of an operation is therefore given by:

$$\mathrm{P}_{IR}(k) = \mathrm{P}\big((\mathrm{HPE}(k) > \mathrm{HAL}) \,\&(\text{no detection within TTA})\big) \tag{5.1}$$

where HPE represents the Horizontal Position Error.

The TTA is the delay allowed for detecting the anomaly after the onset of a positioning failure and warning the user. This last delay of warning transmission in the system is not necessarily well known or easily quantified. Thus, we are going to assume that a loss of integrity happens as soon as a positioning failure occurs. This means

$$P_{IR}(k) = P(\text{HPE}(k) > \text{HAL}) = P\big(HMI(k)\big) \tag{5.2}$$

In order to determine $P\big(HMI(k)\big)$, an analysis of the causes of HMI is necessary. This will be done in the next section.

5.1.2 Fault-free mode and failure modes

An HMI at time epoch k may result from sensor fault-free mode or a failure mode.

The fault-free mode covers nominal errors as well as the causes of HMI that are due to rare normal errors (RTCA, 2009). In general, nominal measurement errors are reliably modeled using large amounts of experimental data. In reality, this is not always true because, in some situations, errors such as GNSS multipath are too complex to be described or modelled. One technique which is used in the civil aviation community, to simplify the model of the nominal distribution, is the Gaussian overbounding technique. This technique refers to the process of replacing the actual error distribution by a simplified conservative model, with the objective of having enough margins to take into account the risk of non-modelled errors (Rife, 2009). There is no requirement that the errors included in the fault-free mode should be detected. They are only taken into account in the calculation of an upper statistical bound of the horizontal position estimation error under the fault-free assumption. This upper bound is called Horizontal Protection Level (HPL) and is addressed in Section 5.2.1.

A failure mode occurs when one (or multiple) sensor measurement(s) is (are) affected by a sufficiently large error that may potentially lead to an HMI. Contrary to nominal errors that correspond to the usual errors, failure modes describe events that are difficult to observe because of their low probability of occurrence and that represent threats to the navigation solution. These events are called “sensor threats” or “sensor failure modes”.

In Section 5.1.3, we will try to list the fault-free and failure mode error sources for the GNSS, the VSLAM and the INS. As for the wheel odometry velocity, this measurement is the linear velocity measurement, computed using the different wheel velocities and eventually the steering wheel angle information as explained in Section 2.3. We will assume that this information, as such, is reliable, and the integrity of its measurements will not be investigated. This assumption is justified by the information redundancy provided by each WSS on each wheel that allows to check the consistency of each WSS velocity measurement with the three other velocity measurements and, probably exclude the inconsistent measurement.

5.1.3 Sensor fault-free and failure mode identification

Sensor fault-free mode describes the nominal sensor measurement errors as well as the rare normal errors. The nominal measurement errors are described in Chapter 2 for GNSS and INS. The failure modes of these two sensors will be detailed thereafter. As for SLAM, we will list the most important errors impacting the navigation solution and try to classify them into fault-free mode errors and failure mode errors. Note that this list may not be exhaustive as it is mostly based on our experience with the SLAM.

5.1.3.1 GNSS failure sources

The GNSS system used in the proposed hybridization solution is a GPS L1 C/A receiver without any augmentation. Thus, the failure modes detailed in this section only refer to GPS constellation signals.

GPS failure modes originate from several sources that are well established in the literature (Martineau, 2008). These sources consist of satellite failure, constellation failure, atmospheric propagation anomalies, local effects such as interference multipath and NLOS[8], etc. An exhaustive list is given in (Bhatti, 2007) and (Montloin, 2014).

5.1.3.1.1 The satellite failure

It is called "Major Service Failure" (MSF) and occurs when a healthy GPS satellite ranging signal error (excluding atmospheric and receiver error) exceeds the range error limit (GPS-SPS, 2001). The range error limit is 4.42 times the URA value[9]. The probability of occurrence of a MSF is given by (GPS-SPS, 2008): $P_{MSF} = 10^{-5}/\text{sat/h}$

After the occurrence of such an event, the GNSS shall detect and alert the user within 6 hours. Realistic Maximum observed alert delays are around 1 hour. Thus the probability that a satellite is faulty at a given instant is equal to the probability that the failure occurs during the previous hour (Brocard, 2016): $P_{MSF(1hour)} = 10^{-5}/\text{sat}$

The probability that a position is calculated with N_{sat} containing n MSF is given by:

$$P_{occ\ of\ n\ MSF} = \binom{N_{sat}}{n} \left(P_{MSF(1hour)}\right)^{n} \left(1 - P_{MSF(1hour)}\right)^{N_{sat}-n} \quad (5.3)$$

5.1.3.1.2 Constellation failure

This is the failure that may affect more than one satellite within the same constellation at a given time and for the same reason. For GPS, the probability of occurrence of this failure is currently assumed to be zero (there is no evidence of such event in the past) (Brocard, 2016).

5.1.3.1.3 Atmospheric propagation anomalies

During its propagation, the GNSS signal may encounter large ionospheric or tropospheric delays, or ionospheric scintillation due to solar and meteorological storms. This can result in larger than usual errors in GNSS and manifest as a channel failure.

5.1.3.1.4 Local effects

Local effects include the signal degradation resulting from the environment surrounding the GNSS antenna. These local effects are divided in two categories: interference and NLOS.

- **Interference**

Interference is defined as any undesired signal that interferes with the reception of GNSS signals, and results in degrading the GNSS signals or even completely loosing the track of this signals. It may be intentional or unintentional. Intentional interference includes mainly

[8] We remind that the NLOS is a reflected GNSS signal received without receiving the direct signal
[9] We remind that the URA is the standard deviation value of the satellite clock and ephemeris errors

jamming[10] and spoofing[11]. Unintentional interference is mainly due to equipment that emits Radio Frequency (RF) signals in the GNSS frequency bands such as Distance Measurement Equipment (DME) in the airports that may interfere with GNSS L5 signals and any other equipment emitting RF signals and generating harmonics in the GNSS frequency bands. Further details about interference are given in (Montloin, 2014).

- **NLOS (Non-Line of Sight signals)**

If the satellite is masked, the receiver may track a NLOS, sometimes resulting in a significant bias in the GNSS processed position domain. As explained in (Brocard, 2016), it is not sufficient to apply a C/N_0 mask to exclude all the possible NLOSs. Some NLOSs have a C/N_0 relatively high and may degrade the GNSS accuracy. The remaining NLOS error is not modeled by the nominal GNSS error presented in 2.1.2 and should be taken into account in GNSS signal integrity monitoring, especially in urban environments. Two ways are proposed in (Brocard, 2016) to consider NLOS. One consists of including the NLOS in the nominal error model by inflating it sufficiently; the second considers the NLOS as a failure mode and relies on additional algorithms to protect the system against it.

5.1.3.1.5 User equipment hardware failure

The user equipment hardware failure consists of failures that may occur in the antenna, receiver, or reference oscillator, etc., or software failures. These errors can sometimes produce erroneous outputs on all channels, so they require detection.

5.1.3.2 VSLAM errors

This section lists the VSLAM errors that are likely to degrade the its performance. There errors can be classified into two categories: feature extraction errors and feature matching errors.

5.1.3.2.1 Feature extraction errors

Feature extraction refers to the process of detecting a feature in the image plane and estimating its pixel coordinates. Feature extraction errors may be due to hardware, software or environmental-dependent errors. Hardware errors occur because of the sensor degradation or malfunctioning (camera noise, sampling, quantization, etc.). Software errors can be due to the sensor modeling including the applicability of the projection model, the intrinsic parameter estimation and the distortion corrections (calibration parameters). Other errors on feature extraction can occur because of the environment surrounding the camera. These errors are mainly due to bad visibility conditions (rain, fog, insufficient light, etc.)

5.1.3.2.2 Feature matching errors

Feature matching errors are one of the most serious SLAM problems. We remind that feature matching consists of associating extracted features with corresponding previously existing ones. This process is sensitive to several issues such as the lack of feature repeatability, the lightening abrupt change, the feature scale change, the point of view change, blurring,

[10] Jamming is an intentional attack consisting of broadcasting radio-frequency power that interferes with a receiver's ability to track the GNSS genuine signals, resulting in denial of service

[11] Spoofing is an intentional attack consisting of broadcasting competing signals that make the position estimated by the user receiver incorrect.

occlusion, etc. This leads to establishing erroneous feature matches (outliers), resulting if not excluded in errors degrading the SLAM accuracy estimation. On the other hand, when the outlier rejection process (RANSAC) decides to incorrectly reject a match, fewer measurements are used for estimation, at the expense of estimation accuracy. These issues are especially due to the navigation visual environment and conditions. Indeed, if the visual environment is highly dynamic with a lot of moving objects and people, then the performance of this process is more likely to be degraded due to the lack of good matches (inliers) between static objects with respect to the matches between dynamic features (outliers). In addition, the dynamics of the vehicle, the frame rate and the power of the computer used for feature matching are determining factors of the robustness of this process. If the vehicle has high dynamics and the frame rate is not sufficiently high to handle the fast change of light, scale, point of view, etc., then the matching process is likely to be degraded. However the frame rate cannot be increased without taking into account the computer processor power. Therefore, feature matching robustness (and consequently the probability of failure occurrence) depends thoroughly on the power of the computer used to perform this process.

As discussed in (Cadena, 2016), feature matching is even more challenging when loop closure process is performed because of light, scale and point of view significant changes. This problem is partially solved with the use of bag of words models explained in Section 3.2.2, but this approach is not capable of handling severe illumination variations. This issue can result in establishing wrong loop closures and consequently degrade severely the SLAM estimation process. In addition, in poorly textured environments (runways and taxiways in airports, highways, etc.), the environment tends to be repeated, making consequently the SLAM algorithm establish wrong matches with features seen a time ago and decide a loop closure while this is not the case. This problem has been noticed on a data collection performed in airport environments near the runway.

5.1.3.2.3 Classification of SLAM faults

To be able to address SLAM issues from an integrity monitoring point of view, it is essential to classify the previously mentioned error sources into fault-free mode and failure mode. We remind that that the fault-free mode includes the large nominal errors and the rare normal events while the failure mode includes the errors that cannot be considered nominal. A classification of these errors is proposed in Table 5.1, assuming that the computer processing power and the frame rate are sufficiently high to consider that they are not a source of SLAM error.

	SLAM errors	Error sources
Fault-free mode	Feature extraction error	- Camera noise - Quantization errors - Sampling error - Calibration errors
	Possible existence of outliers in feature matching but with an acceptable inlier-outlier ratio)	- Existence of some dynamic objects in the environment - Light, feature scale, feature point of view changes
Failure modes	Feature extraction error	Bad visibility conditions
	Existence of outliers in feature matching with an unacceptable inlier-outlier ratio)	- Very dynamic environment (High number of dynamic features) - Badly textured environment (Small number of features) - Camera occlusion and blurring - Illumination variation - Bad initialization - Bad motion model (e.g. if the constant velocity model does not describe the actual vehicle motion)

Table 5.1 Classification of VSLAM errors into fault-free mode and failure modes

Note that the environmental conditions mentioned in Table 5.1 as failure modes (very dynamic environment, badly textured environment, camera occlusion and blurring, illumination variation) can result not only in integrity issues but also in continuity issues consisting of the loss of map tracking and consequently the interruption of the navigation output supply by VSLAM.

5.1.3.3 Inertial sensor errors

Inertial sensor errors are mainly due to hardware and/or mechanical/electrical errors and can manifest as no outputs at all, null readings, repeated readings, or simply much larger errors than specified (Groves, 2008). The IMU errors may also be due to the misalignments either between the gyroscopes and the corresponding accelerometers axis or between the body and the IMU platform frames in the case of strapdown mechanization. In addition, if the IMU axes are not perfectly orthogonal, then the IMU measurements will be affected by errors whose severity depends on non-orthogonality coefficients of the IMU axes. Schuler oscillation is also a possible source of inertial error. It is the resulting error of combining initialization errors with the navigation mechanization equations. For example, when an error is introduced in the navigation equations due to an error in initial conditions or non-perfect measurements, this error can be amplified because of the closed loop of the inertial mechanization (the previous

navigation solution is used to compute the current navigation solution). The IMU fault-free mode includes the errors that are intrinsic to the IMU which are described by the measurement model in equations (2.27). The failure mode corresponds to less common errors such as the misalignment between the body frame and the IMU frame, errors related to the gravity model or to the Schuler oscillation.

As highlighted in (RTCA, 2009), the IMU failures are traditionally detected and sometimes even excluded using sensor redundancy which means using several IMU platforms with the same characteristics and running in parallel. Fortunately, the development of the MEMS inertial sensors in the last decade makes this solution possible without exorbitant cost increase. Based on this argument, we are going to assume that there is no integrity risk coming from the IMU and that the only integrity risks come either from GPS or SLAM.

5.1.4 Integrity risk allocation

The integrity risk allocation consists of distributing the integrity risk among all the possible causes of an HMI for all the considered sensors. Therefore, the integrity risk can be expressed as follows (the time epoch k has been removed to simplify the notations):

$$\mathrm{P}_{IR} = \mathrm{P}_{IR}^{GPS} + \mathrm{P}_{IR}^{SLAM} + \mathrm{P}_{IR}^{IMU} + \mathrm{P}_{IR}^{WSS} \tag{5.4}$$

where P_{IR}^{sensor} consists of the integrity risk induced by the failure of each sensor (or the failure of the navigation module of each sensor including the hardware and software risks).

for each sensor, both fault-free mode and failure modes may lead to an HMI. Consequently, the allowed integrity risk for each sensor can be sub-allocated among the fault-free mode and the different failure modes using (Montloin, 2014):

$$\mathrm{P}_{IR}^{sensor} = \mathrm{P}_{IR,ff}^{sensor} + \sum_{i} \mathrm{P}_{IR,fm_i}^{sensor} \tag{5.5}$$

where

- $\mathrm{P}_{IR,ff}^{sensor}$ is the integrity risk allocated to the sensor fault-free mode
- $\mathrm{P}_{IR,fm_i}^{sensor}$ is the integrity risk allocated to the sensor failure mode fm_i

$$\mathrm{P}_{IR}^{sensor} = \mathrm{P}_{HMI|fm_i}^{sensor} \ \mathrm{P}_{fm_i}^{sensor} \tag{5.6}$$

- $\mathrm{P}_{HMI|fm_i}^{sensor}$ is the HMI probability given that the sensor failure mode fm_i is present and not detected (this probability is equivalent to the probability of missed detection (P_{MD}) of a positioning failure)
- P_{fm_i} is the probability of occurrence of the sensor failure mode fm_i

As highlighted previously, it is assumed that there is no integrity risk coming from the WSS and the IMU thanks to the use of redundancy. Therefore, we can write:

$$\mathrm{P}_{IR} = \mathrm{P}_{IR}^{GPS} + \mathrm{P}_{IR}^{SLAM} \tag{5.7}$$

In addition,

Therefore we can write from equation (5.5) and equation (5.7):

$$P_{IR} = P_{IR,ff}^{GPS} + P_{IR,ff}^{SLAM} + \sum_i P_{IR,fm_i}^{GPS} + \sum_i P_{IR,fm_i}^{SLAM} \quad (5.8)$$

This equation leads to the definition of the following integrity risk tree:

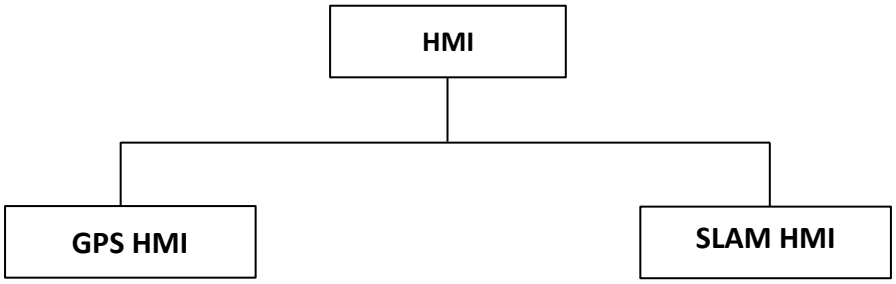


Figure 5-2 Integrity risk tree

Where the GPS and the SLAM fault trees are the following:

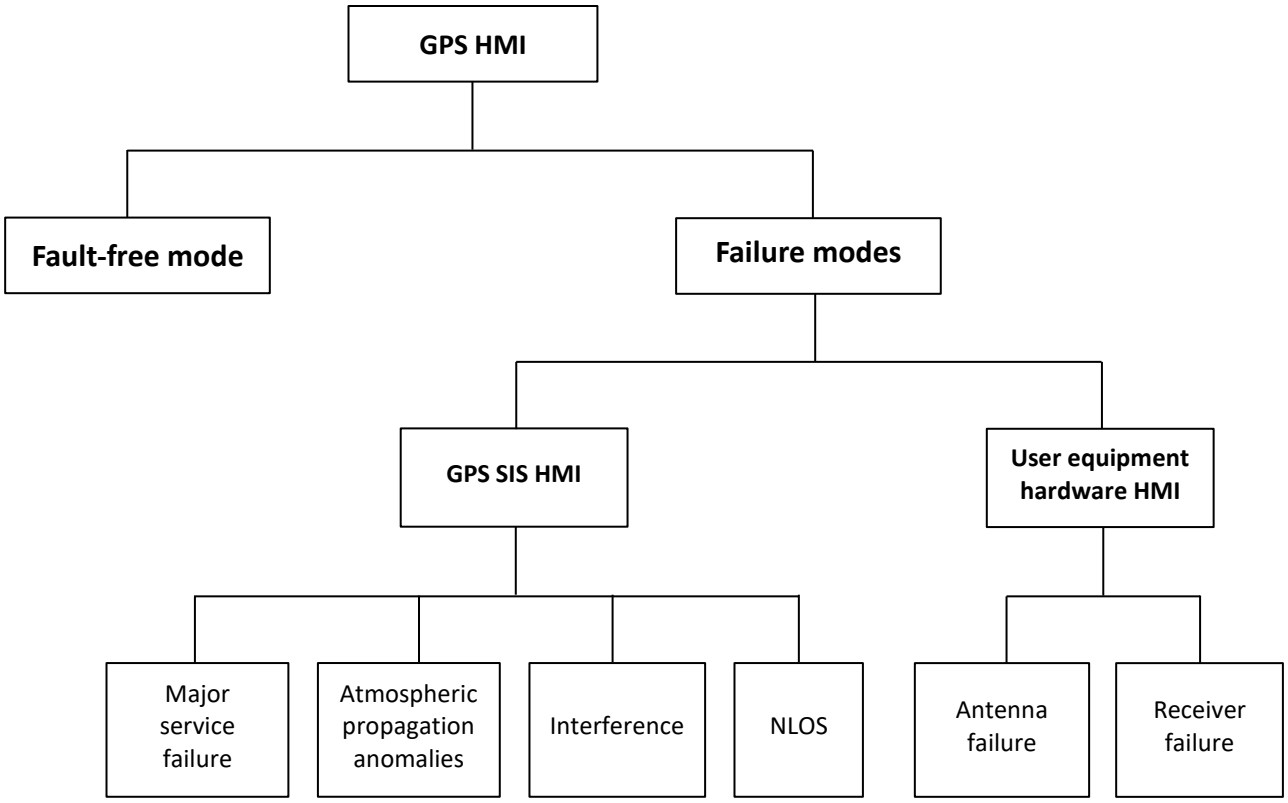


Figure 5-3 GPS fault tree

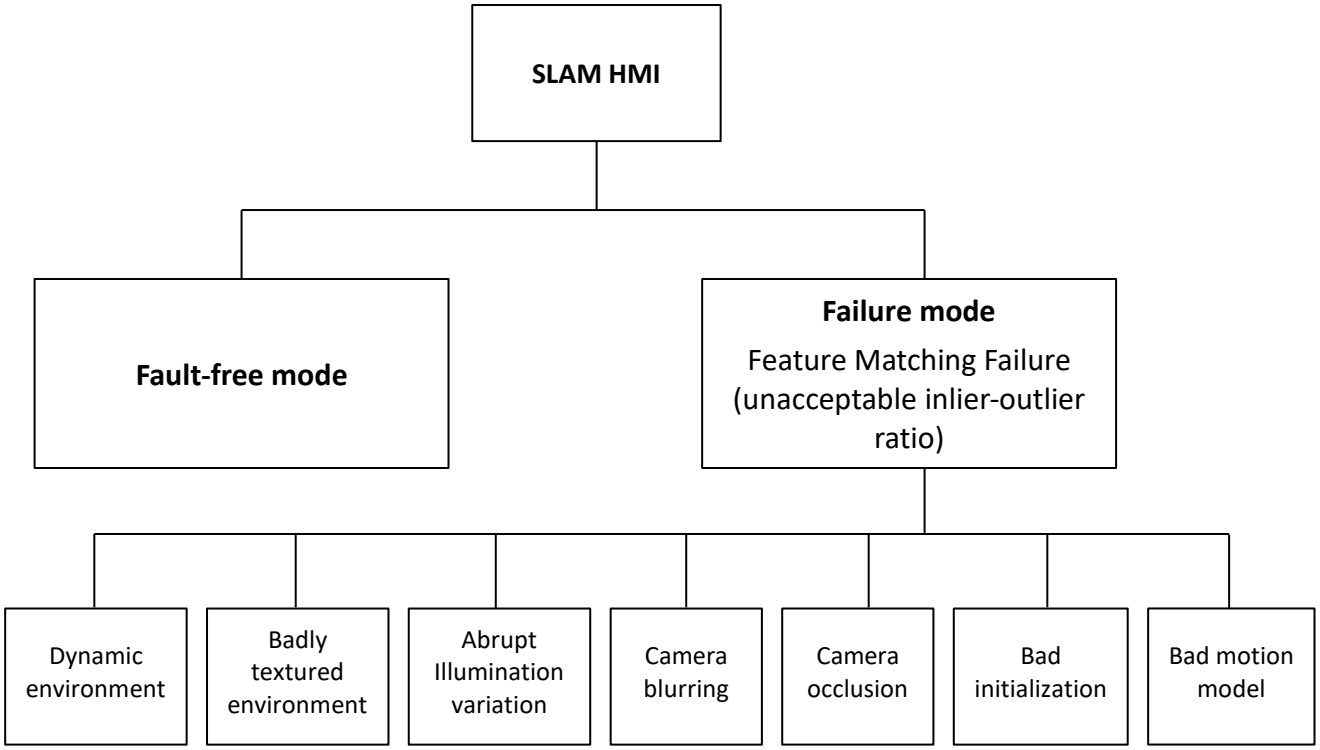


Figure 5-4 VSLAM fault tree

Note that in the scope of this thesis, we do not give numerical allocations to the integrity risks associated to the fault-free and the failure modes. This is mainly because the characterization of the SLAM errors is a challenging task that should be done in future works because it needs a sufficient experience on the SLAM failure mode characterization, which is not the case at present.

5.1.5 Failure mode characterization

Sensor failure modes are generally characterized by their probability of occurrence $P_{fm_i}^{sensor}$, the correlation between multiple failures (simple failure, multiple failures) and their signatures (amplitude, time duration, shape). Their characterization is an important step that must be done prior to the definition of any integrity monitoring algorithm. However, for environment-dependent errors such as GNSS local effects and SLAM data association in harsh environments, the characterization of such errors is a major issue in addressing the integrity problem. For GNSS, some researches have addressed failure characterization issues in urban environment such as multipath and NLOS characterization using environment modeling (BinAhmad, 2015). However, to our knowledge, there is still no work on the characterization of the SLAM error depending on the environment in which the camera moves. In this case, the SLAM failure characterization is a challenging task that should be investigated. A possible way to handle this task is to consider the worst case, consisting of assuming a probability of occurrence of such events equal to one. This assumption can be an appropriate way to handle this issue. However, this is a very conservative assumption.

In fact, as explained in (Montloin, 2014), the probability of occurrence $P_{fm_i}^{sensor}$ of a failure mode fm_i drives the requirements on the integrity monitoring system. Indeed, we can see from

equation (5.6) that the lower the probability of occurrence is, the less demanding the requirement is on the probability of missed detection P_{MD} related to the failure mode. In the worst case where the probability of occurrence of a failure mode is maximum (equals 1), the probability of missed detection should be very low to satisfy the integrity risk. Therefore, considering that a SLAM failure mode has a probability of occurrence of 1 leads to a very low probability of required positioning failure missed detection. The drawback of having a low P_{MD} is explained in Section 5.2.1, after presenting the integrity monitoring function.

5.2 Integrity monitoring function

Integrity monitoring must be able to detect a failure, potentially exclude or mitigate the impact of the erroneous measurement from the navigation solution and provide a measure of the integrity provided in the form of a protection level. In civil aviation navigation using GNSS signals, integrity monitoring may be carried out at the system level (external) using external integrity system such as Ground Based Augmentation Systems (GBAS) or Satellite Based Augmentation System (SBAS) or at the sensor level (internal) using Aircraft Based Augmentation System (ABAS). More details about these augmentation systems can be found in (Milner, 2009). In our case of study, only the sensor level integrity monitoring ABAS is considered since the navigation architecture proposed in Chapter 4 does not involve any external aiding to the used sensors. This integrity system autonomously performs fault detection and potentially exclusion without the need of any external augmentation.

5.2.1 Fault detection and exclusion

Fault Detection and Exclusion (FDE) is a processing scheme that autonomously provides integrity monitoring for the position solution, using redundant measurements. The FDE consists of two distinct parts (Martineau, 2008):

- The fault detection part detects the presence of an unacceptably large position error. If a positioning failure occurs and is not detected, then this event is called *missed detection*. If an alarm is raised without a positioning failure occurs, then this event is called *false alarm*.
- Upon the detection, the fault exclusion may follow to try to exclude the source of the unacceptably large position error, thereby increasing chances for navigation to return to normal performance without an interruption of service. This process is characterized by failed exclusion and by wrong exclusion events.

The FDE function involves three basic parameters: test statistic, decision threshold and Horizontal Protection Level (HPL).

The fault detection is performed by comparing the test statistic to a threshold. If the test statistic exceeds the threshold, then a fault is decided otherwise no fault detection occurs. The selection of the test statistic depends on the integrity monitoring method[12]. The decision threshold is chosen on the basis of statistical characteristics of the test statistic so that a false alarm occurs

[12] Further details will be given later

no more than a specified rate called *false alarm rate* (P_{FA}). For example, if the test statistic has a distribution with a specific Cumulative Distribution Function (CDF), then the threshold is computed based on the CDF denote Q_{dist} and the P_{FA} using:

$$T_h = Q_{dist}^{-1}(1 - P_{FA}) \tag{5.9}$$

The HPL is a statistical upper bound of the estimated HPE satisfying the required integrity risk, thus delimiting the region, around the true position, assured to contain the estimated horizontal position with a specific probability. The HPL is not affected by the actual measurements; it is computed only from the P_{MD}, the P_{FA}, the expected measurement error characteristics and the factors on which depends the precision (e.g. Dilution Of Precision for GPS, feature number and spatial distribution for SLAM). The HPL is an important parameter that determines the availability of the integrity function. That is, if HPL is less than the HAL for a given operation, the integrity function is available, and if it is higher than the HAL then the integrity function is declared unavailable. As explained in (Lee, 1999), as much as it is important for an integrity method to detect that HPE exceeds HPL given the predefined integrity risk, it is also important that HPL be small so that high availability of the integrity function may be provided. Consequently, to keep a small value of HPL, it is important firstly to define an integrity monitor (test statistic and threshold) that detects and excludes the causes of HMI. Secondly, the probability of failure occurrence should be identified instead of considering it equal to 1 as suggested in Section 5.1.5. This increases the value of the required P_{MD} based on equation (5.6), making consequently the HPL value small.

The classical algorithm to perform GNSS fault detection and possibly fault exclusion is the Receiver Autonomous Integrity Monitoring (RAIM) (Parkinson, 1988). It operates by using redundancy in the measurements observed to perform a statistical test based consistency check for the presence of failures. RAIM algorithms may be classified into two groups depending on the algorithm used to estimate the navigation solution: the snapshot RAIM involving only measurements received at the current time epoch and sequential RAIM using both current and historical measurements. The most widely used integrity monitoring algorithm for GNSS signals is the snapshot RAIM (Parkinson, 1988) (Walter, 1995) (Martineau, 2008) that exploits redundant measurements at one time of interest. However, this algorithm is insufficient for our case of study where a Kalman filter is used to fuse sensor measurements. This type of implementation does not match with snapshot RAIM and needs a sequential detection and exclusion algorithm.

5.2.2 State of the art of sequential FDE

Sequential detection approaches have been investigated in the literature. The most popular works are the Autonomous Integrity Monitoring Extrapolation (AIME) method (Diesel, 1996) and the Maximum Solution Separation (MSS) (Brenner, 1995), for GPS/INS hybridization considering that the INS is fault-free.

AIME method uses the Kalman filter innovations[13] rather than the instantaneous least square residuals of the snapshot RAIM, to compute the test statistics to detect GPS range failures. Three test statistics are computed in AIME. The first test is based on the innovation average over the last 2.5 min duration and the others are obtained by averaging the innovations over 10 min and 30 min.

MSS is based on a bank of n sub-filters running in parallel and excluding, each one, a GPS measurement, where n is the number of satellites in view. The sub-filters are then compared to the full-filter solution by computing the difference. This difference defines the test statistic for each sub-filter and is compared to a threshold to detect failures.

A comparison between the AIME and MSS is performed in (Lee, 1999). It reveals that both methods have advantages and disadvantages. While the MSS guarantees satisfactory detection performance based on theory, the HPL values tend to be relatively large so that it cannot achieve good integrity function availability. In addition, MSS runs as many sub-filters as the number of measurements and has consequently a high computational burden. On the other hand, while AIME can achieve significantly higher availability, there is no good way to confirm the detection performance based on theory; therefore, very extensive simulation must be used instead. A common drawback of these two methods is the assumption of a single failure.

Other sequential FDE algorithms based on detecting abrupt changes in random variable distribution are proposed in the literature. These algorithms are mainly based on Kalman innovations (Giremus, 2007) (Sukkarieh, 1999). One of the most known algorithms detecting abrupt changes at the innovation level is the Generalized Likelihood Ratio (GLR) algorithm (Willsky, 1976) (Giremus, 2007). The detection of abrupt changes affecting the components of the state vector is performed by sequentially applying a likelihood ratio hypothesis testing. The advantages of this method are on one hand the possibility to cope with multiple simultaneous failures and on the other hand the robustness to disturbances of small magnitude. The drawback of this approach is that no fault-free solution is maintained. Therefore, whenever an anomaly is detected, a compensation step is necessary to remove the induced errors on the Kalman filter estimates.

All these state of the art methods are based on a monitor (detection test and HPL) used to detect a failure and potentially exclude it, as explained in Section 5.2.1. For the fault-free mode, the HPL depends on the required integrity risk and the error distribution. For the failure mode, the HPL depends on the allowed missed detection rate, which in turn depends on the failure probability of occurrence and the required integrity risk. Since some of the VSLAM and GNSS failure modes are environment-dependent, their probability of occurrence is hard to characterize. Therefore, specific monitors with guaranteed performance cannot be implemented at this stage. However, an overall FDE monitor for Kalman filter can still be implemented. This monitor is the *innovation test* (Sukkarieh, 1999), which is a way of detecting and excluding a measurement by deciding whether it is consistent with the other measurements or not. The normalized innovation (i.e. the innovation divided by its corresponding uncertainty) is

[13] We remind that the Kalman filter innovation is the difference between the actual measurement and the predicted one.

calculated and compared to a threshold. If the normalized innovation is higher than the threshold, than the measurement is excluded, otherwise the measurement is kept. Considering that the Kalman innovation should follow a zero-mean Gaussian distribution, if the threshold is set to 2, then 95% of genuine measurements are passed by the innovation test, and if it set to 3, then 99.73% of genuine measurements are passed by the innovation test. Besides its simplicity, this method is applicable to all the types of measurements.

5.3 Case study of integrity challenges

After reviewing the integrity principle and algorithms for Kalman-based architectures, we are going to recall the most important integrity challenges of our thesis case of study. These challenges can be classified into two categories: the challenges related to the integrity requirements of the targeted applications and the challenges related to the fusion architecture defined in Chapter 4.

5.3.1 Integrity requirements challenges

This thesis targets two different types of land applications. Airport surface moving vehicles and land vehicles moving in urban environments. Unlike aircraft operations which have specific Required Navigation Performance (RNP) (ICAO, 2006) and recently train control (Filip, 2006), until today, there is really no international standard of user performance integrity requirements for these two applications. However, researches are ongoing to define and standardize these requirements for airport movement. For example, the ICAO has developed the Advanced Surface Movement Guidance and Control System (A-SMGCS) concept (ICAO, 2004) to cope with traffic increase on airport surfaces. In addition, this issue has been addressed in some research papers such as (Guilloton, 2011) and (Schuster, 2011). In these researches, the proposed requirements are dependent on the operation (taxiing, approach, etc.), the visibility conditions (good, low, etc.) and the main intended navigation function (Surveillance[14], Guidance[15], Steering[16], etc.). When the hazard risk increases, the requirements become more stringent. For example, when the operation is conducted with a high vehicle speed in degraded visibility conditions and without possible human intervention, the integrity requirements must be very stringent to avoid causing harm to humans or destruction/damage to materials. In our case, we are interested in steering requirements since the automated/autonomous vehicle applications are intended. Table 5.2 provides the navigation integrity requirements in terms of integrity risk and HAL defined in the A-SMGCS and (Guilloton, 2011) for low-speed ground airport vehicles moving near the airport gate:

[14] Surveillance is defined in (ICAO, 2004) as the function that concentrates and displays the position and the identification of all aircraft and ground vehicles on the airport surface.
[15] Guidance is defined in (ICAO, 2004) as the function that relates to facilities, information and advice necessary to enable pilots of aircraft or drivers of ground vehicles to find their way on the airport and keep aircraft or vehicles on the surfaces and areas intended for their use.
[16] Steering can be defined as the function that enables automatic control of the aircraft or the ground vehicles on the airport surface without human aid.

Operation	Integrity risk	HAL
Gate	$10^{-7}/\text{op}$	$1.2\ m$

Table 5.2 Integrity risk and HAL for steering in the airport Gate

As for land vehicles, no standard requirements are defined presently. This makes the integrity study difficult to perform currently. We are going to derive an example of the HAL value using a simple reasoning. Figure 5-5 illustrates a typical traffic scheme where the width of one road way equals $3.5\ m$. We imagine a scenario where a $1.6\ m$ width small urban car running in the middle of one road way meets a $2.5\ m$ width van also running in the middle of its corresponding road way. Then the maximum allowable cross-track error (CTE) to avoid a collision between the two vehicles equals:

$$\text{CTE}_{max} = 3.5 - \frac{1.6 + 2.5}{2} = 1.45\ m \tag{5.10}$$

We define CTE_{max} as the HAL.

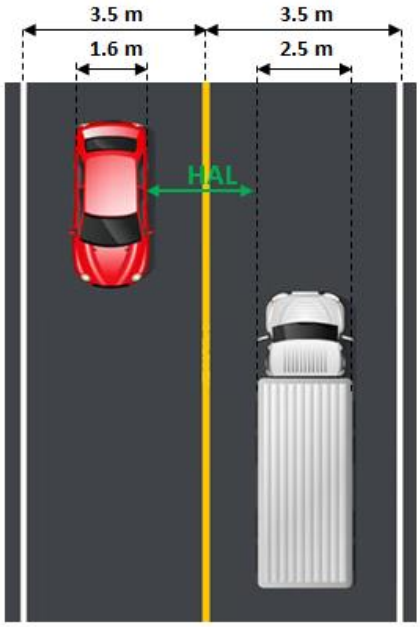


Figure 5-5 HAL computation from a typical traffic example

5.3.2 Fusion architecture integrity challenges

Ideally, in order to take into account the integrity issue in the fusion architecture proposed in Chapter 4, integrity blocks should be added to the architecture as illustrated in Figure 5-6.

Each sensor should have its own integrity block at the sensor level either using sensor redundancy (IMU and WSS) or using measurement redundancy (GNSS and SLAM). A final integrity check based on the redundancy of measurements of the same nature provided by the different sensors should be performed at the hybridization stage.

The most challenging parts are the SLAM and the filter integrity blocks.

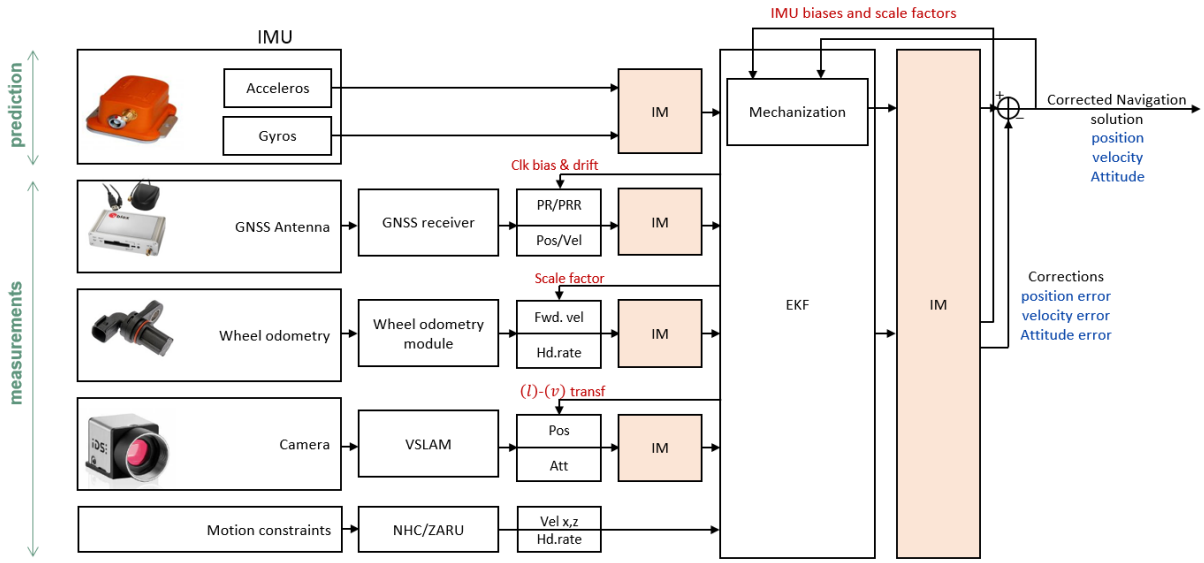


Figure 5-6 multi-sensor architecture integrity monitoring

Although SLAM performs internal outlier detection and exclusion through RANSAC, there are no sufficient elements to elaborate a complete integrity study of this algorithm for several reasons.

First, given the fact that SLAM includes very complex processes to compute the camera pose (navigation solution) from the features detected in the camera images (initial measurements), it is not easy to establish a simple relationship between the SLAM navigation solution and its measurements. This error projection from the measurement domain to the range domain is a necessary step to express the integrity risk in the range domain through the computation of the HPL in case of failure.

Second, the SLAM failure modes and their characteristics in terms of probability of occurrence, number of simultaneous failures, amplitude, time duration and shape need an extensive study and is not performed yet. For the probability of occurrence, if we consider the worst case where it is equal to one, the probability of missed detection corresponds to the lowest possible value the integrity risk value). This makes the HPL value be maximal, reducing therefore the availability of the integrity function as explained in Section 5.2.1. As an example, if we assume that the SLAM provides Gaussian outputs (optimistic assumption) in the fault-free mode (which is the mode supposed to have the lowest HPL), the SLAM HPL, denoted HPL_{ff}^{slam}, can be expressed by:

$$\mathrm{HPL}_{ff}^{slam} = k_{ff}^{slam}\,\sigma_h^{slam} \tag{5.11}$$

where

- σ_h^{slam} is the square root of the maximum eigenvalue of the (2×2) SLAM horizontal position covariance matrix. This matrix must be first transformed from the vision frame to the local frame to be able to talk about horizontal error in meter units.

- k_{ff}^{slam} is a multiplicative coefficient computed from the integrity risk allocated to the fault-free mode $P_{IR,ff}^{slam}$ using:

$$k_{ff}^{slam} = -Q_{norm}^{-1}\left(P_{IR,ff}^{slam}\right) \tag{5.12}$$

where Q_{norm} is the normal CDF.

Table 5.3 expresses different values of k_{ff}^{slam} as a function of different possible values of P_{IR}^{slam} if we assume that the integrity risk is divided equally between the fault-free mode and the failure mode defined by Figure 5-4, i.e.

$$P_{IR,ff}^{slam} = P_{IR,fm}^{slam} = P_{IR}^{slam}/2 \tag{5.13}$$

After k_{ff}^{slam} derivation, the values of $\left(\sigma_h^{slam}\right)_{max}$, defining the maximum allowable value of σ_h^{slam} to meet the availability requirement based on the HAL example given for land vehicle application in equation (5.10), are given in Table 5.3 .

$\boldsymbol{P_{IR}^{slam}}$	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}
$\boldsymbol{k_{ff}^{slam}}$	5.73	5.33	4.89	4.41	3.89
$\boldsymbol{\left(\sigma_h^{slam}\right)_{max}}$	25 cm	27 cm	30 cm	33 cm	37 cm

Table 5.3 Computation of the maximum allowable SLAM uncertainty

Note that, according to Table 5.2 , the HAL for vehicles moving near the airport gate equals 1.2 m and is in the same order of the land vehicle HAL, giving therefore the same order of the required accuracy $\left(\sigma_h^{slam}\right)_{max}$ as land vehicle applications.

Therefore, the maximum allowable value of the SLAM uncertainty to guarantee the availability of the integrity function is about 30 cm. According to the SLAM performance analysis that will be presented in Chapter 6, showing that the SLAM has meter accuracy, the required sub-meter accuracy is not achieved yet, making the integrity function almost always unavailable for the SLAM process.

As for the filter integrity, the same reasoning for the position state estimated by the developed filter applies as for SLAM. According to the filter performance analysis, that will be presented in Chapter 6, the state after fusion has meter accuracy and does not meet the HAL requirement defined by the example in 5.3.1. Consequently, the availability of the integrity function is not guaranteed yet and an integrity algorithm cannot be implemented on the current developed architecture, unless the accuracy is first improved to achieve about 30 cm.

Only the innovation test described in Section 5.2.2 and relying only on the Kalman filter measurement innovation and its covariance is implemented in order to detect and exclude the measurements likely to be faulty based on the consistency between the measurements of the

different hybridization architecture sensors. The contribution of this test to the improvement of the filter accuracy and robustness against faults will be highlighted in Chapter 6.

5.4 Conclusion

In this chapter, the integrity concept of the navigation solution proposed in Chapter 4 is addressed. The sensor errors that may lead to an HMI are identified, particularly for the SLAM and the GPS. The SLAM errors are identified and classified into fault-free mode and failure modes and the GPS errors are recalled in order to define the integrity fault-tree that is used to allocate the integrity risk between the different sensors and the different modes. Furthermore, the integrity monitoring principle through the FD or the FDE function is explained and the state of the art of the existing FD/FDE techniques adapted to our case of study is reviewed. A discussion on the HPL value and the integrity function availability is addressed showing that the lower the HPL value, the better is the integrity function availability.

Then the limitations and the challenges related to GPS, SLAM and filter integrity monitoring are identified assuming that the INS and WSS are fault-free using sensor redundancy. We summarize these challenges in the flowing:

- GPS challenges:
  - The presence of buildings and structures hiding the satellites makes the GPS signal integrity challenging. In fact, integrity monitoring is based on GPS signal redundancy. This redundancy is scarce in urban canyons making the integrity monitoring difficult to perform.
  - The GPS error distribution is assumed to be zero-mean Gaussian. This assumption is invalid in urban scenarios because of the presence of multipath and NLOS. Even with overbounding techniques and environment modeling, this assumption can be violated resulting in wrong modelling of the error distribution.

- SLAM challenges
  - The slam error and particularly failure mode error should be characterized in terms of probability of occurrence, number of simultaneous measurement errors and error signature. This requires having done sufficiently various and long data collections, evaluated the SLAM on it and identified all the possible failures based on these collections. In addition, most of the SLAM failures are environment-dependent making this task a real challenge.
  - The SLAM includes very complex processes making the mapping of the measurement error into the navigation solution domain complicated. However, this step is necessary to be able to establish a theoretical HPL formula.

- Overall system challenges
  - In general, state of the art integrity monitoring techniques are not designed to handle many simultaneous faults and assume a single measurement failure. This assumption is not valid for sensors using several measurements with errors that

are dependent on the environment in which the vehicle moves such as GPS and SLAM.

- In general, sequential integrity monitoring techniques have a high computational complexity making them not adapted to real time. This is the case for the MSS algorithm which considers several banks of sub-filters or the Kalman-based filter of (Joerger, 2013) which requires a batch least-squares algorithm to evaluate the integrity risk. The innovation test is a simpler method that has shown to be efficient (the evaluation of this method is performed in Chapter 6)

- Integrity requirements challenges
  - The most challenging point in the integrity requirements is the value of HAL which should be very low to avoid the risk of human harm or material damage. This value is evaluated to about $1.5\ m$ for the intended applications and requires a much better (submeter level) accuracy than the accuracy provided by the architecture defined in Chapter 4 (meter level).

Chapter 6

Results and Performance Analysis

This chapter describes the tests carried out, along with the details about the followed experimental procedure. The considered scenarios are vehicular navigation in urban canyon and airport surface contexts. The performance of the developed navigation solutions (position, velocity and attitude) is assessed based on these tests.

First, we start by presenting the equipment used to collect real data and describing the environments where tests were carried out.

Second, the GPS navigation solution is assessed, and the limitations of this solution are highlighted.

The third part is dedicated to analyzing ORB-SLAM navigation outputs and to assess its performance based on the localization task. In fact, visual SLAM is advantageous compared to visual odometry thanks to its map and the optimizations using this map to provide an acceptable navigation solution. The case where SLAM explores for the first time an environment is not considered. Only the SLAM localization process based on a previously constructed and adjusted map is studied. In this study, the trajectory coordinates are computed by the SLAM algorithm and are output in its vision frame. Note that this trajectory is computed using the same collected data that was processed to generate the map in a previous step.

The last part presents a synthesis of performance assessment of the hybridized navigation solution proposed in Chapter 4 and improved based on the innovation test described in Chapter 5. The contribution of VSLAM to the hybridized navigation solution is particularly studied.

Contents

6.1 Experimental procedure 100
6.1.1 Equipment 100
6.1.2 Description of the trajectory 102
6.2 Sensors and filter performance analysis 105
6.2.1 GPS solution 105
6.2.2 SLAM solution 110
6.2.3 Hybridized navigation solution 118
6.3 Conclusion 129

6.1 Experimental procedure

6.1.1 Equipment

The developed algorithms are implemented and tested in post-processing on real data collected in Toulouse using an equipped Citroën Jumpy car, illustrated in Figure 6-1.



Figure 6-1 Measuremet campaign equipment

The equipment used in the tests consists of:

- An Xsens-MTI IMU providing the measurements of three axes accelerometers and gyroscopes and running at 100 Hz. The main characteristics of the IMU are given in Table 2.4
- A L1-only-Ublox LEA M8T GNSS receiver running at 1 Hz connected to a low-cost antenna. This receiver provides GPS and GLONASS measurement data, but only GPS data is used in our hybridization to evaluate the impact of single frequency single constellation aiding in multi-sensor fusion framework
- A μEye UI-3240CP-C-HQ camera. Its resolution is equal to 1280×1024 pixels and its frame rate can reach 60 fps. To avoid dealing with huge files, the frame rate is set to 25 fps. The lens is a Tamron M118FM08 with a horizontal and vertical angles of view of 50.8° and 38.6° respectively. The images captured by this camera are processed in post-processing by the ORB-SLAM software.
- The WSS data is generated with the model given in equation (2.41) because of the lack of this information in this dataset.
- A device used for estimating a reference solution for error analysis. This device is the NovAtel SPAN system, consisting of a ProPak6 receiver processing GPS L1/L2 and a tactical grade IMU from Northrop-Grumman Litef Gmbh (UIMU-LCI). The reference solution is computed in post-processing with the NovAtel Inertial Explorer software. The tightly coupled strategy is applied to combine GPS and IMU data in forward and

backward processing. The double difference technique is used for GPS. The base stations used are TLSE, TLIA, TLSG and TLMF whose locations with respect to the car are illustrated in Figure 6-2. for the urban test (this test will be described later)



Figure 6-2 Reference stations distance with respect to the car (urban test)

The accuracy of the reference solution as estimated by the NovAtel software is illustrated in Figure 6-3 for position, for the urban test. It can be noticed that, in this illustration, the reference position has decimeter level uncertainty, up to $1\,m$ in the worst case, as computed by the NovAtel SPAN equipment. The reference velocity and attitude are also used in the performance assessment of the integrated algorithm. The velocity has a cm/s level of accuracy while the attitude accuracy is in the order of arcminute (0.01°).

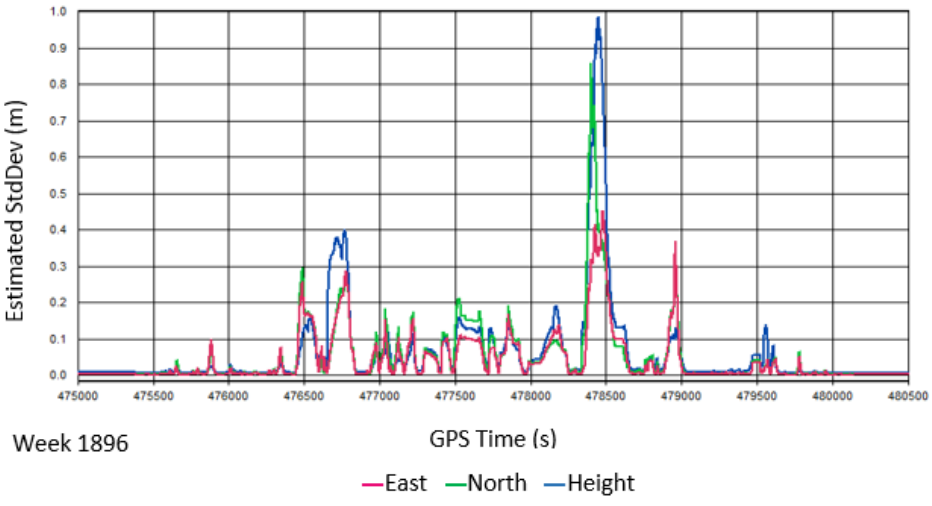


Figure 6-3 Reference position accuracy (standard deviation)

The equipment is mounted on the car with the configuration described in Figure 6-4. The GPS antennas and the camera are mounted on the roof top of the car. The camera is oriented to the forward direction. The IMU is mounted inside the car. The lever arm between the SPAN antenna and the SPAN IMU is set such as only the vertical component is nonzero.

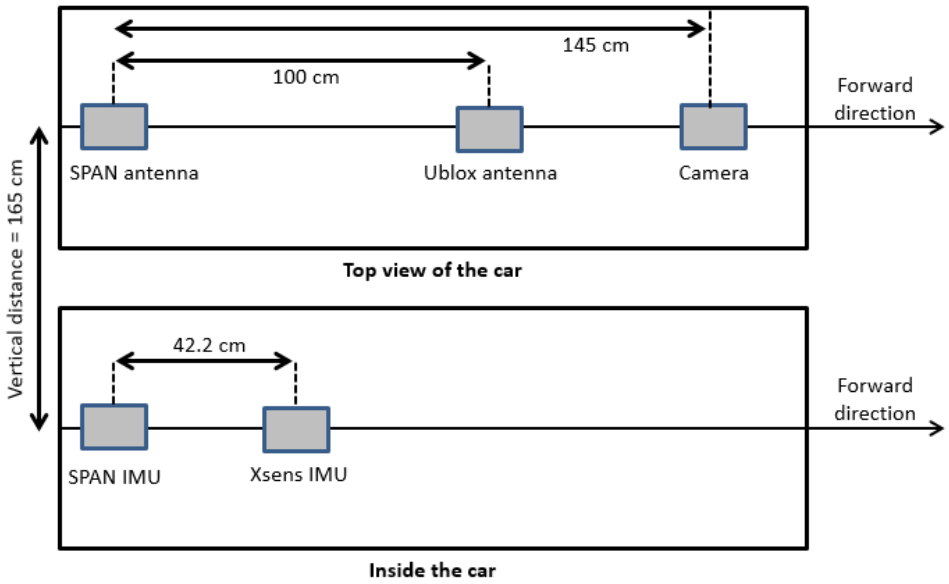


Figure 6-4 Sensor configuration for data collection

Sensor data is collected and recorded using ROS (Quigley, 2009) in a single file called *rosbag*. The rosbag file includes all the sensor information timestamped using the data collection computer Unix Time. This time is computed by the computer clock. To prevent its drift, this time is synchronized with the GPS-Time estimated by the Ublox LEA M8T GNSS receiver. To perform the synchronization task, the Chrony (Chrony, 2016) suite based on the Network Time Protocol (NTP) is used. The Ublox LEA M8T GNSS receiver is connected to the computer serial port using an RS-232 serial connection and outputs PPS (Pulse Per Second) signals. Then, Chrony uses these GPS PPS signals to prevent the computer clock from drift, allowing therefore a good sensor timestamping.

To avoid dealing with big size files because of the presence of image files, the measurement campaigns have been split into several sub-measurement campaigns. For simplicity reasons, only results on measurement sub-campaigns having different characteristics are presented in this report.

6.1.2 Description of the trajectory

6.1.2.1 Suburban and dense urban data collection

The data collection was done on May, 13th 2016 in Toulouse and corresponds approximately to 80 minutes of data. It starts from ENAC, goes through Toulouse downtown and finishes in ENAC. The speed of the vehicle varies from 0 to 50 km/h with frequent stops due to the traffic. The full trajectory is given in Figure 6-5. It is characterized by many loops to test the SLAM Loop Closure and its contribution to the improvement of the navigation performance.

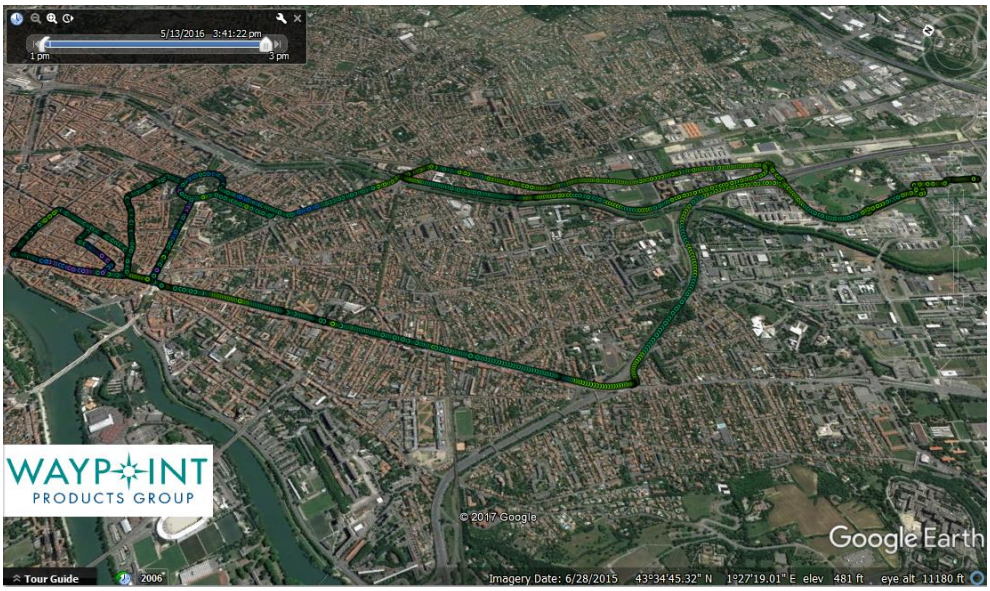


Figure 6-5 Urban and suburban trajectory

The traveled trajectory is characterized by two types of environments:

- A suburban environment with low buildings, trees, etc. but a relatively good satellite visibility. The car travels this type of environment in the beginning and the end of the trajectory.



Figure 6-6 Suburban environment description

- A dense urban environment with urban canyons and important building mask angles with a very high multipath and NLOS occurrence probability. The car travels this type of environment in the middle of the trajectory.



Figure 6-7 Urban environment description

6.1.2.2 Airport environment

The data collection was done on May, 24th 2016 in Toulouse Francazal airport and corresponds approximately to 75 minutes of data. The zones near the hangars, in the apron, in taxiways and in the runway are travelled. The vehicle speed varies from 0 to 90 km/h. The full trajectory is given in Figure 6-8.



Figure 6-8 Francazal airport trajectory

The traveled trajectory is characterized by two types of environments:

- A suburban environment when the vehicle moves next to the hangars and in the apron
- An open sky environment when the vehicle moves in the taxiways and in the runway

6.2 Sensors and filter performance analysis

The 2 measurement campaigns presented in Section 6.1.2 are used to evaluate the performance of each sensor as well as the integrated navigation solution provided by the Kalman filter. First, the L1 GPS performance and challenges, especially in urban environments, are emphasized. Second the monocular visual SLAM performance evaluation is performed. Then the navigation solution output by the Kalman filter is evaluated using different sensor combinations.

This performance analysis is based on the evaluation of the horizontal position, velocity and attitude (heading). The following metrics are used:

- The horizontal position error, the horizontal velocity error and the vehicle heading error, along with the associated 3σ[17] bound. In the case of a zero-mean Gaussian distribution of the error, $99{,}7\%$ of the error must be included in the $[-3\sigma, 3\sigma]$ bound.
- The horizontal position error statistics (after the filter convergence in the case of the assessment of the hybridized solution):
  - The maximum error
  - The 95th and the 99th error percentiles[18].
  - The RMS error

6.2.1 GPS solution

6.2.1.1 Suburban and dense urban environment

The GPS solution is first assessed using the C/N_0 of the received signals. This quantity is an indicator of the received signal quality that is independent of the acquisition and tracking algorithms used by a receiver. Typical values in an L1 C/A code receiver are comprised between 37 and 50 dB-Hz, but high sensitivity receivers can track signals with a much lower C/N_0. This value can drop because of local effects such as multipath and NLOS as explained previously. Figure 6-9 illustrating the C/N_0 of received signals, highlights the drop of their values. A non-negligible ratio of received signals has a C/N_0 lower than 30 dB-Hz, characterizing signals degraded by the surrounding environment. Therefore, we can think of setting a C/N_0 mask to avoid the local effects. The C/N_0 mask is set to 30 dB-Hz.

[17] The parameter σ is the uncertainty of the navigation solution output by the navigation module.
[18] The error percentile is a measure that indicates the value below which a given percentage of the error fall. For example, the 95th percentile is the error value below which we find 95% of errors.

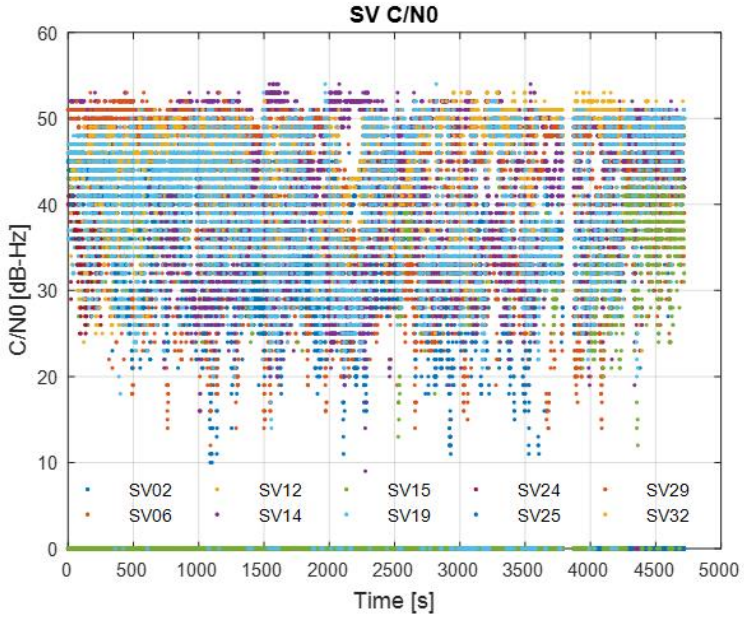


Figure 6-9 C/N_0 of GPS measurements in suburban-urban environment

Figure 6-10 illustrates the number of satellites in view before and after applying the C/N_0 mask. We notice that the number of visible satellites may drop under 4 after applying the C/N_0 mask.

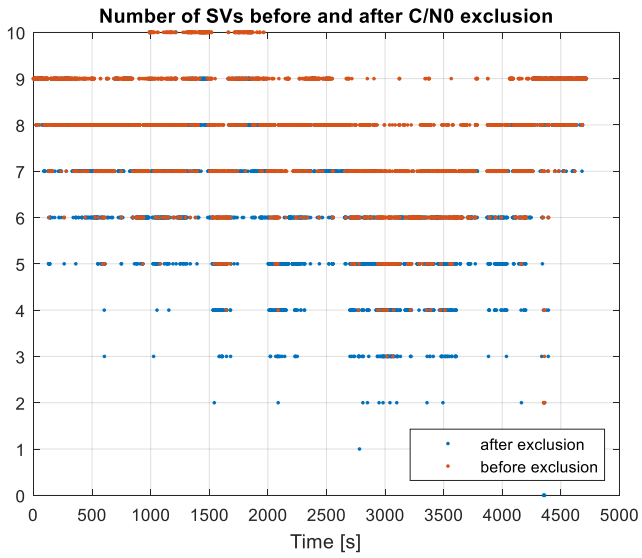


Figure 6-10 Number of GPS measurements before and after exclusion through comparison to $C/N_0 = 30$ dB-Hz in suburban-urban environment

Table 6.1 gives the availability of satellites. For example, we have between 4 and 5 satellites during 3.20% of the measurement campaign duration before exclusion, while after exclusion, we have between 4 and 5 satellites in view 19.98 % of time. We are going to study the impact of satellite removal on the GPS navigation solution computed with the least square method mentioned in Section 2.1.3.

Number of SVs	Before exclusion	After low C/N_0 exclusion
$0 \leq N < 4$	0.13 %	2.63 %
$4 \leq N < 6$	3.20 %	19.98 %
$6 \leq N < 8$	33.82 %	42.98 %
≥ 8	62.84 %	34.92 %

Table 6.1 Satellite availability before and after exclusion through comparison to $C/N_0 = 30$ dB-Hz in suburban-urban environment

The Horizontal Dilution Of Precision (HDOP) describing the impact of the satellite geometry on the horizontal accuracy of the GPS solution is computed before and after the low C/N_0 exclusion and its evolution is compared to the evolution of the horizontal position error and to the horizontal velocity error in Figure 6-11. It can be noticed that the low C/N_0 exclusion degrades dramatically the satellite geometry, and so the accuracy of the navigation solution.

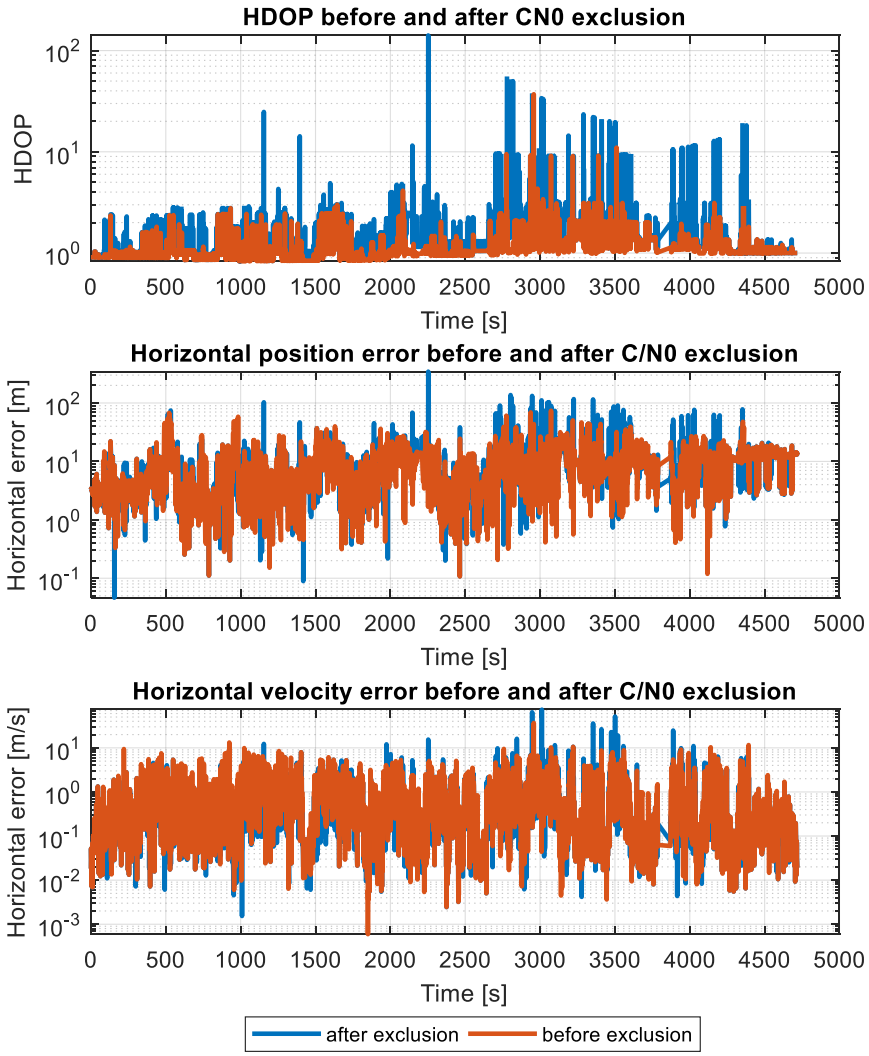


Figure 6-11 Impact of the low C/N_0 exclusion on HDOP, horizontal position error and horizontal velocity error in suburban-urban environment

Table 6.2 gives more details about the GPS accuracy before and after low C/N_0 exclusion. It shows that the exclusion based on a C/N_0 mask of 30 dB-Hz is not an efficient way to improve the navigation solution because it degrades the satellite geometry which degrades in turn the accuracy of the navigation solution. This degradation is especially noticed on the position value while the velocity performances are comparable before and after the low C/N_0 exclusion.

	Horizontal position error (m)		Horizontal Velocity error (m/s)	
	Before exclusion	**After low C/N_0 exclusion**	**Before exclusion**	**After low C/N_0 exclusion**
Max	73.75	344.40	36.64	73.46
RMS	11.69	15.66	1.67	2.30
95th percentile	22.47	23.57	3.50	2.70
99th percentile	38.61	64.00	6.25	6.09

Table 6.2 Comparison of the navigation solution accuracy before and after low C/N_0 exclusion in suburban-urban environment

Note that Figure 6-10 and Figure 6-11 highlight the characteristics of the environment. The beginning of the measurement campaign is characterized by a high number of satellites in view, and consequently a good (low) HDOP and a small position error. The test continues in severe environment with a decrease of the number of satellites in view and their C/N_0, resulting in degradation in the HDOP and the navigation solution.

The full measurements without low C/N_0 exclusion will therefore be considered for the remaining parts of this Chapter, for suburban-urban environments. Figure 6-12 and Figure 6-13 illustrate the GPS horizontal position and velocity errors without low C/N_0 exclusion and their corresponding 3σ bound. It can be seen that the 3σ bound envelops the errors and describes well the error variations. This validates the error models explained in Chapter 2 used to model the different GPS error sources, especially the multipath model in suburban and urban environment as well as the Doppler noise model depending on the C/N_0 and the vehicle speed.

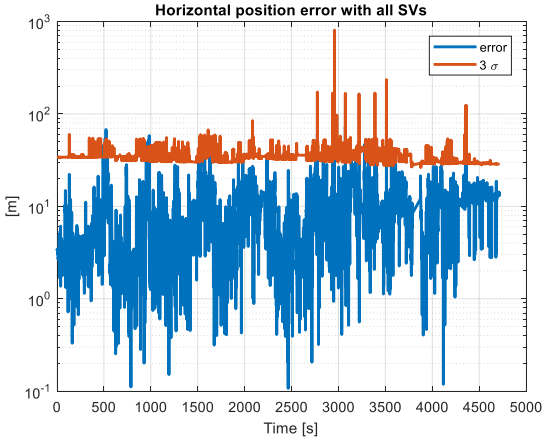


Figure 6-12 GPS L1 horizontal position error and its associated 3σ bound without low C/N_0 exclusion in suburban-urban environment

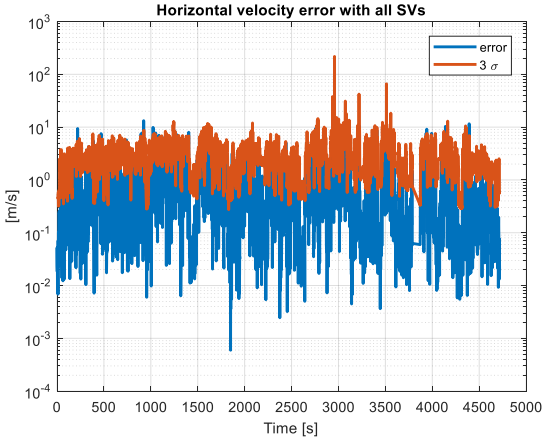


Figure 6-13 GPS L1 horizontal velocity error and its associated 3σ bound without low C/N_0 exclusion in suburban-urban environment

6.2.1.2 Airport environment

Similar to the urban environment, we start first by evaluating the received signals C/N_0 values in Figure 6-14. We can see that the C/N0 values are higher than in the suburban-urban environment. However, these values drop in the beginning and the end of the measurement campaign. This is because the test car has moved along the airport hangars and next to the airport buildings, making the GPS antenna receive multipath signals, in these two periods.

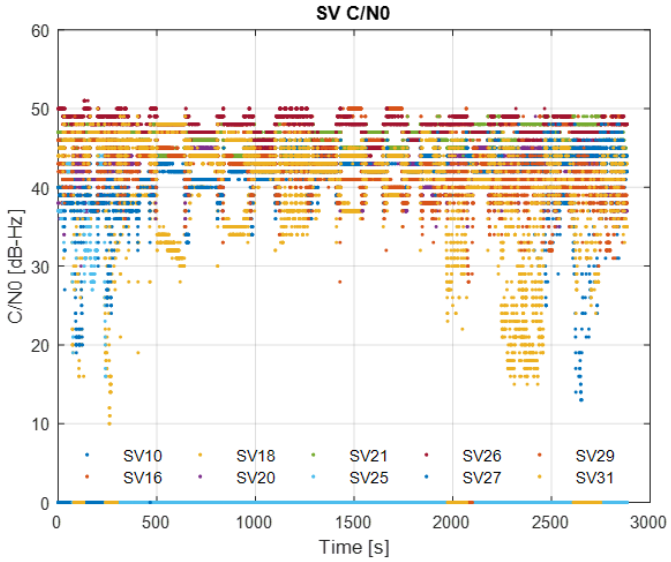


Figure 6-14 C/N_0 of GPS measurements in airport environment

A 30 dB-Hz C/N_0 mask is applied. Figure 6-15 illustrates the number of satellites before and after exclusion. Even after exclusion, the number of satellites in view is not lower than 5. The number of excluded satellites is not important, and the geometry remains satisfactory. This explains the slight improvement of the navigation solution accuracy parameters given in Table 6.3. For this reason, this exclusion is kept for the remaining of the tests in airport environment.

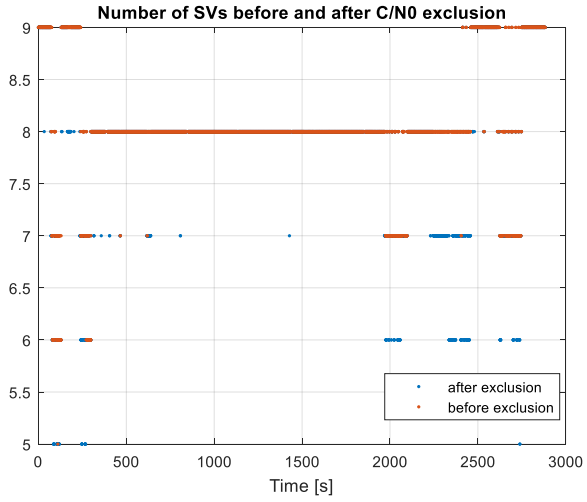


Figure 6-15 Number of GPS measurements before and after exclusion through comparison to $C/N_0 = 30$ dB-Hz in airport environment

	Horizontal position error (m)		Horizontal Velocity error (m/s)	
	Before exclusion	**After low C/N_0 exclusion**	**Before exclusion**	**After low C/N_0 exclusion**
Max	57.88	35.32	4.87	4.87
RMS	6.87	5.80	0.33	0.15
95th percentile	16.52	16.07	0.45	0.22
99th percentile	24.60	17.46	1.36	0.48

Table 6.3 Comparison of the navigation solution accuracy before and after low C/N_0 exclusion in airport environment

To evaluate the open sky measurement model, especially the multipath model given in equation (2.14) and Table 2.2 for pseudorange and pseudorange rate respectively, the GPS horizontal position and velocity errors and their corresponding 3σ bound are plotted in Figure 6-16 and Figure 6-17. We can see that the pseudorange model is satisfying in regions near buildings (beginning and end of the trajectory); however, in runways and taxiways (middle of the trajectory), this model overestimates the navigation solution uncertainty. As for velocity, the error model based on the C/N_0 and the vehicle speed describes well the uncertainty of the velocity error.

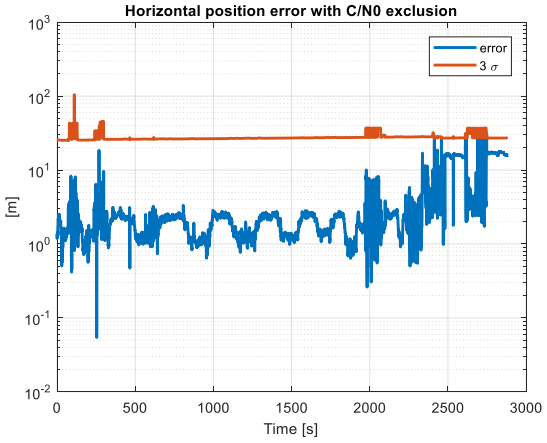


Figure 6-16 GPS L1 horizontal position error and its associated 3σ bound with low C/N_0 exclusion in airport environment

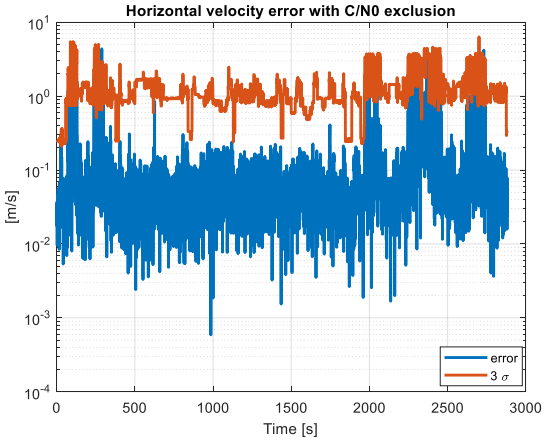


Figure 6-17 GPS L1 horizontal velocity error and its associated 3σ bound with low C/N_0 exclusion in airport environment

6.2.2 SLAM solution

The images collected by the camera are processed using ORB-SLAM. This algorithm performance is assessed using the method described in Section 3.4. We choose the ATE (Absolute Trajectory Error) given by equation (3.17), since loop closures are considered in the data collections.

From now on, the performance assessment is based on 2 tests:

- Urban environment (Test 1)

Test 1 lasting about 12 min belongs to the urban measurement campaign described by Figure 6-5. This part of the trajectory goes through harsh GPS environment. Figure 6-18 illustrates SLAM trajectory, and Figure 6-19 shows some screenshots of SLAM processing, where the

green points are the features detected by SLAM. We can see from the trajectory that, even with map construction and adjustment previously done, there is a map discontinuity that remains at the moment of loop closure, resulting in trajectory discontinuity. Although this discontinuity persists, the drift of the map is less important than in an unknown environment.

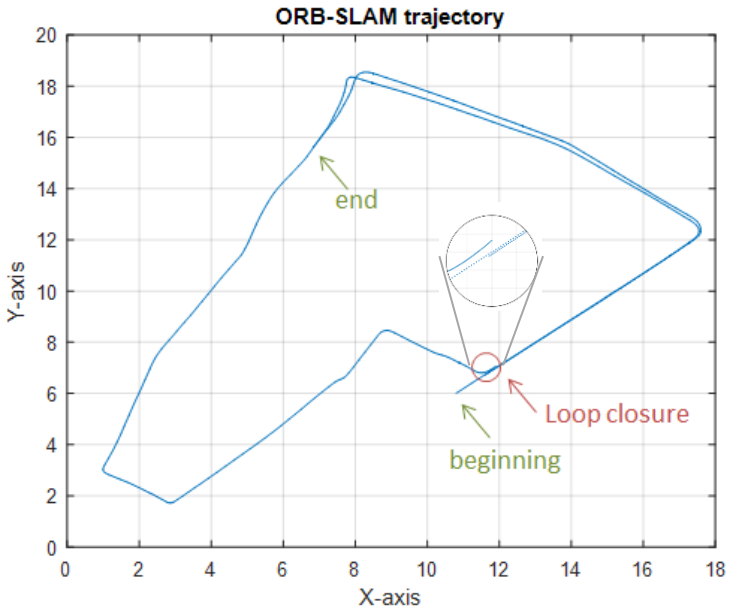


Figure 6-18 ORB-SLAM trajectory in vision frame (Test 1)



Figure 6-19 ORB-SLAM image processing (Test 1)

- Airport environment (Test 2):

Test 2 lasting 4:16 min is performed in airport environment and describes a loop closure trajectory. In the scope of this thesis, only the hangar and apron areas of the airport are studied for many reasons. First, this thesis targets airport ground vehicles that move most of the time in these areas. Second, unlike taxiways and runways where there are no structures resulting in multipath and NLOS, the hangars and apron areas are characterized by buildings and structures that cause this type of GNSS error. As highlighted in Section 6.2.1.2, GNSS needs aid in this type of area where its performance is degraded and has good performance in open-areas like taxiways and runways. The final reason why we only focus

on apron and gate zone is the fact that SLAM has good performance in textured environment and bad performance in non-textured environment. The lack of features illustrated in Figure 6-20 (left figure), especially in the runway, makes SLAM loose the map track as soon as the vehicle turns, and no loop adjustment could be performed in this environment. Therefore, Test 2 is performed only in the airport hangar and apron areas (right figure).

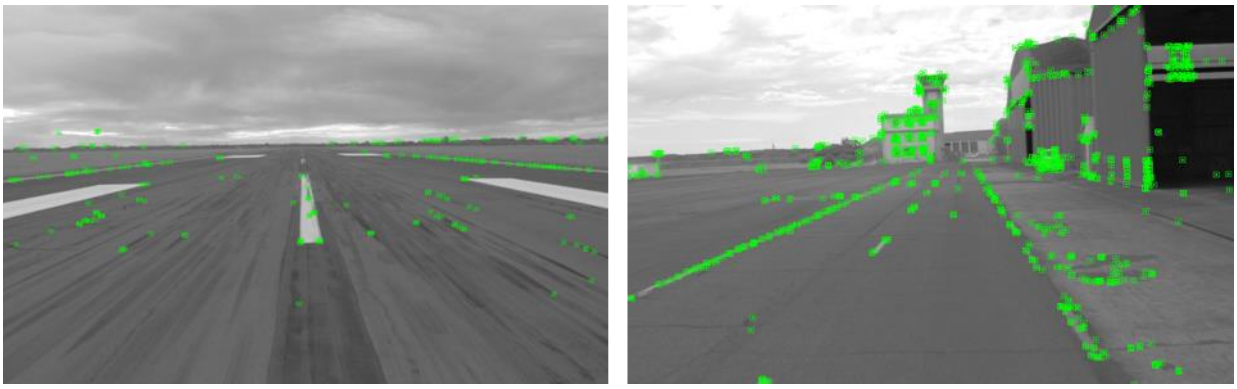


Figure 6-20 ORB-SLAM image processing (Test 2)

Figure 6-21 illustrates the ORB-SLAM trajectory. Similar to Test 1, the map dicontinuity resulting in trajectory discontinuity is also noticed. This means that even with map optimization, a small trajectory drift remains.

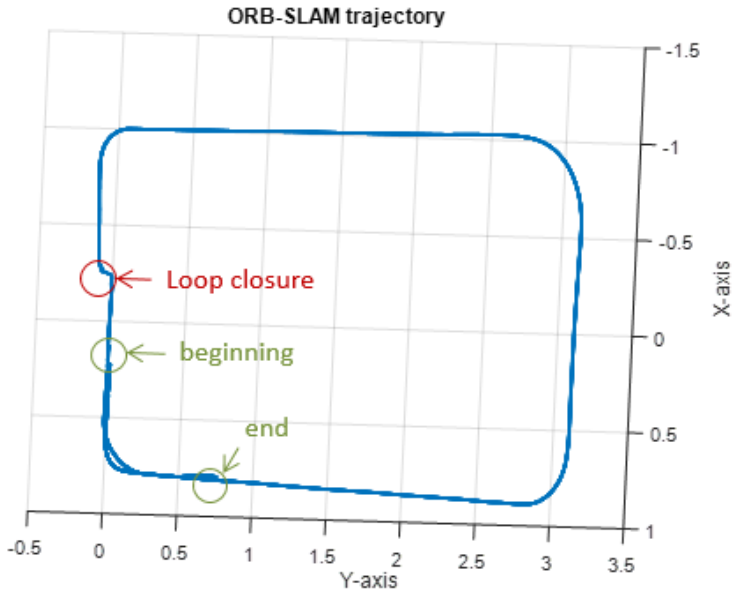


Figure 6-21 ORB-SLAM trajectory in vision frame (Test 2)

6.2.2.1 Test 1: VSLAM performance in urban environment

First of all, we are going to illustrate the car trajectory as it is output by SLAM and compare it to the reference trajectory. This illustration is performed in Figure 6-22 which represents the meter unit trajectory and which shows with Figure 6-18 that the trajectory estimated by SLAM has the same shape as the reference trajectory up to scale, a rotation and a translation as explained in Chapter 3, which is the transformation from the vision to the local frame.

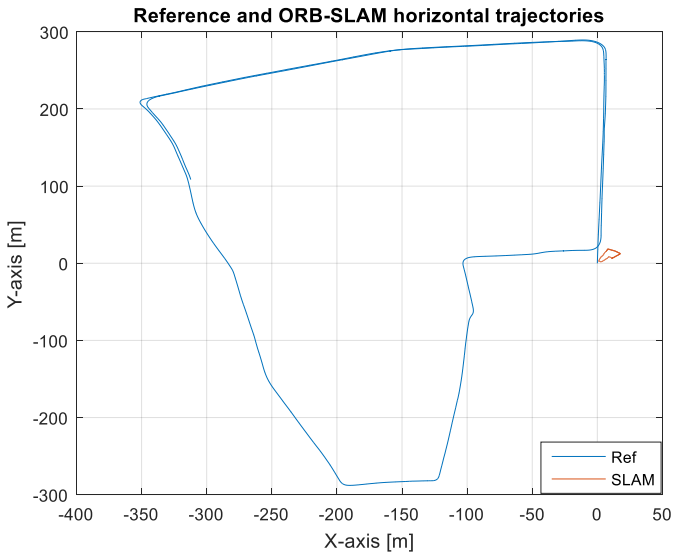


Figure 6-22 Comparison between ORB-SLAM and reference trajectories (Test 1)

The parameters are estimated using Horn method (Horn, 1987) and are given in Table 6.4 ORB-SLAM transformation estimation (Test 1). Note that this method requires at least three different points expressed in both coordinate frames to estimate the transformation between these points.

Scale factor	**Quaternion value (from (l) to (v))**				**Translational offset (between (l) and (v))**		
	q0	qx	qy	qz	px	py	pz
0.0316	0.9296	-0.0251	-0.0514	-0.3640	-19.15	-150.08	12.04

Table 6.4 ORB-SLAM transformation estimation (Test 1)

To validate this estimation, the transformed SLAM trajectory is plotted and compared to the reference trajectory in Figure 6-23.

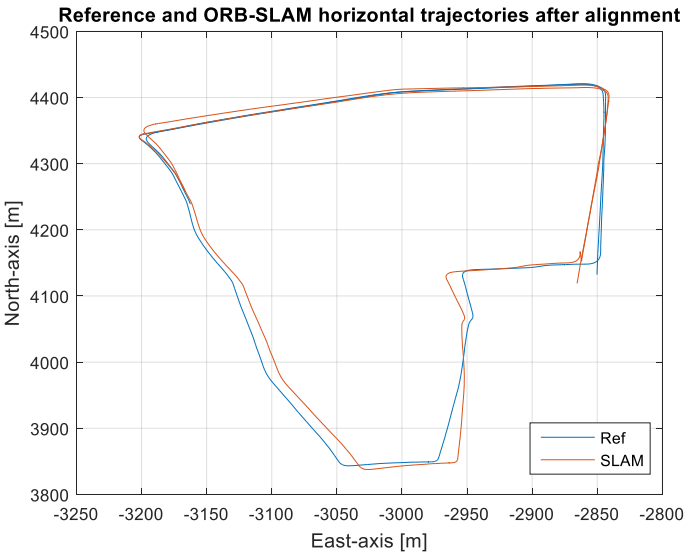


Figure 6-23 Reference and ORB-SLAM trajectories after alignment (Test 1)

We can see in this figure that if we consider the vision to local frame transformation constant, the trajectories do not perfectly match. We can conclude that local-to-vision-frame transformation is not constant in time, which confirms the drift of the map and consequently the drift of the trajectory.

For this reason, we computed the transformation on a temporal sliding window to understand the 7 transformation parameters' behavior. Each sliding window consists of 1500 samples (approximately 1 min). This is illustrated in Figure 6-24**Erreur ! Source du renvoi introuvable.**. In this figure, The SLAM scale factor estimated by the Horn method is compared to the scale factor estimated point by point. This latter estimation is based on equation (3.13) and assumes that the scale factor value does not change between two successive frames:

$$(\tilde{\boldsymbol{p}}_c^v)_{k+1} - (\tilde{\boldsymbol{p}}_c^v)_k = k_v \boldsymbol{C}_{l2v}((\boldsymbol{p}_c^l)_{k+1} - (\boldsymbol{p}_c^l)_k) \tag{6.1}$$

Therefore

$$|k_v| = \left\| \frac{(\tilde{\boldsymbol{p}}_c^v)_{k+1} - (\tilde{\boldsymbol{p}}_c^v)_k}{(\boldsymbol{p}_c^l)_{k+1} - (\boldsymbol{p}_c^l)_k} \right\|_2 \tag{6.2}$$

Note that the point by point scale factor has been filtered using a moving mean function for better readability. The sliding window is the same as the one used with the Horn method.

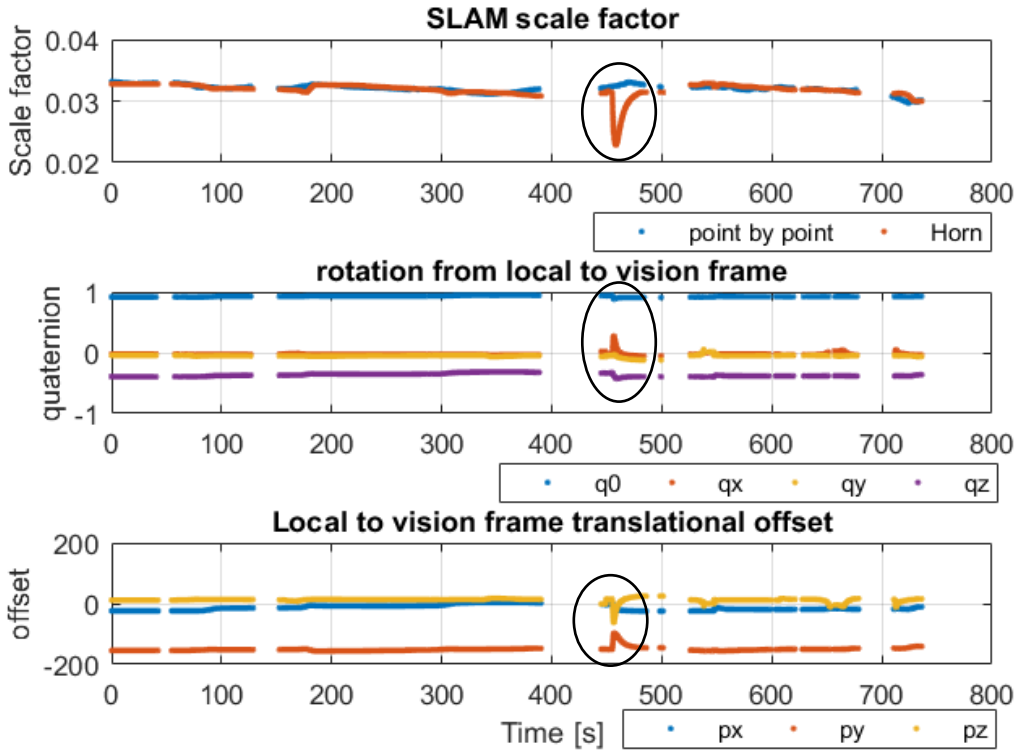


Figure 6-24: ORB-SLAM transformation estimation using a sliding window (Test 1)

First, we notice that the scale factor estimated by Horn method is similar to the scale factor estimated point by point. Second, it can be seen that the 7 transformation parameters vary slowly in time and are not constant. We can also see that the only jump of this transformation parameters' values occurs at time epoch 455s. This jump occurs when the SLAM position switches from using the end of the map to using again the beginning of the map to compute the trajectory. This position jump, in addition to the slow transformation drift, show that, although loop closure reduces considerably the map drift, it does not remove it completely. Despite this little drift, the most convenient way to model these parameters in the Kalman filter is to consider them as constant between two time instants because of their slow variation in time.

The SLAM ATE is computed using the transformation assumed varying in time. Figure 6-25 illustrates this error based on the computation of the previously mentioned transformation. We can see that the error is relatively small assuming a varying transformation except for the loop closure jump instant, where the horizontal error is of few meters. Table 6.5 ORB-SLAM accuracy after alignment (Test 1)Table 6.5 gives the maximum, the RMS, the 95th and the 99th percentile of the SLAM ATE.

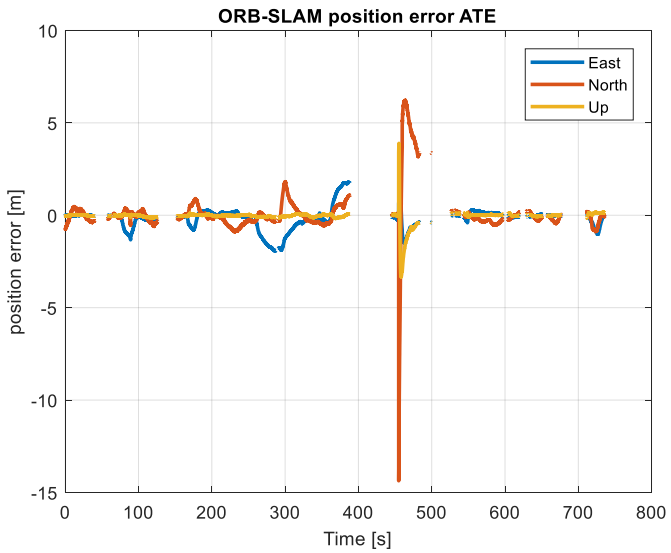


Figure 6-25 ORB-SLAM ATE after alignment (Test 1)

	Max (m)	RMSE (m)	95th percentile	99th percentile
East	1.93	0.56	1.49	1.81
North	14.37	1.08	1.08	5.78
Up	3.89	0.32	0.15	1.76

Table 6.5 ORB-SLAM accuracy after alignment (Test 1)

6.2.2.2 Test 2: VSLAM performance in airport environment

As in the previous test, a comparison is performed between the SLAM and the reference trajectories in Figure 6-26. It shows that the SLAM is affected by a very small scale factor, a rotation and a translational offset as explained in the previous test. The comparison between the scale factor in airport environment and in urban environments shows that the scale factor in airport environment is smaller. This is due to the depth of the observed features. In fact, in urban

environments, the observed features are closer than the features observed in airport. This makes the scale factor higher.

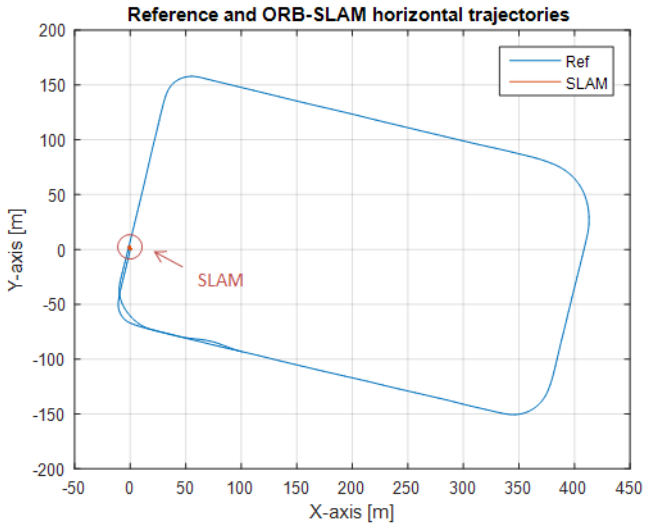


Figure 6-26 Comparison between ORB-SLAM and reference trajectories (Test 2)

As in Test 1, the computation of the local-to-vision frame transformation parameters is performed using Horn method assuming first that this transformation is constant. These parameters are given in Table 6.6.

Scale factor	**Quaternion value (from (l) to (v))**				**Translational offset (between (l) and (v))**		
	q0	qx	qy	qz	px	py	pz
0.0079	0.6245	-0.0329	5e-5	0.7803	-32.75	68.90	-4.51

Table 6.6 ORB-SLAM transformation estimation (Test 2)

Similar to Test 1, this transformation is used to compare the SLAM trajectory to the reference trajectory in Figure 6-27.

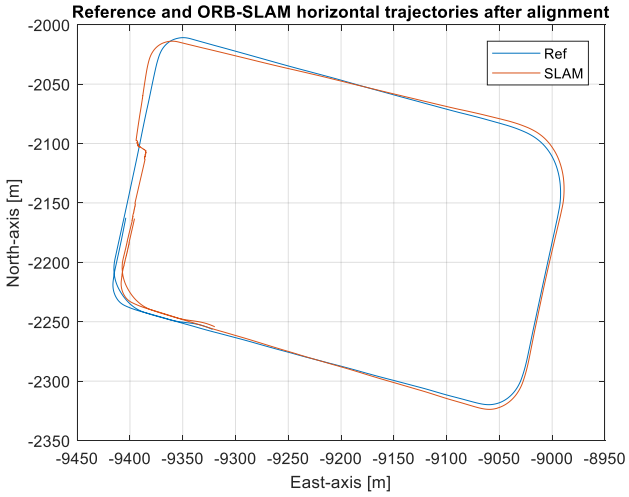


Figure 6-27 Reference and ORB-SLAM trajectories after alignment (Test 2)

The comparison between the two trajectories shows that the constant transformation is not adapted since the trajectories do not totally match. We consider therefore a varying transformation in time, as in Test 1.

Figure 6-28 illustrates the estimated parameters using Horn method on a 1min sliding window. As in Test 1, these parameters vary slowly in time. This highlights the improvement brought by loop closure. However, the jump occurring around instant 180s (time corresponding to loop closure) proves that the map is affected by a discontinuity at the time of loop closure.

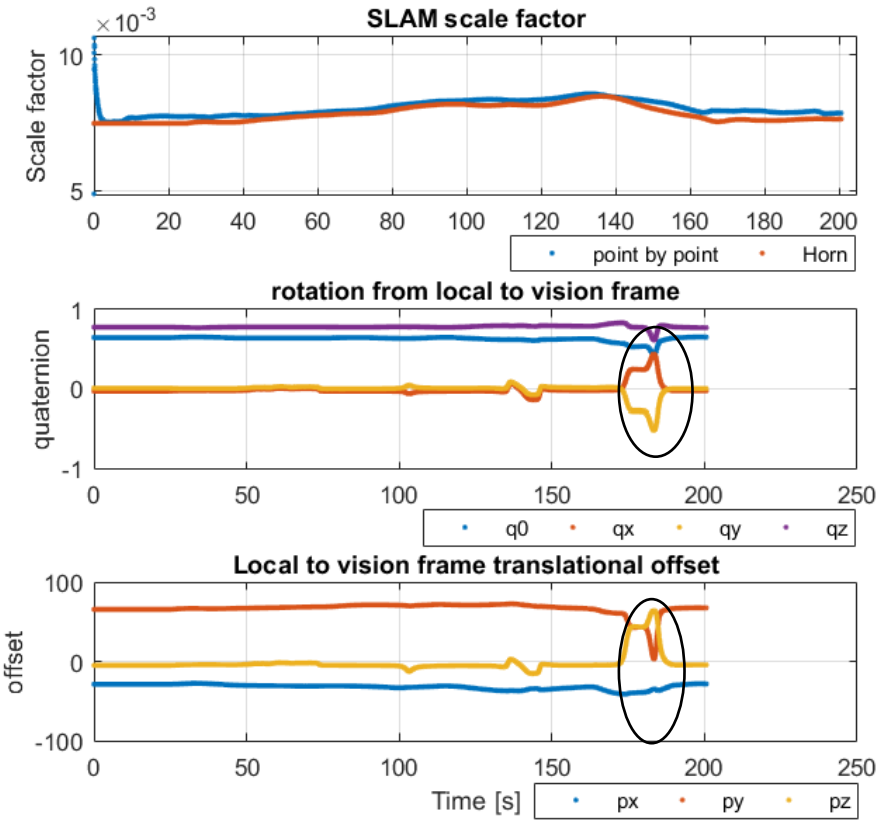


Figure 6-28 ORB-SLAM transformation estimation (Test 2)

The ATE computation is also performed for Test 2. It follows the same steps as for Test 1. The SLAM ATE statistics are summarized in Table 6.7. Compared to urban environments, the ATE is slightly worse. This may be explained by the lack of features first and to their bad spatial distribution (very far, concentrated is one area where there are buildings or aircrafts) compared to urban environment.

	Max (m)	RMSE (m)	95th percentile	99th percentile
East	4.95	1.83	2.18	3.97
North	3.47	1.04	1.24	1.54
Up	4.14	0.78	0.57	3.99

Table 6.7 ORB-SLAM accuracy after alignment (Test 2)

6.2.3 Hybridized navigation solution

6.2.3.1 GPS/INS/WSS/NHC

First of all, the classical GPS/INS/WSS/NHC hybridization using an EKF is implemented and studied. Only Test 1 is discussed since the results are similar for Test 2. This study consists of analyzing the estimated navigation solution error in terms of horizontal position, horizontal velocity and heading. It validates also the filter convergence by analyzing the covariance properties and confirms the observability theoretical results found in Chapter 4. Figure 6-29, Figure 6-30 and Figure 6-31 illustrate respectively the horizontal position error, the horizontal velocity error, the heading error (in blue) and their corresponding 3σ bound (in red), where σ is the state's uncertainty provided by the EKF. We notice that the errors are almost always below the 3σ bound. This validates the measurement models used for the different sensors. We notice also that the 3σ bound reflects the observability issue. When the car does not move, the heading is unobservable, and its associated uncertainty increases to reflect its unobservability at stops.

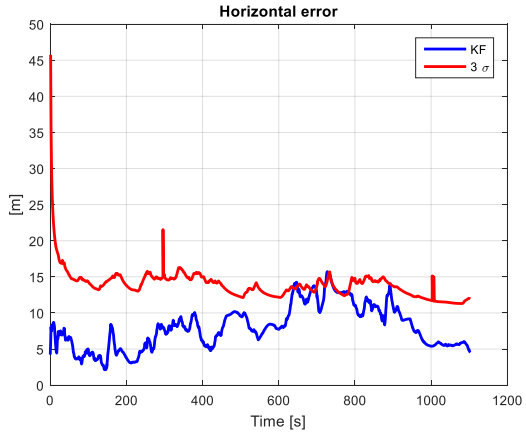


Figure 6-29 Horizontal position error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1)

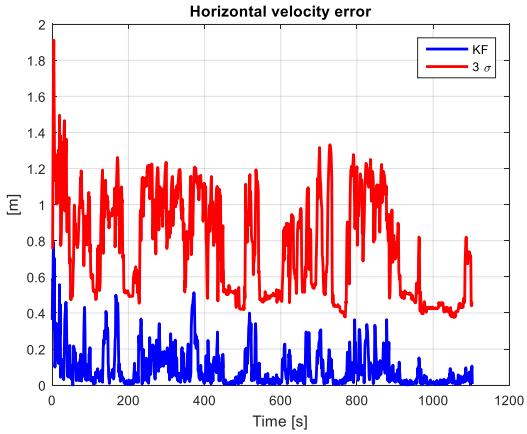


Figure 6-30 Horizontal velocity error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1)

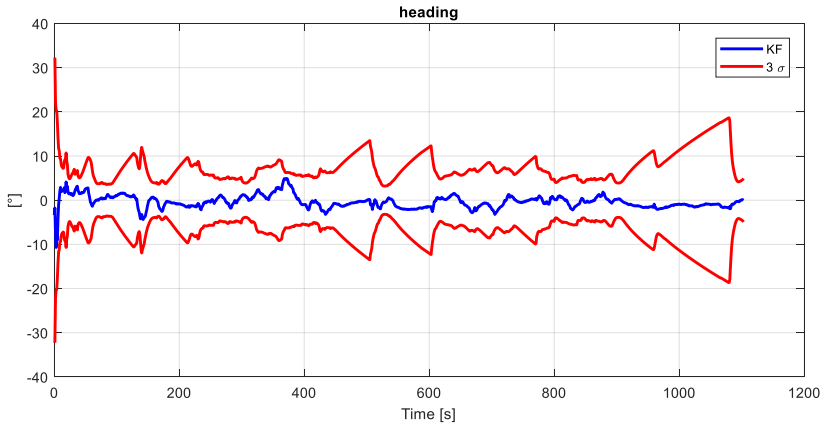


Figure 6-31 Heading error and its associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1)

Figure 6-32 and Figure 6-33 provide the EKF estimation of the IMU biases and scale factors for both accelerometers and gyroscopes. In blue is the bias/scale factor error estimated by the

filter and in red their corresponding 3σ bound. The magenta curve represents the IMU bias and scale factor after summing the estimated errors. It can be noticed that the filter converges since its 3σ bound decreases in time. In addition, the estimated values of IMU biases and scale factors errors are included in this bound. We can also validate the observability theoretical results by analyzing the uncertainty provided by the filter. The accelerometer biases are observable since their uncertainty decreases continuously. The accelerometer scale factors are observable on the y and z axes because these axes correspond respectively to along track and vertical axes that are excited by the car motion and the gravity. However, the uncertainty of the x-axis scale factor does not decrease continuously because the scale factor is only observable when this axis is excited (in turns). For gyroscopes, their biases are theoretically observable if at least two accelerometers are excited. It is almost always the case thanks to the vehicle motion and gravity. When the vehicle does not move, these biases are estimated thanks to ZARU (Zero Angular Rate Update). This is confirmed by Figure 6-33 showing that gyro biases are always estimated. For the scale factors, they are only estimated if their corresponding gyros are excited. For this reason, the uncertainty of the x and y axes gyro scale factors converge much slowly than the scale factor on the z-axis, which is excited much more frequently than the other thanks to horizontal movement.

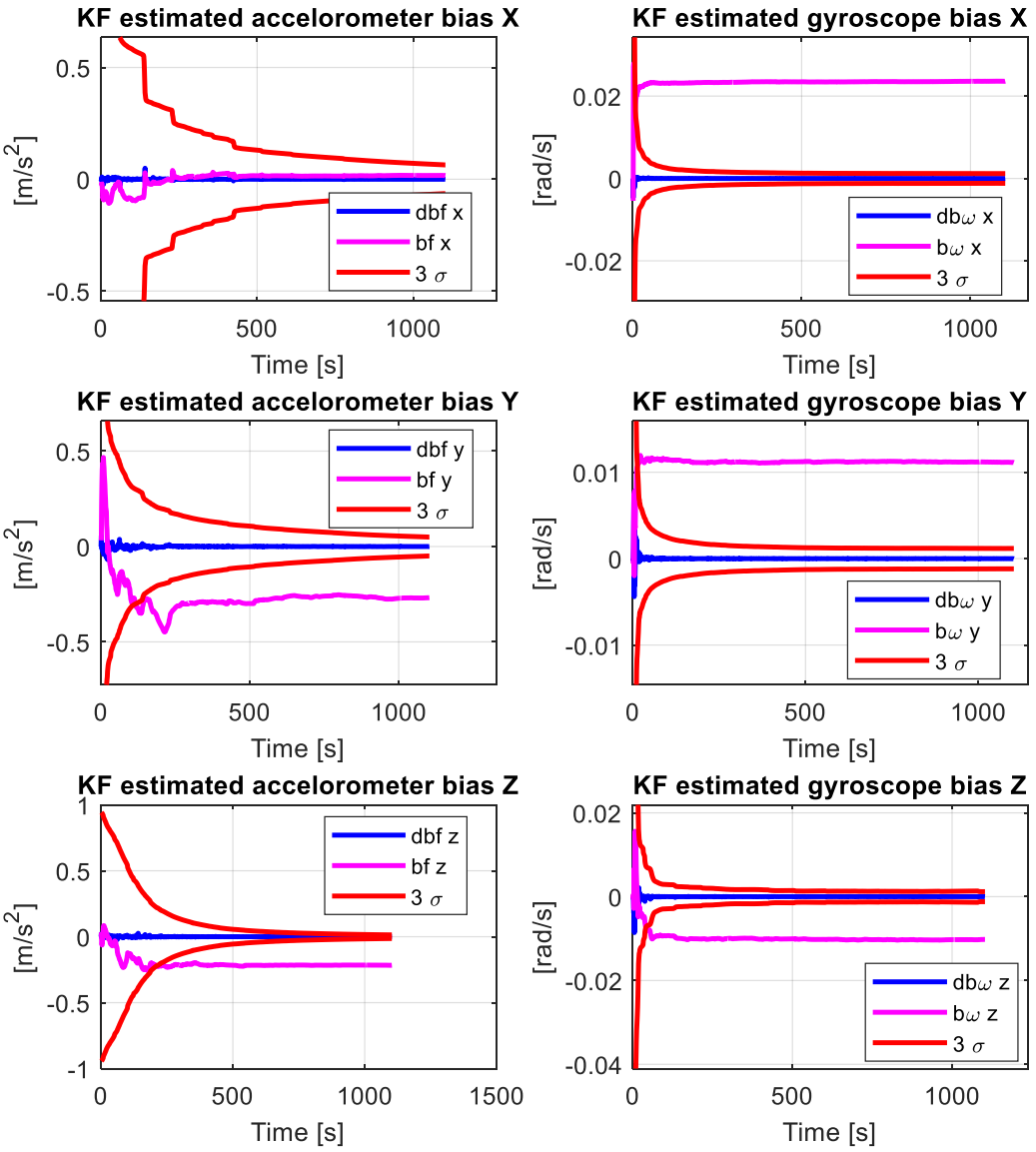


Figure 6-32 IMU accelerometer and gyroscope biases and their associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1)

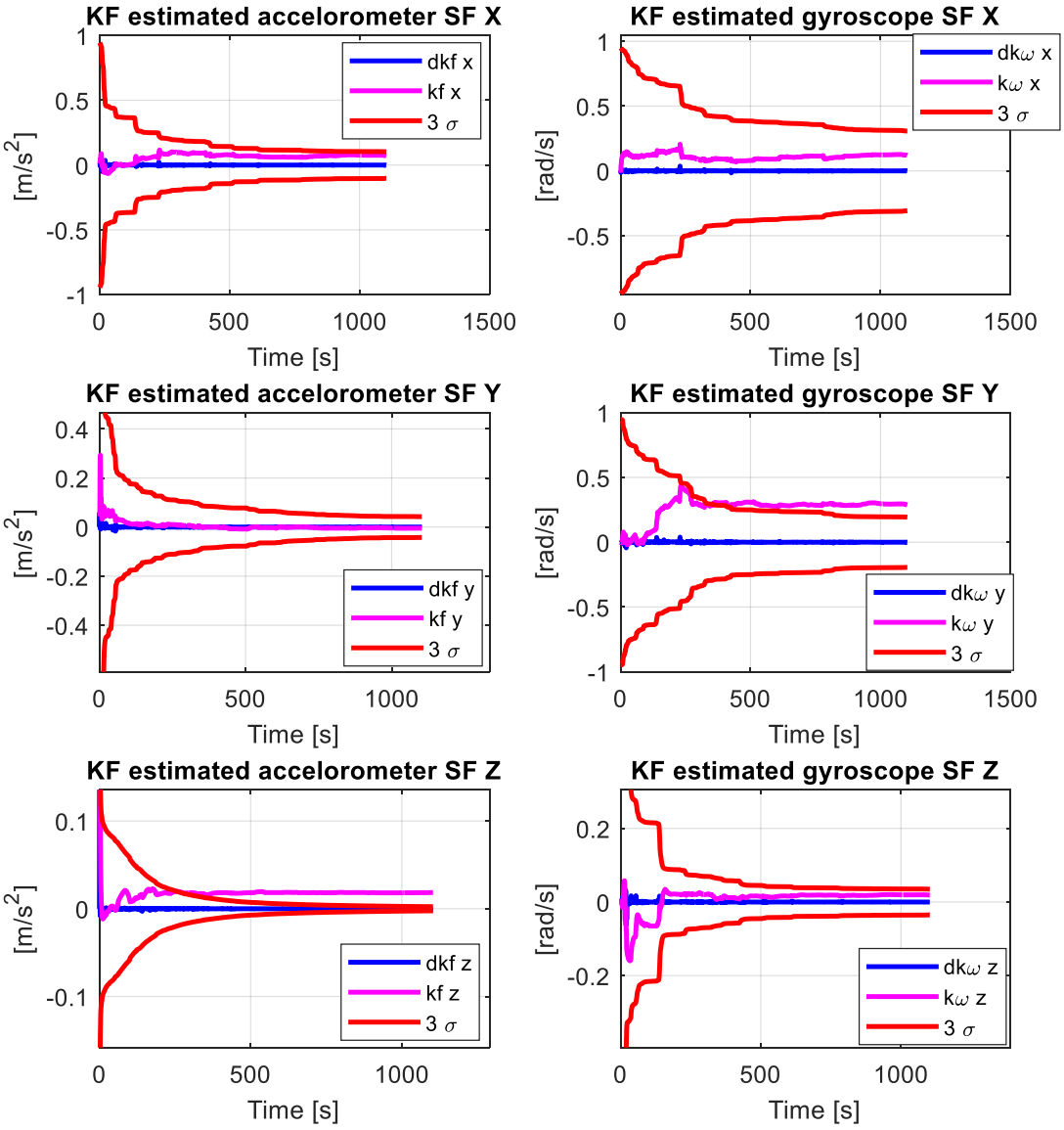


Figure 6-33 IMU accelerometer and gyroscope scale factors and their associated 3σ bound of the GPS/INS/WSS/NHC filter (Test 1)

6.2.3.2 Innovation test contribution

The innovation test is a simple filter test presented in Section 5.2.2 that consists of a consistency check all of the KF measurements with all the filter previous information. This test has been implemented on GNSS normalized innovation with a threshold of 3. The normalization consists of dividing the innovation by its standard deviation estimated by the KF. If the measurement innovation does not pass the test, then it is excluded. A single test is passed by each GNSS pseudorange and pseudorange rate measurement. The comparison between the horizontal position and velocity errors before and after the exclusion based on the innovation test is performed in Figure 6-34, in suburban-urban environment

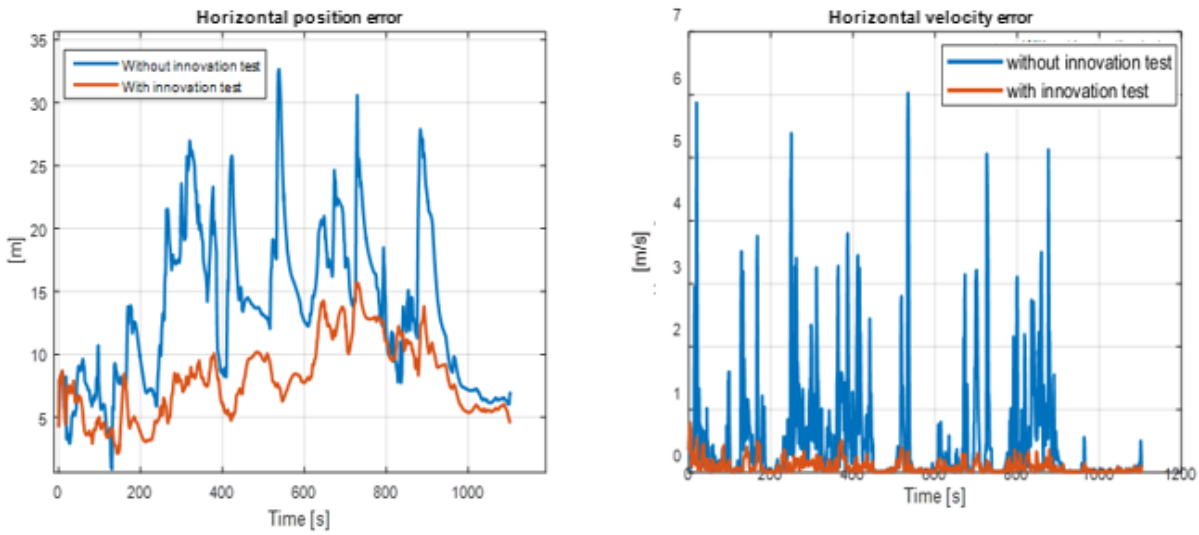


Figure 6-34 Comparison of the horizontal position error and the horizontal velocity error before and after innovation test using GPS/INS/WSS/NHC filter (Test 1)

This comparison reveals that the improvement brought by this simple test makes it a good and simple way to make the navigation solution more robust to faulty measurements by rejecting measurements that are likely to be degraded. It is possible that the pseudorange rate be rejected while its corresponding pseudorange is kept for the navigation solution estimation, but not the inverse because if a pseudorange is of bad quality than the corresponding pseudorange rate is automatically bad. Note that a condition is put on the pseudorange measurement exclusion limiting the number of excluded measurement such as the number of remaining satellites does not go below 4. If the number of identified pseudoranges to be excluded makes this number below 4, then the measurements to be excluded are classified based on their quality and the measurements with better quality than the others are kept to have exactly 4 satellites.

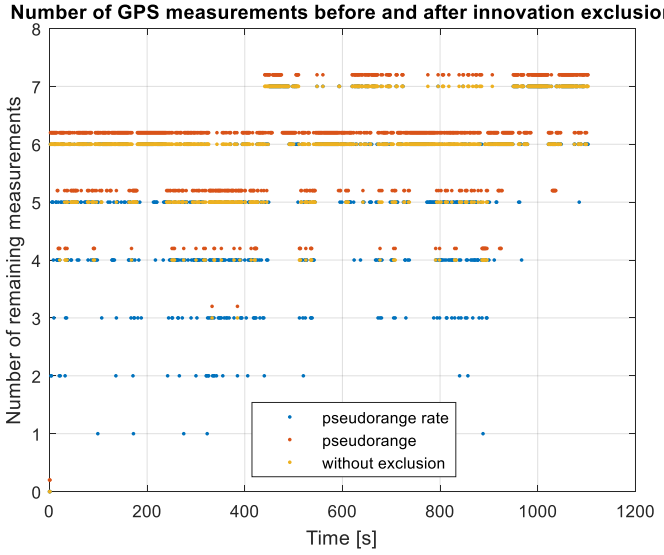


Figure 6-35 Number of GPS measurements before and after innovation test exclusion

Figure 6-35 illustrates the number of all the satellites in view and the number of remaining pseudorange and pseudorange rate. The pseudoranges were shifted by 0.2 to be able to distinguish the number of remaining satellites. We clearly notice on this figure that the pseudorange rate measurements are more excluded than the pseudoranges, as explained previously.

Table 6.8 gives the horizontal performance in terms of maximum, RMS, 95th percentile and 99th percentile before and after exclusion. In addition to the Figure 6-34, this table emphasizes the improvement brought by the innovation test.

	Horizontal position error (m)		**Horizontal Velocity error (m/s)**		**Heading error (°)**	
	Before exclusion	**After exclusion**	**Before exclusion**	**After exclusion**	**Before exclusion**	**After exclusion**
Max	32.63	15.71	6.02	0.79	24.79	4.93
RMS	14.83	8.53	0.99	0.14	0.98	0.14
95th percentile	24.84	13.18	2.35	0.30	11.00	1.98
99th percentile	28.15	14.95	3.96	0.46	19.11	3.68

Table 6.8 Horizontal accuracy performance before and after innovation test using GPS/INS/WSS/NHC filter (Test 1)

The results are only illustrated for Test 1 but the same conclusions are applied for Test 2. Note that the number of satellites excluded is lower in Test 2 thanks to the better quality of measurements in airport area.

6.2.3.3 SLAM contribution

To evaluate SLAM contribution to the hybridized navigation solution, two scenarios are considered:

- Scenario 1: the GPS is always available to study the SLAM contribution in presence of GPS with VSLAM loop-closure-based adjusted map.
- Scenario 2: a GPS outage is manually introduced to highlight the contribution of using VSLAM loop-closure-based adjusted map, when GPS (which is the only absolute sensor) is unavailable.

Therefore, seven configurations are considered:

- Config 1: GPS
- Config 2: GPS/INS/IT (innovation test)
- Config 3: GPS/INS/WSS/NHC/IT
- Config 4: GPS/INS/WSS/NHC/IT/VSLAM
- Config 5: GPS/INS/IT with 300s GPS outage
- Config 6: GPS/INS/WSS/NHC/IT with 300s GPS outage
- Config 7: GPS/INS/WSS/NHC/IT/VSLAM with 300s GPS outage

6.2.3.3.1. SLAM contribution in presence of GPS

Figure 6-36 illustrates the horizontal position and velocity errors obtained by Config 4 that considers all the available aiding measurements. First of all, it is important to mention that the

filter should run and converge before the VSLAM starts, in order to provide good information to estimate the SLAM transformation, as well as the lever arm between the IMU and the camera. Otherwise, these parameters will be badly estimated and the VSLAM innovation measurement quality depending on these parameters will be degraded, resulting in the filter navigation solution degradation. In addition, VSLAM parameters should be initialized by computing the transformation between the SLAM vision frame and the local frame from the first SLAM points. Note that it is mandatory to check the velocity of the vehicle and if it does not move, then wait until the vehicle start moving to launch the initialization. If these parameters are not initialized, the filter converges nonetheless to the values of these parameters, but the convergence takes a lot of time and degrades the navigation solution meanwhile.

Compared to Figure 6-29, Figure 6-36 shows that the use of SLAM in the presence of GPS does not improve the navigation solution, although a negligible decrease in the 3σ bound has been noticed. A possible explanation to this is the fact that the SLAM parameters are estimated thanks to the other filter sensors. These parameters are then utilized to create the SLAM innovation measurement by computing the difference between the SLAM actual output in the vision frame and the IMU prediction transformed in the vision frame; see Equation (4.35).

In addition, the position error exceeds the 3σ bound in some points. This may be due to the bad SLAM modeling. In fact, Kalman filters are not optimal in the case of non-Gaussian measurements, which is the case of SLAM based on the analysis of ATE. This may also be due to the uncertainty provided by the SLAM performance analysis in Section 6.2.2 that underestimates its error.

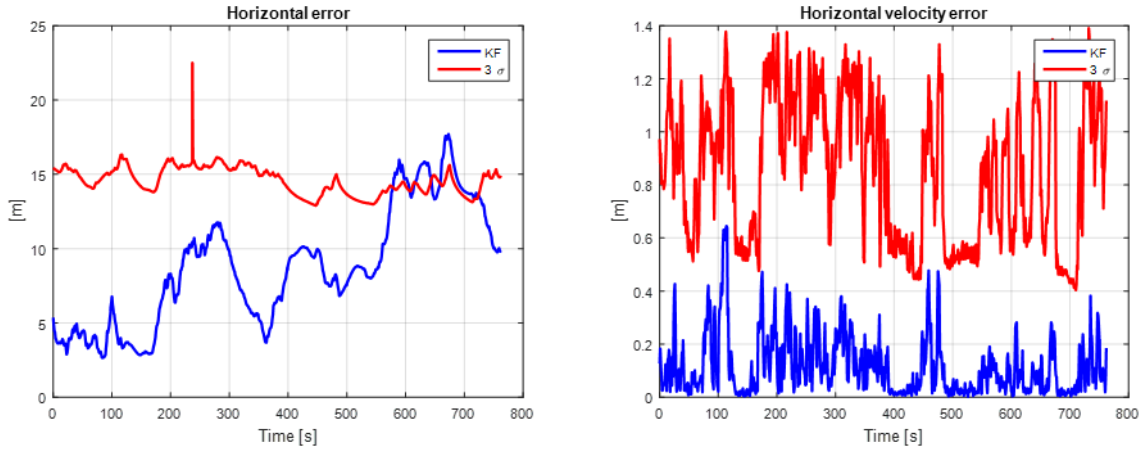


Figure 6-36 Horizontal position and velocity errors and their associated 3σ bound of the GPS/INS/WSS/NHC/IT/VSLAM filter (Test 1)

Figure 6-37 illustrates the SLAM transformation estimation by the filter. These parameters have approximately the same values found with Horn method in Section 6.2.2. The camera-IMU alignment denoted $\boldsymbol{q}_{b2c}$ is also estimated by the Kalman as illustrated in Figure 6-37.

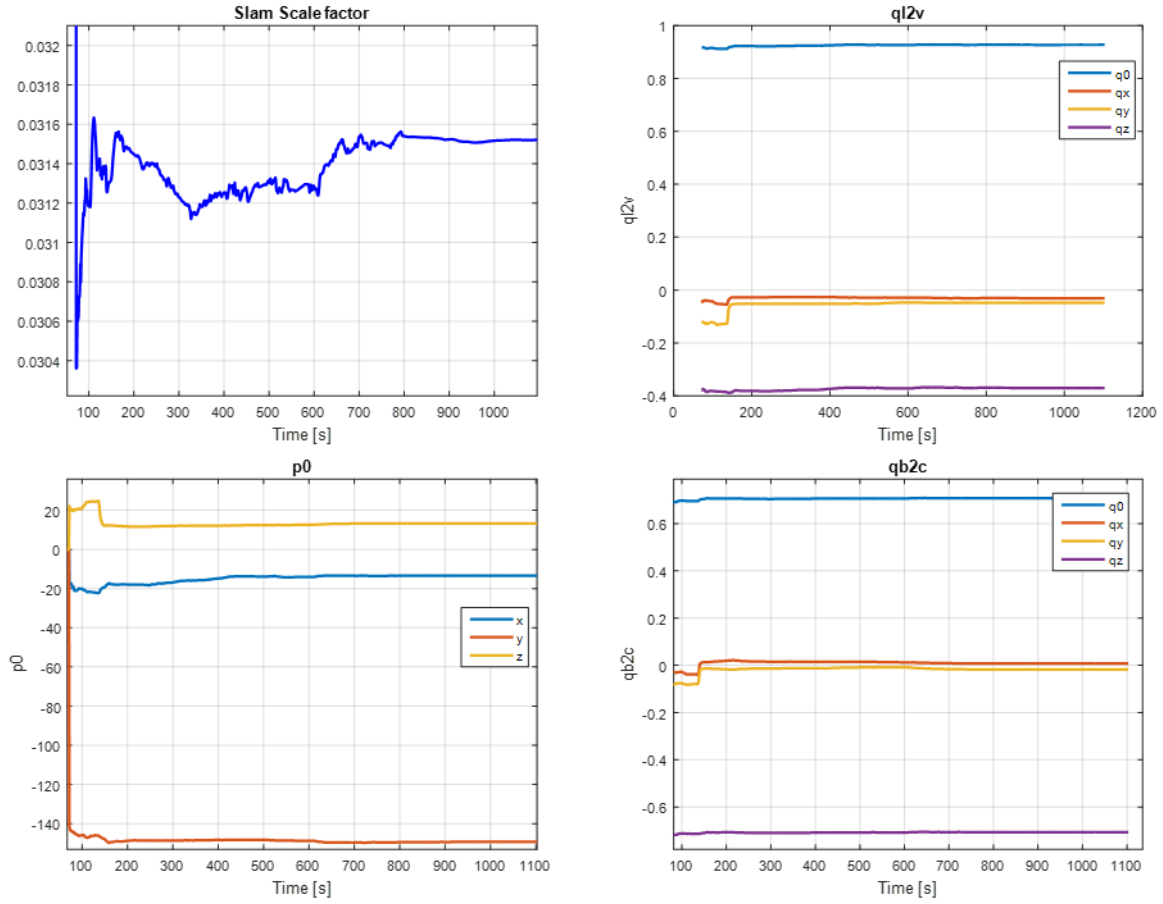


Figure 6-37 ORB-SLAM transformation parameters estimated by the GPS/INS/WSS/NHC/IT/VSLAM filter (Test 1)

5.2.3.3.2. SLAM contribution in GPS outage

Figure 6-38 shows Config 7 where 300 s of GNSS outage have been introduced while using all the other measurements. First, we notice the increase in the error 3σ bound because of GPS outage. This increase is due to the fact that GPS, which is the only global sensor, does not provide any information. In parallel, we notice that the error drifts even in SLAM presence. This is due to the fact that in GPS absence, the SLAM transformation parameters are propagated using its model in the Kalman filter (constant model). This choice of not updating the SLAM transformation parameters with the INS/WSS/NHC mode is justified by the drift of these sensors in GPS outage and by the slowly changing behavior of these parameters. However, due to the small drift remaining even when we use a map adjusted with loop closure, these parameters change in reality making drift the navigation solution using SLAM in GPS outage. If the map was correctly adjusted to completely remove the drift of these parameters, then these parameters would be estimated before GPS outage and the navigation would continue with the parameters that would not change. This would make SLAM like a global sensor thanks to the estimation of its parameters, resulting in navigation performance in GPS outage similar to the case of GPS presence. This suggests reviewing more deeply the loop closure process of SLAM and improving it.

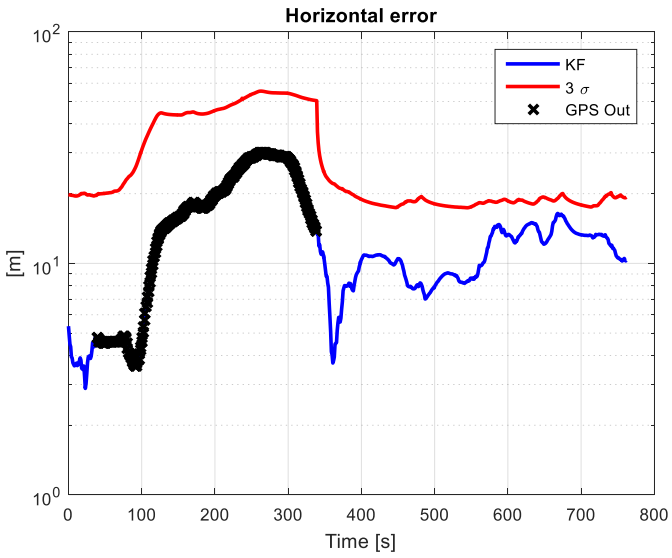


Figure 6-38 Horizontal position error and its associated 3σ bound of the GPS/INS/WSS/NHC/IT/VSLAM filter with 300 s GPS outage (Test 1)

The accuracy performance of the navigation solution is summarized for all the cases of Test1 in Figure 6-39, Figure 6-40 and in Table 6.9 to Table 6.14. We can see that sensor fusion improves dramatically the navigation solution accuracy with respect to standalone GPS. Each time a sensor or constraint is added, the navigation accuracy is improved. As for SLAM use, as it is defined in our proposed architecture, it does not contribute to the improvement of the navigation solution because of the existence of the unknown transformation between the vison frame and the local frame. However, the use of SLAM improves the navigation accuracy in case of GPS outage, even if the achieved performance is still not satisfactory.

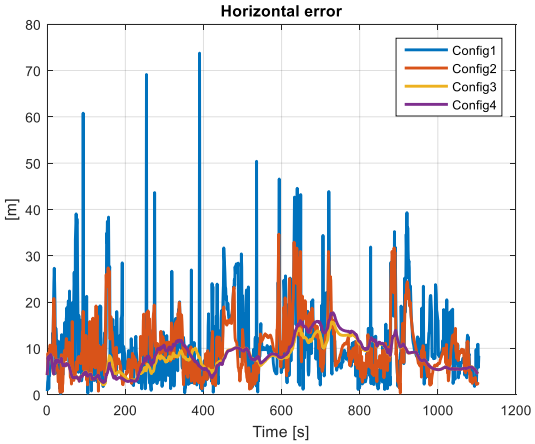


Figure 6-39 Comparison of the different configuration horizontal position error without GPS outage (Test 1)

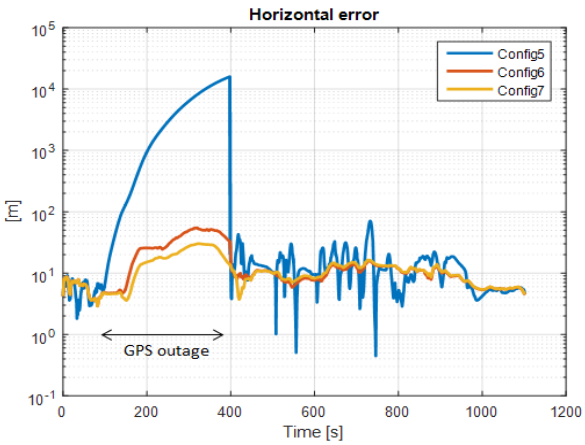


Figure 6-40 Comparison of the different configuration horizontal position error with 300 s GPS outage (Test 1)

	Horizontal position error (m)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS	73.75	15.92	30.42	55.57
GPS/INS/IT	34.65	11.51	21.99	29.74
GPS/INS/WSS/NHC/IT	15.72	8.54	13.19	14.96
GPS/INS/WSS/NHC/IT/SLAM	17.70	9.57	15.57	17.35

Table 6.9 Comparison of the different configuration horizontal position accuracy with no GPS outage (Test 1)

	Horizontal position drift after 300s of GPS outage (m)
GPS/INS/IT	15 967.85
GPS/INS/WSS/NHC/IT	54.92
GPS/INS/WSS/NHC/IT/SLAM	30.21

Table 6.10 Comparison of the different configuration horizontal position drift after 300s of GPS outage (Test 1)

	Horizontal Velocity error (m/s)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS	36.64	2.49	4.39	8.79
GPS/INS/IT	3.21	0.74	1.77	2.53
GPS/INS/WSS/NHC/IT	0.79	0.14	0.31	0.47
GPS/INS/WSS/NHC/IT/SLAM	0.64	0.16	0.35	0.56

Table 6.11 Comparison of the different configuration horizontal velocity accuracy with no GPS outage (Test 1)

	Horizontal velocity drift after 300s of GPS outage (m/s)
GPS/INS/IT	117.81
GPS/INS/WSS/NHC/IT	3.97
GPS/INS/WSS/NHC/IT/SLAM	2.03

Table 6.12 Comparison of the different configuration horizontal velocity drift after 300s of GPS outage (Test 1)

	Heading error (°)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS/INS/IT	25.04	5.21	6.64	19.23
GPS/INS/WSS/NHC/IT	4.94	1.50	1.98	3.68
GPS/INS/WSS/NHC/IT/SLAM	4.68	1.84	3.49	4.56

Table 6.13 Comparison of the different configuration heading accuracy with no GPS outage (Test 1)

	Heading drift after 300s of GPS outage (°)
GPS/INS/IT	153.71
GPS/INS/WSS/NHC/IT	5.58
GPS/INS/WSS/NHC/IT/SLAM	2.66

Table 6.14 Comparison of the different configuration heading drift after 300s of GPS outage (Test 1)

The same conclusions about the filter behavior apply for Test 2. The results of Test 2 are shown in Figure 6-41, Figure 6-42 and in Table 6.15 to Table 6.20. The results show that the addition of INS, WSS and the motion constraints to the GPS solution has improved the accuracy of the navigation solution near buildings and apron. However, when we get far from buildings, the

fusion of GPS with these sensors degrades its performance. This is due to the overestimation of the GPS uncertainly mentioned in Section 6.2.1 using the GPS model described in Section 2.1.2**Erreur ! Source du renvoi introuvable.**. This uncertainty overestimation makes the filter less confident in GPS measurements than it should be, although they are of good quality.

As for the SLAM, its contribution is negligible in presence of GPS. In the absence of GPS, the use of SLAM as it is defined in the architecture proposed in Chapter 4, does not improve neither the navigation performance. This is due to the badly textured airport environment and to the SLAM trajectory drift.

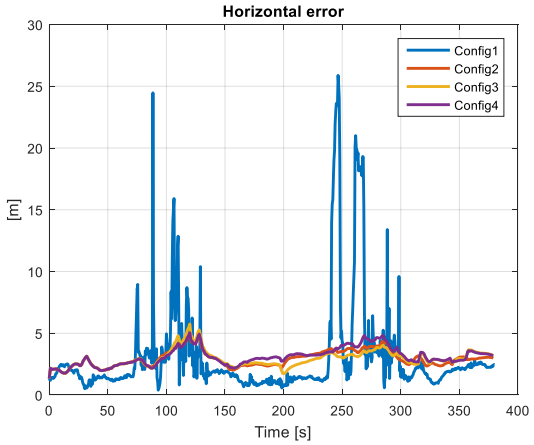


Figure 6-41 Comparison of the different configuration horizontal position error without GPS outage (Test 2)

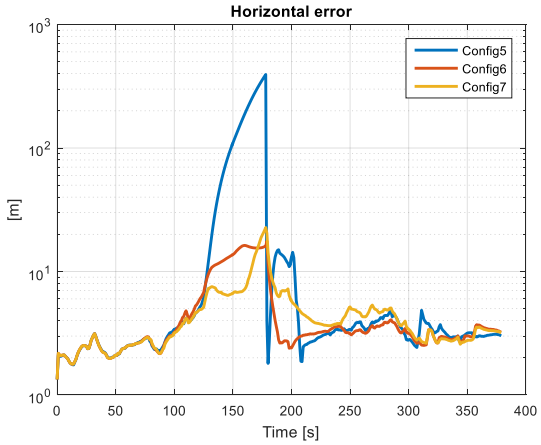


Figure 6-42 Comparison of the different configuration horizontal position error with 60 s GPS outage (Test 2)

	Horizontal position error (m)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS	9.60	2.81	4.92	6.87
GPS/INS/IT	4.94	3.13	4.22	4.64
GPS/INS/WSS/NHC/IT	5.75	3.13	4.41	5.26
GPS/INS/WSS/NHC/IT/SLAM	5.04	3.34	4.59	4.82

Table 6.15 Comparison of the different configuration horizontal position accuracy with no GPS outage (Test 2)

	Horizontal position drift after 60s of GPS outage (m)
GPS/INS/IT	393.93
GPS/INS/WSS/NHC/IT	17.63
GPS/INS/WSS/NHC/IT/SLAM	22.70

Table 6.16 Comparison of the different configuration horizontal position drift after 60s of GPS outage (Test 2)

	Horizontal Velocity error (m/s)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS	2.61	0.24	0.44	1.04
GPS/INS/IT	0.83	0.15	0.32	0.53
GPS/INS/WSS/NHC/IT	0.81	0.17	0.37	0.55
GPS/INS/WSS/NHC/IT/SLAM	0.77	0.16	0.34	0.51

Table 6.17 Comparison of the different configuration horizontal velocity accuracy with no GPS outage (Test 2)

	Horizontal velocity drift after 60s of GPS outage (m/s)
GPS/INS/IT	13.37
GPS/INS/WSS/NHC/IT	0.91
GPS/INS/WSS/NHC/IT/SLAM	1.27

Table 6.18 Comparison of the different configuration horizontal velocity drift after 60s of GPS outage (Test 2)

	Heading error (°)			
	Max	**RMS**	**95th percentile**	**99th percentile**
GPS/INS/IT	15.11	11.33	8.37	13.73
GPS/INS/WSS/NHC/IT	12.03	6.35	5.74	7.99
GPS/INS/WSS/NHC/IT/SLAM	12.01	6.42	5.67	7.93

Table 6.19 Comparison of the different configuration heading accuracy with no GPS outage (Test 2)

	Heading drift after 60s of GPS outage (°)
GPS/INS/IT	80.28
GPS/INS/WSS/NHC/IT	12.18
GPS/INS/WSS/NHC/IT/SLAM	11.78

Table 6.20 Comparison of the different configuration heading drift after 60s of GPS outage (Test 2)

6.3 Conclusion

This chapter presented the accuracy performance analysis of the navigation solution estimated using the hybridization architecture presented in Chapter 4. It described the data collection and the tests used to assess the performance of the navigation solution.

In the first part GPS solution is evaluated. It has been shown based on the received signal C/N_0 that the measurements are degraded with multipath and probably NLOS. The exclusion of measurements based on a C/N_0 mask is proven to be not efficient especially in urban environments where the satellite number is already small. The exclusion degrades the satellite geometry which degrades in turn the navigation solution. However, in airport environment, this exclusion has shown to be useful and is therefore applied in such environment.

In the second part, ORB-SLAM position output was analyzed on a previously explored environment where the feature map has been constructed, optimized thanks to loop closure and saved in the SLAM vision frame. This choice is based on the fact that if the map is not used, then the SLAM drifts and does not help to achieve good accuracy performance. Two tests have been considered. The first one is a trajectory in urban environment and the second one in the airport next to its building. First, the transformation allowing aligning the SLAM frame with

the reference local frame was estimated. Then, an estimation of SLAM position accuracy is given based on the computation of this transformation. This accuracy is used as the SLAM position uncertainty input to the Kalman filter. Note that SLAM accuracy in urban environment (textured) is better than in airport environment (non-textured). This is due to the higher number of features, their better location in terms of proximity and spatial distribution. However, in both cases, the accuracy (of few meters) is not satisfactory.

In the third part, the performance of the hybridized solution is assessed. First, the integrated GPS/INS/WSS/NHC before SLAM integration has been studied. The validation of the measurement models is performed, and the theoretical state observability conclusions are confirmed. Then, the impact of the use of innovation testing and SLAM were discussed. This performance analysis has shown that the innovation test that excludes the GPS measurements that are likely to be degraded because of multipath and NLOS, improves dramatically the navigation solution estimation especially in urban environments. However, the SLAM, as it is used, does not improve the navigation solution performance when GPS is available. However, in case of GPS outage, SLAM can improve the navigation performance if the 7 parameters describing the transformation between the vision frame and the local frame do not change in time. This can be done by improving the SLAM loop closure process to achieve a complete drift removal.

Chapter 7

Conclusions and Proposals on Further Research

The current chapter has a first part which reminds the conclusions of the previous chapters and concludes the work done during this Ph.D. thesis. It also reminds the major contributions of the thesis. In a second part, perspectives of future works that could be carried out so as to continue the work done during this Ph.D. thesis are addressed.

7.1 Thesis conclusions

It is important to remind that this Ph.D. thesis focused on designing and assessing the performances of a hybridization filter integrating low-cost sensors to achieve a high-performance level in terms of accuracy, continuity and integrity, while having a reasonable cost and taking into account the constraints related to the targeted applications.

As a first step, the state of the art of sensors used in literature, their characteristics, their prices and their performance has been reviewed to identify the most adapted sensors to our case of study, taking into account different aspects such as the low-cost constraint and the characteristics of the targeted application (outdoor environment, ground vehicles, etc.). This state of the art has led to the identification of the sensors to be used in the multi-sensor fusion architecture. These sensors consist of a low-cost GNSS receiver, a MEMS IMU, four-wheel speed sensors and visual information provided by an independent processing module which converts the visual information into motion information before its integration in the multi-sensor architecture. This approach of considering the visual processing module as a black box has the advantage of having a multi-sensor system that is independent from the technique used by the visual module. This allows having a modular system capable of continuously providing a navigation solution even if the visual process experiences failures either coming from bad visibility conditions (lack of light, fog, rain, intense luminosity variation, etc.), or an unfavorable environment for visual information processing (dynamic environment, badly textured environment, etc.).

After having identified the sensors that would be part of the hybridization architecture, these sensors have been studied in depth in Chapter 2 and Chapter 3. A review of their technical fundamentals, principles and measurement characteristics has been performed. In particular, we focused on vision-based positioning to identify the techniques and the type of measurement that would be used in the multi-sensor architecture, and to understand the characteristics and challenges of this type of information. First, VO and SLAM techniques have been addressed and compared. SLAM has been identified as the best technique capable of providing navigation

information with the best accuracy performance. Indeed, unlike VO based on simple DR to provide camera motion, SLAM provides the camera pose after being adjusted by several optimization processes. For this reason, this information has been identified as the visual information to be used in the multi-sensor architecture. Second, monocular vision has been preferred to stereovision because of the need of having a large baseline in stereovision to cope with the far objects in large-scale environments, which is not feasible for ground vehicles. Finally, the challenges of these choices are discussed. In particular, issues related to the unknown transformation between the frame in which SLAM outputs its pose and the local well-known frame are highlighted. This transformation includes an unknown scale factor resulting from monocular vision, an unknown rotation and offset between the vision frame and the local frame resulting from the SLAM initialization step.

After having studied sensors and detailed their advantages and their challenges, the state of the art of hybridization techniques has been reviewed, and a comparison between the different strategies has been carried out. This study has led to identify the error-state EKF as the most adapted fusion algorithm. In fact, despite the EKF limits consisting mainly of the linearization process, the low dynamics in the intended applications and the error state configuration make the effect of linearization negligible. In addition, this filter is efficient in terms of computational cost, especially for applications where there are a large number of states and observations, compared to the other techniques.

Based on the previous considerations, the architecture of the multi-sensor fusion system has been proposed in Chapter 4. The IMU is selected as the reference sensor since it is the only sensor continuously and autonomously providing a complete navigation solution at the highest rate, independently of the surrounding environment. The other sensors consisting of a GNSS receiver providing position and velocity information or pseudorange and pseudorange rate depending on the application, WSSs providing velocity on each wheel, and position and attitude output by a SLAM algorithm, are used to correct the IMU navigation solution. This correction is done in closed loop to calibrate the high errors of the bad quality IMU. Motion constraints related to ground movement are also taken into account to improve the navigation solution estimation.

Based on this architecture, the states to be estimated by the filter have been identified for each sensor and gathered in the state vector. The mathematical models defining the Kalman time propagation process and measurement update are detailed for all the sensors. An observability study has been carried out given the high number of states to be estimated by the filter. This study has confirmed the observability of all the states using the proposed architecture if some motion conditions are respected.

The proposed architecture is then evaluated from an integrity point of view in Chapter 5. The challenges related to this topic have been discussed. These challenges are divided into sensor challenges, filter challenges and requirement challenges. Sensor challenges are mainly related to SLAM and GNSS whose errors are environment-dependent. To address this challenge, SLAM errors have been identified and classified into fault-free mode and failure modes and the GNSS failure modes are recalled. Furthermore, the integrity monitoring principle through the

FD/FDE function has been explained and the state of the art of the existing FD/FDE techniques adapted to our case of study is reviewed. This analysis has led to several conclusions regarding the integrity challenges resulting from the presence of SLAM and GNSS. For GNSS, these challenges are mainly related to signal vulnerability to reflections and interferences. As for SLAM, the main challenge consists of characterizing its error and particularly the possible failure modes. This requires having evaluated SLAM on sufficiently various and long data collections and identified all the possible failures based on these collections. In addition, the fact that SLAM error is environment-dependent and that SLAM includes very complex processes, make the mapping of the measurement error (pixel uncertainty) into the navigation solution domain complicated to perform. This prevents the direct exploitation of the classical GNSS signal integrity methods in the SLAM context, and requires thinking about a different approach to handle the SLAM integrity issue.

Filter integrity challenges are mainly due to the single failure assumption which is not adapted to our case of study where the proposed system uses different measurements. In addition, the computational complexity of the existing integrity algorithms makes the real-time processing unfeasible. However, since the proposed architecture takes advantage of information redundancy provided by different sensors, consistency check of these measurements can be simply performed by testing the filter innovation. This method, though simple, has proven to be efficient to detect measurements that are likely to be faulty and exclude them.

The third integrity challenge consists of having no requirements for road applications in terms of integrity risk, time to alert and HAL. Despite that, a simple example based on the HAL definition is proposed in this thesis and has shown that the HAL should be approximately equal to $1.5m$ to avoid collision between vehicles. If we assume the fault free mode, this requires an accuracy of about $40\ cm$, which is a challenging task at present.

Taking into account all the challenges discussed above, the proposed multi-sensor filter has been implemented and evaluated on real data. The ORB-SLAM algorithm is used to output the camera pose. Two tests have been considered. The first one is a trajectory in urban environment and the second one in the airport next to its building.

First the GPS solution is evaluated to highlight the difficulties encountered by GPS in urban environment. It has been shown based on the received signal C/N_0 that the measurements have degraded quality, resulting from multipath and NLOS. The exclusion of measurements based on a C/N_0 mask is proven to be not efficient especially in urban environments where the satellite number is already small. This is due to the degradation of satellite geometry because of exclusion which degrades the navigation solution.

Second, ORB-SLAM position output was analyzed on a previously explored environment where the feature map has been constructed, adjusted and saved in the SLAM vision frame. This choice was based on the fact that if the map is not used, then the SLAM drifts and does not help to achieve good accuracy performance. First, the transformation allowing aligning the SLAM frame with the reference local frame was estimated. Then, an estimation of SLAM position accuracy is given based on the ATE. It has shown that SLAM accuracy in urban textured environment is better than in airport poorly-textured environment. This is due to the

higher number of features, their better location in terms of proximity and spatial distribution. However, in both cases, the accuracy (of few meters) is not satisfactory. The analysis of the SLAM trajectory has shown that the use of a previously adjusted map with loop closure reduces considerably SLAM drift but does not remove it completely.

Then the performance of the hybridized solution is assessed. First, the integrated GPS/INS/WSS/NHC without SLAM has been studied. The validation of the measurement models is performed and the theoretical state observability conclusions are confirmed using the behavior of the 3σ bound output by the filter.

Then, the impact of the use of innovation testing and SLAM were discussed. This performance analysis has shown that the innovation test excluding the GPS measurements that are likely to be degraded by multipath and NLOS, improves dramatically the navigation solution estimation especially in urban environments. However, the SLAM, as it is used, does not improve the navigation solution performance when GPS is available. However, in case of GPS outage, SLAM can improve the navigation performance if the 7 parameters describing the transformation between the vision frame and the local frame do not change in time. This can be achieved by improving the SLAM loop closure process to achieve a complete drift removal.

7.2 Perspective and future works

Further studies can be conducted in the following of this thesis on several axes in order to improve the performance of the proposed architecture:

SLAM improvement

- We have seen in this thesis when studying the SLAM performance that a drift remains in the adjusted map using loop closure. This drift is reflected on the SLAM pose estimation and degrades the quality of SLAM. Note that this problem has not been noticed when processing SLAM in small loop closures (loop closure of few hundreds of meters) and it may be due to the long trajectory we have travelled (few kilometers). It is therefore important to revise the loop closure process especially in the case of large loop closure.

- SLAM can also be improved by modifying the constant velocity model used to predict the vehicle motion. This model can be violated in case of high dynamics resulting in the loss of map track. This problem has been experienced in the airport environments where the car has accelerated. To improve this model, it is recommended to use wheel odometry or inertial navigation to predict vehicle dynamics.

- In badly textured environments such as airport surface, SLAM is likely to loose the map frequently especially in turns. To reduce map tracking loss, the camera field-of-view may be augmented to 180° or even to 360°. This requires the adaptation of the SLAM algorithm to large field-of-view lens, to take into account the correct projection model

- One of the most limiting issues in SLAM, as it is defined in the proposed architecture, is the unknown transformation between the vision frame in which SLAM provides its

outputs and the local frame (or any other known frame). In fact, although we have used the previously adjusted map in our study, this map is defined in the unknown vision frame transferring this unknown transformation to the pose estimation. We can therefore think about geo-referencing the map after its adjustment and before using it to perform the positioning task. The geo-referencing process should be based on a very precise reference system to reduce the SALM error. In that way, SLAM can become an absolute sensor being able to provide the pose directly in the absolute frame. Using this method, we can achieve the sub-metric level of accuracy

SLAM Integrity challenges

- The most challenging integrity task for SLAM is to characterize its error. This task requires testing SLAM on a large amount of data to identify the nominal error statistical distribution and to characterize the failure modes by their probability of occurrence, the correlation between multiple failures (simple failure, multiple failures) and their signatures (amplitude, time duration, shape).

Integration strategy improvement

- The assumption that may be reviewed in this thesis is the incoherence between the EKF requiring Gaussian measurements and the SLAM pose estimation which is not Gaussian. This problem may be handled in the future by choosing one of the following options:
    - the use of a particle filter to hybridize the non-Gaussian SLAM pose measurement with the other sensors. In fact, this filter gives the possibility to overcome the constraint of Gaussianity generally needed for KFs to provide an optimal estimation.
    - the use of the relative information (displacement and angular change instead of position and attitude). This information should be studied in order to be characterized. If it is Gaussian, then, its integration in a Kalman filter may be performed using stochastic cloning described in (Roumeliotis, 2002)

Bibliography

(Angrisano, 2010) Angrisano, A., *GNSS/INS integration methods*, Phd thesis, University of Naples Parthenope, 2010

(Bahrami, 2010) Bahrami, M., and M. Ziebart, *Instantaneous Doppler-Aided RTK Positioning with Single Frequency Receivers*, Proc. IEEE/ION PLANS 2010, Indian Wells, CA, May 2010, pp. 70-78

(Bailey, 2006) Bailey, T., and Durrant-Whyte, H., *Simultaneous localization and mapping: part II*, IEEE Robotics and Automation Magazine, 2006, pp. 108-117

(Barron, 1992) Barron, J.L., Fleet, D.J., Beauchemin, S.S., and Burkitt, T.A., *Performance of optical flow techniques*, Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 1992, pp. 236-242

(Bay, 2008) Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., *SURF: Speeded-Up Robust Features*, Computer Vision and Image Understanding, 2008, 110, (3), pp. 346-359

(Ben-Afia, 2014) Ben-Afia, A., Deambrogio, L., Salos, D., Escher, A.C., Macabiau, C. Soulier, L., and Gay-Bellile, V., *Review and classification of vision-based localisation techniques in unknown environment*, IET Radar, Sonar & Navigation, Volume 8, Issue 9, December 2014, pp. 1059-1072

(BGLE–MIMOSA, 2012) BGLE-MIMOSA, Annexe Technique, 2012

(Bhatti, 2007) Bhatti, U., and Ochieng, W., *Failure Modes and Models for Integrated GPS/INS Systems*, Journal of Navigation, 2007, 60(2), pp. 327-348

(Bhatti, 2008) Bhatti, U., Okou, A.F., and A. Noureldin. *An integrated reduced inertial sensor system - riss / gps for land vehicle*, In Position, Location and Navigation Symposium, IEEE/ION, May 2008, pp. 1014-1021

(BinAhmad, 2015) Bin Ahmad, K., *Reliability Monitoring of GNSS Aided Positioning for Land Vehicle Applications in Urban Environments*, Phd thesis, Université de Toulouse (ISAE), 2015

(Blanco, 2010) Blanco, J.L., *A tutorial on SE(3) transformation parameterization sand on-manifold optimization,* Technical report, University of Malaga, 2010

(Bouguet, 2015) Bouguet, J., *Camera calibration toolbox for Matlab*. Available on http://www.vision.caltech.edu/bouguetj/calib.doc/index.html

(Brenner, 1995) Brenner, M.A., *Integrated GPS/Inertial Fault Detection Availability*, Proc. ION-GPS-95, 8th International Technical Meeting of the Satellite Division of the Institute of Navigation, Palm Springs, CA, 1995.

(Brenner, 1998) Brenner, M., Reuter, R., and Schipper, B., *GPS Landing System Multipath Evaluation Techniques and Results*, Proc. ION GPS-98, Nashville, Tennessee, September 1998

(Brocard, 2016) Brocard, P., *Integrity Monitoring for Mobile Users in Urban Environment*, Phd thesis, Université de Toulouse, 2016

(Brown, 1997) Brown, R.G., and Hwang, P.Y.C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, 1997

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

(Campbell, 2005) Campbell, J., Sukthankar, R., Nourbakhsh, I., and Pahwa, A., *A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision*, Proc. IEEE Int. Conf. Robotics and Automation, 2005, pp. 3421-3427

(Carcanague, 2011) Carcanague, S., Julien, O., Vigneau, W., and Macabiau, C, *A new algorithm for GNSS precise positioning in constrained area*, Proc. International Technical Meeting of The Institute of Navigation, San Diego, CA, January 2011, pp. 24-26

(Carcanague, 2013) Carcanague, S., *Low-cost GPS/GLONASS Precise Positioning Algorithm in Constrained Environment*, Phd thesis, Université de Toulouse, 2013

(Choi, 2009) Choi, S., Kim, T., and Yu, W., *Performance Evaluation of RANSAC Family*, Proc. British Machine Vision Conference, 2009

(Chrony, 2016) *Chrony Manual*, Available on https://chrony.tuxfamily.org/manual.html

(Chu, 2012) Chu, T., Guo, N., Backén, S., and Akos, D., *Monocular Camera/IMU/GNSS Integration for Ground Vehicle Navigation in Challenging GNSS Environments*, Sensors, 2012, 12, (3), pp. 3162-3185

(Collins, 1999) Collins, J.P., *Assessment and Development of a Tropospheric Delay Model for Aircraft Users of the Global Positioning System*, Technical Report N° 203, September 1999

(Concha, 2015) Concha, A., and Civera, J., *DPPTAM: Dense Piecewise Planar Tracking and Mapping from a Monocular Sequence*, Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS), 2015

(Conley, 2006) Conley, R. Cosentino, R., Hegarty, C.J., Kaplan, E.D., Leva, J.L., Uijt de Haag, M., and Van Dyke, K., *Performance of Stand-Alone GPS*, In: *Understanding GPS: Principles and Applications*, 2nd edition, Chapter 7, Artech House Mobile Communications Series, 2006

(Conte, 2006) Conte, G., Moog, C. H., and Perdon, A. M., *Algebraic Methods for Nonlinear Control Systems*, 2nd edition, Springer, 2006

(DeCleene, 2000) De Cleene, B., *Defining Pseudorange Integrity - Overbounding*, Proc. ION GPS 2000, September 2000, pp. 19-22

(Diesel, 1996) Diesel, J., and Dunn, G., *GPS/IRS AIME: Certification for Sole Means and Solution to RF Interference*, Proc. of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996), Kansas City, MO, September 1996, pp. 1599-1606

(Dissanayake, 2001) Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., and Csorba, M., *A Solution to the Simultaneous Localisation and Map-Building Problem*, IEEE Trans. Robotic and Automation, 2001, 17, (3), pp. 229-241

(DoD GPS SPS, 2001) Department of Defense USA, *Global Positioning System Standard Positioning Service Performance Standard*, 3rd edition, October 2001

(DoD GPS SPS, 2008) Department of Defense USA, *Global Positioning Service Standard Positioning Service Performance Standard*, 4th edition, September 2008

(Durrant-Whyte, 2006) Durrant-Whyte, H., and Bailey, T., *Simultaneous localization and mapping: part I*. IEEE Robotics and Automation Magazine, 2006, pp 99-108

(El-Sheimy, 2008) El-Sheimy, N., Hou, H., and Niu, X., *Analysis and modeling of inertial sensors using Allan variance*, IEEE Transactions on Instrumentation and Measurement, 2008, 57(1), pp. 140-149

(Engles, 2006) Engles, C., Stewénius, H., and Nistér, D., *Bundle adjustment rules*, Proc. Symp. Photogrammetric Computer Vision, 2006

(Engel, 2012) Engel, J., Sturm, J., and Cremers, D., *Camera-based navigation of a low-cost quadrocopter*, Proc. Of the Internatioanl Conf. on Intelligent Robot Systems, 2012

(Engel, 2014) Engel, J., Schops, T., and Cremers, D., LSD-SLAM: *Large-scale direct monocular SLAM*, Proc. Of European Conference on Computer Vision (ECCV), Zurich, Switzerland, September 2014, pp. 834-849.

(EU-US CSN, 2010) EU-US Cooperation on Satellite Navigation, *Combined Performances For Open GPS/GALILEO Receivers*, Working Group C, July, 2010

(Farrell, 1999) Farrell, J.A, T*he Global Positioning System & Inertial Navigation*, McGraw Hill Professional, 21 Jan 1999 - Technology & Engineering

(Filip, 2006) Filip, A., Polivka, A., and Suchánek, J., *Practical Analysis of Satellite Navigation Performance for Train Control*. Proc. Of the 7th World Congress on Railway Research, Montréal (WCRR), 2006, Canada

(Forster, 2014) Forster, C. Pizzoli, M., and Scaramuzza, D., *SVO: Fast semi-direct monocular visual odometry*, Proc of the IEEE International Conf. on Robotics and Automation, Hong Kong, China, June 2014, pp. 15-22.

(Gao, 2007) Gao, J., *Development of a Precise GPS/INS/On-Board Vehicle Sensors Integrated Vehicular Positioning System*, Phd thesis, The University of Calgary, Calgary, Alberta, June 2007.

(Gao, 2015) Gao, Y., Liu, S., Atia, M.M., and Noureldin, A., *INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm*, Sensors, 2015, vol. 15, no. 9, pp. 23286-23302,

(Gauglitz, 2011) Gauglitz, S., Höllerer, T., and Turk, M., *Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking*, International Journal of Computer vision, 2011, 94, (3), pp. 335-360

(Giremus, 2007) Giremus, A., and Escher, A.C., *A GLR algorithm to detect and exclude up to two simultaneous range failures in a GPS/Galileo/IRS case*, Proc. of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation, Sep 2007, Fort Worth, USA, pp. 2911-2923

(Goshen-Meskin, 1992) Goshen-Meskin, D., and Bar-Itzhack, I. Y., *Observability analysis of piece-wise constant systems – part I: Theory*, IEEE Transactions on Aerospace and Electronic Systems 28(4), pp. 1056-1067

(Groves, 2008) Groves, P.D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, 2008

(GSA, 2015) GSA, *Report on the Performance and level of Integrity for Safety and Liability Critical Multi-Applications*, Technical Report, May, 2015

(Guilloton, 2011) Guilloton, A., Arethens, J.P., Macabiau, C., Escher, A.C., and Koenig, D., *A Methodology to Elaborate Aircraft Localization Requirements for Airport Navigation*, Proc. of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011, pp. 523-535

(Gustafsson, 2002) Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P., *Particle filters for positioning, navigation, and tracking*, IEEE Trans. Signal Process, 2002, 50, pp. 425-437

(Harris, 1988) Harris, C., and Stephens, M., *A Combined Corner and Edge Detector*, Proc. Alvey Vision Conference, 1988, pp. 147-151

(Hartley, 2004) Hartley, R., and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, March 2004

(Hartmann, 2013) Hartmann, J., Klussendorff, J.H., and Maehle, E., *A comparison of feature descriptors for visual SLAM*, IEEE European Conference on Mobile Robots (ECMR), 2013, pp. 56-61

(Hermann, 1977) Hermann, R., and Krener, A.J., *Nonlinear controllability and observability*, IEEE Transactions on Automatic Control, 1977, 22(5), pp. 728-740

(Hide, 2007) Hide, C., Moore, T., and Hill, C., *A Multi-Sensor Navigation Filter for High Accuracy Positioning in All Environments*, The Journal of Navigation, 2007, 60(3), pp. 409-425

(Hietanen, 2016) Hietanen, A., Lankinen, J., Kämäräinen, J.K., Buch, A.G., and Krüger, N., *A comparison of feature detectors and descriptors for object class matching*, Neurocomputing, 2016, 184, pp. 3-12

(Hiremath, 2014) Hiremath, S.A., Van Der Heijden, G.W., Van Evert, F.K., Stein, A., and Ter Braak, C.J., *Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter*, Computers and Electronics in Agriculture, 2014, 100, pp. 41-50

(Hoffmann-Wellenhof, 1992) Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J., *Global Positioning System: Theory and Practice*, Springer New York, 1992

(Hong, 2012) Hong, S., Song, J.B., Baek, J.H., and Ryu, J.K., *Visual odometry for outdoor environment using a downward-tilting camera and self-shadow removal algorithm*, Proc. IEEE internatioanl Conf. Control, Automation and Systems (ICCAS), 2012, pp. 960-963

(Horn, 1987) Horn, B., *Closed-form solution of absolute orientation using unit quaternions,* Journal of the Optical Society of America A, vol. 4, pp. 629-642

(Hwang, 2011) Hwang S.Y., and Song, J.B., *Monocular vision-based SLAM in indoor environment using corner, lamp, and door features from upward-looking camera*, IEEE Trans. Ind. Electron., 2011, 58, (10), pp. 4804 -4812

(ICAO, 2004) ICAO, *Manual on advanced surface movement guidance and control systems (A-SMGCS)*, Doc 9830 AN/452, 1st edition, 2004

(ICAO, 2006) ICAO, *Annex 10 to the Convention on International Civil Aviation, Aeronautical Telecommunications*, Volume I Radio Navigation Aids, 6th edition, July 2006

(Jahn, 1996) Jahn, A., Bischl, H., and Heib, G., *Channel Characterization for Spread Spectrum Satellite Communication*, IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings, Volume 3, September 22-25, 1996, pp. 1221-1226

(Jiang, 2012) Jiang, Z., and Groves, P.D., *NLOS GPS signal detection using a dual polarisation antenna*, GPS solutions, 2012

(Joerger, 2013) Joerger, M., and Pervan, B., *Kalman Filter-Based Integrity Monitoring Against Sensor Faults*, AIAA Journal of Guidance, Control and Dynamics, Vol. 36, No. 2, March–April 2013, pp. 349-361

(Julien, 2011) Olivier Julien, *Concepts avancés du GNSS*, 3rd Year IENAC Course, 2011

(Julier, 1995) Julier, S.J., Durrant-Whyte, H.F., *Navigation and Parameter Estimation of High Speed Road Vehicles*, Proc. Robotics and Automation Conference, Nagoya, Japan, pp. 101-105

(Julier, 1997) Julier, S.J., and Uhlmann, J., *A new extension of the Kalman filter to nonlinear systems*. Proc. International Symposium on Aerospace/Defense Sensing, Simulation and Controls, 1997, pp. 182-193

(Kalman, 1960) Kalman, R.E., *A new approach to linear filtering and prediction problems*, ASME Transactions, Series D, Journal of Basic Engineering, 1960, 82, pp. 35-45

(Kaplan, 2005) Kaplan, E.D., and Hegarty, C., *Understanding GPS: Principles and Applications*, 2nd edition, Artech House, November 2005

(Kelly, 2010) Kelly, J., and Sukhatme, G., *Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration*, International Journal of Robotics Research, Oct. 2010

(Klein, 2007) Klein, G., and Murray, D., *Parallel tracking and mapping for small AR workspaces*, Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07), Nara, Japan, November 2007.

(Klobuchar, 1987) Klobuchar J.A., *Ionospheric Time-Dealy Algorithm for Single-Frequency GPS Users*; IEEE Transactions on Aerospace and Electronic Systems, vol AES-23, no.3, 1987

(Lee, 1982) Lee, T. S., Dunn, K. P., and Chang, C. B., *On the observability and unbiased estimation of nonlinear systems*, System Modeling and Optimization: Proceedings of the 10th IFIP, Vol. 38/1982 of Lecture Notes in Control and Information Sciences, Springer, pp. 258-266

(Lee, 1999) Lee, Y.C., Laughlin, D.G., *A performance analysis of a tightly coupled GPS/inertial system for two integrity monitoring method*, Proc. of ION GPS, Nashville 14-17 September, pp. 1187-2000

(Li, 2010) Li, T., Petovello, M.G., Lachapelle, G., and Basnayake, C., *Ultra-tightly Coupled GPS/Vehicle Sensor Integration for Land Vehicle Navigation*, Journal of Institute of Navigation, 2010, 54(4), pp. 263-274

(Lowe, 2004) Lowe, D.G., *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, 2004, 60, (2), pp. 91-110

(Martin, 2013) Martin H.W., *U.S. Space-Based positioning, Navigation and Timing Policy and Program Update*, 8th International Committee on GNSS, Dubai, 2013

(Martineau, 2008) Martineau, A., *Performance of Receiver Autonomous Integrity Monitoring (RAIM) for Vertically Guided Approaches*, PhD thesis, Université de Toulouse, 2008

(Martinelli, 2011) Martinelli, A., Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale and Bias Determination, IEEE Transactions on Robotics, 2011, vol. 28(1), pp. 44-60

(Maybeck, 1979) Maybeck, P. S., *Stochastic Models, Estimation and Control*, Vol. 141-1 of Mathematics in Science and Engineering, Academic Press, 1979

(Mezentsev, 2005) Mezentsev, O., *Sensor Aiding of HSGPS Pedestrian Navigation*, PhD Thesis, published as UCGE Report No. 20212, Department of Geomatics Engineering, University of Calgary, Canada, 2005

(Michot, 2010) Michot, J., Bartoli, A., and Gaspard, F., *Bi-objective Bundle Adjustment with Application to Multi-sensor SLAM*, In 3DPVT, 2010

(Milner, 2009) Milner, C., *Determination of the effects of GPS failures on aviation applications*, PhD thesis, Centre for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London, 2009

(Montloin, 2014) Montloin, L., *Impact of GNSS Singular Events on the Integrity of Airport Navigation Systems,* PhD thesis, Université de Toulouse, 2014

(Moravec, 1980) Moravec, H., *Obstacle avoidance and navigation in the real world by a seeing robot rover*, PhD thesis, Stanford University, Stanford, CA, 1980

(Moreno-Noguer, 2007) Moreno-Noguer, F., Lepetit, V., and Fua, P., *Accurate non-iterative O(n) solution to the PnP problem*, Proc. IEEE International Conf. Computer Vision, 2007, pp. 1-8

(Mur-Artal, 2014) Mur-Artal, R., and Tardos, J.D., *Fast relocalisation and loop closing in keyframe-based SLAM*, in IEEE Internatioanl Conf. on Robotics and Automation (ICRA), 2014, pp. 846-853

(Mur-Artal, 2015) Mur-Artal, R., Montiel, J.M.M. and Tardos, J.D., *ORB-SLAM: A versatile and accurate monocular SLAM system*. IEEE Transactions on Robotics, pp. 1-17

(Neri, 2011) Neri, P., *Use of GNSS signals and their augmentations for Civil Aviation navigation during Approaches with Vertical Guidance and Precision Approaches,* PhD thesis, Université de Toulouse, 2011

(Nistér, 2004a) Nistér, D., *An Efficient Solution to the Five-point Relative Pose Problem*, IEEE Trans. Pattern Analysis and Machine Intelligence, 2004, 26, (6), pp. 756-770

(Nistér, 2004b) Nistér, D., Naroditsky, O., and Bergen, J., *Visual odometry*, Proc. International Conf. Computer Vision and Pattern Recognition, 2004, pp. 652-659.

(Nistér, 2006) Nistér, D., Naroditsky, O., and Bergen, J.: *Visual odometry for ground vehicle applications*', Journal of Field Robotics, 2006, 23, (1), pp. 3-20

(Nordh, 2007) Nordh, J., *Ultrasound-based navigation for mobile robots*, Master's thesis ISRN LUTFD2/TFRT- -5789- -SE, Department of Automatic Control, Lund University, Sweden, 2007

(Nourani-Vatani, 2011) Nourani-Vatani, N., Vincius, P., and Borges, K., *Correlation-based Visual Odometry for Ground Vehicles*, Journal of Field Robotics, 2011, 28, (5), pp. 742-768

(Nützi, 2010) Nützi, G., Weiss, S., Scaramuzza, D., and Siegwart, R., *Fusion of IMU and vision for absolute scale estimation in monocular SLAM*, Proc. of the International Conference on Unmanned Aerial Vehicles, Dubai, 2010

(Okuda, 2014) Okuda, R., Yuki K., and Kazuaki T., *A survey of technical trend of ADAS and autonomous driving*, Proc. IEEE Technical Program-2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA), 2014, pp. 1-4

(OpenCV, 2014) *OpenCV Camera calibration function*. Available on http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

(Parkinson, 1988) Parkinson, B.W., and Axelrad, P., *Autonomous GPS Integrity Monitoring Using the Pseudorange Residual*, Journal of the Institute of Navigation, Vol. 35 N°. 2, Summer 1988, pp. 255-274

(Parkinson, 1996) Parkinson, B.W., *GPS Error Analysis*, In: Parkinson, B.W., and Spilker, J.J.Jr., *Global Positioning System: Theory And Applications*, vol. 1, chapter 11. American Institute of Aeronautics and Astronautics, Inc., Washington D.C., USA, 1996

(Quigley, 2009) Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and A. Ng, *ROS: An open-source robot operating system*, Proc. Open-Source Software Workshop International Conf. Robotics and Automation, Kobe, Japan, 2009

(Raquet, 2003) Raquet, J.F., Giebner, M., *Navigation Using Optical Measurements of Objects at Unknown Locations*, Proc. Annual Meeting of ION and CIGTF 22nd Guidance Test Symposium, 2003, pp. 282-290

(Rhee, 2002) Rhee, I., Abdel-Hafez, M.F., and Speyer, J.L., *On the observability of an integrated GPS/INS during manoeuvres*, Proc. ION GPS, Portland, OR, 2002

(Rife, 2009) Rife, J., and Pullen, S., *Aviation Applications*. In: Gleason, S., and Gebre-Egziabher, D., *GNSS Applications and Methods*, Artech House, Boston, MA,2009, pp. 245-267

(Rosten, 2006) Rosten, E., and Drummond, T., *Machine learning for high-speed corner detection*, Proc. 9th European Conference on Computer Vision (ECCV'06), Graz, May 2006.

(Roumeliotis, 1999) Roumeliotis, S.I., Sukhatme, G.S., and Bekey, G.A., *Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization*, Proc. of IEEE International Conference on Robotics and Automation, Detroit, MI, May 10-15 1999, vol. 2, pp. 1656-1663.

(Roumeliotis, 2002) Roumeliotis S.I., and J. Burdick, J., *Stochastic cloning: a generalized framework for processing relative state measurements*, Proc. of IEEE International Conference on Robotics and Automation, vol. 2, pp. 1788-1795

(RTCA, 2009) RTCA DO 316, *Minimum Operational Performance Standards (MOPS) For Global Positioning System/aircraft Based Augmentation System Airborne Equipment*, 2009

(Rublee, 2011) Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., *ORB: an efficient alternative to SIFT or SURF,* Proc. of the IEEE International Conf. on Computer Vision (ICCV), volume 13, 2011

(Scaramuzza, 2006) Scaramuzza, D., Martinelli, A., and Siegwart, R., *A flexible technique for accurate omnidirectional camera calibration and structure from motion*, Proc. IEEE Int. Conf. Computer Vision Systems (ICVS) 2006, Jan. 2006, pp. 45-53.

(Scaramuzza, 2011) Scaramuzza, D., and Fraundorfer, F., *Visual Odometry Part I: The First 30 Years and Fundamentals*, IEEE Trans. Robotics and Automation Magazine , 2011, 18, (4), pp. 80-92

(Schuster, 2011) Schuster, W., and Ochieng, W., *Airport Surface Movement – Critical Analysis of Navigation System Performance Requirements*, The Journal of Navigation (2011), 64, pp. 281-294

(Shi, 1994) Shi, J., and Tomasi, C., *Good Features to track*, Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 1994, pp. 593-600

(Shin, 2001) Shin, E.H., *Accuracy Improvement of low Cost INS/GPS for Land Applications*, Master's thesis, Department of Geomatics Engineering, University of Calgary, 2001

(Siegwart, 2011) Siegwart, R., Nourbakhsh, I.R., and Scaramuzza, D., *Introduction to Autonomous Mobile Robots*, 2nd edition, 2011, Massachusetts Institute of Technology

(Skog, 2009) Skog, I., Handel, P., *In-Car positioning Navigation and Technologies – A survey*, IEEE Trans. Intelligent Transportation Systems, 2009, 10, (1), pp. 4-21

(Sola, 2015) Sola, J., *Quaternion kinematics for the error-state KF,* Institut de Robotica i Informatica Industrial, Barcelona, Tech. Rep., 2015

(Spangenberg, 2007) Spangenberg, M., Calmettes, V., Kubrak, D., and Julien, O., *Optimized low-cost HSGPS/IMU/WSS land vehicle navigation system for urban navigation*, Proc. of the ION ITM Meeting, 2007, Fort Worth Texas, USA, pp. 70-78

(Spilker, 1996) Spilker, J.J.Jr., *Tropospheric Effects on GPS*, In: Parkinson, B.W., and Spilker, J.J.Jr., *Global Positioning System: Theory And Applications*, vol. 1, chapter 13. American Institute of Aeronautics and Astronautics, Inc., Washington D.C., USA, 1996

(Strasdat, 2012) Strasdat, H., Montiel, J.M.M, and Davison, A.J., *Visual SLAM: Why Filter?*, Image and Vision Computing, 2012, 30, (2), pp. 65-77

(Sturm, 2012) Sturm, J., Engelhard, N., Endres, F., Burgrad, W., and Cremers, D., *A benchmark for the evaluation of RGB-D slam systems*, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and System, New York,2012

(Sukkarieh, 1999) Sukkarieh, S., Nebot, E.M., and Durrant-Whyte, H.F., *A high integrity IMU/GPS navigation loop for autonomous land vehicle applications*, IEEE Transactions on Robotics and Automation, 1999, 15(3), pp. 572-578

(Sukkarieh, 2000) Sukkarieh. S, *Low Cost High Integrity Aided Inertial Navigation Systems for Autonomous Land Vehicles*, PhD thesis, The University of Sydney, March 20, 2000

(Syed, 2007) Syed, Z., Aggarwal, P., Niu, X., El-Sheimy, N., *Economical and Robust Inertial Sensor Configuration for a Portable Navigation System*, ION GNSS 2007, Fort Worth, Texas, pp. 24-25

(Trawny, 2005) Trawny, N. and Roumeliotis, S., *Indirect Kalman Filter for 3D Attitude Estimation*, Technical Report Number 2005-002, Department of Computer Science & Engineering, University of Minnesota. March, 2005

(Triggs, 1999) Triggs, B., McLauchlan, P.F., Hartley, R.I., and Fitzgibbon, A.W., *Bundle Adjustment - A Modern Synthesis*, Proc. International Workshop Vision Algorithms: Theory and Practice, Springer-Verlag, 1999

(Verhaegen, 2007) Verhaegen, M., and Verdult, V., *Filtering and System Identification: A Least Squares Approach*, Cambridge University Press, 2007

(Walter, 1995) Walter, T. and Enge, P., *Weighted RAIM for precision approach*, Proc. ION GPS 1995, Palm Springs, CA, Sept. 12-15 1995, pp. 1995-2004

(Wan, 2000) Wan, E. and Van der Merwe, R., *The unscented kalman filter for nonlinear estimation*, Proc. IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control, 2000, pp. 153-158

(Weiss, 2012) Weiss, S. M., *Vision based navigation for micro helicopters*, PhD thesis, Eidgenossische Technische Hochschule ETH Zurich, Nr. 20305, 2012

(Welch, 2006) Welch, G., and Bishop, G., *An introduction to the kalman filter*, Technical Report TR 95-041, University of North Carolina, Chapell Hill, July 2006.

(Willsky, 1976) Willsky, A.S., *A Generalized Likelihood Ratio approach to the detection of mean jumps in linear systems*, IEEE Transactions on Automatic Control, 1976, pp. 108-112

(Woodman, 2007) Woodman, O.J., *An introduction to inertial navigation*, Tech. Rep. UCAMCL-TR-696, University of Cambridge, 2007

(Younes, 2017) Younes, G., Asmar, D., and Shammas, E., *A survey on non-filter-based monocular Visual SLAM systems*, Design, Survey, and Future directions, Robotics and Autonomous Systems, 2017, 98, pp. 67-88

Appendix A

Coordinate frames and Transformations

In this appendix, the different coordinate frames used in this thesis and their corresponding transformations are detailed.

A.1 Coordinate frames

A.1.1 Earth-Centered-Inertial frame ($\boldsymbol{i}$)

An inertial frame is defined as a non-rotating and non-accelerating frame with respect to the fixed stars. The Earth-Centered-Inertial (ECI) frame used in inertial navigation is centered at the Earth's center of mass and oriented with respect to its spin axis and the stars. Consequently, the ECI is not strictly inertial as the Earth undergoes motion in its orbit around the sun. The ECI is defined by:

- Origin: Earth's Center of mass
- $\boldsymbol{z}^i$ axis: Earth's spin axis
- $\boldsymbol{x}^i$ and $\boldsymbol{y}^i$ axes: lie within the equatorial plane and do not rotate with the Earth. The $\boldsymbol{y}^i$ axis points 90° ahead the x^i axis in the direction of the Earth's rotation

A.1.2 Earth-Centered-Earth-Fixed frame ($\boldsymbol{e}$)

The Earth-Centered-Earth-Fixed frame is a frame remaining fixed with respect to the Earth. It is defined by:

- Origin: Earth's Center of mass
- $\boldsymbol{z}^e$ axis: Earth's spin axis
- $\boldsymbol{x}^e$ axis: points from the Earth's center of mass to the intersection of the equator with the Prime Meridian of Greenwich
- $\boldsymbol{y}^e$ axis: completes the right-handed frame.

A.1.3 The East-North-Up local Tangent-Plane frame ($\boldsymbol{l}$)

The East-North-Up (ENU) Tangent-Plane frame is a local frame with a fixed origin with respect to Earth (usually a point on the Earth surface close to the navigation area). It is defined by:

- Origin: a fixed origin with respect to Earth
- $\boldsymbol{z}^l$ or **U** axis: points upwards along the WGS84 ellipsoidal normal
- $\boldsymbol{x}^l$ or **E** axis: points towards the geodetic East
- $\boldsymbol{y}^l$ or **N** axis: points towards the geodetic North.

The ECI, ECEF and ENU frames are shown in Figure A-1.

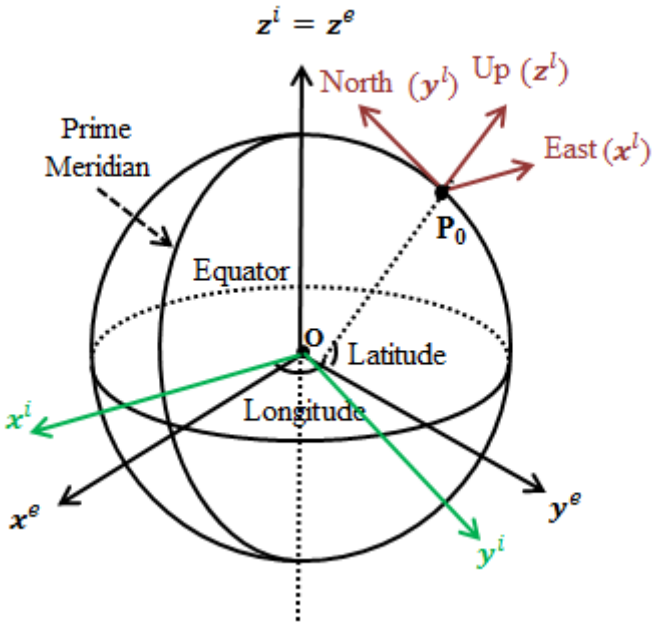


Figure A-1 ECI, ECEF and ENU frames

A.1.4 The body frame (b)

The body frame, sometimes known as the vehicle frame, describes the vehicle position and orientation given by the navigation solution. In the strapdown configuration where the IMU is rigidly mounted on the vehicle to be positioned, this frame describes also the IMU axes. The body frame is defined by:

- Origin: vehicle's Center of mass
- $\boldsymbol{x}^b$ axis: points towards the right of the vehicle
- $\boldsymbol{y}^b$ axis: points towards the front of the vehicle
- $\boldsymbol{z}^b$ axis: completes the right-handed frame (points upwards)

The body frame is illustrated in Figure A-2.

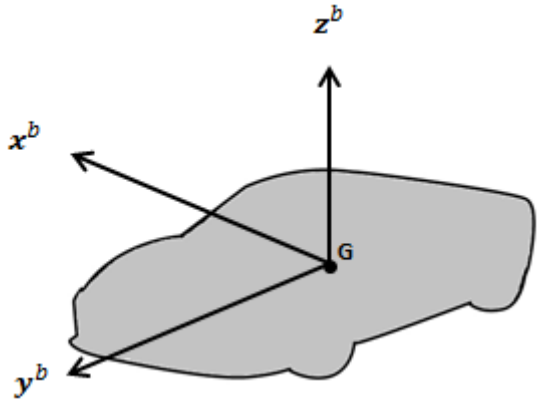


Figure A-2 Body frame

A.1.5 The image frame ($\boldsymbol{I}$)

The image frame is a 2D frame that describes the pixel coordinates of a point lying on the image plane. It is defined by:

- Origin: the top-left corner of the image
- $\boldsymbol{u}$ axis: lies in the image plane and points towards the right
- $\boldsymbol{v}$ axis: lies in the image plane and points downwards

A.1.6 The camera frame ($\boldsymbol{c}$)

The camera frame describes the camera position and orientation. It is defined by:

- Origin: the center of the camera lens (optical center)
- $\boldsymbol{z}^c$ axis: is perpendicular to the image plane and points towards the image plane
- $\boldsymbol{x}^c$ axis: parallel to the image u-axis and points towards the same direction
- $\boldsymbol{y}^c$ axis: parallel to the image v-axis and points towards the same direction

A.1.7 The vision frame ($\boldsymbol{v}$)

The vision frame, also called world frame by the computer vision and robotics communities, is a local frame fixed with respect to the Earth. It is defined by:

- Origin: a fixed point with respect to the Earth
- $\boldsymbol{x}^v$ axis: is an arbitrary axis fixed to the Earth
- $\boldsymbol{y}^v$ and $\boldsymbol{z}^v$ axes: complete the right-handed frame.

The image, camera and vision frames are illustrated in Figure A-3:

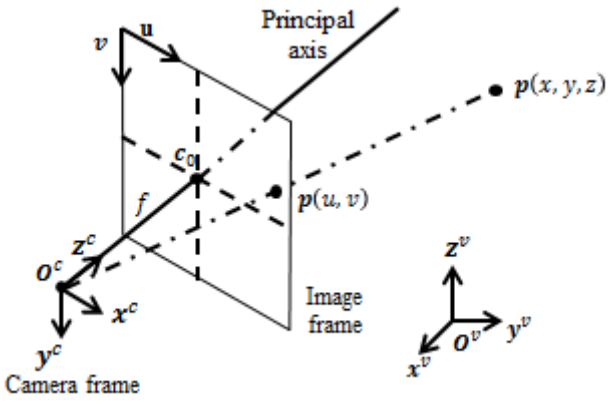


Figure A-3 Image, camera and vision frames

A.2 Frame Transformations

A.2.1 From ECI to ECEF frame

The transformation from ECI to ECEF frame corresponds to a single rotation around the z-axis, because both frames have the same origin and the z-axis. Denoting the Earth angular rate by ω_{ie}, the rotation from ECI to ECEF is given by:

$$\boldsymbol{C}_{i2e} = \begin{bmatrix} \cos(\omega_{ie}(k-k_0)) & \sin(\omega_{ie}(k-k_0)) & 0 \\ -\sin(\omega_{ie}(k-k_0)) & \cos(\omega_{ie}(k-k_0)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

where

- k is the current time epoch
- k_0 is the time epoch when the x and y-axes of both frames are coincident

Therefore, denoting $\boldsymbol{p}^i$ a point expressed in the ECI frame and $\boldsymbol{p}^e$ the same point expressed in the ECEF frame, we have:

$$\boldsymbol{p}^e = \boldsymbol{C}_{i2e}\boldsymbol{p}^i \tag{A.2}$$

A.2.2 From ECEF to ENU frame

The transformation from ECEF to ENU frame includes a rotation and a translation between the ECEF origin (Earth's center of mass) and the ENU origin (local point). The rotation is determined by the vehicle geodetic latitude φ and longitude λ as follows:

$$\boldsymbol{C}_{e2l} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\sin(\varphi)\cos(\lambda) & -\sin(\varphi)\sin(\lambda) & \cos(\varphi) \\ \cos(\varphi)\cos(\lambda) & \cos(\varphi)\sin(\lambda) & \sin(\varphi) \end{bmatrix} \tag{A.3}$$

Therefore, denoting $\boldsymbol{p}^e$ a point expressed in the ECEF frame and $\boldsymbol{p}^l$ the same point expressed in the ENU frame, we have:

$$\boldsymbol{p}^l = \boldsymbol{C}_{e2l}(\boldsymbol{p}^e - \boldsymbol{p}^e_{l0}) \tag{A.4}$$

where $\boldsymbol{p}^e_{l0}$ is the position of the ENU frame origin in the ECEF frame.

A.2.3 From ENU to body frame

The transformation from the ENU frame to the body frame is performed by three consecutive rotations as explained in Appendix B and a translation between the ENU origin (fixed with respect to the Earth) and the body origin (moving with respect to the Earth). The rotation $\boldsymbol{C}_{l2b}$ is given by equation (B.41). Therefore, denoting $\boldsymbol{p}^l$ a point expressed in the ENU frame and $\boldsymbol{p}^b$ the same point expressed in the ENU frame, we have:

$$\boldsymbol{p}^b = \boldsymbol{C}_{l2b}(\boldsymbol{p}^l - \boldsymbol{p}^l_{b0}) \tag{A.5}$$

where $\boldsymbol{p}_{b0}^{l}$ is the position of the vehicle center of mass in the local ENU frame.

A.2.4 From vision to camera frame

The transformation from the vision frame to the camera frame is described by a rotation $\boldsymbol{C}_{v2c}$ and a translation between the vision frame origin and the camera optical center $\boldsymbol{p}_{c0}^{v}$. $\boldsymbol{C}_{v2c}$ and $\boldsymbol{p}_{c0}^{v}$ represent respectively the camera orientation and position is the world frame. These parameters are called the camera extrinsic parameters. Therefore, denoting $\boldsymbol{p}^{v}$ a point expressed in the vision frame and $\boldsymbol{p}^{c}$ the same point expressed in the camera frame, we have:

$$\boldsymbol{p}^{c} = \boldsymbol{C}_{v2c}(\boldsymbol{p}^{v} - \boldsymbol{p}_{c0}^{v}) \tag{A.6}$$

A.2.5 From camera to image frame

The transformation from the camera frame to the image frame is described by two steps. The first step transforms a 3D point $\boldsymbol{p}^{c} = [x^{c} \quad y^{c} \quad z^{c}]^{T}$ expressed in the camera frame into a 2D point $\boldsymbol{p}^{Im}$ lying on the image plane. The second step converts the point $\boldsymbol{p}^{Im}$ expressed in metric unit to $\boldsymbol{p}^{Ip} = [u \quad v]^{T}$ expressed in pixel unit.

To perform this transformation, the points should be expressed in homogenous coordinates to avoid the non-linearity of the transformation:

$$k_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} k_u & s_\gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{2^{nd}step} \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{1^{st}step} \begin{bmatrix} x^{c} \\ y^{c} \\ z^{c} \\ 1 \end{bmatrix} \tag{A.7}$$

where

- k_c is a scale factor
- k_u and k_v represent the number of pixels per distance unit in the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions
- s_γ is the skew factor which equals zero if and only if the $\boldsymbol{u}$ and $\boldsymbol{v}$ directions are perfectly orthogonal
- $\boldsymbol{c_0} = [u_0 \quad v_0]$ is the principal point describing the intersection point between the optical axis and the image plane
- f is the focal length

The parameters k_u ,k_v, s_γ, u_0, v_0 and f are called the camera intrinsic parameters.

Taking into account the transformation from the vision frame to the camera frame and the transformation from the camera frame to the image frame, the projection of a point $\boldsymbol{p}^{v}$ on the image plane is given by the following expression:

$$s \begin{bmatrix} \boldsymbol{p}^{Ip} \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & s_\gamma & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_{v2c} & -\boldsymbol{C}_{v2c}\boldsymbol{p}_{c0}^{v} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{v} \\ 1 \end{bmatrix} \tag{A.8}$$

Appendix B

Rotation mathematics

In three-dimensional space, there are several ways to represent a rotation. The common way to describe a rotation is to use quaternions, rotation matrices and Euler angles.

B.1 Quaternions

B.1.1 Mathematical definition

A quaternion is a hyper-complex number defined with four components q_0, q_1, q_2 and q_3 by:

$$\boldsymbol{q} = q_0 + q_1 i + q_2 j + q_3 k \tag{B.1}$$

where $\boldsymbol{j}$ and $\boldsymbol{k}$ are the hyper-imaginary numbers satisfying the JPL convention (Trawny, 2005):

$$\begin{gathered} i^2 = -1\,, \qquad j^2 = -1\,, \qquad k^2 = -1 \\ ij = -ji = -k\,, \qquad jk = -kj = -i\,, \qquad ki = -ik = -j \end{gathered} \tag{B.2}$$

Note that there are additional conventions that are used such as the Hamilton convention (Sola, 2015). In land navigation, the Hamilton convention is the most widely used while the JPL convention is used in the aerospace field. However, it is more convenient to use the JPL convention to simplify the transformations between the quaternion and the rotation matrices. Further details will be given later about this choice.

The quantity q_0 is the real or scalar part of the quaternion and $q_1 i + q_2 j + q_3 k$ is the imaginary or vector part. Therefore, the quaternion can be written:

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \tag{B.3}$$

B.1.2 Physical definition

A quaternion is a representation of a rotation by a rotation angle α and a unit vector along the axis of rotation $\boldsymbol{u}$:

$$\boldsymbol{q} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) \boldsymbol{u} \end{bmatrix} = \left[\cos\left(\frac{\alpha}{2}\right) \quad \sin\left(\frac{\alpha}{2}\right) u_x \quad \sin\left(\frac{\alpha}{2}\right) u_y \quad \sin\left(\frac{\alpha}{2}\right) u_z\right]^T \tag{B.4}$$

This definition shows the following important property of the quaternion of rotation: A quaternion representing rotation is a unit vector:

$$\|\boldsymbol{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{B.5}$$

B.1.3 Quaternion properties

- **Multiplication**

The quaternion multiplication operator is denoted by $\otimes$. The product of two quaternions is given by:

$$\boldsymbol{q} \otimes \boldsymbol{p} = (q_0 + q_1 i + q_2 j + q_3 k)(p_0 + p_1 i + p_2 j + p_3 k) \tag{B.6}$$

Using the JPL convention defined in equation (B.2), we obtain:

$$\boldsymbol{q} \otimes \boldsymbol{p} = \begin{bmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + q_1 p_0 - q_2 p_3 + q_3 p_2 \\ q_0 p_2 + q_1 p_3 + q_2 p_0 - q_3 p_1 \\ q_0 p_3 - q_1 p_2 + q_2 p_1 + q_3 p_0 \end{bmatrix} \tag{B.7}$$

This can also be written in terms of scalar and vector parts as:

$$\boldsymbol{q} \otimes \boldsymbol{p} = \begin{bmatrix} q_0 p_0 - \boldsymbol{q}_v^T \boldsymbol{p}_v \\ q_0 \boldsymbol{p}_v + p_0 \boldsymbol{q}_v - \boldsymbol{q}_v \times \boldsymbol{p}_v \end{bmatrix} \tag{B.8}$$

This equation shows that the quaternion product is not commutative. However, the quaternion product is associative and distributive over the sum:

$$\begin{gathered} \boldsymbol{q} \otimes \boldsymbol{p} \neq \boldsymbol{p} \otimes \boldsymbol{q} \\ (\boldsymbol{q} \otimes \boldsymbol{p}) \otimes \boldsymbol{r} = \boldsymbol{q} \otimes (\boldsymbol{p} \otimes \boldsymbol{r}) \\ \boldsymbol{q} \otimes (\boldsymbol{p} + \boldsymbol{r}) = \boldsymbol{q} \otimes \boldsymbol{p} + \boldsymbol{q} \otimes \boldsymbol{r} \end{gathered} \tag{B.9}$$

The quaternion product can also be written in matrix form based on equation (B.7):

$$\begin{aligned} \boldsymbol{q} \otimes \boldsymbol{p} &= \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \\ &= \begin{bmatrix} q_0 & -\boldsymbol{q}_v^T \\ \boldsymbol{q}_v & q_0 \boldsymbol{I}_3 - (\boldsymbol{q}_v)\times \end{bmatrix} \begin{bmatrix} p_0 \\ \boldsymbol{p}_v \end{bmatrix} \\ &= [\boldsymbol{q} \quad \boldsymbol{\Psi}(\boldsymbol{q})]\boldsymbol{p} \\ &= \boldsymbol{\mathcal{L}}(\boldsymbol{q})\boldsymbol{p} \end{aligned} \tag{B.10}$$

Or

$$\begin{aligned}\boldsymbol{q} \otimes \boldsymbol{p} &= \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \\ &= \begin{bmatrix} p_0 & -\boldsymbol{p}_v^T \\ \boldsymbol{p}_v & p_0 \boldsymbol{I}_3 + (\boldsymbol{p}_v)\times \end{bmatrix} \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix} \\ &= [\boldsymbol{p} \quad \mathbb{E}(\boldsymbol{p})]\boldsymbol{q} \\ &= \boldsymbol{\mathcal{R}}(\boldsymbol{p})\boldsymbol{q}\end{aligned} \tag{B.11}$$

With:

$$\begin{aligned}\boldsymbol{\mathcal{L}}(\boldsymbol{q}) &= \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}, \qquad \boldsymbol{\mathcal{R}}(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \\ \boldsymbol{\Psi}(\boldsymbol{q}) &= \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & q_3 & -q_2 \\ -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \end{bmatrix}, \qquad \mathbb{E}(\boldsymbol{q}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}\end{aligned} \tag{B.12}$$

- **Identity**

Quaternions have neutral element with respect to multiplication, which is defined as:

$$\boldsymbol{q}_{\mathrm{n}} = [1 \quad 0 \quad 0 \quad 0]^T \tag{B.13}$$

- **Inverse**

The inverse rotation is described by the inverse quaternion $\mathbf{q}^{-1}$, defined as:

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \boldsymbol{q}_{\mathrm{n}} \tag{B.14}$$

Therefore:

$$\mathbf{q}^{-1} = \begin{bmatrix} q_0 \\ -\boldsymbol{q}_v \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\alpha}{2}\right) \\ -\sin\left(\frac{\alpha}{2}\right) \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{-\alpha}{2}\right) \\ -\sin\left(\frac{-\alpha}{2}\right) \boldsymbol{u} \end{bmatrix} \tag{B.15}$$

In addition, the inverse of a quaternion product is the product of the commuted quaternion inverse:

$$(\boldsymbol{q} \otimes \boldsymbol{p})^{-1} = \mathbf{p}^{-1} \otimes \mathbf{q}^{-1} \tag{B.16}$$

- **Vector rotation using quaternions**

Let us consider two different coordinate systems (s) and (t). Let $\boldsymbol{v}^s$ be a (3×1) vector expressed in (s) and $\boldsymbol{q}_{s2t}$ is the quaternion representing the rotation from (s) to (t). Then (Trawny, 2005):

$$\boldsymbol{v}^t = \boldsymbol{q}_{s2t} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \otimes \boldsymbol{q}_{s2t}^{-1} \tag{B.17}$$

- **Rotation quaternion composition**

Let $\boldsymbol{q}_{s2w}$, $\boldsymbol{q}_{w2t}$ and $\boldsymbol{q}_{s2t}$be three rotation quaternions, respectively from the coordinate system (s) to the coordinate system (w), from (w) to (t) and from (s) to (t) and let $\boldsymbol{v}^t$, $\boldsymbol{v}^s$ and $\boldsymbol{v}^w$ be the (3×1) vector $\boldsymbol{v}$ expressed respectively in the (t), (s) and (w) coordinate systems. Then, using the quaternion product associativity in equation (B.9) and the product inverse property in equation (B.16), we have:

$$\begin{aligned} \boldsymbol{v}^t &= \boldsymbol{q}_{w2t} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^w \end{bmatrix} \otimes \boldsymbol{q}_{w2t}^{-1} \\ &= \boldsymbol{q}_{w2t} \otimes \left(\boldsymbol{q}_{s2w} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \otimes \boldsymbol{q}_{s2w}^{-1} \right) \otimes \boldsymbol{q}_{w2t}^{-1} \\ &= (\boldsymbol{q}_{w2t} \otimes \boldsymbol{q}_{s2w}) \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \otimes (\boldsymbol{q}_{w2t} \otimes \boldsymbol{q}_{s2w})^{-1} \end{aligned} \tag{B.18}$$

Then using equation (B.17) we have

$$\boldsymbol{q}_{s2t} = \boldsymbol{q}_{w2t} \otimes \boldsymbol{q}_{s2w} \tag{B.19}$$

B.2 Rotation matrix

A rotation matrix is a (3×3) matrix used to perform a rotation from a coordinate system (s) to another coordinate system (t). Let $\boldsymbol{v}^s$ be a (3×1) vector expressed in (s) and $\boldsymbol{C}_{s2t}$ is the rotation matrix from (s) to (t), then:

$$\boldsymbol{v}^t = \boldsymbol{C}_{s2t} \boldsymbol{v}^s \tag{B.20}$$

- **Multiplication**

As for any matrix product, the rotation matrix multiplication is associative but not commutative:

$$\begin{aligned} \boldsymbol{C}_1(\boldsymbol{C}_2\boldsymbol{C}_3) &= (\boldsymbol{C}_1\boldsymbol{C}_2)\boldsymbol{C}_3 \\ \boldsymbol{C}_1\boldsymbol{C}_2 &\neq \boldsymbol{C}_2\boldsymbol{C}_1 \end{aligned} \tag{B.21}$$

- **Inverse**

Let $\boldsymbol{C}_{s2t}$ be a rotation matrix from (s) to (t). The inverse rotation matrix describe the rotation from (t) to (s):

$$\boldsymbol{C}_{s2t}^{-1} = \boldsymbol{C}_{t2s} \tag{B.22}$$

The rotation matrix is an orthogonal matrix:

$$\boldsymbol{C}_{s2t}\boldsymbol{C}_{s2t}^{T} = \boldsymbol{I}_3 \tag{B.23}$$

Or equivalently:

$$\boldsymbol{C}_{s2t}^{-1} = \boldsymbol{C}_{s2t}^{T} = \boldsymbol{C}_{t2s} \tag{B.24}$$

As for any square matrices, the inverse of the product of two matrices is the product of their inverse:

$$(\boldsymbol{C}_1\boldsymbol{C}_2)^{-1} = \boldsymbol{C}_2^{-1}\boldsymbol{C}_1^{-1} \tag{B.25}$$

- **Rotation matrix composition**

Let $\boldsymbol{C}_{s2w}$, $\boldsymbol{C}_{w2t}$ and $\boldsymbol{C}_{s2t}$be three rotation quaternions, respectively from the coordinate system (s) to the coordinate system (w), from (w) to (t) and from (s) to (t) and let $\boldsymbol{v}^t$, $\boldsymbol{v}^s$ and $\boldsymbol{v}^w$ be the (3×1) vector $\boldsymbol{v}$ expressed respectively in the (t), (s) and (w) coordinate systems. Then, using the matrix product associativity in equation (B.21), we have:

$$\begin{aligned} \boldsymbol{v}^t &= \boldsymbol{C}_{w2t}\boldsymbol{v}^w \\ &= (\boldsymbol{C}_{w2t}\boldsymbol{C}_{s2w})\boldsymbol{v}^s \\ &= \boldsymbol{C}_{s2t}\boldsymbol{v}^s \end{aligned} \tag{B.26}$$

Therefore:

$$\boldsymbol{C}_{s2t} = \boldsymbol{C}_{w2t}\boldsymbol{C}_{s2w} \tag{B.27}$$

B.3 Euler angles

The Euler angles describe the orientation of a rigid body (b) with respect to a fixed coordinate system (the (l) frame for example). There are three Euler angles in a three-dimensional space which are the roll, the pitch and the yaw described in Figure B-1:

- Roll ϕ: the angle of rotation about the forward vehicle axis
- Pitch θ: the angle of rotation about the lateral vehicle axis
- Yaw ψ: the angle of rotation about the vertical vehicle axis

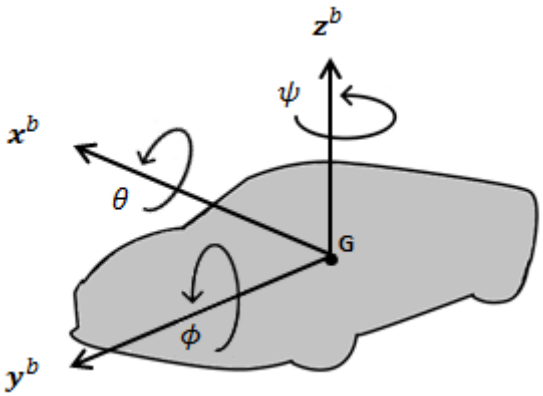


Figure B-1 Euler angles

Figure B-2 illustrates three successive rotations transforming the (l) frame into the (b) frame. The first rotation turns the (l) frame by an angle (ψ) about the z-axis, transforming the x and y axes and leaving the z-axis the same. The second rotation turns the obtained frame by an angle (θ) about the x-axis, transforming the y and z axes and leaving the x-axis the same. Finally, the third rotation turns the frame obained after the previous two rotations by an angle (ϕ) about the y-axis to obtain the (b) frame. The inverse rotation transforms the (b) frame into the (l) frame. This means that in order to transform (b) into (l), we need to turn first about the y-axis by (ϕ), then about the x-axis by (θ) and finally about the z-axis by (ψ).

Note that the order in which the three rotations are carried out is critical. If the rotations are performed in a different order, the orientation of the axes at the end of the transformation is generally different.

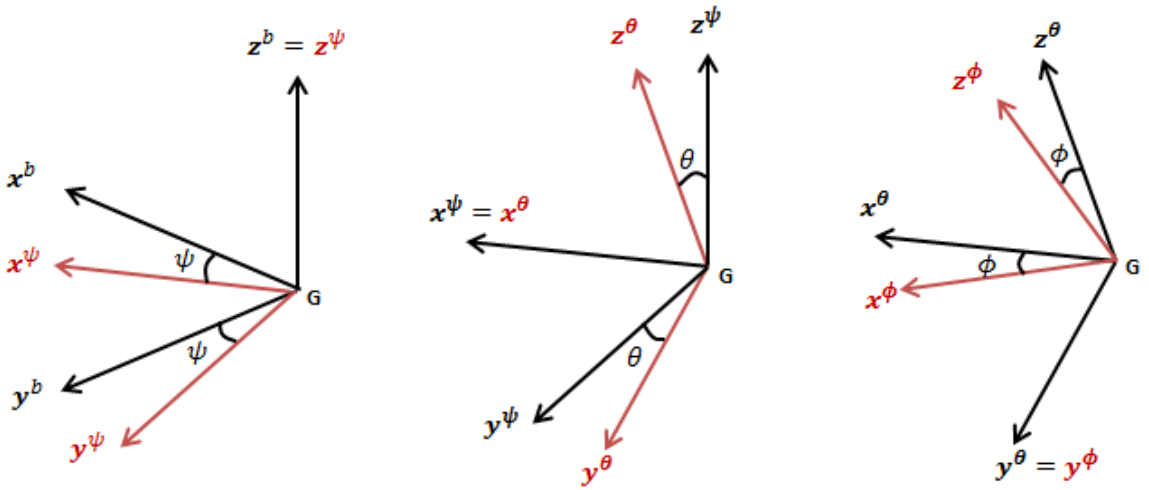


Figure B-2 Transformation from (l) to (v)

Note also that the pitch should satisfy the following constraint $-90 < \theta < 90°$ for the unicity of the angle triplet representing the same attitude.

Based on the notations in Figure B-2, the following relationships between the different axes are derived:

- **Rotation about the z-axis**

The rotation about the z-axis rotates the x and y axes with ψ:

$$\begin{aligned} \boldsymbol{x}^\psi &= \cos\psi\ \boldsymbol{x}^b + \sin\psi\ \boldsymbol{y}^b \\ \boldsymbol{y}^\psi &= -\sin\psi\ \boldsymbol{x}^b + \cos\psi\ \boldsymbol{y}^b \\ \boldsymbol{z}^\psi &= \boldsymbol{z}^b \end{aligned} \tag{B.28}$$

In matrix form, this is written:

$$\begin{bmatrix} \boldsymbol{x}^{\psi} \\ \boldsymbol{y}^{\psi} \\ \boldsymbol{z}^{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{b} \\ \boldsymbol{y}^{b} \\ \boldsymbol{z}^{b} \end{bmatrix} = \boldsymbol{C}_{z}(\psi) \begin{bmatrix} \boldsymbol{x}^{b} \\ \boldsymbol{y}^{b} \\ \boldsymbol{z}^{b} \end{bmatrix} \tag{B.29}$$

- **Rotation about the x-axis**

The rotation about the x-axis rotates the y and z axes with θ:

$$\begin{aligned} \boldsymbol{x}^{\theta} &= \boldsymbol{x}^{\psi} \\ \boldsymbol{y}^{\theta} &= \cos\theta\ \boldsymbol{y}^{\psi} + \sin\theta\ \boldsymbol{z}^{\psi} \\ \boldsymbol{z}^{\theta} &= -\sin\theta\ \boldsymbol{y}^{\psi} + \cos\theta\ \boldsymbol{z}^{\psi} \end{aligned} \tag{B.30}$$

In matrix form, this is written:

$$\begin{bmatrix} \boldsymbol{x}^{\theta} \\ \boldsymbol{y}^{\theta} \\ \boldsymbol{z}^{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{\psi} \\ \boldsymbol{y}^{\psi} \\ \boldsymbol{z}^{\psi} \end{bmatrix} = \boldsymbol{C}_{x}(\theta) \begin{bmatrix} \boldsymbol{x}^{\psi} \\ \boldsymbol{y}^{\psi} \\ \boldsymbol{z}^{\psi} \end{bmatrix} \tag{B.31}$$

- **Rotation about the y-axis**

The rotation about the y-axis rotates the x and z axes with ϕ:

$$\begin{aligned} \boldsymbol{x}^{l} &= \cos\phi\ \boldsymbol{x}^{\theta} - \sin\phi\ \boldsymbol{z}^{\theta} \\ \boldsymbol{y}^{l} &= \boldsymbol{y}^{\theta} \\ \boldsymbol{z}^{l} &= \sin\phi\ \boldsymbol{x}^{\theta} + \cos\phi\ \boldsymbol{z}^{\theta} \end{aligned} \tag{B.32}$$

In matrix form, this is written:

$$\begin{bmatrix} \boldsymbol{x}^{l} \\ \boldsymbol{y}^{l} \\ \boldsymbol{z}^{l} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{\theta} \\ \boldsymbol{y}^{\theta} \\ \boldsymbol{z}^{\theta} \end{bmatrix} = \boldsymbol{C}_{y}(\phi) \begin{bmatrix} \boldsymbol{x}^{\theta} \\ \boldsymbol{y}^{\theta} \\ \boldsymbol{z}^{\theta} \end{bmatrix} \tag{B.33}$$

The use of Euler angles for representing a rotation is not recommended for the following reasons:

- To reverse a rotation, we cannot simply reverse the sign of the Euler angle. More complicated operations are required
- Successive rotations cannot be expressed by simply adding the Euler angle.

For these reasons, it is better to use rotation matrices or rotation quaternions to manipulate rotations.

B.4 Relationship between the different representations

B.4.1 Relationship between quaternion and rotation matrix

- **From quaternion to rotation matrix**

We recall equation (B.17) describing the rotation of a vector with a quaternion:

$$\boldsymbol{v}^t = \boldsymbol{q}_{s2t} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \otimes \boldsymbol{q}_{s2t}^{-1}$$

To simplify the notations, we are going to use the notation $\boldsymbol{q}_{s2t} = \boldsymbol{q} = \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$.

Therefore:

$$\begin{aligned}
\boldsymbol{v}^t &= \begin{bmatrix} q_0 \\ \boldsymbol{q}_v \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ -\boldsymbol{q}_v \end{bmatrix} \\
&= \begin{bmatrix} -\boldsymbol{q}_v^T.\boldsymbol{v}^s \\ q_0\boldsymbol{v}^s - \boldsymbol{q}_v \times \boldsymbol{v}^s \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ -\boldsymbol{q}_v \end{bmatrix} \\
&= \begin{bmatrix} -\boldsymbol{q}_v^T.\boldsymbol{v}^s & -(q_0\boldsymbol{v}^s - \boldsymbol{q}_v \times \boldsymbol{v}^s)^T \\ q_0\boldsymbol{v}^s - \boldsymbol{q}_v \times \boldsymbol{v}^s & -\boldsymbol{q}_v^T.\boldsymbol{v}^s\boldsymbol{I}_3 - (q_0\boldsymbol{v}^s - \boldsymbol{q}_v \times \boldsymbol{v}^s)\times \end{bmatrix} \begin{bmatrix} q_0 \\ -\boldsymbol{q}_v \end{bmatrix} \\
&= \begin{bmatrix} \underbrace{-q_0\boldsymbol{q}_v^T.\boldsymbol{v}^s + q_0\boldsymbol{v}^{sT}.\boldsymbol{q}_v}_{0} - \underbrace{(\boldsymbol{q}_v \times \boldsymbol{v}^s)^T.\boldsymbol{q}_v}_{\text{Dot product of 2 orthogonal vect=0}} \\ q_0^2\boldsymbol{v}^s - q_0\boldsymbol{q}_v \times \boldsymbol{v}^s + \underbrace{\boldsymbol{q}_v^T.\boldsymbol{v}^s.\boldsymbol{q}_v}_{\boldsymbol{q}_v\boldsymbol{q}_v^T\boldsymbol{v}^s} + q_0\boldsymbol{v}^s \times \boldsymbol{q}_v - \underbrace{(\boldsymbol{q}_v \times \boldsymbol{v}^s) \times \boldsymbol{q}_v}_{\boldsymbol{q}_v^T\boldsymbol{q}_v\boldsymbol{v}^s - \boldsymbol{q}_v^T.\boldsymbol{v}^s.\boldsymbol{q}_v} \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ q_0^2\boldsymbol{v}^s - 2\boldsymbol{q}_v \times \boldsymbol{v}^s + 2\boldsymbol{q}_v\boldsymbol{q}_v^T\boldsymbol{v}^s - \underbrace{\boldsymbol{q}_v^T\boldsymbol{q}_v}_{1-q_0^2}\boldsymbol{v}^s \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ (2q_0^2 - 1)\boldsymbol{I}_3 - 2q_0(\boldsymbol{q}_v)\times + 2\boldsymbol{q}_v\boldsymbol{q}_v^T \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ \boldsymbol{C}_{s2t} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{v}^s \end{bmatrix}
\end{aligned} \tag{B.34}$$

Therefore, the rotation matrix is expressed as function of the corresponding quaternion components using:

$$\boldsymbol{C}_{s2t} = (2q_0^2 - 1)\boldsymbol{I}_3 - 2q_0(\boldsymbol{q}_v)\times + 2\boldsymbol{q}_v\boldsymbol{q}_v^T$$

$$= \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2q_0^2 + 2q_2^2 - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \quad \text{(B.35)}$$

Based on the formula in (B.19), we deduce an important property. Let $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ be two rotation quaternions, then their product composes consecutive rotations in the same order as rotation matrices do.

$$\boldsymbol{C}(\boldsymbol{q}_1 \otimes \boldsymbol{q}_2) = \boldsymbol{C}(\boldsymbol{q}_1)\boldsymbol{C}(\boldsymbol{q}_2) \quad \text{(B.36)}$$

- **From rotation matrix to quaternion**

Let us consider that the rotation matrix $\boldsymbol{C}_{s2t}$ is written:

$$\boldsymbol{C}_{s2t} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad \text{(B.37)}$$

Based on (B.35), derived in the previous section, the expressions of the quaternion components as a function of the rotation matrix components are the following:

$$q_0 = \frac{1}{2}\sqrt{1 + c_{11} + c_{22} + c_{33}}$$

$$q_1 = \frac{c_{23} - c_{32}}{4q_0}$$

$$q_2 = \frac{c_{31} - c_{13}}{4q_0} \quad \text{(B.38)}$$

$$q_3 = \frac{c_{12} - c_{21}}{4q_0}$$

In case where q_0 is close to zero, then the first calculated component should be replaced by q_1, q_2 or q_3. For example, if it is q_1, then

$$q_1 = \frac{1}{2}\sqrt{1 + c_{11} - c_{22} - c_{33}}$$

$$q_0 = \frac{c_{23} - c_{32}}{4q_1}$$

$$q_2 = \frac{c_{12} + c_{21}}{4q_1} \tag{B.39}$$

$$q_3 = \frac{c_{13} + c_{31}}{4q_1}$$

The quaternion components should satisfy the unit norm constraint.

B.4.2 Relationship between rotation matrix and Euler angles

- **From Euler angles to rotation matrix**

Euler angles represent successive rotations about the three vehicle axes. By taking the example given in Section B.3, the composition (multiplication) of the rotations defined in this section results in the rotation matrix:

$$\boldsymbol{C}_{l2b} = \boldsymbol{C}_y(\phi)\boldsymbol{C}_x(\theta)\boldsymbol{C}_z(-\psi) \tag{B.40}$$

$\boldsymbol{C}_z(-\psi)$ is considered instead of $\boldsymbol{C}_z(\psi)$ to respect the right-handed transfomation between the (E, N, U) system of the (l) frame and the (x, y, z) system of the (b) frame (with y pointing to the forward direction).

Therefore

$$\boldsymbol{C}_{l2b} = \begin{bmatrix} \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & -\sin\phi\cos\theta \\ \cos\theta\sin\psi & \cos\theta\cos\psi & \sin\theta \\ \sin\phi\cos\psi - \cos\phi\sin\theta\sin\psi & -\sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{B.41}$$

This matrix represents the rotation from the (l) frame to the (b) frame and its inverse (transpose) $\boldsymbol{C}_{b2l}$ represents the vehicle attitude.

- **From rotation matrix to Euler angles**

Based on equation (B.41), derived in the previous section, the expressions of the Euler angles as a function of the rotation matrix components are the following:

$$\phi = -\arctan2(c_{13}, c_{33})$$

$$\theta = -\arcsin(c_{23}) \tag{B.42}$$

$$\psi = \arctan2(c_{21}, c_{22})$$

B.4.3 Relationship between quaternion and Euler angles

- **From Euler angles to quaternion**

The quaternion $\boldsymbol{q}_{l2b}$ representing the rotation from the (l) frame to the (b) frame can be defined by the composition of the Euler rotations about each axis as:

$$\boldsymbol{q}_{l2b} = \boldsymbol{q}_y(\phi) \otimes \boldsymbol{q}_x(\theta) \otimes \boldsymbol{q}_z(-\psi) \tag{B.43}$$

Therefore, to express the relationship between the quaternion and the Euler angles using the JPL convention, we can write:

$$\begin{aligned}
(\boldsymbol{q}_{l2b})_{\text{JPL}} &= \boldsymbol{q}_y(\phi) \otimes \boldsymbol{q}_x(\theta) \otimes \boldsymbol{q}_z(-\psi) \\
&= \begin{bmatrix} \cos\frac{\phi}{2} \\ 0 \\ \sin\frac{\phi}{2} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos\frac{\psi}{2} \\ 0 \\ 0 \\ -\sin\frac{\psi}{2} \end{bmatrix} \\
&= \boldsymbol{\mathcal{L}}\left(\boldsymbol{q}_y(\phi)\right) \boldsymbol{\mathcal{L}}\left(\boldsymbol{q}_x(\theta)\right) \boldsymbol{q}_y \boldsymbol{q}_z(-\psi) \\
&= \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ -\cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} \\ -\cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{bmatrix}
\end{aligned} \tag{B.44}$$

If the Hamilton convention was used, then the composition of the quaternion would have been:

$$(\boldsymbol{q}_{l2b})_{Ham} = \boldsymbol{q}_y(-\phi) \otimes \boldsymbol{q}_x(-\theta) \otimes \boldsymbol{q}_z(\psi) \tag{B.45}$$

This formula does not correspond to the same attitude angles as for the rotation matrices but to their opposite. Therefore, to avoid confusions and apply all the formulas "naturally" the JPL convention is preferred.

- **From quaternion to Euler angles**

Based on equation (B.42) and equations (B.39), the transformation between the quaternion $\boldsymbol{q}_{l2b}$ and Euler angles are determined:

$$\begin{aligned}
\phi &= -\arctan2\left(\left(2(q_0q_2 - q_1q_3)\right), (2q_0^2 + 2q_3^2 - 1)\right) \\
\theta &= -\arcsin\left(2(q_2q_3 + q_0q_1)\right) \\
\psi &= \arctan2\left(\left(2(q_1q_2 - q_0q_3)\right), (2q_0^2 + 2q_2^2 - 1)\right)
\end{aligned} \tag{B.46}$$

B.5 Rotation error model

The rotation error model is only applied to quaternions and rotation matrices. The Euler angles are not considered in this study and are only derived from quaternions or rotations because of the critical manipulation of these angles.

B.5.1 Quaternions

Let $\hat{\boldsymbol{q}}(\alpha)$ be an estimated value of a quaternion and $\boldsymbol{q}(\alpha)$ its actual value. Let $\delta\alpha$ be a small error affecting the quaternion angle of $\hat{\boldsymbol{q}}(\alpha)$, then using the sine and cosine small angle first order approximations ($\cos\delta\alpha \approx 1$) and $\sin(\delta\alpha \approx \delta\alpha)$, we have:

$$\begin{aligned}
\hat{\boldsymbol{q}}(\alpha) &= \boldsymbol{q}(\alpha + \delta\alpha) \\
&= \begin{bmatrix} \cos\left(\dfrac{\alpha+\delta\alpha}{2}\right) \\ \sin\left(\dfrac{\alpha+\delta\alpha}{2}\right)\boldsymbol{u} \end{bmatrix} \\
&= \begin{bmatrix} \cos\dfrac{\alpha}{2} - \dfrac{\delta\alpha}{2}\sin\dfrac{\alpha}{2} \\ \left(\sin\dfrac{\alpha}{2} + \dfrac{\delta\alpha}{2}\cos\dfrac{\alpha}{2}\right)\boldsymbol{u} \end{bmatrix} \\
&= \begin{bmatrix} \cos\dfrac{\alpha}{2} & -\sin\dfrac{\alpha}{2}\boldsymbol{u}^T \\ \sin\dfrac{\alpha}{2}\boldsymbol{u} & \cos\dfrac{\alpha}{2}\boldsymbol{I}_3 + \left(\sin\dfrac{\alpha}{2}\boldsymbol{u}\right)\times \end{bmatrix} \begin{bmatrix} 1 \\ \dfrac{\delta\alpha}{2}\boldsymbol{u} \end{bmatrix} \\
&= \boldsymbol{\mathcal{R}}\big(\boldsymbol{q}(\alpha)\big)\delta\boldsymbol{q} \\
&= \delta\boldsymbol{q} \otimes \boldsymbol{q}(\alpha)
\end{aligned} \tag{B.47}$$

where $\delta\boldsymbol{q}$ is the quaternion error:

$$\delta\boldsymbol{q} = \begin{bmatrix} 1 \\ \dfrac{\delta\alpha}{2}\boldsymbol{u} \end{bmatrix} = \begin{bmatrix} 1 \\ \dfrac{\boldsymbol{\varepsilon}}{2} \end{bmatrix} \tag{B.48}$$

Therefore, we can write:

$$\hat{\boldsymbol{q}} = [\boldsymbol{q} \quad \mathbb{E}(\boldsymbol{q})] \begin{bmatrix} 1 \\ \dfrac{\boldsymbol{\varepsilon}}{2} \end{bmatrix} = \boldsymbol{q} + \frac{1}{2}\mathbb{E}(\boldsymbol{q})\boldsymbol{\varepsilon} \tag{B.49}$$

and

$$\boldsymbol{q} = \delta\boldsymbol{q}^{-1} \otimes \hat{\boldsymbol{q}} = [\hat{\boldsymbol{q}} \quad \mathbb{E}(\hat{\boldsymbol{q}})] \begin{bmatrix} 1 \\ -\dfrac{\boldsymbol{\varepsilon}}{2} \end{bmatrix} = \hat{\boldsymbol{q}} - \frac{1}{2}\mathbb{E}(\hat{\boldsymbol{q}})\boldsymbol{\varepsilon} \tag{B.50}$$

Therefore,

$$\hat{\boldsymbol{q}}^{-1} = (\delta\boldsymbol{q} \otimes \boldsymbol{q})^{-1} = \boldsymbol{q}^{-1} - \frac{1}{2}\boldsymbol{\Psi}(\boldsymbol{q}^{-1})\boldsymbol{\varepsilon} \tag{B.51}$$

and

$$\boldsymbol{q}^{-1} = (\hat{\boldsymbol{q}}^{-1} \otimes \delta\boldsymbol{q})^{-1} = \hat{\boldsymbol{q}}^{-1} + \frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}^{-1})\boldsymbol{\varepsilon} \tag{B.52}$$

B.5.2 Rotation matrix

The rotation error model is derived from the quaternion model in equation(B.47). Based on equation (B.35)expressing a rotation matrix as a function of its corresponding quaternion and equation (B.36)expressing the rotation of a quaternion product as a function of the product of quaternion rotations, the rotation matrix error model can be written:

$$\begin{aligned} \widehat{\boldsymbol{C}} &= \boldsymbol{C}\big(\hat{\boldsymbol{q}}(\alpha)\big) \\ &= \boldsymbol{C}\big(\delta\boldsymbol{q} \otimes \boldsymbol{q}(\alpha)\big) \\ &= \boldsymbol{C}(\delta\boldsymbol{q})\boldsymbol{C}\big(\boldsymbol{q}(\alpha)\big) \\ &= \delta\boldsymbol{C}\,\boldsymbol{C} \end{aligned} \tag{B.53}$$

$\delta\boldsymbol{C}$ is given by the following expression (first order approximation):

$$\delta\boldsymbol{C} \approx \boldsymbol{I}_3 - (\boldsymbol{\varepsilon})\times \tag{B.54}$$

Therefore,

$$\widehat{\boldsymbol{C}} = (\boldsymbol{I}_3 - \boldsymbol{E})\boldsymbol{C} \tag{B.55}$$

where $\boldsymbol{E}$ is the skew-symmetric matrix of $\boldsymbol{\varepsilon}$.

Appendix C

Calculation of the Kalman Filter observation matrices

This appendix justifies the expressions of the Kalman filter observation matrices given in Section 4.6.

C.1 Perturbation model

Based on the following measurement models, the perturbation of quantities to the first order used in the calculation of the observation matrices are expressed by:

C.1.1 Vectors and scalars

IMU-related components:

Position:

$$\widehat{\boldsymbol{p}}_b^l = \boldsymbol{p}_b^l + \delta\boldsymbol{p}_b^l \tag{C.1}$$

Velocity:

$$\widehat{\boldsymbol{v}}_b^l = \boldsymbol{v}_b^l + \delta\boldsymbol{v}_b^l \tag{C.2}$$

Gyroscope measurement:

$$\delta\boldsymbol{\omega}_{ib}^b = \widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{\omega}_{ib}^b = \boldsymbol{\gamma}(\boldsymbol{k}_\omega)\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_\omega + \boldsymbol{\eta}_\omega \tag{C.3}$$

with $\gamma(\boldsymbol{k}_\omega) = \text{diag}\left(\frac{1}{1+k_{\omega x}}, \frac{1}{1+k_{\omega y}}, \frac{1}{1+k_{\omega z}}\right)$

GNSS-related components:

Position:

$$\widetilde{\boldsymbol{p}}_G^l = \boldsymbol{p}_G^l + \delta\boldsymbol{p}_G^l \tag{C.4}$$

Velocity:

$$\widetilde{\boldsymbol{v}}_G^l = \boldsymbol{v}_G^l + \delta\boldsymbol{v}_G^l \tag{C.5}$$

Pseudorange:

$$\widetilde{\boldsymbol{\rho}}_G = \boldsymbol{\rho}_G + \delta\boldsymbol{\rho}_G \tag{C.6}$$

Pseudorange rate:

$$\widetilde{\dot{\boldsymbol{\rho}}}_G = \dot{\boldsymbol{\rho}}_G + \delta\dot{\boldsymbol{\rho}}_G \tag{C.7}$$

Clock bias:

$$\hat{b}_G^{clk} = b_G^{clk} + \delta b_G^{clk} \tag{C.8}$$

Clock bias:

$$\hat{\dot{b}}_G^{clk} = \dot{b}_G^{clk} + \delta \dot{b}_G^{clk} \tag{C.9}$$

SLAM-related components:

Position:

$$\tilde{\boldsymbol{p}}_c^v = \boldsymbol{p}_c^v + \delta \boldsymbol{p}_c^v \tag{C.10}$$

Scale factor:

$$\hat{k}_v = k_v + \delta k_v \tag{C.11}$$

(l)-(v) frame offset:

$$\hat{\boldsymbol{p}}_{lv}^v = \boldsymbol{p}_{lv}^v + \delta \boldsymbol{p}_{lv}^v \tag{C.12}$$

WSS-related components:

Velocity:

$$\tilde{\boldsymbol{v}}_w^b = \boldsymbol{v}_w^b + \delta \boldsymbol{v}_w^b \tag{C.13}$$

Scale factor:

$$\hat{k}_w = k_w + \delta k_w \tag{C.14}$$

C.1.2 Rotations

The following equations expressing the rotation error models are derived from equation (B.49)and equation (B.55).

(b) to (l) rotation:

$$\begin{aligned} \widehat{\boldsymbol{C}}_{b2l} &= (\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\boldsymbol{C}_{b2l} \\ \widehat{\boldsymbol{q}}_{b2l} &= \boldsymbol{q}_{b2l} + \frac{1}{2}\mathbb{E}(\boldsymbol{q}_{b2l})\boldsymbol{\varepsilon}_{b2l} \\ \boldsymbol{q}_{l2b} &= \widehat{\boldsymbol{q}}_{l2b} + \frac{1}{2}\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{l2b})\boldsymbol{\varepsilon}_{b2l} \end{aligned} \tag{C.15}$$

(l) to (v) rotation:

$$\begin{aligned}\widehat{\boldsymbol{C}}_{l2v} &= (\boldsymbol{I}_3 - \boldsymbol{E}_{l2v})\boldsymbol{C}_{l2v} \\ \widehat{\boldsymbol{q}}_{l2v} &= \boldsymbol{q}_{l2v} + \tfrac{1}{2}\mathbb{E}(\boldsymbol{q}_{l2v})\boldsymbol{\varepsilon}_{l2v} \\ \boldsymbol{q}_{v2l} &= \widehat{\boldsymbol{q}}_{v2l} + \tfrac{1}{2}\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\varepsilon}_{l2v}\end{aligned} \tag{C.16}$$

(b) to (c) rotation

$$\begin{aligned}\widehat{\boldsymbol{C}}_{b2c} &= (\boldsymbol{I}_3 - \boldsymbol{E}_{b2c})\boldsymbol{C}_{b2c} \\ \widehat{\boldsymbol{q}}_{b2c} &= \boldsymbol{q}_{b2c} + \tfrac{1}{2}\mathbb{E}(\boldsymbol{q}_{b2c})\boldsymbol{\varepsilon}_{b2c} \\ \boldsymbol{q}_{b2c} &= \widehat{\boldsymbol{q}}_{b2c} - \tfrac{1}{2}\mathbb{E}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\varepsilon}_{b2c}\end{aligned} \tag{C.17}$$

C.2 Observation matrices

The calculation of the observation matrices is based on the measurements models relating the sensors outputted measurements to the state vector components, given in Section 4.5, and the perturbation models of the measurements and the state vector components given in equations (C.1) to (C.17)

C.2.1 GNSS observation matrices

Loose coupling

The position observation model is given by:

$$\widetilde{\boldsymbol{p}}_G^l = \widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\ \Delta\boldsymbol{p}_G^b \tag{C.18}$$

By replacing the quantities by their perturbation model, we have:

$$\boldsymbol{p}_G^l + \delta\boldsymbol{p}_G^l = \boldsymbol{p}_b^l + \delta\boldsymbol{p}_b^l + (\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_G^b \tag{C.19}$$

Therefore, knowing that $\boldsymbol{p}_G^l = \boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\ \Delta\boldsymbol{p}_G^b$, we obtain:

$$\begin{aligned}\delta\boldsymbol{p}_G^l &= \delta\boldsymbol{p}_b^l - \boldsymbol{E}_{b2l}\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_G^b \\ &= \delta\boldsymbol{p}_b^l + \left(\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_G^b\right) \times \boldsymbol{\varepsilon}_{b2l}\end{aligned} \tag{C.20}$$

where $\left(\delta\boldsymbol{p}_G^l\right)$ is the difference between the GNSS measured position and the prediction of this position using the IMU propagation (first Kalman filer step):

$$\delta\boldsymbol{p}_G^l = \widehat{\boldsymbol{p}}_b^l - \widetilde{\boldsymbol{p}}_G^l + \widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_G^b \tag{C.21}$$

The same development applies for the GNSS velocity. Using the velocity observation model:

$$\tilde{\boldsymbol{v}}_G^l = \hat{\boldsymbol{v}}_b^l + \hat{\boldsymbol{C}}_{b2l}\left(\left(\hat{\boldsymbol{\omega}}_{lb}^b\right) \times \Delta \boldsymbol{p}_G^b\right) \tag{C.22}$$

As justified in Section 4.6.2, $\left(\hat{\boldsymbol{\omega}}_{lb}^b\right)$ can be approximated by $\left(\tilde{\boldsymbol{\omega}}_{ib}^b\right)$ for a land vehicle using a MEMS IMU, therefore

$$\boldsymbol{v}_G^l + \delta \boldsymbol{v}_G^l = \boldsymbol{v}_b^l + \delta \boldsymbol{v}_b^l + (\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\boldsymbol{C}_{b2l}\left(\left(\boldsymbol{\omega}_{ib}^b + \delta \boldsymbol{\omega}_{ib}^b\right) \times \Delta \boldsymbol{p}_G^b\right) \tag{C.23}$$

$$\begin{aligned} \delta \boldsymbol{v}_G^l &= \delta \boldsymbol{v}_b^l - \boldsymbol{E}_{b2l}\boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{ib}^b \times \Delta \boldsymbol{p}_G^b\right) + \boldsymbol{C}_{b2l}\left(\left(\underbrace{\boldsymbol{\gamma}(\boldsymbol{k}_\omega)\boldsymbol{\omega}_{ib}^b}_{\boldsymbol{W}^b \boldsymbol{k}_\omega} + \boldsymbol{b}_\omega\right) \times \Delta \boldsymbol{p}_G^b\right) \\ \delta \boldsymbol{v}_G^l &= \delta \boldsymbol{v}_b^l + \left(\boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{ib}^b \times \Delta \boldsymbol{p}_G^b\right)\right) \times \boldsymbol{\varepsilon}_{b2l} - \boldsymbol{C}_{b2l}\left(\Delta \boldsymbol{p}_G^b\right) \times \boldsymbol{b}_\omega \\ &\quad - \boldsymbol{C}_{b2l}\left(\Delta \boldsymbol{p}_G^b\right) \times \boldsymbol{W}^b \boldsymbol{k}_\omega \end{aligned} \tag{C.24}$$

where $\boldsymbol{W}^b$ is the (3×3) diagonal matrix whose elements are the x, y and z componenets of $\boldsymbol{\omega}_{ib}^b$.

We conclude that the expression of $\boldsymbol{H}_G$ is given by:

$$\boldsymbol{H}_G = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \left(\hat{\boldsymbol{C}}_{b2l}\,\Delta \boldsymbol{p}_G^b\right)\times & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_{3\times 11} \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \left(\hat{\boldsymbol{C}}_{b2l}\left(\omega_{ib}^b \times \Delta \boldsymbol{p}_G^b\right)\right)\times & \boldsymbol{0}_3 & -\hat{\boldsymbol{C}}_{b2l}(\Delta \boldsymbol{p}_G^b)\times & \boldsymbol{0}_3 & -\hat{\boldsymbol{C}}_{b2l}(\Delta \boldsymbol{p}_G^b) \times \boldsymbol{W}^b & \boldsymbol{0}_{3\times 11} \end{bmatrix} \tag{C.25}$$

Tight coupling

The pseudorange observation model for a satellite si is given by:

$$\tilde{\rho}_G^i = \underbrace{\sqrt{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\,\Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)}}_{d_i} + b_G^{clk} \tag{C.26}$$

The linearization around the state vector prediction point using the first order Taylor approximation leads to the following expression:

$$\tilde{\rho}_G^i \approx \hat{\rho}_G^i + \left.\frac{\partial \rho_G^i}{\partial \boldsymbol{X}}\right|_{\widehat{\boldsymbol{X}}} \left(\boldsymbol{X} - \widehat{\boldsymbol{X}}\right) \tag{C.27}$$

where $\hat{\rho}_G^i$ is the prediction of the ρ_G^i at $\widehat{\boldsymbol{X}}$.

Therefore:

$$\begin{aligned}\tilde{\rho}_G^i - \hat{\rho}_G^i &= \left.\frac{\partial \rho_G^i}{\partial \boldsymbol{X}}\right|_{\widehat{\boldsymbol{X}}} (\boldsymbol{X} - \widehat{\boldsymbol{X}}) \\ &= \left.\frac{\partial \rho_G^i}{\partial \boldsymbol{p}_b^l}\right|_{\widehat{\boldsymbol{p}}_b^l} \underbrace{\left(\boldsymbol{p}_b^l - \widehat{\boldsymbol{p}}_b^l\right)}_{-\delta \boldsymbol{p}_b^l} + \left.\frac{\partial \rho_G^i}{\partial b_G^{clk}}\right|_{\hat{b}_G^{clk}} \underbrace{\left(b_G^{clk} - \hat{b}_G^{clk}\right)}_{-\delta b_G^{clk}} \\ &= \frac{-\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T}{\hat{d}_i} \delta \boldsymbol{p}_b^l - \delta b_G^{clk}\end{aligned} \tag{C.28}$$

As for the pseudorange rate, the observation model for a satellite si is given by:

$$\tilde{\dot{\rho}}_G^i = \frac{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \left(\boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\boldsymbol{\omega}_{lb}^b\right) \times \Delta \boldsymbol{p}_G^b - \boldsymbol{v}_{si}^l\right)}{\sqrt{\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)}} + \dot{b}_G^{clk} \tag{C.29}$$

By linearizing as previously, we have:

$$\begin{aligned}\tilde{\dot{\rho}}_G^i - \hat{\dot{\rho}}_G^i &= \left.\frac{\partial \dot{\rho}_G^i}{\partial \boldsymbol{X}}\right|_{\widehat{\boldsymbol{X}}} (\boldsymbol{X} - \widehat{\boldsymbol{X}}) \\ &= \left.\frac{\partial \dot{\rho}_G^i}{\partial \boldsymbol{p}_b^l}\right|_{\widehat{\boldsymbol{p}}_b^l} \underbrace{\left(\boldsymbol{p}_b^l - \widehat{\boldsymbol{p}}_b^l\right)}_{-\delta \boldsymbol{p}_b^l} + \left.\frac{\partial \dot{\rho}_G^i}{\partial \boldsymbol{v}_b^l}\right|_{\widehat{\boldsymbol{p}}_b^l} \underbrace{\left(\boldsymbol{v}_b^l - \widehat{\boldsymbol{v}}_b^l\right)}_{-\delta \boldsymbol{v}_b^l} + \left.\frac{\partial \dot{\rho}_G^i}{\partial \dot{b}_G^{clk}}\right|_{\hat{\dot{b}}_G^{clk}} \underbrace{\left(\dot{b}_G^{clk} - \hat{\dot{b}}_G^{clk}\right)}_{-\delta \dot{b}_G^{clk}} \\ &= -\left(\frac{\left(\widehat{\boldsymbol{v}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\left(\widehat{\boldsymbol{\omega}}_{lb}^b\right) \times \Delta \boldsymbol{p}_G^b - \boldsymbol{v}_{si}^l\right)^T}{\hat{d}_i} \right. \\ &\quad \left. + \frac{\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \hat{\dot{d}}_i}{\left(\hat{d}_i\right)^2}\right) \delta \boldsymbol{p}_b^l - \frac{\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T}{\hat{d}_i} \delta \boldsymbol{v}_b^l \\ &\quad - \delta \dot{b}_G^{clk}\end{aligned} \tag{C.30}$$

We conclude that:

$$\boldsymbol{H}_G = \begin{bmatrix} \boldsymbol{H}_{G,pPR_1} & \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 26} & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{G,pPR_n} & \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 26} & -1 & 0 \\ \boldsymbol{H}_{G,pPRR_1} & \boldsymbol{H}_{G,vPRR_1} & \boldsymbol{0}_{1\times 26} & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{G,pPRR_n} & \boldsymbol{H}_{G,vPRR_n} & \boldsymbol{0}_{1\times 26} & 0 & -1 \end{bmatrix} \tag{C.31}$$

where

- $\boldsymbol{H}_{G,pPR_i} = \boldsymbol{H}_{G,vPRR_i} = \frac{-\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T}{\hat{d}_i}$
- $\boldsymbol{H}_{G,pPRR_i} = \frac{-\left(\widehat{\boldsymbol{v}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\left(\widehat{\boldsymbol{\omega}}_{lb}^b\right) \times \Delta \boldsymbol{p}_G^b - \boldsymbol{v}_{si}^l\right)^T}{\hat{d}_i} - \frac{\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\, \Delta \boldsymbol{p}_G^b - \boldsymbol{p}_{si}^l\right)^T \hat{\dot{d}}_i}{\left(\hat{d}_i\right)^2}$

C.2.2 SLAM observation matrix

The SLAM position observation model is given by:

$$\tilde{\boldsymbol{p}}_c^v = \hat{k}_v\, \hat{\boldsymbol{C}}_{l2v} \left(\hat{\boldsymbol{p}}_b^l + \hat{\boldsymbol{C}}_{b2l}\, \Delta\boldsymbol{p}_c^b\right) + \hat{\boldsymbol{p}}_{lv}^v \tag{C.32}$$

By replacing the quantities by their perturbation models, we have:

$$\begin{aligned} \boldsymbol{p}_c^v + \delta\boldsymbol{p}_c^v = (k_v + \delta k_v)(\boldsymbol{I}_3 - \boldsymbol{E}_{l2v})\boldsymbol{C}_{l2v}\left(\boldsymbol{p}_b^l + \delta\boldsymbol{p}_b^l + (\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_c^b\right) + \boldsymbol{p}_{lv}^v \\ + \delta\boldsymbol{p}_{lv}^v \end{aligned} \tag{C.33}$$

Therefore by neglecting the terms of order higher then 1, we obtain

$$\begin{aligned} \delta\boldsymbol{p}_c^v &= k_v\boldsymbol{C}_{l2v}\, \delta\boldsymbol{p}_b^l - k_v\boldsymbol{C}_{l2v}\boldsymbol{E}_{b2l}\boldsymbol{C}_{b2l}\, \Delta\boldsymbol{p}_c^b - k_v\boldsymbol{E}_{l2v}\boldsymbol{C}_{l2v}\boldsymbol{p}_b^l - k_v\boldsymbol{E}_{l2v}\boldsymbol{C}_{l2v}\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_c^b \\ &\quad + \delta k_v\boldsymbol{C}_{l2v}\boldsymbol{p}_b^l + \delta k_v\boldsymbol{C}_{l2v}\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_c^b + \delta\boldsymbol{p}_{lv}^v \\ &= k_v\boldsymbol{C}_{l2v}\, \delta\boldsymbol{p}_b^l + k_v\boldsymbol{C}_{l2v}(\boldsymbol{C}_{b2l}\, \Delta\boldsymbol{p}_c^b)\times\boldsymbol{\varepsilon}_{b2l} \\ &\quad + k_v\left(\left(\boldsymbol{C}_{l2v}\boldsymbol{p}_b^l\right)\times + (\boldsymbol{C}_{l2v}\boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_c^b)\times\right)\boldsymbol{\varepsilon}_{l2v} \\ &\quad + \boldsymbol{C}_{l2v}\left(\boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\Delta\boldsymbol{p}_c^b\right)\delta k_v + \delta\boldsymbol{p}_{lv}^v \end{aligned} \tag{C.34}$$

The SLAM attitude measurement model is given by:

$$\tilde{\boldsymbol{q}}_{v2c} = \boldsymbol{q}_{b2c} \otimes \boldsymbol{q}_{l2b} \otimes \boldsymbol{q}_{v2l} \tag{C.35}$$

By replacing the quaternions by their perturbation models given in equations (C.15)-(C.16)-(C.17), we have:

$$\begin{aligned} \tilde{\boldsymbol{q}}_{v2c} = \left(\hat{\boldsymbol{q}}_{b2c} - \frac{1}{2}\mathbb{E}(\hat{\boldsymbol{q}}_{b2c})\boldsymbol{\varepsilon}_{b2c}\right) \otimes \left(\hat{\boldsymbol{q}}_{l2b} + \frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{b2l})\boldsymbol{\varepsilon}_{b2l}\right) \\ \otimes \left(\hat{\boldsymbol{q}}_{v2l} + \frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{v2l})\boldsymbol{\varepsilon}_{l2v}\right) \end{aligned} \tag{C.36}$$

We expand this equation. After neglecting the second and third error orders, we obtain:

$$\begin{aligned} \tilde{\boldsymbol{q}}_{v2c} = \hat{\boldsymbol{q}}_{b2c} \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \hat{\boldsymbol{q}}_{v2l} + \left(-\frac{1}{2}\mathbb{E}(\hat{\boldsymbol{q}}_{b2c})\boldsymbol{\varepsilon}_{b2c}\right) \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \hat{\boldsymbol{q}}_{v2l} + \hat{\boldsymbol{q}}_{b2c} \\ \otimes \left(\frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{b2l})\boldsymbol{\varepsilon}_{b2l}\right) \otimes \hat{\boldsymbol{q}}_{v2l} + \hat{\boldsymbol{q}}_{b2c} \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \left(\frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{v2l})\boldsymbol{\varepsilon}_{l2v}\right) \end{aligned} \tag{C.37}$$

Therefore

$$\begin{aligned} \hat{\boldsymbol{q}}_{b2c} \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \hat{\boldsymbol{q}}_{v2l} - \tilde{\boldsymbol{q}}_{v2c} \\ = \left(\frac{1}{2}\mathbb{E}(\hat{\boldsymbol{q}}_{b2c})\boldsymbol{\varepsilon}_{b2c}\right) \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \hat{\boldsymbol{q}}_{v2l} - \hat{\boldsymbol{q}}_{b2c} \otimes \left(\frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{b2l})\boldsymbol{\varepsilon}_{b2l}\right) \\ \otimes \hat{\boldsymbol{q}}_{v2l} - \hat{\boldsymbol{q}}_{b2c} \otimes \hat{\boldsymbol{q}}_{l2b} \otimes \left(\frac{1}{2}\boldsymbol{\Psi}(\hat{\boldsymbol{q}}_{v2l})\boldsymbol{\varepsilon}_{l2v}\right) \end{aligned} \tag{C.38}$$

In matrix form:

$$\begin{aligned}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{l2b})\boldsymbol{q}_{b2c} - \widetilde{\boldsymbol{q}}_{c2v} \\ = \frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{l2b})\mathbb{E}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\varepsilon}_{b2c} - \frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{l2b})\,\boldsymbol{\varepsilon}_{b2l} \\ - \frac{1}{2}\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{l2b})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{v2l})\,\boldsymbol{\varepsilon}_{l2v}\end{aligned} \tag{C.39}$$

Therefore, the SLAM observation matrix is written:

$$\boldsymbol{H}_v = \begin{bmatrix} \boldsymbol{H}_{v,pp} & \boldsymbol{0}_3 & \boldsymbol{H}_{v,p\varepsilon_{bl}} & \boldsymbol{0}_{3\times13} & \boldsymbol{H}_{v,pk_v} & \boldsymbol{H}_{v,p\varepsilon_{lv}} & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \{\boldsymbol{0}_{3\times2}\} \\ \boldsymbol{0}_{4\times3} & \boldsymbol{0}_{4\times3} & \boldsymbol{H}_{v,q\varepsilon_{bl}} & \boldsymbol{0}_{4\times13} & \boldsymbol{0}_{4\times1} & \boldsymbol{H}_{v,q\varepsilon_{lv}} & \boldsymbol{0}_{4\times3} & \boldsymbol{H}_{v,q\varepsilon_{cb}} & \{\boldsymbol{0}_{4\times2}\} \end{bmatrix} \tag{C.40}$$

where

- $\boldsymbol{H}_{v,pp} = \hat{k}_v\,\widehat{\boldsymbol{C}}_{l2v}$
- $\boldsymbol{H}_{v,p\varepsilon_{bl}} = \hat{k}_v\,\widehat{\boldsymbol{C}}_{l2v}\left(\widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)\times$
- $\boldsymbol{H}_{v,pk_v} = \widehat{\boldsymbol{C}}_{l2v}\left(\widehat{\boldsymbol{p}}_b^l + \widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)$
- $\boldsymbol{H}_{v,p\varepsilon_{lv}} = \hat{k}_v\left(\left(\widehat{\boldsymbol{C}}_{l2v}\widehat{\boldsymbol{C}}_{b2l}\,\Delta\boldsymbol{p}_c^b\right)\times + \left(\widehat{\boldsymbol{C}}_{l2v}\widehat{\boldsymbol{p}}_b^l\right)\times\right)$
- $\boldsymbol{H}_{v,q\varepsilon_{bl}} = -\frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{l2b})$
- $\boldsymbol{H}_{v,q\varepsilon_{lv}} = -\frac{1}{2}\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{b2c})\boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{q}}_{l2b})\boldsymbol{\Psi}(\widehat{\boldsymbol{q}}_{v2l})$
- $\boldsymbol{H}_{v,q\varepsilon_{cb}} = \frac{1}{2}\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{v2l})\boldsymbol{\mathcal{R}}(\widehat{\boldsymbol{q}}_{l2b})\mathbb{E}(\widehat{\boldsymbol{q}}_{b2c})$

C.2.3 WSS and NHC observation matrix

The WSS and NHC measurement model is given by:

$$\widetilde{\boldsymbol{v}}_w^b = \widehat{\boldsymbol{D}}\left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\widehat{\boldsymbol{\omega}}_{lb}^b\right)\times\Delta\boldsymbol{p}_w^b\right) \tag{C.41}$$

Based on equation (C.15) giving the perturbation model of $\boldsymbol{C}_{b2l}$, we have:

$$\widehat{\boldsymbol{C}}_{l2b} = \left(\widehat{\boldsymbol{C}}_{b2l}\right)^T = \left((\boldsymbol{I}_3 - \boldsymbol{E}_{b2l})\boldsymbol{C}_{b2l}\right)^T = \boldsymbol{C}_{l2b}(\boldsymbol{I}_3 + \boldsymbol{E}_{b2l}) \tag{C.42}$$

By replacing the quantities by their perturbation models, and taking $\left(\widehat{\boldsymbol{\omega}}_{lb}^b \approx \widetilde{\boldsymbol{\omega}}_{ib}^b\right)$, we have:

$$\begin{aligned}\boldsymbol{v}_w^b + \delta\boldsymbol{v}_w^b = \left(\boldsymbol{D} + \begin{bmatrix}0 & 0 & 0\\ 0 & \delta k_{wss} & 0\\ 0 & 0 & 0\end{bmatrix}\right)\left(\boldsymbol{C}_{l2b}(\boldsymbol{I}_3 + \boldsymbol{E}_{b2l})\left(\boldsymbol{v}_b^l + \delta\boldsymbol{v}_b^l\right)\right. \\ \left. + \left(\boldsymbol{\omega}_{ib}^b + \boldsymbol{W}\boldsymbol{k}_\omega + \boldsymbol{b}_\omega\right)\times\Delta\boldsymbol{p}_w^b\right)\end{aligned} \tag{C.43}$$

Thus, keeping only the first order errors, we have:

$$\begin{aligned}\delta \boldsymbol{v}_w^b &= \boldsymbol{D}\boldsymbol{C}_{l2b}\delta \boldsymbol{v}_b^l + \boldsymbol{D}\boldsymbol{C}_{l2b}\boldsymbol{E}_{b2l}\boldsymbol{v}_b^l + \boldsymbol{D}(\boldsymbol{W}\boldsymbol{k}_\omega)\times \Delta \boldsymbol{p}_w^b + \boldsymbol{D}(\boldsymbol{b}_\omega)\times \Delta \boldsymbol{p}_w^b \\ &\quad + \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\widehat{\boldsymbol{\omega}}_{lb}^b\right)\times \Delta \boldsymbol{p}_w^b\right)\delta k_w \\ &= \boldsymbol{D}\boldsymbol{C}_{l2b}\delta \boldsymbol{v}_b^l - \boldsymbol{D}\boldsymbol{C}_{l2b}\left(\boldsymbol{v}_b^l\right)\times \boldsymbol{\varepsilon}_{b2l} - \boldsymbol{D}(\Delta \boldsymbol{p}_w^b)\times \boldsymbol{W}\boldsymbol{k}_\omega \\ &\quad - \boldsymbol{D}(\Delta \boldsymbol{p}_w^b)\times \boldsymbol{b}_\omega \times + \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\widehat{\boldsymbol{\omega}}_{lb}^b\right)\times \Delta \boldsymbol{p}_w^b\right)\delta k_w\end{aligned} \tag{C.44}$$

Therefore:

$$\boldsymbol{H}_w = [\boldsymbol{0}_3 \quad \boldsymbol{H}_{w,v} \quad \boldsymbol{H}_{w,\varepsilon_{b2l}} \quad \boldsymbol{0}_3 \quad \boldsymbol{H}_{w,b_\omega} \quad \boldsymbol{0}_3 \quad \boldsymbol{H}_{w,k_\omega} \quad \boldsymbol{H}_{w,k_w} \quad \boldsymbol{0}_{3\times 10} \quad \{\boldsymbol{0}_{3\times 2}\}] \tag{C.45}$$

where

- $\boldsymbol{H}_{w,v} = \widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{C}}_{l2b}$
- $\boldsymbol{H}_{w,\varepsilon_{b2l}} = -\widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{C}}_{l2b}\left(\widehat{\boldsymbol{v}}_b^l\right)\times$
- $\boldsymbol{H}_{w,b_\omega} = -\widehat{\boldsymbol{D}}\,(\Delta \boldsymbol{p}_w^b)\times$
- $\boldsymbol{H}_{w,k_\omega} = -\widehat{\boldsymbol{D}}\,(\Delta \boldsymbol{p}_w^b)\times \boldsymbol{W}$
- $\boldsymbol{H}_{w,k_w} = \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\left(\widehat{\boldsymbol{C}}_{l2b}\widehat{\boldsymbol{v}}_b^l + \left(\boldsymbol{\omega}_{lb}^b\right)\times \Delta \boldsymbol{p}_w^b\right)$

Appendix D

Proofs of observability analysis

All the proofs are based on the observability matrix expression defined in equation (4.61) and Table 4.2 which expresses the observability matrix components. To simplify reading, let us recall their expressions :

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \nabla L^0 \boldsymbol{h}_1 \\ \nabla L^0 \boldsymbol{h}_2 \\ \nabla L^0 h_{32} \\ \nabla L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \\ \nabla L^0 h_6 \\ \nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_2 \\ \nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \\ \nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \end{bmatrix} \tag{D.1}$$

	$\boldsymbol{p}_b^l$	$\boldsymbol{v}_b^l$	$\boldsymbol{q}_{b2l}$	$\boldsymbol{b}_f$	$\boldsymbol{b}_\omega$	$\boldsymbol{k}_f$	$\boldsymbol{k}_\omega$	k_w
$\nabla L^0 \boldsymbol{h}_1$	$\boldsymbol{I}_3$	$\boldsymbol{0}_3$	$\boldsymbol{g}_1(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{0}_{3\times1}$
$\nabla L^0 \boldsymbol{h}_2$	$\boldsymbol{0}_3$	$\boldsymbol{I}_3$	$\boldsymbol{g}_2(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{g}_3(\boldsymbol{X})$	$\boldsymbol{0}_3$	$\boldsymbol{g}_4(\boldsymbol{X})$	$\boldsymbol{0}_{3\times1}$
$\nabla L^0 h_{32}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_5(\boldsymbol{X})$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_6(\boldsymbol{X})$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{g}_7(\boldsymbol{X})$	$g_8(\boldsymbol{X})$
$\nabla L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_9(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{10}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times1}$
$\nabla L^0 h_6$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$2\boldsymbol{q}_{b2l}^T$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	$\boldsymbol{0}_{1\times3}$	0
$\nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_2$	$\boldsymbol{0}_3$	$\boldsymbol{0}_3$	$\boldsymbol{g}_{11}(\boldsymbol{X})$	$\boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f)$	$\boldsymbol{g}_{12}(\boldsymbol{X})$	$\boldsymbol{g}_{13}(\boldsymbol{X})$	$\boldsymbol{g}_{14}(\boldsymbol{X})$	0
$\nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{15}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{16}(\boldsymbol{X})$	$\boldsymbol{g}_{17}(\boldsymbol{X})$	$\boldsymbol{g}_{18}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times1}$
$\nabla L_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{19}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{0}_{9\times3}$	$\boldsymbol{g}_{20}(\boldsymbol{X})$	$\boldsymbol{g}_{21}(\boldsymbol{X})$	$\boldsymbol{0}_{9\times1}$

Table D.1 Expression of the $\boldsymbol{\mathcal{O}}$ matrix

D.1 Step 1: Observability analysis of the WSS scale factor $\boldsymbol{k_w}$

The WSS measurement expression is given by:

$$h_{32} = (1 + k_w)\left(\boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l + l_{vy}\right) \tag{D.2}$$

Where l_{vy} is the y-component of $\boldsymbol{l}_v = \left(\boldsymbol{\omega}_{ib}^b\right)_{true} \times \Delta \boldsymbol{p}_w^b$ representing the lever arm between the WSS velocity computation point and the IMU.

Therefore;

$$\nabla L^0 h_{32} = [\mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_5(\boldsymbol{X}) \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_6(\boldsymbol{X}) \quad \mathbf{0}_{1\times3} \quad \boldsymbol{g}_7(\boldsymbol{X}) \quad \boldsymbol{g}_8(\boldsymbol{X})] \tag{D.3}$$

where

$$\boldsymbol{g}_8(\boldsymbol{X}) = \boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l + l_{vy} \tag{D.4}$$

This expression is equal in two possible cases:

- Either the vehicle does not move ($\boldsymbol{v}_b^l = 0$ and $l_{vy} = 0$)
- Or $\boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l = -l_{vy}$. This implies that $\left\|\boldsymbol{v}_b^l\right\|_2 = \left|l_{vy}\right| = \left|\omega_{ib}^{bz}\,\Delta p_w^{bx} - \omega_{ib}^{bx}\,\Delta p_w^{bz}\right|$ with ω_{ib}^{bx} very close to zero for a ground vehicle application and Δp_w^{bx} is also close to zero since the IMU is often mounted along the longitudinal axis of the vehicle, which is the same axis along which lies the rear axle center. Therefore, the condition $\boldsymbol{C}_{l2b}(2,:)\boldsymbol{v}_b^l = -l_{vy}$ implies that $\left\|\boldsymbol{v}_b^l\right\| \approx 0$.

Thus, we conclude that the only possibility that $\boldsymbol{g}_8(\boldsymbol{X})$ is not full rank (null) is to have a stationary vehicle. In this case, the WSS scale factor is not locally weakly observable.

Otherwise, when the vehicle moves, the WSS scale factor is locally weakly observable.

D.2 Step 2: Observability analysis of the attitude $\boldsymbol{q_{b2l}}$ and the accelerometers scale factors $\boldsymbol{k_f}$

The GNSS velocity measurement expression is given by:

$$\boldsymbol{h}_2 = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta \boldsymbol{p}_G^b \tag{D.5}$$

Therefore;

$$\begin{aligned} \nabla L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 &= \nabla(\nabla L^0 \boldsymbol{h}_2 . \boldsymbol{f}_2) \\ &= \nabla\left(\boldsymbol{C}_{b2l}\gamma\left(\boldsymbol{k}_f\right)\right) \\ \nabla L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 &= [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_9(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{10}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times1}] \end{aligned} \tag{D.6}$$

where $\boldsymbol{\gamma}\left(\boldsymbol{k}_f\right) = \text{diag}\left(\frac{1}{1+k_{fx}}, \frac{1}{1+k_{fy}}, \frac{1}{1+k_{fz}}\right) = \text{diag}\left(\gamma_{f_x}, \gamma_{f_y}, \gamma_{f_z}\right)$

$$\boldsymbol{g}_9(\boldsymbol{X}) = \frac{\partial}{\partial \boldsymbol{q}_{b2l}}\left(\boldsymbol{C}_{b2l}\boldsymbol{\gamma}\left(\boldsymbol{k}_f\right)\right) \tag{D.7}$$

Considering the expression of $\boldsymbol{C}_{b2l}$ as a function of the $\boldsymbol{q}_{b2l}$ components given in equation (B.35), we compute $\boldsymbol{g}_9$ and $\boldsymbol{g}_{10}$:

$$\boldsymbol{g}_9(\boldsymbol{X}) = \begin{bmatrix} 0 & 0 & -4\gamma_{f_x} q_2 & -4\gamma_{f_x} q_3 \\ 2\gamma_{f_x} q_3 & 2\gamma_{f_x} q_2 & 2\gamma_{f_x} q_1 & 2\gamma_{f_x} q_0 \\ -2\gamma_{f_x} q_2 & 2\gamma_{f_x} q_3 & -2\gamma_{f_x} q_0 & 2\gamma_{f_x} q_1 \\ -2\gamma_{f_y} q_3 & 2\gamma_{f_y} q_2 & 2\gamma_{f_y} q_1 & -2\gamma_{f_y} q_0 \\ 0 & -4\gamma_{f_y} q_1 & 0 & -4\gamma_{f_y} q_3 \\ 2\gamma_{f_y} q_1 & 2\gamma_{f_y} q_0 & 2\gamma_{f_y} q_3 & 2\gamma_{f_y} q_2 \\ 2\gamma_{f_z} q_2 & 2\gamma_{f_z} q_3 & 2\gamma_{f_z} q_0 & 2\gamma_{f_z} q_1 \\ -2\gamma_{f_z} q_1 & -2\gamma_{f_z} q_0 & 2\gamma_{f_z} q_3 & 2\gamma_{f_z} q_2 \\ 0 & -4\gamma_{f_z} q_1 & -4\gamma_{f_z} q_2 & 0 \end{bmatrix} \tag{D.8}$$

and

$$\boldsymbol{g}_{10}(\boldsymbol{X}) = \begin{bmatrix} -\gamma_{f_x}^2(1 - 2q_2^2 - 2q_3^2) & 0 & 0 \\ -2\gamma_{f_x}^2(q_1 q_2 + q_0 q_3) & 0 & 0 \\ -2\gamma_{f_x}^2(q_1 q_3 - q_0 q_2) & 0 & 0 \\ 0 & -2\gamma_{f_y}^2(q_1 q_2 - q_0 q_3) & 0 \\ 0 & -\gamma_{f_y}^2(1 - 2q_1^2 - 2q_3^2) & 0 \\ 0 & -2\gamma_{f_y}^2(q_0 q_1 + q_2 q_3) & 0 \\ 0 & 0 & -2\gamma_{f_z}^2(q_0 q_2 + q_1 q_3) \\ 0 & 0 & -2\gamma_{f_z}^2(q_2 q_3 - q_0 q_1) \\ 0 & 0 & -\gamma_{f_z}^2(1 - 2q_1^2 - 2q_2^2) \end{bmatrix} \tag{D.9}$$

The $\boldsymbol{q}_{b2l}$ quaternion norm constraint is given by:

$$h_6 = \boldsymbol{q}_{b2l}^T . \boldsymbol{q}_{b2l} \tag{D.10}$$

Therefore,

$$\nabla L^0 h_6 = [\mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad 2\boldsymbol{q}_{b2l}^T \quad \mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad \mathbf{0}_{1\times3} \quad 0] \tag{D.11}$$

Let us prove that the matrix $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{g}_9(\boldsymbol{X}) & \boldsymbol{g}_{10}(\boldsymbol{X}) \\ 2q_{b2l}^T & \mathbf{0}_{1\times3} \end{bmatrix}$ has full column rank (equal to 7). To prove this, we have to prove that the rank of at least one submatrix of $\boldsymbol{A}$ is 7. To do this, we compute the determinant of a subset of square submatrices of $\boldsymbol{A}$. These submatrices are formed by choosing a combination of 7 rows from the 10 rows of $\boldsymbol{A}$. The last row which expresses the quaternion norm constraint is kept for all the chosen submatrices.

The first three rows of $\boldsymbol{A}$ correspond to acceleration along the x-axis of (b), rows 4 to 6 correspond to acceleration along the y-axis of (b) and finally rows 7 to 9 correspond to acceleration along z-axis of (b).

We notice that we must have the three accelerations excited in order to have a full rank matrix $\boldsymbol{A}$. In fact, with two accelerations (the minimum number of accelerations since we must choose 6 rows apart from the seventh row corresponding to the quaternion row constraint), the $\boldsymbol{A}$ matrix is rank deficient and the rank of $\boldsymbol{A}$ is equal to 6. The missing rank corresponds to the IMU acceleration scale factor of the missing axis. Therefore, in our case of vehicular applications, the scale factor k_{f_x} is weakly observable when the vehicle turns, only. The other scale factors are observable thanks to the vehicle displacement and the gravity effect.

Now let us consider the submatrix of $\boldsymbol{A}$ after removing the rows corresponding to the cross acceleration along the x-axis, and let us prove that the attitude is locally weakly observable with at least two acceleration axes excited. This means that we must prove that the rank of $\boldsymbol{A}_1$ is equal to 6 with $\boldsymbol{A}_1$ given by:

$$\boldsymbol{A}_1 = \begin{bmatrix} -2\gamma_{f_y}q_3 & 2\gamma_{f_y}q_2 & 2\gamma_{f_y}q_1 & -2\gamma_{f_y}q_0 & -2\gamma_{f_y}^2(q_1q_2 - q_0q_3) & 0 \\ 0 & -4\gamma_{f_y}q_1 & 0 & -4\gamma_{f_y}q_3 & -\gamma_{f_y}^2(1 - 2q_1^2 - 2q_3^2) & 0 \\ 2\gamma_{f_y}q_1 & 2\gamma_{f_y}q_0 & 2\gamma_{f_y}q_3 & 2\gamma_{f_y}q_2 & -2\gamma_{f_y}^2(q_0q_1 + q_2q_3) & 0 \\ 2\gamma_{f_z}q_2 & 2\gamma_{f_z}q_3 & 2\gamma_{f_z}q_0 & 2\gamma_{f_z}q_1 & 0 & -2\gamma_{f_z}^2(q_0q_2 + q_1q_3) \\ -2\gamma_{f_z}q_1 & -2\gamma_{f_z}q_0 & 2\gamma_{f_z}q_3 & 2\gamma_{f_z}q_2 & 0 & -2\gamma_{f_z}^2(q_2q_3 - q_0q_1) \\ 0 & -4\gamma_{f_z}q_1 & -4\gamma_{f_z}q_2 & 0 & 0 & -\gamma_{f_z}^2(1 - 2q_1^2 - 2q_2^2) \\ 2q_0 & 2q_1 & 2q_2 & 2q_3 & 0 & 0 \end{bmatrix} \tag{D.12}$$

Let us consider the following submatrices of $\boldsymbol{A}_1$ and let us compute their determinants:

$$\begin{aligned} \det 1 &= \det(\boldsymbol{A}_1\{1,2,3,4,5,7\}) = -32(q_1q_0 + q_2q_3)\gamma_{f_y}^4\gamma_{f_z}^3 \\ \det 2 &= \det(\boldsymbol{A}_1\{1,2,3,4,6,7\}) = -16(q_1^2 - q_2^2 + q_3^2 - q_0^2)\gamma_{f_y}^4\gamma_{f_z}^3 \\ \det 3 &= \det(\boldsymbol{A}_1\{1,2,4,5,6,7\}) = -16(q_1^2 + q_2^2 - q_3^2 - q_0^2)\gamma_{f_y}^3\gamma_{f_z}^4 \\ \det 4 &= \det(\boldsymbol{A}_1\{1,3,4,5,6,7\}) = -32(q_1q_0 - q_2q_3)\gamma_{f_y}^3\gamma_{f_z}^4 \end{aligned} \tag{D.13}$$

To prove that $\boldsymbol{A}_1$ has full column rank, it suffices to prove that $\{\exists\, i \in \{1, \cdots, 4\}\ ;\ \det i \neq 0\}$

To do so, let us conduct a proof by contradiction and let us assume that $\{\forall\, i \in \{1, \cdots, 4\}\ ;\ \det i = 0\}$.We know that $\gamma_{f_y} \neq 0$ and $\gamma_{f_z} \neq 0$, therefore:

$$\begin{aligned} \det 1 = 0 \ &\rightarrow\ q_1q_0 = -q_2q_3 \\ \det 2 = 0 \ &\rightarrow\ q_1^2 + q_3^2 = q_2^2 + q_0^2 \\ \det 3 = 0 \ &\rightarrow\ q_1^2 - q_3^2 = -q_2^2 + q_0^2 \\ \det 4 = 0 \ &\rightarrow\ q_1q_0 = q_2q_3 \end{aligned} \tag{D.14}$$

By adding and then subtracting the two lines corresponding to $(\det 2 = 0)$ and to $(\det 3 = 0)$, we have:

$$q_1^2 = q_0^2 \ \text{ and } \ q_2^2 = q_3^2 \tag{D.15}$$

Using the result of $(\det 1 = 0)$, we have:($q_1^2 q_0^2 = q_2^2 q_3^2$) and applying the previously found equalities, we deduce that:

$$q_1^2 = q_2^2 = q_3^2 = q_0^2 \tag{D.16}$$

Now using $(\det 1 = 0)$ and $(\det 4 = 0)$, we deduce that $q_1 q_0 = 0$ which implies that $q_1 = 0$ or $q_0 = 0$. This means that $q_1^2 = q_2^2 = q_3^2 = q_0^2 = 0$ which is contradictory with the fact that the quaternion norm is unit.

Therefore;

$$\{\exists\, i \in \{1, \cdots, 4\}\ ;\ \det i \neq 0\} \tag{D.17}$$

We conclude that $\boldsymbol{A}_1$ has full column rank which means that the attitude and the scale factors of the two excited acceleration axes are locally weakly observable. If only one acceleration axis is exited, then the attitude is not fully observable. This result is confirmed in (Rhee, 2002) where the observability of the linearized GPS/INS system is performed.

D.3 Step 3: Observability analysis of the accelerometers biases $\boldsymbol{b_f}$

We recall the GNSS velocity measurement expression given in equation (D.5):

$$\boldsymbol{h}_2 = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta \boldsymbol{p}_G^b$$

Therefore

$$\begin{aligned} \nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_2 &= \nabla(\nabla L^0 \boldsymbol{h}_2 . \boldsymbol{f}_0) \\ &= \nabla\left(\boldsymbol{C}_{b2l}\gamma\left(\boldsymbol{k}_f\right)\boldsymbol{b}_f\right) \end{aligned} \tag{D.18}$$

$$\nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_2 = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{g}_{11}(\boldsymbol{X}) & \boldsymbol{C}_{b2l}\gamma\left(\boldsymbol{k}_f\right) & \boldsymbol{g}_{12}(\boldsymbol{X}) & \boldsymbol{g}_{13}(\boldsymbol{X}) & \boldsymbol{g}_{14}(\boldsymbol{X}) & 0 \end{bmatrix}$$

This step is very simple since we need to prove that $\boldsymbol{C}_{b2l}\gamma\left(\boldsymbol{k}_f\right)$ is full rank (equal to 3). This expression is the product of a rotation matrix which is invertible and a (3×3) diagonal matrix with nonzero elements. Therefore, the rank of both matrices is 3 and their product has consequently a full rank. Thus, the three accelerometer biases are locally weakly observable if the GNSS provides a 3D velocity information.

D.4 Step 4: Observability analysis of the gyroscopes biases $\boldsymbol{b_\omega}$

We recall the GNSS velocity measurement expression given in equation (D.5):

$$\boldsymbol{h}_2 = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta \boldsymbol{p}_G^b$$

Therefore

$$\begin{aligned}\nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 &= \nabla(\nabla(\nabla L^0 \boldsymbol{h}_2 . \boldsymbol{f}_2) . \boldsymbol{f}_0) \\ &= \nabla\left(\nabla L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 \, . \boldsymbol{f}_0\right) \\ &= \nabla\left(\frac{-1}{2} \boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\, \boldsymbol{q}_{b2l}) \gamma(\boldsymbol{k}_\omega) \boldsymbol{b}_\omega + \boldsymbol{g}_{10}(\boldsymbol{X}) \boldsymbol{\beta}_{k_f} \boldsymbol{k}_f\right)\end{aligned} \tag{D.19}$$

$$\nabla L_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} \boldsymbol{h}_2 = [\boldsymbol{0}_{9\times3} \quad \boldsymbol{0}_{9\times3} \quad \boldsymbol{g}_{15}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times3} \quad \boldsymbol{g}_{16}(\boldsymbol{X}) \quad \boldsymbol{g}_{17}(\boldsymbol{X}) \quad \boldsymbol{g}_{18}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times1}]$$

With

$$\begin{aligned}\boldsymbol{g}_{16}(\boldsymbol{X}) &= -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{b}_\omega}(\boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\, \boldsymbol{q}_{b2l}) \gamma(\boldsymbol{k}_\omega) \boldsymbol{b}_\omega) \\ &= \frac{-1}{2} \boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\, \boldsymbol{q}_{b2l}) \gamma(\boldsymbol{k}_\omega)\end{aligned} \tag{D.20}$$

Let us analyze the rank of $\boldsymbol{g}_{16}(\boldsymbol{X})$ which is the product of three matrices.

$\gamma(\boldsymbol{k}_\omega)$ is a diagonal matrix with nonzero elements and consequently it has full rank. Therefore, it suffices to analyses the rank of the matrix: $\boldsymbol{B} = \frac{-1}{2} \boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\, \boldsymbol{q}_{b2l})$.

Let us conduct the same analysis as in Step 2. We must demonstrate that the rank of $\boldsymbol{B}$ is equal to3. To do this, we compute the determinant of a subset of square submatrices of $\boldsymbol{B}$. These submatrices are formed by choosing a combination of 3 rows from the 9 rows of the matrix $\boldsymbol{B}$. The determinants of these submatrices are computed and we must prove that there exists at least one nonzero determinant.

First of all, when we compute the determinant of $\boldsymbol{B}$ with only one excited accelerometer axis, we obtain a null determinant and the rank of $\boldsymbol{B}$ is 2. Therefore at least 2 accelerometer axes must be excited to be able to estimate the three gyro biases. As explained in Step 2, the x-axis describes the lateral movement of the vehicle and the corresponding acceleration is excited during turns, only. Therefore, only the y and z axes are considered because they correspond respectively to the forward movement and the gravity acceleration effect.

Let us consider the following submatrices of $\boldsymbol{B}$ and let us compute their determinants:

$$\begin{aligned}\det 1 &= \det(\boldsymbol{B}\{5,6,8\}) = -4(q_1^2 q_2^2 - q_3^2 q_0^2) \gamma_{f_y}^2 \gamma_{f_z} \\ \det 2 &= \det(\boldsymbol{B}\{6,8,9\}) = 4(q_1^2 q_3^2 - q_2^2 q_0^2) \gamma_{f_y} \gamma_{f_z}^2 \\ \det 3 &= \det(\boldsymbol{B}\{4,6,7\}) = -(q_1^4 + q_2^4 - q_3^4 - q_0^4 - 2q_1^2 q_2^2 + 2q_3^2 q_0^2)\, \gamma_{f_y}^2 \gamma_{f_z} \\ \det 4 &= \det(\boldsymbol{B}\{4,7,8\}) = -(q_1^4 - q_2^4 + q_3^4 - q_0^4 - 2q_1^2 q_3^2 + 2q_2^2 q_0^2)\, \gamma_{f_y} \gamma_{f_z}^2 \\ \det 5 &= \det(\boldsymbol{B}\{4,5,7\}) = -2(q_1 q_0 + q_2 q_3)(q_1^2 - q_2^2 - q_3^2 + q_0^2) \gamma_{f_y}^2 \gamma_{f_z}\end{aligned} \tag{D.21}$$

$$\det 6 = \det(\boldsymbol{B}\{5,8,9\}) = -4(q_1q_3 + q_2q_0)(q_1q_2 + q_3q_0)\gamma_{f_y}\gamma_{f_z}^2$$

$$\det 7 = \det(\boldsymbol{B}\{5,6,9\}) = -4(q_1q_3 - q_2q_0)(q_1q_2 - q_3q_0)\gamma_{f_y}^2\gamma_{f_z}$$

As for step 2, let us conduct a proof by contradiction and let us assume that $\{\forall\, i \in \{1,\cdots,7\}\ ;\ \det i = 0\}$.We know that $\gamma_{f_y} \neq 0$ and $\gamma_{f_z} \neq 0$, therefore:

$$\begin{aligned} \det 1 = 0 \ \ &\rightarrow \ \ q_1^2q_2^2 = q_3^2q_0^2 \\ \det 2 = 0 \ \ &\rightarrow \ \ q_1^2q_3^2 = q_2^2q_0^2 \end{aligned} \tag{D.22}$$

Using the results of $(\det 1 = 0)$ and $(\det 2 = 0)$ we have,

$$\begin{aligned} \det 3 = 0 \ \ &\rightarrow q_1^4 + q_2^4 - q_3^4 - q_0^4 = 0 \\ \det 4 = 0 \ \ &\rightarrow q_1^4 - q_2^4 + q_3^4 - q_0^4 = 0 \end{aligned} \tag{D.23}$$

By adding and then subtracting the previous equations, we obtain:

$$q_1^4 = q_0^4 \ \text{ and } \ q_2^4 = q_3^4 \ \text{ or equivalently } q_1^2 = q_0^2 \ \text{ and } \ q_2^2 = q_3^2 \tag{D.24}$$

By applying the previously found equalities in the result of $(\det 5 = 0)$, we have:

$$\det 5 = -2(q_1^2 - q_2^2)(q_1q_0 + q_2q_3) = 0 \ \ \rightarrow q_1q_0 = -q_2q_3 \ \ \text{or} \ \ q_1^2 = q_2^2 \tag{D.25}$$

In both cases we deduce that:

$$q_1^2 = q_2^2 = q_3^2 = q_0^2 \tag{D.26}$$

Indeed:

- If $(q_1q_0 = -q_2q_3)$ then($q_1^2q_0^2 = q_2^2q_3^2$) and applying the equalities found in equation (D.24), we obtain equation (D.26)
- If $(q_1^2 = q_2^2 = 0)$ then using equation (D.24), we obtain directly equation (D.26)

Finally, summing and subtracting the results of $(\det 6 = 0)$ and $(\det 7 = 0)$, we deduce that at least one quaternion component is zero. This implies that $(q_1^2 = q_2^2 = q_3^2 = q_0^2 = 0)$ which is contradictory with the fact that the quaternion is a unit vector.

Therefore

$$\{\exists\, i \in \{1,\cdots,7\}\ ;\ \det i \ \neq 0\} \tag{D.27}$$

We conclude that $\boldsymbol{B}$ has full column rank. Therefore, the three gyroscope biases are locally weakly observable if at least two acceleration axes are excited.

D.5 Step 5: Observability analysis of the gyroscopes cales factors k_{ω}

We recall the GNSS velocity measurement expression given in equation (D.5):

$$\boldsymbol{h}_2 = \boldsymbol{v}_b^l + \boldsymbol{C}_{b2l}\left(\gamma(\boldsymbol{k}_\omega)\left(\widetilde{\boldsymbol{\omega}}_{ib}^b - \boldsymbol{b}_\omega\right)\right) \times \Delta\boldsymbol{p}_G^b$$

Therefore;

$$\begin{aligned}\nabla L_{\boldsymbol{f}_1} L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 &= \nabla(\nabla(\nabla L^0 \boldsymbol{h}_2 . \boldsymbol{f}_2) . \boldsymbol{f}_1) \\ &= \nabla\left(\nabla L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 \, . \boldsymbol{f}_1\right) \\ &= \nabla\left(\frac{1}{2}\boldsymbol{g}_9(\boldsymbol{X})\boldsymbol{\Psi}(\,\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\right)\end{aligned} \tag{D.28}$$

$$\nabla L_{\boldsymbol{f}_1} L_{\boldsymbol{f}_2}^1 \boldsymbol{h}_2 = [\boldsymbol{0}_{9\times3} \quad \boldsymbol{0}_{9\times3} \quad \boldsymbol{g}_{19}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times3} \quad \boldsymbol{0}_{9\times3} \quad \boldsymbol{g}_{20}(\boldsymbol{X}) \quad \boldsymbol{g}_{21}(\boldsymbol{X}) \quad \boldsymbol{0}_{9\times1}]$$

With

$$\boldsymbol{g}_{21}(\boldsymbol{X}) = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{k}_\omega}\left(\boldsymbol{g}_9(\boldsymbol{X})\boldsymbol{\Psi}(\,\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\right) \tag{D.29}$$

After removing the zero rows, we obtain the following matrix:

$$\boldsymbol{g}_{21}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{g}_{21}^{cx} & \boldsymbol{0}_{1\times6} & \boldsymbol{0}_{1\times6} \\ \boldsymbol{0}_{1\times6} & \boldsymbol{g}_{21}^{cy} & \boldsymbol{0}_{1\times6} \\ \boldsymbol{0}_{1\times6} & \boldsymbol{0}_{1\times6} & \boldsymbol{g}_{21}^{cz} \end{bmatrix} \tag{D.30}$$

with

$$\boldsymbol{g}_{21}^{cx} = \begin{bmatrix} (2q_1q_3 - 2q_2q_0)\gamma_{\omega_x}^2\gamma_{f_x} \\ (-2q_3q_0 - 2q_1q_2)\gamma_{\omega_x}^2\gamma_{f_x} \\ (2q_1q_0 + 2q_2q_3)\gamma_{\omega_x}^2\gamma_{f_y} \\ (q_1^2 - q_2^2 + q_3^2 - q_0^2)\gamma_{\omega_x}^2\gamma_{f_y} \\ (-q_1^2 - q_2^2 + q_3^2 + q_0^2)\gamma_{\omega_x}^2\gamma_{f_z} \\ (2q_1q_0 - 2q_2q_3)\gamma_{\omega_x}^2\gamma_{f_z} \end{bmatrix}, \quad \boldsymbol{g}_{21}^{cy} = \begin{bmatrix} -(2q_1q_3 - 2q_2q_0)\gamma_{\omega_y}^2\gamma_{f_x} \\ (q_1^2 - q_2^2 - q_3^2 + q_0^2)\gamma_{\omega_y}^2\gamma_{f_x} \\ -(2q_1q_0 + 2q_2q_3)\gamma_{\omega_y}^2\gamma_{f_y} \\ (2q_1q_2 - 2q_3q_0)\gamma_{\omega_y}^2\gamma_{f_y} \\ (q_1^2 + q_2^2 - q_3^2 - q_0^2)\gamma_{\omega_y}^2\gamma_{f_z} \\ (2q_1q_3 + 2q_2q_0)\gamma_{\omega_y}^2\gamma_{f_z} \end{bmatrix}$$

$$\boldsymbol{g}_{21}^{cz} = \begin{bmatrix} (2q_1q_2 + 2q_3q_0)\gamma_{\omega_z}^2\gamma_{f_x} \\ (-q_1^2 + q_2^2 + q_3^2 - q_0^2)\gamma_{\omega_z}^2\gamma_{f_x} \\ (-q_1^2 + q_2^2 - q_3^2 + q_0^2)\gamma_{\omega_z}^2\gamma_{f_y} \\ (2q_3q_0 - 2q_1q_2)\gamma_{\omega_z}^2\gamma_{f_y} \\ (-2q_1q_0 + 2q_2q_3)\gamma_{\omega_z}^2\gamma_{f_z} \\ -(2q_2q_0 + 2q_1q_3)\gamma_{\omega_z}^2\gamma_{f_z} \end{bmatrix} \tag{D.31}$$

Again, the acceleration along the x-axis is not taken into account and the observability is analyzed independently of this axis for the land vehicle. The observability of $\boldsymbol{k}_\omega$ depends also on the gyro measurements. We take into consideration the fact that the land vehicle rotation around the y-axis (the roll) is negligible. In this case, $\boldsymbol{g}_{21}$ is rank deficient and its rank is equal to 2. The missing rank corresponds to the IMU angular rate scale factor about the y-axis. Let us prove that the other gyroscope scale factors are observable with the measurements that are available. First, we define the new matrix $\boldsymbol{G}_{21}^{cx}$ whose rows correspond to the available measurements (2 accelerations along the y and z axes and a rotation about the x axes):

$$\boldsymbol{G}_{21}^{cx} = \begin{bmatrix} 2(q_1q_0 + q_2q_3)\gamma_{\omega_x}^2\gamma_{f_y} \\ (q_1^2 - q_2^2 + q_3^2 - q_0^2)\gamma_{\omega_x}^2\gamma_{f_y} \\ (-q_1^2 - q_2^2 + q_3^2 + q_0^2)\gamma_{\omega_x}^2\gamma_{f_z} \\ 2(q_1q_0 - q_2q_3)\gamma_{\omega_x}^2\gamma_{f_z} \end{bmatrix} \tag{D.32}$$

Let us prove that it has a rank equal to 1.

$$\begin{aligned} \det 1 &= \det(\boldsymbol{G}_{21}^{cx}\{1\}) = 2(q_1q_0 + q_2q_3)\gamma_{\omega_x}^2\gamma_{f_y} \\ \det 2 &= \det(\boldsymbol{G}_{21}^{cx}\{2\}) = (q_1^2 - q_2^2 + q_3^2 - q_0^2)\gamma_{\omega_x}^2\gamma_{f_y} \\ \det 3 &= \det(\boldsymbol{G}_{21}^{cx}\{3\}) = (-q_1^2 - q_2^2 + q_3^2 + q_0^2)\gamma_{\omega_x}^2\gamma_{f_z} \\ \det 4 &= \det(\boldsymbol{G}_{21}^{cx}\{4\}) = 2(q_1q_0 - q_2q_3)\gamma_{\omega_x}^2\gamma_{f_z} \end{aligned} \tag{D.33}$$

Let us assume that $\{\forall\, i \in \{1, \cdots, 4\}\ ;\ \det i = 0\}$.We know that $\gamma_{f_y} \neq 0$,$\gamma_{f_z} \neq 0$ and $\gamma_{\omega_x} \neq 0$ therefore:

$$\begin{aligned} \det 1 = 0 \ \ \rightarrow \ \ q_1q_0 &= -q_2q_3 \\ \det 2 = 0 \ \ \rightarrow \ \ q_1^2 + q_3^2 &= q_2^2 + q_0^2 \\ \det 3 = 0 \ \ \rightarrow \ \ q_1^2 - q_3^2 &= -q_2^2 + q_0^2 \end{aligned} \tag{D.34}$$

By adding and then subtracting $(\det 2 = 0)$ and $(\det 3 = 0)$, we have:

$$q_1^2 = q_0^2 \ \text{ and } \ q_2^2 = q_3^2 \tag{D.35}$$

Using $(\det 1 = 0)$, we have: $q_1^2q_0^2 = q_2^2q_3^2$ and applying the previously found equalities, we deduce that:

$$q_1^2 = q_2^2 = q_3^2 = q_0^2 \tag{D.36}$$

Now using $(\det 1 = 0)$ and $(\det 4 = 0)$, we deduce that $q_1q_0 = 0$ which implies that $q_1 = 0$ or $q_0 = 0$. This means that $q_1^2 = q_2^2 = q_3^2 = q_0^2 = 0$ which is contradictory with the fact that the quaternion norm is unit with the fact that the quaternion norm is unit.

Therefore;

$$\{\exists\, i \in \{1, \cdots, 4\}\ ;\ \det i \ \neq 0\} \tag{D.37}$$

We conclude that $\boldsymbol{G}_{21}^{cx}$ has full column rank and consequently that the gyro scale factor about the x-axis is locally weakly observable.

Now let us prove that $\boldsymbol{G}_{21}^{cz}$ has full rank with:

$$\boldsymbol{G}_{21}^{cz} = \begin{bmatrix} (-q_1^2 + q_2^2 - q_3^2 + q_0^2)\gamma_{\omega_z}^2\gamma_{f_y} \\ 2(q_3q_0 - q_1q_2)\gamma_{\omega_z}^2\gamma_{f_y} \\ -2(q_1q_0 - q_2q_3)\gamma_{\omega_z}^2\gamma_{f_z} \\ -2(q_2q_0 + q_1q_3)\gamma_{\omega_z}^2\gamma_{f_z} \end{bmatrix} \tag{D.38}$$

$$\begin{aligned} \det 1 &= \det(\boldsymbol{G}_{21}^{cz}\{1\}) = (-q_1^2 + q_2^2 - q_3^2 + q_0^2)\gamma_{\omega_z}^2\gamma_{f_y} \\ \det 2 &= \det(\boldsymbol{G}_{21}^{cz}\{2\}) = 2(q_3q_0 - q_1q_2)\gamma_{\omega_z}^2\gamma_{f_y} \\ \det 3 &= \det(\boldsymbol{G}_{21}^{cz}\{3\}) = -2(q_1q_0 - q_2q_3)\gamma_{\omega_z}^2\gamma_{f_z} \\ \det 4 &= \det(\boldsymbol{G}_{21}^{cz}\{4\}) = -2(q_2q_0 + q_1q_3)\gamma_{\omega_z}^2\gamma_{f_z} \end{aligned} \tag{D.39}$$

Let us assume that $\{\forall\, i \in \{1, \cdots, 4\}\ ;\ \det i = 0\}$.We know that $\gamma_{f_y} \neq 0$,$\gamma_{f_z} \neq 0$ and $\gamma_{\omega_z} \neq 0$ therefore:

$$\begin{aligned} \det 1 = 0 \ &\rightarrow \quad q_1^2 + q_3^2 = q_2^2 + q_0^2 \\ \det 2 = 0 \ &\rightarrow \quad q_3q_0 = q_1q_2 \ \rightarrow \quad q_3^2q_0^2 = q_1^2q_2^2 \\ \det 3 = 0 \ &\rightarrow \quad q_1q_0 = q_2q_3 \ \rightarrow \quad q_1^2q_0^2 = q_2^2q_3^2 \\ \det 4 = 0 \ &\rightarrow \quad q_2q_0 = -q_1q_3 \end{aligned} \tag{D.40}$$

By summing $(\det 2 = 0)$ and $(\det 3 = 0)$, and using $(\det 1 = 0)$, we have:

$$(q_2^2 + q_0^2)(q_2^2 - q_0^2) = 0 \text{ and } (q_1^2 + q_3^2)(q_1^2 - q_3^2) = 0 \tag{D.41}$$

which implies

$$q_2^4 = q_0^4 \text{ or equivalently } q_2^2 = q_0^2 \text{ and } q_1^4 = q_3^4 \text{ or equivalently } q_1^2 = q_3^2 \tag{D.42}$$

Using the previously found equalities in $(\det 1 = 0)$, we obtain:

$$q_1^2 = q_2^2 \tag{D.43}$$

and we deduce that

$$q_1^2 = q_2^2 = q_3^2 = q_0^2 \tag{D.44}$$

Let us assume that $q_1 \neq 0$, then multiplying $(\det 3 = 0)$ by q_1 and using $(\det 4 = 0)$, we have:

$$q_1^2q_0 = q_1q_2q_3 = -q_2q_0q_2 = -q_2^2q_0 \tag{D.45}$$

This implies that:

$$(q_1^2 + q_2^2)q_0 = 0 \tag{D.46}$$

This means that $q_2 = 0$ or $q_0 = 0$. In both cases this implies using equation (D.44) that

$$q_1 = q_2 = q_3 = 0 \tag{D.47}$$

which is contradictory with the unit norm of the quaternion. Therefore $q_1 = 0$ and consequently all the other quaternion components are equal zero which is again contradictory with the quaternion unit norm.

Finally, we conclude that $\{\exists\ i \in \{1, \cdots, 4\}\ ;\ \det i \neq 0\}$ and that the gyro scale factor about the z-axis is locally weakly observable.

D.6 Note on the observability using only position information

It can be noticed in the observability analysis done that the GNSS velocity was the most used. However, the observability analysis applies also to any velocity information provided to the filter if the GNSS velocity is not available (for example the WSS velocity). Note that in this study, it was intended to not use the lever arm information between the sensors to study specifically the contribution of each measurement without the lever arm contribution. The lever arm information includes the attitude of the vehicle and its presence increases the observability of the attitude.

However, if the position is the only available information, then let us have a look at the system observability using this information:

$$\boldsymbol{h}_1 = \boldsymbol{p}_b^l + \boldsymbol{C}_{b2l}\ \Delta\boldsymbol{p}_G^b \tag{D.48}$$

Therefore

$$\nabla L^0 \boldsymbol{h}_1 = [\boldsymbol{I}_3 \quad \boldsymbol{0}_3 \quad \boldsymbol{g}_1(\boldsymbol{X}) \quad \boldsymbol{0}_3 \quad \boldsymbol{0}_3 \quad \boldsymbol{0}_3 \quad \boldsymbol{0}_3 \quad \boldsymbol{0}_{3\times1}] \tag{D.49}$$

where $\boldsymbol{g}_1(\boldsymbol{X})$ is the component resulting from the lever arm presence.

The first Lie derivative of $\boldsymbol{h}_1$ along $\boldsymbol{f}_0$ brings up the velocity term that we have used to analyze the observability in the previous section:

$$L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_1 = \nabla L^0 \boldsymbol{h}_1 . \boldsymbol{f}_0 = \boldsymbol{v}_b^l - \frac{1}{2}\boldsymbol{g}_1(\boldsymbol{X})\boldsymbol{\Psi}(\boldsymbol{q}_{b2l})\gamma(\boldsymbol{k}_\omega)\boldsymbol{b}_\omega \tag{D.50}$$

Therefore

$$\nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_1 = [\boldsymbol{0}_3 \quad \boldsymbol{I}_3 \quad \boldsymbol{g}_{22}(\boldsymbol{X}) \quad \boldsymbol{0}_3 \quad \boldsymbol{g}_{23}(\boldsymbol{X}) \quad \boldsymbol{0}_3 \quad \boldsymbol{g}_{24}(\boldsymbol{X}) \quad \boldsymbol{0}_{3\times1}] \tag{D.51}$$

which proves that the velocity is locally weakly observable.

We show that the same conclusions as in the previous section can be drawn thanks to $\nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_1$.

<u>Correspondence 1</u>:

$$L_{\boldsymbol{f}_2}^1 L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_1 = \nabla L_{\boldsymbol{f}_0}^1 \boldsymbol{h}_1 . \boldsymbol{f}_2 = \boldsymbol{C}_{b2l}\gamma(\boldsymbol{k}_f) \tag{D.52}$$

Therefore

$$\nabla L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 = [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_9(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{10}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times1}] \tag{D.53}$$

This shows the attitude error and the accelerometer scale factors observability as detailed in Step 2.

Correspondence 2:

$$L^2_{\boldsymbol{f}_0} \boldsymbol{h}_1 = \nabla L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 . \boldsymbol{f}_0 = \boldsymbol{C}_{b2l} \gamma(\boldsymbol{k}_f) \boldsymbol{b}_f \tag{D.54}$$

Therefore

$$\nabla L^2_{\boldsymbol{f}_0} \boldsymbol{h}_1 = [\mathbf{0}_3 \quad \mathbf{0}_3 \quad \boldsymbol{g}_{11}(\boldsymbol{X}) \quad \boldsymbol{C}_{b2l} \gamma(\boldsymbol{k}_f) \quad \boldsymbol{g}_{12}(\boldsymbol{X}) \quad \boldsymbol{g}_{13}(\boldsymbol{X}) \quad \boldsymbol{g}_{14}(\boldsymbol{X}) \quad 0] \tag{D.55}$$

This shows the accelerometer biases observability as detailed in Step 3.

Correspondence 3:

$$L^1_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 = \nabla L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 . \boldsymbol{f}_0 = \frac{-1}{2} \boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\boldsymbol{q}_{b2l}) \gamma(\boldsymbol{k}_\omega) \boldsymbol{b}_\omega + \boldsymbol{g}_{10}(\boldsymbol{X}) \boldsymbol{\beta}_{k_f} \boldsymbol{k}_f \tag{D.56}$$

Therefore

$$\begin{aligned} &\nabla L^1_{\boldsymbol{f}_0} L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 \\ &= [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{15}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{16}(\boldsymbol{X}) \quad \boldsymbol{g}_{17}(\boldsymbol{X}) \quad \boldsymbol{g}_{18}(\boldsymbol{X}) \quad \mathbf{0}_{9\times1}] \end{aligned} \tag{D.57}$$

This proves the gyroscope biases observability as detailed in Step 4

Correspondence 4:

$$L^1_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 = \nabla L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 . \boldsymbol{f}_1 = \frac{1}{2} \boldsymbol{g}_9(\boldsymbol{X}) \boldsymbol{\Psi}(\boldsymbol{q}_{b2l}) \gamma(\boldsymbol{k}_\omega) \tag{D.58}$$

Therefore

$$\nabla L^1_{\boldsymbol{f}_1} L^1_{\boldsymbol{f}_2} L^1_{\boldsymbol{f}_0} \boldsymbol{h}_1 = [\mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{19}(\boldsymbol{X}) \quad \mathbf{0}_{9\times3} \quad \mathbf{0}_{9\times3} \quad \boldsymbol{g}_{20}(\boldsymbol{X}) \quad \boldsymbol{g}_{21}(\boldsymbol{X}) \quad \mathbf{0}_{9\times1}] \tag{D.59}$$

This proves the gyroscope scale factors observability as detailed in Step 5.

Résumé

Les exigences en termes de précision, intégrité, continuité et disponibilité de la navigation terrestre, consistant à estimer la position, la vitesse et l'attitude d'un véhicule, sont de plus en plus strictes, surtout depuis le développement des véhicules autonomes. Ce type d'applications nécessite un système de navigation non seulement capable de fournir une solution de navigation précise et fiable, mais aussi ayant un coût raisonnable. Durant les dernières décennies, les systèmes de navigation par satellites (GNSS) ont été les plus utilisés pour la navigation, surtout avec la baisse continue des coûts des récepteurs. Cependant, malgré sa capacité à fournir des informations de navigation absolue avec une bonne précision dans des milieux dégagés, l'utilisation du GNSS dans des milieux contraints est limitée à cause des problèmes liés à la propagation des signaux. Ce problème peut être surmonté en fusionnant les bonnes mesures GNSS avec les mesures d'autres capteurs ayant des caractéristiques complémentaires. Les algorithmes d'hybridation les plus largement mis en œuvre pour les véhicules terrestres fusionnent les mesures GNSS avec des données inertielles et / ou odométriques. Cependant, les performances obtenues par cette hybridation dépendent énormément de la qualité du capteur inertiel/odométrique utilisé, surtout lorsque les signaux GNSS sont dégradés ou indisponibles. Par conséquent, cette thèse, vise à enrichir l'architecture d'hybridation en incluant d'autres mesures de capteurs capables d'améliorer les performances de navigation tout en disposant d'un système bas coût et facilement embarquable. C'est pourquoi l'utilisation de la technique de navigation SLAM basées sur la vision pour fournir des informations supplémentaires est proposée dans cette thèse. Par conséquent, ce travail se concentre sur le développement d'une architecture de fusion multi-capteurs fusionnant l'information visuelle fournie par le SLAM avec les capteurs précédemment mentionnés et étudie en particulier la contribution de l'utilisation de cette information pour améliorer les performances du système de navigation.

Mots-clés : fusion multi-capteurs, navigation par vision, GNSS, INS, SLAM, filtre de Kalman

Abstract

For land vehicles, the requirements of the navigation solution in terms of accuracy, integrity, continuity and availability are more and more stringent, especially with the development of autonomous vehicles. This type of application requires a navigation system not only capable of providing an accurate and reliable position, velocity and attitude solution continuously but also having a reasonable cost. In the last decades, GNSS has been the most widely used navigation system especially with the receivers decreasing cost over the years. However, despite of its capability to provide absolute navigation information, this system suffers from problems related to signal propagation especially in urban environments where buildings, trees and other structures hinder the reception of GNSS signals and degrade their quality. A possible way to overcome these problems is to fuse good GNSS measurements with other sensors having complementary characteristics. Generally, the most widely implemented hybridization algorithms for land vehicles fuse GNSS measurements with inertial and/or odometric data. However, the performance achieved by this hybridization depends thoroughly on the quality of the inertial/odometric sensor used especially when GNSS signals are degraded or unavailable. Therefore, this Ph.D. thesis, aims at extending the classical hybridization architecture by including other sensors capable of improving the navigation performances while having a low cost and being easily embeddable. For this reason, the use of vision-based navigation techniques to provide additional information is proposed in this thesis. In particular, the SLAM technique is investigated. Therefore, this work focuses on developing a multi-sensor fusion architecture integrating visual information with the previously mentioned sensors. In particular, the study of the contribution of this information to improve the vision-free navigation system performance is perfomrmed.

Keywords : Multi-sensor fusion, vision-based navigation, GNSS, INS, SLAM, Kalman filter