



HAL
open science

Autonomous Integrity Monitoring of Navigation Maps on board Intelligent Vehicles

Clément Zinoune

► **To cite this version:**

Clément Zinoune. Autonomous Integrity Monitoring of Navigation Maps on board Intelligent Vehicles. Robotics [cs.RO]. UTC Compiègne, 2014. English. NNT: . tel-01092628v1

HAL Id: tel-01092628

<https://hal.science/tel-01092628v1>

Submitted on 9 Dec 2014 (v1), last revised 30 May 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

THESIS

Submitted for the degree of

Doctor of Philosophy

Area of specialization: Information Technologies and Systems

by

Clément ZINOUNE

**Autonomous Integrity Monitoring of
Navigation Maps on board
Intelligent Vehicles**

Heudiasyc Laboratory, UMR UTC/CNRS 7253
and Renault S.A.

Defended on the Eleventh of September, 2014

Thesis Committee:

President:	Isabelle Fantoni	Université de Technologie de Compiègne
Reviewers:	Eduardo Nebot	The University of Sydney - Australia
	François Peyret	IFSTTAR Nantes
Examiners:	Pierre-Yves Gilliéron	EPFL - Switzerland
	Jean-Philippe Lauffenburger	Université de Haute Alsace
	Mohammed A. Quddus	Loughborough University - United Kingdom
Supervisors:	Philippe Bonnifait	Université de Technologie de Compiègne
	Javier Ibañez-Guzmán	Renault S.A.

À ma Mère.

Remerciements

Je tiens avant tout à remercier Javier et Philippe grâce à qui cette thèse s'est déroulée dans de bonnes conditions. Merci pour le temps que vous avez accordé à nos discussions sur l'importance des cartes encore sous-estimée dans la communauté, sur les différents concepts scientifiques que l'on pourrait appliquer à leur étude ainsi qu'aux épineuses questions de vocabulaire. Vous représentez chacun une face de la recherche et pour chacune d'elles, les questions soulevées vis-à-vis d'une solution proposée : la recherche industrielle (*Pourquoi faire* et *Combien cela coûte-t-il*) et la recherche académique (*Comment*). Vous avez su me faire profiter de ces deux visions tout en évitant les blocages idéologiques.

Je remercie aussi tout particulièrement les membres de mon jury pour avoir accepté de m'accorder de leur temps et pour leurs précieux avis sur mes travaux.

Cette thèse s'est déroulée en trois temps et deux lieux : le Technocentre Renault à Guyancourt pour la première et la dernière année et le laboratoire Heudiasyc à Compiègne pour la deuxième année. J'ai eu la chance de toujours côtoyer des personnes qui m'ont donné l'envie d'aller travailler tous les jours.

Je suis très reconnaissant à Charles, Sébastien et Johann qui m'ont accueilli à Renault et grâce à qui j'ai pu travailler dans une bonne ambiance. Malgré mon statut particulier de doctorant, j'ai eu le sentiment de faire pleinement partie de leur équipe. Ceci a aussi été rendu possible grâce aux autres membres de cette équipe. Nous avons fait nos premières armes sur les véhicules et démêlé des kilomètres de câbles dans l'Espace avec Nabil et Jeremy. Nous avons formé une solide équipe de doctorants à travers nos galères et nos bons moments avec Stéphanie, Laetitia et Alexandre. Merci aussi à Jong-Hoon, Victor et tous les membres des équipes ADAS Amont et Développement pour leur bonne humeur et leur accueil.

Les moments passés à Compiègne ont aussi été très agréables et pleins d'enseignements. Je tiens donc à remercier Julien, Vincent, Farah, Nicole, Adam, Felipe, Hoda, Gilles pour leur accueil chaleureux au sein de la grande équipe des doctorants du laboratoire. Les chercheurs ainsi que les équipes administratives ont aussi largement contribué à cette qualité de travail (Béregère, Nathalie, Vincent, Véronique, Gérard, Gabriel).

J'ai mis ma thèse en pause pendant six mois pour cause de maladie. Toutes ces personnes m'ont aussi permis de rester connecté au travail durant cette période. J'ai pu garder une oreille ouverte sur ce qui se passait à Renault et à l'UTC tout en étant préservé de toute tâche à accomplir. Ceci m'a permis de me consacrer à ma

guérison et de faire un retour en douceur au travail lorsque la santé fut revenue. Je n'aurais pas été capable de terminer cette thèse sans la compétence et la qualité de suivi du Dr Cereja et des équipes infirmières du Centre Hospitalier Sud Francilien.

Je pense que l'on ne peut s'épanouir pleinement que dans une famille unie. C'est aussi grâce à eux que j'en suis arrivé là. Merci à ma Grand Mère pour nous avoir accueilli Nabil et moi lors d'une campagne d'acquisition de données qui ont servi à l'élaboration d'une grande partie de ces travaux. Mes parents et mon petit frère qui m'ont soutenu pendant toutes mes études même si cela n'a pas toujours été facile. Ma mère qui m'a soutenu pendant ma première année à Compiègne et qui n'est malheureusement pas là pour en voir le résultat. Je remercie finalement Julie pour avoir adapté sa vie autour de ma thèse pendant ces trois ans et pour m'avoir supporté au jour le jour et pendant les moments difficiles.

Abstract

Several Intelligent Vehicles capabilities from Advanced Driving Assistance Systems (ADAS) to Autonomous Driving functions depend on *a priori* information provided by navigation maps. Whilst these were intended for driver guidance as they store road network information, today they are even used in applications that control vehicle motion. In general, the vehicle position is projected onto the map to relate with links in the stored road network. However, maps might contain faults, leading to navigation and situation understanding errors. Therefore, the integrity of the map-matched estimates must be monitored to avoid failures that can lead to hazardous situations. The main focus of this research is the real-time autonomous evaluation of faults in navigation maps used in intelligent vehicles.

Current passenger vehicles are equipped with proprioceptive sensors that allow estimating accurately the vehicle state over short periods of time rather than long trajectories. They include receiver for Global Navigation Satellite System (GNSS) and are also increasingly equipped with exteroceptive sensors like radar or smart camera systems. The challenge resides on evaluating the integrity of the navigation maps using vehicle on board sensors.

Two types of map faults are considered: Structural Faults, addressing connectivity (e.g., intersections). Geometric Faults, addressing geographic location and road geometry (i.e. shape). Initially, a particular structural navigation map fault is addressed: the detection of roundabouts absent in the navigation map. This structural fault is problematic for ADAS and Autonomous Driving. The roundabouts are detected by classifying the shape of the vehicle trajectory. This is stored for use in ADAS and Autonomous Driving functions on future vehicle trips on the same area. Next, the geometry of the map is addressed. The main difficulties to do the autonomous integrity monitoring are the lack of reliable information and the low level of redundancy. This thesis introduces a mathematical framework based on the use of repeated vehicle trips to assess the integrity of map information. A sequential test is then developed to make it robust to noisy sensor data. The mathematical framework is demonstrated theoretically including the derivation of definitions and associated properties. Experiments using data acquired in real traffic conditions illustrate the performance of the proposed approaches.

Résumé

Les véhicules dits intelligents actuellement développés par la plupart des constructeurs automobiles, ainsi que les véhicules autonomes nécessitent des informations sur le contexte dans lequel ils évoluent. Certaines de ces informations (par exemple la courbure de la route, la forme des intersections, les limitations de vitesses) sont fournies en temps réel par le système de navigation qui exploite les données de cartes routières numériques. Des défauts résultant de l'évolution du réseau routier ou d'imprécisions lors de la collecte de données peuvent être contenus dans ces cartes numériques et entraîner le dysfonctionnement des systèmes d'aide à la conduite. Les recherches menées dans cette thèse visent à rendre le véhicule capable d'évaluer, de manière autonome et en temps réel, l'intégrité des informations fournies par son système de navigation.

Les véhicules de série sont désormais équipés d'un grand nombre de capteurs qui transmettent leurs mesures sur le réseau central interne du véhicule. Ces données sont donc facilement accessibles mais de faible précision. Le défi de cette thèse réside donc dans l'évaluation de l'intégrité des informations cartographiques malgré un faible degré de redondance et l'absence de données fiables.

On s'adresse à deux types de défauts cartographiques : les défauts structurels et les défauts géométriques. Les défauts structurels concernent les connections entre les routes (intersections). Un cas particulier de défaut structurel est traité : la détection de ronds-points qui n'apparaissent pas dans la carte numérique. Ce défaut est essentiel car il est fréquent (surtout en Europe) et perturbe le fonctionnement des aides à la conduite. Les ronds-points sont détectés à partir de la forme typique de la trajectoire du véhicule lorsqu'il les traverse, puis sont mémorisés pour avertir les aides à la conduite aux prochains passages du véhicule sur la zone.

Les imprécisions de représentation du tracé des routes dans la carte numérique sont quant à elles désignées comme défauts géométriques. Un formalisme mathématique est développé pour détecter ces défauts en comparant l'estimation de la position du véhicule d'après la carte à une autre estimation indépendante de la carte. Cette seconde estimation pouvant elle aussi être affectée par un défaut, les anciens trajets du véhicule sur la même zone sont utilisés. Un test statistique est finalement utilisé pour améliorer la méthode de détection de défauts géométriques dans des conditions de mesures bruitées. Toutes les méthodes développées dans le cadre de cette thèse sont évaluées à l'aide de données réelles.

Contents

Nomenclature	xi
List of Figures	xiii
1 Introduction	1
1.1 Intelligent Vehicles and Maps	2
1.1.1 Maps in Driving Assistance Systems	2
1.1.2 Vehicle Navigation Function	5
1.1.3 Map Technologies for Intelligent Vehicles	6
1.2 Map Faults	9
1.2.1 Definitions	9
1.2.2 Pathology	10
1.3 Problem Statement and Objectives	15
1.4 Thesis Approach	15
1.5 Contributions	17
1.6 Thesis Content	17
2 Navigation System Overview and Key Processes	19
2.1 Introduction	19
2.2 Vehicle Localization	20
2.2.1 Reference Coordinate Systems	20
2.2.2 GNSS Principles	24
2.2.3 Dead Reckoning	26
2.2.4 GNSS and DR Fusion	32
2.2.5 Road Centreline Positioning	32
2.3 Navigation System	35
2.3.1 Navigation Maps	35
2.3.2 Map-Matching	37
2.3.3 Electronic Horizon	38
2.4 Integrity Monitoring	39
2.5 Conclusion	42
3 Map Structural Faults Identification	45
3.1 Introduction	45
3.2 Case Study: Roundabouts	46

3.3	Description of the Proposed Method	48
3.3.1	Pre-Processing	48
3.3.2	Descriptor	49
3.3.3	Recognition	51
3.3.4	Classes	52
3.3.5	Roundabout Placement	57
3.4	Experimental Validation	59
3.4.1	Experimental Set-up	59
3.4.2	Method Performance Evaluation	59
3.5	Conclusion	61
4	Map Geometric Faults Detection, Isolation and Adaptation	65
4.1	Introduction	65
4.2	Problem Statement	66
4.2.1	Monitoring System	66
4.2.2	Spatial Sampling	68
4.2.3	Assumptions	69
4.3	Fault Detection, Isolation and Adaptation Method	73
4.3.1	Sets of Faults	73
4.3.2	Residual Processing	73
4.3.3	Relationships Between Faults and Residuals	75
4.3.4	Fault Detection and Isolation	76
4.3.5	Conditions of Isolability	77
4.4	Formalism Properties	79
4.5	Illustrative Examples	82
4.5.1	Navigation Map Geometric Fault	83
4.5.2	Isolation of GNSS Faults	84
4.6	Practical Implementation	85
4.6.1	Experimental Set-up	85
4.6.2	Spatial Sampling	86
4.6.3	Relaxing Equality Constraints	86
4.6.4	Gating Effects	87
4.6.5	In-Vehicle Algorithm	88
4.7	Experimental Evaluation	89
4.7.1	Metrics	89
4.7.2	Automatic Map Fault Generation	92
4.7.3	Urban Test Track	93
4.7.4	Rural Test Track	99
4.8	Conclusion	102
5	Extension of the FDIA Method to Handle Uncertainties	105
5.1	Introduction	105
5.2	Page's Trend Test	106
5.2.1	Signal Generation	106

5.2.2	Formulation of the Test	108
5.2.3	Experimental Evaluation of Fault Detection	112
5.3	FDIA with Page’s Trend Test	117
5.3.1	Method	117
5.3.2	Experimental Results in the Urban Scenario	120
5.4	Conclusion	123
6	Conclusions	125
6.1	Synthesis	125
6.2	Perspectives	127
	Bibliography	131
A	Experimental Set-up, Maps and Software Tools	141
A.1	Test Vehicle	141
A.2	Maps	142
A.2.1	OpenStreetMap	143
A.2.2	Map for PAMU Project	145
A.3	Navigation Systems	146
A.3.1	ADAS-RP	147
A.3.2	Navigation System based on OpenStreetMap	148
A.4	Software Tools for Navigation Maps Edition	151
A.4.1	Java OpenStreetMap	152
A.4.2	Quantum GIS	156
A.4.3	Map Faults Generator	160
B	Demonstration of the Rules of Non-Isolability	163
B.1	Introduction	163
B.2	Notations	164
B.3	Properties	165
B.4	New Problem Statement	166
B.4.1	If $l, m < K$	167
B.4.2	If $l = K$	167
B.4.3	If $m = K$	167
B.5	Conclusion	168
C	Automatic Truth Tables Generation for FDIA Implementation	171

Nomenclature

ACC	Adaptive Cruise Control
ADAS	Advanced Driving Assistance System
CC	Cruise Control
CUSUM	Cumulative Sum
DR	Deduced Reckoning
ECEF	Earth-Centred Earth-Fixed
ECI	Earth-Centred Inertial
ENU	East, North, Up
ESC	Electronic Stability Control
FDIA	Fault Detection, Isolation and Adaptation
GDOP	Geometric Dilution of Precision
GIS	Geographic Information System
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
LKA	Lane Keeping Assistant
NLOS	Non-Line-Of-Sight
OA	Overtaking Assistant
OSM	OpenStreetMap
PFL	Predictive Front Lighting
RAIM	Receiver Autonomous Integrity Monitoring

Contents

RCS	Reference Coordinate System
RP	Range Prediction
SLAM	Simultaneous Localization and Mapping
USERE	User Equivalent Range Error
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-X

List of Figures

1.1	Map accuracy needs	4
1.2	Functional diagram of the navigation functions	5
1.3	The concept of integrity.	10
1.4	Needs in localization and map accuracy for the development of driving assistance functions	11
1.5	Effect of road offset on intersection warning systems	14
1.6	Integrity monitoring in the Intelligent Vehicle framework	16
2.1	Vehicle localization and extraction of contextual information from the navigation map.	20
2.2	Ellipsoidal model of Earth	22
2.3	Heights of a point of space S	24
2.4	Kinematic model for front steered vehicle.	28
2.5	Example of scene detection by the vehicle front camera	34
2.6	Camera lane marking model	35
2.7	Electronic horizon	39
2.8	High confidence map-matching on a map fault	40
2.9	Residual generation for system monitoring	41
3.1	Conflict points for uncontrolled intersection without and with a roundabout	46
3.2	Pre-processing step	49
3.3	Buffer track with the profiles	51
3.4	Descriptor vector	51
3.5	Roundabout intersection with associated vehicle trajectories	53
3.6	Circumscribed circle of three points	57
3.7	Determination of the roundabout centre and radius in a buffer track	58

LIST OF FIGURES

3.8	Detail of vehicle's path	60
3.9	Results of the roundabout detection method	62
3.10	Example of application of the algorithm in rural environment	63
4.1	Structure of fault detection isolation and adaptation to standard passenger vehicle.	67
4.2	Roads spatially sampled with a 5 metres interval	68
4.3	Vehicle position estimates from sensors for three vehicle trips on the same road	69
4.4	Illustration of the notation convention	70
4.5	Systematic errors due to navigation map fault	71
4.6	Non-systematic faults of navigation function	71
4.7	A faulty map area	83
4.8	A correct map area close to a large building	84
4.9	Situation causing a theoretically impossible residual	88
4.10	Metrics employed for method evaluation	90
4.11	Five randomly generated navigation maps for method evaluation	92
4.12	GNSS tracks and navigation maps	94
4.13	False isolation rate, false validation rate and overall efficiency in urban test track	97
4.14	Information and spatial availabilities in the urban test track	98
4.15	Rural test track	100
4.16	Results of the FDIA for the rural test track	101
5.1	Page's trend test for fault detection in navigation integrity monitoring context.	106
5.2	The distance between the estimate from sensors G and the estimate from navigation N taken as random variables	107
5.3	Illustrative example of Page's test	111
5.4	Definition of the metrics used for tests assessment based on a simple example	113
5.5	Global view of the test areas	115
5.6	Sequential outcomes of three trend tests	116
5.7	Example of the use of Page's trend test with the FDIA framework	119

5.8	False isolation rate, false validation rate and overall efficiency in urban test track	121
5.9	Information and spatial availabilities in the urban test track	122
A.1	Test vehicle	142
A.2	Example of OSM map and corresponding OSM file.	144
A.3	Autonomous valet parking vehicle (PAMU project).	145
A.4	Navigation map for PAMU project.	146
A.5	ADAS-RP user interface.	147
A.6	Structure of the navigation system based on OpenStreetMap developed for testing purposes.	148
A.7	User interface of the navigation system	150
A.8	Electronic Horizon data structure	151
A.9	Java OpenStreetMap download interface	153
A.10	Result of the conversion of NMEA recording into OSM map	154
A.11	Simplification of the way resulting from NMEA recording	154
A.12	Creation of parallel ways	155
A.13	Quantum GIS user interface	157
A.14	Display of PAMU data in WGS84 reference coordinate system	158
A.15	Export of geographic information into Shapefile format.	159
A.16	Result of Shapefile import into JOSM.	160
A.17	PAMU map exported by QGIS in SVG and opened in Inkscape.	161
A.18	Result of automatic generation of five faulty maps	162

Chapter 1

Introduction

Contents

1.1	Intelligent Vehicles and Maps	2
1.2	Map Faults	9
1.3	Problem Statement and Objectives	15
1.4	Thesis Approach	15
1.5	Contributions	17
1.6	Thesis Content	17

From ancient times, maps are strategic issues as they represent a comprehensive and systematic compilation of geographic knowledge. For military operations, agriculture, environmental resource management, precise and complete knowledge of terrain is essential. Since paper maps cannot represent all the features of an area, a selection is made according to the information needed. For example, a hiking map has a small scale and shows mainly foot paths, contour lines and landmarks (e.g., rivers, power lines, localities). Typical vehicle navigation maps represent the road network geometry (i.e. roads and junctions) and attributes (e.g., speed limits, access restrictions, traffic signs, etc.).

Today, there is a rapid transformation of passenger vehicles, in which navigation maps are major components. Initially, maps were introduced as part of Navigation Systems and used in an informative-guidance manner. Now, they provide context information to informative Advanced Driving Assistance Systems (ADAS) and their usage is extended to actuating ADAS. Moreover, the future adoption of connected vehicles as part of what is known as V2X (Vehicle-to-Vehicle and Vehicle-to-Infrastructure) applications means that navigation maps play a fundamental role: they establish the spatial relationships between the road and the various communicating entities. There is currently much interest in the automotive industry on the deployment of Autonomous Vehicles because these represent major technological and societal changes [44]. Autonomous vehicles are sensor-based computer controlled systems capable of navigating by themselves. To attain such a capability, maps are central components. They are used as *a priori* information on top of

which the perceived environment is represented. The machine can therefore understand the current situation with respect to its environment and then decide which manoeuvre to perform.

With these developments, the frontier between mobile robotics and passenger vehicles starts to disappear. This evolution can be seen also through the usage of navigation maps. Mobile robotics have led the use of maps dedicated to the robot mission. Their coverage was limited but these were very accurate and complete. Intelligent vehicles nevertheless require worldwide maps, so navigation maps are used as a basis for developments but their level of quality is uncertain. The focus of this research is on the real-time and autonomous evaluation of faults in navigation maps used for intelligent vehicles.

This chapter presents the context of this thesis in terms of maps usage for different vehicle navigation applications. It also provides an overview of the principles of the map technologies used in intelligent vehicles. As this work focuses on map autonomous integrity monitoring, the pathology of map faults is also introduced. The problem is then formally stated and the main contributions of this thesis are finally given.

1.1 Intelligent Vehicles and Maps

1.1.1 Maps in Driving Assistance Systems

For the last decades, passenger vehicle manufacturers aim at improving safety, reducing pollution and enhancing driving comfort. Systems that provide advices to the driver or that take control instead of the driver are made possible by the available technology. These vehicles therefore gained perception, reasoning, decision and actuation capabilities and are qualified of Intelligent Vehicles.

There are many kinds of sensors used in intelligent vehicles. Some of them measure the vehicle state or ego-motion and are called proprioceptive sensors like for example speedometer, odometer, accelerometers, gyroscopes. Others perceive the immediate environment of the vehicle and are called exteroceptive sensors like for example radar, lidar, camera. A navigation map can be employed in different ways: as a source of raw information (e.g., speed limit or road curvature) or as a mean of interpretation of the perceived information. In combination with V2X communication it is also possible to benefit from information (e.g., traffic incidents, congestion, weather conditions) well beyond the constraints imposed by on board sensors.

The following paragraphs describe major ADAS either on current passenger vehicle or under development. This provides an overview of the evolution in the use of navigation maps.

Longitudinal control Statistics show that over-speed is a major cause of traffic fatalities [6]. The first ADAS therefore aimed at controlling the vehicle speed.

The Cruise Control (CC) system actuates on the throttle command to maintain a driver-defined constant speed. However, this system doesn't take into account other vehicles. This implies that the vehicle could collide with vehicles travelling at lower speeds. The next version, Adaptive Cruise Control (ACC), employs exteroceptive sensors to maintain a minimum distance to the vehicle ahead without any driver intervention. ACC doesn't take into consideration road geometry and infrastructure. The Contextual ACC intends to reduce the vehicle speed approaching a sharp bend or a toll booth [29]. The navigation map is central providing required contextual information.

Coupled with communication capabilities, navigation maps permit to extend the Contextual ACC to abnormal and temporary events. A central unit broadcasts road incidents to the vehicles. Traffic congestions, accidents, difficult weather conditions can then be anticipated by the vehicle by reducing the speed and telling the driver to take controls back [57].

Energy-management is made possible through the use of navigation maps. This is especially used for electric vehicle due to the low level of autonomy permitted by the batteries is a source of anxiety. The Range Prediction (RP) system determines whether a destination can be reached by the vehicle. It also informs the driver of the available charging facilities if the destination is beyond the predicted range [91]. According to the road slopes provided by the navigation map, eco-routing applications can determine the most efficient speed profile and extend the vehicle autonomy [21].

Lateral control Road departure is another major cause of traffic fatalities [6], thus active safety systems were developed to assist the driver to keep control and to keep in lane. After the successful introduction of stability control systems like the Electronic Stability Control (ESC) also known as Electronic Stability Program (ESP) that prevent vehicle slippage by acting on the brakes, active steering systems are now developed. These either warn the driver or correct the vehicle trajectory in case of lane departure based on lane detection by a front camera. These systems are named Lane Departure Warning (LDW), Lane Keeping Assistant (LKA) or Lane Centring Assistant (LCA) depending on control strategies. The lane markings are tracked on approximately twenty metres in front of the vehicle with a camera. However, in case of dense traffic, the camera only perceives lane marking a few metres ahead. Information on the oncoming road curvature and on the presence of junctions is used to improve tracking.

Driving comfort Driving comfort is a central topic for manufacturers because it represents the perceived quality of the vehicle by the driver. Improving comfort is therefore an objective for ADAS. Suspension setting is a matter of compromise between reactivity, passenger comfort and road conditions. Active suspension systems under development include the driving context (urban, inter-urban, rural) to refine the suspension stiffness and therefore propose the

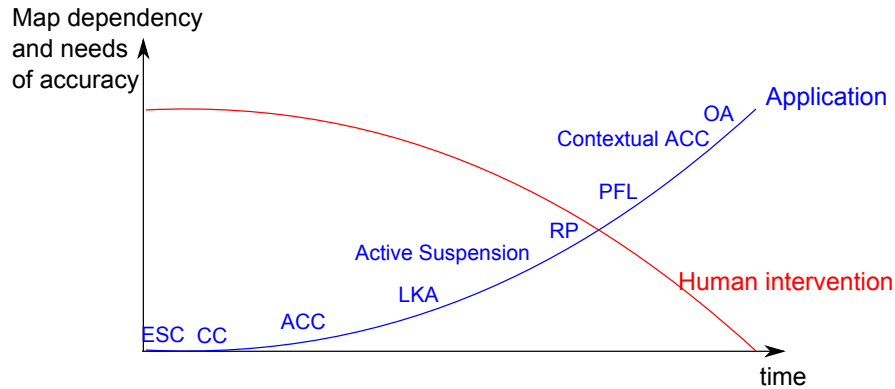


Figure 1.1: Evolution of the needs of map accuracy with time and development of automated driving functions.

best compromise between reactivity and comfort for various road conditions.

Driver perception is also addressed by ADAS such as front light beam control [29]. Adaptive lightening systems control the beam direction based on the steering wheel angle. However, in S-shaped bends the result is not acceptable. The navigation map is then used to anticipate the road curvature and to provide the optimal lightening with a system usually named Predictive Front Lighting (PFL).

It finally can be remarked that overtaking is a dangerous manoeuvre, especially on single carriageway roads. Current developments examine how to provide overtaking permission or restriction according to the road context. To do so, the Overtaking Assistant (OA) uses the line-of-sight based on road curvature and terrain models, junctions ahead and the type of lane markings. The navigation map is employed to get this information [30].

It can be observed from this overview that vehicle automation goes along with a significant use of navigation maps. Being an *a priori* knowledge of the environment, it supersedes the human for inferring the current and future vehicle context. However, like any other source of information, navigation maps are imperfect and must be considered with caution. Efforts made in the recent years in the domain of intelligent vehicles often employ maps as a perfect and comprehensive source of information. This originates from robotics oriented maps that were manually made with high precision. This assumption is no longer valid when using global maps. Their imperfections are not significant when the map is interpreted by a human but they can have serious consequences as the degree of automation of the vehicle increases. Figure 1.1 schematically shows the evolution of the map needs when vehicle automation degree increases. The purpose of this thesis is to make the vehicle able to evaluate autonomously the quality of the data provided by its navigation map in order to prevent malfunctions of ADAS and automated driving functions.

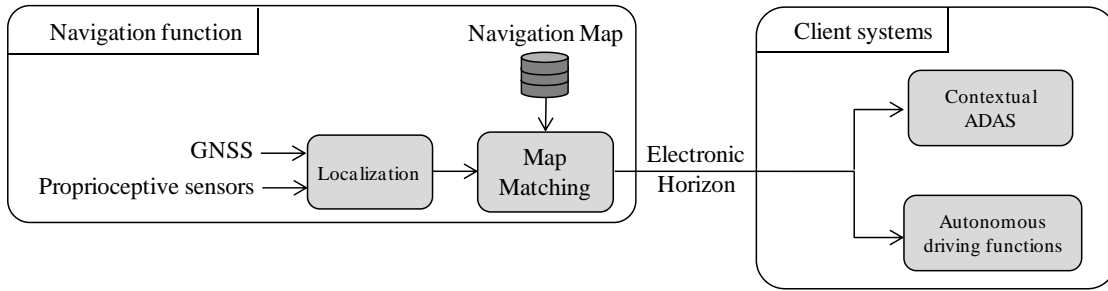


Figure 1.2: Functional diagram of the navigation functions for intelligent vehicles addressed in this thesis.

1.1.2 Vehicle Navigation Function

According to the focus of this work, it is important to understand how navigation maps are used in intelligent or autonomous vehicles. A navigation on board the vehicle stores all the *a priori* knowledge concerning the road network.

For this information to be relevant, it must be selected according to the area of interest that corresponds to the current vehicle position. It is therefore necessary to estimate firstly the vehicle position and then to project this estimation onto the road network. The current vehicle position is then associated with the stored information. The resulting information is provided either to map dependent ADAS applications or to autonomous driving functions [40]. The principles are shown in Figure 1.2. This function can be summarised in three parts which are namely the Navigation Map, the Localization system and the Map-Matching process. These are detailed as follows.

Navigation Map The navigation map is a representation of the road network with its relevant features according to the applications. For intelligent vehicles, it stores the road geometry, topology (junctions) and related attributes. The on board navigation map is a compiled version of the geographic database owned by the map provider. This compilation aims at reducing the memory required to store the map by simplifying road geometry and ignoring irrelevant features. Moreover, compilation improves data access for map-matching search algorithms.

Localization Even the most precise map is useless without knowing the position onto this map. Landmark localization relies on the use of perceivable features in the environment, like traffic signs, bridges, road junctions, etc. The vehicle could be positioned relatively to its surrounding environment. Instead, GNSS (Global Navigation Satellite System) receivers are widely used in intelligent vehicle domain mainly because of their affordable price. The principles of GNSS are substantially equivalent to landmark localization as they rely on measuring pseudo-ranges between the receiver and known satellites. The prin-

cial drawbacks of GNSS is the relatively low accuracy and the need to be in open-sky environments. Proprioceptive sensors are typically used for vehicle positioning when satellite signal loss occurs. Wireless communications can also be used to receive differential corrections that improve position quality.

Map-Matching The map-matching consists in finding the road, within the map, on which the vehicle is travelling. The vehicle calculated position is projected onto the navigation map. However, due to the potentially low position accuracy, this position may not fit any road of the navigation map. Assuming that the vehicle is actually on a road stored in the navigation map, the map-matching finds the road that best fits the position. Map-matching can be difficult especially in urban areas because the road network is dense and GNSS conditions are difficult. In this situation, the GNSS error is large and there are many road candidates. Thus efforts have been made in the literature to improve map-matching performance using several criteria (e.g., distance between the road and the vehicle measured position, deviation between vehicle heading and road direction, coherency between vehicle speed and road type) [98]. A confidence index is usually provided with the map-matched position which corresponds to the final solution score.

Once the vehicle position is projected on a road of the navigation map, relevant information about the vehicle current road environment is sent to the client systems. As this information represents a set of context events the vehicle will meet as it travels, this is known as Electronic Horizon (EH) [103].

This thesis takes place at a vehicle manufacturer where these three functions are developed and provided by a supplier as a sole function. This is the reason why they are grouped and named Navigation Function. It is to note that this is usually a black box from the vehicle manufacturer perspective. This means that there is no access permitted to any of internal variables or data of this function. Thus, when a failure occurs on the application functions it is difficult to identify their origin, to correct them or reduce their effect. This is a fundamental consideration in this thesis as it constraints on the use of existing approaches to monitor the integrity of the map data available in the EH.

1.1.3 Map Technologies for Intelligent Vehicles

There are two ways for representing geographical features in navigation maps: raster and vector. Raster consists in representing geographical data by an array in which each cell contains information. This format is suitable for continuous data representation like elevation, temperature, rainfall, population density, or satellite imagery. On the other hand, in vectorial data (also named topological maps), information is represented as a set of connected nodes and attributes. This format is convenient for representing semantic objects and relationships like for example road network, buildings and areas with attributes (e.g., type, name, speed limits, number

of lanes). A navigation map may store a set of raster and vector data classified into layers. This data can be managed using a Geographic Information System (GIS).

The following sections give an overview of the map contents and formats according to the common usages in the intelligent vehicles domain.

1.1.3.1 Guidance

Guidance maps were introduced for navigation assistance. These are dedicated to the computation of the optimal path from a starting point to a desired destination. In these vectorial maps, the road network is represented as a graph where the nodes are road junctions and costs are associated to the edges. These maps in which the representation of the connectivity between nodes (places) and the cost associated to the transition between two nodes are referred as topological maps. The absolute localization of the nodes is of secondary importance. Several parameters can be used as cost function to compute the best route (e.g., distance, travel time, toll). Methods such as Dijkstra's algorithm [36], Genetic Algorithm [82] or Rapidly-exploring Random Trees [118, 85] provide solutions for solving graphs and find the optimal route.

The numerical formats of such maps are designed to facilitate the use of graph search algorithms. The ESRI Shapefile is usable with most GIS software [45]. This data stores geographical coordinates as well as its geographical index. This allows faster access to data by search algorithms. Other proprietary binary formats are developed by the map providers but data can only be accessed using their own software. Data base oriented formats such as Spatialite are also employed for guidance purpose since they provide a fast and easy access to the data [4].

Guidance applications require large coverage maps, the size of the navigation map is therefore substantial. When changes occur on the road network, the map update must be done using physical memory storage like DVD or flash card. The map providers are now developing tile based formats in order to make possible the update of small areas. The updates can therefore be sent via GSM wireless connection for instance.

1.1.3.2 Robotics

A mobile robotic platform must know where it is and how to reach its destination. For this guidance purpose, vector maps similar to those described in the previous paragraph are employed. The robot deduces a set of way points from the mission planning system. The robot immediate environment must be modelled in terms of free spaces and obstacles in order to reach the next way point safely. Raster maps are often employed for this purpose. Each cell of the map contains the knowledge of the robot about the presence of an obstacle. This value can be binary (i.e. 0 for free cell and 1 for occupied cell) or continuous: the probability [113] or the credibility [79]

for the cell to be occupied. The robot must find a path to the next way point that crosses only free cells.

The Simultaneous Localization And Mapping (SLAM) algorithm used in robotics led the development of dedicated representation of the environment. In this process, the mobile robot has no *a priori* information on the environment. The purpose of SLAM is for the robot to build create the map based on the perception of landmarks and to be able to use this map to navigate. The challenge in this task is to simultaneously reduce the uncertainty on the robot position and on the landmark position. The SLAM maps therefore consist in listing the landmarks positions and uncertainty. These have different forms depending on the method used for solving SLAM (e.g., Kalman Filter, Particle Filter) [41, 13].

Autonomous vehicle led the development of map standards that originate from research challenges, from laboratories or from industry research and development [26, 106]. The file formats that are specified vary (XML text formats, database oriented navigation maps or binary files) but the principles shared by these formats is to represent a set of features that make the vehicle able to position, plan the mission and avoid obstacles.

1.1.3.3 Infotainment and location based services

The main public use of digital map is made through web services. The large quantity of geo-localized data centralised on web servers allow anyone in a few seconds to locate a restaurant, a train station, a friend, etc.

Digital maps are stored on servers and accessed through dedicated web sites. Projects such as OpenStreetMap [2, 1, 5], provide access to the the whole geographic information under open source licence. In addition to usage detailed before in this paragraph, an easy and complete access to data makes other applications possible that range from humanitarian aid to urban management [109].

1.1.3.4 Crowd Sourcing

The trend is for crowd sourcing, that is how to acquire information in an ubiquitous manner from other sources. For example an accident has occurred, this is geo-localized and registered in a GIS in a cloud server. The server then stores vehicle speeds queried from vehicles approaching the event point and infer traffic density and compare with statistics stored elsewhere. It then queries on weather information, determine alternative roads for traffic flow relief, etc. This crowd sourced and harvested information is then used to contextualise the event and provide relevant information to approaching vehicles [62].

1.2 Map Faults

Intelligent and autonomous vehicles applications require the information provided by the navigation function in the form of the EH as detailed in the previous section and illustrated in Figure 1.2. Erroneous data in the EH may result in undesirable behaviour of client systems and generate hazardous situations. These can be the consequence of faults that arise at any step of the EH generation. Localization system used in combination with the map may be perturbed and provide a position estimate that contains large errors. This may result in large errors on the estimation of the vehicle position onto the map. A map is a complex entity that represents the environment that is constantly evolving, it then necessarily contains Faults. The scope of this thesis centres on faults originating in the navigation map.

This section first defines the concepts of fault and error which are essential in this work. Next the pathology of faults found in navigation maps and their effects on the vehicle applications is described.

1.2.1 Definitions

The terms of fault, error, failure and integrity may have different meanings according to the application domain. The following definitions are used in the context of this research and are based on those given in [96]:

- Fault: Error generative process. The presence of a fault may not lead to an error.
- Error: A discrepancy between a computed, observed or measured value and the true, specified or theoretically correct value.
- Failure: Instance in time when a required function exceeds the acceptable limits or is terminated.
- Integrity: Reliability of the confidence indicator associated with an information with respect to specifications of the client application.

There is a loss of integrity of EH data if the error is greater than the estimation.

Figure 1.3a shows the geometry of the segment of a road network. The shaded shape represents admissible error associated to it. In this illustration, a fault occurs if the EH data is outside this envelop. Figure 1.3b shows the loss of integrity for the position estimate. The true vehicle position is indeed outside the confidence domain of the error associated to the position estimate.

The accuracy of the vehicle position estimate provided by the navigation function in the EH is closely related to both the localization and the navigation map. Figure 1.4 highlights this dependency and the consequences on the development of driving assistance functions, from the assisted navigation to autonomous driving [51].

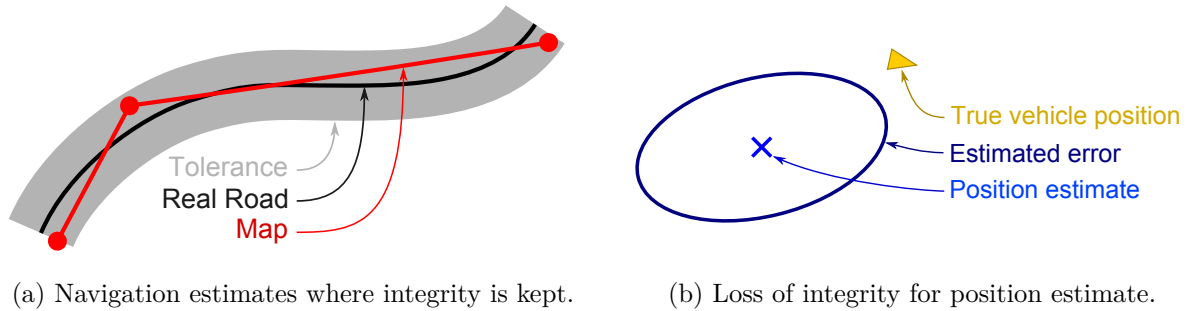


Figure 1.3: The concept of integrity.

Passenger vehicles are mainly used either for commuting or to travel on known roads. Except for occasional journeys like going on vacations, the vehicle is therefore driven on roads on which it has already been driven. It must be noticed that the dysfunctions due to map faults are mainly systematic. Whenever the vehicle crosses again the erroneous road, the EH provides the same false data to ADAS which, themselves, will have the same uncomfortable behaviour. The quality perceived by the driver decreases significantly due to the frustration of repetitive illogical warnings or reactions of the vehicle.

1.2.2 Pathology

The road network is constantly evolving. Map providers consider that 15% of the road network of a mature country change each year. This value increases in developing countries. With the exception of few countries like Japan, road changes are not monitored by a central administration. It is therefore difficult for cartographers to keep the geographic databases up-to-date.

Map creation is a complex process that takes a long time. When loaded in the vehicle, the navigation map data is therefore already several months old and partially outdated. Mapmakers adopt two approaches to pursue fast map updates. The first is to facilitate mapping surveys by deploying a large fleet of vehicles. These are equipped with a specific set of sensors (e.g., lidar, inertial measurement systems, differential GNSS) for rapid mapping of the road network. Priorities are given to main roads and areas where changes have been reported. This strategy allows precise mapping but is costly in material and human resources. The second is to rely on user data. The records of the journeys travelled with the navigation devices are automatically uploaded to the map provider server. Data mining algorithms are then applied to update the central database. This approach is less expensive and allows to update the road geometry or the mean travel times. However other structural elements such as traffic signs, speed limits still require field surveys.

The typical faults are summarized in a pathology. It provides an overview of the

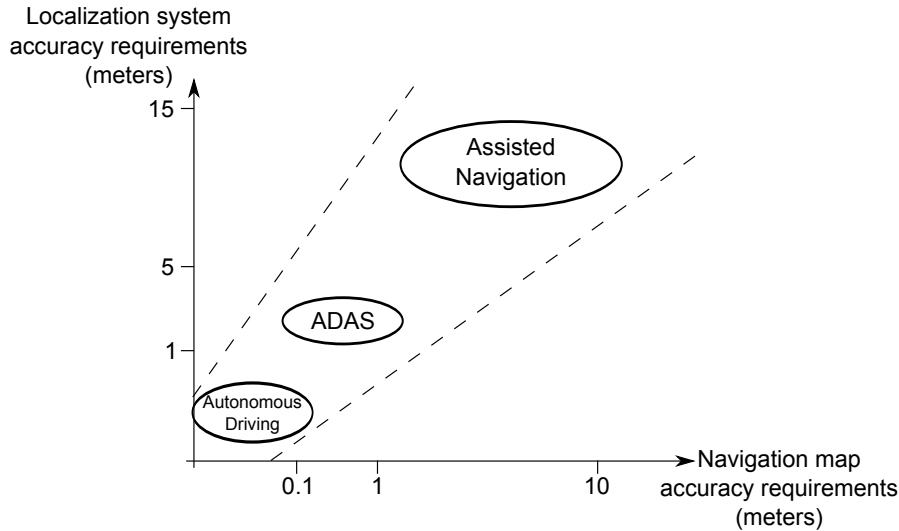


Figure 1.4: Needs in localization and map accuracy for the development of driving assistance functions (after [51])







issues involved by each fault and associated scientific challenges. The distinction is made here between fault in the navigation map structure, geometry and attributes. Structural faults are related to the manner in which the elements of the map are connected or are identified in the map. Geometric faults are related to the shape or the geographic placement of these entities. This distinction is made because they require different approaches to be detected and corrected. These are detailed in the following paragraphs and summarised in Table 1.1.

1.2.2.1 Structural Faults

Road connectivity The correct representation of road junction in the navigation map is essential for optimal path planning. The missing connection between two road may cause the path planner choose a suboptimal path and bother the driver. Reciprocally, a connection between two roads in the navigation map that does not exist in reality may result in impossible path. In this case, the driver may be misled by the navigation assistant and cause hazardous driving situations.

Type of intersection The significant information at road junctions is the right way of one road with respect to the others. Hazardous situations could occur if this information is missing. In navigation maps, the type of intersection can be associated to junctions in order to describe implicitly the order of priorities. Over the past years, a particular type of intersection that is roundabouts (or traffic circles) is preferred to others types and is built in many places. They indeed reduce fatalities

Table 1.1: Navigation map pathology

Map Fault		Example	Affected ADAS and Consequences
Structure	Road Connectivity		Navigation assistant: - Suboptimal path - Driver misled
	Type of Intersection		Contextual ACC - Inappropriate speed Navigation assistant: - Incoherent instructions
Geometry	Road shape and Road Offset		Contextual ACC: - Inappropriate speed - Target loss PFL: - Poor visibility
	Missing Road		Every Contextual ADAS
Attributes	Speed Limits		Navigation assistant: - Suboptimal path - Driver misled
	Vehicle Restriction		Navigation assistant: - Suboptimal path - Driver misled

and increase traffic exchange between roads. Their construction means that the network structure is changed [104, 117]. The manner in which the vehicle crosses them is different to a classic crossroad and the reachability to the intersection branches is different. The inclusion of a roundabout must be registered in the navigation map in order to provide convenient guidance information to the driver and adapt the approaching manoeuvre (e.g., speed reduction, lane placement).

1.2.2.2 Geometric fault

Road shape and road offset For optimisation purposes, the on board navigation maps are compressed and compiled. In this process, some road shape points are removed. The road curvature estimated using the shape points is therefore perverted. In the GIS domain, absolute and relative accuracies are distinguished [24]. The former describes the accuracy of the geographic feature with respect to a global reference coordinate system whilst the later describes accuracy relative to other features. Roads with low relative accuracy result in poor shape definition. The curve warning related ADAS (i.e. Contextual ACC, LKA, PFL) are directly affected by low relative accuracy of the road. In case of low absolute and high relative accuracies (road offset), the map-matching algorithm may not choose the right road candidate, especially in dense road network area. Moreover, Figure 1.5 shows that the offset on junctions induces malfunctions of intersection warning systems for vehicles running on the other roads [9].

Missing roads Road networks change over time and some roads are created or closed. Until the next cartographic survey, every contextual ADAS application cannot operate properly. The case in which the vehicle is driven on a new road is split in two situations. First, if the new road is far from the roads stored in the navigation map, the map-matching function will switch to failure mode and provide the output *off road*. The contextual ADAS applications could then adopt a suitable strategy and limit the consequences. Second, if the new road goes along a road stored in the navigation map, the vehicle position is likely to be matched on the wrong road with a high confidence level. The contextual ADAS could operate based on inconsistent data and cause hazardous situations.

1.2.2.3 Attributes faults

Speed limits The speed limit can be displayed to the driver in Over-Speed Prevention (OSP) system or used to set the vehicle cruise speed in Contextual ACC applications. The navigation map stores the speed limit as an attribute for each road. This attribute originates from direct surveys or is inferred based on the road class, the number of lanes and the area (inside or outside built-up area). Smart cameras are nowadays capable of detecting speed signs in real time [68, 81], however,

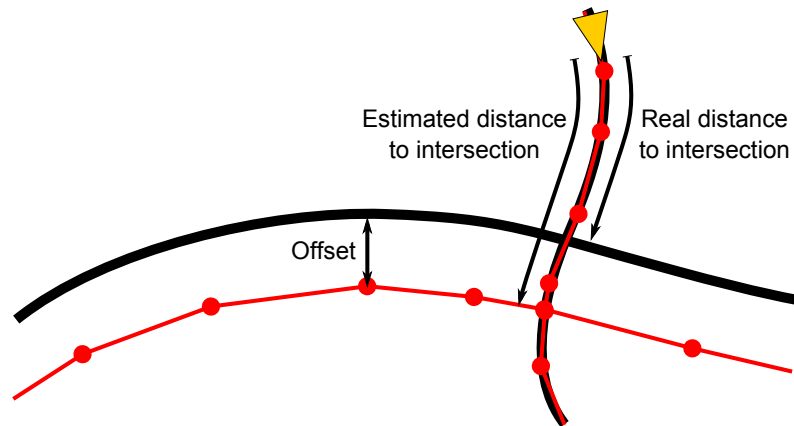


Figure 1.5: Consequences of road offset on intersection warning systems. The true roads are in black. The red polylines are the roads stored in the navigation map. The vehicle is the yellow triangle.

the challenge resides in determining whether the detected sign is applicable to the vehicle. Speed limits indicated by traffic signs may be dedicated to one particular lane (e.g., for a motorway exit), to a class of vehicles (e.g., trucks, vehicles towing caravans, buses) or to special whether conditions. An inappropriate estimation of the speed limit by the vehicle would bother the driver and decrease the perceived vehicle quality.

Driving directions, vehicle restrictions In navigation maps, roads are assumed to be drivable in both direction unless a dedicated attribute is associated to the road. Similarly, attributes are defined in order to establish some traffic restrictions (e.g., trucks, pedestrian, maximum height). The path planner uses these attributes to exclude wrong-way roads and roads that does not comply with the vehicle type. Faults in these attribute may cause the path planner to choose a suboptimal route or ask the driver to take a forbidden route. This may have severe consequences especially for large goods vehicles that cannot manoeuvre easily.

As a conclusion on this fault pathology, it must be noticed that the variety of the information stored in the navigation map induces a variety in the faults in the map. The methods employed to address the detection and correction of these faults are necessarily diverse in terms of sensors and formalisms. Some of the formalisms that could permit to address this problem require an *a priori* knowledge on the correctness of the navigation map. However, to the author's knowledge, no trustworthy study on the navigation map overall reliability has been done. In this thesis the navigation map is considered to be globally correct but potentially locally erroneous.

1.3 Problem Statement and Objectives

The literature review and vehicles manufacturers roadmaps show an increased dependence of *a priori* geographic knowledge stored in navigation maps for ADAS functions and autonomous driving applications. It was found that navigation maps include several issues due to data acquisition and the evolution of the road network. Further, it is impossible to expect a perfect and up-to-date map on board of the vehicle with the current set-up. Depending on the type of application, a fault in the navigation map induces hazardous situations or makes the driver uncomfortable, that is, reduces safety and causes client dissatisfaction. This thesis addresses the problem of evaluating in real-time and autonomously the integrity of geometry and structure of the road network data stored in navigation maps. This is done by relying on proprioceptive sensors and GNSS receivers and by taking benefit of the repetition of vehicle trips.

For this purpose the following objectives have been defined:

- To provide the underlying technological and theoretical background necessary to address the detection and isolation of fault in intelligent vehicle navigation maps.
- To infer the presence of new roundabout as a new structural element of the road network.
- To be able to solve the ambiguity associated to likely faults in the vehicle position estimates despite the low sensor redundancy.
- To make the fault detection isolation and adaptation more robust to inaccuracies that affect the on board vehicle sensors.
- To demonstrate the validity of the theoretical framework presented in a experimental manner using data acquired in real situations.

1.4 Thesis Approach

The solution to the autonomous integrity monitoring problem needs to be embedded into existing systems. For this purpose, the work in this thesis defines an Integrity Monitoring Framework applied to intelligent vehicles. A block diagram representation of it is shown in Figure 1.6. The existing navigation function is considered as a black box. It provides information to the various applications (Client Systems) embedded on what is known as the Electronic Horizon. Since this is a closed system that has been optimised for cost and performance, access to internal data is not available outside the system. Additional inputs available are all the proprioceptive vehicle data available from the vehicle standard on board sensors (e.g., odometry, steering wheel angle, vehicle speed, yaw rate). A standard GNSS receiver independent of the navigation system is also assumed to be available. Other inputs

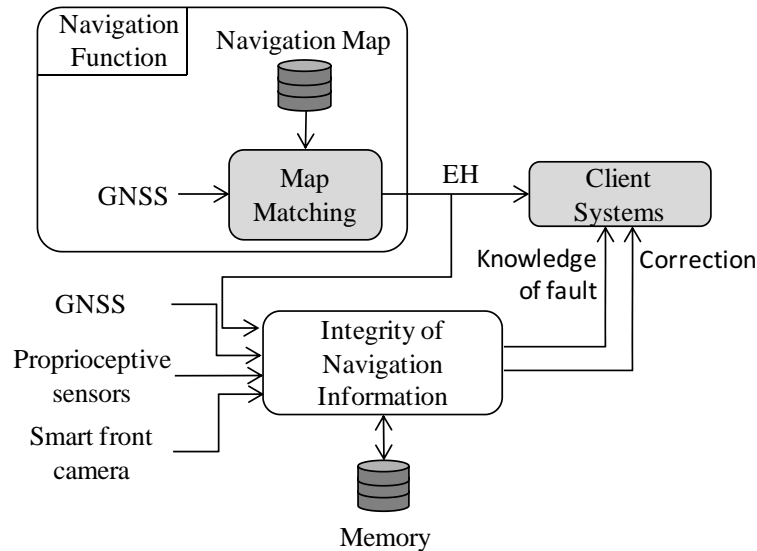


Figure 1.6: Integrity monitoring in the Intelligent Vehicle framework

are possibly available from exteroceptive on board sensors like smart front video cameras or radars. Some passenger vehicles are indeed being equipped with such sensors depending on the category of vehicle. The methods developed in this thesis do not rely on such sensors but these can benefit from improvement if exteroceptive sensors are available. As the wireless communications are still not used in passenger vehicles, these are not included in this thesis.

In the proposed framework, and as shown in Figure 1.6, the information provided by the navigation function is compared with the information estimated using the available inputs in what it is labelled as *Integrity of Navigation Information*. A very important consideration is that it is assumed that the estimation made is likely to be erroneous due to inaccuracies and faults that could be associated to the low-level performance of the sensors used. An ambiguity on the identification of the fault exists when discrepancy between the sensor estimation and the navigation estimation is detected. The repetition of trips made by the vehicle is used to solve this ambiguity. While the vehicle is driven, the data provided by the navigation system and the diagnoses made by the proposed methods are stored within the method memory. This provides the redundancy required to identify the faults and allows to correct navigation information when the vehicle crosses the road again.

Finally and as shown by the outputs *knowledge of fault* in Figure 1.6, the proposed framework is intended to provide to client systems a real-time evaluation of the data in the EH. As soon as the proposed method detected erroneous information in the EH, the correction to apply is sent to the client systems by the output *correction*. It is therefore assumed that the client systems are designed to take into account this evaluation and adopt dedicated failure mode in case of loss of integrity.

1.5 Contributions

The contributions of this thesis can be summarised as follows:

- The formulation of an algorithmic framework that allows for the integration of a map integrity monitoring system independent of standard vehicle on board navigation function solutions, based on the sole use data available on conventional passenger vehicles. It takes into account the low level of redundancy in the available sensor set.
- A detection method for the presence of roundabouts in replacement of crossroads based on the recognition of the patterns that the vehicle trajectory takes as it crosses them and the adaptation of a classification method. After a single crossing, it is possible to establish that a new roundabout exists that is unregistered in the navigation map.
- A mathematical framework that identifies and excludes faults in the geometry of the road network stored in navigation maps. It takes into account the uncertainty associated to the vehicle position estimates due to limits on the sensors used.
- The mathematical proof that enables the detection of a geometric or position estimate fault after a limited number of journeys around the same navigation area.
- A validation method that allows for the random generation of geometric network faults with the use of proprioceptive and exteroceptive data to determine the robustness of the convergence of the fault detection algorithms.
- The application of a sequential test to the detection of the divergence between two estimates of the vehicle regardless of the noise disturbing the measurements.
- A demonstration of the performance of the proposed algorithms using data acquired in real traffic conditions.

1.6 Thesis Content

The thesis is organised as follows.

Chapter 2 presents the principles of the functions involved in the computation of the contextual information provided to vehicle ADAS or autonomous driving functions, namely vehicle localization, navigation maps and map-matching. This allows to understand the reasons of possible loss of integrity prior to address this problem. Work on the detection of map changes and system integrity monitoring is also detailed.

One type of fault in the map structure is addressed in Chapter 3: new roundabouts. This chapter shows that this particular intersection is significant for contextual

ADAS and autonomous driving functions since it affect driving manoeuvres and must be anticipated. In addition, roundabouts significantly reduce traffic fatalities so they are built in many place and are likely to be missing in the navigation map. In order to detect this semantic element, a method based on the probabilist recognition of the roundabout as a pattern in the vehicle trajectory is employed. The method is then validated on real vehicle data.

Chapter 4 addresses the detection of geometric faults in the navigation map. The integrity of the estimation of the vehicle position made by the navigation system is investigated based on the comparison with another vehicle position estimate independent of the map. Since this last estimate is also likely to be faulty, an ambiguity remains in case of large discrepancy between both estimates. This chapter presents a formalism that solves this ambiguity using the multiple vehicle trips on the same locations. The principles and the interest of this formalism are first mathematically demonstrated. The application to navigation map Fault Detection Isolation and Adaptation is then detailed. The evaluation of the formalism performance finally shows that the noise affecting real data degrades the results.

The purpose of Chapter 5 is therefore to make the FDIA formalism more robust to sensor noise. A statistical test is employed to detect discrepancy between position estimates that is similar to the noise on the data. The performance of the FDIA formalism associated are established based on the same data and metrics as in the previous chapter which permit objective comparison.

Chapter 6 finally concludes this thesis by providing perspectives for further work.

Chapter 2

Navigation System Overview and Key Processes

Contents

2.1	Introduction	19
2.2	Vehicle Localization	20
2.3	Navigation System	35
2.4	Integrity Monitoring	39
2.5	Conclusion	42

2.1 Introduction

The output of the Navigation Function is the result of several processes running in separate components, each with different performance. For the purpose of this research, the navigation is considered as a system and it is shown in Figure 2.1. In order to monitor the integrity of navigation system (in particular in order to detect and isolate faults in the map), it is necessary to provide an understanding of the physics of this system and to introduce the theoretical concepts involved.

The vehicle localization principles are presented first, followed by the description of some methods used to determine the vehicle position onto the navigation map known as map-matching. Integrity evaluation of each of these steps is also presented. System monitoring approaches are finally introduced.

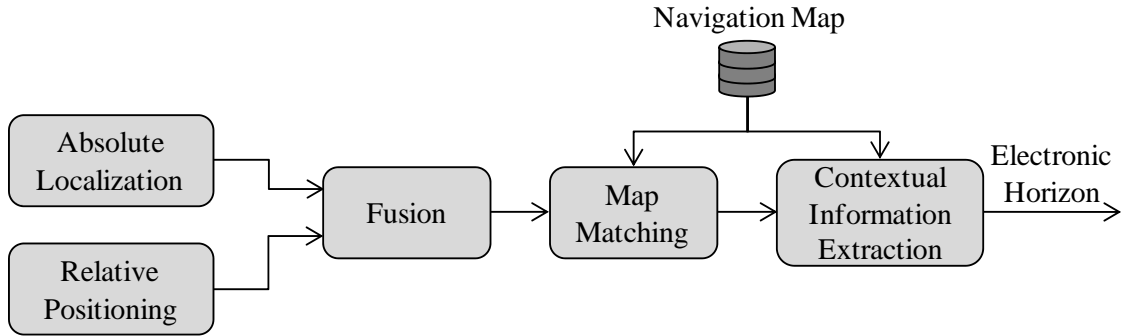


Figure 2.1: Vehicle localization and extraction of contextual information from the navigation map.

2.2 Vehicle Localization

Vehicle localization¹ consists in determining the vehicle coordinates with respect to a reference coordinate system. This section therefore introduces the usual reference coordinate systems used for vehicle localization and the methods involved in the estimation of the vehicle position. This allows to understand the potential error introduced by coordinate projection and fault that could occur in the localization process.

2.2.1 Reference Coordinate Systems

The computation of the vehicle position and the projection of this position onto a navigation map require the use of clearly defined Reference Coordinate Systems (RCS). Coordinate projections are used for representing large scale 3-dimensional geographic information in 2-dimensional maps. The principal RCS and projections are introduced in the following sections.

Earth-Centred Inertial Coordinate System

The Earth-Centred Inertial (ECI) coordinate system origin coincides with the Earth's centre of mass. Its xy -plane lies on the equatorial plane and the z -axis goes through the geographical North pole. In this RCS, the x -axis points towards a constant direction of the celestial sphere. The y -axis is chosen so that the ECI is a right-handed coordinate system. The Earth is supposed to have a constant rotation motion with respect to this system. The ECI coordinate system is employed to describe the motion of the satellites using Newton's laws but it is inconvenient for application related to the Earth surface [67, 73].

¹Sometimes entitled "ego-localization" or "positioning". In this manuscript, the term "localization" is used.

Earth-Centred Earth-Fixed Coordinate System

Like the ECI, the origin of the Earth-Centred Earth-Fixed (ECEF) coordinate system is the Earth's centre of mass, the xy -plane coincides with the equatorial plane and the North pole lies on its z -axis. The x -axis is defined to lie on the zero-longitude plane. The Earth is therefore stationary in the ECEF reference frame. This RCS is convenient for GNSS positioning that relies on measuring the time of flight of a signal between a satellite constellation and the receiver. The satellites position is estimated with a high precision in ECI, converted into ECEF by the GNSS ground segment and is transmitted to the receiver. This can then measure the distance to the satellites and solve its position. Cartesian coordinates in Earth centred reference systems are not convenient for human understanding since the order of magnitude of the distance to this reference frame is $6 \cdot 10^6$ m [67, 73].

Geodetic Coordinates

The geodetic coordinates (i.e. latitude, longitude and height) are used instead of ECEF since they provide a rapid understanding of the Earth area where the receiver is located. They are expressed in decimal or sexagesimal degrees. In the sexagesimal notation, usually denoted by DMS for Degree Minute Second, one degree is divided into 60 arc-minutes and one arc-minute into 60 arc-seconds.

A physical model of the Earth is necessary to transform Cartesian ECEF receiver coordinates into latitude (ϕ), longitude (λ) and height (h) as shown in Figure 2.2. The most used representation is the World Geodetic System 1984 (WGS84) in which the Earth is described by an ellipsoid [86]. In this model, any cross section of the Earth that contain the z -axis is an ellipse, its semi-major axis a is the mean equatorial radius, given as $6.3781370000 \cdot 10^6$ m. The ellipse semi-minor axis b is the Earth polar radius, given as $6.3567523142 \cdot 10^6$ m. Any cross-section of the WGS84 Earth model parallel to the equatorial plane is circular.

Given this model and the ECEF Cartesian coordinates of the receiver $R = (x_r, y_r, z_r)$, the longitude λ is the angle made by the projection of the user in the xy (i.e. equatorial) plane and the x -axis. By convention, the longitude ranges from -180° to 180° and positive longitudes correspond to degrees East. It can be calculated from Cartesian coordinates with the following equation:

$$\lambda = \begin{cases} \arctan\left(\frac{y_r}{x_r}\right) & x_r \geq 0 \\ 180^\circ + \arctan\left(\frac{y_r}{x_r}\right) & x_r < 0, y_r \geq 0 \\ \arctan\left(\frac{y_r}{x_r}\right) - 180^\circ & x_r < 0, y_r < 0 \end{cases} \quad (2.1)$$

Due to the ellipsoidal model, a difference occurs for the definition of the latitude

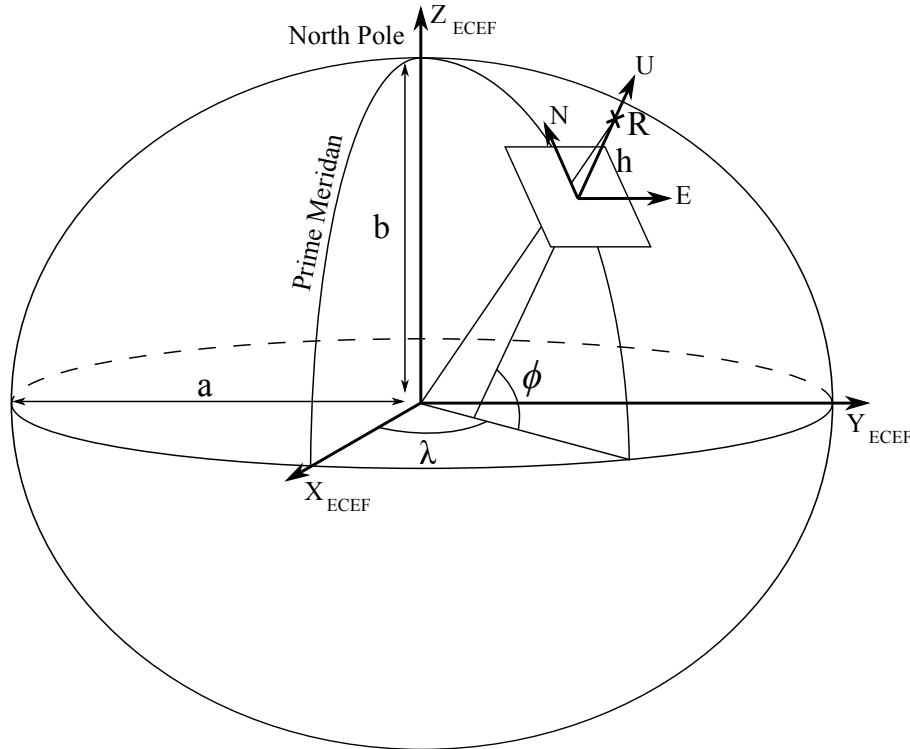


Figure 2.2: Ellipsoidal model of Earth (after [67]).

ϕ and height h with respect to a classical spherical coordinate system. These are defined according to the ellipsoid normal; the Earth centre is on this normal only when the receiver is on the pole or Equator. As shown in Figure 2.2, the latitude ϕ is the angle made by the ellipsoid normal at the receiver position U with the Equator plane and ranges from -90° to 90° with positive values on the Northern hemisphere. The height h is the shortest distance between the ellipsoid and the user. It is positive if the user is outside the ellipsoid.

Algorithm 2.1 details the computation of the geodetic latitude and height from ECEF coordinates. The choice of the convergence threshold ϵ is a compromise between the time of convergence and the precision of the calculated coordinates. Usually, $\epsilon = 10^{-9}^\circ$ is chosen because it represents an error of approximately 1 mm on the Earth surface which is convenient for navigation purposes. This algorithm is involved in several parts of the method developed in this thesis. It can be noticed that closed forms for this transformation also exist and can be found in [48].

The height h defined before represents the distance between a point and the ellipsoid taken as a simplified model for the Earth surface. Historically, the measure of height used in cartography is the distance with the sea level modelled by the geoid. The geoid is the surface of constant gravitational potential that corresponds to the mean sea level. This is a complex shape since the gravitational potential depends on the density of the subsurface Earth layers. The distance between the ellipsoid and the

Algorithm 2.1 Computation of the geodetic latitude and height from ECEF coordinates

Inputs: ECEF coordinates (x_r, y_r, z_r) , the convergence threshold ϵ

Outputs: Geodetic latitude and height (ϕ, h)

Begin

$$e \leftarrow \sqrt{\frac{a^2 - b^2}{a^2}}$$

$$p \leftarrow \sqrt{x_r^2 + y_r^2}$$

$$\phi \leftarrow \infty$$

$$\phi' \leftarrow \arctan\left(\frac{z_r}{p}\right)$$

While $\|\phi - \phi'\| > \epsilon$, **do**

$$w \leftarrow \sqrt{1 - e^2 \sin^2(\phi')}$$

$$N \leftarrow \frac{a}{w}$$

$$h \leftarrow p \cdot \cos(\phi') + z_r \cdot \sin(\phi') - a \cdot w$$

$$\phi \leftarrow \arctan\left(\frac{z_r}{p} \cdot \frac{N+h}{N(1-e^2)+h}\right)$$

End while

End

geoid ranges from -105 m (at the south of India) to +85 m (at New Guinea) [67]. The shape of the geoid cannot be described mathematically but tabular models are available. A diversity of regional and national grids exists but the most largely used worldwide model is the US NASA Earth Gravitational Model 1996 (EGM96). The distance between a point of space and the geoid is named orthometric height or height above mean sea level. In air vehicle domain, the height refers to the distance between the aircraft and the ground surface. The terrain topography must then be taken into account. Figure 2.3 summarises the conventions employed for denoting the heights of a point of space. It can be noticed that the distance between the geoid and the ellipsoid (GE in this figure), is called geoidal separation.

East, North Up Coordinates

Local East, North Up (ENU) is a Cartesian coordinate system defined to be tangent to the Earth geoid at a defined origin. The North axis is tangent to the meridian that contains the ENU origin and in the North direction. The East axis is normal to the North axis and is in the positive longitudes direction. Finally, the Up axis is chosen so that the ENU is a right-handed coordinate system (Figure 2.2). ENU, also known as local tangent plane or navigation plane, is particularly convenient for the computation of motion on short distances (few kilometres). As Cartesian coordinates facilitates the computation of distances, ENU is widely used in the navigation tasks (e.g., map-matching, path planning). Conversion from ECEF to ENU coordinates can be found in [38].

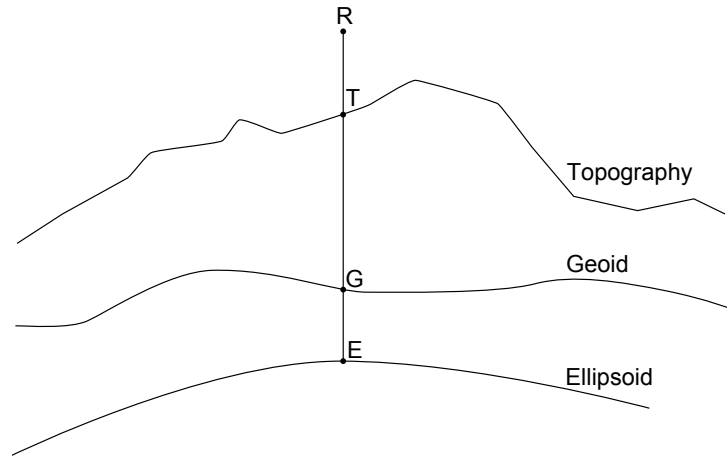


Figure 2.3: Heights of a point of space R . The ellipsoid height is RE and the orthometric height is RG (after [67])

2.2.2 GNSS Principles

Several technologies have been developed to compute the vehicle global position using satellites and referred by the generic term GNSS. The first and the most widely used is the American GPS (Global Positioning System). Today, the Russian GLONASS (GLObalnaya NAVigatsionnaya Sputnikovaya Sistema) also provides world coverage facilities for positioning. The European Union with Galileo project and the China with Compass (the extension of the regional system Beidou) are currently developing other alternatives for global positioning. These all rely on the same basic principle which is the measurement of the time-of-flight of a radio-signal between satellites whose position is known and the receiver.

An electromagnetic signal is emitted by every satellite and contains the emission time. The receiver decodes this signal and measures the time difference between the emission and the reception. Knowing the speed of the signal, the pseudo-range between the visible satellites and the receiver can be measured. All the satellites carry precise atomic clock and are precisely synchronised (in fact, the offsets and the drifts of their clocks are precisely monitored). However the receiver usually does not benefit from this technology. It is assumed that the receiver clock is precise only on short period of time but drifts with time. The receiver clock offset must be therefore solved along with the coordinates. If the coordinates of the satellites in ECEF are denoted by (x_i, y_i, z_i) and the ρ_i the corresponding pseudo-range, the receiver position (x_r, y_r, z_r) and the clock offset can be calculated by solving a non-linear system made of the following equations:

$$\rho_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} + c \cdot \Delta_t \quad (2.2)$$

where c is the signal speed. There are four unknowns, so four time-of-flight measurements are required to solve the receiver position.

The GNSS principles are simple but it involves complex systems and models, and the errors on the calculated position result in the combination of many smaller errors.

The measured pseudo-range must be corrected according to known effects like atmospheric delays and diffraction, relativity effects, antenna signal propagation, etc. It is assumed that the errors due to all these effects can be allocated to pseudo-ranges individually which is named User Equivalent Range Error (UERE). Models are used to estimate the contribution of each effect on the pseudo-range error in order to compensate them. In addition, the geometry of the satellite constellation at the measurement time changes the observability of the receiver coordinates. For example, if all the satellites are in the same meridian plane, the normal distance to this plane is fully correlated with the clock offset. Both cannot be determined simultaneously [52]. The contribution of the constellation geometry to the position error is denoted by the Geometric Dilution Of Precision (GDOP).

In order to reduce error on the position solution, the underlying idea of many approaches rely on the use of redundant pseudo-ranges. The redundancy degree is the difference between the number of available measurements and the minimum number of measurements required to solve the position (four in GNSS positioning). Assuming that all these measurements are reliable, a least squares approach can be used in order to improve the solution accuracy. However, the main issue in GNSS positioning is the presence of non reliable pseudo-ranges.

The first reason for non reliable pseudo-range measurement is dysfunction of satellite clocks. The drift of each satellite clock is regularly evaluated by the GNSS control segment and broadcast to the receiver via the satellite. It can therefore be compensated by the receiver. However, before time drift update, the error induced may be unacceptable of a given application. In the Receiver Autonomous Integrity Monitoring (RAIM) algorithm, these redundant measurements can be used to perform consistency checks and detect if one of the pseudo-ranges causes an unacceptably large error according to specifications [23]. Based on this and given false-alarm and miss-detection probabilities, it is possible to determine a domain that contains the position solution which is called Protection Level [67]. In these approaches the assumption taken is that there is at most one faulty measurement at each time the measurements are taken.

The second reason for non reliable measurement is the reflection of the satellite signal on elements of the receiver surrounding environment like buildings or mountains. This causes the signal to travel a longer distance than the actual distance between the satellite and the receiver. This occurs mainly in urban areas where a limited number of satellites is within the receiver line-of-sight. The Non-Line-Of-Sight (NLOS) satellites can be identified using information on the terrain elevation or building heights provided by a three-dimensional navigation map [93, 92, 89]. According to an *a priori* position, the NLOS satellites are identified by comparing their

elevation to the heights of surrounding environment and excluded for the position calculation. When reflections occur, the signal is polarised and the signal-to-noise (SNR) ratio decreases. NLOS signals can therefore be excluded using polarised antenna and by increasing the SNR threshold [56]. Others simply reject inconsistencies in the measurements based on the use of Interval-based methods and the relaxed intersection of constraints [65, 39].

GNSS will take benefit of the development of new satellite constellations like Galileo or Compass thanks to the increasing number of available satellites and the improvement of error models. However, localization in places where the sky is not visible from the antenna will remain impossible to solve with GNSS technology. Different methods exist to enhance the performance of GNSS by combining its output with proprioceptive and exteroceptive sensors data.

2.2.3 Dead Reckoning

Passenger vehicles are now equipped with a wide variety of sensors originally designed for basic functions like engine control, steering control, brake control with ABS (Anti-lock Braking System), trajectory control with ESC (Electronic Stability Control), etc. In order to reduce the wire length in the vehicle, the sensors and the control units (ECU) communicate via a central network named CAN-bus (Controller Area Network). Sensor data is therefore easily accessible for new functions with localization purpose.

In the marine navigation domain the task of estimating the motion of the vessel without any global references available (when the stars were not visible) was named Deduced Reckoning (DR). The shorten name *Ded. Reckoning* is usually written *Dead Reckoning*. DR provides estimates of the vehicle location relative to the starting point, however as the distance increases, errors are introduced due to calibration, slippage, noise, etc. Fusion of DR and GNSS are employed to benefit from the advantages of both techniques: reduce the drift, improve availability and provide a global position [20]. Since the rate of DR sensors is usually significantly higher than the one of GNSS, the idea of the methods presented here is to estimate the vehicle motion and to merge it with the GNSS position when available.

2.2.3.1 Modelling

According to the available data, different kinematic models can be used. The simplest model consists in representing the vehicle by a single oriented point. Assuming that the vehicle has a 2D motion and that its speed (v_t) and yaw rate (ω_t) can

be measured, its coordinates (x, y) in a planar working frame are given as follows:

$$\begin{cases} x_{t+1} = x_t + v_t \cdot T \cdot \cos(\psi_t + T \cdot \omega_t) \\ y_{t+1} = y_t + v_t \cdot T \cdot \sin(\psi_t + T \cdot \omega_t) \\ \psi_{t+1} = \psi_t + T \cdot \omega_t \end{cases} \quad (2.3)$$

where (x_t, y_t) are the vehicle position coordinates at time t . ψ_t is the orientation of the vehicle speed with respect to East (i.e. horizontal) axis and positive anticlockwise at t equals 0. The Subscripts t and $t + 1$ indicate the time steps. T is the sampling period.

Due to technological and cost constraints, the data available is limited in rate, accuracy, precision and resolution (see Appendix A). For example, the gyroscope measuring the vehicle yaw rate was designed initially to be highly dynamic in order to correct the vehicle trajectory by acting on brakes (ESC) on short periods of time. The measured yaw rate (ω_g) can be modelled as a function of the vehicle yaw rate and a bias b which mainly depends on the sensor temperature. This is presented as follows:

$$\omega_g = \omega + b \quad (2.4)$$

Without absolute measurements like GNSS for instance, the problem is to estimate this bias. A technique often used and named Zero Velocity Update is to estimate it based on the mean of the yaw rate when the vehicle is stationary. This method is efficient when the vehicle does frequent stops. However, passenger vehicles may travel long distances with few stops. Other approaches based on wheel speed sensors allow to estimate the vehicle yaw rate and deduce the gyroscope bias while the vehicle is moving. Front or rear wheel speed measurements can be used for this purpose based on the kinematic model presented in Figure 2.4 [100]. These are presented as follows.

When using the front wheels, the following equations can be set:

$$\begin{cases} \sin(\beta_l) = \frac{e}{R_{fl}} \\ \sin(\beta_r) = \frac{e}{R_{fr}} \end{cases} \quad (2.5)$$

where R_{fl} and R_{fr} are the curvature radii of front left and front right vehicle wheels respectively. β_l and β_r are the left and right wheel angles respectively. e is the wheelbase.

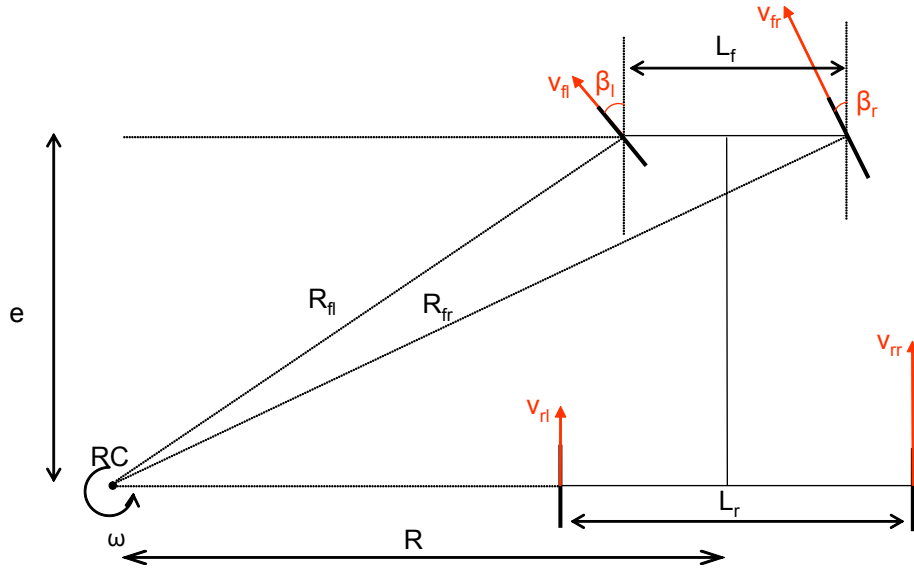


Figure 2.4: Kinematic model for front steered vehicle.

In addition:

$$\omega = \frac{v_{fl}}{R_{fl}} = \frac{v_{fr}}{R_{fr}} \quad (2.6)$$

where v_{fl} and v_{fr} are the linear speeds of front left and front right wheels respectively. The yaw rate can therefore be estimated with these two equations:

$$\begin{cases} \omega = v_{fl} \cdot \frac{\sin(\beta_l)}{e} \\ \omega = v_{fr} \cdot \frac{\sin(\beta_r)}{e} \end{cases} \quad (2.7)$$

There is usually no sensor measuring the steering angles β_l and β_r separately in the vehicle. However, lookup tables can be used to estimate them based on the steering wheel angle.

The use of estimates of the steering angles of the front wheels can induce errors in the kinematic model. Those are due to wheel slippage especially in high speed and/or high steering angles. Moreover, passenger vehicles are designed to be under-steered for stability purposes which is the case of the test vehicle used in this thesis. For this reason, the gyroscope bias is estimated using the speed difference between the rear wheels (this data is indeed available on the test vehicle).

Given

$$\begin{cases} v_{rl} = \omega \left(R - \frac{L_r}{2} \right) \\ v_{rr} = \omega \left(R + \frac{L_r}{2} \right) \end{cases} \quad (2.8)$$

where R is the curvature radius at the midpoint of the rear vehicle axle and L_r is the rear vehicle track. v_{rl} and v_{rr} are the linear speeds of rear left and rear right wheels respectively (as shown in Figure 2.4). The difference of the rear wheel speeds is as follows:

$$r = \omega \cdot L_r \quad (2.9)$$

Where $r = v_{rr} - v_{rl}$.

2.2.3.2 Proprioceptive Sensors Fusion

Bayesian techniques are efficient to fuse vehicle positioning sensors. In the particle filter approach, the uncertainty of the vehicle position is represented by a set of possible positions called particles. A weight is associated to each particle to represent its “quality”. The position of each particle is predicted and the particle weights are updated when a measurement is available. The lowest weighted particles are eliminated during the re-sampling process [102]. This recursive filter is convenient to deal with non-Gaussian processes since the re-sampling can be processed according to any combination of probability density functions. The prediction and update processes are done independently on each particle, so this filter requires high computational resources or dedicated hardware architectures [10].

In the Extended Kalman Filter (EKF), the Gaussian uncertainty model assumption makes possible a closed form of the fusion problem and an efficient implementation [19].

The equations for the EKF are described hereafter as they are used as part of the formulated approach. Let X_t be the vehicle state vector at time t :

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \omega_t \\ b_t \\ \psi_t \end{bmatrix} \quad (2.10)$$

where $(x_t y_t)$ are the vehicle coordinates in the East-North-Up (ENU) local reference frame. ψ_t stands for the orientation of the vehicle speed with respect to East axis and positive anticlockwise. ω_t is the vehicle yaw rate positive anticlockwise. Finally, b is the vehicle's gyroscope bias as defined in (2.4). The observation vector Y is defined as follows:

$$Y_{DRt} = \begin{bmatrix} \omega_{gt} \\ r_t \end{bmatrix} \quad (2.11)$$

where ω_g is the yaw rate as measured by the vehicle gyroscope and r is the speed difference between the rear vehicle's wheels. Based on (2.9):

$$r_t = \omega \cdot L_r \quad (2.12)$$

The vehicle state representation for DR is:

$$\begin{cases} X_{t+1} = f(X_t, v_t) + \alpha_t \\ Y_{DRt} = H_{DR} \cdot X_t + \beta_t \end{cases} \quad (2.13)$$

The DR observation matrix H_{DR} is as follows:

$$H_{DR} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & L_r & 0 & 0 \end{bmatrix} \quad (2.14)$$

The kinematic model introduced in Section 2.2.3 is used to derive the state transition. This results in the following vector:

$$f(X_t, v_t) = \begin{bmatrix} x_t + T \cdot v_t \cdot \sin(\psi_t) \\ y_t + T \cdot v_t \cdot \cos(\psi_t) \\ \omega_t \\ b_t \\ \psi_t + T \cdot \omega_t \end{bmatrix} \quad (2.15)$$

where the measured input is the vehicle's speed v_t and T is the sampling period.

The model and observation noises (α_t and β_t respectively) are supposed to be white and zero mean. Noises on each evolution model equation are assumed to be inde-

pendent, so the model noise covariance matrix Q_X is as follows:

$$Q_X = \begin{bmatrix} \sigma_{X_1}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{X_2}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{X_3}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{X_4}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{X_5}^2 \end{bmatrix} \quad (2.16)$$

The driving conditions in which the vehicle is used comply with the vehicle kinematic model chosen here (normal driving). The model noise is therefore set at a low value for evolution of x , y and ψ : $\sigma_{X_1}^2 = \sigma_{X_2}^2 = \sigma_{X_5}^2 = 10^{-8}$ (SI units). However, the constant vehicle yaw rate model is less realistic, so the standard deviation of the corresponding noise is set higher: $\sigma_{X_3}^2 = 10^{-2}$. The gyroscope bias is assumed to be constant in the model but it is actually likely to change over long periods of time (due to sensor temperature for instance). $\sigma_{X_4}^2 = 10^{-8}$ is therefore chosen.

A ground truth positioning sensor (inertial measurement unit) has been used to empirically determine the first and second statistical moments of the speedometer, yaw rate gyroscope and differential rear wheel speedometer. Consequently, the input variance is chosen as $Q_v = 4.9 \cdot 10^{-3}$. Moreover, the DR observation covariance matrix is as follows:

$$R_{DR} = \begin{bmatrix} \sigma_{\omega_g}^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \quad (2.17)$$

Where $\sigma_{\omega_g}^2 = 3.6 \cdot 10^{-5}$ and $\sigma_r^2 = 10^{-3}$.

DR requires initialisation of the state vector X_0 and covariance matrix P_0 . In order to facilitate the algorithm convergence during the first iterations, they can be set with coherent values. If no *a priori* information is available, the following values are used:

$$X_0 = [0 \ 0 \ 0 \ 0 \ 0]^T \quad (2.18)$$

$$P_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.19)$$

2.2.4 GNSS and DR Fusion

The GNSS receiver provides position observation at a lower rate than vehicle proprioceptive sensor. The full observation vector Y when GNSS is available is as follows:

$$Y = \begin{bmatrix} x_{gnss} \\ y_{gnss} \\ \omega_g \\ r \end{bmatrix} \quad (2.20)$$

The observation model and noise covariance matrix are H and R respectively:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & L_r & 0 & 0 \end{bmatrix} \quad (2.21)$$

$$R = \begin{bmatrix} \sigma_{x_{gnss}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y_{gnss}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\omega_g}^2 & 0 \\ 0 & 0 & 0 & \sigma_r^2 \end{bmatrix} \quad (2.22)$$

Since the GNSS receiver used in the first experiments did not provide latitude and longitude standard deviations, $\sigma_{x_{gnss}}^2$ and $\sigma_{y_{gnss}}^2$ are set to 4 m^2 according to a study done with the positioning ground truth.

The extended Kalman filter is presented in Algorithm 2.2.

2.2.5 Road Centreline Positioning

Road features like lane markings can be used to provide relative position estimates. Lane detection using cameras is becoming a standard feature on recent passenger vehicles. Computer vision and computing power is making such progress, that the level of reliability increases and thus cameras are being used for multiple functions (e.g., pedestrian detection, vehicle detection, lane markings) [43, 78, 47]. Lane marking detection can be used to provide lateral constraint position estimates. Further it can be associated to data stored in the navigation maps to provide better localisation estimates as in [112]. This approach and results of the knowledge acquired in this thesis were used on board of an autonomous vehicle research platform [63]. An example of on board camera detection is given in Figure 2.5.

Algorithm 2.2 Extended Kalman filter for vehicle localization

Inputs: Sensor observations $(x_{gnss}, y_{gnss}, \omega_g, r)$ and vehicle speed v **Outputs:** Vehicle state X and Covariance matrix P **Begin**

$$X \leftarrow X_0$$

$$P \leftarrow P_0$$

$$K \leftarrow P \cdot H^T \cdot (H \cdot P \cdot H^T + R)^{-1}$$

$$K_{DR} \leftarrow P \cdot H_{DR}^T \cdot (H_{DR} \cdot P \cdot H_{DR}^T + R_{DR})^{-1}$$

When new proprioceptive data is available

$$A \leftarrow \frac{\partial f}{\partial X}(X, v)$$

$$J \leftarrow \frac{\partial f}{\partial v}(X, v)$$

$$X \leftarrow f(X, v)$$

$$P \leftarrow A \cdot P \cdot A^T + J \cdot Q_v \cdot J^T + Q_X$$

If new GNSS data is available

$$Y \leftarrow [x_{gnss} \ y_{gnss} \ \omega_g \ r]^T$$

$$K \leftarrow P \cdot H^T \cdot (H \cdot P \cdot H^T + R)^{-1}$$

$$X \leftarrow X + K \cdot (Y - H \cdot X)$$

$$P \leftarrow (I_5 - K \cdot H) \cdot P \cdot (I_5 - K \cdot H)^T + K \cdot R \cdot K^T$$

Else

$$Y_{DR} \leftarrow [\omega_g \ r]^T$$

$$K_{DR} \leftarrow P \cdot H_{DR}^T \cdot (H_{DR} \cdot P \cdot H_{DR}^T + R_{DR})^{-1}$$

$$X \leftarrow X + K_{DR} \cdot (Y_{DR} - H_{DR} \cdot X)$$

$$P \leftarrow (I_5 - K_{DR} \cdot H_{DR}) \cdot P \cdot (I_5 - K_{DR} \cdot H_{DR})^T + K_{DR} \cdot R_{DR} \cdot K_{DR}^T$$

End if**Return** (X, P) **End when****End**



Figure 2.5: Example of scene detection by the vehicle front camera (from Mobileye development software).

In this thesis, the position of the road centreline is required to evaluate the navigation system integrity. If a smart camera is on board the vehicle, it can be estimated based on the lane detections and the vehicle estimated position². The following paragraphs detail this process.

The reference coordinate system used by the camera is linked to the vehicle. Its origin is at the middle of the front bumper and the z -axis goes in the vehicle direction. The x -axis is horizontal and goes to the right hand side of the vehicle. The y -axis is vertical and downward. The lane markings are described by a $x(z)$ second order polynomial function and the c_0 , c_1 and c_2 coefficients are provided. These are illustrated by Figure 2.6 defined as follows:

$$x(z) = c_0 + c_1 \cdot z + \frac{c_2}{2} \cdot z^2 \quad (2.23)$$

The lane marking type and the confidence in the detection are also provided for any lane marking detected by the camera. Using this information, the position of the road centreline as recorded in the navigation map can be measured based on the vehicle position estimation and the camera lane marking detection. In local tangent plane coordinates, the centreline coordinates at the vehicle position for a single lane

²This sensor was not available in the test vehicle

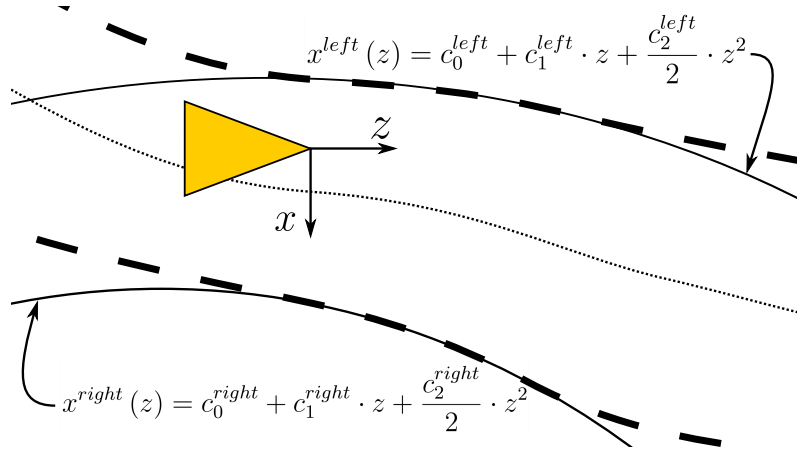


Figure 2.6: Camera lane markings model. The vehicle is in yellow and is represented with the camera coordinate system. The real lane marking is in dashed line and the regression made by the camera in solid line. The dotted line is the road centreline.

road is (x_l, y_l) :

$$x_l = x + \frac{c_0^{left} + c_0^{right}}{2} \cos(\psi) \quad (2.24)$$

$$y_l = y - \frac{c_0^{left} + c_0^{right}}{2} \sin(\psi) \quad (2.25)$$

(x_l, y_l) can finally be converted in geodetic coordinates.

2.3 Navigation System

For the purpose of this research, the Navigation System is defined by the following components: the Navigation Maps, the Map-Matching algorithms and the Electronic Horizon. Details of each of these components are included in the following sections.

2.3.1 Navigation Maps

The navigation maps are understood here as databases that contain information on the road context and dedicated to be used in real-time in the vehicle. Multiple purposes are addressed by the navigation maps which determine constraints on its content and data structure. The trip planning functionality is required to make the

vehicle able to reach a desired destination from a given starting point. This implies that the navigation map must describe the topology (i.e. connectivity) between places and the cost to reach one place from another (e.g., distance, trip duration, energy). Functions like trajectory planning require a more precise representation of the road to operate. There are therefore several navigation maps formats.

In order to maintain driving comfort and safety, the road construction rules require the road curvature to be continuous [116]. Roads are then sequences of straight lines, transition section, constant curvature bends and transition section before the next straight line. In the navigation map formats presented in [17, 105, 27], the road network is modelled using three basic elements, straight lines, clothoids (i.e. curves whose curvature changes linearly with the curve length) and circles. The road in the navigation map therefore accurately reproduces the characteristics of the true road using a few parameters. This is therefore an efficient way to model the road geometry. The RoadXML standard defines a similar way to model the road that handles information on the transversal profile and road material [3]. This format is suitable for vehicle dynamics simulation. However, this format uses local coordinates which makes applications to large scale navigation functions inappropriate.

In most of the global navigation maps, geometry is described by ordered sequences of points named Nodes. The coordinates and the connectivity between each node is stored in the database and it is assumed that the geometry between parent and child nodes is a straight line named Links. As a consequence, a large number of nodes is then required to model accurately a curve. Nevertheless, the navigation task is made easy using this format since graph search algorithm are applied to find the optimal path between two nodes like for example A*, Dijkstra or Genetic Algorithm [36, 82]. The other functions of the intelligent vehicle require more than navigation planning. In [106, 26], multi-layer navigation maps are proposed in which one polyline is drawn for each lane, lane marking, curb, etc. A final raster layer represents the obstacles perceived by the sensors in order to plan local trajectories.

In the navigation maps currently used in passenger vehicles and which are considered in this work, the geometry is also modelled by polylines. One polyline represents a whole carriageway in order to reduce the size of the database. Navigation maps that cover several countries can therefore be used in the vehicle despite the limited memory available in passenger vehicles. These maps were initially designed for trip planning and turn-by-turn guidance so cartographers centred on the connectivity between roads. Attributes are associated to each road to describe them qualitatively. Due to the development of location based services and contextual ADAS, the quantity of information grew mainly by adding new attributes (e.g. number of lanes, marking types, driving direction, speed limits).

The automatic evaluation of the quality of a navigation map is usually addressed in a centralised manner. A reference (i.e. ground truth) navigation map can be used to evaluate the vehicle map (subject map). For each road of the subject map, the corresponding road in the reference map is found and the comparison is made.

This process is straightforward if the roads have the same ID in both maps. If not, the matching can be done by different methods like buffer zones, bounding boxes, Voronoï diagrams, etc. [35]. The similarity between two corresponding roads is then calculated. The Euclidean distance is often used when comparing two points but it must be adapted when dealing with curves or polylines. The similarity between two curves may be measured as the maximum gap between them using the Hausdorff distance [80]. This is a good metric to detect road offsets and some dissimilarities in the curves shapes. However, it may be misled if the segments have different lengths or very different shapes [11]. The mean distance introduced in [77] takes into account the surface between the curves. In [55], the authors use fuzzy logic to compute an outlier index that expresses how a geographical object belongs to its spatio-temporal neighbourhood. This approach aims at detecting faults as well as temporal changes in maps. Studies were led on large scale databases and especially in volunteered geographical information like OpenStreetMap (OSM) in [53, 87, 59]. Methods inspired from the SLAM (Simultaneous Localization And Mapping) domain can also be employed. The position information given by the navigation map is treated as an observation like other sensors observations. In [22], an Extended Kalman Filter is implemented according to this idea.

To be considered as a ground truth, the reference navigation must be created by precise and complete survey. In the literature some works aim at using other sources of information like aerial imagery [34, 32, 94] or the mining of a large number of GNSS tracks [107, 7] to create the reference map. In these approaches, the assumption that the disparities between reference and subject maps are due to faults in the subject map is made. The problem of fault in the reference map (due to offset in aerial imagery or recent evolutions of the road network) or in both maps are not addressed. In the approach presented in this thesis, a reference is unavailable. Moreover, due to the autonomous and real-time requirements, the data-mining approaches are inappropriate.

2.3.2 Map-Matching

Map-matching is required to determine in which road of the navigation map the vehicle is travelling and to be able to extract relevant information for client systems (ADAS or autonomous driving functions). The vehicle position estimated by the GNSS receiver contains significant errors and may not fit exactly the road of the navigation map. Map-matching consists therefore in finding the road of the navigation map that best fits the GNSS estimate.

The simplest way is to find the closest point in the navigation road network to the GNSS measurement [54]. This geometric matching method is limited when travelling in dense road networks, when GNSS conditions are poor (typically in urban environments) or approaching intersections. The GNSS position is likely to be matched to the wrong road, thus the vehicle heading is used to discard inconsistent road

candidates. A buffer of the vehicle trajectory can also be used instead of singular GNSS measurement to find the road portion on which the vehicle is travelling [66]. Monte Carlo based approaches like Particle Filter can be used when the available computational resources are sufficient. The particles are spread on the whole road network (or on a area large enough to contain the final solution). The particle population evolves with available measurements (e.g., GNSS measurement, DR estimation) and finally concentrates on the final solution [108, 58, 111]. In [49, 76], the road candidates are represented by a set of hypotheses. A Bayesian framework is used to choose the most likely road. The Evidence theory can also be used since it is a convenient way to handle conflict in data fusion [83].

In [64], the map-matching process is considered as a localization problem within an *a priori* constrained localization space. A Bayesian approach is used to evaluate the likelihood of the map-matched positions given GNSS and navigation map observation. Once the map-matched position is found, the authors propose a multi-hypotheses tracking formalism to determine the matched position at the consecutive time steps. In [90], a similar way to proceed is used by distinguishing the computation of the first map-matched position, known as Initial Matching Process and the tracking from the initial matched position called Subsequent Matching Process. The map-matching algorithm of navigation function developed for testing purpose in this thesis is based in this method (details are in Appendix A).

Fuzzy logic can also be used to address the complexity of the map-matching problem and the large number of criteria involved in choosing the road candidate. In [97], the author use a Sugeno fuzzy inference system to choose the road in the navigation map based on the position uncertainty, the distance between the road and the vehicle position estimate and the angle between the road direction and the vehicle heading.

These approaches include a measure of confidence in the map-matched position. The nature of this index depends on the method employed. It can be a function of the distance between the GNSS estimate and the map-matched position for geometric matching, the posterior road probability for Bayesian approaches or the final conflict in evidential algorithms. Section 2.4 details why this index cannot be taken as a map quality index.

2.3.3 Electronic Horizon

Once the position of the vehicle is matched onto the navigation road network, the relevant contextual information can be extracted and sent to client systems like ADAS or autonomous driving functions. In most applications, the vehicle needs to anticipate the near future situation, thus client systems also need this information several metres ahead. For instance, Contextual ACC requires the future road curvature in order to smoothly adapt the vehicle speed when approaching a bend. This information is sent in the EH (Electronic Horizon) also known as ADAS horizon. It includes navigation map information at the vehicle position and for all the likely routes ahead

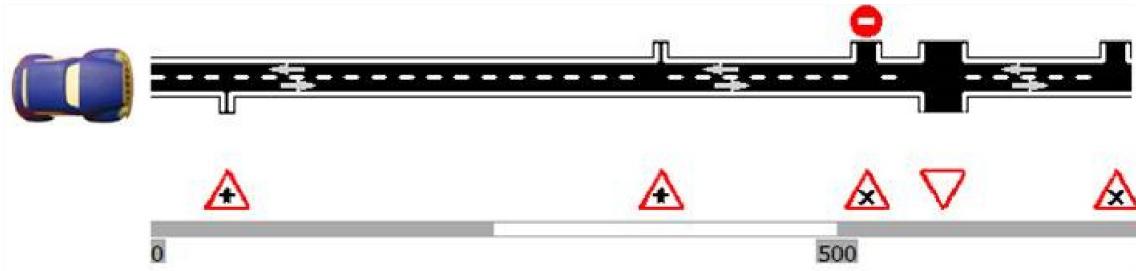


Figure 2.7: Electronic horizon from ADASIS protocol specifications [103].

as shown in Figure 2.7: map-matched position, road curvature, speed limit, distance to next intersection, etc. Unless the driver has a planned route, the routes that the vehicle is likely to follow needs to be determined. Several approaches exist and can be found in [28, 50]. A communication standard named ADASIS (Advanced Driver Assistance Interface Specification) has been developed [103]. It defines the CAN-bus communication protocol that allows the transmission of the EH to the client systems independently from the navigation system. Appendix A includes further details on the EH used in the experiments of this thesis.

2.4 Integrity Monitoring

The vehicle navigation system usually provides a confidence index associated with the map-matched vehicle position. This corresponds to the final score of the optimisation process employed in the map-matching according to a given GNSS estimate and a given navigation map. However, this index should not be taken as a measure of quality of the navigation map.

This issue is illustrated by taking the following road network example as shown in Figure 2.8. A sparse road network can be observed, the road stored in navigation map is 8 m to the side of the real road. However, as the number of road candidates for the map-matching is low, the algorithm would select the closest road with a high confidence index, despite the fault in the navigation map. From the point of view of the client systems, the map-matched position estimate seems to be reliable but this geometric fault may cause malfunction of intersection or curve warning systems.

This section details approaches that aim to evaluate the integrity of the information provided by navigation systems.

The approach in [110] is applied to the domain of indoor mobile robotics. It presents a similar architecture to the navigation systems studied in this thesis. The navigation system consists in DR localization and path planning based on the obstacles stored in the navigation map and perceived by sensors. Some obstacles are neither



Figure 2.8: High confidence map-matching on a map fault. The road (circled in red) in the navigation map is 8 m aside the real road (assuming a perfect imagery). Since the road network is sparse, the confidence would be high despite this offset (OpenStreetMap data on top of Bing imagery).

perceived nor recorded in the map. They block the robot causing navigation failures. The robot has a limited number of low cost sensors. The localization is done using a Kalman Filter. It is shown that detecting unexpected large deviations of the innovation signal is not sufficient since it is very sensitive to outlier due to the noise of poor quality sensors. A Cumulative Sum (CUSUM) test is therefore implemented to reduce the number of false alarms. A similar method is applied in the fault detection task formulated in this thesis (Chapter 5).

An integrated localization and map-matching process based on lane level clothoid maps and using a particle filter is presented in [114]. An integrity monitoring approach is also introduced and is associated to the position solution which therefore takes into account both steps. Two metrics are defined: the lane occupancy probability which indicates the trust in the lane assignment and the lane positioning protection level which shows the confidence on the vehicle position within the assigned lane.

The concept of user-level integrity is introduced in [115] to emphasize the necessity of taking into account every step of the positioning process, GNSS, navigation map and map-matching in the intelligent vehicle position integrity monitoring problem. A strategy relying on successive evaluation of GNSS integrity, map complexity and map-matching solution integrity is presented.

The structural elements in navigation maps (e.g., lane markings, traffic sign) are also addressed in the literature. The introduction of smart cameras having capabilities to detect features associated to roads has been a major enabler. A typical application is

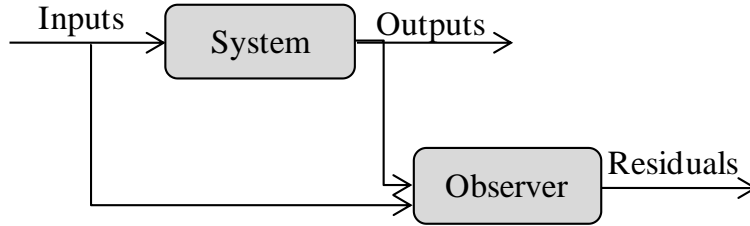


Figure 2.9: Residual generation for system monitoring

the detection of speed limit signs. Several approaches exist to determine the speed limit when the camera perceives a speed limit sign different from the one stored in the navigation map. The belief theory is convenient for the fusion of discreet information with high level of uncertainty or conflict. This formalism can be used to assess the current speed limit based on the navigation map information (e.g., position confidence, navigation map speed limit, road class, driving context) [31, 84]. A similar approach based on Bayesian fusion is developed in [12].

The pertinence of information stored in the navigation map applied to roadworks situations is studied in [75]. In such cases the stored map is irrelevant and should be ignored by client ADAS or automated driving functions. For this purpose a probabilistic framework based on Gaussian Processes and the use of a Hidden Markov Model is proposed. This detects the presence of roadworks based on environment features perceived by the vehicle front camera as well as information fed from a server that centralises roadwork information.

The system monitoring approach is a promising domain since it enables the real time and autonomous evaluation of the integrity of a system as a whole.

The underlying principle is shown in Figure 2.9. The system inputs and outputs are fed into an observer which transforms them into a set of residuals [37]. These are null when no fault affects the system. If faults exist, the residuals is non-null. When faults are detected, the consequences they have on the system are observed. Thus an association between the faults and their effects on the system would lead to their unambiguous identification and hence isolation. These faults can be excluded and/or corrected from the system to keep it operating correctly or at a different level of performance. This process is known as Fault Detection, Isolation and Adaptation (FDIA).

Based on the system model and the available measurements, a logical link between faults and the residual value can be established which is summarised in a signature matrix. This principle is illustrated by an example issued from [15]. Given a system model, let the three residuals r_i depend on faults f_j affecting measurements such as detailed in Table 2.1. In this example, the faults f_1 and f_2 produce the same effects on the residuals and are therefore detectable but not isolable. The fault f_3 has a unique signature and can therefore be isolated. A complete framework that aim at

Table 2.1: Signature table

	f_1	f_2	f_3
r_1	1	1	0
r_2	1	1	0
r_3	0	0	1

detecting multiple faults in a system is presented in [69]. The sensitivity of a set of residual is established using on the system model. Diagnoses to apply are settle based on the observed residuals. A similar approach is developed in this thesis for navigation system fault detection isolation and adaptation (in Chapter 4).

This boolean approach applied to the residual value must be adapted when real signals are used. Due to the noise in them, different change detection strategies must be applied. An extensive description of the mathematical tools available for signal change detection can be found in [14].

2.5 Conclusion

The principles of the systems as part of the navigation function in intelligent vehicles were studied in this chapter. It was shown that estimating the vehicle localization remains a challenge due to the physics involved and cost constraints on the sensors used. GNSS based localization system are convenient, However, these are subject to disturbances in complex environments. Different methods were presented to make the localization robust to signal multipath or satellite faults based on the use on vehicle sensors. It was then shown that the vehicle localization is used to determine the vehicle location on the navigation map. This map-matching process was presented and methods for improving its performance and evaluating its integrity introduced. It was finally shown that monitoring the integrity of each step of the navigation system independently is insufficient to evaluate the integrity of the final result. System monitoring approaches that can be applied to the entire system were therefore introduced.

Most of the contributions presented in this chapter use raw information as the basis of integrity monitoring. For example GNSS methods detect the presence of multipath by observing independently the pseudo-ranges. The map-matching integrity evaluation is based on the examination of the surrounding roads in the navigation map. However, with the assumptions taken in this thesis, the entire navigation system must be taken as a black box. The challenge therefore relies on the evaluation of the information contained in the EH without access to the raw data.

This review showed that the integrity evaluation of the navigation function cannot be accomplished using only one formalism. The information provided by the navigation

system in the EH is diverse (e.g., map-matched position, road curvature, speed limit, type of intersection) so the methods applied to evaluate it must reflect this diversity. A common feature of all these methods is to create redundancy. At least one observation independent from the navigation system is required. The level of redundancy introduced and the type of generated information depend on the variables observed in the navigation map. In general, for this purpose, additional sensors are introduced. However, their use is constrained by the application domain, their layout, cost, etc. This is a major constraint from an industrial perspective.

The methods developed in the next chapters to determine the integrity of the navigation function are based on the use of minimal sensor redundancy to comply with industrial constraints.

Chapter 3

Map Structural Faults Identification

Contents

3.1	Introduction	45
3.2	Case Study: Roundabouts	46
3.3	Description of the Proposed Method	48
3.4	Experimental Validation	59
3.5	Conclusion	61

3.1 Introduction

A navigation map contains a variety of information that is used in different manners by the client systems within intelligent vehicles. As introduced in Section 1.2, a distinction has been made between different structural navigation map faults.

The identification of road connectivity faults prevents suboptimal planning and avoids the driver to be asked to drive on restricted roads for instance. Road connectivity faults are in practice very rare and so are not addressed in this manuscript.

Another structural fault listed in the pathology is the wrong representation of intersections in the navigation map. For instance, access lanes, exit lanes, roundabouts, undefined traffic zones can be missing. In order to address this kind of fault, the monitoring system has to interpret the information provided by the sensors up to a high level of understanding with semantic attributes. A classical methodology to do so is to use machine learning and classification methods similarly to pedestrian recognition in intelligent camera systems.

In this chapter, the focus is on the particular case of the identification of missing roundabouts in the navigation map which is of particular importance in Europe. A recognition system based on supervised learning and Bayesian classification is proposed and evaluated.

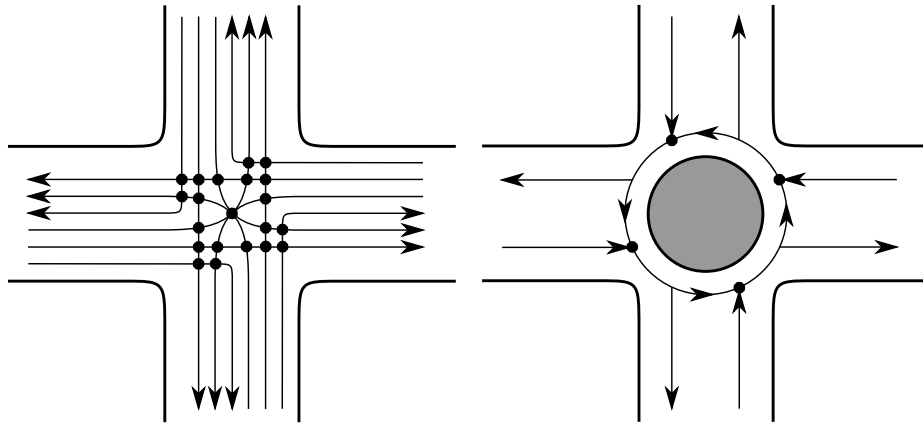


Figure 3.1: Conflict points for an uncontrolled two-lane intersection without and with a roundabout (after [46])

3.2 Case Study: Roundabouts

In the European Union, approximately 22% of annual road fatalities occur at junctions. Despite the fall of the overall number of fatalities, this proportion remained constant over the last decade [88]. Efforts are made to reduce the approaching vehicle speed by changing the shape and the hierarchy of the junction. In this context, roundabouts (i.e. traffic circles) are extensively introduced. Even if the crash frequencies at roundabouts is similar to other intersections, the severity is lessened since it reduces up to 90% of the crashes involving death or incapacitation injuries compared to a classical four-stop intersection [104]. In addition to the lower speed imposed by the junction shape, the number of conflict points in a roundabout is significantly reduced compared to a classical crossing as depicted by Figure 3.1. Further and especially in European Union countries, the decision of creating a new roundabout can be taken by different authorities depending on the road class on which they are located. Similarly to any other road changes, there is no central register for the creation of new roundabouts. This makes very difficult for the map providers to keep databases up-to-date.

The roundabout is also a significant road feature in terms of ADAS. As these are created to reduce traffic speed on dangerous areas, they often result in changing the order of priority of the road. For example, severe accident occur at intersections on countryside primary roads due to the high speed difference between vehicles driving on the primary road and vehicle that just entered at the junction. Roundabouts are built to reduce oncoming vehicle speed. The primary road then lost the right of way at the intersection. In this situation, the ADAS should warn the driver or reduce the vehicle speed approaching the new roundabout which is not possible if it is not recorded in the navigation map. The first experimentations of autonomous driving in open road environment also showed that the roundabout is a challenging driving

situation [16].

In this chapter, the problem of detecting this particular structural fault which is the presence of a roundabout that is not recorded in the navigation map is addressed. Only a few contributions in the literature address this problem, however, as detailed in Chapter 2 road features can be detected with aerial imagery. In [101], an active contour algorithm is developed to detect object in the image and filters based on the object surface and mean curvature variation are applied to reject undesirable objects like cars and shadows. A similar approach is led with gradient propagation in [33]. As detailed in the thesis problem statement in Section 1.3, the solution studied in this work must be autonomous and use only embedded sensors. These methods do not meet these requirements.

In [42], a neural network is trained to classify the GPS traces with respect to driving scenarios like *road*, *local street*, *roundabout* or *traffic light stop* according to a set of parameters (speed, rate of acceleration, curvature or change in travel direction).

The observation made in this work is that the driving pattern while crossing a roundabout is characteristic. This term refers to the sequence of actions taken by the driver. When approaching a roundabout, the driver brakes until a low or zero speed, then accelerates and turn right, then turn left at constant radius and with moderate speed, then switches the right indicator on, then finally accelerates while turning right. This driving pattern can be felt by the car passengers especially with the eyes closed. The data related to this driving pattern is available in any recent passenger vehicle via the CAN-bus.

Roundabout detection methods based on classifying temporal sequences of signals such as speed, yaw rate, indicator states were investigated during the preliminary research of this thesis. However, treating these signals independently is not efficient. This results in unnecessary and complex computations which is not suitable for real time application. The approach presented here classifies the vehicle trajectory which intrinsically depends on the signals listed above. In addition, the trajectory is more robust to driving pattern variations. For example, in dense traffic, the vehicle may stay at zero speed for a few seconds before entering the roundabout; if there is no other vehicle, it may conversely go through the roundabout without reducing speed. The trajectory of the vehicle is less affected by these variations.

Roundabout detection can then be treated as a two-dimensional signal classification. Signal classification requires several steps and many mathematical methods have been developed in the literature depending on the application. These phases can be summarised as follows. First, a significant signal must be generated. It can be raw sensor data or signal obtained by calculation like a residual for instance. Second, a pre-processing step is often required to facilitate the use of the signal. This usually consists in normalisation and filtering. Third, in the classification (or pattern recognition) stage, the similarity of the signal with models is calculated. The signal models are samples that represent the typical value of the signal for each class. The process of creating these models is called learning and can be done

manually or automatically. The choice of the feature as well as the measure of similarity function is essential because it must reflect the variety of the classes and be robust to insignificant parameters. The signal is finally associated to the class with the higher similarity score.

The proposed approach is based on the classification of the past vehicle trajectory. The spatial representation of the trajectory that corresponds to a roundabout is characteristic. To take benefit from this, a method inspired from handwriting recognition domain is employed. The classification step can be done using various mathematical tools like Fuzzy Logic [72], Neural Networks, Hidden Markov Models [8], K-Nearest Neighbours, Support Vector Machines [18, 95, 25]. Probabilistic classification provides a good compromise between complexity and performance. A Bayesian classification is then used in this approach.

3.3 Description of the Proposed Method

In this section, the steps for identifying roundabout in the vehicle trajectory are detailed. First, the pre-processing consists in generating the vehicle trajectory on a fixed length sliding window. Second, the variables chosen to describe the shape of this trajectory are detailed. Third, the classes used and the recognition method are introduced. Finally the placement of the roundabout centre and the computation of its radius are detailed.

3.3.1 Pre-Processing

The pre-processing step consists in preparing the signal that will be used for classification so that it represents explicitly the feature to classify. Here, the trajectory of the vehicle must be computed. The localization is performed by the Extended Kalman Filter described in Section 2.2.3.2. The area of interest is isolated from the whole vehicle track using the concept of *sliding window*. The vehicle successive positions are accumulated in the memory until a finite distance. In other words, only the last L metres of the vehicle's trajectory are used to search for features. The vehicle's odometer is used to spot the time since which the vehicle covered L metres. The vehicle positions are stored in a buffer memory and dismissed as soon as it goes out of the sliding window. The track present in the buffer memory at any time is called Buffer Track.

The distance is chosen as trigger to fill the buffer instead of time for the following reasons. First, maximum distance can be easily selected with respect to the usual size of a roundabout while choosing a maximum time would depend on traffic conditions. Second, useless redundant points are avoided when the vehicle stops when using a distance-based buffer track.

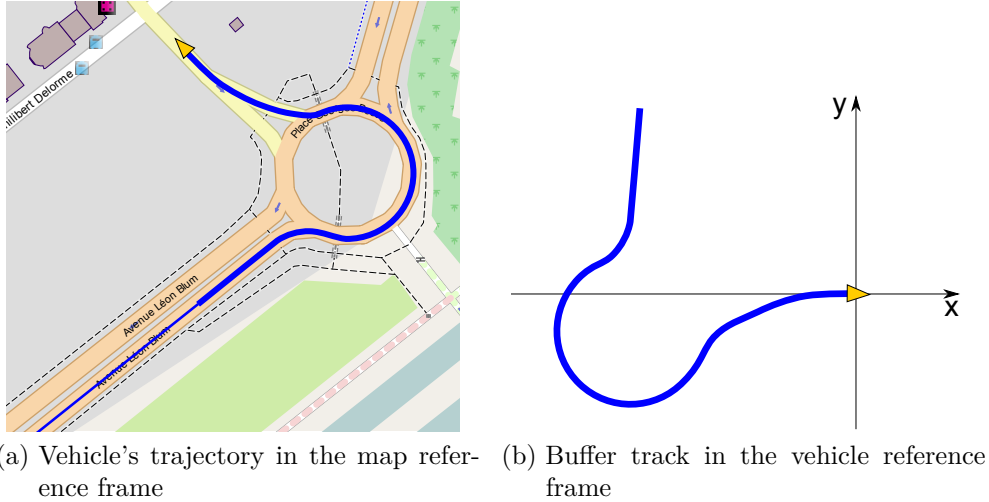


Figure 3.2: Pre-processing step. The vehicle is the yellow arrow. Its trajectory (thin blue line) is accumulated and only the L last metres are kept in the buffer track (thick blue line) and are expressed in the vehicle coordinate system.

Pre-processing consists also in changing the reference frame of the buffer track to reduce the diversity of the possible roundabout shapes. The shapes and roundabout orientations are then made independent. For this purpose, we propose to use an egocentric approach. The buffer track is transformed into the current vehicle frame using the rotation matrix Z defined in (3.1) where ψ denotes the vehicle's heading with respect to the East (horizontal axis).

$$Z_k = \begin{bmatrix} \cos(\psi_t) & -\sin(\psi_t) \\ \sin(\psi_t) & \cos(\psi_t) \end{bmatrix} \quad (3.1)$$

Figure 3.2 shows an example of the buffer track extraction and rotation in the pre-processing step.

3.3.2 Descriptor

In classification and recognition tasks, the choice of a descriptor is crucial. Here, the vehicle's trajectory is considered as a graphical 2D objects, the descriptor should then illustrate these two dimensions independently. Let the descriptor be composed of two halves: left and down profiles. The left (resp. down) vector stores the distances between the left (resp. down) side of the interval and the first point met on this line (resp. column) as shown in Figure 3.3. The profiles contain n equally

Algorithm 3.1 Down profile computation

Inputs: Buffer track coordinates in the vehicle reference frame $X_j, Y_j, j \in \{1, \dots, m\}$, number of profile samples n .

Outputs: Down profile $c_i, i \in \{1, \dots, n\}$.

Begin

$$\bar{X} \leftarrow \max_{j:1 \rightarrow m} (X_j)$$

$$\underline{X} \leftarrow \min_{j:1 \rightarrow m} (X_j)$$

For $i \leftarrow 1 : n$, **do**

$$x_i \leftarrow \underline{X} + (i - 1) \frac{\bar{X} - \underline{X}}{n - 1}$$

For $j \leftarrow 2 : m$, **do**

If $x_i \in [X_{j-1}, X_j]$, **do**

$$y_i \leftarrow Y_{j-1} + (Y_j - Y_{j-1}) \frac{x_i - X_{j-1}}{X_j - X_{j-1}}$$

$$c_i \leftarrow \min(c_i, y_i)$$

End if

End for

End for

$$\underline{c} \leftarrow \min_{i:1 \rightarrow n} (c_i)$$

For $i \leftarrow 1 : n$, $c_i \leftarrow c_i + \underline{c}$ **End for**

End

spaced elements: l_i for the left profile and c_i for the down profile. The computation of the profile is comparable to a linear interpolation of a function however, since one function input may have several outputs (e.g., c_2 in Figure 3.3), the interpolation algorithm must be slightly modified to keep the minimum value. Algorithm 3.1 details the computation of the down profile of the buffer track.

One issue when one deals with features such as roundabouts is the diversity in shapes of the elements named roundabouts. This comes, on the one hand from the number of exits of the roundabout and the one chosen by the driver, and on the other hand the roundabout radius. The first difficulty will be solved by defining a suitable number of classes, as developed later on. Secondly, to prevent scale effects, the descriptor is offset and normalized as shown by the two last lines of Algorithm 3.1.

The descriptor of one buffer track is therefore a vector D of $2n$ elements. An illustration is given in Figure 3.4.

$$D = [l_1 \quad l_2 \quad \cdots \quad l_n \quad c_1 \quad c_2 \quad \cdots \quad c_n] \quad (3.2)$$

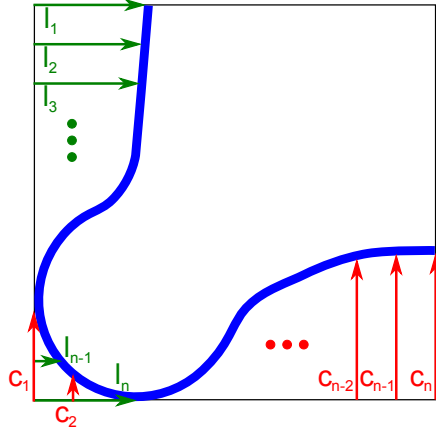


Figure 3.3: Buffer track with the profiles. The left profile is in green and the down profile is in red.

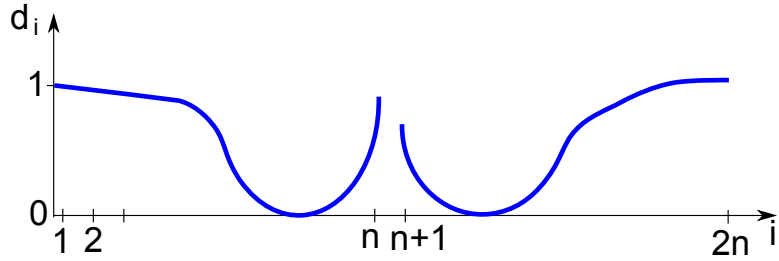


Figure 3.4: Descriptor vector that corresponds to the example given in Figure 3.3.

As soon as the vehicle goes straight, the projection of the buffer track along the Y axis becomes very small. In this situation, the vehicle cannot have crossed any roundabout. Therefore, buffer tracks that have a vertical range less than threshold ΔY_{min} are neglected.

3.3.3 Recognition

Let us consider a closed frame of discernment Ω and a set of classes $\{w_i\}$ ($i = 1 \rightarrow k$) being a partition of Ω . According to total probability law and to the Bayes' theorem, the conditional probability of a sample D to belong to the i^{th} class is given by (3.3).

In this equation, it is assumed that every class has the same prior probability.

$$P(w_i|D) = \frac{P(D|w_i)}{\sum_{j=1}^k P(D|w_j)} \quad (3.3)$$

We consider sample D as a vector containing $2n$ random variables. Let us assume that, within a given class w_i , samples are distributed around the representative sample D_i according to a Gaussian distribution. Thus, if D belongs to w_i then $D \sim \mathcal{N}(D_i, P) \forall i \in \{1, \dots, k\}$ where P is the covariance matrix.

The posterior probability of w_i to be the correct class associated to the sample D is then given by (3.4).

$$P(w_i|D) = \frac{\exp\left(-\frac{(D-D_i)^T \cdot P^{-1} \cdot (D-D_i)}{2}\right)}{\sum_{j=1}^k \exp\left(-\frac{(D-D_j)^T \cdot P^{-1} \cdot (D-D_j)}{2}\right)} \quad (3.4)$$

The sample D is then associated to the maximum posterior probability class. This method minimises error probability.

3.3.4 Classes

In order to keep this system as simple as possible a supervised learning process is chosen. Consistent classes must be defined to make the recognition process efficient. It is not suitable to create only two main classes: one containing every buffer track that contains a roundabout and the other with anything else. Indeed, a too wide variety of shapes inside the same class (due to the large diversity of roundabout shapes as said previously) makes them inconsistent and the learning and recognition steps very tough. The solution found here has been to split the roundabout class into sub-classes. Each one being associated to one kind of buffer track containing a roundabout. Likewise, several classes associated to non-roundabout buffer tracks have been designed. To comply with the Bayesian recognition process introduced previously, the design of the classes is made in order to be orthogonal and exhaustive. Any buffer track should therefore belong to one, and only one class.

The classes are manually chosen in order to reflect the variety of the possible trajectories that contain a roundabout. The roundabout detection method is run every time new sensor data are available. Consequently, the method is run several times on the same roundabout, until it gets out the buffer track. The classes are chosen to represent the roundabout at the beginning and at the end of the buffer track in

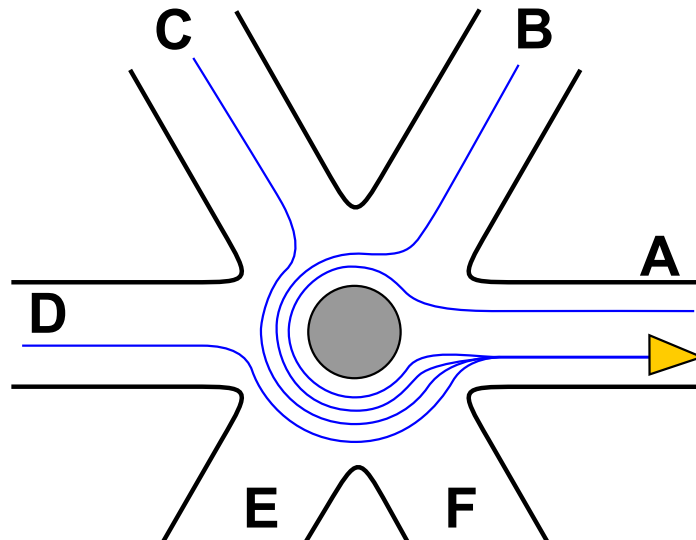


Figure 3.5: Roundabout intersection with associated vehicle trajectories. The vehicle is the yellow triangle and is going to the right.

order to increase the probability of detecting a roundabout before having completely passed it. It also permits to detect roundabouts with small diameter with respect to the buffer track length.

A generic roundabout intersection is used to highlight the choice made on the classes and is shown in Figure 3.5. A roundabout with 6 roads is chosen because the left profile descriptor is sensitive to the relative orientation of the entry and the exit of the roundabout. According to the previous requirement and the notation introduced in Figure 3.5, classes 1 and 2 represent buffer tracks for which the vehicle took the *A* entry (i.e. the vehicle made a U-turn at the roundabout). Similarly, classes 3 and 4 are for entry *B*, classes 5 and 6 for entry *C* and classes 7 and 8 are for entry *D*.

When crossing a roundabout by the entries *E* and *F* for instance, the drivers usually optimize the trajectory and do not strictly follow the circle. In addition, the presence of a roundabout has almost no effect on the trajectory. The roundabout is therefore not observable so these are not addressed by the method.

Similarly, the non-roundabout classes are selected to represent left and right turns with different curvature. Classes 11, 14 and 15 are for left turns and classes 12 and 13 depict right turns. Finally, classes 9 and 10 are for buffer tracks that are nearly straight but above the threshold.

Buffer tracks computed based on real vehicle data were finally manually selected in order to comply with these specifications. These are presented in Tables 3.1, 3.2 and 3.3 with their corresponding descriptors.

Table 3.1: Classes used for roundabout inference, buffer tracks and corresponding descriptors. Roundabout classes 1 to 5.

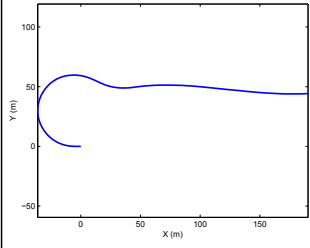
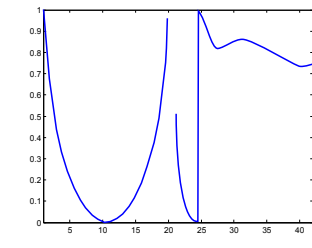
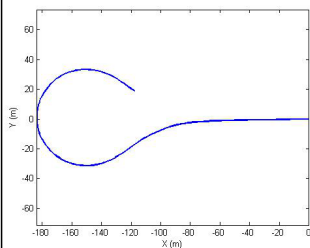
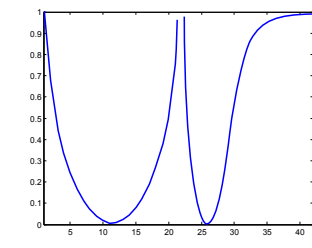
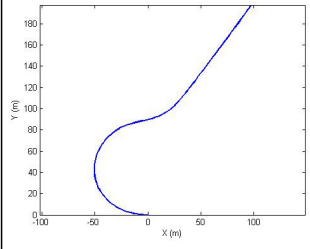
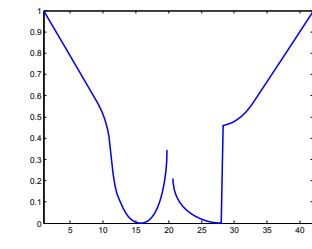
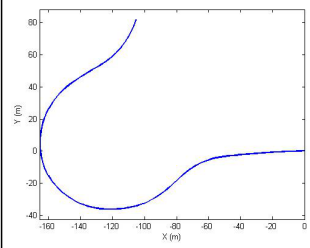
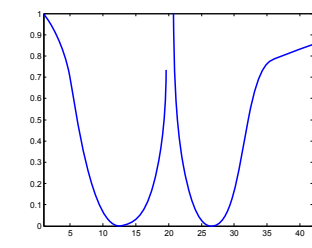
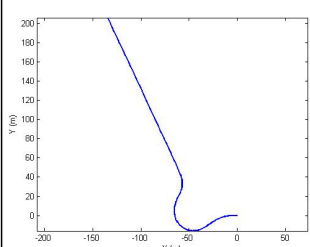
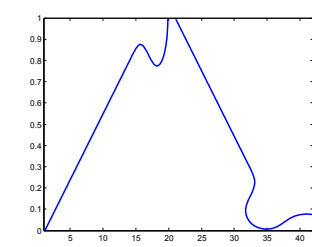
	Buffer Tracks	Descriptors
Class 1		
Class 2		
Class 3		
Class 4		
Class 5		

Table 3.2: Classes used for roundabout inference, buffer tracks and corresponding descriptors. Roundabout classes 6 to 8 and non-roundabout classes 9 and 10.

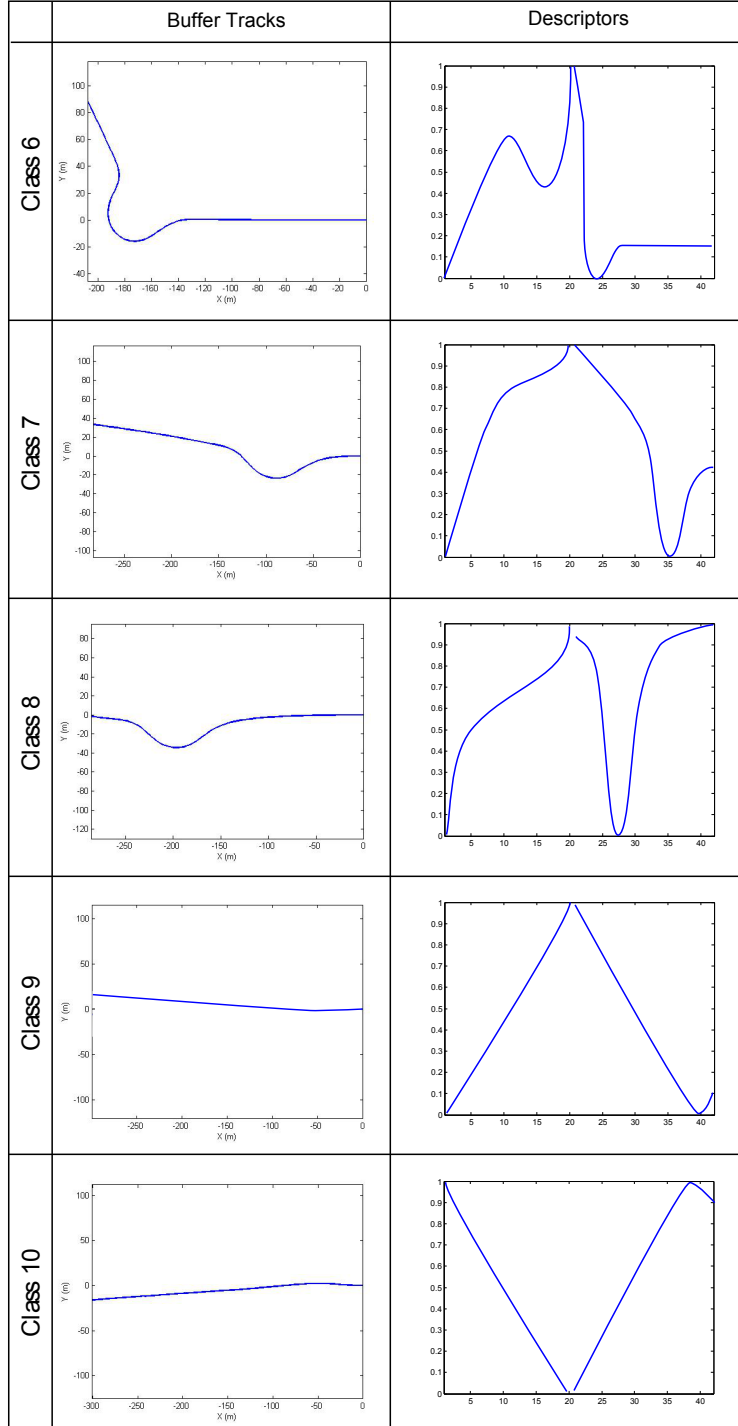
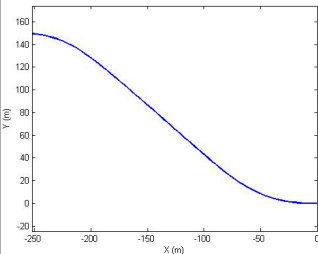
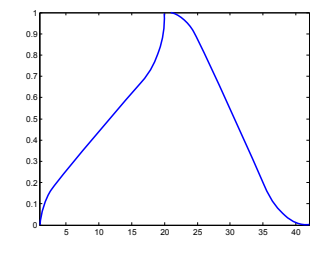
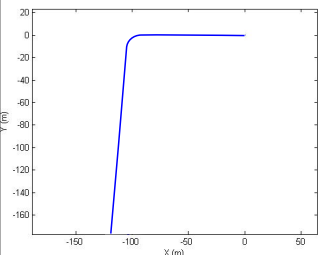
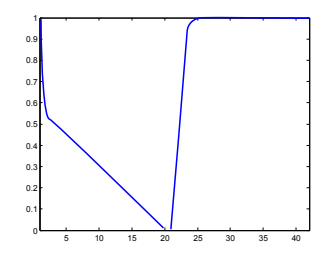
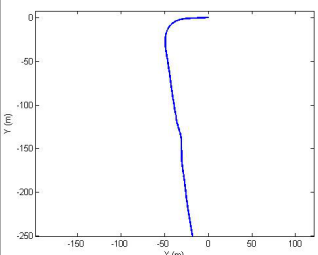
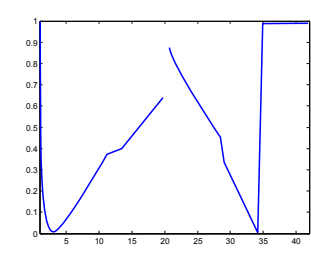
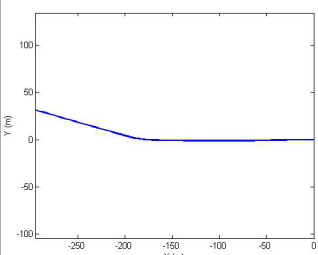
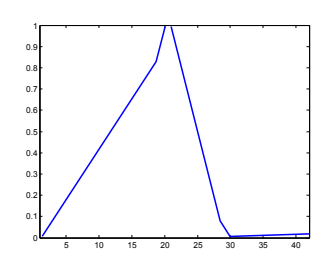
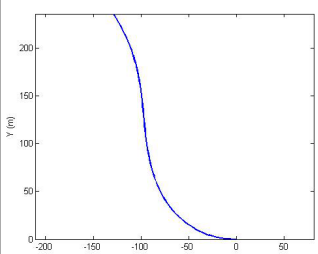
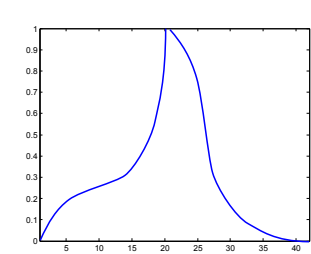


Table 3.3: Classes used for roundabout inference, buffer tracks and corresponding descriptors. Non-roundabout classes 10 to 15.

	Buffer Tracks	Descriptors
Class 11		
Class 12		
Class 13		
Class 14		
Class 15		

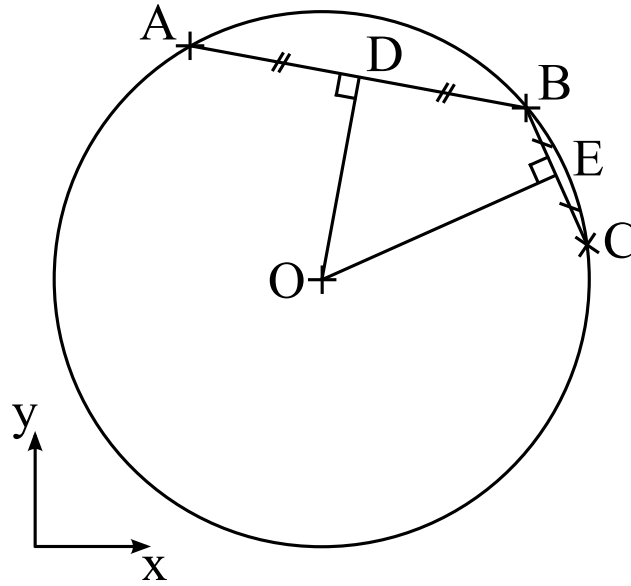


Figure 3.6: Circumscribed circle of three points A , B and C in (x, y) coordinates. D and E are the midpoints of $[AB]$ and $[BC]$ respectively. The centre is denoted by O .

The classes presented here are designed to right-hand drive and should be twisted around the horizontal axis for application in left-hand traffic countries.

3.3.5 Roundabout Placement

As soon as a roundabout has been detected in a buffer track, the final step is to find its centre and its radius. This allows to convert the discovery of a roundabout back into absolute coordinates and to identify it clearly into the internal memory. This step is based on the clustering of the successive rotation centres. It is easy to prove that the centre $O(x_O, y_O)$ of the circle passing by three points $A(x_A, y_A)$, $B(x_B, y_B)$ and $C(x_C, y_C)$ that are not aligned is unique as illustrated by Figure 3.6. The coordinates of the centre are found by solving the following system of equations:

$$\begin{cases} \vec{DO} \cdot \vec{AB} = 0 \\ \vec{EO} \cdot \vec{BC} = 0 \end{cases} \quad (3.5)$$

where D and E are the midpoints of chords $[AB]$ and $[BC]$ respectively. The system

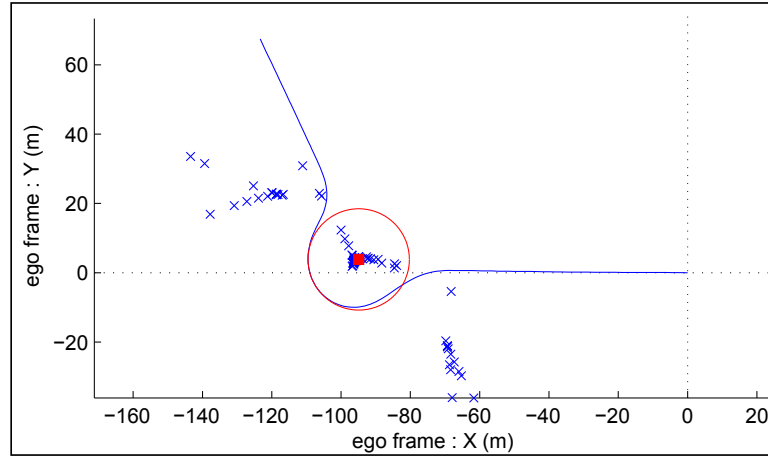


Figure 3.7: Determination of the roundabout centre and radius in a buffer track. Blue line represents the buffer track itself and blue crosses are its successive rotation centres. Red square denotes the centre retained associated to its circle (red line).

of equations (3.5) is therefore written as follows:

$$\begin{cases} (x_B - x_A)x_O + (y_B - y_A)y_O = \frac{1}{2}(x_B^2 - x_A^2 + y_B^2 - y_A^2) \\ (x_C - x_B)x_O + (y_C - y_B)y_O = \frac{1}{2}(x_C^2 - x_B^2 + y_C^2 - y_B^2) \end{cases} \quad (3.6)$$

The analytical solution is found using the matrix formulation.

The centre and radius of every points trio is computed along the whole buffer track as presented by blue crosses in Figure 3.7. Considering the fact that the curvature radius remains almost constant while the vehicle is in the roundabout, the true centre belongs to the densest cluster. For each rotation centre, the number of neighbours within a predefined distance λ_R perimeter is computed. The rotation centre that counts the highest number of neighbours is selected. The final roundabout centre is the barycentre of the neighbourhood of the selected rotation centre and is shown in Figure 3.7 by the red square. This method is a good compromise between simplicity and robustness.

The centre coordinates are converted back into absolute ENU using the reciprocals of (3.1) and then into geodetic coordinates.

Usually, the *roundabout* information is actually one of possible value of the attribute *road type* linked to a road segment (as *motorway*, *parking*, *tunnel*,...). If a roundabout has been detected for the first time by the vehicle and is missing in the navigation map, its location is added to the map and a “*don't use the map here*” flag is raised for ADAS. Then, next time the vehicle goes through this place, ADAS will be able to reject the map information.

3.4 Experimental Validation

3.4.1 Experimental Set-up

The method has been tested in real conditions thanks to a *Renault Espace* passenger vehicle driven in various conditions, urban, suburban and rural environments. The localization system on board was a standard single frequency Ublox 4T receiver. A Ixsea LandIns Inertial Navigation System (INS) tightly coupled to a Novatel GPS receiver has been used as position ground truth. The vehicle odometer, speed, rear wheel speed difference and yaw rate were available in the vehicle CAN-bus. More details on the experimental facilities are in Appendix A.

The few parameters of the algorithm have been tuned as follows. First the length L of the buffer track must be larger than the perimeter of the biggest roundabout the vehicle may cross. We have set $L = 250$ m which corresponds to a roundabout radius of 40 m which is also the value chosen for the maximum detectable radius R_{Max} in the filter described before. The neighbour radius chosen in the roundabout centre clustering λ_R is set to 3 m. Then, the minimum Y range defined in this filter is set so that $\Delta Y_{min} = 10$ m which corresponds to minimum detectable radius of 5 m for U-turns and 10 m for straight crossings.

3.4.2 Method Performance Evaluation

The most relevant way to assess the method is to use it on real data. A large amount of data representing more than 150 km have been recorded in various environments. Rural, suburban and urban areas have been covered totalling up 37 different roundabouts. Some of them have been crossed several times by different ways and under different traffic conditions.

Firstly, the performance of the localization sub-system has been assessed thanks to the ground truth provided by the INS. The maximum distance error is below 15 m and equals 5 m in average.

The consistency of the filter is highlighted in Figure 3.8. It shows that the true position (in green) and the measured position (in red) are always inside the 3-sigmas covariance ellipses (blue ellipses) which corresponds to a 99.7% probability. One can notice that when the vehicle is stopped (bottom side of the picture), these ellipses decrease due to a larger number of observations for the same position.

Secondly, the classifier performance has been deduced from more than 100 km of driving sequences. The classes assigned by the classifier have been manually compared to true classes. The results have been merged to provide the overall roundabout detection rates. The method is evaluated in terms of classification abilities. A True Positive occurs when the method classifies in one of the roundabout classes

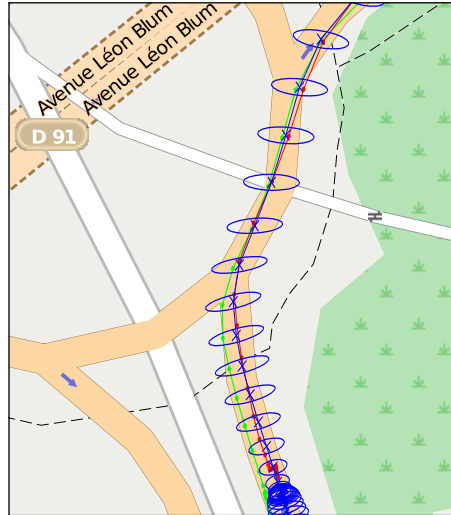


Figure 3.8: Detail of vehicle's path. The position estimations are associated to their covariance ellipses (blue line, crosses and ellipses). GPS fixes used in the filter are denoted in red and ground truth is in green.

(classes 1 to 8) a buffer track that actually contains the trajectory that corresponds to a roundabout crossing. A False Positive is when a buffer track with no roundabout crossing is classified in one of the roundabout classes. In order to make these figures more understandable, these are divided by the total number of positive detections, i.e. the number of buffer tracks that were classified in a roundabout class. The reciprocal is employed with True Negatives and False Negatives that measure the ability of the method to detect non-roundabout buffer tracks.

The computation of these performance metrics was done partly manually. For each dataset, a ground truth of the places where the vehicle actually crossed a roundabout was manually built. The time stamps, in the dataset, that correspond to observable roundabouts (as described in Section 3.3.4) were found and the corresponding buffer tracks were tagged. The statistical results were then automatically generated by comparing, for each time stamp, the output of the classification method with the ground truth.

The results are summarized in Table 3.4. One can notice that every buffer track classified as roundabout actually contains a roundabout, i.e. there are no false positive. This is an important result because the method is expected to not detect roundabout with no reason. Moreover, data used here contains ambiguous manoeuvres like U-turn and left turns with no roundabout. The false negatives show that some buffer tracks with roundabouts are missed by the classifier. Some of these are roundabouts which were crossed nearly straight which makes them almost not observable. Others correspond to observable roundabouts missed by the method. However, as stated previously, the evaluation of a buffer track is independent from one time step to the other. A roundabout miss-detected when it was at the beginning of the track may be detected a few time steps later. This justifies the uses of two

Table 3.4: Method performance

True positives: 100%	False negatives: 7.2%
False positives: 0%	True negatives: 92.8%

classes for one roundabout shape as denoted in Section 3.3.4. On these experiments every observable roundabout was detected

Figure 3.9 represents a part of the outputs of the algorithm. One can notice that the roundabouts have been correctly detected and their centres located on the map. The large roundabout at the intersection of avenues *de l'Europe* and *Leon Blum* is not addressed by the method because the vehicle took the first exits. A direct application of the algorithm is shown in Figure 3.10. The vehicle has been driven on a recently modified road. On road D926, a new roundabout has been built and another one is still under construction. We can see the roundabout centres (red stars) detected by the algorithm while the vehicle was driven along the blue path. It can also be noticed on this picture that the U-turn is successfully classified as a non-roundabout. The algorithm makes then possible to detected that the map is obsolete and to store this information for next passages in the internal memory. We see in this situation that a cruise control system must be henceforth turned off on road D924.

The background map used in Figure 3.9 and Figure 3.10 to illustrate the results is taken from OpenStreetMap project [61, 2].

The method has been coded and tested in Matlab and runs more than 10 times faster than the real data. Then, even if it has not been implemented in real time yet, it makes conceivable an embedded application.

3.5 Conclusion

A method for the identification of missing roundabouts based on the trajectory estimated by the vehicle has been presented. This particular type of structural fault deserves to be addressed for two main reasons. First, due to their positive effect on road fatalities a large number of roundabouts is built, especially in European countries. Second, a roundabout intersection modifies significantly the driving rules and the approaching manoeuvres. To be correctly anticipated by the driver or by automated driving function (speed reduction, lane placement,...) the roundabout must be correctly anticipated.

Within the framework of autonomous monitoring of navigation map integrity taken in this thesis and according to the black box assumption on the navigation system,

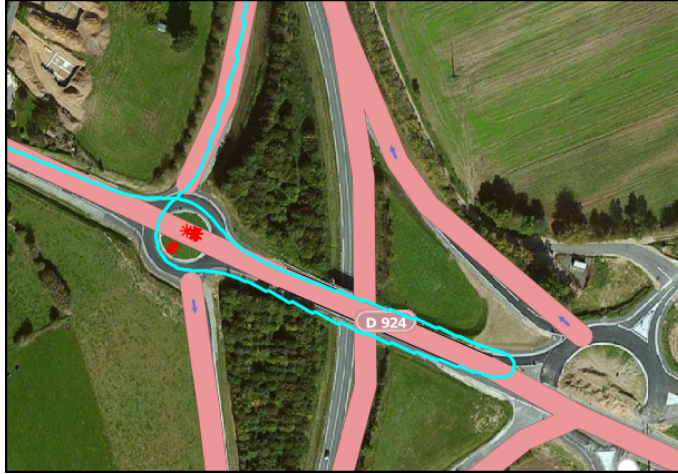


Figure 3.10: Example of application of the algorithm in rural environment. Here the road network has changed recently, one roundabout has been created and another one is being built. These are not registered in the navigation map (red lines) yet. The vehicle track is in blue and the centres of the roundabouts found by our algorithm are in red stars.

the idea developed in this chapter was to determine when the vehicle crossed a roundabout based on the shape of its trajectory. This trajectory corresponding to the last 250 m travelled by the vehicle is computed using the standard on board vehicle sensors (odometer, speeds, yaw rate and GNSS). The trajectory of the vehicle when crossing a roundabout has a typical shape, a probabilistic classifier based on the left and down trajectory profiles therefore implemented. Fifteen classes are manually learnt in order to represent the diversity of vehicle trajectories with and without a roundabout. The roundabout centre and radius is finally found by the clustering of rotation centre of the trajectory. The presence of a roundabout not represented in the navigation map is stored in the method memory in order to warn client systems when the vehicle comes again on the same location.

The method finally has shown good performance during the test on real vehicle data. Every roundabout crossed by the vehicle was appropriately detected and the centres were localized with a sufficient accuracy. Moreover, no roundabout was detected inappropriately (no false positive). We have also noticed that the classifier is not very sensitive to the tuning parameters. For instance, the thresholds can be chosen within a certain margin and the covariance matrix P has not a strong effect.

The method has been tested to detect new roundabouts in rural areas where the road network was recently modified, but it can be significantly improved. Even if the Matlab prototype of this method shows computation times that are compatible with real time applications, it could be accelerated by using the log-likelihood in the recognition process. The design of classes is the main topic that could be addressed for improvement. A comprehensive study of the contribution of each class to the true, false positive and to the true, false negatives would allow to evaluate their

individual relevance. In association with an evaluation of the correlation between the classes, it could permit to improve the design of the classes. Moreover, the situation in which the vehicle makes one or several full rotation(s) around the roundabout was not directly addressed. According to the classes defined in this chapter, the roundabout is likely to be detected when the vehicle made the three-quarters of the first rotation. The buffer track would indeed be similar to the third class. After several rotation the buffer track would be a complete circle; a dedicated class should be designed to address this situation properly.

Chapter 4

Map Geometric Faults Detection, Isolation and Adaptation

Contents

4.1	Introduction	65
4.2	Problem Statement	66
4.3	Fault Detection, Isolation and Adaptation Method . . .	73
4.4	Formalism Properties	79
4.5	Illustrative Examples	82
4.6	Practical Implementation	85
4.7	Experimental Evaluation	89
4.8	Conclusion	102

4.1 Introduction

The quality of the road representation within the navigation map in terms of geometry has a direct impact on the performance of intelligent vehicles navigation systems, as introduced in Chapter 1. The knowledge of the road shape in front of the vehicle is used in existing intelligent vehicles to improve sensor tracking (e.g., lane markings for lane keeping functions or leading vehicle for adaptive cruise control applications) and anticipate hazardous situation by adapting the vehicle speed. The navigation map road geometric description is also essential for highly automated driving like for path planning, decision making and control functions.

To avoid dysfunction of these systems, the quality of the geometric description of the road in the navigation map must be monitored. In a standard passenger vehicle, a method is to compare the map with the positioning information given by the available sensors. Since this provides only a one degree of freedom in terms of redundancy, the challenge relies on solving the ambiguity on the fault source in order to get an indication of the integrity of the navigation system. Nevertheless,

if the vehicle travels several times on the same roads, it is possible to increase the redundancy and therefore to isolate (i.e. identify with no ambiguity) the faults. A way to perform this task is to store from one trip to another the estimates of the vehicle position from the navigation system and from vehicle sensors. Thanks to this information, it is possible to implement a Faults Detection, Isolation and Adaptation (FDIA) strategy [37, 74, 14]. The knowledge on the methods and systems involved in vehicle navigation functions allows to establish a set of assumptions on which a FDIA framework can be based.

In this chapter, the theory involved in the proposed formalism and the resulting properties are introduced and proved in a generic and mathematical manner. The interest of this formalism is illustrated by two detailed examples. Quantitative measures of the method performance obtained with real vehicle data are finally presented and discussed for rural and urban scenarios.

4.2 Problem Statement

4.2.1 Monitoring System

According to the black box assumption on the vehicle systems, the only available observation of the road geometry is the data in the EH.

The geometry of the road is described through two physical quantities, namely the position of the shape nodes and the curvature. Even if those are encoded statically in the navigation map, the road curvature is calculated based on the shape nodes by the map provider and recorded in the database. A discrepancy may occur when the curvature is calculated based on the shape nodes described in the navigation map in the vehicle because the navigation map is often simplified to reduce its size. Even if the road curvature appears to be a better description of the road shape, the road shape node geographical coordinates are chosen here as a description of the road geometry for two reasons. First, the curvature attribute is presently not defined for the entire road network in the commercial navigation maps used in the vehicles. Minor roads outside cities do not have this attribute. Second, if a lateral offset is detected on a road, one can easily deduce that it will lead to a longitudinal error for all perpendicular connected roads. Dysfunctions of intersection warning can therefore be prevented.

The systemic diagram of integrity monitoring in passenger vehicle introduced in Figure 1.6 is extended by the one of Figure 4.1. This is the basis of the developments done in this chapter. The aim is to provide an indication of the integrity of the navigation system (particularly in case of map road geometry faults) to the systems that use this information (i.e. client systems). If a loss of integrity is detected, the correction to make can be provided to the client systems.

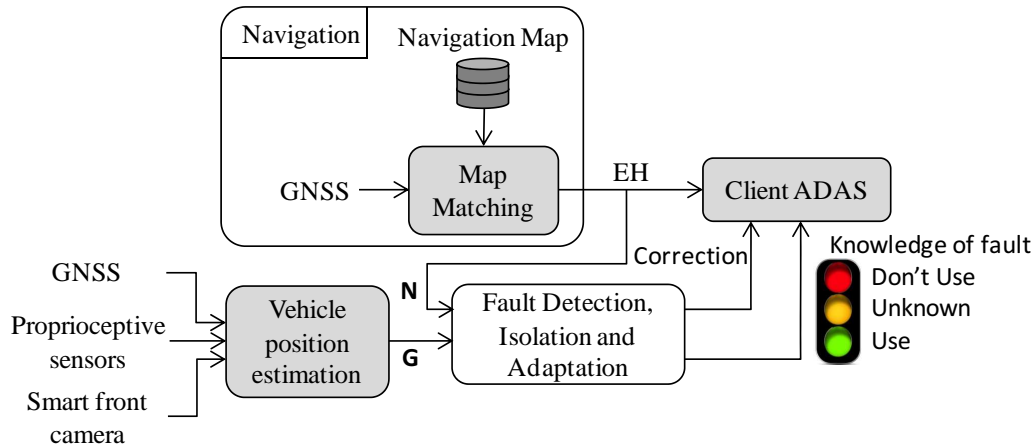


Figure 4.1: Structure of fault detection isolation and adaptation to standard passenger vehicle.

To do so, an estimation of the vehicle position independent of the navigation system is used. A GNSS receiver is needed to compute this estimate and other sensors as proprioceptive sensors (odometer, gyroscope,...) can be employed to improve its precision, accuracy and availability. The vehicle position estimation function of Figure 4.1 provides this estimate G which is called the position estimate from sensors. The internal calculations of this function was developed in Chapter 2. In contrast, the estimation of the vehicle position made by the navigation function is the observation of the navigation map geometry since it results of the matching of GNSS navigation receiver onto the navigation map. As it is seen as a black box, it is only available from the EH and is named position estimate from navigation and denoted by N . This estimate is compared with estimate G to determine whether it is affected by a fault (i.e. faulty). Since both estimates may be affected by faults, the formalism introduced in this chapter takes benefit from the repetitive vehicle trips in order to solve this ambiguity.

The output *Knowledge of fault* of Figure 4.1 has three possible values:

- *Use*. The estimate provided by the navigation function to client systems is not affected by a fault.
- *Unknown*. A fault has been detected but is not isolated. The position estimate from the navigation system is possibly affected by a fault.
- *Don't use*. A fault affects the current estimate from the navigation system and the method provides a fault-free estimate to client systems through the output *Correction*.

As a recall, the fault detection step consists in stating that at least one of the evaluated estimate is affected by a fault. The isolation step consists in determining which estimate(s) is (are) affected by a fault.

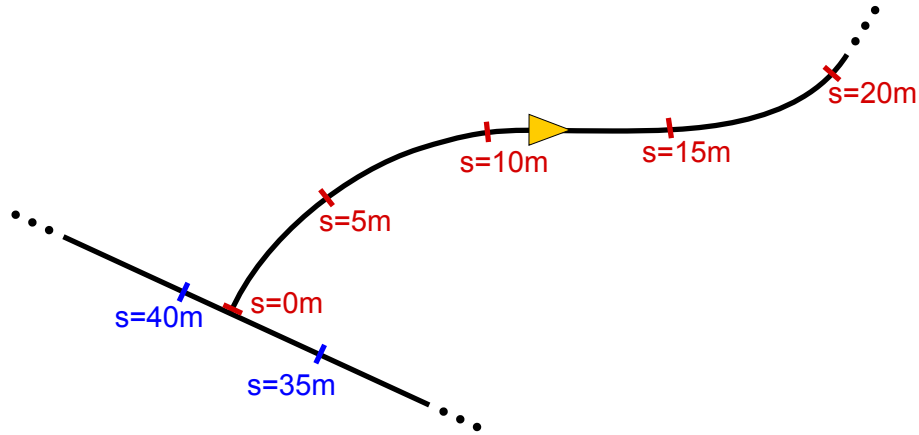


Figure 4.2: Roads spatially sampled with a 5 metres interval

The vehicle position estimate provided by the vehicle navigation function is evaluated thanks to an additional estimation calculated with vehicle sensors. However, this estimation is also possibly affected by a fault. If the two estimations differ, there is an ambiguity on the identification of the faulty estimate. This ambiguity cannot be resolved due to the low level of redundancy (the degree of freedom being only one). The main idea that is studied in this framework is to develop a method that takes advantage of the repetitive vehicle trips to resolve this ambiguity.

4.2.2 Spatial Sampling

In the proposed approach, integrity of the vehicle position estimate from the navigation system is spatially evaluated. Each location of the road network is considered as an operating point of the system to monitor (i.e. navigation system). For a given location of the vehicle, the presence of a fault is investigated using all the estimates recorded at this location during the previous vehicle trips.

The method is spatially sampled with respect to the curvilinear abscissa of the road. The vehicle curvilinear abscissa on a given road is the distance along the carriageway with respect to its origin and is written $s \in \mathbb{R}^+$ as shown by Figure 4.2.

Let $K \in \mathbb{N}$ denote the total number of times the vehicle went on a given road. The true vehicle position at abscissa s of a given road and at the k^{th} trip is written P_k^s . This can be encoded as a vector that contains the vehicle's geographic coordinates longitude, latitude and ellipsoidal height.

Using the same notation convention, G_k^s and N_k^s are estimates of the vehicle position P_k^s provided by the sensors and the navigation respectively. Every time the vehicle is at abscissa s of a given road and for the k^{th} time, these two estimates are recorded. Figure 4.3 shows some estimates from sensors recorded during three trips.

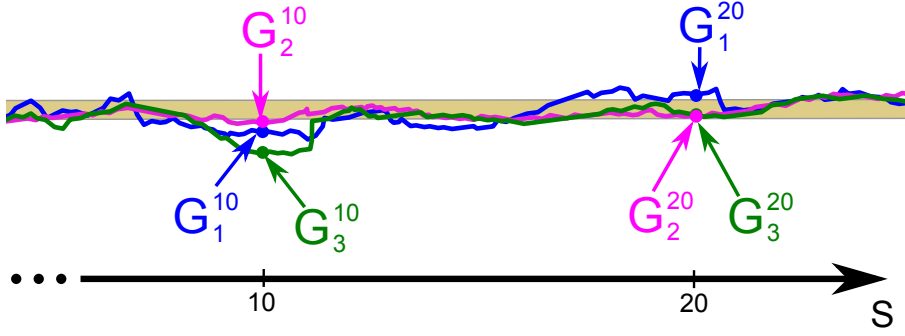


Figure 4.3: Vehicle position estimates from sensors for three vehicle trips on the same road. Samples G used by the method are represented for two abscissa with 10 m interval.

Faults may affect the navigation as well as the position estimate from sensors and cause their value to be significantly different from the ground truth (if a multipath affects a GNSS receiver for example). In this case, the estimates are said to be faulty.

Let us define the faults $f_{N_k^s}$ and $f_{G_k^s}$ with:

$$f_{N_k^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } N_k^s \neq P_k^s \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

$$f_{G_k^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } G_k^s \neq P_k^s \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

Figure 4.4 illustrates the notation convention. Since physical quantities cannot be strictly equal, a threshold on the distance between the estimates is employed for real implementation which will be detailed in Section 4.6. Chapter 5 will develop a statistic approach that improves this approximation.

Since the true vehicle position is not measurable directly, the fault detection and isolation relies on the pairwise comparison of estimates.

4.2.3 Assumptions

A fault is an error generative process according to the definition stated in Section 1.2.1. An error is so the discrepancy caused by the fault and is measurable by an appropriate external observer. A faulty navigation system (from the client ADAS

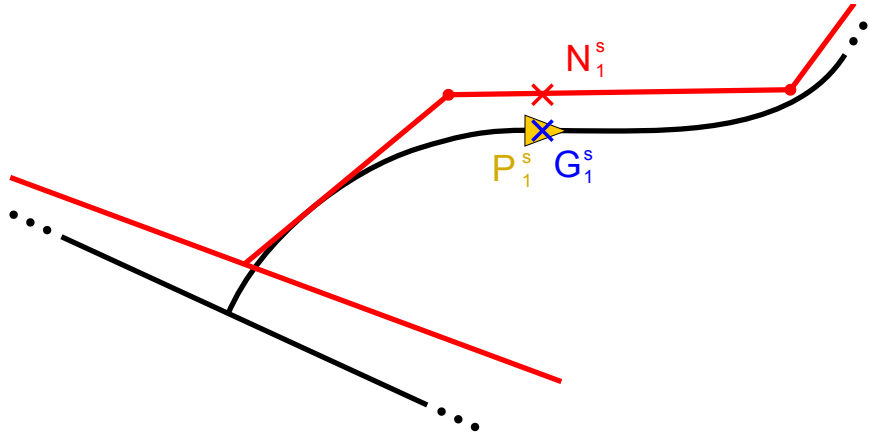


Figure 4.4: Illustration of the notation convention. The true (i.e. real) road is denoted by the black splines while the map is in red. The yellow arrow represents the true vehicle pose. Estimation of the vehicle position by the navigation function is the red cross. The blue cross shows the estimation from vehicle sensors. The example taken in this figure leads to $f_{N_1^s} = 1$ and $f_{G_1^s} = 0$ according to (4.1) and (4.2).

point of view) can result from a navigation map fault. Map faults cause systematic errors. Indeed, as shown in Figure 4.5, every time the vehicle goes on this area, the navigation system provides the same faulty position estimate.

A faulty state of the navigation system is not only due to map faults. Indeed, the map-matching may choose a wrong road candidate due to difficult road configuration like in junction for instance as shown by Figure 4.6. In this situation the map-matching may or may not choose the right road segment from one trip to another; faults are not systematic. Nevertheless, one can notice that, in case of a fault, the resulting error is always the same since the output of the navigation estimate is constrained by the map. Errors on the navigation system, when they occur, are systematic.

In this chapter, the problem of determining the reason of an error on the estimation from the navigation is not addressed because this would require to have access to the internal variables of this system.

The estimation of the vehicle position from the on board vehicle sensors G mainly rely on GNSS information even if it is merged with DR sensors. A fault on a G estimate can therefore result from two main reasons which are closely related to the receiver environmental conditions. Multipath (i.e. satellite signal reflection on buildings for instance) and poor satellites configuration can induce large position errors which are considered as faults in our framework. The question is now to see if errors and faults on G are correlated from one trip to another. Studies on this topic are often addressing static receivers. For instance, the magnitude of the

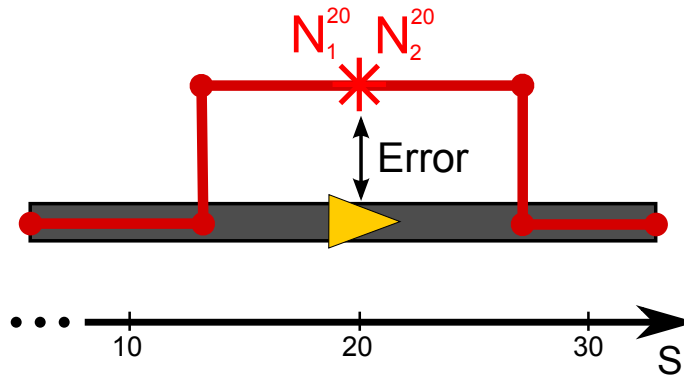


Figure 4.5: Systematic errors due to navigation map fault (the map is displayed in red). The faulty estimates of the vehicle position provided by the navigation function at abscissa $s = 20$ are the same.

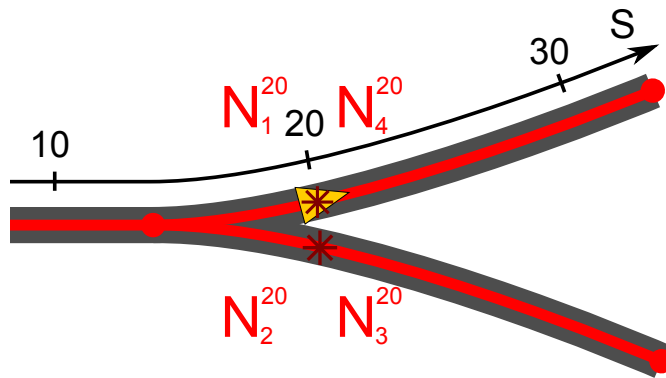


Figure 4.6: Non-systematic faults of navigation function. The map-matching chooses either the right or the wrong road from one trip to another. The navigation estimate may or may not be faulty. Errors on the faulty estimates are nevertheless equal (here N_2^{20} and N_3^{20}).

day-to-day multipath correlation of a base station is typically around 85% [99]. We have no knowledge of similar results for a moving receiver since the receiver motion mitigates the multipath effects. Moreover, a GNSS multipath error can repeat itself only at the same place with the same satellites configuration (the ground track repeat is 23hr 56min for GPS and 10 days for Galileo). Therefore, we believe that the repetition of the same multipath errors and faults is very unlikely from one trip to another. Errors and faults due to poor satellites geometry (multipath aside) result from the propagation of pseudo-range random measurement errors (Dilution Of Precision DOP) and have therefore also a weak correlation between two vehicle trips. Finally, faults of position estimate from sensors are not systematic. When a G fault occurs, it causes errors which are assumed to be different from one trip to another.

The monitoring system relies on the observation of the difference between G and N , each being affected by errors (with respect to the unknown ground truth). Since two independent GNSS receivers and position computation algorithms are used (as shown in Figure 4.1), it is unlikely that the estimate from sensors G falls on a wrong road that exactly corresponds to an incorrect map-matched solution N . Therefore, errors due to faults on G are assumed to be different to errors due to faults on N .

The assumptions on which is based the FDIA framework can be summarized as follows:

- When travelling several times on a road, the vehicle follows the same path with small deviations (which can be compensated if necessary by the lane markings measurements of a front camera).
- Every fault on position estimates from sensors can cause errors of any values. In other words, errors on faulty vehicle position estimates from sensors are different from one trip to the other, at a given abscissa.
- The navigation map does not change from one trip to another. Errors on the vehicle position estimates from the navigation (when they occur due to a fault) are therefore always the same, at a given abscissa.
- Errors on the vehicle position estimates from sensors and from the navigation are different from each other, at a given abscissa.

According to these assumptions, one can state that:

$$P_i^s = P_{i+1}^s, \forall i \in \{1, \dots, K - 1\} \quad (4.3)$$

In the following, the formalism is developed from the system monitoring point of view, putting temporarily aside the application to intelligent vehicles. The estimates from sensors and from the navigation (G and N respectively) are seen as estimates of the same physical quantity P which comply with the assumptions cited before. The curvilinear abscissa denoted by s is understood as an operating point of the

system to monitor. The multiple vehicle trips are iterations of this system and is denoted by K .

4.3 Fault Detection, Isolation and Adaptation Method

This section shows how to detect and isolate faulty estimates and perform adaptation according to the assumptions made previously. The adaptation process consists in providing a non-faulty estimate to a client system for it to operate normally if a fault affects the current estimate. This therefore requires to identify unambiguously a not faulty estimate. The concepts of Sets of Faults and Residuals are defined first. The mathematical relationship between these two concepts is then demonstrated. It is finally shown how to isolate a set of faults based on the observation of residuals.

4.3.1 Sets of Faults

Let the set of faults e_K^s be composed of all $f_{G_k^s}$ and $f_{N_k^s}$ for the considered iterations K at a given abscissa s :

$$e_K^s \stackrel{\text{def}}{=} \{f_{G_i^s}, f_{N_j^s}\}, \forall i, j \in \{1, \dots, K\} \quad (4.4)$$

The cardinality of e_K^s is $2K$. Each element of e_K^s is a boolean value so there are 2^{2K} possible sets written $e_{K,n}^s$:

$$\left\{ \begin{array}{l} e_K^s \in \{e_{K,n}^s\} \\ e_{K,n}^s \in \mathbb{B}^{2K}, \forall n \in \{1, \dots, 2^{2K}\} \end{array} \right\} \quad (4.5)$$

Let us take an example with $K = 2$. There are $2^{2 \cdot 2} = 16$ different sets. The cardinality of each one is $2 \cdot 2 = 4$. For instance, $e_{2,5}^s = \{0 \ 0 \ 1 \ 0\}$ means $\{f_{G_2^s} = 0 \text{ and } f_{N_2^s} = 0 \text{ and } f_{G_1^s} = 1 \text{ and } f_{N_1^s} = 0\}$.

4.3.2 Residual Processing

At a given abscissa s and at system iteration K , every available estimate at the current iteration is compared to all the others and the result is stored in a residual

vector R_K^s . The elements of R_K^s are defined as:

$$r_{G_i^s G_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } G_i^s \neq G_j^s \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (4.6)$$

$$r_{G_i^s N_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } G_i^s \neq N_j^s \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\} \quad (4.7)$$

$$r_{N_i^s N_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } N_i^s \neq N_j^s \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (4.8)$$

Equations (4.6) and (4.8) are restricted to $i > j$ to avoid useless redundant residuals. R_K^s is therefore made of $C(2K, 2)$ boolean elements where $C(2K, 2)$ stands for the number of 2-combinations from a given set of $2K$ elements.

$$C(2K, 2) = \frac{(2K)!}{2! \cdot (2K - 2)!} \quad (4.9)$$

where ! denotes the factorial operation. This can be written as

$$C(2K, 2) = \frac{2K(2K - 1)(2K - 2)!}{2(2K - 2)!} \quad (4.10)$$

so

$$C(2K, 2) = K(2K - 1) \quad (4.11)$$

The residual vector therefore contains $K(2K - 1)$ elements.

For example, at the second iteration, the size of R_2^s is 6:

$$R_2^s = \left[r_{N_2^s G_2^s} \quad r_{G_2^s G_1^s} \quad r_{N_1^s G_2^s} \quad r_{N_2^s G_1^s} \quad r_{N_1^s N_2^s} \quad r_{N_1^s G_1^s} \right] \quad (4.12)$$

If, for example, the estimates are such as $G_1^s \neq N_1^s = G_2^s = N_2^s$ then the residual vector is $R_2^s = \left[0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \right]$.

4.3.3 Relationships Between Faults and Residuals

Let \vee and \oplus denote boolean *or* and *exclusive or* operators respectively.

Proposition 1. *The elements of the residual vector are the result of boolean operations between the faulty states of the estimates according to the following equations:*

$$r_{G_i^s G_j^s} = f_{G_i^s} \vee f_{G_j^s}, \forall i, j \in \{1, \dots, K\}, i > j \quad (4.13)$$

$$r_{G_i^s N_j^s} = f_{G_i^s} \vee f_{N_j^s}, \forall i, j \in \{1, \dots, K\} \quad (4.14)$$

$$r_{N_i^s N_j^s} = f_{N_i^s} \oplus f_{N_j^s}, \forall i, j \in \{1, \dots, K\}, i > j \quad (4.15)$$

Proof. Equation (4.13) is demonstrated first. Let i and j be such as $i, j \in \{1, \dots, K\}$ and $i > j$. First of all, one can consider the case in which no fault affects the estimates. According to (4.2):

$$f_{G_i^s} = 0 \iff G_i^s = P^s \quad (4.16)$$

$$f_{G_j^s} = 0 \iff G_j^s = P^s \quad (4.17)$$

Then:

$$G_i^s = G_j^s \quad (4.18)$$

Secondly, the same reasoning is used if a fault affects one of the estimates: $f_{G_i^s} = 1$ and $f_{G_j^s} = 0$

$$f_{G_i^s} = 1 \text{ and } f_{G_j^s} = 0 \iff G_i^s \neq G_j^s \quad (4.19)$$

Finally, if both estimates are faulty:

$$f_{G_i^s} = 1 \iff G_i^s \neq P^s \quad (4.20)$$

and

$$f_{G_j^s} = 1 \iff G_j^s \neq P^s \quad (4.21)$$

Due to the assumptions on G errors, two faulty G cannot be equal. Then:

$$G_i^s \neq G_j^s \quad (4.22)$$

One can conclude that if there is at least one fault on G_i^s or G_j^s , then $r_{G_i^s G_j^s}$ is equal to one. This proves (4.13).

Secondly, the same reasoning scheme is used to demonstrate (4.14).

A similar deduction is finally applied to justify (4.15). Equation (4.1) allows stating that:

$$f_{N_i^s} = 0 \text{ and } f_{N_j^s} = 0 \iff N_i^s = N_j^s \quad (4.23)$$

Similarly, if a fault affects a navigation estimates:

$$f_{N_i^s} = 1 \text{ and } f_{N_j^s} = 0 \iff N_i^s \neq N_j^s \quad (4.24)$$

Contrarily to the previous cases and due to the assumption made in Section 4.2, two faulty N are equal, then:

$$f_{N_i^s} = 1 \text{ and } f_{N_j^s} = 1 \iff N_i^s = N_j^s \quad (4.25)$$

Finally, $r_{N_i^s N_j^s}$ is equal to one if there is only one fault among $f_{N_i^s}$ and $f_{N_j^s}$. This proves (4.15). \square

Equations (4.13), (4.14) and (4.15) of Proposition 1 establish a link between available estimates (i.e. G and N) and the faults which affected them (i.e. f_G and f_N). The first two equations tell that if there is at least one fault on the considered estimates then the residual will be affected. In (4.15), the residual equals one if there is a unique fault among both estimates. It is now possible to deduce the presence of faults based on observation and comparison of the estimates.

4.3.4 Fault Detection and Isolation

The fault detection and isolation strategy relies on listing all the possible sets of faults for a given iteration K and calculating the corresponding theoretical residual

Table 4.1: Truth table for one iteration ($K = 1$). The first residual $r_{G_1 N_1} = 0$ is only once in the table so it is in green since it makes isolation possible. Oppositely, $r_{G_1 N_1} = 1$ is due to more than one set on faults and is coloured in red. The residual used as explanation example is in bold.

Sets of faults $e_{K,n}$			Residuals
	f_{G_1}	f_{N_1}	$r_{G_1 N_1} = f_{G_1} \vee f_{N_1}$
$e_{1,1}$	0	0	0
$e_{1,2}$	1	0	1
$e_{1,3}$	0	1	1
$e_{1,4}$	1	1	1

vectors with (4.13), (4.14) and (4.15). This forms the truth table for K . In parallel, available estimates are used to compute the observed residual vector based on (4.6), (4.7) and (4.8). This vector, found in the truth table, allows determining the corresponding set of faults. Faults affecting each estimate can be finally deduced from this set. It can be noticed that the truth tables are valid for every operating point so the superscript s is omitted in the tables.

Let us take the example of Section 4.3.2. At the first system iteration at operating point s , two estimates are available: G_1^s and N_1^s . The truth table for one iteration can be found in Table 4.1. It is assumed in this example that $G_1^s \neq N_1^s$ is observed, therefore $r_{G_1^s N_1^s} = 1$ according to (4.6). Table 4.1 shows that this residual can be due to three sets of faults: $e_{1,2}^s$, $e_{1,3}^s$ and $e_{1,4}^s$. After one system iteration, it can be concluded that there is at least one faulty estimate among G_1^s and N_1^s but it is not possible to determine which one. Fault is detected but not isolated.

At the second system iteration at the operating point s , a new pair of estimates is available: G_2^s and N_2^s . The truth table for two system iterations is calculated with equations (4.13), (4.14) and (4.15) and is shown in Table 4.2. In this example and similarly to Section 4.3.2, it is assumed that $G_1^s \neq N_1^s = G_2^s = N_2^s$ is observed. This leads to the residual $R_2^s = [0 \ 1 \ 0 \ 1 \ 0 \ 1]$. Table 4.2 shows that this residual (in bold) is exclusively due to the set of faults $e_{2,5}^s$. After the second system iteration, fault isolation is performed by concluding that $\{f_{G_2^s} = 0$ and $f_{N_2^s} = 0$ and $f_{G_1^s} = 1$ and $f_{N_1^s} = 0\}$.

4.3.5 Conditions of Isolability

By definition, the truth table is exhaustive; the observed residual vector does make part of it. However, some sets of faults induce the same residual vector as denoted by the red colour in Tables 4.1 and 4.2. In this case, isolation is not possible. These are

Table 4.2: Truth table for two iterations ($K = 2$). Unique residuals are in green since they make isolation possible. Oppositely, residuals that are due to more than one set on faulty states are in red. The residual used as explanation examples is in bold.

	Sets of faults $e_{K,n}$				Residuals					
	f_{G_2}	f_{N_2}	f_{G_1}	f_{N_1}	$r_{N_2G_2}$	$r_{G_2G_1}$	$r_{N_1G_2}$	$r_{N_2G_1}$	$r_{N_1N_2}$	$r_{N_1G_1}$
$e_{2,1}$	0	0	0	0	0	0	0	0	0	0
$e_{2,2}$	1	0	0	0	1	1	1	0	0	0
$e_{2,3}$	0	1	0	0	1	0	0	1	1	0
$e_{2,4}$	1	1	0	0	1	1	1	1	1	0
$e_{2,5}$	0	0	1	0	0	1	0	1	0	1
$e_{2,6}$	1	0	1	0	1	1	1	1	0	1
$e_{2,7}$	0	1	1	0	1	1	0	1	1	1
$e_{2,8}$	1	1	1	0	1	1	1	1	1	1
$e_{2,9}$	0	0	0	1	0	0	1	0	1	1
$e_{2,10}$	1	0	0	1	1	1	1	0	1	1
$e_{2,11}$	0	1	0	1	1	0	1	1	0	1
$e_{2,12}$	1	1	0	1	1	1	1	1	0	1
$e_{2,13}$	0	0	1	1	0	1	1	1	1	1
$e_{2,14}$	1	0	1	1	1	1	1	1	1	1
$e_{2,15}$	0	1	1	1	1	1	1	1	0	1
$e_{2,16}$	1	1	1	1	1	1	1	1	0	1

called Adverse sets. At least one more system iteration is then required to perform isolation.

Being adverse depends on the number of faults affecting the estimates as stated in the Proposition 2.:

Proposition 2. *A set of faulty states is adverse if and only if it complies with one of the following rules:*

$$f_{N_i} = 1, \forall i \in \{1, \dots, K\} \text{ and } \exists! j \in \{1, \dots, K\} \text{ such as } f_{G_j} = 0$$

$$f_{G_i} = 1, \forall i \in \{1, \dots, K\}$$

In other words it is not possible to isolate faults if:

1. Every estimates N is faulty and there is a unique fault-free G .
2. Every G is faulty.

The proof of Proposition 2 is given in Appendix B.

One can notice in the example developed previously that after the first system iteration (i.e. $K = 1$), the situation actually complied with the second rule of this proposition because $f_{N_1^s} = 0$ and $f_{G_1^s} = 1$. This is why fault isolation was not possible. However, after the second iteration, the set chosen for the example $\{f_{G_2^s} = 0 \text{ and } f_{N_2^s} = 0 \text{ and } f_{G_1^s} = 1 \text{ and } f_{N_1^s} = 0\}$ does not comply with any of these rules. Fault isolation was then possible.

Proposition 2 is capital for demonstrating the internal formalism properties. These are detailed and demonstrated as follows.

4.4 Formalism Properties

Once the formalism bases are set, a set of properties are shown here through the listed propositions. These demonstrate the formalism stability in terms of number of iterations and are essential to justify the interest of the method for integrity monitoring of navigation systems.

Proposition 3. *Guaranteed fault detection: The formalism always detects the presence of faulty estimates. In other words, as soon as there is a faulty estimate, the formalism detects it (but may not be able to isolate the faulty estimates).*

Proof. This is demonstrated by showing that the formalism is always able to identify the only set that contains no fault. Proposition 2 indeed shows that this set $e_{K,1}^s$ (for which $f_{N_i} = f_{G_j} = 0, \forall i, j \in \{1, \dots, K\}$) is not adverse. \square

Proposition 4. *Isolation convergence: The ratio between the number of adverse sets of faulty states and the total number of sets goes to zero when the number of iterations increases. In other words, increasing K improves fault isolation capabilities.*

Proof. Let $A(K)$ stand for the number of adverse sets of faults for K iterations. $A(K)$ is therefore the sum of the number of sets induced by the two rules of Proposition 2:

$$A(K) = C(K, 1) + \sum_{j=0}^K C(K, j) \quad (4.26)$$

Where $C(a, b)$ stands for the number of b -combinations from a given set of a elements. The binomial formula applied for coefficients 1 and 1 gives $\sum_{j=0}^K C(K, j) = 2^K$.

Then:

$$A(K) = K + 2^K \quad (4.27)$$

The ratio $q(K)$ between $A(K)$ and the total number of possible sets is:

$$q(K) = \frac{K + 2^K}{2^{2K}} \quad (4.28)$$

The limit of $q(K)$ as K goes to infinity is 0 which demonstrates the proposition. \square

Proposition 5. *Conservation of isolability: Once fault isolation is performed, fault isolation will be performed at any new iteration.*

Proof. Let I_k be the set of non-adverse e_k . Since this proof does not depend on the operating point, the superscript s is omitted. Reciprocally, let I_k^c be the complement of I_k , i.e. the set of adverse sets of faults. Moreover, let e'_{k+1} such as:

$$e'_{k+1} = \{f_{G_{k+1}}, f_{N_{k+1}}, e_k\} \quad (4.29)$$

The proposition can be written:

$$e_k \in I_k \implies e'_{k+1} \in I_{k+1} \quad (4.30)$$

This can be proved by demonstrating the contrapositive:

$$e'_{k+1} \in I_{k+1}^c \implies e_k \in I_k^c \quad (4.31)$$

Let e'_{k+1} be adverse at iteration $k+1$ such as $e'_{k+1} \in I_{k+1}^c$. Then e'_{k+1} complies with one of the rules stated in Proposition 2. On the one hand, if e'_{k+1} is adverse due to rule 1, this may be caused by two reasons:

1.a. $f_{N_i} = 1, \forall i \in \{1, \dots, k+1\}$ and $\exists! j \in \{1, \dots, k+1\}$ such as $f_{G_j} = 0$ and $j \neq k+1$, or

1.b. $f_{N_i} = 1, \forall i \in \{1, \dots, k+1\}$ and $\exists! j \in \{1, \dots, k+1\}$ such as $f_{G_j} = 0$ and $j = k+1$.

If e'_{k+1} complies to rule 1.a, then:

$$f_{N_i} = 1, \forall i \in \{1, \dots, k\},$$

$$\text{and } \exists! j \in \{1, \dots, k\} | f_{G_j} = 0 \quad (4.32)$$

so

$$e_k \in I_k^c \quad (4.33)$$

If e'_{k+1} complies to rule 1.b, then:

$$f_{G_i} = 1, \forall i \in \{1, \dots, k\} \quad (4.34)$$

so

$$e_k \in I_k^c \quad (4.35)$$

This shows that, if e'_{k+1} is adverse due to rule 1, then (4.31) is satisfied.

On the other hand, let us consider the case for which e'_{k+1} is adverse due to rule 2.

$$f_{G_i} = 1, \forall i \in \{1, \dots, k+1\} \implies f_{G_i} = 1, \forall i \in \{1, \dots, k\} \quad (4.36)$$

then

$$e_k \in I_k^c \quad (4.37)$$

This finally fulfils the demonstration of (4.31) and proves the conservation of isolability property. \square

Proposition 6. *Adaptation: If fault detection and isolation are performed, then adaptation is possible. As a reminder, adaptation consists in identifying a fault-free estimate once detection and isolation have been performed.*

Proof. The only set $e_{K,n}^s$ in which every estimate is faulty is adverse (the faults are not isolable because of the second rule of Proposition 2). The consequence is that every isolable fault configuration contains at least one true estimate. As isolation is performed, the fault-free estimate is perfectly identified within the set. \square

Proposition 7. *Conservation of adaptation: If fault isolation is achieved at the K^{th} iteration, adaptation is possible at iteration $K + 1$ whatever the faults affecting the new estimates.*

Proof. The Proposition 5 states that, if fault isolation is performed at iteration K , isolation will also be performed at $K + 1$. Moreover, Proposition 6 shows that adaptation is always possible as soon as faults are isolated. Then, if faults are isolated at iteration K , adaptation is possible at $K + 1$. \square

These propositions have important consequences for the application of the method for intelligent vehicles. First, Proposition 3 shows that the presence of a fault among the available estimates is always detected by the method. This means that the method is able to state with certainty that there is no fault, even from the first system iteration and permits client systems to operate. In this sense, integrity monitoring is possible with this method. Second, since fault isolation is based on the use of multiple system iterations, the stability is important and shall be demonstrated. Proposition 4 indeed shows that a new iteration always bring information for fault isolation and justifies the use of multiple system iterations. Third, according to Propositions 5 to 7, as soon as isolation is performed, it is possible to provide a fault-free estimate to client systems and to do so for any future iteration. Client systems can then anticipate they will be able to operate properly at any future iteration.

4.5 Illustrative Examples

The proposed FDIA formalism is applied to the integrity monitoring of the navigation vehicle position estimate as introduced in Figure 4.1 and each step is detailed through two simple examples. In the first example, the map contains an error and we show how the method performs fault detection, isolation and adaptation. Each step of the proposed method is detailed and the properties introduced in Section 4.4 are illustrated. In the second example, the map geometry is correct but a large building may cause errors on vehicle position estimates from sensors. This shows how the method rejects faulty estimates.

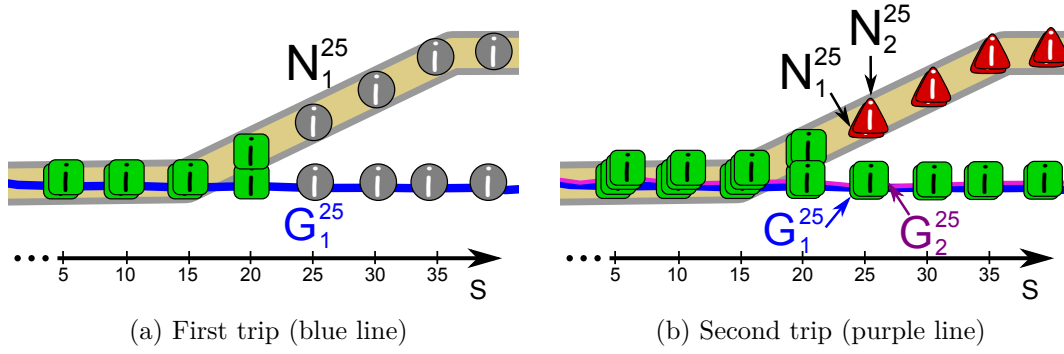


Figure 4.7: A faulty map area. Circular grey marks are for estimate for which the method has detected but not isolated a fault. Green squares are for true estimates and red triangles are the faulty estimates.

4.5.1 Navigation Map Geometric Fault

In this example (depicted in Figure 4.7), the true road goes straight whilst there is an error in the representation of the road in the map. Let us detail the proposed formalism at abscissa 25 m of the first trip shown in Figure 4.7a.

The first time the vehicle is at abscissa $s = 25$, position estimates are provided by the vehicle state (G_1^{25}) and by the navigation (N_1^{25}) functions. One can compute the observed residual with equation (4.7):

$$G_1^{25} \neq N_1^{25} \implies r_{G_1^{25} N_1^{25}} = 1$$

This residual is found three times in the truth table for one trip FDI (Table 4.1): the sets of faults $e_{1,2}^{25}$, $e_{1,3}^{25}$ and $e_{1,4}^{25}$ give $r_{G_1 N_1} = 1$. The proposed method then detects a faulty estimate among G_1^{25} and N_1^{25} but is not able to isolate it. The integrity monitoring system cannot state on the faultiness of N_1^{25} , it sends *Knowledge of fault: Unknown* to client systems as shown by circular grey marks on Figure 4.7a.

The second time the vehicle crosses abscissa $s = 25$ of the same road (Figure 4.7b), a new pair of position estimates becomes available: G_2^{25} and N_2^{25} . The residual vector dimension increases to 6. The elements are calculated using (4.6), (4.7) and (4.8):

$$N_2^{25} \neq G_2^{25} \implies r_{N_2^{25} G_2^{25}} = 1$$

$$G_2^{25} = G_1^{25} \implies r_{G_2^{25} G_1^{25}} = 0$$

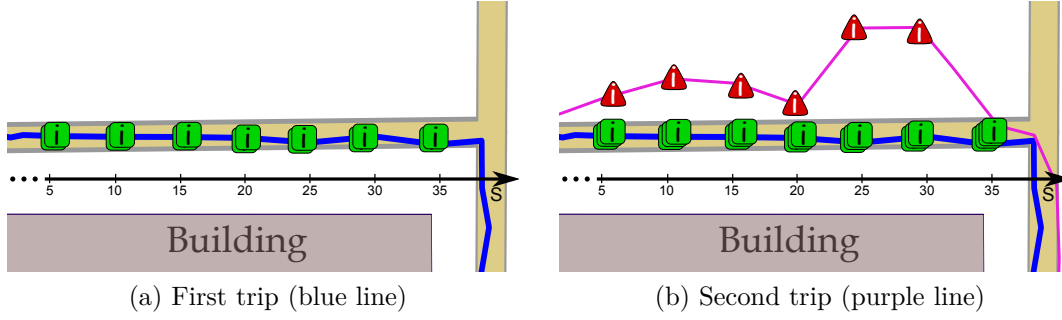


Figure 4.8: A correct map area close to a large building. Red triangles (resp. green squares) are for estimates which the method has isolated as faulty (resp. true).

$$N_1^{25} \neq G_2^{25} \implies r_{N_1^{25}G_2^{25}} = 1$$

$$N_2^{25} \neq G_1^{25} \implies r_{N_2^{25}G_1^{25}} = 1$$

$$N_1^{25} = N_2^{25} \implies r_{N_1^{25}N_2^{25}} = 0$$

$$G_1^{25} \neq N_1^{25} \implies r_{G_1^{25}N_1^{25}} = 1$$

$$\text{Then } R_2^{25} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Table 4.2 is the truth table for two trips. According to the first trip observation, one knows that $f_{G_1^{25}}$ and $f_{N_1^{25}}$ are not both null hence the first four rows of Table 4.2 could be ignored. The observed residual is found only once in this table (caused by the set of faults $e_{2,11}^{25}$), one can then conclude that $f_{G_2^{25}} = 0$, $f_{N_2^{25}} = 1$, $f_{G_1^{25}} = 0$ and $f_{N_1^{25}} = 1$.

The integrity monitoring system returns the instruction *Knowledge of fault: don't use the navigation position estimate* (N_2^{25}) and provides a fault-free estimate instead in the output *Correction* (either G_2^{25} or G_1^{25}). On Figure 4.7b, faulty (resp. true) estimates are represented by red triangles (resp. green squares). Due to Proposition 7, the integrity monitoring system will be able to perform adaptation, that is to provide an error-free position estimate, for all future vehicle trips on this road whatever the faults affecting the future estimates.

4.5.2 Isolation of GNSS Faults

In the second example, shown in Figure 4.8, there is no error in the map but the road is close to a large building that may cause GNSS faults. No fault affects vehicle position estimate the first time the vehicle goes in this area (Figure 4.8a). The method steps is detailed for abscissa $s = 5$ but in this particular example, these are the same for $s = \{10, 15, \dots, 35\}$. At the first trip, available estimates are such that

$G_1^5 = N_1^5$. According to (4.7), the residual is $R_1^5 = 0$. Table 4.1 indicates that only one set gives this residual, i.e. $e_{1,1}^5 = \{f_{G_1^5} = 0 \text{ and } f_{N_1^5} = 0\}$. The method states with certainty that there is no fault at this abscissa and provides the information *Use* to client systems that can work properly.

According to Proposition 7, one knows that fault isolation and adaptation will be performed at any future trip on this road whatever the faults encountered. This can be noticed in the second trip truth table (Table 4.2) because the residual corresponding to the sets in which $f_{G_1^5} = 0$ and $f_{N_1^5} = 0$ are unique ($e_{2,1}$, $e_{2,2}$, $e_{2,3}$ and $e_{2,4}$ are in green in the table). Moreover, in these four sets, there is at least one estimate that is not affected by a fault and that can be used for performing adaptation, if necessary.

Figure 4.8b illustrates this with the second trip which is perturbed by the buildings. At abscissa $s = 5$, the available estimates are now such as $G_2^5 \neq N_2^5 = G_1^5 = N_1^5$. The residual calculated with (4.6), (4.7) and (4.8) is $R_2^5 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$. This residual is solely caused by $e_{2,2}^5 = \{f_{G_2^5} = 1 \text{ and } f_{N_2^5} = 0 \text{ and } f_{G_1^5} = 0 \text{ and } f_{N_1^5} = 0\}$ as shown in the truth table. The method identifies the faulty vehicle position estimates and keeps confidence in current and past navigation estimates; adaptation is not necessary here.

Two opposite cases have been shown by these examples: one erroneous map area in good GNSS conditions and one correct map area in poor GNSS conditions. The method works properly even if faults affect both estimates. In such a situation, fault detection is directly made but isolation and adaptation require more trips. This is shown in the next section.

4.6 Practical Implementation

The method has been evaluated using data collected with a standard passenger vehicle. The vehicle was driven several times on the same path. Quantitative results were obtained by running the method offline with the real vehicle data and using randomly erroneous navigation maps. The performance is evaluated according to several criteria. This section describes these metrics and the results obtained.

4.6.1 Experimental Set-up

All data used in the following were acquired from a standard GPS receiver on board a passenger vehicle. The map-matching function uses an editable navigation map. The stored road geometries were locally changed to simulate errors found in such navigation maps. All algorithms were coded in C++ language.

For all the experiments, two estimates are used for the analysis: the estimation of the vehicle position G and the projection of the vehicle position onto the road

network of the navigation map provided by the navigation function N (as defined in Figure 4.1). During the experiments, the subject vehicle travelled several times over the same roads, thus several estimates of the observed variables were available at any abscissa s . The code for automatic generation of the truth tables required for the method implementation is provided in Appendix C.

4.6.2 Spatial Sampling

As stated in the problem formulation and in order to take benefit of the repetitive vehicle trips, the method has to be spatially sampled. This makes possible to compare the estimates of multiple trips for one common vehicle true position which is seen as a system operating point. The method is triggered on the vehicle abscissa onto the map road which is provided by the navigation function through the EH.

Available sensors are timely triggered and there frequency is limited. In these experiments, spatial sampling is obtained by sub-sampling temporal data, i.e. sensor data is rejected unless they correspond to a desired abscissa with a tolerance λ_s . According to the Nyquist-Shannon theorem and given the vehicle speed, the method spatial sampling must be at least twice the distance between two consecutive temporal data. To avoid any aliasing problem, we require the spatial sampling to be at least five times the temporal data for a given vehicle speed. The GPS receiver used had a maximum frequency of 10 Hz. Moreover, we choose to evaluate the map information every 10 m which represents a satisfying distance to alert from the ADAS point of view. The maximum vehicle speed is then 20 m/s, i.e. 72 km/h. The navigation estimates are rejected unless they correspond to a sampled abscissa with a $\lambda_s = 1$ m tolerance.

4.6.3 Relaxing Equality Constraints

As stated in Section 4.3.2, the residual processing is based on the pairwise comparison of vehicle position estimates. Strict equalities were employed to settle the mathematical background but they have to be relaxed for a practical implementation with real estimates since two physical quantities cannot be rigorously equal. The residuals are then processed based on the distance between the estimates and using a given threshold λ_d . Equations (4.6), (4.7) and (4.8) become:

$$r_{G_i^s G_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \text{dist}(G_i^s, G_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (4.38)$$

$$r_{G_i^s N_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \text{dist}(G_i^s, N_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\} \quad (4.39)$$

$$r_{N_i^s N_j^s} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \text{dist}(N_i^s, N_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (4.40)$$

4.6.4 Gating Effects

Relaxing the equality constraints may cause unwanted effects. Indeed, with strict equalities:

$$G_1 = G_2, G_2 = N_1 \implies G_1 = N_1 \quad (4.41)$$

This implication used in the method is no longer possible. Indeed, with $\lambda_d > 0$:

$$\text{dist}(G_1, G_2) \leq \lambda_d, \text{dist}(G_2, N_1) \leq \lambda_d \not\Rightarrow \text{dist}(G_1, N_1) \leq \lambda_d \quad (4.42)$$

Gating effects may therefore arise as illustrated by Figure 4.9 which shows one configuration of estimates at the second trip ($K = 2$) that leads to a theoretically impossible residual, i.e. a residual that is due to none of the sets of fault and that is therefore not in the truth table. The configuration chosen in this example gives the residual $R_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$ which is not in the truth table (Table 4.2).

The threshold λ_d must then be chosen with care. A large threshold leads to a great number of impossible residuals and masks the faulty estimates. On the other hand, a very small λ_d can cause fault-free estimates to be different from each other. The minimum acceptable value of λ_d is given by the curvilinear abscissa tolerance. Two navigation estimates N_i and N_j (with abscissa s_{N_i} and s_{N_j} respectively) are associated to the same abscissa s if $s_{N_i} \in [s - \lambda_s, s + \lambda_s]$ and $s_{N_j} \in [s - \lambda_s, s + \lambda_s]$. On an error-free road, these must be considered as equal, the threshold λ_d is therefore such as $\lambda_d \geq 2\lambda_s$. It is finally set to be:

$$\lambda_d = 2\lambda_s \quad (4.43)$$

In case of impossible residual, the Hamming distance can be used to find the closest residual in the truth table. The Hamming distance is defined as the number of different digits in two strings [60]. In the example introduced by Figure 4.9, the closest residual of the truth table is $R_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$ and is the consequence of the set $e_{2,5}^s = \left\{ f_{G_2^s} = 0 \text{ and } f_{N_2^s} = 0 \text{ and } f_{G_1^s} = 1 \text{ and } f_{N_1^s} = 0 \right\}$. One can notice that this set of faulty states is a credible explanation for this example since the three

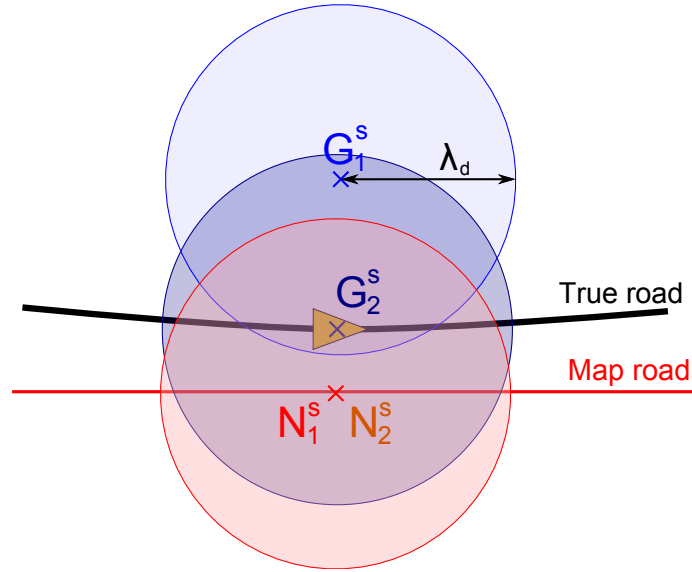


Figure 4.9: Situation causing a theoretically impossible residual at the second vehicle trip (yellow arrow) on the true road (black curve). Vehicle position estimates are G_1^s and G_2^s . The navigation estimates N_1^s and N_2^s are superimposed. Each estimate is at the centre of a circle that represents the threshold. For example, if G_2^s is inside the G_1^s circle, then the method considers $G_1^s = G_2^s$.

clustered estimates are judged not faulty (N_1^s , G_2^s and N_2^s) and the estimate aside is said faulty (G_1^s). In order to evaluate the formalism by itself, the Hamming distance is not used in the experimental evaluation.

4.6.5 In-Vehicle Algorithm

The FDIA framework as introduced so far, used the estimates stored in the vehicle memory that correspond to all the vehicle trips. However, due to the properties demonstrated in Section 4.4, it may not be necessary to use all the available estimates. In order to reduce the computations made by the method, the principles of the algorithm used for the experimental evaluation are to use passed trips until the framework isolates faults as illustrated by Algorithm 4.1. If the current pair of estimates induces an adverse residual, the estimates of the previous trip are used and so on until the residual vector is isolable. When an isolable residual is found, the corresponding state of fault of the current estimate from navigation is sent to client systems. If this estimate is faulty, a fault-free estimate is found and is provided to client system as a correction. If the number of vehicle trips is insufficient the output *Knowledge of fault* is set to *Unknown*.

The truth tables used by the framework to determine the sets of faults based on

Algorithm 4.1 In-vehicle FDIA algorithm

Inputs: Current estimate from sensors G_k and estimate from navigation N_k **Outputs:** State of fault of the current estimate from navigation N_k and correction if needed**Begin**Store G_k and N_k in the internal memory**If** $r_{G_k N_k} = 0$, **Return** Use // *The current navigation estimate is not faulty***Else** // *A fault is detected***If** G_{k-1} **and** N_{k-1} **exist** // *Try an isolation with previous estimates*

Compute residual vector for two trips

Find this residual in the truth table

If this residual is unique**If** $f_{N_k} = 1$, **Return** Don't Use, provide a fault free estimate for adaptation**Else**, **Return** Use**Else**, Use trip $k - 2$ if available and so on, else return Unknown**Else**, **Return** Unknown**End**

the residual vector are independent on the observed estimates. They are therefore computed once for all and stored in the memory. Some of the table rows can also be deleted in order to reduce the memory allocated to their storage and to accelerate the search processes. In the K^{th} trips table, the rows that correspond to isolable residual with $K - 1$ trips can be deleted since they will never be used by the algorithm. For example, if we consider the truth table for two trips shown in Table 4.2, the four first rows (i.e. $e_{2,1}$, $e_{2,2}$, $e_{2,3}$ and $e_{2,4}$) can be deleted because they correspond to sets of faults that are isolable with only one trip (row $e_{1,1}$ in Table 4.1).

4.7 Experimental Evaluation

4.7.1 Metrics

It was shown that the proposed method has three possible output states that refer to the current navigation estimate integrity, namely *Use*, *Unknown*, *Don't use*. The navigation map can be actually *correct* or *faulty*. A set of metrics is introduced as follows and illustrated by Figure 4.10. These are evaluated with respect to the number of vehicle trips to evaluate precisely the method performance.

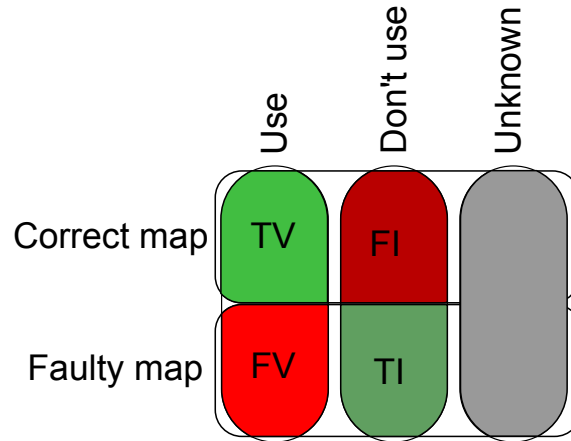


Figure 4.10: Metrics employed for method evaluation. The columns are the output of the method and the lines are the actual state of the navigation map.

True Isolation (TI)

A TI occurs when a point of the map is faulty and the method informs correctly the client system of this fault. For good performance, the method must produce high number of TI.

False Isolation (FI)

A FI occurs when the method detects a fault on the current navigation estimate but the map is correct at this given point. From the user's point of view, a FI can be seen as a false alarm and should be as rare as possible. The False Isolation Rate (FIR) is defined as:

$$FIR \stackrel{\text{def}}{=} \frac{FI}{\Omega_c} \quad (4.44)$$

where Ω_c is the number of correct points in the map. A high FIR would affect the client ADAS availability.

True Validation (TV)

A TV occurs when a correct point of the navigation map has been declared with no fault. Similarly to TI, the number of TV is expected to be high.

False Validation (FV)

A FV occurs when the method trusts a faulty navigation estimate. The False Validation Rate (FVR) is defined as:

$$FVR \stackrel{\text{def}}{=} \frac{FV}{\Omega_f} \quad (4.45)$$

where Ω_f is the number of faulty points in the map. This is a first importance figure since the proposed method intends to avoid malfunction of client systems due to faulty estimates; the FVR must therefore be as low as possible.

Overall Efficiency (OE)

The OE counts the relevant diagnoses of the method. This is therefore the sum of TI and TV. The Overall Efficiency Rate (OER) is:

$$OER \stackrel{\text{def}}{=} \frac{TV + TI}{\Omega - \Omega_{unknown}} \quad (4.46)$$

where Ω is the number of navigation points evaluated by the method and $\Omega_{unknown}$ is the number of navigation estimates for which the method states *Unknown*. An OER close to one would indicate that as soon as the method provides an output different from *Unknown*, this diagnosis is reliable.

Information Availability (IA)

The output *Unknown* doesn't provide information on the integrity of the navigation estimate from the point of view of client systems. From the applicative point of view, this output must be as rare as possible. The performance of the method in terms of information availability is measured by the Information Availability Rate (IAR):

$$IAR \stackrel{\text{def}}{=} \frac{\Omega - \Omega_{unknown}}{\Omega} \quad (4.47)$$

It is expected to converge to one as the number of trips increases.

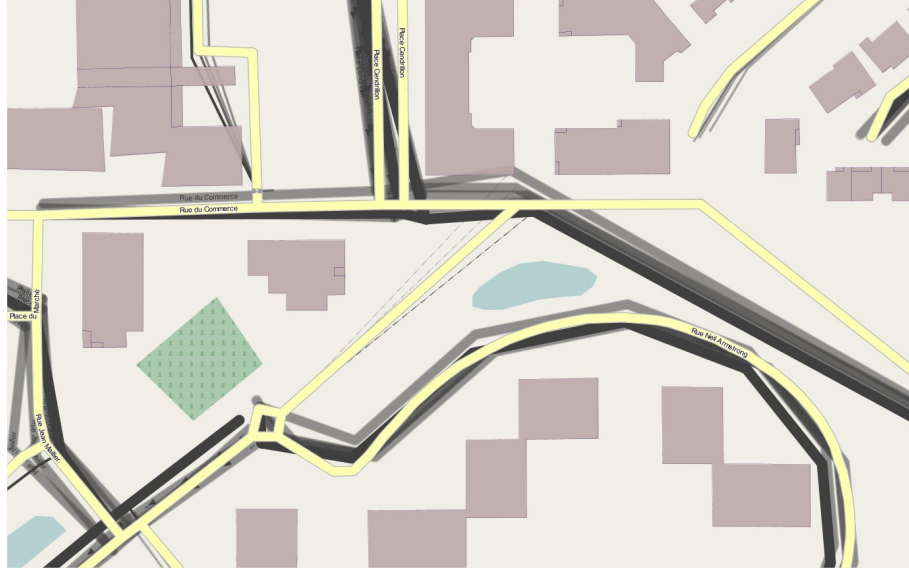


Figure 4.11: Five randomly generated navigation maps for method evaluation. The correct map is on top in yellow and the generated maps are in shades of grey.

Spatial Availability (SA)

The method requires spatial sampling of sensor data. However, current available sensors provide data at a constant rate. To address this problem, these are re-sampled according to the vehicle curvilinear abscissa on the road. Some evaluating points may be missed because of the high vehicle speed or to the low frequency of the sensors. This is characterized by the Spatial Availability Rate (SAR):

$$SAR \stackrel{\text{def}}{=} \frac{\Omega}{\Omega_{map}} \quad (4.48)$$

where Ω_{map} is the number of points to evaluate.

4.7.2 Automatic Map Fault Generation

A large variety of navigation maps that contain different geometric faults is required to evaluate the method. The faults contained in these maps must be similar to those that can be encountered in usual commercial navigation maps. In order to be independent of the FDIA method, these faults are randomly created in the navigation map in a two stages process. First, a random set of ways is chosen within the navigation map. Those are affected by a random spatial bias (i.e. all the nodes of

the way are translated according to the same vector). This simulates a road offset fault as introduced in Section 1.2.2. Second, the road shape nodes are sub-sampled to represent mapping simplifications. Figure 4.11 shows an example of a navigation map created by this process. The original map is generated manually for each experiment area with high precision localization sensors (inertial navigation sensor coupled with dual-frequency GPS receiver).

For the experiments, the navigation map is loaded in the navigation software. This provides the navigation estimate in real time through the electronic horizon to the method. This tool is described in Appendix A.

4.7.3 Urban Test Track

In this experiment, the vehicle was driven close to large buildings. The GPS receiver was perturbed by multipath effects caused by the signal reflection on buildings. As shown by Figure 4.12a, some of the ways benefit from better conditions and the deviation of the GPS measurements is less than the road width. In other roads, the noise that affects the GPS measurements is larger. These measurement are expected to be isolated by the method. The length of each trip is 1100 m and the spatial sampling is set to 10 m as said previously. Hence $\Omega = 110$ points of the navigation map should be evaluated at each trip. This value varies of a few points for one trip to the other because the data recordings were not started and stopped rigorously at the same positions.

The automatic map error method were run five times on the correct map to produce the maps used in this test and shown in Figures 4.12b to 4.12f. The probability of fault generation on a map way was set to 0.5. For each road selected by this first random process, the mean and standard deviation of the random spatial biases applied to the road were 0 m and 5 m respectively in both directions. Finally, one every two road shape nodes were deleted. One can notice that this last step does not change much the road shape because the test track is made of nearly straight roads. In this experiment, each driving lane of the test area is modelled by a one-way road in the navigation map. The ways on which the navigation system matches the vehicle position in therefore different from one driving direction to the other.

The method is evaluated using the metrics introduced in the previous section. Figures 4.13 and 4.14 show the values of each metric is plotted according to the trip number. The value is plotted on the first graph and the second graph sums up the trends of the values with box plots. The box plots represent the statistics of the value taken by each metric at each trip. The lower and upper edges of the box are the first and third quartiles respectively. The median is shown by the central red line. The whiskers show the minimum and maximum values omitting outliers which are shown by red stars. The box plot is a more complete representation a set

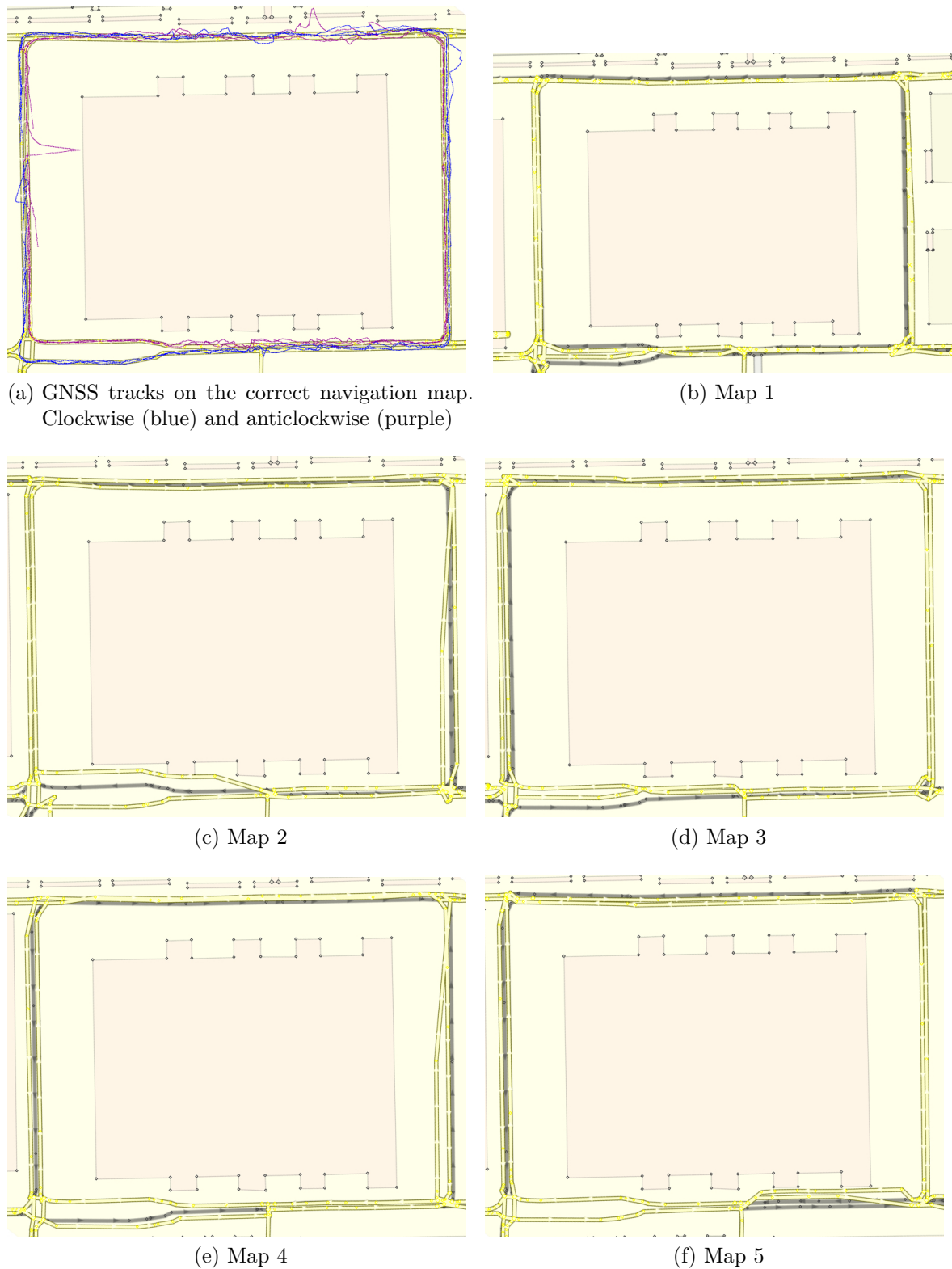


Figure 4.12: GNSS tracks and navigation maps. The maps used in the experiment are in yellow and the correct map is in background and in grey. The buildings are in light pink.

of results than the first and second moments and it does not require to make any assumption on the probabilistic distribution.

Figures 4.13a and 4.13b show the FIR with respect to the number of trips. At the first vehicle trip, the method potentially returns the output *Use* or *Unknown*. Since isolation is never performed at the first trip, the FIR equals zero at the first trip for all the tested maps. The first significant trip for the FIR is then the second trip in which the minimum is 1%, the maximum is 11% and the median is 5.0%. The FIR are all due to situations in which the vehicle position estimates from sensors of the two trips (G_1 and G_2) are faulty but close enough from each other to be considered as equal by the method. The distance between those estimate and the estimate from navigation (N_1 and N_2) is however just above the threshold $\lambda_d = 2$ m (the distance is close to 3 m). The residual calculated by the method is $R_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$. The method wrongly isolates the estimates from navigation because the situation does not comply with the first assumption made in Section 4.2. In the worst test which is the map 3 clockwise, it concerns 11 points. At the third trip, all the statistics shown by the box plot decrease: the minimum is 0%, the maximum is 6.6% and the median is 3.3%. The FIR point at this trip originate from the same reason as at the second trip. However, some of the FIR points of the second trip are isolated correctly at the third trip because the new estimates satisfy the assumption.

The evolution FVR with the number of trips is detailed by Figures 4.13c and 4.13d. Two tests show FVR equal to zero for the three trips however the other tests shows higher rates which are between 2% and 12%. The anticlockwise tests on the map 1 and 3 and clockwise test on map 4 show especially high FVR with 27% which represents 14 points. Those are mainly located on the horizontal lower road of the map 1 and on the horizontal upper road of the maps 3 and 4. Those are due to faulty estimates from sensors which are close to faulty estimates from navigation. This does not comply with the third assumption of Section 4.2 and makes the method wrongly validate the current estimate from navigation.

Figures 4.13e and 4.13f depict the global method performance for the tested maps. The OER is in general greater than 75%. At the first vehicle trip, the method cannot perform isolation. The OE is then only composed of TV. The OE is between 77% and 100% due to the FVR which is between 0% and 20% at the first trip as said previously. It must be noticed here that, given a test map, the OE is not strictly equal to $1 - FVR$ because the OE does not take into account the point for which the method stated *Unknown* as detailed in (4.45) and (4.46). At the second vehicle trip the OE decreases for nearly all the maps. This is mainly due to the not FIR which is not null from the second journey. It can also be noticed that a cluster appears for which the OE is lower which is made of the tests 1, 3 anticlockwise and 4 clockwise. These tests present a low OE because of their high FVR at the second trip. The third trip finally improves the OE for all the tests since it is between 82% and 97% with a median at 90%.

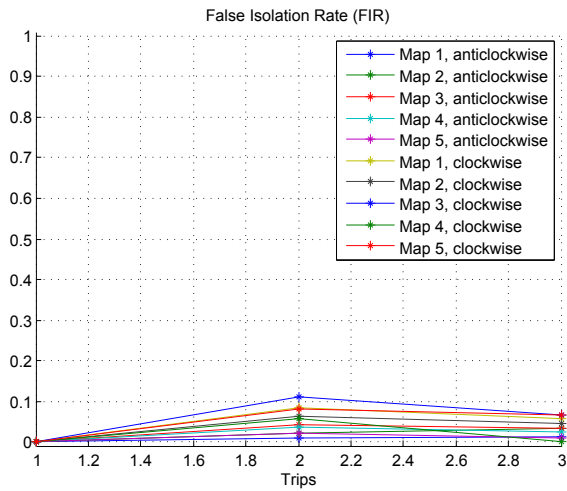
Figure 4.14a and 4.14b show the evolution of the IAR with the number of vehicle

trips. In this experiment, the navigation maps used contain a large part of erroneous roads (50% of the roads were made erroneous). Moreover, the GPS receiver suffers from multipath in many measurements. The method states *Unknown* every time there is at least one faulty estimate among G and N , the IAR is therefore low at the first trip. It can be noticed that this rate grows fast and exceeds 95% at the third trip and for all the maps. This shows that the *Unknown* output is a transition states and that the method is able to converge to either *Use* or *Don't use* for all the points of the navigation map in a reasonable number of trips. The isolation convergence property (stated in Proposition 4) is verified here experimentally.

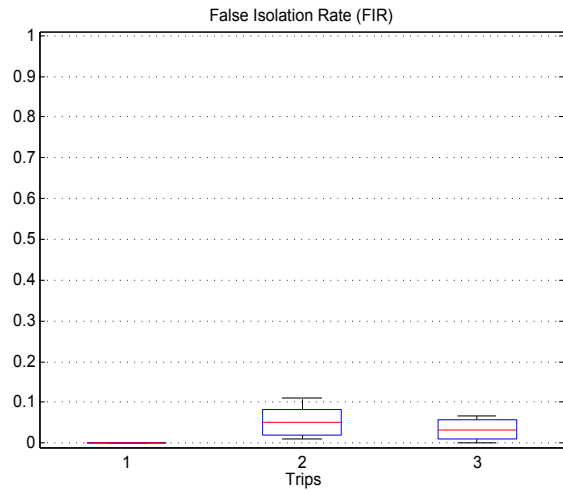
Figures 4.14c and 4.14d finally detail the SAR with the number of trips. It can be noticed that from the first trip, the method evaluates between 82% and 94% of the points of the map. The SAR increases for all the maps and reaches 90% to 100% of the points for the third trip. The points not evaluated by the method are mainly located close to the intersections and especially the one at the bottom left of the maps shown in Figure 4.12. This is due to the delay of the map-matching algorithm within the navigation function to provide an estimate N when the vehicle is close to the intersections. Moreover, a very few estimates are missed because of the re-sampling of temporal data which is due to the relatively low speed of the vehicle (less than 40 km/h). The box plot shows that the SAR tends to increase with the number of trips. The reason is that a point that was missed at one trip, due for instance to spatial sampling, is evaluated at the next trip. All the evaluated points are kept in the method memory so the spatial availability tends to increase at each new trip.

The first conclusion deduced from this experiment is that the method shows good performance in terms of information availability. The convergence of the IAR to one for all the tests and at the third trip justifies the basic idea of using multiples trips as a source of redundancy. It can also be noticed that, the experimental IAR is better than the theoretical value which is 83% at the third trip according to (4.28). The high level of spatial availability from the first trip and the increase of the SAR with the number of trips show that the spatial sampling of temporal data can be correctly achieved when the vehicle speed remains relatively low.

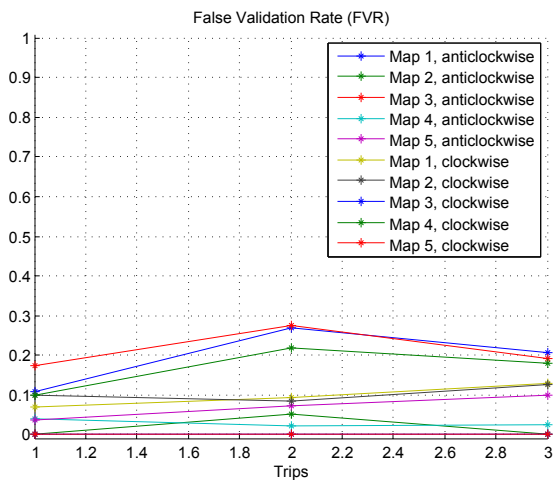
Nevertheless, the method shows less convincing performance in terms of its ability to identify faulty estimates. The reason for every wrong identification (FI and FV) is that either the first or the third assumption made in Section 4.2 as the basis of the method development are not verified. The distance between two estimates is less than the distance threshold and the method considers them as equal. For a large part of the wrong identifications, this distance is close to the threshold so reducing λ_d would improve the FIR, the FVR and the OE. However the minimum of this threshold is set by the tolerance on the curvilinear abscissa λ_s as stated in (4.43). The use of sensors triggered directly on the vehicle abscissa would permit to reduce λ_s and therefore λ_d .



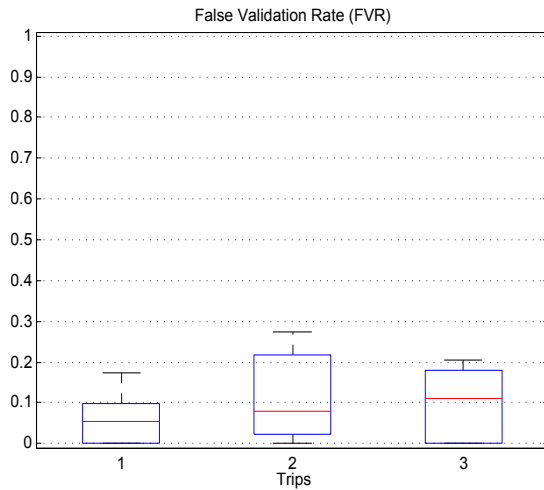
(a) False Isolation Rate



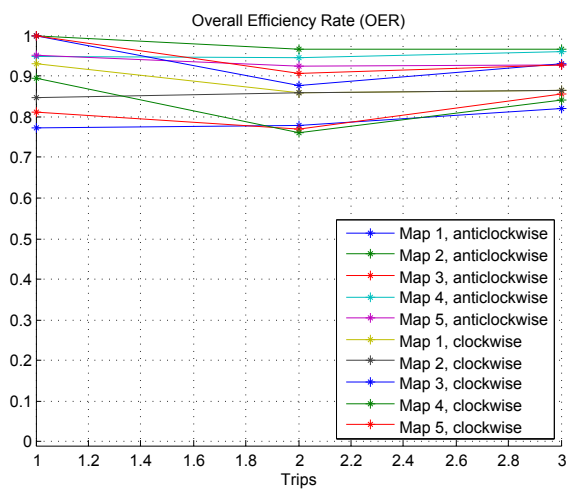
(b) Box plot of FIR.



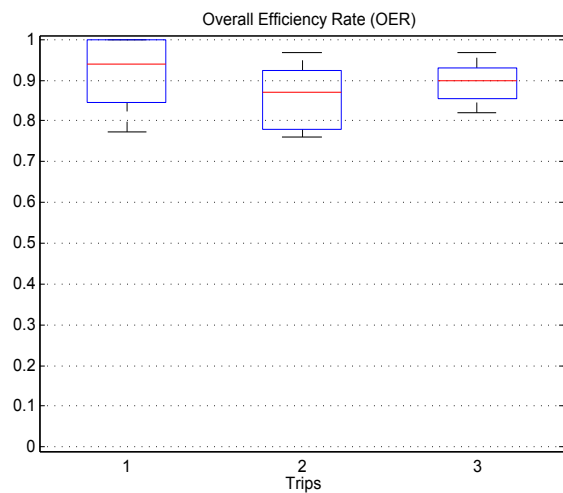
(c) False Validation Rate.



(d) Box plot of FVR.



(e) Overall Efficiency Rate.



(f) Box plot of OER.

Figure 4.13: False isolation rate, false validation rate and overall efficiency in urban test track

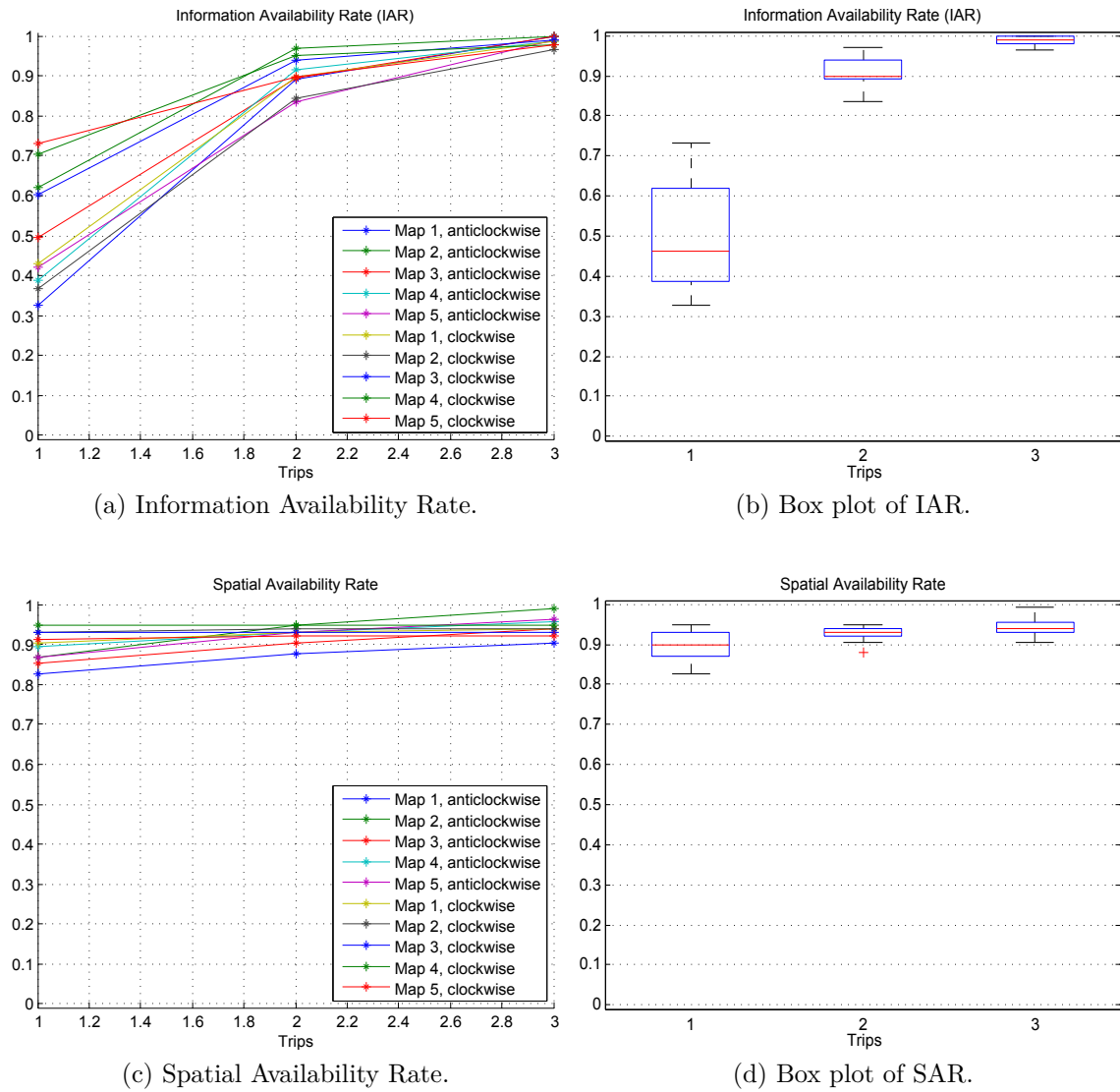


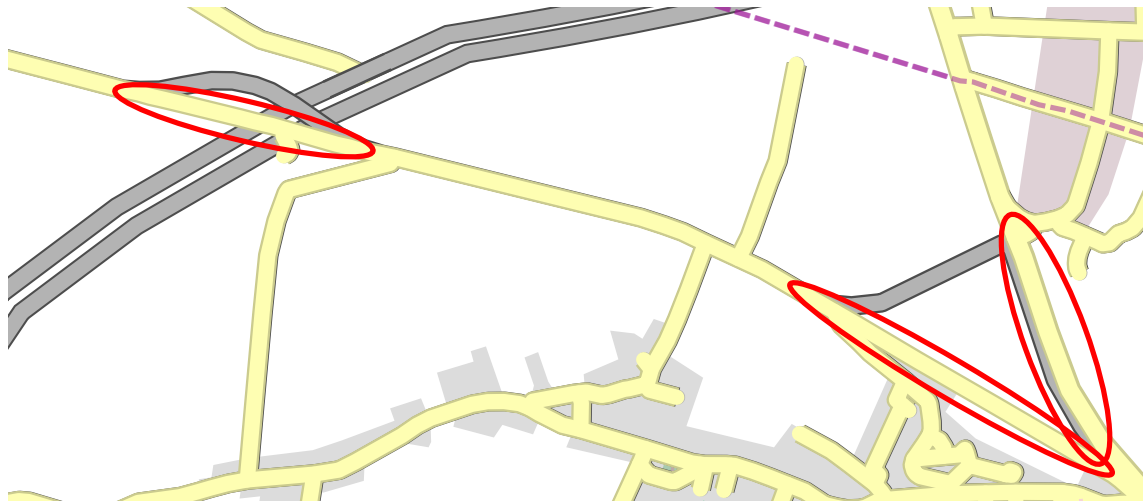
Figure 4.14: Information and spatial availabilities in the urban test track

It can be noticed that this experiment were led in tough conditions since the errors which were investigated were of the same order of magnitude than the road width. Furthermore, the GPS used in this experiment provides noisy data so some estimates G are a few metres aside the true position due to this noise but the mean value of this estimates is actually the true vehicle position. The problem is then to not compare each sample of G with the navigation map but to consider it as a random variable and to evaluate its mean over a few samples. Chapter 5 proposes a sequential test to address this problem.

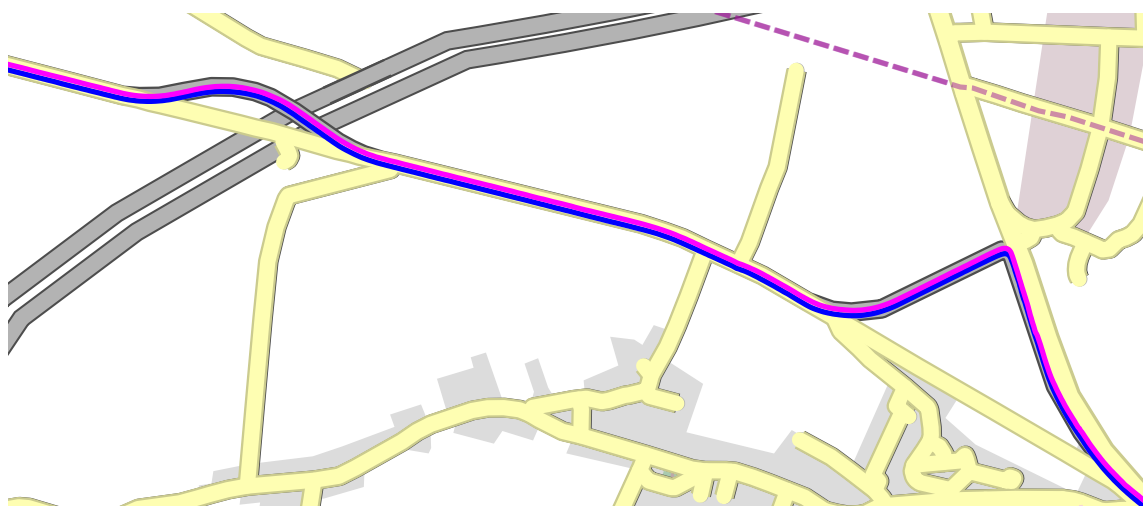
4.7.4 Rural Test Track

This paragraph details the method performance on an area where real map errors were found. The road was modified due to the construction of a motorway. A 2008 *Navetq* navigation map has been reproduced in the OpenStreetMap format in order to run the FDIA method. Figure 4.15a shows that this map contains three major errors described from left to right. The first one is a deviation of the new road where it passes over the motorway. The second is the creation of a completely new road that deviates from the old one. For these two, the confidences associated to both estimates from sensors and from navigation are high. Indeed, in rural environment, many satellites are in the receiver line-of-sight which increases the level of confidence and reduces the position standard deviations and Dilution of Precision. Moreover, the road network is quite simple so the map-matching algorithm provides a high level of confidence even if the GNSS measurement is a few metres aside the road. The challenge is therefore to determine precisely the reason of a disparity between estimates from sensors and from navigation, that is to determine which estimate is affected by a fault. When the true road is too far from the map road, the map-matching confidence index suddenly decreases and the navigation function switches to *off-road* mode and stops providing navigation estimates. The third error is the creation of a new road parallel the ancient one. Even if the estimate from navigation is relatively close to the true vehicle position in this area, the method should identify the fault. Figure 4.15b shows the estimates from sensors for the two trips used in this experiment.

Figure 4.16a shows the result of the FDIA applied to this dataset after the first trip. The green roads are those for which the method returned *Use* and the black roads are the roads for which the output was *Unknown*. The FIR is null due to the fact that the method cannot isolate a fault at the first trip, as detailed before. It can nevertheless be noticed that the FVR is also 0% and the OER is 100%. This means that the method correctly identified the situation in which no fault affected the estimates by providing the output *Use* and detected the situations in which at least one fault affected the estimates by providing the output *Unknown* to client systems. The IAR of this first trip is 77% which corresponds to the proportion of erroneous



(a) The errors of the navigation map are circled in red.



(b) The estimates from sensors of the first (resp. second) trip is in blue (resp. purple).

Figure 4.15: Rural test track. The navigation map used in the experiment is in yellow, the correct map is in grey in background. The vehicle goes from left to right.

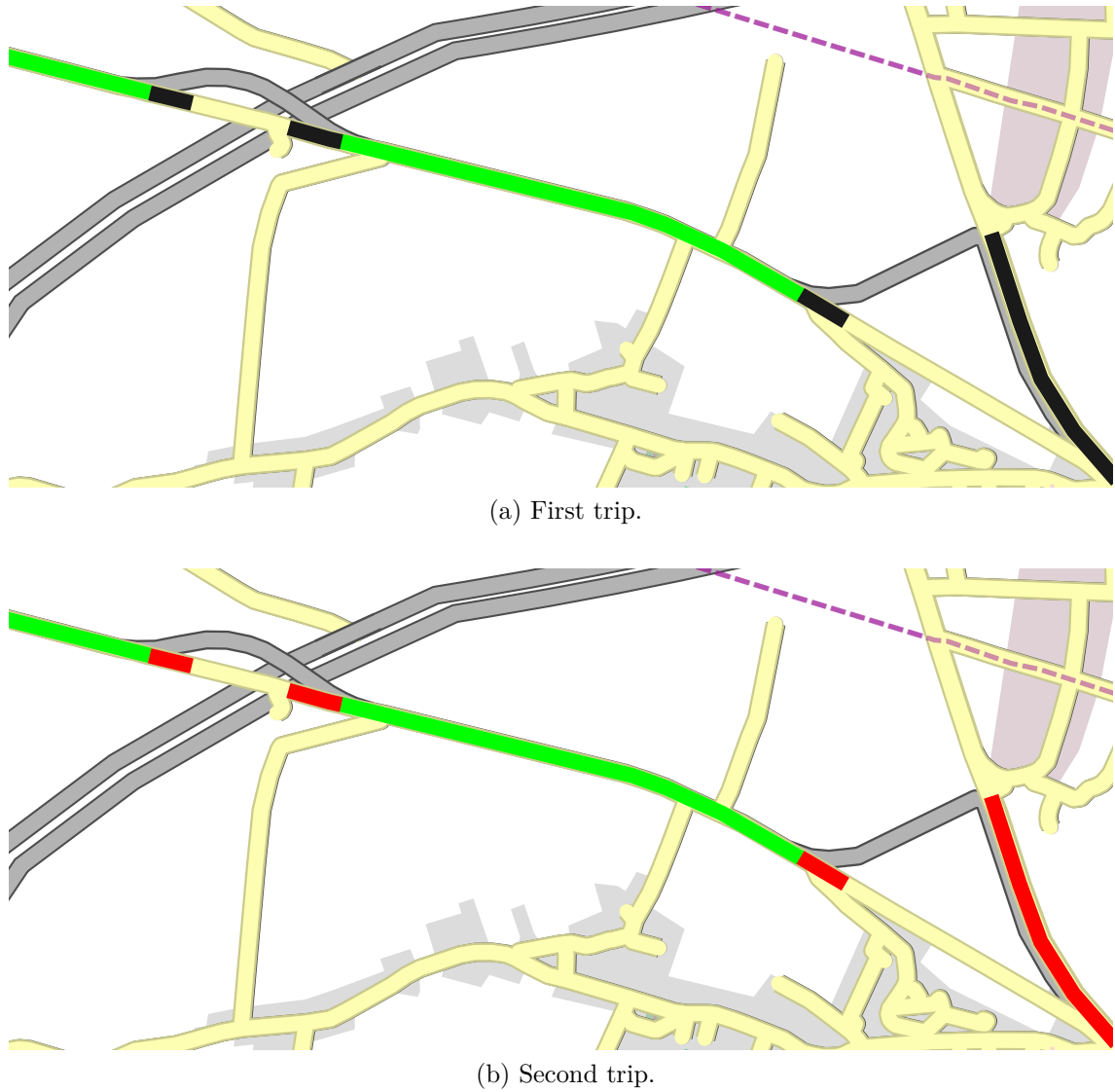


Figure 4.16: Results of the FDIA for the rural test track. The road sections for which the method states *Use* are in green, those for which the method states *Don't use* are in red and those for which the method states *Unknown* are in black. The true navigation map is in background in grey.

roads of the navigation map. The SAR is finally 87% for this first trip which means that 13% of the evaluation points were missed. Contrary to urban test track, the road network of this rural scenario contains only a few intersections, however, the vehicle speed was higher which did not facilitate the data spatial sampling.

The results obtained after the second vehicle trip in this area are shown by Figure 4.16b. Here again, $FIR = 0 \%$, $FVR = 0 \%$ and $OER = 100 \%$ which means that every estimate which is not said *Unknown* at the second trip is correctly identified. Moreover, every point that benefits from two trips are said either *Use* or *Don't use* so the Information Availability Rate is 100%. Finally, the SAR that corresponds to the second trip is 77%. For every point at which the estimate from navigation is said *Don't use*, the method performs adaptation by providing a fault-free estimate through the output *Correction*. Here, the fault-free estimate is the one from sensors of the first or second trip.

This experiment indicates that the method shows good performance when using real vehicle data and a real navigation map error. The absence of False Isolations and particularly False Validations and the high Information Availability in these conditions make the FDIA framework realistic for navigation integrity monitoring. The decrease of the Spatial Availability could be solved by using sensors triggered on vehicle position or using interpolation of the vehicle position estimates. This last wasn't used in this experiment in order to show only the FDIA performance which was the scope of this chapter.

4.8 Conclusion

This chapter has introduced a framework for monitoring the integrity of the navigation map geometry by detecting and isolating faults on the estimate of the vehicle position from the navigation system. It was shown that the context of intelligent vehicles in which this work takes place limits the quality of the sensors and the redundancy of the sources of information. The FDIA framework detailed in this chapter has proposed to fill this gap by taking benefit of the repetition the vehicle trips.

The principle used in this framework was the pairwise comparison of spatially-sampled vehicle position estimates at the current and past vehicle trips that forms residual vectors. In order to develop a FDIA framework with only an additional GNSS receiver which provides poor redundancy, several assumptions have been made: navigation faults are non-systematic and, when they occur, cause systematic errors on the position estimate; faults on position estimate from sensors are non-systematic and cause non-systematic errors; errors on estimates from the navigation are different from errors on estimate from sensors. Based on these three assumptions, it was shown that the observed residual vectors are also the result of Boolean operations between faults affecting the estimates: logical Exclusive OR for

a pair of estimates from the navigation and logical OR for other pairs of estimates. The observed residuals, found in the truth table, allow to determine the fault(s) affecting the estimates (i.e. to detect and isolate faults) and to identify fault-free estimates (i.e. to perform adaptation).

It was demonstrated that, as long as the assumptions are verified, the proposed FDIA framework is always able to perform fault detection. However, depending on the number of faults that affect the estimates and on the number of vehicle trips, it may not be possible to perform isolation, that is, to determine without ambiguity which estimate(s) is (are) affected by a fault. The mathematical definition of such sets of faults given in this chapter allowed to demonstrate that the fault isolation and adaptation capabilities of the method are improved as the number of vehicle trips increases.

The proposed framework was tested using real sensor data and based on navigation maps faults randomly and artificially generated or real faults existing in a database coming from a map provider. The results were evaluated using metrics defined in order to measure the method availability and its ability to correctly identify faulty and non-faulty navigation estimates. Navigation maps containing randomly generated geometric faults were used first to illustrate the method performance in complex scenarios. The isolation convergence property was clearly verified since the number of points for which the method cannot perform isolation decreases and is close to zero at the third vehicle trip. The Overall Efficiency was acceptable (greater than 80%) but a significant number of estimates remained wrongly identified (False Isolations and False Validations). Those were due to the fact that the map faults were of the same order of magnitude than the noise of estimates from sensors. This shows the limitations of the assumptions made to build the FDIA system in complex conditions (small map errors in difficult GNSS environment). Chapter 5 proposes an improvement of the detection method in order to get closer to the assumptions. A navigation map that contains a set of real geometric error was used next to test the FDIA framework in more usual conditions. The method showed a good performance since the Overall Efficiency was 100% for both trips. This highlights the interest of using this autonomous FDIA approach in intelligent vehicles.

A perspective for the work led in this chapter is to compare the results with a simplified version of the FDIA formalism. Based on the assumption of non-systematic errors of estimate from sensors, an approach for identifying fault-free estimates could be to find two estimates from sensors that are equal. Every estimate equal to those would be fault-free and others would be faulty. This approach would require less computations as the truth tables would no longer be necessary. However at least two vehicle trips at which the estimate from sensors are fault-free would be required. This is likely to increase the number of vehicle trips to settle especially in difficult GNSS areas.

Chapter 5

Extension of the FDIA Method to Handle Uncertainties

Contents

5.1	Introduction	105
5.2	Page’s Trend Test	106
5.3	FDIA with Page’s Trend Test	117
5.4	Conclusion	123

5.1 Introduction

The FDIA framework presented at the previous chapter is based on the comparison of the position estimate from sensors with the estimate from the navigation system at a given location of the vehicle. If these two estimates are equal, these are said fault-free. If not, a fault is detected and the estimates recorded at the past vehicle trips are used to isolate the fault (i.e. determine which estimate is affected by the fault). So far, a threshold on the distance between the estimates was used to relax the strict equality constraint required by the framework. However, the estimates from sensors are calculated based on a low quality GNSS receiver which equips standard passenger vehicles. The threshold can be reached due to the usual noise that affect the receiver. The comparison of the estimates form sensors and from navigation should no longer be done on individual samples but on their means. As shown by Figure 5.1, this comparison is done here by Page’s trend test.

This chapter introduces the mathematical formulation of Page’s trend test and its application to the detection of discrepancy between the vehicle position estimates. The performance of this test in terms of fault detection is presented first independently from the FDIA process. Two other statistical tests, namely Neyman Pearson and the mean tests, are used to compare the performance and discuss the choice of Page’s trend test. The manner in which this test can be used in combination with

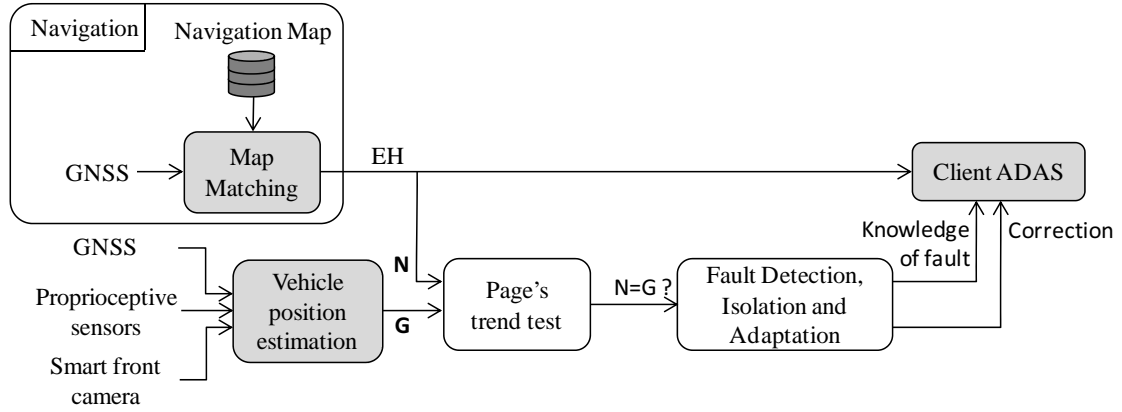


Figure 5.1: Page’s trend test for fault detection in navigation integrity monitoring context.

the FDIA framework presented in Chapter 4 is then detailed. The contribution of this test on the FDIA performance is finally measured using metrics and based on the same real vehicle data as in the evaluation of the FDIA without Page’s trend test.

5.2 Page’s Trend Test

Statistical tests are appropriate to evaluate parameters of a probability law based on set of outcomes. In our application, we aim at detecting a change of the mean of the probability density function (PDF) of a set of observed data while the standard deviation of this PDF is in the same order of magnitude than the expected mean gap. Page’s trend test works sequentially and is especially efficient for stream data. The problem is therefore formulated as the detection of a change of the mean of a random variable that represents the distance between the estimates from sensors and from navigation as illustrated by Figure 5.2.

5.2.1 Signal Generation

This section details how the distance signal¹ is generated and described in terms of mean and standard deviation. Let us consider the estimate N from the navigation as the result of a random process based on the true vehicle position P in a frame \mathcal{R}_1 aligned with the road:

$$N = P + \alpha \tag{5.1}$$

¹The term “signal” is used here to make a clear distinction with the residuals used in the previous chapter

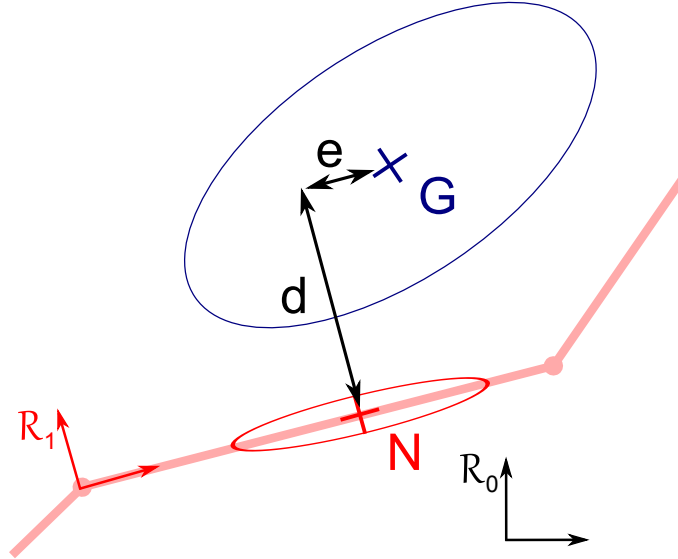


Figure 5.2: The distance between the estimate from sensors G and the estimate from navigation N taken as random variables. d and e are the lateral and longitudinal offsets between the estimates from navigation and from sensor respectively. \mathcal{R}_0 is the East-North plane locally tangent to the Earth surface and \mathcal{R}_1 is the frame aligned with the road segment on which the position is matched by the navigation function.

$$\Sigma_\alpha = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & 0 \end{bmatrix}_{\mathcal{R}_1} \quad (5.2)$$

where α is a noise supposed zero-mean with a diagonal covariance matrix Σ_α . Indeed, since the roads are represented in the navigation map by zero width poly-lines, the variance of the navigation map-matched error normal to the road segment is by definition null. However, a map-matched position error along the road segment exists, and σ_a denotes the longitudinal standard deviation of the navigation estimate.

The estimate of the vehicle position from sensors G can be encoded as a two-dimensional point $G = (x, y)^T$ in the East-North plane \mathcal{R}_0 locally tangent to Earth with the covariance matrix Σ_β of the estimation error β :

$$G = P + \beta \quad (5.3)$$

$$\Sigma_{\beta} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix}_{\mathcal{R}_0} \quad (5.4)$$

In order to make the distance signal independent of the road direction, an isotropic approach is chosen and it consists in using the outer circle of the ellipsoid. Its radius is $\eta = \max(\eta_i)$, η_i being the eigenvalues of Σ_{β} . So, the covariance matrix expressed in \mathcal{R}_1 is $\eta \cdot I$ (with I being the identity matrix).

In \mathcal{R}_1 , the difference between the map-matched and estimated positions is given by a vector L which has two independent components.

$$L = \begin{bmatrix} d \\ e \end{bmatrix} = N - G = \alpha - \beta \quad (5.5)$$

Under the hypothesis of independent errors, the lateral d and longitudinal e signals have the following variances:

$$\begin{aligned} \sigma_d^2 &= \eta \\ \sigma_e^2 &= \eta + \sigma_a^2 \end{aligned} \quad (5.6)$$

For the reasons detailed in Section 4.2, the relevant information in terms of the application is the lateral position of the roads in the navigation map. The fault detection is therefore made by detecting mean changes of the signal d .

5.2.2 Formulation of the Test

Page's test consists in statistically detecting a change in the mean of a random variable [14]. Let us consider q samples d_i of a random variable D . The likelihood of two hypotheses H_0 and H_1 are compared. The first hypothesis states that D has a constant mean μ_0 among the q samples. The second one assumes that, given $0 < r \leq q$, the mean of D is μ_0 for the first $r - 1$ samples and μ_1 for samples r to q :

$$\begin{aligned} H_0 &: d_i = \mu_0 + b_i, \quad i = 1, \dots, q \\ H_1 &: \begin{cases} d_i = \mu_0 + b_i, & i = 1, \dots, r - 1 \\ d_i = \mu_1 + b_i, & i = r, \dots, q \end{cases} \end{aligned} \quad (5.7)$$

where b is a zero-mean noise of standard deviation σ . The generalized likelihood

ratio of both hypotheses is given by (5.8).

$$\Lambda(D) = \frac{\prod_{i=1}^q p(d_i, r|H_1)}{\prod_{i=1}^q p(d_i|H_0)} \quad (5.8)$$

Since the likelihood of the alternative hypothesis H_1 depends of an unknown parameter r , its maximum likelihood estimation is considered.

$$\Lambda(D) = \frac{\sup_r \left(\prod_{i=1}^{r-1} p(d_i|H_1) \prod_{i=r}^q p(d_i|H_1) \right)}{\prod_{i=1}^q p(d_i|H_0)} \quad (5.9)$$

As the likelihood of the null hypothesis H_0 does not depend on r and having $\prod_{i=1}^{r-1} p(d_i|H_1) = \prod_{i=1}^{r-1} p(d_i|H_0)$, the likelihood ratio can be simplified as follows:

$$\Lambda(D) = \sup_r \left(\frac{\prod_{i=r}^q p(d_i|H_1)}{\prod_{i=r}^q p(d_i|H_0)} \right) \quad (5.10)$$

Let δ denote the mean gap ($\delta = \mu_1 - \mu_0$). Under Gaussian assumption, one can get [74]:

$$\ln(\Lambda(D)) = \frac{\delta}{\sigma^2} \sup_r \left(\sum_{i=r}^q \left(d_i - \mu_0 - \frac{\delta}{2} \right) \right) \quad (5.11)$$

The decision of choosing either H_0 or H_1 is made by comparing the likelihood ratio with a threshold λ_Λ :

$$\begin{cases} H_0 : \ln(\Lambda(D)) < \ln(\lambda_\Lambda) \\ H_1 : \ln(\Lambda(D)) > \ln(\lambda_\Lambda) \end{cases} \quad (5.12)$$

For a real-time implementation, it is especially convenient to formulate the test sequentially. Let us then define the cumulative sum as in (5.13):

$$S_r^q(\mu_0, \delta) = \delta \sum_{i=r}^q \left(d_i - \mu_0 - \frac{\delta}{2} \right) \quad (5.13)$$

which can be re-written as:

$$S_1^q(\mu_0, \delta) = S_1^{r-1}(\mu_0, \delta) + S_r^q(\mu_0, \delta) \quad (5.14)$$

The decision rule then becomes:

$$\begin{cases} H_0 : S_1^q(\mu_0, \delta) - \inf_r (S_1^{r-1}(\mu_0, \delta)) < \gamma \\ H_1 : S_1^q(\mu_0, \delta) - \inf_r (S_1^{r-1}(\mu_0, \delta)) > \gamma \end{cases} \quad (5.15)$$

where $\gamma = \sigma^2 \ln(\lambda_\Lambda)$.

Let δ_m be the minimum value of δ which must be detected. The test is split into two sub-tests running in parallel, the first aiming at detecting a mean growth and the other a decrease of the mean.

Finally, at the current time k , a mean growth is detected as soon as (5.16) is true:

$$U_k - m_k > \gamma \quad (5.16)$$

where

$$U_k = U_{k-1} + d_k - \mu_0 - \frac{\delta}{2} \quad (5.17)$$

$$m_k = \min(m_{k-1}, U_k) \quad (5.18)$$

Conversely, a mean decrease is detected when:

$$M_k - T_k < \gamma \quad (5.19)$$

where

$$T_k = T_{k-1} + d_k - \mu_0 + \frac{\delta}{2} \quad (5.20)$$

$$M_k = \max(M_{k-1}, T_k) \quad (5.21)$$

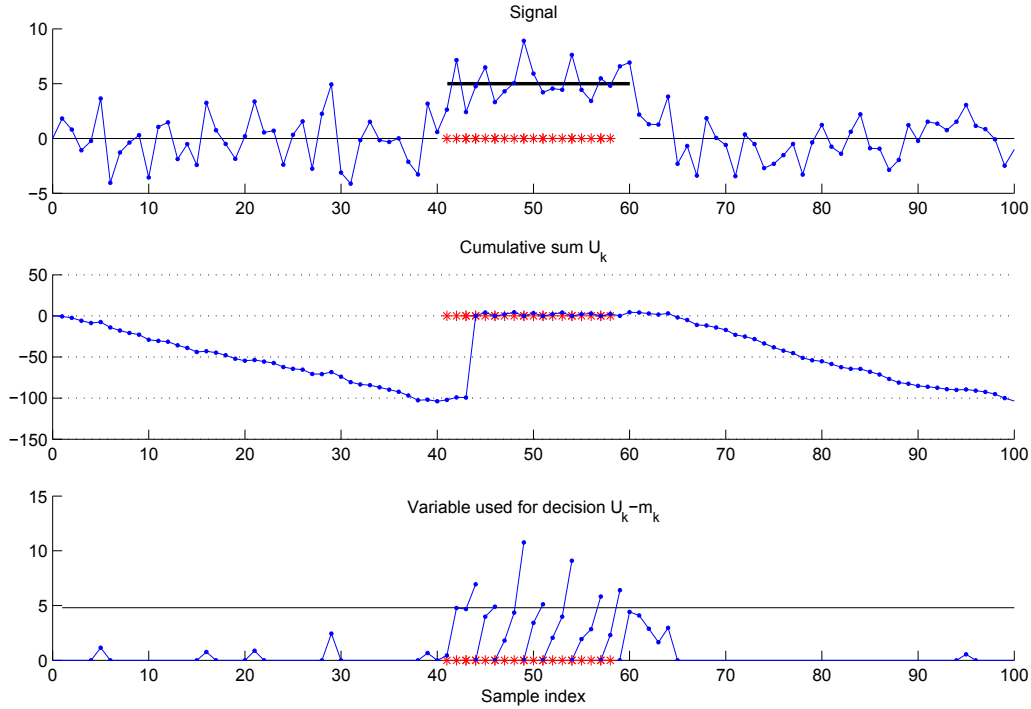


Figure 5.3: Illustrative example of Page's test. The signal to consider is plotted on the top graph. The corresponding cumulative sum and decision variable are beyond. The red stars denote the sample indexes for which the test detected a mean change.

As soon as threshold γ is reached, the cumulative sums are reset to zero. The actual mean change happened at the last time m (resp. M) has reached its minimum (resp. maximum) before crossing γ . Another formulation for the localization of the mean change which is used in this work is to find the last time the decision variable ($U_k - m_k$ for increase detection or $M_k - T_k$ for decrease detection) is null. This test is then very efficient in terms of required computational load. Indeed, at every new time step, it only requires to make additions and comparisons of scalar variables. The mean change cannot be detected at the samples prior to the last one at which the decision variable is null. Those samples are useless and can be removed from the computer memory.

The choice of γ has consequences on the false alarm probability. It is not possible to express it formally because the probability density function of hypothesis H_1 depends of an unknown parameter r . However, it can be set based on the number h of estimated parameters in the probability density function (PDF) and on the number n_σ of standard deviations: $\gamma = 2 \cdot h \cdot n_\sigma \cdot \sigma / \delta_m$ [74]. Here the mean is the only estimated parameter of the PDF and n_σ is set to 2 as nominal tuning which is a good compromise between false alarm rate and time to detection. Then $\gamma = 4 \cdot \sigma / \delta_m$

Let us consider the example shown in Figure 5.3. The signal plotted in the upper

graph of the figure is generated using a Gaussian function with a constant standard deviation of 2. The mean of this signal is 0 from sample 1 to sample 40 and from sample 61 to sample 100. The mean is set to 5 for sample indexes 41 to 60. Page's trend test is run using this signal as an input and the cumulative sum U_k as well as the decision variable $U_k - m_k$ are plotted in order to illustrate the behaviour of the test. A larger threshold than indicated previously is chosen in order to make the explanation more understandable; here $n_\sigma = 6$ thus $\gamma = 4.8$. When a sample of the signal deviates from zero and get close to 5 (at samples 5, 18, 21 for instance) the cumulative sum increases and thus the decision variable increases. In those situations a mean change of the signal is expected but not confirmed. Since the actual mean of the signal is 0 from sample 1 to 40, the cumulative sum keeps decreasing and the decision variable goes back to zero before crossing the threshold. The expectation of a mean change is therefore rejected. At the sample indexes 41 to 43, the cumulative sum and the decision variable increase like before but the decision variable finally crosses the threshold at index 44 which confirms the mean change. Samples 41 to 44 are therefore marked with red stars on the graph. The cumulative sum and the decision variable are set to zero which is indicated by discontinuities of the plots. The Page's test starts again: it detects at the sample 46 that a mean change occurred since the sample 44 and so on. In this example the test requires 3 or 4 samples to detect a mean change and missed the sample 59 and 60 because the decision variable was close to the threshold but did not reach it.

5.2.3 Experimental Evaluation of Fault Detection

5.2.3.1 Analysis Methodology

The critical issues of the fault detection are the reactivity and the ability to detect the fault when it occurs. It was shown in the previous section that the sequential formulation of Page's trend test resets depending on the value of the samples. It therefore has the particularity to constantly adapt the size of the sliding window. The reactivity of this test is not known before hand. In order to measure this, the test is evaluated in terms of distance-to-alert, distance-to-recovery and accuracy of map error localization, as shown in Figure 5.4. The distance-to-alert and the distance-to-recovery are derived from the usual time-to-alert and time-to-recovery and adapted in the context of this work in which the distance travelled by the vehicle is the reference. The distance-to-alert $\delta_{H_0 \rightarrow H_1}$ is the distance travelled by the vehicle before detecting a fault. Reciprocally, $\delta_{H_1 \rightarrow H_0}$ is the distance-to-recovery, that is, the distance after which the test detects the end of a fault. Since the test may detect a fault a few samples in the past, the *a posteriori* accuracy of the test is measured by $e_{H_1|H_0}$ and $e_{H_0|H_1}$. $e_{H_1|H_0}$ denotes the length of the road that has been identified as faulty while being actually fault-free. This can be named false alarm distance. Reciprocally, $e_{H_0|H_1}$ stands for the length of road that actually contains a fault that has not been detected by the algorithm which can be seen as a missed detection

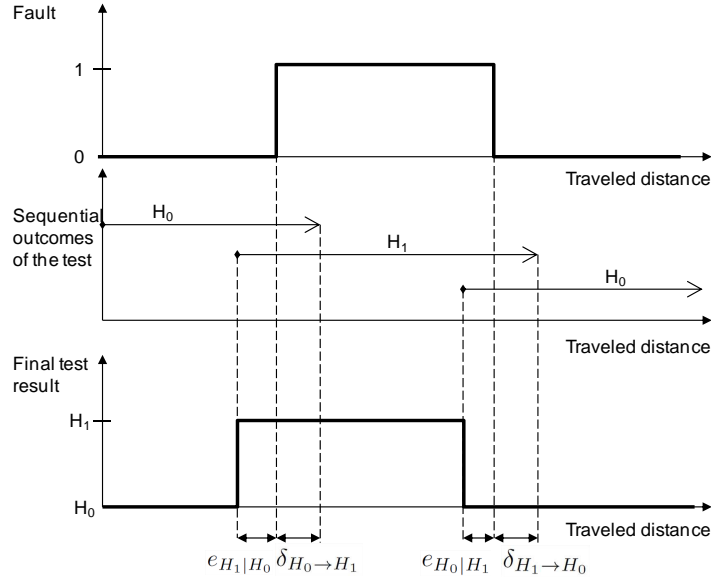


Figure 5.4: Definition of the metrics used for tests assessment based on a simple example. The fault to detect is represented on the upper part of the figure. As the test may go back in the past to localise a detected fault, the *a posteriori* outcomes of the test appear on the second line. The fault finally detected and localised appears on the bottom graph.

distance.

The performance of Page's trend test is compared to two other usual methods based on fixed length sliding window. These both aim to discriminate between H_0 and H_1 :

$$\begin{cases} H_0 : d_i = b_i, i = 1, \dots, q \\ H_1 : d_i = \delta_m + b_i, i = 1, \dots, q \end{cases} \quad (5.22)$$

where $b_i \sim \mathcal{N}(0, \sigma^2)$.

On the one hand, a simple decision rule based on the empirical mean of the sliding window was implemented:

$$\frac{1}{q} \sum_{i=1}^q d_i \underset{H_0}{\overset{H_1}{\gtrless}} \delta_m \quad (5.23)$$

On the other hand, the Neyman Pearson probabilistic decision rule was used for comparison. This is based on the generalised likelihood ratio of the hypotheses.

Under Gaussian noise assumption. The choice follows the rule (5.24) [14, 74]:

$$\sum_{i=1}^q d_i \underset{H_0}{\overset{H_1}{\geq}} \sigma \sqrt{2 \cdot q \cdot \log(\Phi)} \quad (5.24)$$

Where the threshold Φ arises from a compromise between desired false alarm (type I error) and missed detection probabilities (type II error) of the decision rule. The false alarm probability has been set to its usual value (0.1%) for an appropriate comparison with the proposed method.

The size q of the sliding window must be large enough to be statistically representative and short enough to detect map errors as fast as possible. Moreover, the samples D must be reinitialised as soon as the vehicle leaves one road for another which happens frequently in urban environment. It has then been set to $q = 20$ which is equivalent to approximately 200 m of travelled distance.

5.2.3.2 Experimental Evaluation

The tests were run on a set of roads that was recently modified due to the construction of a new motorway in Normandy, France (see Figure 5.5). This area is representative of typical geometric errors that a map may hold and which may cause severe malfunctions in driving assistance systems. In area 1 and 2, sharp bends were added to the road which was previously straight. This would make a curve warning system inefficient. In area 3 a new carriage way was added to the old single track road. This induces a constant lateral offset of the new road which would make obsolete intersection warning systems on crossing roads. On the fourth area, the lateral road offset decreases while the vehicle goes. This situation is very useful to highlight the distance to recovery of the tests.

Since the experiment is done in a rural environment and also for explanation convenience, it is assumed here that a fault always originates from an error in the navigation map. The actual strategy of fault isolation that allows to determine the fault originates whether from the estimate from navigation or from the estimate from sensors is employed in Section 5.3.

Tests have been run on the lateral euclidean distance d . The most restrictive constraint comes from intersection warning systems which require a longitudinal precision of 10 m for the placement of an intersection on a road link. Indeed, a lateral offset of a road link induces a longitudinal misplacement of intersection on crossing roads. This value has then been chosen as the mean change to detect $\delta_m = 10m$. Finally, the cumulative sum threshold is set dynamically to be $\gamma = 4 \cdot \sigma / \delta_m$ which represents a good compromise between false alarm rate and time to alert.

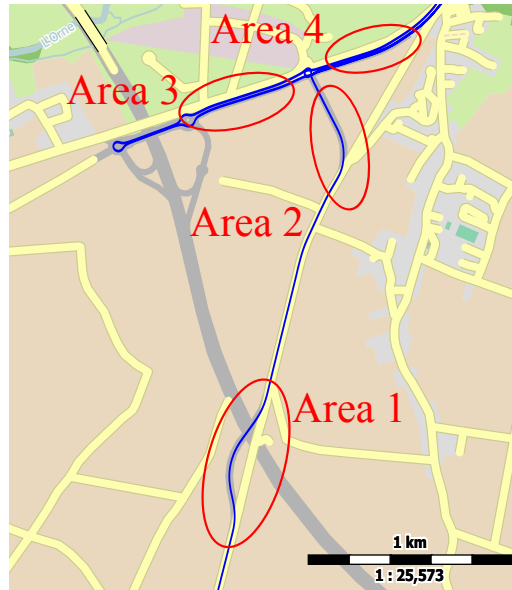


Figure 5.5: Global view of the test areas. The out-of-date map being assessed is shown by yellow lines. A map of the actual road network is at the background in grey. The vehicle trajectory is in blue (starting from the bottom of the figure). The four test zones are circled in red.

The results summarized in Table 5.1 show that Page's trend test is appropriate for detection and localization of faults. Indeed, faults are detected less than 20 m after the beginning of the fault and well localized. The two other methods provide less suitable false alarm and missed detection rates. This is mainly due to the fact that fixed length sliding windows are used. The fault detection has then undesirable collateral effects on the tail of the window. These methods show bad results when the road error is small with respect to the size of the sliding window.

Let us focus on area 1, to better understand the strength and weakness of the three methods. The upper part of Figure 5.6 shows the lateral error between the vehicle's estimated position and its map-matched position against the travelled distance. The lower part shows the sequential outcomes of each methods while the vehicle is driven. It can be seen on this figure that Page's test is very efficient for detecting the fault since it chooses H_1 as soon as the road is actually erroneous. Moreover, in this example, it locates perfectly the fault (from abscissa 520 to 1520 m). The outcomes of the two other tests are less accurate. Indeed, the faults are detected later and locate it very poorly. This is due to the fact that is not possible to know where the change happened within the sliding window. The whole window is supposed to belong to H_1 as soon as the threshold is crossed. Correct road points are then declared faulty while they are not and *vice versa*. This example illustrates why these methods induce false alarms, missed detections and inaccuracies in fault localization.

Table 5.1: Comparative results with nominal tuning of each test

	Test	$\delta_{H_0 \rightarrow H_1}$	$\delta_{H_1 \rightarrow H_0}$	$e_{H_0 H_1}$	$e_{H_1 H_0}$
Area 1	Page	0 m	0 m	0 m	0 m
	Mean	73 m	150 m	90 m	177 m
	N.P.	46 m	255 m	220 m	0 m
Area 2	Page	20 m	n.a. ³	20 m	0 m
	Mean	n.d. ⁴	n.a.	35 m	0 m
	N.P.	80 m	n.a.	0 m	170 m
Area 3	Page	0 m	0 m	0 m	0 m
	Mean	50 m	100 m	150 m	200 m
	N.P.	0 m	300 m	0 m	300 m
Area 4	Page	0 m	20 m	0 m	20 m
	Mean	50 m	0 m	250 m	40 m
	N.P.	0 m	250 m	0 m	210 m

³ not applicable

⁴ not detected

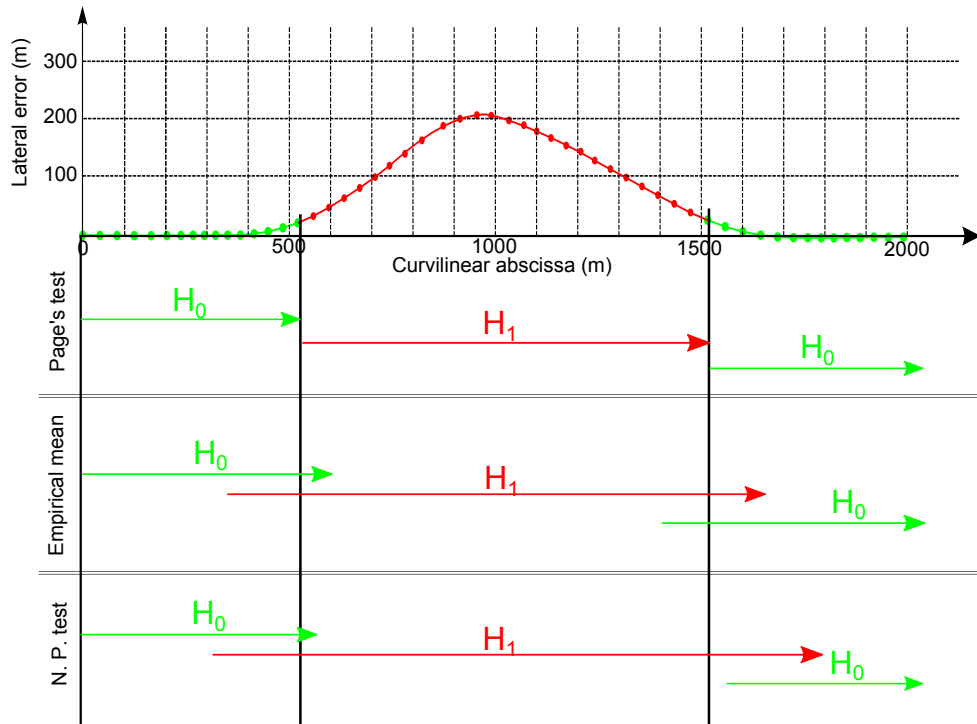


Figure 5.6: Sequential outcomes of three trend tests on hypotheses H_0 (correct road segment) and H_1 (erroneous road segment). The lateral distance to the map-matched position is denoted on the upper part. The plot colour shows the true state of the road: correct in green, with fault in red.

5.3 FDIA with Page's Trend Test

Section 5.2.2 has presented a sequential test to detect a mean gap of a random variable. This test showed its relevance for detecting a discrepancy between the estimates provided by sensors and by the navigation function. Such a discrepancy can be interpreted as a presence of a fault in one (or both) of the two estimates. However, it is not possible to determine which one. Faults can be detected but not isolated. This section proposes to use the FDIA framework introduced in Chapter 4 to take benefit from the past vehicle trips and therefore perform fault isolation.

5.3.1 Method

The FDIA framework introduced in the previous chapter is based on the calculation of a residual vector R_K^s (where s is the vehicle curvilinear abscissa on the road and K is the number of trips on this road recorded by the method). The elements of R_K^s were defined as the result of comparisons of the distance between every pair of available estimates N and G with a threshold λ_d and denoted by $r_{G_i^s G_j^s}$, $r_{G_i^s N_j^s}$ and $r_{N_i^s N_j^s}$, $\forall i, j \in \{1, \dots, K\}$ which are recalled hereafter:

$$r_{G_i^s G_j^s} = \begin{cases} 1 & \text{if } \text{dist}(G_i^s, G_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (5.25)$$

$$r_{G_i^s N_j^s} = \begin{cases} 1 & \text{if } \text{dist}(G_i^s, N_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\} \quad (5.26)$$

$$r_{N_i^s N_j^s} = \begin{cases} 1 & \text{if } \text{dist}(N_i^s, N_j^s) > \lambda_d \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in \{1, \dots, K\}, i > j \quad (5.27)$$

However, it was shown that the estimate from sensor G can be different from the estimate from navigation N due to the usual noise that affect the GNSS receiver. The idea developed here is to use Page's trend test instead of the distance measurement for comparing G and N , as shown in Figure 5.1. According to this new formulation, the residual vector element $r_{G_K^s N_K^s}$ equals zero if the Page's test states that the mean of the signal d is zero. Reciprocally, $r_{G_K^s N_K^s}$ equals one if the test detects a mean change in d . The manner in which the other residual elements ($r_{G_i^s G_j^s}$ and $r_{N_i^s N_j^s}$) are calculated remains unchanged.

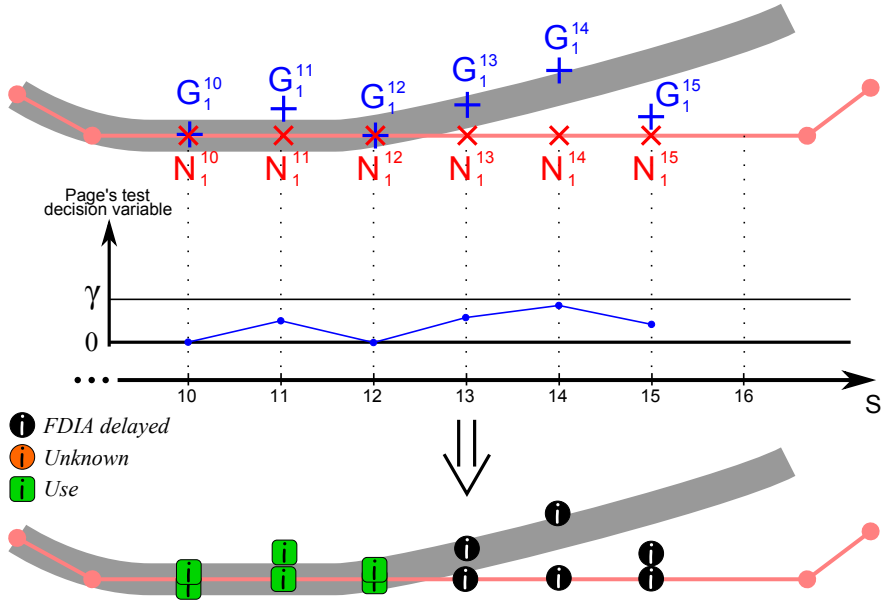
As shown in Section 5.2.2, the Page's test may require a few samples to settle. This was highlighted by the metrics distance-to-alert and distance-to-recovery. In such

a situation, the estimates are buffered until the Page's test provides a definitive output. The FDIA framework is then run on each estimate taking as an input the result of the test.

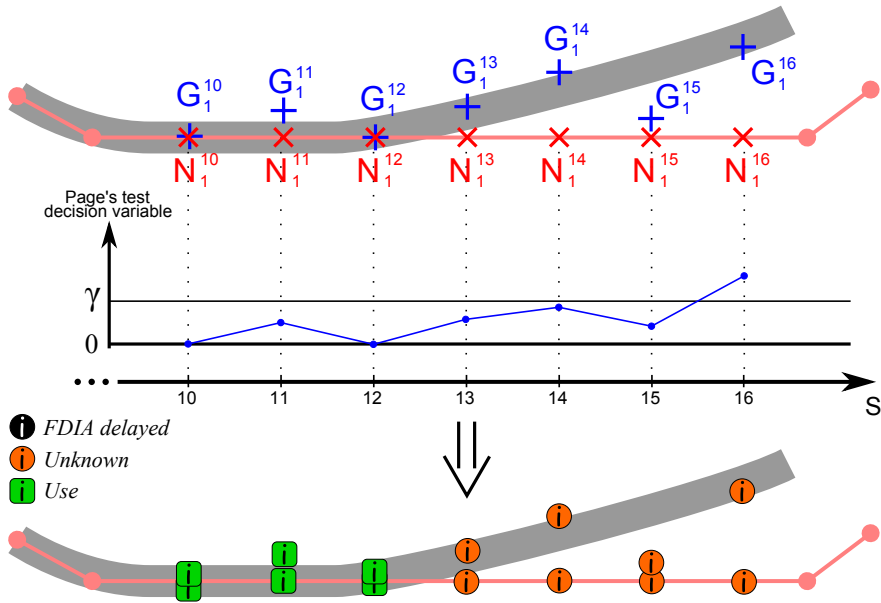
As an example, Figure 5.7 shows the evolution of the Page's test and illustrates the strategy employed here for the FDIA. The first trip ($K = 1$) on this road is considered. The method is detailed step by step, based on this figure. When the vehicle is at abscissa $s = 10$, the estimates from sensors G_1^{10} and N_1^{10} are very close to each other. The Page's test decision variable equals zero therefore the estimates are said equal. The FDIA framework is run with $r_{G_1^{10}N_1^{10}} = 0$: this residual is found only once in the truth table so it is not necessary to use previous trip and the FDIA concludes that no fault affects the estimates (neither from navigation N_1^{10} nor from sensors G_1^{10}). At abscissa $s = 11$, the Page's test decision variable is not null but has not reached the threshold γ . A discrepancy between the estimates G and N is likely but still not detected. The estimates that correspond to abscissa $s = 11$ (i.e. G_1^{11} and N_1^{11}) are stored in the buffer memory until the Page's decision variable either crosses the threshold or goes back to zero. When the vehicle reaches abscissa $s = 12$, the estimates G_1^{12} and N_1^{12} are such that the decision variable goes back to zero which indicates that the Page's test judges that there is no discrepancy for the two last abscissas (since the last abscissa at which the decision variable is null). The FDIA is run at every abscissa in the buffer memory:

- At $s = 11$ and given $r_{G_1^{11}N_1^{11}} = 0$, the FDIA concludes that there is no fault affecting G_1^{11} and N_1^{11}
- At $s = 12$ and given $r_{G_1^{12}N_1^{12}} = 0$, the FDIA concludes that there is no fault affecting G_1^{12} and N_1^{12} and the buffer memory is emptied.
- At $s = 13$ to $s = 15$, the estimates are such as the decision variable is not null so these are buffered as shown by black round marks on Figure 5.7a.
- At abscissa $s = 16$ the decision variable finally crosses γ so the Page's test finally states that a discrepancy is detected and started at $s = 13$.
- The FDIA is then successively run at $s = 13$, $s = 14$, $s = 15$ and $s = 16$ with $r_{G_1^{13}N_1^{13}} = 1$, $r_{G_1^{14}N_1^{14}} = 1$, $r_{G_1^{15}N_1^{15}} = 1$ and $r_{G_1^{16}N_1^{16}} = 1$ respectively. Since these residuals are adverse, the FDIA uses past trips to perform isolation and its final diagnostic depends on the past estimates at each abscissa as shown by Figure 5.7b. The buffer memory is emptied and the decision variable is set to zero for the following evaluation points.

It can be noticed in this example that, at $s = 11$, the Page's test avoided a detection of a fault where the estimate from sensor was affected by a usual noise. Moreover, at $s = 15$, the FDIA only would have evaluated the estimates as fault free and return the output *Use* for a faulty navigation estimate. The Page's test avoided a False Validation at this abscissa.



(a) The vehicle is at abscissa $s = 15$. Page's decision variable is greater than 0 and lower than the threshold. The FDIA is then delayed since $s = 13$.



(b) The Page's decision variable crosses the threshold at $s = 16$. FDIA is run for abscissas 13 to 16 with $r_{G_1 N_1} = 1$. This results in *Knowledge of fault: Unknown* in this example.

Figure 5.7: Example of the use of Page's test with the FDIA framework. The road as recorded in the navigation map is the red poly-line. The estimates from navigation N_1^s are the red crosses and the estimates from sensors G_1^s are the blue crosses. The vehicle goes on this road from left to right so its curvilinear abscissa is denoted by the axis S . The decision variable used in the Page's test is plotted on the middle graph. The result of FDIA with Page's test is shown on the lower parts of the figures.

5.3.2 Experimental Results in the Urban Scenario

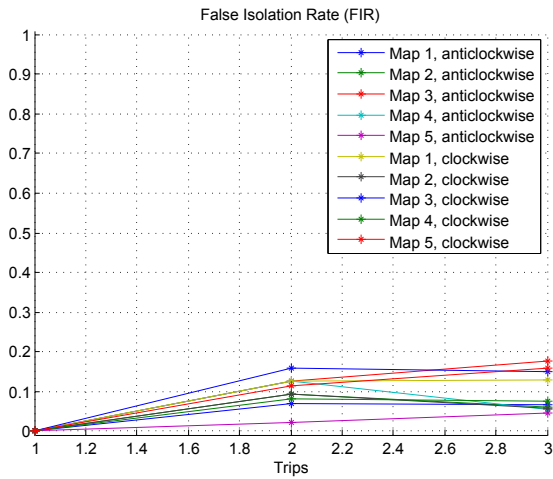
The influence of the Page's test on the fault detection and isolation is shown using the same data and navigation maps as in Section 4.7.3. Page's trend test is set to detect a discrepancy of $\delta_m = 2$ m between the estimates with the detection threshold $\gamma = 4\sigma/\delta_m$. In the FDIA functions, the parameters are set as follows: the spatial sampling of integrity monitoring is 10 m, the tolerance on the vehicle curvilinear abscissa is $\lambda_s = 1$ m, the distance threshold used to compare estimates from navigation is $\lambda_d = \delta_m = 2$ m.

The performance of the FDIA method coupled with the Page's test for fault detection is evaluated using the metrics introduced in Section 4.7.1 and detailed by Figures 5.8 and 5.9.

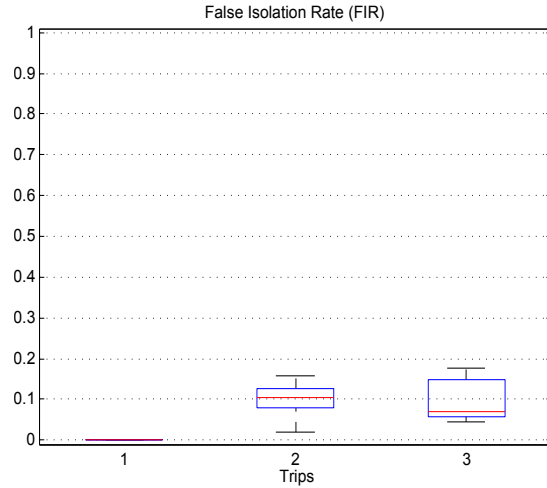
Figures 5.8a and 5.8a show the proportion of correct points from the navigation map that are isolated as faulty by the method. Since the method cannot perform isolation at the first trip, the FIR equals zero at the first trip for the ten tests. The FIR at the second trip are between 2.1% and 16% with the median equal to 10%. At the third trip, the FIR of maps 3 and 5 anticlockwise as well as map 5 clockwise increases. The others remain constant or decrease. The median decreases at the third trip with 7.1% but the range is shifted upwards since the FIR is between 4.5% and 18%. These figures are greater than in the experiment without the Page's test. The reason originates from the manner in which this metric is generated. The performance calculation algorithm used here computes the distance between each estimate recorded by the FDIA function and the road in the correct navigation map. If this distance is lower than the threshold λ_d used in the method, this estimate is said correct. The FIR depicts the proportion of such correct estimates that were isolated as faulty by the FDIA method. In the area where a large noise affects the estimates, the Page's test takes a few samples to settle. If the test finally concludes that a mean gap affected the samples, some of these, for which the distance taken individually is below the threshold, would generate a FI.

Figures 5.8c and 5.8d depict the ratio of faulty map points that have been said fault-free by the method. Four tests keep a FVR equal to zero for the three trips (maps 2 and 4 anticlockwise and maps 3 and 5 clockwise). One test (map 1 anticlockwise) presents a notably high FVR at the second and third trips with 20% and 13% respectively. The box plot shows that in general the FVR tends to decrease at the third trip since the median goes from 4.1% to 2.7%. The comparison of the FDIA with and without the Page's test shows a significant decrease of the FVR when using the Page's test. It also permits to cancel the upward trend of the median at the third trip.

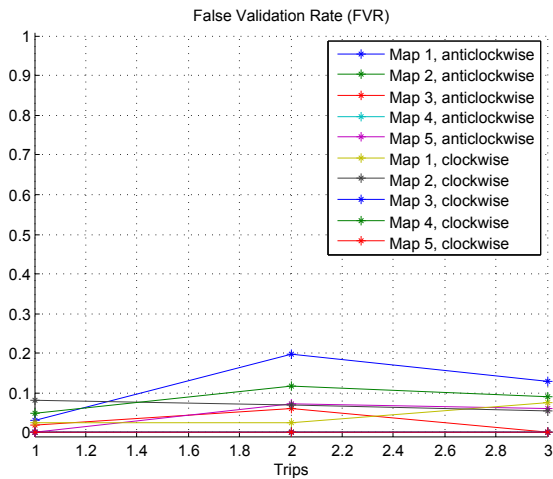
Figures 5.8e and 5.8f show ratio of correctly identified points to the number of isolated or validated points. As detailed in Section 4.7.3, the OER at the first trip is favoured by the absence of false validation; the OER of five of the ten tests therefore



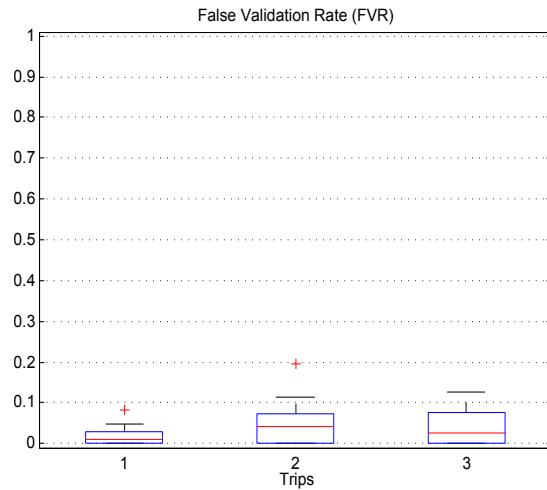
(a) False Isolation Rate



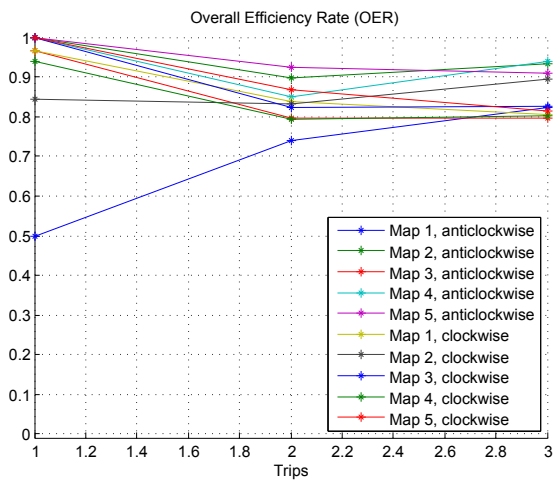
(b) Box plot of FIR.



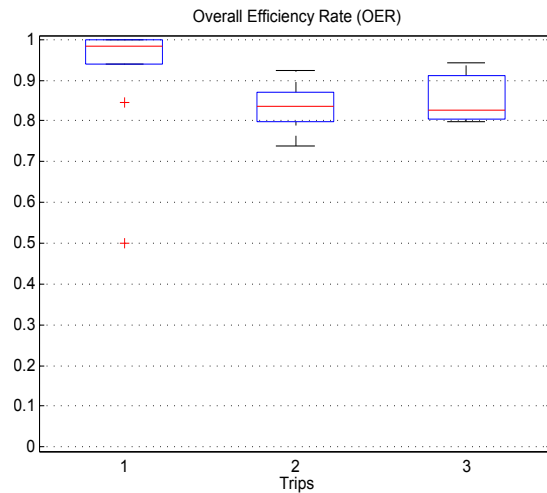
(c) False Validation Rate.



(d) Box plot of FVR.

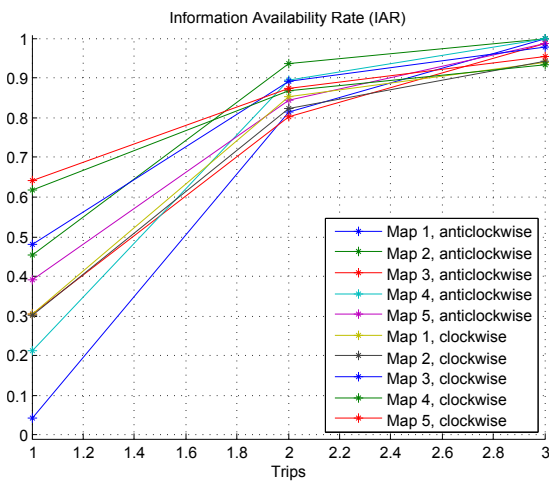


(e) Overall Efficiency Rate.

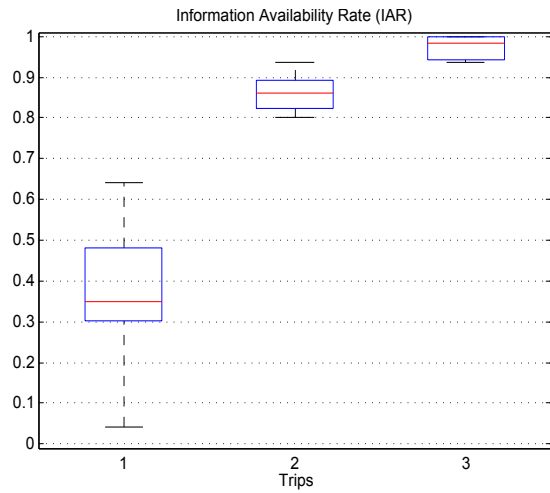


(f) Box plot of OER.

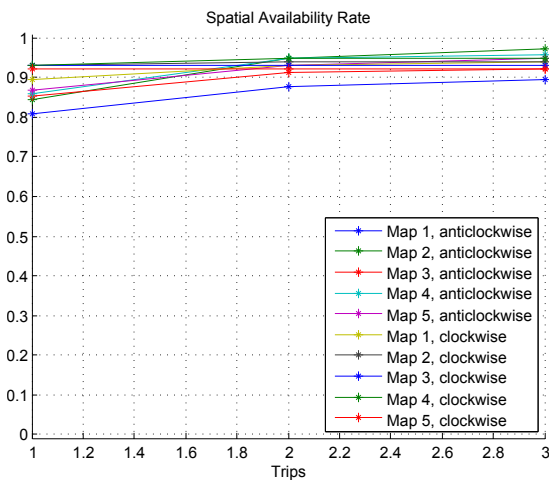
Figure 5.8: False isolation rate, false validation rate and overall efficiency in urban test track



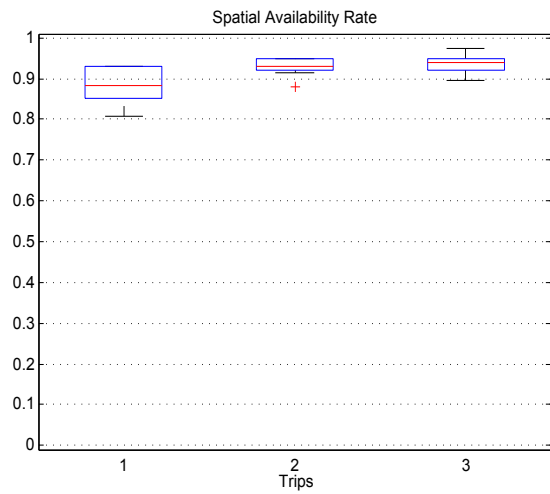
(a) Information Availability Rate.



(b) Box plot of IAR.



(c) Spatial Availability Rate.



(d) Box plot of SAR.

Figure 5.9: Information and spatial availabilities in the urban test track

equal one. It can be noticed that the OER of map 1 anticlockwise is especially low at the first trip (50%), however it is not significant because it is calculated using only four points. The OER tends to remain constant from the second to the third trip with medians equal to 84% and 83% respectively. The OER with and without the Page's test are substantially identical since the grow of FIR is balanced by the decrease of FVR.

Figures 5.9a and 5.9b summarise the ratio between the number of validated or isolated points and the number of points said *Unknown*. The IAR increases with the number of trip for all the tests and exceeds 90% at the third trip. The convergence of the FDIA method is therefore observed when evaluating the FIDA with the Page's test however the IAR is slightly lower than without the Page's test. The reason is the same than the one introduced in the description of Figures 5.8a and 5.8b. Some points taken individually are close enough to be considered as equal and judged fault-free by the FDIA framework. When using Page's trend test, these points are finally said different because of the other points in the test buffer. In this situation the FDIA possibly associates to these points the output *Unknown* which reduces the IAR.

Figures 5.9c and 5.9d finally show the proportion of evaluated points with the number of trips. All tests together, the SAR exceeds 80% and increases at each new trip. A very small decrease of 1% can be observed on the minimum and maximum of SAR when using the Page's test compared to the SAR calculated without the Page's test. This is due to the few sample which may not be evaluated at the end of the recording. Apart from this, the results are the same. This highlight the fact that the SA mainly depends on the ability of the sensors to provide spatially sampled data.

As a conclusion, it can be noticed that the use of the Page's test allows to improve the performance of the FDIA method in terms of false validation which is a significant advantage for the integrity monitoring application in which this work takes place. This test makes the FIDA more cautious from the applicative point of view since it reduces the number of times the method wrongly provides the output *Use* to the client applications but slightly increases the number of unjustified *Don't use*.

5.4 Conclusion

This chapter presented Page's trend test applied to the fault detection of navigation systems. This statistical test is used in the literature to detect a change in the mean of a random variable affected by a large noise with respect to the change that is investigated. It was adapted here in order to evaluate the similarity between the estimates of the vehicle position provided by the navigation system and by the sensors despite the noise affecting the estimate from sensors. Indeed the sensors used in this work are the standard vehicle sensors.

The mathematical formulation of this test and its application to the comparison of vehicle position estimates was detailed. Given this, its ability to detect a discrepancy between the estimate from sensors and the estimate from navigation was evaluated using real vehicle data. Since the metrics usually employed to measure the performance of this test are not relevant in the intelligent vehicle context, the evaluation was done using metrics introduced in this chapter. The test showed convincing performance since it detected quickly (short distance-to-detection and distance-to-recovery) and accurately (short false alarm and missed detection distances) the discrepancies between the estimates.

Page's trend test allows to detect with robustness when the two estimates of the vehicle position are significantly different and to conclude that a fault affects at least one of them. However, it does not permit to determine which one is affected by the fault. In this sense, this test performs fault detection but does not perform isolation. The second part of this chapter showed how to use the FDIA formalism introduced in the previous chapter in combination with the Page's test to fill this gap.

The performance of this complete framework was evaluated based the same data and using the same metrics as in the previous chapter in order to highlight the efficiency of the Page's test on the ability of the FDIA framework to accurately isolate faults of navigation system. It was shown that the Page's test makes the FDIA more cautious, that is, significantly decreases the rate of wrongly validated navigation estimates but slightly increases the rate of wrongly isolated navigation estimates. This positive effect, from the point of view of the intelligent vehicle applications, justifies the use of Page's trend test in the integrity monitoring system presented in this work.

Chapter 6

Conclusions

Contents

6.1	Synthesis	125
6.2	Perspectives	127

6.1 Synthesis

Navigation maps are increasingly used on board vehicles as the applications are transformed from simple driving assistance systems to computer actuated vehicles. However, navigation maps cover extensive and constantly evolving road networks. They are therefore often out-of-date and include faults.

The research described in this thesis proposed a framework to detect faults in navigation maps. The main singularity of the approach was to be autonomous and to rely solely on the use of vehicle proprioceptive sensors and GNSS receiver available as parts of ADAS. The purpose was to reduce and/or eliminate faults in the contextual data fed by the *navigation function* to client systems like ADAS or autonomous driving functions. The approach was based on the assumption that passenger vehicles are most of the time driven in known areas, that is, on roads on which the vehicle has already been driven. The principle was therefore to take benefit from the repetition of trips as a source of redundancy.

The underlying theory and principles required for the detection and isolation of faults in navigation maps were developed. For this purpose, the techniques needed for the basic components of the navigation function (Localization, Map-Matching and Navigation Maps) were established. The principles of vehicle localization were detailed through the description of coordinates systems, GNSS positioning and methods aiming at improving accuracy and availability by fusing GNSS with vehicle proprioceptive sensors. The study of the map-matching function allowed to determine its weaknesses in case of faults in the navigation map. It was shown that this function is likely to provide erroneous data even with a high confidence index. A pathology

of navigation map faults was established and classified into structure, geometry and attribute faults. The first two were addressed in this work.

The detection of roundabouts replacing conventional crossroads was achieved by looking at the vehicle trajectory. It uses proprioceptive sensors on a fixed length window expressed in the vehicle reference frame. A set of fifteen patterns was manually learnt representing the variety of trajectories that either include or do not include a roundabout. A Bayesian classifier is then applied onto the estimated trajectory. If the trajectory is classified as containing a roundabout, its radius and the coordinates of its centre are found through the clustering of the successive rotation centres. The findings are geo-localized and stored in an auxiliary map in order to warn the ADAS the next time the vehicle goes in the area. This was validated experimentally on real traffic conditions with success.

The study of the system integrity monitoring, showed that most of the methods addressing fault detection and isolation are developed for linear or linearisable systems. They therefore require system models of various degrees of granularity. Since navigation maps are difficult to model, a novel approach based was developed. It was based on a separate estimation of the vehicle position, independent of the navigation map. This was compared with the output from the navigation function. The principle was to compare these estimates and to assume that these were equal only if no fault affected them. With this formulation, a disparity between the estimates indicated at least one fault among the estimates but an ambiguity on the identification of the faulty estimate(s) remained. A FDIA (Fault Detection, Isolation and Adaptation) formalism was developed in order to solve this ambiguity based on the redundancy obtained by using the past vehicle trips on the same location. Several properties were theoretically demonstrated. The convergence of isolability property showed that the isolation capabilities are improved as the number of trips increases. It was demonstrated that the method is able to provide a correction in case of fault on navigation function. It was also demonstrated that this formalism anticipates its ability to correct faults. This allows the client system to adopt the proper strategy when approaching a road on which the vehicle has already travelled.

The proposed FDIA formalism was validated using data acquired in real traffic conditions and a road network that was randomly modified to introduce random geometric faults. The isolation convergence property has been verified by these tests. Further, the method can correctly identify most of the faulty estimates. Some faults were wrongly identified due to the noise in the data acquired by the vehicle sensors and the spatially-sampled strategy.

Page's trend test was used to reduce the sensitivity to the FDIA framework to noisy data. This statistical test aims at detecting changes of the mean of a random variable. Its specificity resides on its sequential formulation which is very convenient for real-time applications. It allows to detect but not isolate faults. The combined use of this test and the FDIA framework has been proposed and studied. The experimental evaluation based on the same data showed a clear improvement of

Page’s test on the FDIA performance.

6.2 Perspectives

Complementary Experimental Evaluation

To evaluate the proposed approach, several software tools were developed: a navigation map that includes sufficient variety of geometric changes for testing purposes, a navigation function that generates the EH, the implementation of the FDIA framework, data acquisition from vehicle proprioceptive sensors and the GNSS receiver for use with the map, the comparison and analysis of the results. These took a substantial effort. Whilst much testing has been done, this could be extended to longer trajectories, different road configurations, etc. Further, the results could be associated to existing contextual ADAS (e.g., Contextual ACC) to compare the performance gains of such functions with the use of the Integrity Monitoring component.

Experimental Comparison with other approaches

The proposed FDIA method can be compared with other autonomous approaches based on different assumptions. For instance, the assumption on the fact that the errors on the navigation system are not systematic can be relaxed which conduct to simplifications in the formalization of the method. A simplified version of the formalism based on the sole comparison of pairs of estimates of the vehicle position from sensors could be designed. Since this approach relies on less assumptions, it would be robust to road changes between two vehicle trips for example. However, it is expected to require a large number of vehicle trips to converge (i.e. to perform Isolation and Adaptation).

Integrity Monitoring Horizon

The method used in this work aimed to monitor the integrity of the navigation system at the vehicle current position. The current vehicle position was estimated, on the one hand, by the navigation function and on the other hand with vehicle sensors. The integrity of navigation estimate was deduced from the comparison of these estimates. A promising approach would be to extend this integrity monitoring to future vehicle positions. On the one hand, the navigation function provides, through the electronic horizon, the future road geometry. On the other hand, the future vehicle position could be predicted with a kinematic vehicle model to make the comparison. The vehicle proprioceptive sensors would be used for this task but these would permit to compute a position estimate only a few metres ahead.

Moreover, under the assumption that a smart front camera is surely available on board the vehicle, the vehicle position could be estimated on a larger distance based on lane markings detection.

Adaptation for Autonomous Vehicles

Over the last two years the automotive industry has shown a strong interest on computer controlled vehicles, namely autonomous vehicles. Efforts are consequently made on lane-level navigation maps that includes precise description of every road lane and associated features (e.g., lane markings, traffic signs, kerbs). Most of the techniques developed as part of this thesis are consistent with the use of enhanced maps in autonomous vehicles. A major extension of this research would be to apply it to the correction and completion of such maps using the FDIA framework. Integrity monitoring is a major feature needed for the safe deployment of autonomous vehicles. This has provided over the past months a different perspective on the potential of this work and have influenced the future directions that this could take.

Map Building and Continuous Learning

In case of large geometric faults in the navigation map, the confidence in the map-matched vehicle position provided by usual navigation function decreases until the navigation function finally switches to *off-road* mode. The vehicle is no longer associated to any road. This situation is easily detectable and was therefore not addressed in this work. An extension of the proposed approach would be to create local maps that represent the missing roads. The repeated vehicle trips on the same roads would be used to enrich and refine these new maps using approaches similar to [7]. Monitoring the refinement level of the new maps would permit to finally include them into the navigation map and permit client systems to work properly. This would also require to adapt the current approach to concurrent mapping and localization techniques (SLAM). An extension of this approach could be the application of these techniques to build maps in areas where no map exists like in rural areas of emerging countries.

The embedded editable vehicle memory used in this thesis would permit the vehicle to constantly learn from its environment. An application for this could be related to the improvement of object tracking algorithms that requires information on the type of objects that are likely to be tracked [70]. The navigation map can be used in this kind of approaches in order to determine the type of road and the context (rural, urban,...) of the vehicle to adapt the tracking strategy. For instance, a pedestrian is more likely to appear in front of the vehicle on a street close to a primary school than on a rural motorway. These parameters can be deduced from static information recorded in the map. A more sophisticated approach made possible by this work should be to continuously refine these parameters by learning at every trip of the vehicle in each area.

Information Sharing, Collaborative Vehicles

The assumption of Vehicle to Vehicle (V2V) or to Infrastructure (V2I) communication capabilities was not made in this work. The redundancy required for FDIA was provided by the repetition of trips by the subject vehicle. However, it could also be obtained by sharing the data of several vehicles. In this way, the subject vehicle could benefit from recent information on the navigation map integrity and from the correction to apply, prior to go for the first time in an area. The research issues to be addressed include data security, optimisation of communications links (i.e. the size of the information files shared), centralisation of information in the cloud and the extension to other sources of information.

Bibliography

- [1] Crisismappers project homepage: <http://crisismappers.net/> accessed 01 april 2014.
- [2] Openstreetmap project homepage: <http://www.openstreetmap.org/> accessed 30 january 2012.
- [3] Road network description xml format specification.
- [4] Spatialite project homepage: <https://www.gaia-gis.it/fossil/libspatialite/index> accessed in june 2014.
- [5] Ushahidi project homepage: <http://ushahidi.com/> accessed 01 april 2014.
- [6] Traffic accident causation in europe project (trace). <http://www.trace-project.org>, 2008. Accessed 30 January 2012.
- [7] G. Agamennoni, J.I. Nieto, and E.M. Nebot. Robust and accurate road map inference. In *Robotics and Automation (ICRA)*, pages 3946–3953, may 2010.
- [8] N. Arica and F.T. Yarman-Vural. Optical character recognition for cursive handwriting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):801–813, jun 2002.
- [9] A. Armand, D. Filliat, and J. Ibañez-Guzmán. Modelling stop intersection approaches using gaussian processes. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems-ITSC*, 2013.
- [10] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, and S. W. Williams. The landscape of parallel computing research: A view from berkeley. Technical report, Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.
- [11] T. Badard and C. Lemarie. Associer des données, l'appariement. In *Généralisation et représentation multiple*, 2002.
- [12] C. Bahlmann, M. Pellkofer, J. Giebel, and G. Baratoff. Multi-modal speed limit assistants: Combining camera and gps maps. In *Intelligent Vehicles Symposium, IEEE*, pages 132–137, June 2008.
- [13] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

BIBLIOGRAPHY

- [14] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc, 1993.
- [15] N. Belkherchi. *Contribution a l'etude du diagnostic et de la commande tolérante aux fautes par l'approche structurelle. Application aux procedes biologiques*. PhD thesis, Universite Paul Sabatier - Toulouse III, 2011.
- [16] M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, S. Debattisti, R. I. Fedriga, M. Felisa, L. Gatti, A. Giacomazzo, P. Grisleri, M. C. Laghi, L. Mazzei, P. Medici, M. Panciroli, P. P. Porta, and P. Zani. The vislab intercontinental autonomous challenge: 13,000 km, 3 months, no driver. In *World Congress on ITS*, Busan, South Korea, October 2010.
- [17] D. Betaille and R. Toledo-Moreo. Creating enhanced maps for lane-level vehicle navigation. *Intelligent Transportation Systems, IEEE Transactions on*, 11(4):786–798, dec. 2010.
- [18] K. M. Bin Abdl and S. Z. M. Hashim. Handwriting identification: a direction review. In *Signal and Image Processing Applications (ICSIPA)*, pages 459–463, nov. 2009.
- [19] Ph. Bonnifait, P. Bouron, P. Crubille, and D. Meizel. Data fusion of four abs sensors and gps for an enhanced localization of car-like vehicles. In *Robotics and Automation, ICRA. IEEE International Conference on*, volume 2, pages 1597–1602 vol.2, 2001.
- [20] J. Borenstein and Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. *Robotics and Automation, IEEE Transactions on*, 12(6):869–880, Dec 1996.
- [21] K. Boriboonsomsin, M. J. Barth, W.eihua Zhu, and A. Vu. Eco-routing navigation system based on multisource historical and real-time traffic information. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1694–1704, 2012.
- [22] C. Boucher and J.-C. Noyer. Automatic detection of topological changes for digital road map updating. *Instrumentation and Measurement, IEEE Transactions on*, 61(11):3094–3102, Nov 2012.
- [23] A. K. Brown. Receiver autonomous integrity monitoring using a 24-satellite gps constellation. In *Institute of Navigation, Technical Meeting*, volume 1, pages 256–262, 1987.
- [24] D. J. Buckley. The gis primer an introduction to geographic information systems. Technical report, Innovative, 1997.
- [25] H. Bunke. Recognition of cursive roman handwriting: past, present and future. In *Document Analysis and Recognition. International Conference on*, pages 448 – 459 vol.1, aug. 2003.

- [26] DARPA Urban Challenge. Route network definition file (rndf) and mission data file (mdf) formats. Technical report, Defense Advanced Research Projects Agency, 2007.
- [27] A. Chen, A. Ramanandan, and J. A. Farrell. High-precision lane-level road map building for vehicle navigation. In *Position Location and Navigation Symposium (PLANS), IEEE/ION*, 2010.
- [28] L. Chen, M. Lv, Q. Ye, G. Chen, and J. Woodward. A personal route prediction system based on trajectory data mining. *Information Sciences*, 181(7):1264–1284, 2011.
- [29] M. Chowdhary. Driver assistance applications based on automotive navigation system infrastructure. In *Consumer Electronics, Digest of Technical Papers. International Conference*, pages 38 – 39, 2002.
- [30] J. Craig. Map data for adas. In *Handbook of Intelligent Vehicles*, pages 881–892. Springer, 2012.
- [31] J. Daniel and J.-P. Lauffenburger. Conflict management in multi-sensor dempster-shafer fusion for speed limit determination. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 987–992, June 2011.
- [32] N. de Freitas, R. Dearden, F. Hutter, R. Morales-Menendez, J. Mutch, and D. Poole. Diagnosis by a waiter and a mars explorer. *Proceedings of the IEEE*, 92(3):455 – 468, mar 2004.
- [33] M. E. de Gunst and J. E. den Hartog. Knowledge-based updating of maps by interpretation of aerial images. In *Pattern Recognition, Computer Vision Image Processing., Proceedings of the 12th IAPR*, 1994.
- [34] L. Deren, S. Haigang, and X. Ping. Automatic change detection of geo-spatial data from imagery. *Geo-Spatial Information Science*, 6:1–7, 2003.
- [35] T. Devogele. *Processus d’intégration et d’appariement de bases de données géographiques; application à une base de données routières multi-échelles*. PhD thesis, Université de Versailles-Saint Quentin en Yvelines, 1997.
- [36] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
- [37] S. X. Ding. *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.
- [38] S. P. Drake. Converting gps coordinates [phi, lambda, h] to navigation coordinates (enu). 2002.
- [39] V. Drevelle and Ph. Bonnifait. Robust positioning using relaxed constraint-propagation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 4843–4848, 2010.
- [40] S. Durekovic and N. Smith. Architectures of map-supported adas. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 207 –211, june 2011.

BIBLIOGRAPHY

- [41] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.
- [42] F. Ekpenyong, D. Palmer-Brown, and A. Brimicombe. Updating of road network databases: Spatio-temporal trajectory grouping using snap-drift neural network.
- [43] M.ENZWEILER and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2179–2195, 2009.
- [44] A. Eskandarian. *Handbook of Intelligent Vehicles*. Number vol. 2 in Handbook of Intelligent Vehicles. Springer, 2012.
- [45] ESRI. Esri shapefile technical description. Technical report, ESRI, 1998.
- [46] R. Ewing and E. Dumbaugh. The built environment and traffic safety: A review of empirical evidence. *Journal of Planning Literature*, 23(4):347–367, May 2009.
- [47] N. Fairfield and C. Urmson. Traffic light mapping and detection. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 5421–5426, May 2011.
- [48] W. E. Featherstone and S. J. Claessens. Closed-form transformation between geodetic and ellipsoidal coordinates. *Studia geophysica et geodaetica*, 52:1–18, 2008.
- [49] C. Fouque and Ph. Bonnifait. Multi-hypothesis map-matching on 3d navigable maps using raw gps measurements. In *Intelligent Transportation Systems, IEEE International Conference on*, pages 1498–1503, Madeira Island Portugal, 09 2010.
- [50] J. Froehlich and J. Krumm. Route prediction from trip observations. Technical report, SAE Technical Paper, 2008.
- [51] P.-Y. Gilliéron, H. Gontran, and B. Merminod. Cartographie routière précise pour les systèmes d’assistance à la conduite. In *Proceedings of the GIS-SIT Conference*, number TOPO-CONF-2006-015, 2006.
- [52] P. Y. Gilliéron and B. Merminod. Éléments de géomatique. EPFL MOOC, <https://www.coursera.org/course/geomatique>, accessed in June 2014.
- [53] J.-F. Girres and G. Touya. Quality assessment of the french openstreetmap dataset. *Transactions in GIS*, 14(4):435–459, 2010.
- [54] J. S. Greenfeld. Matching gps observations to locations on a digital map. In *Transportation Research Board 81st Annual Meeting*, 2002.
- [55] G. Grekousis and Y. Fotis. A fuzzy index for detecting spatiotemporal outliers. *GeoInformatica*, 16:597–619, 2012.

- [56] P.D. Groves, Z. Jiang, L. Wang, and M.K. Ziebart. Intelligent urban positioning using multi-constellation gnss with 3d mapping and nlos signal detection. In *Proceedings of ION GNSS*, 2012.
- [57] C. Guo, J. Meguro, Y. Kojima, and T. Naito. Cadas: A multimodal advanced driver assistance system for normal urban streets based on road context understanding. In *Intelligent Vehicles Symposium (IV)*, pages 228–235, June 2013.
- [58] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.
- [59] M. Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning. B, Planning & design*, 37(4):682, 2010.
- [60] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2), April 1950.
- [61] M. Hentschel and B. Wagner. Autonomous robot navigation based on openstreetmap geodata. In *Intelligent Transportation Systems (ITSC)*, pages 1645–1650, sept. 2010.
- [62] J. Ibanez-Guzman. Soft-safety for autonomous driving and contextual adas. Technical report, Renault S.A., 2014.
- [63] J. Ibanez-Guzman, N. Minoiu-Enache, H. G. Chale Gongora, J. Lesaing, and Chauveau F. Control of the autonomous mode of bimodal vehicles, 2013.
- [64] M. Jabbour, Ph. Bonnifait, and V. Cherfaoui. Map-matching integrity using multihypothesis road-tracking. *Journal of Intelligent Transportation Systems*, 12(4):189–201, 2008.
- [65] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74(18):1772–1782, 2001.
- [66] R. R. Joshi. A new approach to map matching for in-vehicle navigation systems: the rotational variation metric. In *Intelligent Transportation Systems, IEEE*, pages 33–38. IEEE, 2001.
- [67] E. Kaplan. *Understanding GPS - Principles and applications*. Artech House, 2nd edition edition, December 2005.
- [68] C. G. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff. Real-time recognition of u.s. speed signs. In *Intelligent Vehicles Symposium, IEEE*, pages 518–523, June 2008.
- [69] M. Krysander, F. Heintz, J. Roll, and E. Frisk. Flexdx: A reconfigurable diagnosis framework. *Engineering Applications of Artificial Intelligence*, 23(8):1303 – 1313, 2010.

BIBLIOGRAPHY

- [70] L. Lamard, R. Chapuis, and J.P. Boyer. Multi target tracking with cphd filter based on asynchronous sensors. In *Information Fusion (FUSION), International Conference on*, pages 892–898, July 2013.
- [71] O. Le Marchand. *Approche autonome pour la localisation et la surveillance de l'intégrité d'un véhicule automobile en environnement complexe*. PhD thesis, Université de Technologie de Compiègne, 2010.
- [72] A. Malaviya and L. Peters. Extracting meaningful handwriting features with fuzzy aggregation method. In *Document Analysis and Recognition, International Conference on*, volume 2, pages 841–844 vol.2, aug 1995.
- [73] D. H. Maling. Coordinate systems and map projections for gis. *Geographical Information Systems: Principles and Applications*. John Wiley & sons, pages 135–146, 1991.
- [74] D. Maquin and J. Ragot. *Diagnostic des systèmes linéaires*. Lavoisier, 2000.
- [75] B. Mathibela, M. A. Osborne, I. Posner, and P. Newman. Can priors be trusted? learning to anticipate roadworks. In *Intelligent Transportation Systems (ITSC), IEEE International Conference on*, pages 927–932, Sept 2012.
- [76] O. Mazhelis. Using recursive bayesian estimation for matching gps measurements to imperfect road network data. In *Intelligent Transportation Systems (ITSC), IEEE International Conference on*, pages 1492–1497. IEEE, 2010.
- [77] R. McMaster. A statistical analysis of mathematical measures for linear simplification. *The American Cartographer*, 23, 1986.
- [78] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1484–1497, Dec 2012.
- [79] J. Moras, V. Cherfaoui, and Ph. Bonnifait. Credibilist occupancy grids for vehicle perception in dynamic environments. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 84–89, May 2011.
- [80] J.-M. Morvan. Distance and projection. In *Generalized Curvatures*, volume 2 of *Geometry and Computing*, pages 47–56. Springer Berlin Heidelberg, 2008.
- [81] F. Moutarde, A. Bargeton, A. Herbin, and L. Chanussot. Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system. In *Intelligent Vehicles Symposium, IEEE*, pages 1122–1126, June 2007.
- [82] G. Nagib and W. Gharieb. Path planning for a mobile robot using genetic algorithms. In *Electrical, Electronic and Computer Engineering, ICEEC International Conference on*, pages 185–189, Sept 2004.
- [83] G. Nassreddine, F. Abdallah, and T. Denoeux. Map matching algorithm using interval analysis and dempster-shafer theory. In *Intelligent Vehicles Symposium, IEEE*, pages 494–499, June 2009.

- [84] D. Nienhuser, T. Gump, and J.M. Zollner. A situation context aware dempster-shafer fusion of digital maps and a road sign recognition system. In *Intelligent Vehicles Symposium, IEEE*, pages 1401–1406, June 2009.
- [85] J. Nieto, E. Slawinski, V. Mut, and B. Wagner. Online path planning based on rapidly-exploring random trees. In *Industrial Technology (ICIT), IEEE International Conference on*, pages 1451–1456, March 2010.
- [86] NIMA. Department of defense world geodetic system 1984. its definition and relationships with local geodetic systems. Technical report, National imagery and mapping agency, 2000.
- [87] V. Noronha and M. F. Goodchild. Map accuracy and location expression in transportation, reality and prospects. *Transportation Research Part C: Emerging Technologies*, 8(1 - 6):53 – 69, 2000.
- [88] European Road Safety Observatory. Traffic safety basic facts 2012: Junctions. Technical report, European Commission, 2012.
- [89] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik. Multipath detection with 3d digital maps for robust multi-constellation gnss/ins vehicle localization in urban areas. In *Intelligent Vehicles Symposium (IV), IEEE*, pages 184–190, June 2012.
- [90] W. Y. Ochieng, M. Quddus, and R. B. Noland. Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 2(55), 2009.
- [91] P. Ondruska and I. Posner. Probabilistic attainability maps: Efficiently predicting driver-specific electric vehicle range. In *Intelligent Vehicles Symposium (IV), IEEE*, 2014.
- [92] S. Peyraud, D. Bétaille, S. Renault, M. Ortiz, F. Mougel, D. Meizel, and F. Peyret. About non-line-of-sight satellite detection and exclusion in a 3d map-aided localization algorithm. *Sensors*, 13(1):829–847, 2013.
- [93] C. Pinana-Diaz, R. Toledo-Moreo, D. Betaille, and A.F. Gomez-Skarmeta. Gps multipath detection and exclusion with elevation-enhanced maps. In *Intelligent Transportation Systems (ITSC), 14th International IEEE Conference on*, pages 19–24, Oct 2011.
- [94] O. Pink and C. Stiller. Automated map generation from aerial images for precise vehicle localization. In *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, pages 1517 –1522, sept. 2010.
- [95] R. Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE*, 22(1):63 –84, jan 2000.
- [96] V. Popovic and B. Vasic. Review of hazard analysis methods and their basic characteristics. *FME Transactions*, 36(4):181–187, 2008.

BIBLIOGRAPHY

- [97] M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies*, 14(4):283–302, 2006.
- [98] M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312 – 328, 2007.
- [99] R. S. Radovanovic. High accuracy deformation monitoring via multipath mitigation by day-to-day correlation analysis. In *13th International Technical Meeting of the SAT Division of the ION, September*, pages 19–22, 2000.
- [100] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer, 2011.
- [101] M. Ravanbakhsh and C. S. Fraser. Road roundabout extraction from very high resolution aerial imagery. *International Archives of Photogrammetry and Remote Sensing*, 2009.
- [102] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical report, Centre for Intelligent Machines, McGill University, 2004.
- [103] C. Röss, D. Balzer, A. Bracht, S. Durekovic, and J. Löwenau. Adasis protocol for advanced in-vehicle applications. In *15th World Congress on Intelligent Transport Systems*, 2008.
- [104] R. A. Retting, B. N. Persaud, P. E. Garder, and D. Lord. Crash and injury reduction following installation of roundabouts in the united states. *American Journal of Public Health*, 91(4):628 – 631, 2001.
- [105] A. Schindler, G. Maier, and F. Janda. Generation of high precision digital maps using circular arc splines. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 246–251, June 2012.
- [106] M. Schreiber, C. Knoppel, and U. Franke. Laneloc: Lane marking based localization using highly accurate maps. In *Intelligent Vehicles Symposium (IV), IEEE*, pages 449–454, June 2013.
- [107] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson. Mining gps traces for map refinement. *Data Mining and Knowledge Discovery*, 9:59–87, 2004.
- [108] A. Selloum, D. Betaille, E. Le Carpentier, and F. Peyret. Robustification of a map aided location process using road direction. In *Intelligent Transportation Systems (ITSC), 13th International IEEE Conference on*, pages 1504–1510, Sept 2010.
- [109] R. Soden and L. Palen. From crowdsourced mapping to community mapping: The post-earthquake work of openstreetmap haiti. In *11th International Conference on the Design of Cooperative Systems (COOP 2014), Nice, France*, 2014.

- [110] P. Sundvall and P. Jensfelt. Fault detection for mobile robots using redundant positioning systems. In *Robotics and Automation, ICRA. IEEE International Conference on*, pages 3781–3786, may 2006.
- [111] I. Szottka. Particle filtering for lane-level map-matching at road bifurcations. In *Intelligent Transportation Systems, ITSC. IEEE International Conference on*, pages 154–159, Oct 2013.
- [112] Z. Tao, Ph. Bonnifait, V. Fremont, and J. Ibañez-Guzman. Lane marking aided vehicle localization. In *Intelligent Transportation Systems, ITSC. IEEE International Conference on*, pages 1509–1515, 2013.
- [113] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [114] R. Toledo-Moreo, D. Betaille, and F. Peyret. Lane-level integrity provision for navigation and map matching with gns, dead reckoning, and enhanced maps. *Intelligent Transportation Systems, IEEE Transactions on*, 11(1):100–112, March 2010.
- [115] N. R. Velaga, M. A. Quddus, A. L. Bristow, and Y. Zheng. Map-aided integrity monitoring of a land vehicle navigation system. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):848–858, June 2012.
- [116] M. Vertet and S. Giauxserand. *Comprendre les principaux paramètres de conception géométrique des routes*. Sétra, 2006.
- [117] R. Wang and H. J. Ruskin. Modeling traffic flow at a single lane urban roundabout. *Computer Physics Communications*, 147:570 – 576, 2002. Proceedings of the Europhysics Conference on Computational Physics Computational Modeling and Simulation of Complex Systems.
- [118] C. Zinoune. Path planning for cooperative autonomous vehicles. Master’s thesis, School of Engineering, Cranfield University, 2010.

Appendix A

Experimental Set-up, Maps and Software Tools

Contents

A.1 Test Vehicle	141
A.2 Maps	142
A.3 Navigation Systems	146
A.4 Software Tools for Navigation Maps Edition	151

The experimental studies presented in this manuscript have been realized using equipped vehicles and different maps managed by several software tools. This appendix gives an overview of these materials. In particular, the management of different road databases, the generation of faults and the Electronic Horizon elaboration are presented with a focus on crucial procedures necessary to provide the input information to test the Fault Detection and Isolation methods.

A.1 Test Vehicle

The experiments presented in this thesis were done with a standard passenger vehicle: 2008 *Renault Espace IV* equipped with a 2.0 litres turbo petrol engine (Figure A.1). This vehicle benefits from various equipments such as Anti-lock Braking System (ABS), Electronic Stability Program (ESP), automatic headlamp alignment. Consequently, the vehicle is equipped with a large number of sensors that provide data on the vehicle internal network (CAN-bus) [71]. These data are therefore accessible and is used for the experiments:

- The four wheel speeds are available at a 50 ms period.
- The steering wheel angle is available at a 20 ms period.
- Two MEMS accelerometers measure the longitudinal and transversal vehicle accelerations and provide data every 40 ms.



Figure A.1: Test vehicle

- One MEMS gyroscope provides measurements of the vehicle yaw rate every 40 ms.

In addition to this information on the vehicle dynamics, the state of all the driver actuations are available on the CAN-bus (e.g., pedals positions, indicator position, state of the cruise control system, wiper state). The CAN-bus data is recorded on the test computer through a serial communication port.

The GNSS receiver used for the experiments is a *u-blox 6T* GPS receiver. This is a standard sensors equipped with RAIM capabilities. The internal Kalman Filter was disabled to comply with assumptions made in this work.

For validation purposes, this vehicle is also equipped with high precision localisation sensors. A high grade *Ixsea (IxBlue) PHINS* Inertial Measurement Unit (IMU) provides vehicle localization to the test computer by a serial port. Its fibre-optic gyroscopes and accelerometers measure the relative vehicle motion and global localization is done by fusing IMU data from a dual frequency *Novatel* GPS receiver.

The data is recorded with RTMAPS. This software allows to synchronise sensor data and record it. Replay of recorded data in the same configuration as during the recording is also possible. This functionality is used for the evaluation of the methods developed in this thesis.

A.2 Maps

The navigation maps used in this thesis are described in this section. Most of the developments and experiments done in this thesis are based on OSM maps. This

file format is therefore introduced first. The navigation maps used in the PAMU project were also used as a basis for developing reference map for this work. These also led some considerations about the format and reference frames that can be used for vehicle navigation systems. They are therefore also detailed in this section.

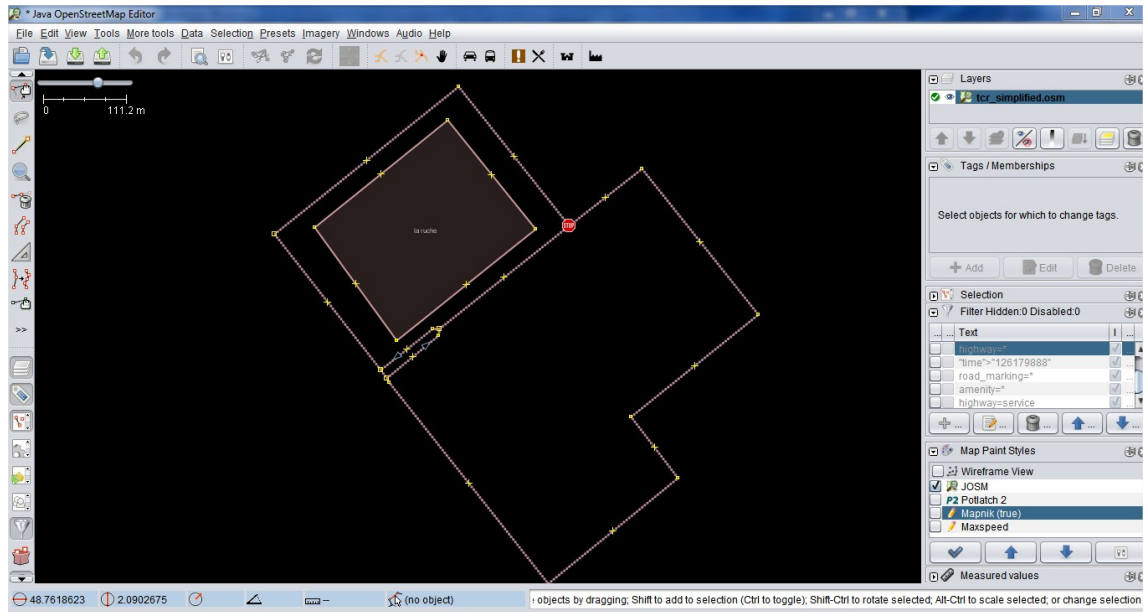
A.2.1 OpenStreetMap

OpenStreetMap is a worldwide collaborative mapping project. The geographic data is available through a website and can be consulted and modified online by any registered user (signing up is free and requires a valid e-mail address) [2].

In OSM standard, geographic information is represented in 2 dimensions by three basic entities: nodes, ways and relations. They are defined as follows:

- *Nodes*. These allow to describe the geometry of all the elements in the map. These are defined by their geodetic coordinates (latitude and longitude). Tags (i.e. attributes) can be associated to each node. For example if a node describes a road stop sign, the latitude and longitude of the node describe the position of the sign, and the tag *highway* with the value *stop* is associated to the node.
- *Ways*. Each way is an ordered list of nodes previously defined. They are used to model poly-lines (e.g., roads, rivers, power lines) and areas (e.g., buildings, boundaries, forests) in the map. The description of the ways are provided by attributes associated to them. For example all the ways related to roads have the tag *highway*; the value of this tag depends on the road class (e.g., *primary*, *secondary*, *residential*).
- *Relations*. These are made of one or several tags, nodes and ways to describe logical or geographical relationships. They can be used, for example, to describe a turn restriction at an intersection: the relation *no turn* is established between a pair of ways at the intersection.

The map is stored in an XML file with *.osm* extension. It can be opened with a text editor. Figure A.2a shows an example of a simple map and Figure A.2b shows the corresponding OSM file. The structure of this file corresponds to the three basic entities described previously. The first part is dedicated to the definition of nodes. For each node, an ID and the coordinates are defined. If necessary, the tags are on the following lines. For the example taken here, the stop intersection is represented by the tag *highway=stop* of the node -67 at line 16 (Figure A.2b). The second part of the file defines the ways of the map. They consist in a list of node ID previously defined. The tags associated to the way is after the node list. In this example, one can notice that the way defined at lines 73 to 81 has the same starting and ending nodes. The *building* tag shows that this closed way describes the office building *La Ruche*. The last part of the OSM file is for the definition of relations (no relation is defined in this example).



(a) Map view.

```

1  <?xml version='1.0' encoding='UTF-8'?>
2  <osm version='0.6' upload='true' generator='JOSM'>
3    <node id='-91' visible='true' lat='48.76054365402234' lon='2.081654403010075' />
4    <node id='-89' visible='true' lat='48.76053944947221' lon='2.0817824385517576' />
5    <node id='-87' visible='true' lat='48.76045984747889' lon='2.0817643254546545' />
6    <node id='-85' visible='true' lat='48.76166828417479' lon='2.078785055922189' />
7    <node id='-83' action='modify' visible='true' lat='48.759952648696014' lon='2.080819855608538' />
8    <node id='-81' action='modify' visible='true' lat='48.76005110830078' lon='2.0807119102263774' />
9    <node id='-79' visible='true' lat='48.759905093519706' lon='2.080874509184484' />
10   <node id='-77' visible='true' lat='48.75750739103097' lon='2.0837549485881306' />
11   <node id='-75' visible='true' lat='48.75876543010232' lon='2.0861003353035152' />
12   <node id='-73' action='modify' visible='true' lat='48.75949697888917' lon='2.0852536723030117' />
13   <node id='-71' visible='true' lat='48.760712455317886' lon='2.0875574234277967' />
14   <node id='-69' visible='true' lat='48.76245331135705' lon='2.0854482835206856' />
15   <node id='-67' action='modify' visible='true' lat='48.76177674883916' lon='2.0841257482011586'>
16     <tag k='highway' v='stop' />
17   </node>
18   <node id='-65' visible='true' lat='48.7634332594107' lon='2.0821340762835305' />
19   <node id='-63' visible='true' lat='48.76174790726769' lon='2.079517142226621' />
20   <node id='-61' visible='true' lat='48.763027434353326' lon='2.0819324627349975' />
21   <node id='-59' visible='true' lat='48.761733366999856' lon='2.0835206186857107' />
22   <node id='-57' visible='true' lat='48.760402914679645' lon='2.0810060384304148' />
23   <way id='-103' action='modify' visible='true'>
24     <nd ref='-91' />
25     <nd ref='-89' />
26     <nd ref='-87' />
27     <nd ref='-85' />
28     <nd ref='-83' />
29     <nd ref='-81' />
30     <nd ref='-79' />
31     <nd ref='-77' />
32     <nd ref='-75' />
33     <nd ref='-73' />
34     <nd ref='-71' />
35     <nd ref='-69' />
36     <nd ref='-67' />
37     <nd ref='-65' />
38     <nd ref='-63' />
39     <nd ref='-61' />
40     <nd ref='-59' />
41     <nd ref='-57' />
42     <nd ref='-55' />
43     <nd ref='-53' />
44     <nd ref='-51' />
45     <nd ref='-49' />
46     <nd ref='-47' />
47     <nd ref='-45' />
48     <nd ref='-43' />
49     <nd ref='-41' />
50     <nd ref='-39' />
51     <nd ref='-37' />
52     <nd ref='-35' />
53     <nd ref='-33' />
54     <nd ref='-31' />
55     <nd ref='-29' />
56     <nd ref='-27' />
57     <nd ref='-25' />
58     <nd ref='-23' />
59     <nd ref='-21' />
60     <nd ref='-19' />
61     <nd ref='-17' />
62     <nd ref='-15' />
63     <nd ref='-13' />
64     <nd ref='-11' />
65     <nd ref='-9' />
66     <nd ref='-7' />
67     <nd ref='-5' />
68     <nd ref='-3' />
69     <nd ref='1' />
70     <nd ref='3' />
71     <nd ref='5' />
72     <nd ref='7' />
73     <nd ref='9' />
74     <nd ref='11' />
75     <nd ref='13' />
76     <nd ref='15' />
77     <nd ref='17' />
78     <nd ref='19' />
79     <nd ref='21' />
80     <nd ref='23' />
81     <nd ref='25' />
82     <nd ref='27' />
83     <nd ref='29' />
84     <nd ref='31' />
85     <nd ref='33' />
86     <nd ref='35' />
87     <nd ref='37' />
88     <nd ref='39' />
89     <nd ref='41' />
90     <nd ref='43' />
91     <nd ref='45' />
92     <nd ref='47' />
93     <nd ref='49' />
94     <nd ref='51' />
95     <nd ref='53' />
96     <nd ref='55' />
97     <nd ref='57' />
98     <nd ref='59' />
99     <nd ref='61' />
100    <nd ref='63' />
101    <nd ref='65' />
102    <nd ref='67' />
103    <nd ref='69' />
104    <nd ref='71' />
105    <nd ref='73' />
106    <nd ref='75' />
107    <nd ref='77' />
108    <nd ref='79' />
109    <nd ref='81' />
110    <nd ref='83' />
111    <nd ref='85' />
112    <nd ref='87' />
113    <nd ref='89' />
114    <nd ref='91' />
115    <tag k='building' v='office' />
116    <tag k='name' v='la ruche' />
117  </way>
118 </osm>

```

(b) OSM file of the navigation map.

Figure A.2: Example of OSM map and corresponding OSM file.



Figure A.3: Autonomous valet parking vehicle (PAMU project).

The text format of the OSM map is convenient for developing applications based on OSM data. A simple text reader routine or XML parser can be used to load the map in a custom application (e.g., in Matlab, C++, Java).

A.2.2 Map for PAMU Project

The Renault PAMU (Urban Mobility Application Platform) project aimed at developing an autonomous valet parking vehicle. It was designed to operate in the Renault research centre (Technocentre). The challenge relied on the use of only standard sensors for passenger vehicles. This electrical vehicle (Figure A.3) is therefore equipped front and rear radar, front cameras and GPS receivers.

The navigation map is an important topic of this project. It therefore led efforts in map specification, creation and evaluation. A mapping survey was done on the PAMU test track in order to produce a highly accurate map dedicated for autonomous navigation. This map contains three main elements described as follows:

- *Links*. This contains the ways that the vehicle is likely to use. These are described in terms of connectivity by a set of *nodes* and in geometry by a set of *shape points*.
- *Points of Interest*. These points (or areas) represent every road contextual static features. They can be for example pedestrian crossings, traffic signs, bumpers, battery charging areas, driver pick-up points.
- *Link border*. This map layer contains the road lane markings. Since the vehicle is equipped with a camera-based lane detection system, this layer is used to improve the vehicle positioning by fusing the detected lanes with GNSS estimates.

In order to facilitate the processing of vehicle objects detection, path planning and control functions, the choice was made to use a local ENU reference frame for the

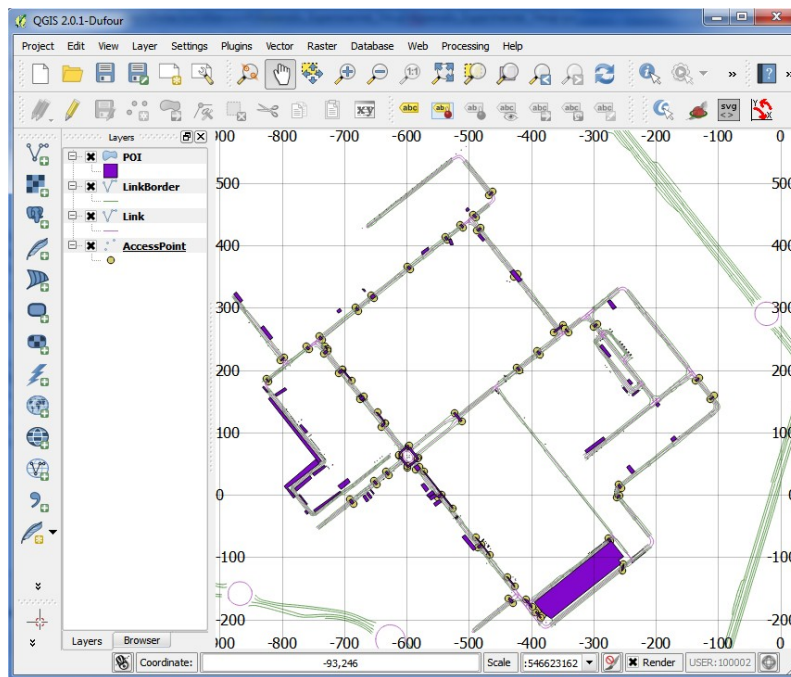


Figure A.4: Navigation map for PAMU project.

map. The closest reference point of the French levelling network is used as origin of the reference frame: boundary marker K.A.M3-226 (48.759444 ° North, 2.088889 ° East).

A company in charge of mobile mapping (Viametris) provided the map for PAMU according to the Renault specifications. A vehicle equipped with camera, laser scanners and high precision localization systems was driven on the test track. The road features were then manually drawn on point cloud images. The result shown in Figure A.4 was exported in a Spatialite database which is used in the PAMU vehicles. Spatialite is a library that allows to use SQL databases for storing geographic information. It also permits to perform optimized spatial queries in the database.

Since a dedicated survey was done for this map, it is used in this thesis as a reference (i.e. ground truth). However, no navigation system that uses Spatialite database as navigation maps is available. Conversions into other formats are therefore required to make this map usable for this thesis. This is detailed in the following sections.

A.3 Navigation Systems

This section details the software tools used as vehicle on board navigation systems. These therefore aim at providing contextual road information to client systems in real time while the vehicle is driven. The Navteq research platform ADAS-RP is

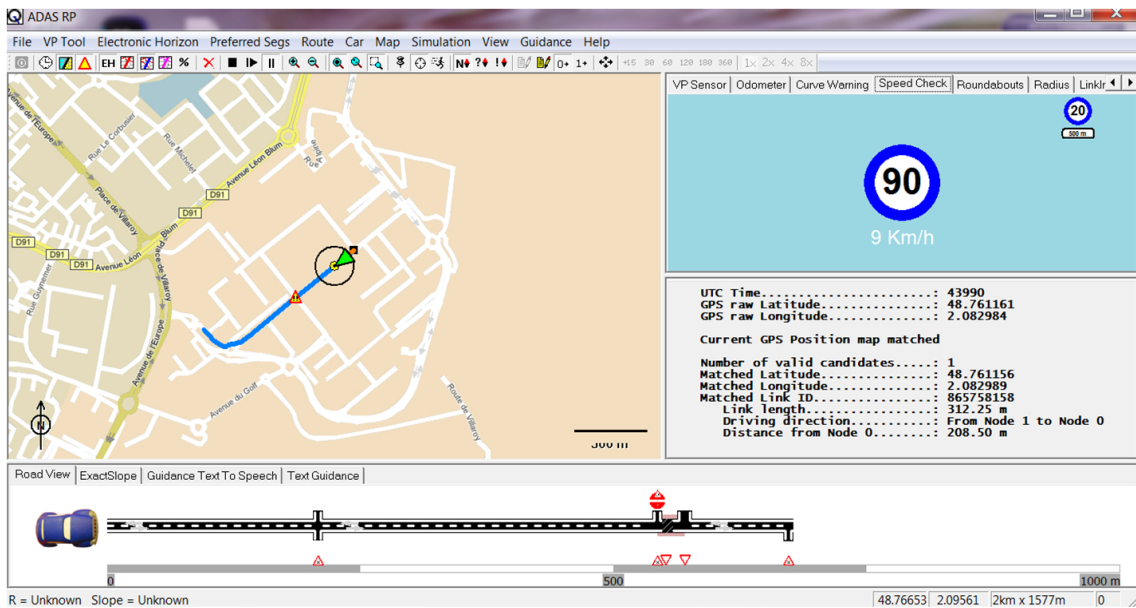


Figure A.5: ADAS-RP user interface.

presented first since it has been used for the first experiments led in this thesis. This software is representative of passenger vehicle navigation systems because it is based on the same internal algorithms and navigation maps. Moreover, ADAS-RP offers a framework for the development of ADAS functions (known as plug-ins). However, the modification of the navigation map used in ADAS-RP is forbidden which is not convenient to create map faults and to evaluate the methods presented in this work. A navigation system based on an editable navigation map has therefore been developed. The experiments presented in this thesis were done using this software which is introduced in this section.

A.3.1 ADAS-RP

At the beginning of this research, the Navteq research platform ADAS-RP was used as navigation system. This software, shown in Figure A.5, takes as an input the GPS data (in real time or in replay), performs the map-matching and extracts navigation data from the map. A development kit is also available to prototype and develop in-vehicle applications. The user interface of the developed application appears on the right-hand side of ADAS-RP interface. Figure A.5 shows the current value of the *Speed Check* plug-in which is an example of development provided with ADAS-RP.

Even if more recent versions were available, this software was used with a 2008 *Navteq* database in order to benefit from a larger number of map errors. The data of this navigation map is not directly accessible. It is therefore impossible for an application to search within the map database. Map information is solely accessible

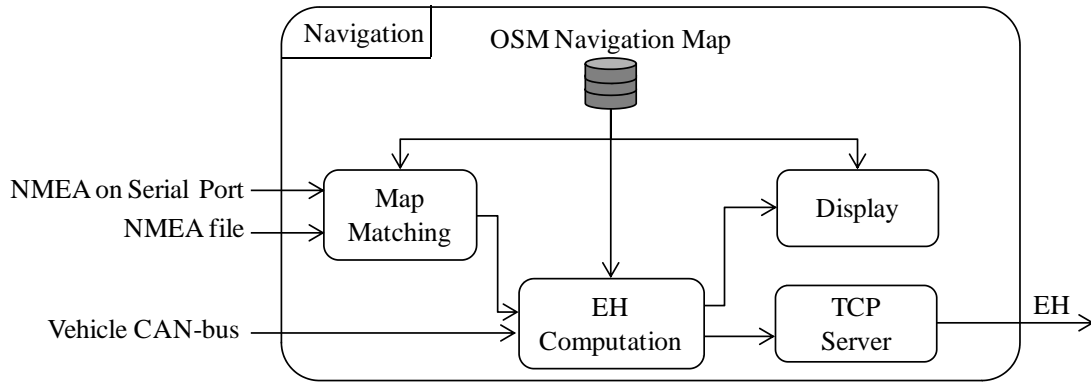


Figure A.6: Structure of the navigation system based on OpenStreetMap developed for testing purposes.

by the Electronic Horizon (EH); this means that available map data refers only to the close vehicle environment. As shown in Figure A.5, the EH is displayed on top of the map by a blue line. A linear view is also provided on the lower part of the figure.

The EH data structure within ADAS-RP is similar to the ADASIS communication protocol. Information on the oncoming road features are located with their offset with respect to the current vehicle position. Probabilities are associated to every road segment that the vehicle is likely to take. The size of the EH is defined by a maximum length (distance) and depth (number of route changes).

A.3.2 Navigation System based on OpenStreetMap

As the navigation map is not editable in ADAS-RP, we used to search the database manually using the user interface, to find map faults and then to drive on these areas. This led to a 1000 km driving campaign. To facilitate this process and to be able to create typical faults in the map, a navigation system based on an editable navigation map (i.e. OpenStreetMap) was developed.

A.3.2.1 Software

Figure A.6 shows the structure of this software. Its two first inputs are the current vehicle position in the NMEA standard (the National Marine Electronics Association, NMEA, standard defines a communication protocol dedicated to positioning and navigation devices). This can be provided in real time by a GNSS receiver through a serial communication port or in replay by a NMEA text file recording. The last input is the vehicle CAN-bus which is used to provide the indicator state which is used in the EH computation.

The map-matching algorithm is based on fuzzy logic and inspired from [90]. The vehicle position, speed and heading are used in this process. Three algorithms are used depending on the road context for optimization purposes (reactivity and computational time):

- *Initial Map-matching Process* (IMP): It is used when no previous map-matched position is available. The algorithm tests every road in the close vehicle environment (search zone).
- *Subsequent Map-matching Process 1* (SMP1): A previous map-matched position is available and the vehicle is not approaching an intersection. The algorithm propagates the past position along the road segment and fuses it with the NMEA input.
- *Subsequent Map-matching Process 2* (SMP2): A previous map-matched position is available and the vehicle is approaching an intersection. The past position is propagated along every road of the intersection and the NMEA input is used to find the best candidate.

The future vehicle path is estimated in the *EH Computation* function based on the road classes, road shapes and the state of the vehicle indicators. Fuzzy logic is also used in this function.

The EH is finally sent by TCP communication; client application can therefore be developed independently from the navigation system by using a TCP client to retrieve EH data. The Integrity Monitoring methods presented in this thesis are clients of this navigation system.

A.3.2.2 User Interface

Figure A.7 shows the user interface of the navigation system. The main part of the window (left-hand side) is dedicated to the display of the vehicle position, EH and navigation map. The map rendering engine is based on the *libosmscout* C++ library. Basic controls are also available (pan map, zoom in and out, centre view on current vehicle position). The application parameters are on the right-hand side and are detailed as follows:

- *Database*: Selection of a compiled OSM database. An OSM file can be compiled by clicking on *File/Import Database* to be then used for navigation.
- *NMEA Input*: Selection of positioning input. If a NMEA file is selected, controls allow to start, stop, accelerate and jump in the replay time line.
- *CAN Bus*: Parameters for indicator signal decoding.
- *TCP Server*: IP address and port for the TCP server used to broadcast the EH.

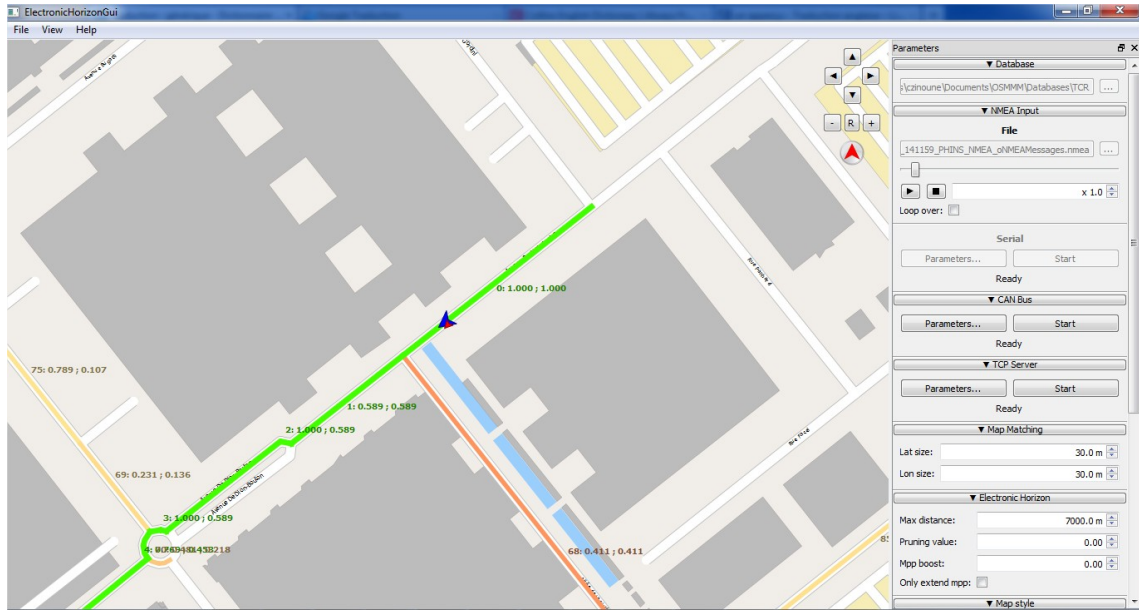


Figure A.7: User interface of the navigation system. The vehicle position estimate from GNSS (input of the navigation system) is the red arrow. The map-matched vehicle position is the blue arrow. The EH is displayed in green (most probable path) and in scale of orange (secondary paths). The score (i.e. probability) of each path is also displayed.

- *Map Matching*: Dimensions of the search zone used in IMP algorithm.
- *Electronic Horizon*: Maximum length of the EH and minimum probability of path of the EH.
- *Map Style*: Display settings of the OSM map.
- *Electronic Horizon Style*: Display settings for the EH.
- *Performance*: Information on current software performance. Computation time for map-matching and EH, final score of the map-matching estimate and algorithm used to provide this estimate.

A.3.2.3 Electronic Horizon Data Structure

The EH is described using four data structures which are sent by TCP server. These are detailed as follows:

- *Satellite_data()*: Information on the vehicle position estimate prior to the matching process (i.e. as provided by the NMEA input and shown by the red arrow in the interface). This includes latitude, longitude, standard deviations, etc.

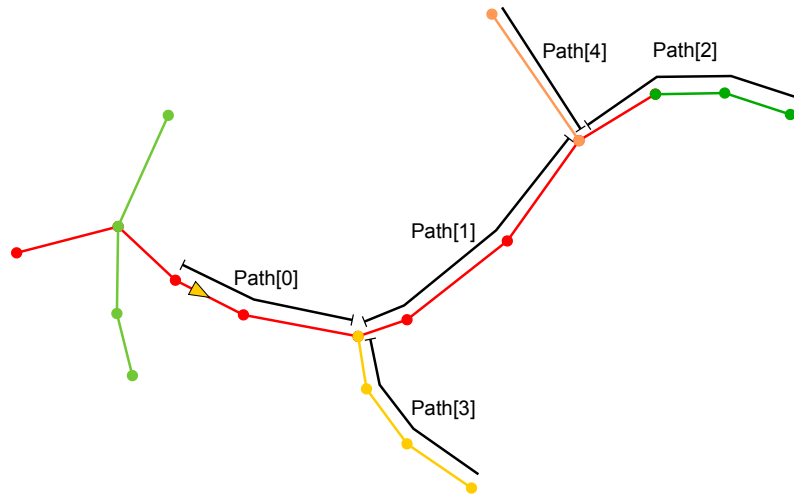


Figure A.8: Electronic Horizon data structure. Map road segments that belong to the same way are represented in the same colour. The electronic horizon paths are ordered according to their probability.

- *Map_matching_data()*: Information on the vehicle map-matched position (i.e. blue arrow in the interface). This includes for example, position, offset with respect to the first path, confidence, algorithm used.
- *Electronic_horizon_data()*: Paths that compose the EH. Several information are linked to each path of the EH (e.g. probability, parent and child path, is it part of the most probable path, map way ID).
- *Way_list()*: This structure contains the ways of the OSM map that take part of the EH and the data recorded in the OSM map. It contains the nodes coordinates and attributes, and the way description.

Figure A.8 shows the data contained in the EH provided by this navigation software. Except for the first one (Path[0]), each path of the EH starts and ends at an intersection. However, in the OSM standard, a way is defined to be a set of road segments which all have the same attributes. Thus a way do not necessarily fits to a path. This is shown in Figure A.8: each OSM way is denoted by a colour and the EH path are shown by black lines.

A.4 Software Tools for Navigation Maps Edition

This section describes the software tools used in this thesis to edit and display navigation maps. Java OpenStreetMap is presented first as it is very convenient for handling of maps in OSM format. Quantum GIS is then introduced because this generic Geographic Information System allows to manage a large variety of map

formats. This is therefore used to display PAMU maps. The software that generates faults in navigation maps and which was developed for this thesis is finally described.

A.4.1 Java OpenStreetMap

Java OpenStreetMap (JOSM) is an open-source software dedicated to the edition of OpenStreetMap files. This software has a number of basic functionalities (e.g., map download and upload from and to OSM server, map edition, display of aerial imagery) and can be completed by installing plug-ins. The typical use of JOSM consists in downloading OSM data from server on small area, displaying it on top of aerial imagery, mapping missing features and attributes and uploading the result back to the OSM server.

JOSM is used in this thesis to edit maps in OSM format for test purposes. The data from OSM server is used as a basis for creating the test maps; the upload to OSM server of these test map is not used here. This section details the main functions of JOSM as used in this thesis. The basic tools can be found by using the software however, in this section, the focus is on tools that may not appear to new users and which are particularly useful for this work.

A.4.1.1 Map Download from OSM Server

The last version of OSM map available on the server can be downloaded by clicking on *File/Download from OSM*. The user should then define an area to download as shown in Figure A.9. Once downloaded, every element of the map is editable (i.e. geometry, topology and attributes). It can also be saved locally in a *.osm* text file.

NMEA recordings can be imported into JOSM: opening an NMEA file (*File/Open*) automatically creates a new data layer in JOSM dedicated to this file. Graphical properties of the NMEA layer (right click on the layer and *Customize track drawing*) allow to display other parameters of the recording like for instance circles representing the Horizontal Dilution of Precision (HDOP) of each position (*Draw a circle from HDOP value*) or colours showing the speed associated to each position (*Track and Point Colouring/Velocity*). If the plug-in *ElevationProfile* is installed, a graph of heights along the NMEA recording is also displayed.

NMEA recordings are also used in this thesis to create reference maps. The *Espace* test vehicle is equipped with high precision sensors that can provide estimates of the vehicle position in the NMEA format (see Section A.1). This data is recorded while driving on the centre of the lane (as well as possible). These recordings are imported into JOSM and converted into an OSM map (right click on the NMEA layer and *Convert to Data Layer*). This process converts each NMEA position estimate into a map nodes. All the nodes are to belong to one single way. Figure A.10 shows the result of the conversion of a 10 Hz IMU NMEA recording on top of the OSM

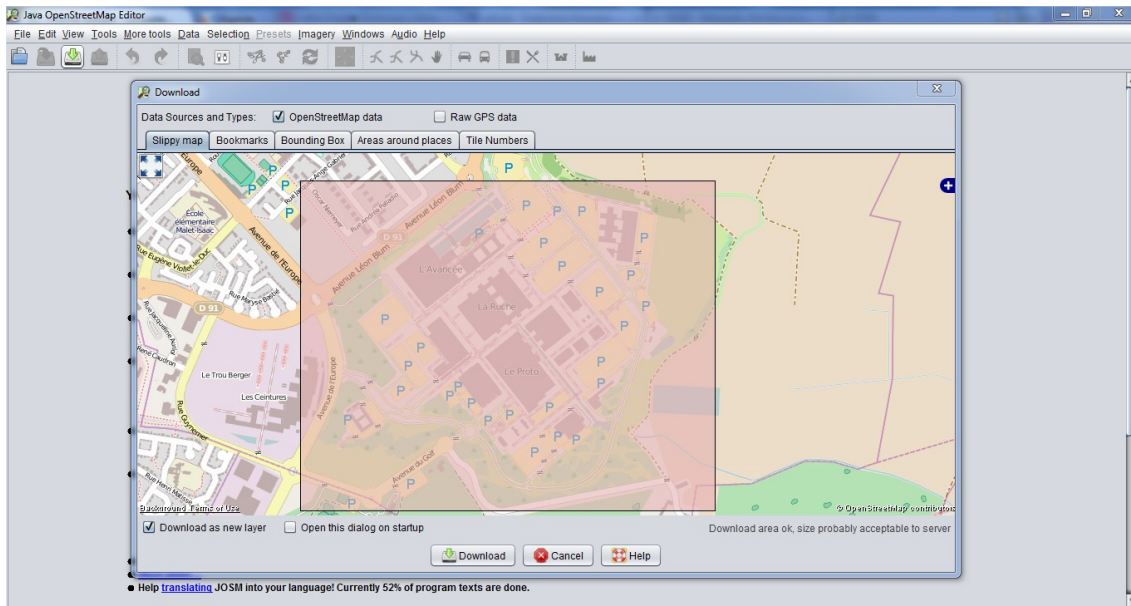


Figure A.9: Java OpenStreetMap download interface. The Renault research centre (Technocentre) is chosen as the download area.

map (of the Technocentre) downloaded from the server. It must be noticed that for this experiment the vehicle did not follow exactly the road centreline due to other vehicles present on the track. The resulting geometry can finally be modified to create a complete and usable map.

A.4.1.2 Simplifying Ways

JOSM offers the a simple command that simplifies ways (i.e. delete unnecessary nodes) in the map. This is used here for two reasons. First, the conversion of NMEA data into a map may create a large number of way nodes (shape points) which may be unnecessary and could slow down the map-based algorithms. Second, navigation map providers use this process on maps used in intelligent vehicle; doing it with JOSM allows to simulate faults of usual navigation maps.

To simplify a way which is selected in JOSM, click on *Tools/Simplify Way*. The nodes of the way are then deleted until the error induced by the deletion reaches a threshold. The default value of this threshold is 3 m. This value can be changed in *Edit/Preferences*, the *Expert Mode* should be enabled. In the *Advanced Preferences* tab, the value should be set (in metres) in the key *simplify-way.max-error*. Figure A.11 shows a part of the map resulting from the conversion of NMEA recording simplified with two thresholds. The blue line was simplified with 3 m threshold and the red line was simplified with 0.1 m threshold. The background map is the Technocentre map from OSM server.

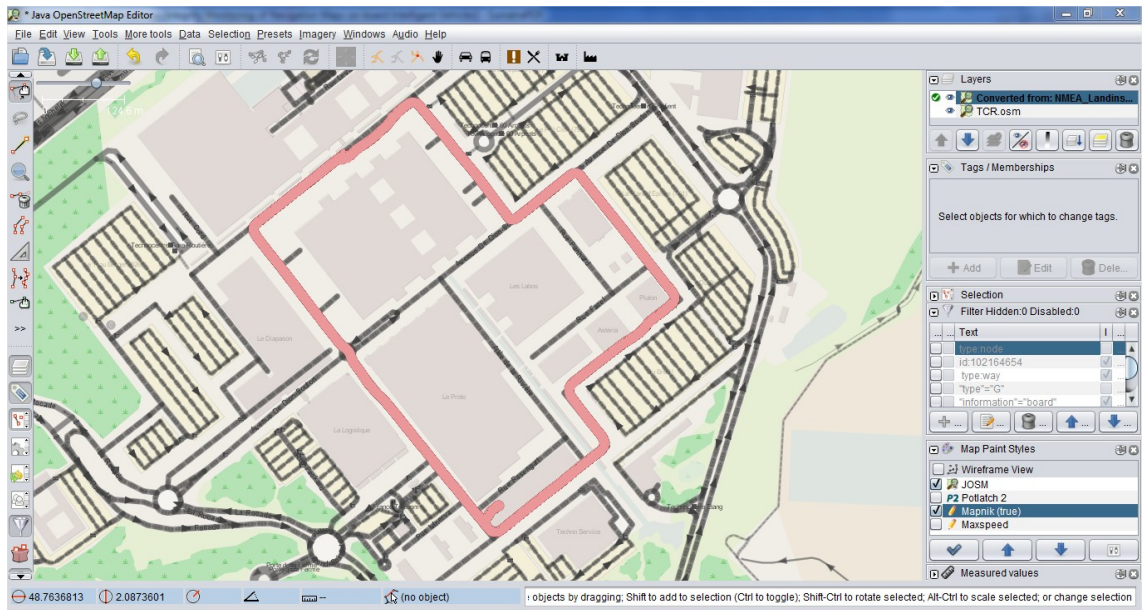


Figure A.10: Result of the conversion of NMEA recording into OSM map (red line). The background map is the data downloaded from OSM server.

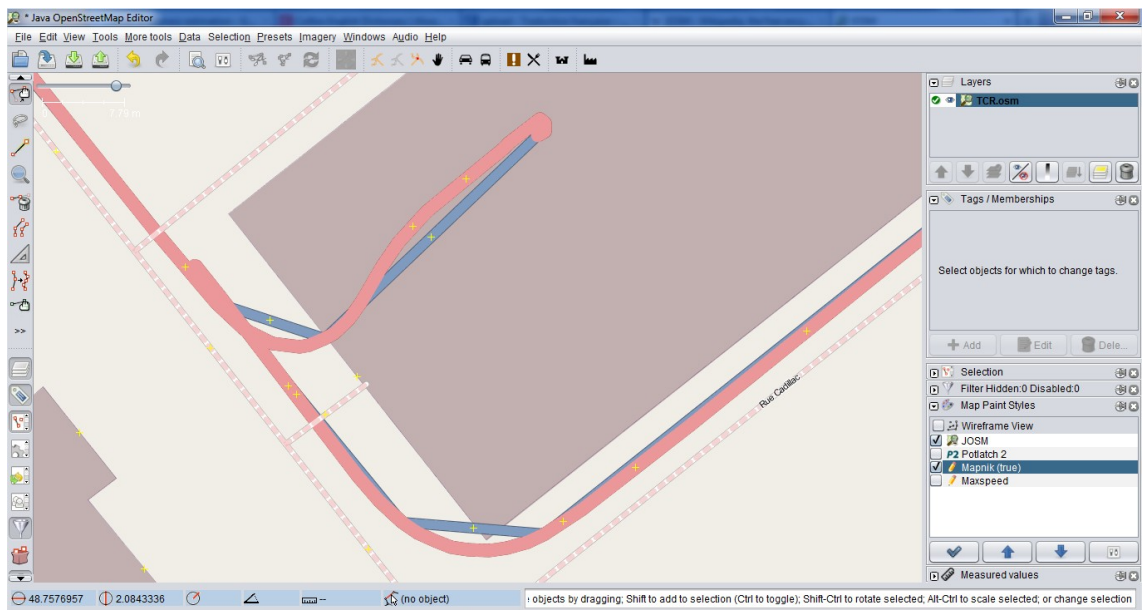


Figure A.11: Simplification of the way resulting from NMEA recording with 3 m threshold (in blue) and with 0.1 m threshold (in red).

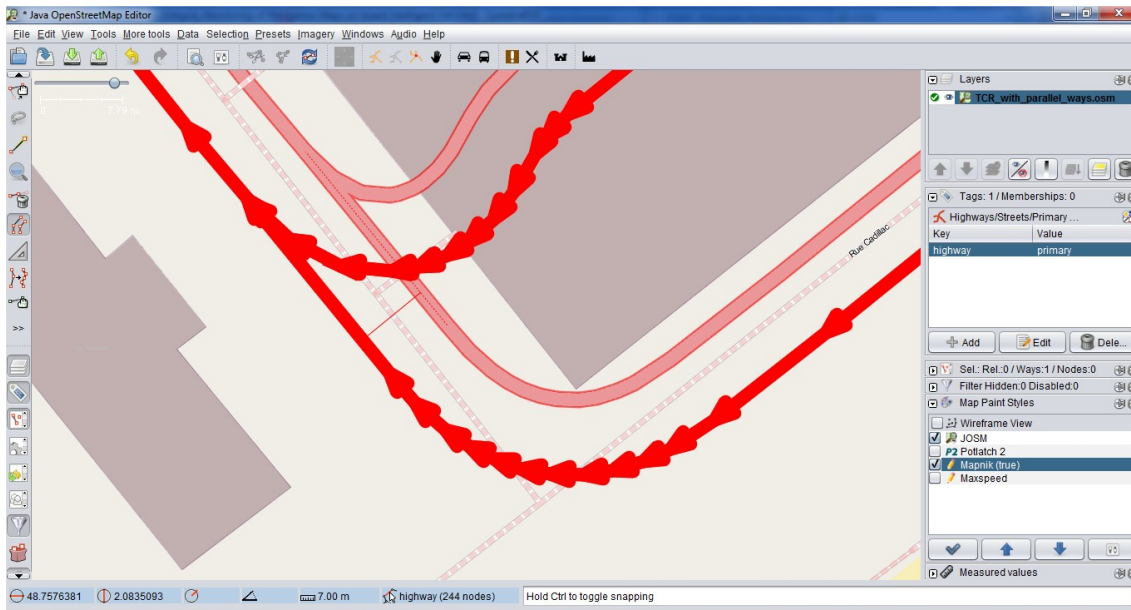


Figure A.12: Creation of parallel ways. The original way is in light red and the way which is being created is in bright red. The current offset value is displayed at the bottom of the figure (7 m).

A.4.1.3 Making Parallel Copies of Ways

This tool (*Edit Toolbar/Make Parallel Copies of Ways*) allows to duplicate a way into another way with a constant dilatation as shown in Figure A.12. The new way is obtained by click-and-drag of the original way. When dragging, the dilatation value is displayed at the bottom of the window (7 m in the example of Figure A.12). This is convenient for creating lane level maps based on a single NMEA recording. Each lane of the map can therefore be created by duplicating the NMEA recording with several offsets. However this only provides an approximate model of the lanes; high accuracy lane level maps should be created based on appropriate mapping survey.

A.4.1.4 Plug-ins

Additional functions are provided to JOSM by plug-ins. Those are developed by the JOSM community members and made available to all users. The plug-ins are installed thanks to the tab *Plug-ins* of the JOSM preferences window (*Edit/Preferences*). The plug-ins used in this thesis are listed as follows:

- *Measurement*. This provides tools for making measurements of distances and surfaces on the map.
- *Download Along*. A box is usually employed to define the download area from OSM server. This box may include a two large amount of data when one want

to get OSM map along a long vehicle trip. This plug-in downloads OSM data only along the NMEA recording.

- *RoadSigns*. This provides tools for defining road signs in the map as well as road signs rendering.
- *OpenData*. This makes possible to import Shapefile data into JOSM.
- *ElevationProfile*. This displays the height profile of NMEA recordings in JOSM
- *UtilsPlugin2*. This provides additional tools for map edition (e.g., adding nodes where two ways cross each other, aligning nodes, creating circle arcs).

A.4.2 Quantum GIS

Quantum GIS (QGIS) is a geographic information system under the GNU General Public License. QGIS is a general purpose GIS so it allows to manage several types and formats of geographic data. Navigation map handling is not intuitive with this software, however some functionalities were used in this thesis like format conversion, display of several maps in the same window, etc. The key operations required to achieve this are detailed in this section.

A.4.2.1 Display Geographic Information

The user interface is shown in Figure A.13. The geographic information is represented as layers which can result from various files. Some geographic information file formats do not contain the Reference Coordinate System (RCS) associated to the information. The file therefore contains the coordinates of the geographic elements but not the reference in which these coordinates are provided. This is problematic when one tries to display several layers with data that do not have the same RCS (known as CRS for Coordinate Reference System in QGIS).

An example is given in Figure A.13. A GPS recording of the PAMU test vehicle is displayed on top of a map of the test track (in Renault Technocentre). The GPS recording is in blue in the NMEA standard. It therefore contains geodetic vehicle coordinates (Latitude, Longitude, Height). However, the map layers are in local ENU reference frame: Points of Interest (POI), junction nodes (Node), lane markings (LinkBorder), drivable lanes (Link) and access points for POI (AccessPoint). The RCS of each layer is not implicitly detected by QGIS, it must therefore be defined manually for each layer (right click on the layer and select *Set layer CRS*). The WGS84 must be chosen for data with geodetic coordinates. Local ENU reference frame used for this experiment is dedicated to PAMU project. It must therefore be manually defined in the *PROJ.4* standard. A new custom projection must be defined in *Settings/Custom CRS*. For the example chosen here, it is done by the following command:

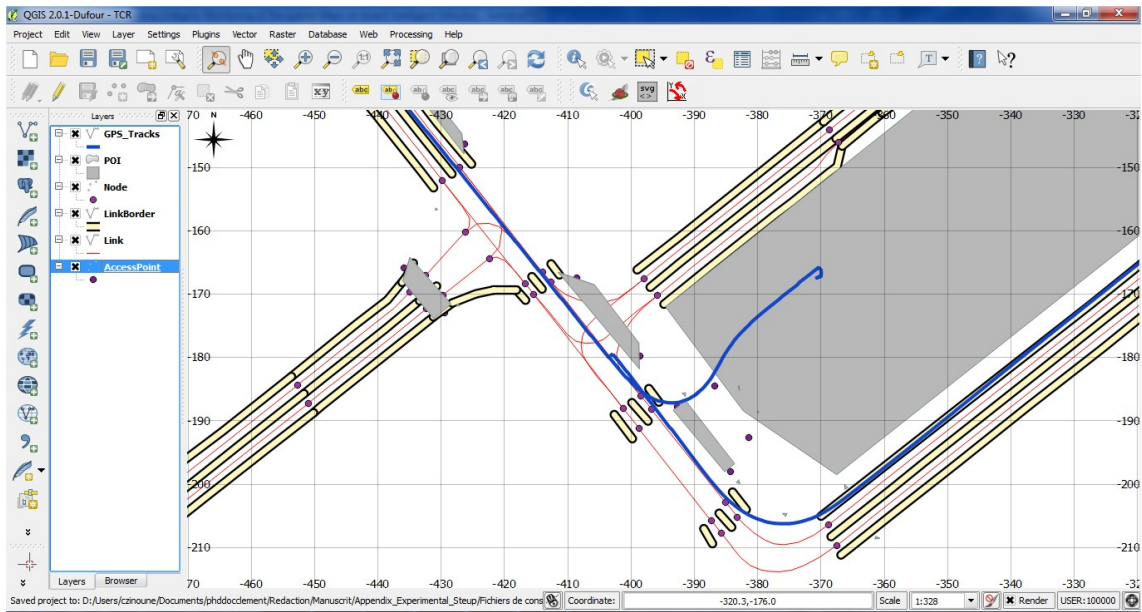


Figure A.13: Quantum GIS user interface. GPS tracks of the PAMU vehicle (blue line) on top of the test track map. Data is displayed in local ENU reference frame.

```
+proj=tmerc +lat_0=48.759444 +lon_0=2.088889
+x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs
```

where:

- *+proj=tmerc* defines the projection type as Transverse Mercator cylindrical projection.
- *+tlat_0* and *+tlon_0* are the geodetic coordinates of the ENU tangent plane origin (latitude and longitude respectively) in decimal degrees.
- *+x_0* and *+y_0* are the false Easting and false Northing respectively. It defines a translation of the ENU frame with respect to the projection point.
- *+ellps=WGS84* defines the reference ellipsoid as WGS84.
- *+units=m* defines the coordinate units as metres.

The RCS of the display window should also be defined for in *Project/Project Properties/CRS*. The option *Enable 'on the fly' CRS transformation* must be activated in order to allow the projection of layers with different RCS to be displayed. Figure A.14 shows the same data as in Figure A.13 in WGS84 reference frame. The horizontal axis represents degrees of longitude and vertical axis represents degrees of latitude.

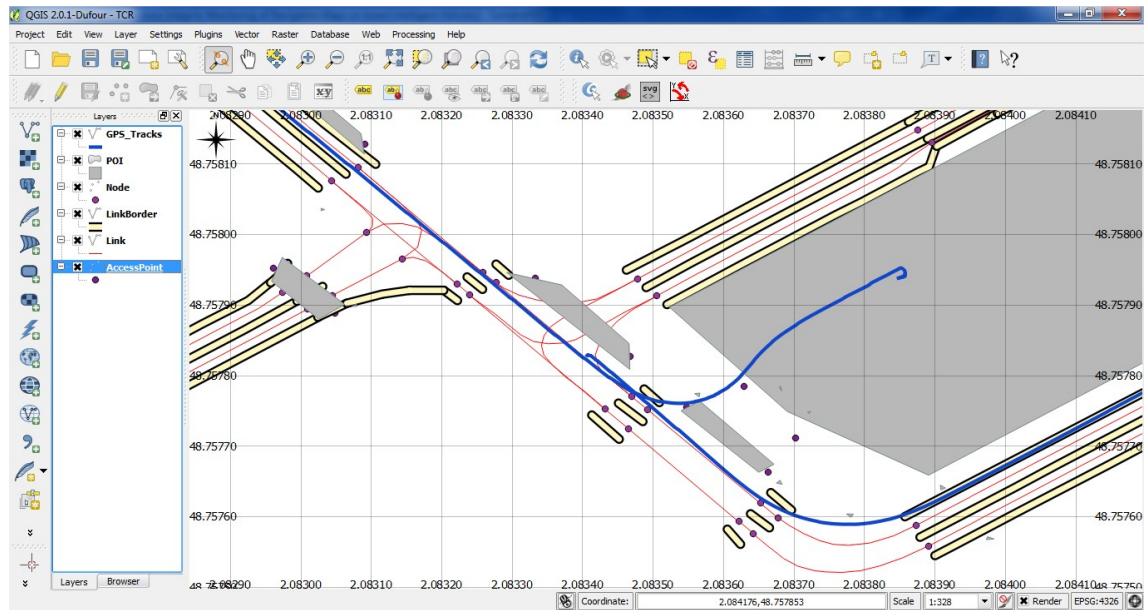


Figure A.14: Display of geographic data in WGS84 reference coordinate system. GPS tracks of the PAMU vehicle (blue line) on top of the test track map.

A.4.2.2 Format Conversion

The main benefit of QGIS for this thesis is its ability to handle many data formats. It is therefore possible to import data from various sources and export it for further uses in convenient format. For example, the test track map used in the PAMU project were developed by a supplier and delivered in a Spatialite Database. We intended to use this map as part of the experiments and therefore to use it in the Electronic Horizon software. However, this software uses as input OSM maps. No conversion from Spatialite to OSM is available so QGIS is used to do the conversion.

The Spatialite data is imported first in QGIS and the corresponding RCS is specified as introduced in the previous section. Since the export to OSM is not handled by QGIS, the Shapefile format is used. The layer is saved into a Shapefile (right click on the layer and *Save As*). As shown in Figure A.15, the RCS of the destination file must be specified as WGS84 to be correctly imported into other software (e.g., JOSM). The option *Skip attribute creation* should be activated in order to prevent failures during the use of the Shapefile by other GIS. This must be done for every layer.

Once saved as Shapefile, the data can be imported into JOSM (*File/Open*). The OpenData plug-in must have been installed to make JOSM able to read Shapefiles. Since the attributes were not created in the Shapefile, they must be added manually in JOSM. The principal attribute that defines road links is *highway*. The value of this attribute (e.g., *primary*, *secondary*, *track*) is chosen based on the road class

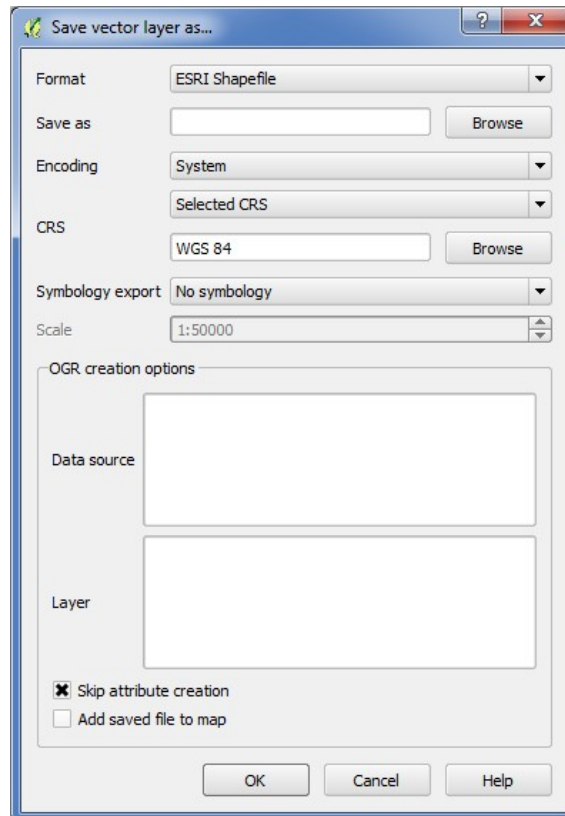


Figure A.15: Export of geographic information into Shapefile format.

but is not of major importance for the use of the navigation map in the Electronic Horizon software. Each Shapefile is imported as a separate layer into JOSM. A new layer must be created (*File/New Layer*) and the layers resulting from Shapefile import are merged (right click on the layer and select *Merge*). The result is shown in Figure A.16 and can finally be saved in OSM format (right click on the layer and select *Save As*).

In this process the geometry of the geographic data has not been modified so the resulting OSM file benefits from the same accuracy as the Spatialite database which was created by a mapping survey. The resulting OSM map is therefore considered as the reference (i.e. ground truth) map for the developments done in this thesis. The JOSM software also allows to locally modify the map to create map faults (for evaluation of the methods developed in this thesis) or include additional information to the map.

QGIS also allows to export the current view to vector image format which is especially convenient to create graphical illustrations based on the map or graphical user interfaces. The current view in QGIS can be exported in SVG (Scalable Vector Graphic) file which can be used in vector graphics editors like Adobe Illustrator and

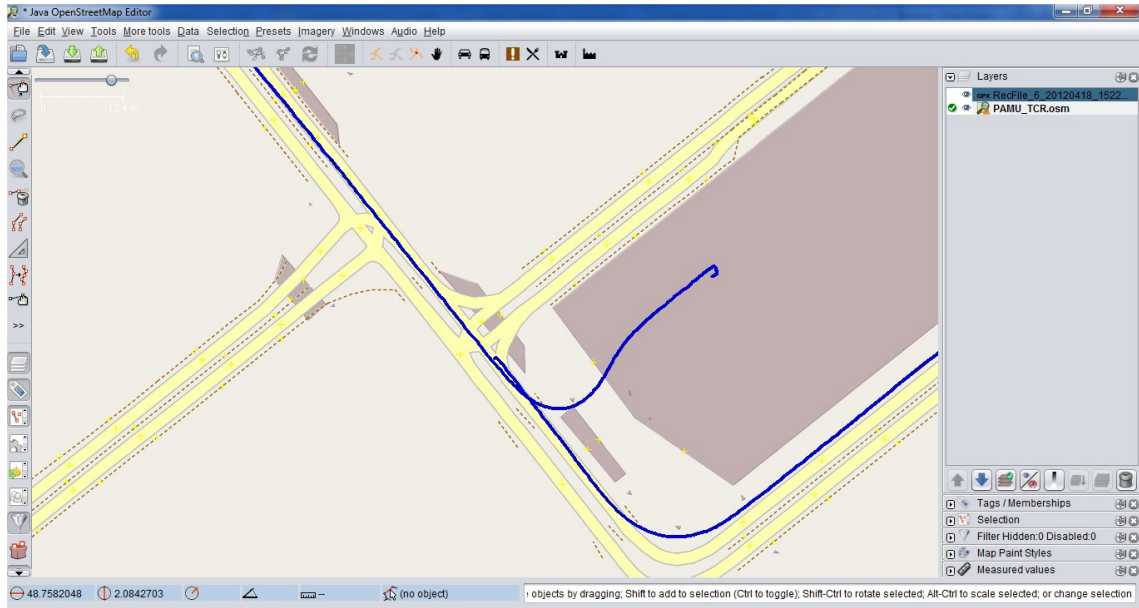


Figure A.16: Result of Shapefile import into JOSM.

Inkscape. The *SimpleSvg* plug-in must be installed in QGIS to add this functionality (*Plugins/Manage and Install Plugins*). The resulting file opened with Inkscape is shown in Figure A.17.

A.4.3 Map Faults Generator

Faulty navigation maps are required to evaluate the performances of the methods developed in this work. As introduced in Section A.3.2, a navigation system that uses an editable navigation map was developed for this thesis. Faults on the navigation map can therefore be intentionally created and injected into the navigation system. A method for automatic fault generation in OSM maps is described in this section.

The map faults are generated in two steps in order to represent the common map geometric faults in standard navigation maps: offset and sub-sampling.

A.4.3.1 Offset

The offsetting process consists in shifting a whole way according to a given vector. This therefore decreases the absolute accuracy but keeps the relative accuracy of the way. In order to keep some ways in the map from fault generation, a random process is used in order to determine which way will be affected by the offset. This is done using an uniform probability distribution and a user-defined threshold. For example, the probability of way offsetting can be set to 0.5.

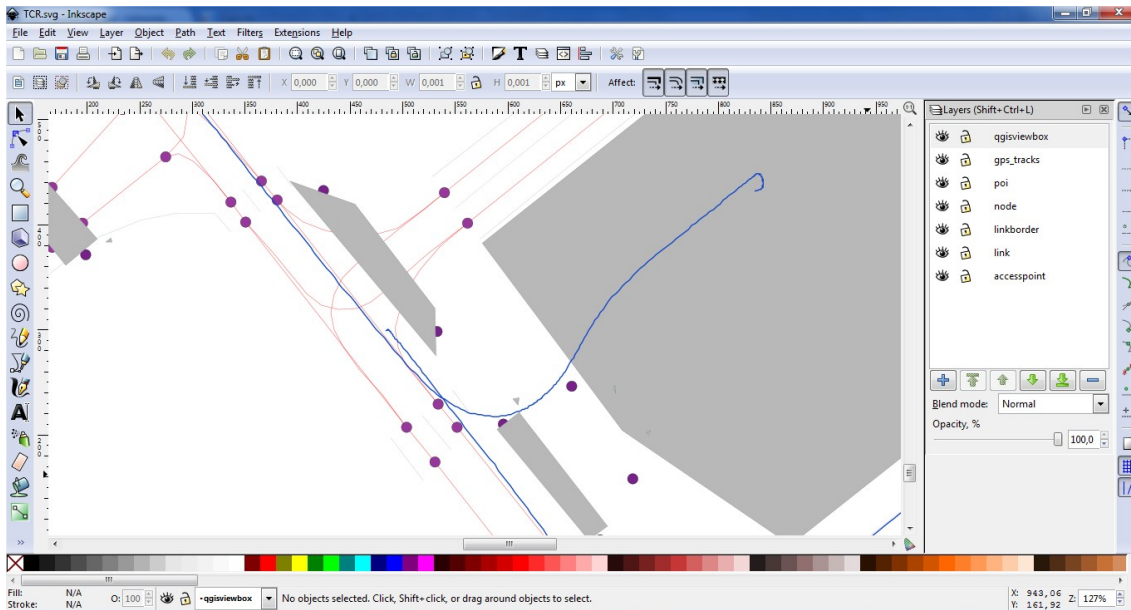


Figure A.17: PAMU map exported by QGIS in SVG and opened in Inkscape.

If this first random process indicated to offset the way, an offset vector is calculated based on a second random process. The latitude and longitude of this vector are random variables that follow a Gaussian distribution. The mean and the standard deviation of this distribution are chosen by the user before running the algorithm.

Once the offset vector is chosen, it is added to the coordinates of every nodes of the way.

A.4.3.2 Sub-sampling

The sub-sampling process consists in removing a defined ratio of nodes of each way. In order to keep the map topology, only shape nodes are considered in this process. Nodes that belong to more than one way (i.e. connection nodes) are indeed not removable by the sub-sampling. For example, one in three shape nodes are removed from the map. This decreases the relative accuracy of the way and simulates the map compilation process made by usual navigation map providers.

As detailed in Section A.2.1, ways are defined in the OSM map file by a ordered list of nodes (identified by their ID), the sub-sampling process therefore simply consists in deleting the desired node ID in the way definition.

A.4.3.3 Result Examples

The map faults generator is coded in C++ with Visual Studio 2010 and does not require any additional library. Figure A.18 shows an example of 5 maps resulting

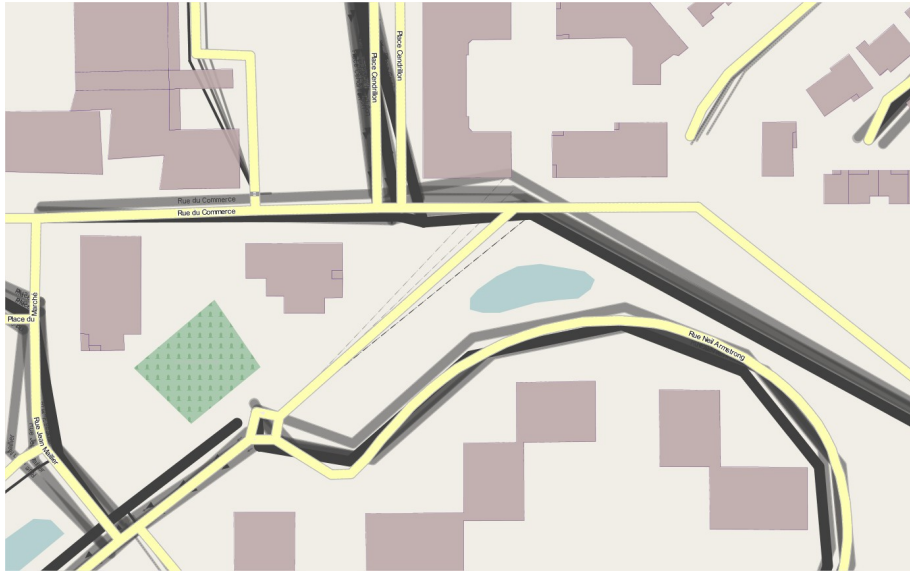


Figure A.18: Result of automatic generation of five faulty maps. The generated maps are in grey. The original map is in yellow.

from this program. The parameters used to create them are as follows:

- Offsetting probability: 0.5.
- Offset mean: 0 m.
- Offset standard deviation: 3 m.
- Sub-sampling ratio: one in two nodes.

Appendix B

Demonstration of the Rules of Non-Isolability

Contents

B.1 Introduction	163
B.2 Notations	164
B.3 Properties	165
B.4 New Problem Statement	166
B.5 Conclusion	168

B.1 Introduction

A Fault Detection, Isolation and Adaptation (FDIA) formalism is introduced in Chapter 4 and this appendix demonstrates the two rules of non isolability of Proposition 2.

The proposed FDIA framework assumes that one system observation produces two estimates of the same quantity: G and N . These estimates are stored in FDIA memory. K observations of the system are made.

Faults affecting G_i and N_i cause their value to be different from the true one P_i . The index i stands for the i^{th} observation. The state of G_i (resp. N_i) of being faulty or not is denoted by the boolean variable f_{G_i} (resp. f_{N_i}). For instance, $f_{G_i} = 1$ means that a fault affects G_i then $G_i \neq P_i$. At the K^{th} observation, the faulty states of every estimate is summarised by e called the set of faulty states. The purpose of FDIA is to isolate (i.e. determine) e which means to ascertain the faulty state of the estimates G_i and N_i for every observation $i \leq K$.

FDIA is based on the use of the residual vector $R(e)$. The terms of R are the results of boolean operations between the faulty states f_{G_i} and f_{N_i} . Some sets of faulty states produce unique residuals, isolation is then possible. However, some residuals are generated by several sets of faulty states, isolation is not possible.

The aim of this report is to prove the conditions on f_{G_i} and f_{N_i} for e to be not isolable. These are stated by two rules in Proposition 8:

Proposition 8. A set of faulty states is not isolable if and only if, it complies with one of the following rules:

1. $f_{N_i} = 1, \forall i \in \{1, \dots, K\}$ and $\exists! j \in \{1, \dots, K\}$ such as $f_{G_j} = 0$
2. $f_{G_i} = 1, \forall i \in \{1, \dots, K\}$

In other words it is not possible to isolate faults if:

1. Every estimates N is faulty and there is a unique true G .
2. Every G is faulty.

Section B.2 introduces the notations required for the demonstration. Section B.3 deduces mathematical properties of the proposed formalism. According to these properties, Section B.4 reformulates the problem and demonstrate the proposition. Finally Section B.5 concludes the demonstration.

B.2 Notations

A set of faulty states e is defined as $e = (f_{G_i}, f_{N_i})_{1 \leq i \leq K} \in \{0, 1\}^{2K}$ for some $K \geq 1$.

Let E be the set of sets of faulty states e .

$$e \in E \tag{B.1}$$

R is the function that associates a residual to a set of faulty states. $R(e)$ is the residual of e .

$$R = \left\{ r_{G_i G_j}, r_{N_i N_j}, r_{G_p N_q} \right\}_{1 \leq i, j, p, q \leq K, i \neq j}$$

with

$$r_{G_i G_j} = f_{G_i} \vee f_{G_j}, \forall i, j \in \{1, \dots, K\}, i > j \tag{B.2}$$

$$r_{G_i N_j} = f_{G_i} \vee f_{N_j}, \forall i, j \in \{1, \dots, K\} \tag{B.3}$$

$$r_{N_i N_j} = f_{N_i} \oplus f_{N_j}, \forall i, j \in \{1, \dots, K\}, i > j \quad (\text{B.4})$$

where \vee and \oplus are boolean *or* and *exclusive or* respectively.

σ_K is the set of permutations of $\{1, \dots, K\}$

I stands for the set of isolable sets of faulty states.

$$I \subset E \quad (\text{B.5})$$

I^c is the complement of I (i.e. the set of non-isolable sets of faulty states)

$C_{l,m}$ is the set of sets of faulty states $e = (f_{G_i}, f_{N_i})_{1 \leq i \leq K}$ such as there are l $f_{G_i} = 1$ and m $f_{N_i} = 1$.

Given a set of faulty states $e \in E$ and two permutations $\sigma, \sigma' \in \sigma_K$, the set of faulty states obtained by permuting variables f_{G_i} (with σ) and variables f_{N_i} (with σ') is denoted by

$$(\sigma, \sigma') \cdot e = (f_{G_{\sigma^{-1}i}}, f_{N_{\sigma'^{-1}i}})_{1 \leq i \leq K} \quad (\text{B.6})$$

where σ^{-1} (resp. σ'^{-1}) stands for the inverse permutation of σ (resp. σ').

$e_{l,m}$ stands for canonical form of $e \in C_{l,m}$. $e_{l,m} = (f_{G_i}, f_{N_i})_{1 \leq i \leq K}$ with

$$\left\{ \begin{array}{l} f_{G_i} = \left(\underset{l}{1, \dots, 1}, \underset{K-l}{0, \dots, 0} \right) \\ f_{M_i} = \left(\underset{m}{1, \dots, 1}, \underset{K-m}{0, \dots, 0} \right) \end{array} \right. \quad (\text{B.7})$$

B.3 Properties

Proposition 9. *The class $C_{l,m}$ contains all the permutations of the canonical set of faulty states $e_{l,m} : C_{l,m} = \{(\sigma, \sigma') \cdot e_{l,m} | \sigma, \sigma' \in \sigma_K\}$*

Proof. The permutation σ (resp. σ') doesn't change the number of f_{G_i} (resp. f_{N_i}) equals to one. Thus, if $e \in C_{l,m}$ and $\sigma, \sigma' \in \sigma_K$ then $(\sigma, \sigma') \cdot e \in C_{l,m}$. Reciprocally, every set of faulty states $e \in C_{l,m}$ can be obtained by permuting the variables f_{G_i} and f_{N_i} and the proposition follows. \square

Remark 1. $E = \bigcup_{1 \leq l, m \leq K} C_{l,m}$

$C_{l,m}$, $1 \leq l, m \leq K$ is a partition of E .

Let denote by \tilde{E} the disjoint union of $C_{l,m}$ where $(l, m) \neq (K-1, K)$ and $(l, m) \neq (K, K)$

$$\tilde{E} = \bigcup_{1 \leq l, m \leq K} C_{l,m} \mid (l, m) \neq (K-1, K) \text{ and } (l, m) \neq (K, K) \quad (\text{B.8})$$

Proposition 10. *If $e \in E$ and $\sigma, \sigma' \in \sigma_K$ then*

$$e \in I \iff (\sigma, \sigma') \cdot e \in I \quad (\text{B.9})$$

Proof. $e \notin I \implies \exists e' \neq e \mid R(e') = R(e)$. For a pair of permutations $\sigma, \sigma' \in \sigma_K$, then $(\sigma, \sigma') \cdot e \neq (\sigma, \sigma') \cdot e'$ and $R((\sigma, \sigma') \cdot e) = R((\sigma, \sigma') \cdot e')$; then $(\sigma, \sigma') \cdot e \notin I$.

Reciprocally, if $(\sigma^{-1}, \sigma'^{-1}) \cdot e \notin I$, then there exists $(\sigma^{-1}, \sigma'^{-1}) \cdot e' \in E$ such as $(\sigma^{-1}, \sigma'^{-1}) \cdot e' \neq (\sigma^{-1}, \sigma'^{-1}) \cdot e$ and $R((\sigma^{-1}, \sigma'^{-1}) \cdot e') = R((\sigma^{-1}, \sigma'^{-1}) \cdot e)$. One can apply the same permutation (σ, σ') to those sets of faulty states.

$$(\sigma, \sigma') \cdot (\sigma^{-1}, \sigma'^{-1}) \cdot e' \neq (\sigma, \sigma') \cdot (\sigma^{-1}, \sigma'^{-1}) \cdot e$$

$$\text{and } R((\sigma, \sigma') \cdot (\sigma^{-1}, \sigma'^{-1}) \cdot e') = R((\sigma, \sigma') \cdot (\sigma^{-1}, \sigma'^{-1}) \cdot e)$$

$$\implies e' \neq e \text{ and } R(e') = R(e)$$

$$\implies e \notin I$$

□

If a set of faulty states e of E is isolable, then every permutation of e is isolable. More precisely, the following proposition holds :

Corollary 1. *For all $e \in C_{l,m}$, $e \in I \iff C_{l,m} \in I$*

This is true in particular for the canonical set of faulty states $e_{l,m}$ of class $C_{l,m}$. The canonical set of faulty states describes it entire class in terms of isolability.

B.4 New Problem Statement

According to the previous developments, studying isolability of a set of faulty states is equivalent to evaluating the isolability of the canonical set of faulty states of

every class. The residuals are calculated using boolean operations between f_G and f_N variables. As stated in Section B.2, the *OR* operator is used for $f_G f_G$ and $f_G f_N$ pairs combination and *Exclusive OR* is used for $f_N f_N$ combination. The canonical sets of faulty states of class $C_{l,m}$ are represented in the Tables B.1a, B.1b and B.1c. For f_G (resp. f_N), the l (resp. m) ones are written first and the $K - l$ (resp. $K - m$) zeros are written then. The the result of the boolean operation is written in the table which forms the residual.

These tables offer the advantage of showing clearly the consequence of the parameters l and m on the residual of a set of faulty states. It has been shown previously that the isolability of a canonical set of faulty states is the same as the class it belongs to. The isolability study of a class is made by looking at the number of zeros in the tables with respect to l and m .

B.4.1 If $l, m < K$

If $l < K$ and $m < K$, there is a rectangle of zeros in Table B.1c of size $n_c = (K - l) \cdot (K - m) > 0$. Assume that $e \in E$ is another set of faulty states with the same residual as $e_{l,m}$. Looking at Table B.1c we see that the $K - l$ last variables f_{G_i} and the $K - m$ last variables N_i are necessarily 0. Then, because of the two sub-tables made of 1, we see that all the others variables are 1. In other words, $e = e_{l,m}$. As a consequence, we have:

$$C_{l,m} \subset I, \forall l, m \in \{1, \dots, K - 1\} \quad (\text{B.10})$$

B.4.2 If $l = K$

If $l = K$, Tables B.1a and B.1c are full of ones for all m and Table B.1b is the only one which can make a difference in the residual. Moreover, $N_i \oplus N_j = \overline{N_i} \oplus \overline{N_j} \forall i \neq j$. Thus, for all $m \in \{1, \dots, K\}$ and $e = (f_{G_i}, f_{N_i}) \in C_{l,m}$, replacing N_i by $\overline{N_i}$ doesn't change the residual of e . Hence:

$$C_{K,m} \subset I^c, \forall m \in \{1, \dots, K\} \quad (\text{B.11})$$

B.4.3 If $m = K$

If $m = K$ and for all l , the Table B.1b is full of zeros and the Table B.1c is full of ones. Then the most significant table is Table B.1a.

It must be noticed first that, $l = K$ and $l = K - 1$ makes Table B.1a be full of ones. $C_{K,K}$ and $C_{K-1,K}$ have therefore the same residual. Then:

$$C_{K,K} \subset I^c \tag{B.12}$$

$$C_{K-1,K} \subset I^c \tag{B.13}$$

Secondly, if $l \leq K - 2$, there are $(K - l)^2 - (K - l) > 0$ zeros in Table B.1a. Assume that e is a set of faulty states with the same residual as $e_{l,K}$. For the same reasons as in the case B.4.1, this implies that the l first variables f_{G_i} are 1 and the others are 0: the variables f_{G_i} of e are the same as f_{G_i} of $e_{l,K}$. Moreover, in view of Table B.1c, we see that all the variables f_{N_i} of e are 1 or else a 0 would appear in Table B.1a corresponding to the residual of e . So, $e = e_{l,K}$ and we have shown that $e_{l,K}$ is isolable. This is sufficient to obtain the following inclusion:

$$C_{l,K} \subset I, \forall l \in \{1, \dots, K - 2\} \tag{B.14}$$

B.5 Conclusion

It was demonstrated that:

$$I^c = C_{K-1,K} \cup \bigcup_{l=0}^K C_{K,l} \tag{B.15}$$

In other words, a set of faulty states is not isolable if and only if, it complies to one of the following rules:

- $N_j = 1, \forall j \in \{1, \dots, K\}$ and $\exists ! i \in \{1, \dots, K\}$ such as $G_i = 0$
- $G_i = 1, \forall i \in \{1, \dots, K\}$

Which concludes the demonstration. TBS¹.

¹Terminé Bonsoir

Table B.1: Residual generation

(a) $f_G f_G$ residual generation

\vee f_G	f_G	l				$K-l$			
		1	1	...	1	0	0	...	0
l	1	n.a.	1	...	1	1	1	...	1
	1	1	n.a.	...	1	1	1	...	1
	\vdots	\vdots	\ddots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
	1	1	1	...	n.a.	1	1	...	1
$K-l$	0	1	1	...	1	n.a.	0	...	0
	0	1	1	...	1	0	n.a.	...	0
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
	0	1	1	...	1	0	0	...	n.a.

(b) $f_N f_N$ residual generation

\oplus f_N	f_N	m				$K-m$			
		1	1	...	1	0	0	...	0
m	1	n.a.	0	...	0	1	1	...	1
	1	0	n.a.	...	0	1	1	...	1
	\vdots	\vdots	\ddots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
	1	0	0	...	n.a.	1	1	...	1
$K-m$	0	1	1	...	1	n.a.	0	...	0
	0	1	1	...	1	0	n.a.	...	0
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
	0	1	1	...	1	0	0	...	n.a.

(c) $f_G f_N$ residual generation

\vee f_G	f_N	m		$K-m$			
		1	...	1	0	...	0
l	1	1	...	1	1	...	1
	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
	1	1	...	1	1	...	1
$K-l$	0	1	...	1	0	...	0
	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
	0	1	...	1	0	...	0

Appendix C

Automatic Truth Tables Generation for FDIA Implementation

This appendix provides the Matlab code for automatically generating the truth tables used in the FDIA framework presented in Chapter 4. The number of vehicle trips for which the truth table is created is set at the beginning of the program. In order to facilitate the use of the table in the program, two tables are created. First, The residual table contains every possible residual and which are identified by a unique ID. This table also indicates whether the residual is distinguishing (i.e. not adverse). Second, the diagnosis table contains the residual ID and the corresponding sets of faults. These tables are created in a SQL database.

Matlab Program

```
1 clear all
2 close all
3 clc
4
5 % C.Z. 2013
6
7
8 conn = database('test','root','utcrenault');
9
10 % Number of vehicle trips:
11 K=5;
12
13 % number of columns in diagnostic and residual tables
14 n_diag = 2*K + 2;
15 n_res = K*(2*K - 1) + 2;
16
17 % number of rows in diagnostic and residual tables
18 m_diag = 2^(2*K);
19 m_res = 2^(2*K);
20
21 % table names
22 name_diag = ['k' num2str(K-1) '_diagnostic_auto'];
23 name_res = ['k' num2str(K-1) '_residuals_auto'];
24
25
26 %% columns names
27 MG = cell(0);
28 MG_descr = cell(0);
29 for i=0:2*K-1
30     if mod(i,2)==0 % si i est paire
31         MG{end+1} = ['G' num2str(i/2)];
32     else
33         MG{end+1} = ['M' num2str(floor(i/2))];
34     end
35
36     MG_descr{end+1} = 'INT(1)';
37
38 end
39 head_diag = [{'id' 'residual_id'} MG];
40 descr_diag = [{'INT(1)' 'UNSIGNED' 'NOT NULL' 'AUTO_INCREMENT' 'INT(1)'} MG_descr];
41
```

Appendix C Automatic Truth Tables Generation for FDIA Implementation

```

42
43 R = cell(0);
44 operateur = cell(0);
45 R_descr = cell(0);
46 for i=1:2*K
47     for j=i+1:2*K
48         R{end+1} = ['r_', MG{i} '_', MG{j}];
49         if (mod(i,2)==0 && mod(j,2)==0)
50             operateur{end+1} = 'xor';
51         else
52             operateur{end+1} = 'or';
53         end
54         R_descr{end+1} = 'INT(1)';
55     end
56 end
57 head_res = [{'id' 'distinguishing'} R];
58 descr_res = [{'INT(1) UNSIGNED NOT NULL AUTO_INCREMENT' 'INT(1)'} R_descr];
59
60
61 %% Tables creation
62 % diagnostic
63 req_create_diag = ['CREATE TABLE IF NOT EXISTS' ' ' name_diag ' ']; % head_diag]; % ' PRIMARY KEY
64 (' head_diag{1} ') ENGINE=INNODB;';
65
66 for i=1:length(head_diag)
67     req_create_diag = [req_create_diag head_diag{i} ' ' descr_diag{i} ' , '];
68 end
69
70 req_create_diag = [req_create_diag ' PRIMARY KEY (' head_diag{1} ') ENGINE=INNODB;'];
71
72 exec(conn, req_create_diag);
73
74
75 % residual signatures
76 req_create_res = ['CREATE TABLE IF NOT EXISTS' ' ' name_res ' '];
77
78 for i=1:length(head_res)
79     req_create_res = [req_create_res head_res{i} ' ' descr_res{i} ' , '];
80 end
81
82 req_create_res = [req_create_res ' PRIMARY KEY (' head_res{1} ') ENGINE=INNODB;'];
83
84 exec(conn, req_create_res);
85
86
87 %% filling the tables
88 % diagnostic
89
90 for i=1:m_diag
91     erreur = cell(0);
92     word = num2str(dec2bin(i-1));
93     n_missing_zeros = 2*K - length(word);
94     for z=length(word)+1 : 2*K
95         erreur{z} = '0';
96     end
97     for z=length(word) : -1 : 1
98         erreur{z} = word(end-z+1);
99     end
100
101     req_fill_diag = ['INSERT INTO' , name_diag , ' VALUES (' , num2str(i) , ' , NULL'];
102     for j=1:length(erreur)
103         req_fill_diag = [req_fill_diag , ' , ' erreur{j}];
104     end
105     req_fill_diag = [req_fill_diag , ')'];
106
107     exec(conn, req_fill_diag);
108 end
109
110 %% residual signatures
111 curs = exec(conn, ['SELECT * FROM' , name_diag , ' ORDER BY id;']);
112 setdbprefs('DataReturnFormat', 'numeric');
113 curs = fetch(curs);
114 diagnostic = curs.Data(:, 3:end);
115
116 i_signature = 1;
117
118 for i=1:m_res % for each row...
119     req_res = ['INSERT INTO' , name_res , ' VALUES (' , num2str(i) , ' , NULL '];
120     % ...the signature is built
121     signature = cell(0);
122     i_operateur = 1;

```

```

131     for j=1:length(MG)-1
132         for k=j+1 : length(MG)
133
134             if strcmp(operateur{i_operateur} , 'or')
135
136                 signature = [signature , num2str(or(diagnostic(i,j) , diagnostic(i,k))) ];
137
138             elseif strcmp(operateur{i_operateur} , 'xor')
139
140                 signature = [signature , num2str(xor(diagnostic(i,j) , diagnostic(i,k))) ];
141
142             end
143             i_operateur = i_operateur + 1;
144         end
145     end
146 end
147
148
149 % we determine whether the signature is unique
150 recherche_signature = ['SELECT_*_FROM_' , name_res , '_WHERE_'];
151 for j=1 : length(signature)
152     recherche_signature = [recherche_signature , R{j} , '=' , signature{j}];
153
154     if j<length(signature)
155         recherche_signature = [recherche_signature , '_AND_'];
156     end
157 end
158 recherche_signature = [recherche_signature , ';'];
159
160 curs = exec(conn,recherche_signature);
161 setdbprefs('DataReturnFormat','numeric');
162 curs=fetch(curs);
163
164 if strcmp(curs.data,'NoData')
165     % the row is added in the table residual with the value discriminating = 1
166     insert_signature = ['INSERT INTO_' , name_res , '_VALUES_(' , num2str(i_signature) , '_,' ,
167         '_'];
168     for j=1: length(signature)
169         insert_signature = [insert_signature , ',' , signature{j}];
170     end
171     insert_signature = [insert_signature , ')'];
172     exec(conn,insert_signature);
173
174     % the ID is updated in the diagnostic table
175     update_diag = ['UPDATE_' , name_diag , '_SET_residual_id_=' , num2str(i_signature) , '_ ' ,
176         'WHERE_id_=' , num2str(i) , ';'];
177     exec(conn,update_diag);
178
179     i_signature = i_signature + 1;
180 else
181     % the row is not added in the table but we change discriminating => 0 in the existing
182     row
183     update_signature = ['UPDATE_' , name_res , '_SET_distinguishing_=' , num2str(
184         (curs.data(1,1)) , ';');
185     exec(conn,update_signature);
186     % the ID is updated in the diagnostic table
187     update_diag = ['UPDATE_' , name_diag , '_SET_residual_id_=' , num2str(curs.data(1,1)) , ' ' ,
188         'WHERE_id_=' , num2str(i) , ';'];
189     exec(conn,update_diag);
190 end
191 end
192
193 %% sort the columns of table residuals in order to keep the column order from table K-1 to table
194 K
195
196 if K==2
197     req1 = 'ALTER TABLE_' , test , 'k2_residuals_auto' , '_CHANGE_COLUMN_' , 'r_MO_G1' , '_ ' , 'r_MO_G1' , '_int
198         (1)_NULL_DEFAULT_NULL_AFTER_' , 'r_GO_M1' , ';';
199     exec(conn,req1);
200     req2 = 'ALTER TABLE_' , test , 'k2_residuals_auto' , '_CHANGE_COLUMN_' , 'r_MO_M1' , '_ ' , 'r_MO_M1' , '_int
201         (1)_NULL_DEFAULT_NULL_AFTER_' , 'r_MO_G1' , ';';
202     exec(conn,req2);
203     req3 = 'ALTER TABLE_' , test , 'k2_residuals_auto' , '_CHANGE_COLUMN_' , 'r_G1_M1' , '_ ' , 'r_G1_M1' , '_int
204         (1)_NULL_DEFAULT_NULL_AFTER_' , 'r_MO_M1' , ';';
205     exec(conn,req3);
206
207     disp('Columns sorted for K=2!');
208 end
209
210 if K==3
211     req1 = 'ALTER TABLE_' , test , 'k3_residuals_auto' , '_CHANGE_COLUMN_' , 'r_MO_G1' , '_ ' , 'r_MO_G1' , '_int

```

Appendix C Automatic Truth Tables Generation for FDIA Implementation

```
212         (1) NULL DEFAULT NULL AFTER 'r_G0_M1';';
213     exec(conn, req1);
214     req2 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_M0_M1' 'r_M0_M1' int
215         (1) NULL DEFAULT NULL AFTER 'r_M0_G1';';
216     exec(conn, req2);
217     req3 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G1_M1' 'r_G1_M1' int
218         (1) NULL DEFAULT NULL AFTER 'r_M0_M1';';
219     exec(conn, req3);
220     req4 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G0_G2' 'r_G0_G2' int
221         (1) NULL DEFAULT NULL AFTER 'r_G1_M1';';
222     exec(conn, req4);
223     req5 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G0_M2' 'r_G0_M2' int
224         (1) NULL DEFAULT NULL AFTER 'r_G0_G2';';
225     exec(conn, req5);
226     req6 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_M0_G2' 'r_M0_G2' int
227         (1) NULL DEFAULT NULL AFTER 'r_G0_M2';';
228     exec(conn, req6);
229     req7 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_M0_M2' 'r_M0_M2' int
230         (1) NULL DEFAULT NULL AFTER 'r_M0_G2';';
231     exec(conn, req7);
232     req8 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G1_G2' 'r_G1_G2' int
233         (1) NULL DEFAULT NULL AFTER 'r_M0_M2';';
234     exec(conn, req8);
235     req9 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G1_M2' 'r_G1_M2' int
236         (1) NULL DEFAULT NULL AFTER 'r_G1_G2';';
237     exec(conn, req9);
238     req10 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_M1_G2' 'r_M1_G2' int
239         (1) NULL DEFAULT NULL AFTER 'r_G1_M2';';
240     exec(conn, req10);
241     req11 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_M1_M2' 'r_M1_M2' int
242         (1) NULL DEFAULT NULL AFTER 'r_M1_G2';';
243     exec(conn, req11);
244     req12 = 'ALTER TABLE 'test'.'k3_residuals_auto' CHANGE COLUMN 'r_G2_M2' 'r_G2_M2' int
245         (1) NULL DEFAULT NULL AFTER 'r_M1_M2';';
246     exec(conn, req12);
247     disp('Columns sorted for K=3!');
248 end
```

