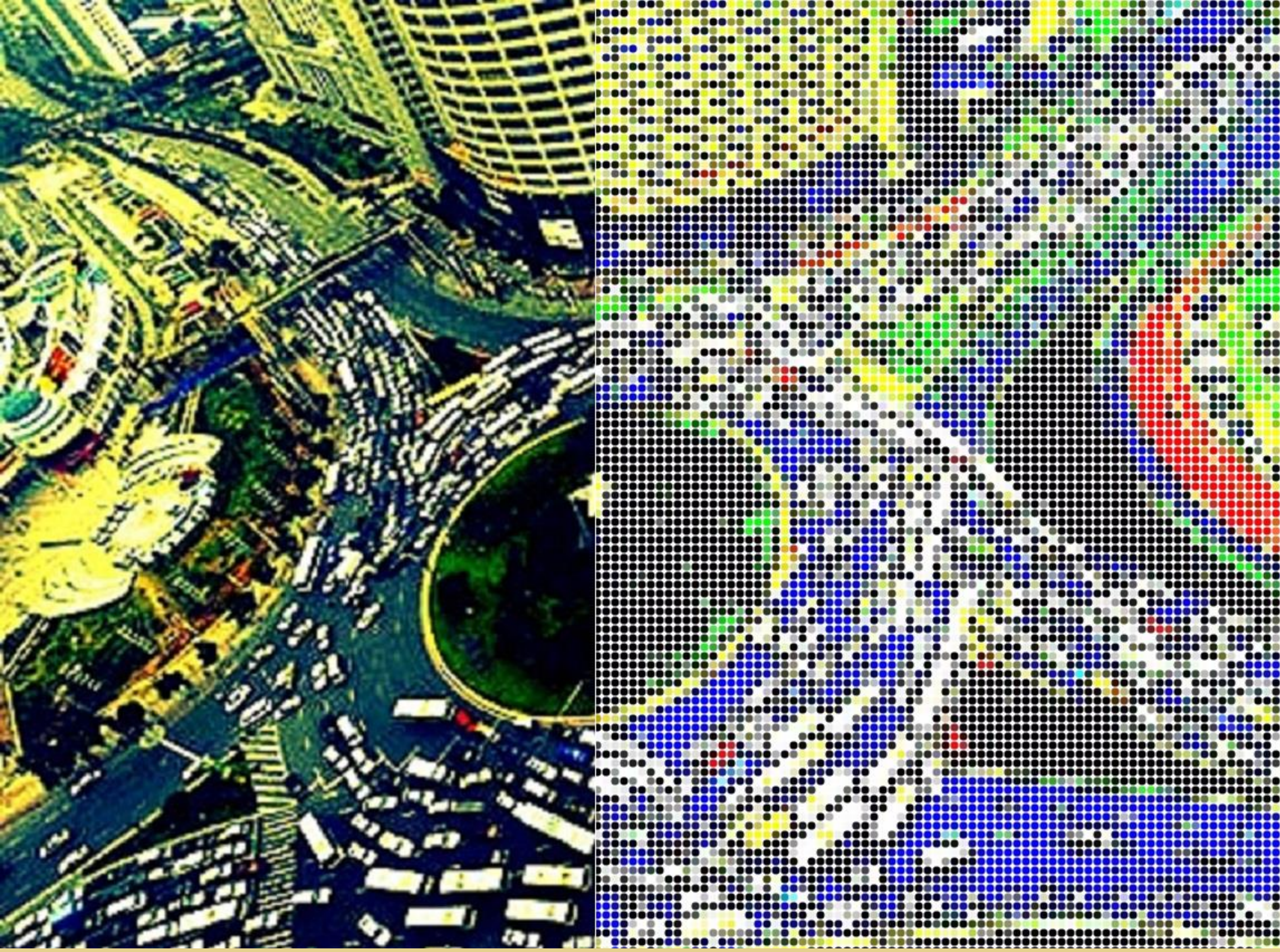


Luís Manuel Conde Bento

Intelligent Traffic Management Algorithms for Environmental Impacts Reduction at city centers aided by GNSS Positioning

Tese de doutoramento em Engenharia Eletrotécnica e de Computadores, ramo de especialização em Automação e Robótica, orientada pelo Professor Doutor Urbano José Carreira Nunes e apresentada ao Departamento de Engenharia Eletrotécnica de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Junho/2016

• U C •

UNIVERSIDADE DE COIMBRA

Intelligent Traffic Management Algorithms for Environmental impacts reduction at city centers aided by GNSS Positioning

Luís Manuel Conde Bento

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Coimbra, Portugal

under supervision of

Prof. Dr. Urbano Nunes (advisor)

Acknowledgments

When I started this journey, the *”Intelligent Transportation Systems”* topic was a new area of research for me. To get here, I received many direct and indirect contributions that helped me to overcome the challenges of this multidisciplinary research area.

Undoubtedly, my first thanks go to my supervisor Professor Urbano Nunes. I owe to Professor Urbano Nunes much of my training as a student and researcher, he has accompanied me since Bachelor student in 2000. The guidance of Professor Urbano Nunes was excellent, thank you for your methodological contributions, enlightening discussions, continuous motivation and indispensable friendship to overcome the most difficult phases of the development work. I also appreciate the freedom that he gave me to follow my own ideas.

To Professors Hesham Rakha and Philippe Bonnifait, I want to thank them for the fruitful exchange of knowledge and ideas as well as methodological contributions.

I would also like to give my thanks to the Ricardo Parafita, Sergio Santos and Rafael Chelim for their valuable work in the implementation of algorithms and construction of *”datasets”*.

This thesis results from research work conducted at the Institute of Systems and Robotics (ISR), University of Coimbra. I thank ISR for the excellent facilities made available, and the Polytechnic Institute of Leiria for the support provided within the scholarship PROTEC SFRH/BD/50066/2009. This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) and COMPETE under grants PTDC/SEN-TRA/099413/2008 (EVSIM09 Project) and UID/EEA/00048/2013. We also acknowledge the support of Project B-Mobility4People (QREN-MaisCentro SCT-2011-01).

For the constant concern of my wellbeing, encouragement and support I thank my parents, my sister, in-laws and nephews. To my grandparents for serving as a model in their tenacity, determination and perseverance, a thank posthumously.

Although there is a number of friends and family to whom I must thank as they contributed to be able to reach to this point, I can not mention them all, to them the most sincere thanks and the certainty that they were not forgotten.

Finally a special word to my wife, Filipa, for her love and altruism, especially in difficult times, and to the lights of my life, my daughters Inês and Mariana, for their unconditional love.

Abstract

Traffic congestion in city centers and its associated environment impact are among the major challenges pose to mobility researcher nowadays. This thesis aims to contribute for urban mobility improvement, in terms of road capacity and vehicle emissions. The current work is included in the Intelligent Transportation Systems (ITS) research area, namely traffic simulation of autonomous vehicles and CO_2 emissions assessment at a microscopic level. The thesis focus on the development of methodologies for the traffic management in intersections aided by precise multi-sensor positioning. This research explores a simulation framework for the development of Intelligent Traffic Management (ITM) algorithms in road intersections, namely roundabout and crossroads. The research work provides an assessment of novel intersection traffic management algorithms for automated vehicles on fuel consumption and greenhouse gas emissions of CO_2 relative to traditional traffic signal and roundabout intersection control. An accurate vehicle's pose estimation and control is of extreme importance for the inner working of the ITM algorithms, therefore its development is also performed and analyzed. The research work is divided in three major topics: simulation framework, ITM algorithms and vehicle pose estimation.

In the framework of this research the ISR - Traffic Simulator (ISR-TRAFSIM) was developed as the simulation environment. It is an open-source Matlab-based simulator and it has the potential for being used in diverse studies such has: automatic traffic control, vehicle emissions analysis, vehicle path-following control and sensor fusion for vehicle awareness (available at: `http://www2.isr.uc.pt/~conde/isr-trafsim/`).

Several ITM algorithms were developed, aiming to minimize accidents, traffic congestion and consequently the environmental costs of road traffic. The developed intelligent traffic management ITM algorithms, which are based on a spatio-temporal reservation scheme, ensure that vehicles proceed through the intersection without colliding with other vehicles while at the same time reducing the intersection delay and environmental impacts. This research demonstrates that there is the potential of emission reduction and optimized traffic flow through the use of these ITM algorithms.

To implement the traffic management in cooperative driving scenarios, the estimated pose and speed of vehicles must be both accurate and robust. When vehicles localization systems involve standalone Global Navigation Satellite System (GNSS) receivers, the resulting accuracy can be affected by satellite-specific errors of several meters. This work studied how road-features like lane marking detected by on-board cameras can be exploited to reduce absolute position errors of cooperative vehicles sharing information in real-time in a network. The algorithms considered in this work are based on a error bounded set membership strategy, these methodologies were used to compute GNSS guaranteed risk integrity zone. A robust set-inversion based algorithm was also developed, to solve the problem of Non Line Of Sight (NLOS) multipath and its error propagation when using collaborative algorithms.

Resumo

O congestionamento do tráfego no centro das cidades e impacto ambiental associado estão entre os principais desafios colocados hoje em dia aos investigadores na área da mobilidade. Este trabalho tem como objetivo contribuir para a melhoria da mobilidade urbana, em termos do aumento da capacidade das infraestruturas rodoviárias e da redução das emissões dos veículos. O presente trabalho está incluído no domínio dos *"Sistemas de Transportes Inteligentes"*, nomeadamente na simulação de tráfego de veículos autónomos e análise das emissões de CO_2 ao nível microscópico. O trabalho incidirá sobre o desenvolvimento de metodologias para a gestão de tráfego em interseções, auxiliadas por um sistema de localização multi-sensorial. Um ambiente de simulação integrado é analisado no contexto do desenvolvimento de algoritmos *"Inteligentes de Gestão de Tráfego"* (Intelligent Traffic Management-ITM) aplicados em interseções, nomeadamente em rotundas e e cruzamentos. Os resultados deste trabalho, disponibilizam dados relevantes sobre consumo de combustível e sobre as emissões de gases de efeito estufa CO_2, habilitando a avaliação dos algoritmos inteligentes de gestão de tráfego em interseções para veículos autónomos, relativamente às técnicas tradicionais de regulação de trânsito em rotundas e cruzamentos. A estimação precisa da localização de um veículo e o controlo do mesmo, é de extrema importância para o desenvolvimento e operação de algoritmos ITM, portanto, sendo o seu desenvolvimento também realizado e analisado. O trabalho desenvolvido encontra-se dividido em três grandes temas: ambiente de simulação integrado, algoritmos ITM e estimação da localização de veículos

.

No âmbito deste trabalho foi desenvolvido um ambiente de simulação denominado ISR-TRAFSIM. Este simulador é baseado em Matlab, o seu código fonte encontra-se disponível, e tem o potencial para ser usado em estudos diversos na área dos transportes inteligentes, nomeadamente: controlo automático de tráfego rodoviário, análise de emissões de veículos automóveis, controlo de seguimento de trajetória de veículos autónomos e de fusão de sensorial para efeitos de localização precisa e percepção de situações anómalas ou perigosas (disponível em: `http://www2.isr.uc.pt/~conde/isr-trafsim/`).

Vários algoritmos ITM foram desenvolvidos, visando minimizar o número de acidentes, o congestionamento do tráfego e consequentemente reduzir os custos ambientais do tráfego rodoviário. Os algoritmos ITM desenvolvidos, são baseados em metodologias de reserva espaço-temporal. Estas metodologias asseguram que os veículos se deslocam através do cruzamento sem colidir com outros veículos, ao mesmo tempo que conseguem reduzir o tempo de espera no cruzamento assim como os seus impactos ambientais. Este trabalho demonstra que existe o potencial para a redução de emissões e aumento do fluxo de tráfego através da utilização de algoritmos ITM.

Para implementar a gestão de tráfego em cenários de condução cooperativa, a estimativa da localização e velocidade dos veículos deve ser precisa e robusta. Em sistemas localização de veículos baseados apenas em receptores GNSS, a precisão resultante pode ser afetada por erros de vários metros. Neste trabalho avaliou-se a potencial melhoria na localização de veículos equipados câmaras de detecção da linha de marcação lateral da estrada, em cenários cooperativos para reduzir os erros de posição absoluta de veículos através da partilha em rede de informação em tempo real. As técnicas utilizadas nos algoritmos de localização são baseadas em métodos intervalares, usados para determinar conjuntos de pertença com erro delimitado, estas metodologias foram usadas para determinar localização baseada em GNSS, com garantia da solução se encontrar dentro de um conjunto com um determinado risco. Para resolver o problema das *”pseudoranges”* com multi-percurso do tipo NLOS e propagação do erro quando se utiliza algoritmos colaborativos, desenvolveram-se algoritmos robustos, baseados na detecção de medidas aberrantes de satélites, e no relaxamento das restrições usadas na computação do conjunto de pertença.

Contents

List of Tables

List of Figures

List of Acronyms

2WS	Two Wheel Steering
4WD	Four Wheel Drive
4WS	Four Wheel Steering
ADAS	Advanced Driver Assistance Systems
API	Application Program Interface
APP	application layer
BeiDou	BeiDou Satellite Navigation System
CAN-BUS	Controller Area Network Bus
CG	Center of Gravity
CI	Crossroad Intersections
CIITM	Car Intention Intelligent Traffic Management
CP	Control Point
CFC	Chained Form based path following Controller
CS	Communication System
DDM	Dual Driving Mode
DFCO	Deceleration Fuel Cut Off
DGPS	Differential GPS
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EKF	Extended Kalman Filter
EMITM	Early Method Intelligent Traffic Management
ENU	East North Up
FLC	Fuzzy-Logic based path following Controller
FMITM	Forward Method for Intelligent Traffic Management
GDOP	Geometric Dilution of Precision

GIS	Geographic Information Systems
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GOMNE	Guaranteed Minimum Outlier Number Estimator
GPS	Global Positioning System
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HAL	horizontal alarm limit
HCI	Human Computer Interaction
HPE	horizontal position error
HPL	Horizontal Protection Level
HMI	Hazardously Misleading Information
IDM	Intelligent Driver Model
IGP	Iterative Least Squares GNSS positioning
IMU	Inertial Measurement Unit
IP	Internet Protocol
ISR	Institute of Systems and Robotics
ISR-TRAFSIM	ISR - Traffic Simulator
ITS	Intelligent Transportation Systems
ITM	Intelligent Traffic Management
ITMD	Intelligent Traffic Management Deactivated
KF-CT	Kinematic sensors Fusion with Confidence Tests
KF(-CT)	Kinematic sensors Fusion without Confidence Tests
KF-CT+MAG	Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection
KF-CT+LASER(RTK)	Kinematic sensors Fusion with Confidence Tests + Laser landmarks detection with Real Time Kinematic gps accuracy

KF-CT+MAG+LASER	Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection + Laser landmarks detection
KF-CT+MAG+LASER(RTK)	Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection + Laser landmarks detection with Real Time Kinematic gps accuracy
LC	Lateral Controller
LEMITM	Legacy Early Method for Intelligent Traffic Management
LIDAR	Light Detection And Ranging
LB-AIGP	Lane Boundary Augmented Iterative least squares GNSS Positioning
LB-ASGP	Lane Boundary Augmented Set-membership GNSS Positioning
LB-CAIGP	Lane Boundary Collaborative Augmented Iterative least squares GNSS ositioning
LB-CASGP	Lane Boundary Collaborative Augmented Set-membership GNSS Positioning
LB-RAIGP	Robust Lane Boundary Augmented Iterative least squares GNSS Positioning
LB-RCAIGP	Robust Lane Boundary Collaborative Augmented Iterative least squares GNSS Positioning
LB-RASGP	Robust Lane Boundary Augmented Set-membership GNSS Positioning
LB-RCASGP	Robust Lane Boundary Collaborative Augmented Set-membership GNSS Positioning
LB-RSSMFDE	Lane Boundary Relaxed Set-membership Satellite NLOS Multipath Fault Detection and Exclusion
LTE	Long-Term Evolution
MAC	medium access control layer
MAS	Multi-Agent System
MAFS	Master Antenna Fixed Station
MI	Misleading Information
MSS	Magnets Sensing System
MTDT	Multi-Target Detection and Tracking
NET	network layer

NLOS	Non Line Of Sight
PDM	Park Driving Mode
PRN	Pseudo-Random Noise
PFC	Path Following Controller
PHY	physical layer
SGP	Set-membership GNSS positioning
RI	Roundabout Intersections
RMS	Root Mean Square
RTK-GPS	Real Time Kinematic-GPS
SFR	Sensor Fusion and Reasoning
SGP	Set-membership GNSS Positioning
SIVIA	Set Inversion Via Interval Analysis
RSIVIA	Robust Set Inversion Via Interval Analysis
STLS	Standard Traffic Light System
ST-SP	STart-StoP
TFP	Traffic Flow Profile
UMTS	Universal Mobile Telecommunications System
VP	Velocity Planner
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Vehicle + Vehicle to Infrastructure
VANET	Vehicular Ad hoc NETwork
WMITM	Waiting Method Intelligent Traffic Management
WGS	World Geodetic System
VPE	Vehicle Pose Estimator

Chapter 1

Introduction

Contents

In this chapter, we introduce the main problems to be addressed, and the major goals to be achieved. The key contributions of the dissertation are also outlined.

1.1 Motivation

In a society where time is valuable and that the automobile remains the primary means adopted for transportation, it is necessary to develop systems capable of smoothly flowing traffic, aiming to reduce the time spent by drivers on their travels thus minimizing the environmental cost of traffic caused by the growing number of circulating vehicles in urban areas. If possible the systems developed should contribute to increasing road safety thereby minimizing the social and economic impact of accidents. The frequency, volume and speed of transportation vehicles have been increasing at a high rate. Alongside this growth, significant improvements have also undergone in capacity, quality and effectiveness of infrastructure such as roads, crossroad and roundabouts intersections. However, these improvements are not sufficient to prevent traffic congestion and accidents. Drivers behavior can generate adverse situations such as the occurrence of traffic jams and accidents due to conflict of interests between the drivers involved. During the high traffic hours the infrastructures are used well above capacity if traditional traffic management techniques are used. In England, the costs due to traffic jams were estimated in the order of 34 000 million Euros [1]. A topic that is being one of the biggest concerns globally is related to environmental impact of vehicular transportation. The whole transportation system in Japan is responsible for 21% of total Japanese CO_2 emissions and 90% out of these emissions are generated by motor vehicles [2]. Roundabouts and signalized intersections are two of the most commonly used infrastructure for regulating intersections, thus being given attention by the new ITS technologies. Much work has been done in the field of autonomous Crossroad Intersections (CI). Dresner and Stone developed a simulator for autonomous CI management using agents [3]. Mateo and Sacsha used a market base strategy for the ITM system of multiple CI [4]. Roundabout Intersections (RI) are a very effective way of increasing traffic safety and specially at reducing fatalities [5], in Korea the reduction of fatalities by replacing 192 crossroads with roundabouts accounted for a 50% decrease in fatalities and 51% in major injuries (injuries requiring more than three weeks of hospitalization). Although the evident safety increase and the fact that there is a quite significant number of RI nowadays, not much work has been done to autonomously regulate traffic in RI. The early RI emerged as a very effective solution to regulate road traffic in road intersections. Because of its good performance, modern RI are widely used, especially in situations where there are a high number of roads joining at the intersection. Besides the obvious economic benefits, a more efficient RI traffic management leads to environmental benefits, by reducing the circulation time of vehicles, and improve drivers and passenger life quality. The inefficiency of RI in dealing with very high traffic, led to a group of solutions with main focus on regulating traffic in RI using traffic lights [6]-[7] and the optimization of the signaling control. Other approaches have also been tried by exploring new RI designs [8]. Different approaches and tools are needed to assist ITM algorithms on solving the aforementioned problems, namely, employing Vehicle to Vehicle + Vehicle to Infrastructure (V2X) communication technology and accurate positioning systems. The implementation and testing of new ITM solutions in real environments are very expensive, therefore it is necessary to evaluate the performance of new ITM solutions in simulated environments. There are numerous simulators used to study the performance of traffic in intersections, however, most of these have a closed architecture, preventing the application and development of new techniques for traffic management. Due to the absence of an integrated simulation system, for testing novel and efficient traffic control techniques, taking advantage of the potential of intersections as well as of autonomous vehicles, a new simulation framework is needed.

In a foreseeable future, autonomous vehicles will be introduced into the commercial market, and it is expected to be closely integrated with ITS. Meanwhile vehicle manufacturers and research units are integrating Advanced Driver Assistance Systems (ADAS). ADAS are systems designed to assist the driver along the driving process, including adaptive cruise control, lane departure warning, traffic sign recognition or automatic parking. One key requirement of ITS, to work seamlessly, is accurate vehicle localization (absolute and relative positioning). The ability to provide traffic information such as traffic patterns or collision warnings, relies mainly on the capacity of the network to identify the absolute and relative position of each vehicle. Vehicle localization is also a vital part in the autonomous navigation, it is mandatory to accurately compute the position of the vehicle for map matching, velocity profiling and path planning.

Accurate and continuous position calculation is essential for an autonomous vehicle to perform acceptably. Even for human driven vehicles, significant attention is being given in intelligent traffic management algorithms, whereby the positioning accuracy has crucial impact on the performance of the algorithm. The methods of vehicle localization based on motion integration, also referred as relative positioning methods, lead to localization errors which will accumulate gradually and unbounded. Absolute localization systems are based on multiple data source with precisely known positions. Both relative positioning methods and absolute positioning methods have advantages and shortcomings. Relative methods have high precision in short term and high frequency sampling but are subject to cumulative errors. Absolute methods based on GNSS are subject to multi-path errors, limited number of satellites in the field of view, especially in urban or forest areas, where, tall buildings or trees block or reflect GNSS signals, or can be totally unavailable, *e.g.* tunnels. GNSS positioning is also susceptible to lack of integrity of the computed solutions. The development of achievable and inexpensive solutions with improved position estimate accuracy, for on-road vehicles, guaranteeing the integrity of the computed solution, is one of the requirements for a ITS scenario. A single sensor cannot provide all information required for ITM algorithms, as these algorithms require not only accurate positioning but also a guarantee of the integrity of the computed solution. Therefore the fusion of multiple sensory information plays a key role in ITM algorithms for both autonomous and human driven vehicles.

1.2 Goals and key contributions

The research is aimed on the development and evaluation of intelligent traffic management algorithms in a realistic scenario. To accomplish this goal, a vehicular traffic simulator for intersections was developed where several support systems to new technologies of intelligent transport systems were integrated, enabling the measurement and thorough analysis of the intelligent traffic management algorithms in a simulated environment.

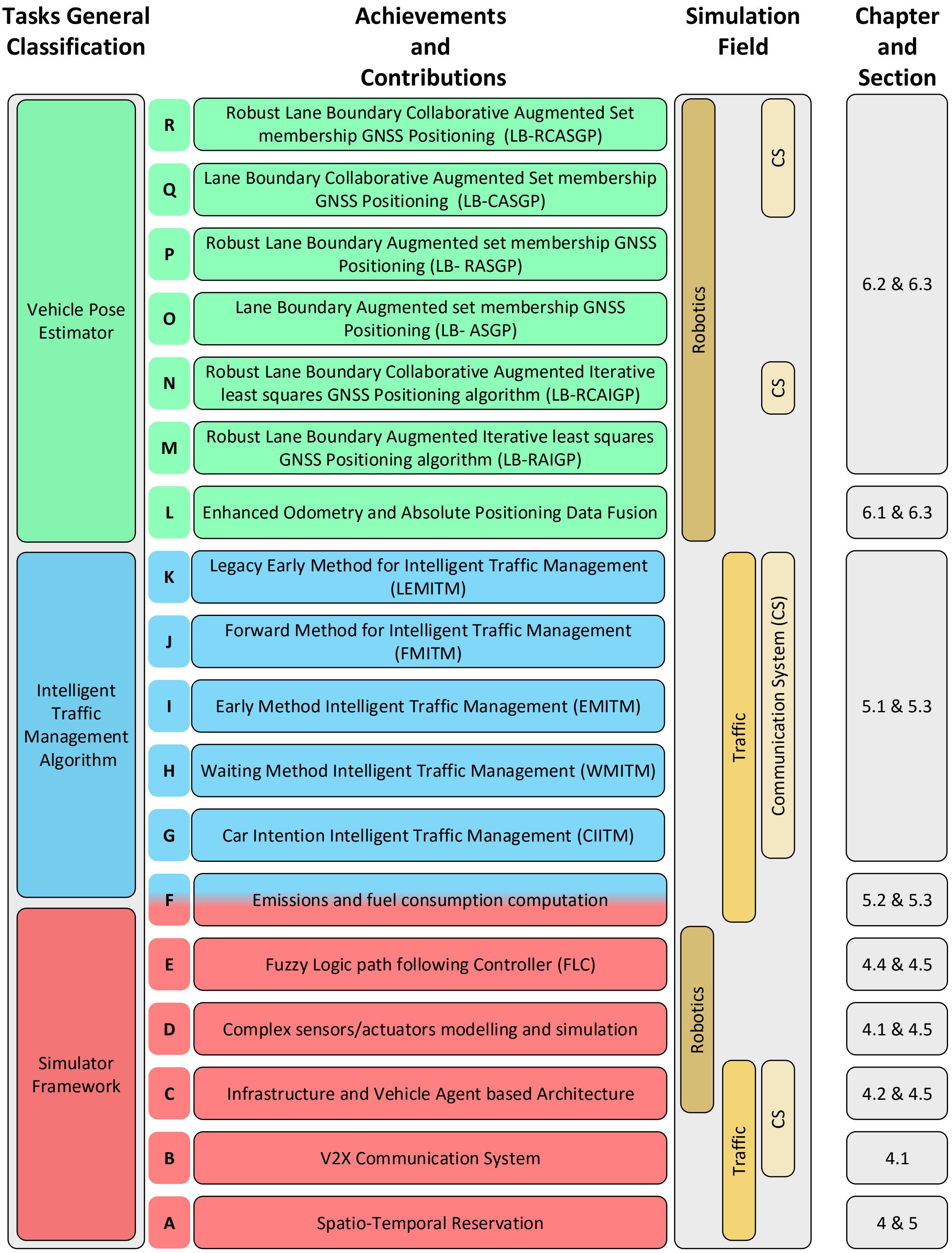
The integrated tools support complex positioning systems and enable the simulation of a simple Communication System (CS) for V2X. Another goal was to integrate on the simulator a fuel consumption and emissions model as well as a module supporting non-Ackermann steering vehicles, namely four-wheel steering system present in many modern vehicles. The simulator must also have a graphical interface so that a user can configure different scenarios in a simple way.

Another objective of this research is the development of intelligent and novel traffic management algorithms, using mechanisms of pre-reservation of spatio-temporal matrices supported by V2X. The ITM algorithms were tested and evaluated using the integrated vehicular traffic simulator.

Vehicle control and localization play a decisive role in autonomous vehicular traffic management, therefore its control for Ackermann and non-Ackermann steering vehicles, as well as, localization using multi-vehicle sensory systems was also an objective.

This research lead to the following achievements and contributions depicted in Figure 1.1:

- [**A-E**]- A microscopic traffic simulator framework for the simulation of traffic in intersections based on discrete events was developed. The simulator integrates the kinematics of recent 4WS vehicles and enables the simulation of V2X and a broad set of sensors models, including GNSS, Magnets Sensing System (MSS), LIDAR, odometers, steering actuators, etc. This simulator allows the testing of ITM algorithms, Path Following Controller (PFC), sensor fusion, etc. The simulator enables a non-skilled programmer to develop and integrate new modules and its source code is already disclosure to the academic community.
- [**H-K**]- Proposal of algorithms to enable fully automated vehicles equipped with V2X CS to cross intelligent roundabout and crossroad intersections:
  - a method to decelerate the vehicle and reserve the upcoming layer cells after the last layer occupied cells of a spatio-temporal matrix, Waiting Method Intelligent Traffic Management (WMITM);
  - a method to decelerate the vehicle and reserve the earliest layer free cells of a spatio-temporal matrix, Early Method Intelligent Traffic Management (EMITM);
  - a method to accelerate the vehicle and reserve the layer free cells of a spatio-temporal matrix prior than it would be expected if the vehicle followed a regular speed profile, Forward Method for Intelligent Traffic Management (FMITM);
- [**G,K**]- Proposal of algorithm to enable an intelligent intersection to accommodate vehicles, not equipped or with faulty V2V and V2I communications, Legacy Early Method for Intelligent Traffic Management (LEMITM).
- [**G-Q**]- In reservation-based algorithms for ITM at intersections it was used the estimated poses instead of the error-free poses. Each vehicle send its estimated pose (using positioning systems, namely a simulated GNSS) to the intersection traffic management system. This procedure reflects real-world information exchange more accurately and is thought to have have not been studied before.



Tasks General Classification		Achievements and Contributions	Simulation Field	Chapter and Section
Vehicle Pose Estimator	R	Robust Lane Boundary Collaborative Augmented Set membership GNSS Positioning (LB-RCASGP)	Robotics CS	6.2 & 6.3
	Q	Lane Boundary Collaborative Augmented Set membership GNSS Positioning (LB-CASGP)		
	P	Robust Lane Boundary Augmented set membership GNSS Positioning (LB- RASGP)		
	O	Lane Boundary Augmented set membership GNSS Positioning (LB- ASGP)		
	N	Robust Lane Boundary Collaborative Augmented Iterative least squares GNSS Positioning algorithm (LB-RCAIGP)	CS	
	M	Robust Lane Boundary Augmented Iterative least squares GNSS Positioning algorithm (LB-RAIGP)		
	L	Enhanced Odometry and Absolute Positioning Data Fusion		6.1 & 6.3
Intelligent Traffic Management Algorithm	K	Legacy Early Method for Intelligent Traffic Management (LEMITM)	Traffic Communication System (CS)	5.1 & 5.3
	J	Forward Method for Intelligent Traffic Management (FMITM)		
	I	Early Method Intelligent Traffic Management (EMITM)		
	H	Waiting Method Intelligent Traffic Management (WMITM)		
	G	Car Intention Intelligent Traffic Management (CIITM)		
	F	Emissions and fuel consumption computation		5.2 & 5.3
Simulator Framework	E	Fuzzy Logic path following Controller (FLC)	Robotics	4.4 & 4.5
	D	Complex sensors/actuators modelling and simulation		4.1 & 4.5
	C	Infrastructure and Vehicle Agent based Architecture	Traffic CS	4.2 & 4.5
	B	V2X Communication System		4.1
	A	Spatio-Temporal Reservation		4 & 5

Figure 1.1: Achievements/Contributions.

- **[F-K]**- Proposal of a detailed fuel consumption and emissions model, from low level to high level, to evaluate the environmental benefits of the reservation-based intelligent intersection control algorithms. The low level emissions model, is detailed to the level of piston displacement, encompasses accurate models of engines and vehicle dynamics.

- **[F-K]**- The use of the well-to-wheel analysis, to compute the total CO_2 emissions, is thought to be firstly introduced in the context of the reservation-based algorithm in this research. Most of the existing emissions models used in fuel consumption studies do not consider the fuel production, processing and delivery emissions stages.
- **[B,G-K]**- Integration of an hierarchical four-layered structure for the wireless CS. The use of the proposed CS, provides a more realistic communication environment, compared to the use of a fixed percentage of packet loss rate used in reservation-based algorithm previous studies. This extension accounts for several factors that were not considered in previous research efforts on reservation-based algorithms.
- **[B,L]**- Proposal of a cooperative localization system supported by V2V/V2I in semi-structured outdoor environments for position estimation, where a fusion of redundant encoders data and absolute positioning data provided by landmarks is achieved by means of an Extended Kalman Filter (EKF). Landmarks can either be a front laser detected vehicle or a detected magnetic marker. A front laser detected vehicle, is considered as a landmark, if at a given time, it either detected a magnetic marker or it has been able to compute a Real Time Kinematic-GPS (RTK-GPS) positioning solution.
- **[O-R]**- Use of absolute positioning data, acquired through a lane boundary system, in a set-membership algorithm, for vehicle position estimation, Lane Boundary Augmented Set-membership GNSS Positioning (LB-ASGP), and cooperative vehicle position estimation, LB-CASGP. The use of absolute positioning data, namely lane boundary detection system, for the position estimation, in a set-membership algorithm, rather than classical iterative least square, is thought to have not been studied elsewhere.
- **[M-R]**- Proposal of a guaranteed risk integrity zone computation algorithm for robust vehicle positioning, constrained by geo-referenced lane boundary measurements and GNSS pseudoranges to remove the constellation satellites with NLOS multipath pseudoranges affecting each vehicle individually, using Lane Boundary Relaxed Set-membership Satellite NLOS Multipath Fault Detection and Exclusion (LB-RSSMFDE). The use of geo-referenced lane boundary measurements and GNSS pseudoranges, to remove the constellation, is thought to be firstly introduced in the context of bounded-error models. The removal of NLOS multipath pseudoranges contaminated satellites, at the level of each individual vehicle, creates a new

class of positioning algorithms in the context of this research, this class is denominated *robust*: Robust Lane Boundary Augmented Iterative least squares GNSS Positioning (LB-RAIGP), Robust Lane Boundary Augmented Set-membership GNSS Positioning (LB-RASGP), Robust Lane Boundary Collaborative Augmented Iterative least squares GNSS Positioning (LB-RCAIGP) and Robust Lane Boundary Collaborative Augmented Set-membership GNSS Positioning (LB-RCASGP).

This work has produced the following publications.

Papers in international scientific Journals with referees:

2016 *(Submitted)* L.C.Bento, P.Bonnifait and U.Nunes, "Cooperative GNSS Positioning aided by Road-Features Measurements", Transportation Research Part C: Emerging Technologies, Elsevier.

2016 *(Submitted)* L.C.Bento, P.Bonnifait and U.Nunes, "Set-Membership Position Estimation with GNSS Pseudorange Error Mitigation using Lane-Boundary Measurements", IEEE Transactions on Intelligent Transportation Systems.

2014 *(Submitted)* L.C.Bento, R.Parafita, H.Rakha and U.Nunes, "A Study of the Environmental Impacts of Intelligent Automated Vehicle Control at Intersections via V2V and V2I Communications",Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, Taylor & Francis.

2007 U.Nunes, L.C.Bento, "Data Fusion and Path-Following Controllers Comparison for Autonomous Vehicles", Nonlinear Dynamics, International Journal of Nonlinear Dynamics and Chaos in Engineering Systems, Springer Netherlands.

Papers in conference proceedings with referees:

2015 L.C.Bento, R.Chelim, U.Nunes, "Collaborative Vehicle Self-Localization Using Multi-GNSS Receivers and V2V/V2I Communications" 18th IEEE ITSC - Intelligent Transportation Systems Conference, Canary Islands, Spain.

2013 L.C.Bento, R.Parafita, U.Nunes, "Intelligent Traffic Management at Intersections: Legacy Mode for Vehicles not Equipped with V2V and V2I Communications" 16th IEEE ITSC - Intelligent Transportation Systems Conference, Hague, Netherlands.

2012 L.C.Bento, R.Parafita, U.Nunes, "Inter-Vehicle Sensor Fusion for Accurate Vehicle Localization Supported by V2V and V2I Communications", 15th IEEE ITSC -

Intelligent Transportation Systems Conference, Anchorage, USA.

2012 L.C.Bento, R.Parafita, U.Nunes, ”Intelligent traffic management at intersections supported by V2V and V2I communications”, 15th IEEE ITSC - Intelligent Transportation Systems Conference, Anchorage, USA.

1.3 Thesis outline

The remainder of this thesis is organized as follows (see Figure 1.1):

Chapter 2 It is presented the ITS background for basic understanding of our work. It is not an exhaustive description of this scientific topic, but a rather focused presentation of main problems of ITM and contributions made to it in recent years. Moreover, a short state-of-the-art allows the reader to frame the contributions of this thesis.

Chapter 3 The theoretical background and methods used through out the thesis are presented in this chapter, namely it is discussed the components of the GNSS error, GNSS single epoch receiver positioning, lane boundary sensor modelling, set-inversion based on interval analysis and the measurement bounds setting. It gives the necessary background to understand the proposed methodologies in remaining chapters.

Chapter 4-7 These are the core chapters, where the Simulator Framework, the Intelligent Traffic Management Algorithms and the Inter-vehicle Sensor Fusion and Pose Estimator algorithms are formulated and presented. A subset of the simulation and experimental results used to test, compare and validate the algorithms are here presented. The performance evaluation of each solution is summarized on the final remarks of these chapters.

Chapter 8 In this chapter we review the main results, draws some conclusions and suggest some future directions.

Appendices A and B Briefly reviews some background concepts regarding fuzzy-logic background and chained form controllers respectively.

Chapter 2

ITS background and state of the art

Contents

In this chapter, it will be presented the ITS background for basic understanding of the research conducted. It is not an exhaustive description of this scientific topic, but a rather focused presentation of main problems of ITM and contributions made to it in recent years. Moreover, this short state-of-the-art allows the reader to frame the contributions of this thesis.

2.1 ITS Simulation

Although real world experiments provides realistic results, the physical implementation is a time consuming procedure and expensive as it may require a lot of hardware and human resources. Alternatively, simulation is affordable and provides good results and in a cost effective way enable to evaluate the performance of proposed protocols/schemes. Hence simulation tools have been preferred over outdoor experiments because their simple, easy and cheap. Many tools exist for this purpose but most of them have the problem with the proper interaction among them. In this section we evaluate the available simulation models, distributed among three categories: vehicular road traffic simulators; vehicle emission models and multi-purpose integrated simulators.

Vehicular road traffic simulators include models described by the framework, which includes topological maps like lanes, roads, streets, obstacles in mobility and communication model, vehicle velocities, the attraction and repulsion points, based on traffic densities relating to how the simulation time could vary, vehicular distribution on roads and intelligent driving pattern [9]. The vehicle emission models are concerned with computing fuel consumption and emission at a macroscopic level by tracking average link speeds or on a microscopic level by tracking individual vehicles instantaneously. The multi-purpose integrated simulators include V2X communications and their integration with simulators with the capacity to simulate sensors and its noise characteristics as well as the actuators and its performance.

2.1.1 Vehicular Road Traffic Simulators

Intelligent vehicular traffic management solutions may target vast deployments that comprise tens, hundreds or even thousands of vehicles. Large-scale experimental test beds are too expensive and complex to scale to more than a few vehicles, therefore simulation is the means of choice for their performance evaluation [10].

Traffic simulators are divided into three broad categories according to the level of representation of vehicles [10, 11, 12]:

Macroscopic simulators capture traffic dynamics of large networks, *i.e.* model traffic as a continuous flow. In this category of simulators, vehicles are not considered individually, its movements are inspired by fluid theoretical models, leading to computationally faster simulators. The models describe the collective vehicle dynamics in terms of the spatial vehicle density, the average flow, and average speed. They are mainly used in situations where it is necessary to simulate a large number of vehicles over a long period of time;

Mesoscopic simulators model individual vehicles at an aggregate level, *i.e* model the individual vehicles but not their interactions. In this type of simulators, traffic flow is described in less detail than in microscopic simulators and in greater detail than in macroscopic. The behavior of vehicles and drivers are not described individually, rather on small groups of vehicles of a specific user-class classified by their position, velocity, and desired velocity at an instant of time;

Microscopic simulation environment provide a detailed representation of the traffic, and thus capture the behavior of vehicles and drivers in great detail, *i.e.* each vehicle is considered individually. This class of simulators considers lane changes, drivers and vehicles types, as well as the interactions between them and the environment, meaning that the physics of individual vehicles as they interact with the driver models and the infrastructure are characterized in this model. Modeling individual vehicles at a microscopic level requires a significant computational capability, making this type of simulators more suitable for local traffic study.

In Table 2.1, we provide an overview of the most relevant microscopic traffic simulators [10, 11, 13, 14, 15, 12].

Table 2.1: Studies on relevant microscopic traffic simulators

Simulator	*OS/License type*	*Interface with other tools*	*Vehicle representation*	*Traffic representation*	*User interface/OS*	*Special remarks*
AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks)	Windows / Unix, proprietary, time trial or limited functionality demo version available	an API feature, enable users to modify the default simulation routine and test their own models	the user may define as many vehicle types as desired and provide the vehicle parameters required. Classes of vehicles can be defined to group vehicle types	it can be configured as input flows (flows on the incoming links at the boundary of the network) and turning percentages (the percentage of vehicles turning left, right or going straight for each intersection) or it can be provided in the form of origin-destination matrices	graphical interface enable user to edit input networks, manage experiments and view results	a significant advantage of AIMSUN is that the gap-acceptance behavior of drivers is modified based on their delay time. Most other models do not represent such phenomena. It has reasonable level of fidelity
CORSIM (CORridor microscopic SIMulation)	Windows, proprietary, time trial or limited functionality demo version available	an API feature, enable users to modify the default simulation routine and test their own models	FREESIM allows nine different vehicle types while for NETSIM it can reach up to 16. Vehicle type characteristics include vehicle length, maximum acceleration and deceleration	the specification of the trip table, to perform traffic assignment is accomplished via origin-destination matrices. Sources and/or destinations (sinks) of traffic that are internal to the network can also be specified	input and output is via ASCII text files, however there are tools to graphically create these input files and display results	the lack of appropriate modeling of vehicle movements in the intersection box is an disadvantage of CORSIM. Although the current model includes *micronode* modeling of intersection box movements, the calculation of conflicts and output statistics is based on approximation principles, which are not considered a viable model for intersection vehicle movements
PARAMICS (PARAllel MICroscopic Simulation)	Windows, proprietary, time trial or limited functionality demo version available	an API feature, enable users to modify the default simulation routine and test their own models	there are seven predefined vehicle classes but the user may add more as required	origin-destination matrices to derive traffic volumes	user friendly graphical interface for network building and visualization of results	it can model roundabouts by default and enables users to model unconventional circles. The aspects that make it attractive are: the capability of microscopically modeling the vehicle-following/lane-changing behavior of individual vehicles and the employment of an algorithm that defines a general purpose method to steer a vehicle over a realistic path between its current position to any target position, taking angles of orientation and steering limits into account. It has reasonable level of fidelity
SUMO (Simulator of Urban MObility)	Windows / Unix, free, open source	It allows the access to a running road traffic simulation to retrieve values of simulated objects and to manipulate their behavior *on-line* via TraCI (Traffic Control Interface). A TCP-based client/server architecture is established where SUMO acts as a server and the *controller* is the client	default values for acceleration, maximum speed and desired speed distributions are given but can be changed by the user to reflect local traffic conditions. Various car types, truck types, trams, buses and pedestrians can be defined	while data needed to describe the departure times and a route origin and destination are given, the routes themselves must be computed, for that purpose SUMO suite includes an application for converting origin-destination matrices to single vehicle trips	it reads the input information, processes the simulation, gathers results and produces outputs via ASCII text files, however it also has an optional graphical interface called SUMO-GUI	SUMO uses two significantly influencing parameters for queue and travel time, *i.e.* it uses parameters of the driver imperfection and also the driver reaction time
VISSIM (Verkehr In Städten - SIMulationsmodell) german for Traffic in cities - simulation model	Windows, proprietary, time trial or limited functionality demo version available	an API feature, enable users to modify the default simulation routine and test their own models	default values for acceleration, maximum speed and desired speed distributions are given but can be changed by the user to reflect local traffic conditions. Various vehicle types, truck types, trams, buses and pedestrians can also be defined	origin-destination matrices, a route choice model is also included	graphical interface enable to edit data such as network definition of roads and tracks, vehicle and behavioral driver specifications, car volumes and paths, transit routes and schedule	the priority rules feature of VISSIM appears to allow complex modeling of junction behavior, including friendly merging, (*i.e.* situations where following vehicles will slow for merging vehicles to create a gap), as it occurs in the real world. Another advantage of VISSIM is the representation of on-street parking behavior and double parking. It has good level of fidelity

Microscopic traffic simulators produce individual synthetic vehicular traces, and therefore have emerged as the most viable solution to faithfully represent the unique dynamics of vehicle movement patterns. The fidelity of the synthetic traces generation process has drastically improved over the last few years, as a result, it is today widely acknowledged that, intelligent vehicular traffic management system at intersections may only be accurately reproduced on microscopic-level simulation.

The majority of the most representative simulators are proprietary and therefore the internal systems modeled are unavailable to the developer. Nevertheless all presented simulators enable the development of external modules to modify the default simulation routine and test custom models. The external module development for the proprietary simulators is a difficult task and in some cases the documentation is scarce, while for the open source simulator SUMO the development of new features requires good programming skills and deep knowledge of the inner working of the simulator. The main advantage of microscopic traffic flow models is that the behavior of the drivers and vehicles are described in detail. Therefore, they can provide relatively more information regarding the characteristics of the traffic flow than other types of models. The main limitation of microscopic models is that they require a large memory size and they are very slow when used for large traffic networks making these models are mostly useful for local traffic studies. Moreover, microscopic models require large number of parameters, which are most often difficult to calibrate [12].

The autonomous vehicle paradigm has been around in researchers' minds for some years now, as it envisions to radically change the concept of mobility and traffic management. With the technology rapid growth, researchers are now more focused on the software challenges that such a complex system requires, as hardware itself is becoming more affordable [16]. Major automobile manufacturers predict the availability of autonomous vehicles in the near future. In [17] Juliussen anticipates that nearly all of the vehicles in use are likely to be self-driving cars or self-driving commercial vehicles sometime after 2050.

It is common among microscopic simulators to enable vehicle and driver customization but there is no support for the performance evaluation of path-following controllers and non-Ackermann steered vehicles, such as the four wheel steering vehicles (4WS), are not possible to be modeled. 4WS vehicles account for more than forty commercially available models from major automobile industry manufacturers. Another major drawback of microscopic simulators in the context of autonomous vehicles, is the lack high quality sensor simulation and physics engine. Therefore traffic simulation of autonomous vehicles requires further module extensions, this class of traffic simulators is referred has nanoscopic [18]. An integrated framework that aims coupling robotics and a microscopic traffic simulator is presented in [12]. This work developed an integrated framework, using SUMO and USARSim (Unified System for Automation and Robot Simulation), enabling autonomous vehicles to be deployed in a rather realistic traffic flow at the same time it simulates all its sensors and actuators. Although the software developed was multi-platform, no software was released to the academic community neither the tools used (Unreal scripting language and C++) enable a non-skilled programmer to develop and integrate new modules.

2.1.2 Vehicle Emission Models

Global warming is a very serious problem, which is worsening with the growth of traffic congestion, due to the increasing number of vehicles on roads. As traffic congestion increases, more fuel is consumed and consequently CO_2 emissions are increased. Given the current economic crisis, the rising price of fuel and the social pressure for ecological awareness, the reduction of fuel consumption and CO_2 emissions is paramount. According to Nagurney [19], 15% of the world CO_2 emissions come from road vehicles, this figures can be further compromising for developed countries. The average fuel consumption of vehicles is provided by manufacturers, this information is mainly derived from measurements based on engine test benches or from a standard test cycle using a chassis dynamometer, rather than on real-life driving cycles, therefore the information provided does not make it possible to derive CO_2 emissions [20].

A number of research efforts have attempted to develop vehicle fuel consumption and emission models. Due to their simplicity, macroscopic fuel consumption and emission models have been proposed [21]. These models utilize the aggregated characteristics, such as the average speed and total length of a trip to estimate the emission rate. A major drawback of these models is the fact that they compute fuel consumption and emissions based on average link speeds. Using the single average speed is not sufficient to characterize the precise driving behavior of a vehicle, and therefore, not accurate to estimate the emission level of a vehicle, *i.e.* various vehicle operations, such as variable instantaneous vehicle speed, vehicle acceleration and gear change pattern may lead to the same average speed with different fuel consumption. Average speed cannot capture detailed spatial information, and thus do not allow accurate emission estimations of vehicles at small segments of a road network making the use this type of model unsuitable for intersections. These models are suitable to be used in scenarios where emission prediction at a coarse granularity is expected, namely in large-scale transportation analyses where the average speed is adequate to characterize traffic conditions [22].

To overcome the limitations of macroscopic fuel consumption and emission models in small segments of a road network, research methods using instantaneous speed and acceleration variables were proposed, these methods are referred as microscopic fuel consumption and emission models (also referred as instantaneous fuel consumption and emission models). Microscopic models can be classified into three general categories: emission map based models, data driven statistical based models and physical based model.

Emission map based models use so-called *engine maps* to obtain fuel consumption. The fuel consumption is obtained through a search on a table mapping the performance of the engine, where each cell of the table stores instantaneous fuel consumption given the velocity and acceleration. Although emission maps are simple and easy to use, they can be sparse and too sensitive to the driving cycles;

Data driven statistical based models are also referred as regression-based models. Mathematical functions of instantaneous vehicle speed and acceleration are used to estimate fuel consumption. These mathematical functions are built based on information from vehicle speed/acceleration and emission datasets. These models lack clear physical interpretations;

Physical based model also referred as load-based model, estimate fuel consumption using detailed physical parameters of vehicles. The vehicle physical dynamics is simulated as well as the corresponding dynamic power flow and energy losses within the powertrain. Physical models can be classified into forward-facing and backward-facing based on the way in which the dynamic calculations are performed. A forward-facing model features a driver model that provides torque demand to meet the desired speed, while the backward-facing model the vehicle speed is known and the required power throughout the system is computed. Physical based models are usually complicated and require high computational effort as they consider a wide range of parameters related to the fuel consumption, such as vehicle speed and acceleration, road grade, gear ratio, engine max power, engine power demand, *etc* [22].

Physical based models are the most comprehensive microscopic models for fuel consumption estimation. Table 2.1 describes briefly several physical based microscopic fuel consumption and emissions models.

A global view of fuel consumption models reveals that the majority are proprietary and therefore the internal systems modeled are unavailable to the developer, focusing on the short list of fuel consumption models presented, only one is free and open source. The external module development is not available for the proprietary models, while for the open source simulator QSS-TB the development of new features is not natively available, but since it is implemented in Matlab and open source, it makes it possible for non-skilled programmers to develop and integrate new modules. Most of the common consumption

Table 2.2: Studies on physical based microscopic fuel consumption and emissions models

Model	*OS/Licen type*	*Interface with other tools*	*User interface*	*Special remarks*
EMIT (EMIssions from Traffic)	Windows, proprietary	not available	integrated visualization and manipulation utility called the EMIT Mapper	EMIT was developed as a hybrid model to overcome the disadvantages of both data driven statistical based models and physical based models. EMIT combines physical factors from the physical based model in order to increase the accuracy of fuel consumption estimation, but it only requires simple parameters, leading to the estimation of the fuel consumption to be able to run fast. EMIT only requires vehicle speed and acceleration with a sampling time of one second for the computation of the fuel consumption. EMIT provides reasonable results compared to actual measurements, over a wide range of operating conditions. A major limitation of EMIT is that the model does not take into account road grade that may significantly influence the fuel consumption estimation neither is able to interface with other tools [22]
CMEM (Comprehensive Modal Emission Model)	Windows / Unix, proprietary	Paramics plugin provides an interface between CMEM and Paramics	Java Based GUI	The CMEM model is based on a parameterized physical approach that breaks down the entire emission process into components that correspond to the physical events associated with vehicle operation and fuel emissions production. To produce accurate estimations, it requires detailed vehicle specific parameters for the estimations such as the engine friction coefficient, and the vehicle engine speed. This type of model is deterministic, since it is based on causal parameters rather than statistical variables which are not necessarily linked to physical phenomena. Results obtained are very good, with no significant bias. CMEM can be seen as a state-of-the-art microscopic emission model because of its ease of application [20, 22, 23]
QSS-TB (QuasiStatic Simulation Toolbox)	Windows / Unix, free, open source	not available	Matlab / Simulink environment	The QSS-TB is implemented using a *reversed causality quasistatic* approach. Where *quasistatic* means that the dynamic evolution is broken into a sequence of stationary states at discrete-time instants and *reversed causality* means that, using the driving cycle speeds, the accelerations are computed and the necessary forces are determined, based on the vehicle features and mechanical transmission. The QSS-TB makes it possible for powertrain systems to be designed quickly in a flexible manner and to calculate easily the fuel consumption of such systems. The QSS TB library integrates various elements, such as driving cycles, vehicle dynamics, internal combustion engine, electrical motor and mechanical transmission. Batteries, supercapacitor and fuel cell are also included enabling the modeling of the hybrid electric vehicles. There are two major drawbacks. The first one is related to the input variables, as it does not deal with real measurable quantities in a vehicle, making it not suited to control systems development. The second comes from the assumption that *speed vs. time* trace is *followed*, making it not suited to predict best-effort performance under component limited conditions. The major advantages of QSS-TB are the low computational requirements and the fact that it is capable of great accuracy as well as its validation being documented in the literature [24, 25, 26]

models are only suited for ICE (internal combustion engine), from the short list of fuel consumption models analyzed only the QSS-TB is able to model hybrid or full electric vehicles.

An in-vehicle energy management control strategy was developed by Kang [27], the method uses microscopic data from networked vehicles, for the optimization of vehicle fuel and energy consumption to minimize total trip costs. The Paramics simulator was used to generate traffic traces and the QSS-TB was used for the batteries and fuel consumption models. In this study the simulators were decoupled, no communications were simulated and therefore it was not possible to run a real-time simulation and to evaluate the performance under a realistic scenario.

The studied models focus on vehicle's *tank-to-wheel* power flow, *i.e.* it depends mainly on the powertrain and on the on-board energy carrier, neglecting the *well-to-tank* power conversion. Traditional ICE vehicles rely on combustion of fossil propellant. The fossil must be extracted from natural reserve, refined and transported to the end user. All these steps introduce additional losses that depend on the fuel used and on the fossil extracting process [28], and they should be addressed as part of the emissions model.

2.1.3 Multi-purpose Integrated Simulators

Most simulators lack important features for the simulation of autonomous vehicles in traffic modeling context, mainly because *Vehicular Road Traffic Simulators* are not detailed enough to simulate autonomous vehicles, these simulators are over-simplified, while *Robotic Simulators* cannot simulate large scale traffic and V2X communications [29].

Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) will see volume deployment around 2020 and has enough value to be used as part of autonomous vehicles. V2X provide additional and useful information and this will improve the capability to avoid accidents at low cost, *i.e.* approximately $200 in volume production [10].

The proposed solutions for vehicular communications include the communication through a telecommunication infrastructure or directly between vehicles. The solutions are then divided into conventional communications that can use infrastructures General Packet Radio Service (GPRS), Universal Mobile Telecommunications System (UMTS), Long-Term Evolution (LTE), etc.) and in infrastructureless communications and Vehicular Ad hoc NETwork (VANET) [10]. Mehlfuhrer *et al.* developed a MATLAB-based link and system level simulation environments for UMTS and LTE [30]. Ray *et al.* developed a MATLAB-based link and system level simulation environments for IEEE 208.11 abg [31]. The source code of both simulators is available under an academic non-commercial use license allowing researchers full access to standard-compliant simulation environments. Due to the open source availability, the simulators enable reproducible research in wireless communications and comparison of novel algorithms.

The simulation of VANET typically employs two concepts, the vehicles mobility simulation and wireless network simulation, seldom this two concepts are merged together, *vide* Figure 2.1. VANET simulation requires that a traffic and network simulator can be jointly used with feedback between them to render the simulation results as accurate as real life. Hassan *et al.* [9, 32] discussed the shortcomings of current VANET simulators, namely:

- Separate Traffic and Network simulator: VanetMobiSim and NS-2; Problem: Traces

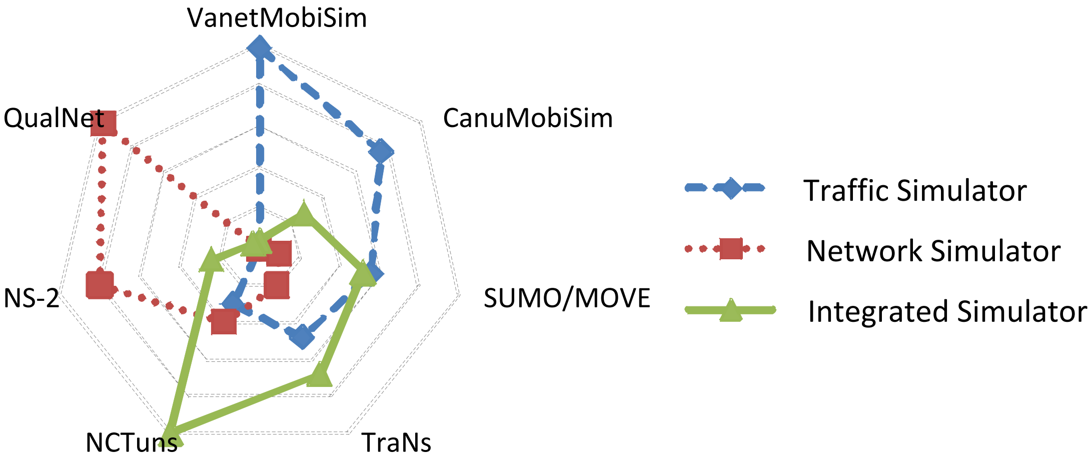


Figure 2.1: Application strength relations between simulator types.

are generated once and thus no feedback is allowed;

- Integrating Traffic and Network simulator: TraNs; Problem: Loose coupling, the feedback process is slow;
- Federating Traffic and Network simulator: MOVE and NS-2 / QualNet; Problem: Still lack interaction.

Recommendations on how to perform useful VANET simulations were also presented in [9, 32]: to integrate the two simulators in time and space, a bidirectional communication should take place, with the autonomous vehicles providing kinematic variables to the traffic simulator; the traffic simulator then calculates its surroundings and return its data back to the autonomous vehicle simulator; all of this transactions should occur in the same time step.

Figure 2.1 presents the overall performance of several simulators with respect to its suitability as a traffic simulator, as network simulator or as an integrated VANET simulator.

The NCTUns simulator revealed to be most suitable. NCTUns is a high-fidelity and extensible network simulator and emulator capable of simulating various protocols used in both wired and wireless Internet Protocol (IP) networks. NCTUns has been deprecated because it is was superseded by EstiNet, which is proprietary.

Pereira [33] developed an integrated architecture for autonomous vehicles simulation. The main objective was the integration of two types of simulators, namely a robotics USARSim and a traffic simulator SUMO [33]. However part of the proposed solutions for the architecture was proven to be unfeasible giving the semi-closed nature of the Unreal Engine working as a basis for USARSim.

Even after so much effort of the academic community in the development of simulation environments, we are still far away from having an multi-purpose simulator for traffic simulation of autonomous vehicles. The more features simulators have the more realistic they are, and consequently more resources they need to perform simulations. A trade-off on the realism of the simulation as to be chosen, *i.e.* an oversimplified simulation is unuseful as well as a simulation time beyond a reasonable period of time [29].

2.2 Environment Impact Reduction through Traffic Management

The number of automobiles has been increased on the road in the past few years, this increase is essential for the economic development, but it is also harmful to the environment and to human health. Recent developments in green transportation have heightened the importance of research techniques in this area, the research on the reduction of CO_2 emissions from road transportation has grown notably [20]. In this context traffic management assumes a pivotal importance. Traffic management is the planning, monitoring and control or influencing of traffic [34]. Traffic management is a broad definition that assumes different meanings function of the context it is applied. From a *macro* perspective, traffic management represents an integrated management of both roadways infrastructures and vehicles at a higher hierarchical level, *e.g.* inter-city and intra-city road networks aggregated traffic management. From a *micro* or *nano* perspective, traffic management may represent traffic control at a single intersection. The studies addressing strategies for fuel efficiency and CO_2 emissions reduction can be divided into four main categories:

1. traffic reduction and transportation hardware improvements;
2. traffic routing (Table 2.3);
3. vehicle speed control and traffic signal management (Table 2.4);
4. reservation-based ITM (Table 2.5).

Table 2.3 to 2.5 list some prominent recent studies including the tools that were used, namely the *emission models* and the *scenario type*.

The *Emission Models* represent the instantaneous vehicle emission models applied in each study. These models can be divided into:

- *Emission map based models (e.g. HBEFA)*
- *Data driven statistical based models (e.g. VT-Micro and EM-2)*
- *Physical based model (e.g. CMEM, EMIT, SIDRA TRIP, and VT-CPFM)*
- *Real experiments (e.g. EM-1 and EM-3)* - fuel consumption is directly computed/estimated based on field measurements. A CAN bus data logger was used in EM-1 to record the fuel consumption [35], while in EM-3 the mass air flow and vehicle speed were recorded using an on-board diagnostic 2 (OBD2) port [36].

The *scenario type* represents the environment used in each study to model traffic dynamics. These scenario types can be divided into:

- *Microscopic simulation (e.g. SUMO, VEINS, PARAMICS, INTEGRATION, VGSim, DIVERT, ST-5, ST-6, ST-9, ST-10, TransModeler, AIMSUN and VISSIM)*
- *Mesoscopic simulation (e.g. DYNEMO, DYNASMART, DYNAMIT DYNUS-T, and MATSim)*
- *Macroscopic simulation (e.g. EMME/3 and ST-4)*
- *Node/link tour simulation (e.g. ST-1, ST-2 and ST-3)* - model a scenario with a collection of possible routes from a departure place to a number of cross-through geographically scattered places subject to side constraints.
- *Real experiments (e.g. ST-7 and ST-8)* - where data acquired for the analyzed scenario is taken from real experiments.

Additionally the *scenario type* may include V2X communication simulators, namely OMNET, NS-2/3, NetSim and NCTUNS, but they are often missed in most of the studies.

Vehicle emissions reduction through traffic reduction and transportation hardware improvements

Suthaputchakun *et al.* [37] present a survey of the fuel efficiency and CO_2 emissions reduction based on vehicular communications as well as the envisaged technical challenges in this research area. Traffic reduction, can be achieved through improved city planning, with major places within walking distance; public transportation improvement and through the promotion of car pooling and car sharing. These strategies can indirectly reduce fuel consumption and CO_2 emissions [37]. Transportation hardware improvements can also achieve fuel consumption and CO_2 emission reductions through improvements in infrastructure design (e.g. road paving materials and slope elimination) and vehicles (e.g. engine electronics, periodical vehicle checks).

Studies on vehicle emission reduction by traffic re-routing

Table 2.3 presents a summary of the emission reductions reported in the literature as it relates to traffic re-routing.

The majority of the studies focus on optimizing an objective function composed by the travel time and/or fuel consumption and emission levels. A computational market distributed allocation of an urban road network is introduced in [42]. In this *out of the box* algorithm, driver agents trade the use of the capacity inside the intersections with intersection manager agents.

Table 2.3: Studies on vehicle emission reduction by traffic re-routing

Algorithm	*Emission Model*	*Scenario Type*	*Results/Contributions*
pollution-routing problem; integer linear programming	based on CMEM	ST-1	objective function that accounts for the labour costs, greenhouse emissions, fuel consumption, travel times and their costs [38]
disjoint candidate path; Dijkstra; A^*	N/A	ST-2	reliable paths; real-time search [39]
re-routing traffic to parallel lanes	EMIT	VEINS (OMNET + SUMO)	optimization of two conflicting metrics: fuel consumption and emissions versus the travel time [40]
Energy/Emissions Operational Parameter Set using multivariate regression analysis, for path selection	CMEM	ST-3	use of the three following independent variables in the route selection: vehicle characteristics, e.g., vehicle type, model year, and loaded weight; roadway characteristics, e.g., roadway type, vertical grade, and type of intersection at link ends (stop-sign, signalized, or none); traffic characteristics, e.g., speed, density, or congestion level; other explanatory variables, e.g., driver characteristics and the environment [41]
competitive computational market for the distributed allocation of an urban road network	N/A	based on DYNEMO	market dynamics affect the driver agent decision making, contributing to benefits by means of lower average travel times and less congestion [42]
Estimation of fuel consumption and emissions using GNSS data	VT-Micro	Field data collection using GNSS data	the faster highway route choice is not always the best route from an environmental and energy consumption perspective; macroscopic emission estimation tools can produce erroneous conclusions by ignoring transient vehicle behavior along a route; air quality can be improved significantly by minimizing high-emitting driving behavior [43]
Agent-based eco-routing algorithm	VT-Micro	INTEGRATION Software on simple and large-scale networks	Developed a modeling framework to model eco-routing strategies and tested them on two large-scale networks. Fuel savings ranging between 3.3 and 9.3 percent were observed compared to typical travel time minimization routing strategies. The study showed that the configuration of the network had a significant impact on the results [44, 45]

Studies on vehicle emissions reduction through the vehicle speed control and traffic signal management

The algorithms controlling vehicle speed and traffic signal timings to achieve emissions reductions present in the current literature are summarized in Table 2.4.

Vehicular fuel consumption and CO_2 emissions are proportional to the frequency of accelerations and decelerations. The majority of the presented studies attempt to reduce vehicle accelerations, decelerations and avoid stops, by controlling the vehicle speed. Some studies explicitly optimize the vehicle fuel consumption and CO_2 emission levels (e.g. [56]). The proposed methods for the control of the vehicle speed can be divided into two categories: direct and indirect methods. The direct speed control method can alter the vehicle speed or provide an eco-friendly speed advice to the driver [35]. The indirect speed control can make use of infrastructure variable speed limits or variable traffic light system (TLS) timings in order to comply with a low level of jerk along the path and reduce engine idling [36, 46]. A more complex system is presented in [53], the proposed algorithm aggregates the control of several traffic signal systems (TLSs) timings, achieving significant

Table 2.4: Studies on vehicle emissions reduction through the vehicle speed control and traffic signal management

Algorithm	*Emission Model*	*Scenario Type*	*Results/Contributions*
speed limit	N/A	ST-4	acknowledge that a speed limit law may play the same role as a toll charge scheme and perform better than some negative (rebate) toll schemes under certain conditions for network flow management [46]
vehicle-following controllers comparison	CMEM	ST-5	acknowledge that the performance of a vehicle-following controller is mainly determined by the spacing policy employed rather than by its form [47]
Intelligent Speed Adaptation (ISA)	CMEM	PARAMICS	acknowledge that ISA has the potential to mitigate congestion by smoothing traffic flow during congested conditions, which also leads to lower fuel consumption and pollutant emissions [48]
Variable-speed limit sign (VSLS)	N/A	PARAMICS	assessment of expected benefits of a practical VSLS control strategy in terms of VSLS response activity and upon modifications to the control algorithm; modeling of relative safety benefit modeling of measures of crash potential parameters [49]
vehicle speed adaptation function of TLS	VT-Micro	INTEGRATION	TLS communication range; ROI/emissions assessment on speed adaptation [50]
imperialist competitive algorithm (ICA) on TLS	N/A	ST-6	control the number of cars in the under-construction roads [51]
bus loading and traffic conditions modeled in time groups; driver peer competition	EM-1	ST-7	incentive system has been developed for the case company Tampere City Transport [35]
carbon-footprint/fuel consumption-aware variable (FC-VSL) (variable speed limit control with optimal fuel consumption)	SIDRA Trip	VGSim	optimal vehicular fuel consumption under live traffic conditions [52]
Virtual Traffic Light (VTL)	EMIT	DIVERT; NS-3	CO_2 reduction impact using VTL [53]
passenger-based adaptive traffic signal control (PATSC) mechanism; Green Signal Time Extension and Termination (GSTET)	SIDRA Trip	NCTUNS	assessment of the benefit value *w.r.t.* the number of passengers [54]
nonlinearly constrained optimization problem for a speed profile minus maximum 5 km/h	EM-2	N/A	real-time feedback system, including visual instructions, to enable drivers to alter their driving styles in response to actual driving conditions to improve fuel efficiency [55]
deceleration patterns when approaching traffic signals that force or may force the vehicle to stop	EM-3	ST-8	not requiring the installation of infrastructure on the road [36]
Dynamic programming using an A* minimum path algorithm	VT-CPFM	Agent-based simulation	optimized vehicle trajectories at signalized intersection approaches using traffic signal phasing and timing (SPaT) information [56]
Heuristic non-linear optimization	VT-CPFM	iCACC tool	optimizing vehicle trajectories while proceeding through an intersection [57, 58]

CO_2 reductions.

Studies on vehicle emissions reduction through the use of reservation-based ITM

A summary of studies on vehicle emission reduction through the use of reservation-based ITM is presented in Table 2.5.

Table 2.5: Studies on vehicle emissions reduction through the use of reservation-based ITM

Algorithm	*Emission Model*	*Scenario Type*	*Results/Contributions*
reservation-based approach based on a detailed communication protocol	N/A	ST-9	reservation-based approach significantly outperforms current intersection control technology-traffic lights and stop signs [59]
Priority-based Policy, With-Lane-based Policy, First come, first serve	HBEFA	ST-10	reduced significantly the fuel consumption and vehicle emissions compared to traditional signal control systems [60]
dynamic hierarchical reservation protocol	CMEM	VISSIM	assigning different priorities to incoming reservation requests; evaluation of the benefits from a mobility and an environmental point of view [61]
Heuristic non-linear optimization	VT-CPFM	iCACC tool	optimization of vehicle trajectories to prevent vehicle collisions and minimize the total intersection delay [57]

The reservation-based ITM was first proposed by Dresner *et al.* [3]. The vehicles' pose were considered to be known, no emission model was considered and the V2I wireless communications were simulated through a basic *protocol-less* communication system. Jin *et al.* [60] developed a single lane crossroad intersection time-space occupancy reservation scheme. A Priority-based Policy, With-Lane-based Policy, First-come-first-serve (FCFS) algorithm was applied. The vehicles' pose were considered to be known and the V2I wireless communications were simulated through a basic *protocol-less* communication system. The proposed strategy reduced the vehicle fuel consumption and emission levels significantly compared to traditional signal control systems. Zohdy and Rakha developed a centralized intersection controller that optimizes the vehicle trajectories to minimize the total intersection delay, the algorithm was extended to operate on roundabouts where any failure in communication would have a traditional roundabout intersection control [58], the assumption of an existent communication system was made and it was also assumed that the vehicles' pose was known *a priori*. A speed profile for each vehicle crossing an intersection was proposed by Huang *et al.* [61] and applied in their integrated microscopic traffic simulator. An integrated framework was used to develop a dynamic hierarchical reservation-based ITM and to conduct studies of the environmental impacts of its application. The ITM algorithm assigns different priorities to incoming reservation requests. Results revealed significant mobility benefits, in terms of increased capacity to handle traffic and reductions in fuel consumption, emission levels and travel time. The communication layer used a basic *protocol-less* communications system to simulate V2I wireless communications and the vehicles' pose were considered to be known.

2.3 Vehicle Localization

This study presented in this section should not be viewed as a thorough study of positioning techniques that have been recently proposed in the research literature, nevertheless it has a value of a survey for multi-sensor GNSS aided positioning, as it makes a description and comparison of several algorithms.

Augmenting GNSS localization with other sensory information to improve the positioning accuracy, is common in the context of intelligent vehicles [62]. In [63], N.M. Drawil developed a V2V communication assisted localization. This localization technique takes advantage of the fact that GNSS receivers operating in close proximity and observing the same constellation of satellites have strongly correlated errors. These errors are largely canceled when a relative positioning system is taken into consideration. Both W. Li, *et al.* [64] and M. Woo, *et al.* [65] used V2V, GNSS and the distance among vehicles given by a vision and/or ranging sensor to compute a relative vehicle positioning, this method added more robustness to relative positioning. Although relative vehicle positioning is sufficient for platooning it is not enough to implement autonomous driving. GNSS accuracy can be enhanced by using carrier phase measurements (RTK-GPS). G. Challitasing, *et al.* [66] used V2V communications, RTK-GPS and a vehicle to vehicle ranging system (vision-based ranging system) for absolute positioning. Although this configuration is more robust, the RTK-GPS system employed is expensive and therefore not available for massive vehicle distribution, additionally the robustness is supported by relative measurements.

An approach taken for outdoor absolute positioning was the use of magnetic sensors and magnets (landmarks) buried along a path [62]. The *California Partners for Advanced Transit and Highways* (PATH) used on-board sensing system acquiring both the vehicle's state and road reference system based on magnets [67]. However in order to achieve good results the distance between magnets is required to be low, increasing the infrastructure costs. Both [68] and [69] used lane marking to improve the pose estimation. A localization method where the fusion of a mono-camera, a low-cost GNSS and a map data for intelligent vehicle is presented in [68]. This method uses lateral spatial information provided by a vision-based lane detection module and longitudinal and lateral spatial information provided by a vision-based traffic sign detection module for high-accuracy localization. The proposed method is economically feasible, does not need any change on the infrastructure environment. The proposed method achieved centimeter-level localization accuracy. In [69] a lane marking aided vehicle localization is presented. This paper proposes a solution that uses a lane detection system to retrieve accurate lateral and orientation vehicle information with respect to road lane markings, and then combine this information with GNSS estimates and deadreckoning sensors in order to provide localization information with high availability. A GNSS shaping filter combining random constant and first order auto-regressive models was implemented. Results show that the filter modeling, is able to fuse continuously GNSS fixes even if they are affected by large errors.

In [70] and [71], vehicles determine their positions in a collaborative way, by fusing their own sensor data with data shared by other users via a common communication network. These papers present collaborative navigation algorithms to increase the accuracy of vehicle positioning via the sensor information sharing through a V2V network. In [70] the algorithm generates GNSS differential corrections from a set of GNSS equipped vehicles by fusing GNSS measurements with a camera-based lane-boundary sensor. The results show that it is possible to generate an error-free differential correction that estimates the projection into the ground plane of the satellite-specific GNSS biases (ionosphere, troposphere, satellite clock) experienced by all collaborators in a local area. The benefits of the proposed method are more noticeable when the user density is high. In [71] an algorithm for differential GNSS corrections with no stationary reference receiver is proposed. The algorithm generates differential corrections using data from moving vehicles, thus eliminating the need for an infrastructure of stationary receivers. This algorithm generates individual differential corrections for each satellite, shared among vehicles with different satellites in view. Results show that the measurements sharing, significantly improve positioning accuracy in the crosstrack and in the along-track direction.

The knowledge of localization uncertainties is of prime importance when the navigation of intelligent vehicles has to deal with safety issues. To quantify the localization confidence V. Drevelle and P. Bonnifait [72] [73] developed several algorithms based on interval analysis and constraint propagation. The developed algorithms can handle several hypotheses in cases of ambiguous solutions simply by computing disconnected solution sets and are able to compute location zones in which the user is guaranteed to be located. A set-membership based satellite positioning aided by height data from a digital elevation model (DEM) for high integrity was developed in [72]. The integrity zone is computed recursively using a set-inversion method in a bounded-error context through set-bisection. Results show that the additional altitude information enabled more precise positioning while tolerating GNSS outliers, especially with a small number of visible satellites. In [73] V. Drevelle and P. Bonnifait further evolved their previous algorithm into two stages. On a first stage, tightly coupled position domains are computed by constraint propagation on GNSS measurements and precise 3D maps of the drivable space. A second stage provides localization integrity and information availability by the use of a position and proprioceptive data history. Results show that the algorithm is able to handle erroneous positions with a chosen integrity risk and in experiments carried out in urban canyons, despite bad satellite visibility, full positioning availability is obtained, and errors are less than 5.1 m during 95% of the trials. The measurements sharing, to improve positioning accuracy in the crosstrack and in the along-track direction, while maintaining localization integrity is still an open research area to be explored.

2.4 Final Remarks

A review on ITS simulation was presented, focusing on three subcategories: vehicular road traffic simulators; vehicle emission models and multi-purpose integrated simulators. The study of vehicular road traffic simulators revealed a wide range of simulators available, nevertheless all simulators lacked characteristics required for the mentioned thesis work goals, namely being proprietary and lack of detailed sensors simulation. The vehicle emission models studied were accurate but most of them did not allow to add new custom modules. The simulation of sensors and its noise characteristics as well as the actuators and its performance, were available in robotic simulators, but these simulators lacked the V2X communications.

The review of the state of the art regarding intelligent environment impact reduction through traffic management focused on four topics:

- vehicle emissions reduction through traffic reduction and transportation hardware improvements;
- vehicle emission reduction by traffic re-routing;
- vehicle emissions reduction through the vehicle speed control and traffic signal management;
- vehicle emissions reduction through the use of reservation-based intelligent traffic management.

The findings of this study revealed that by means of the reservation-based algorithms the emissions could be considerably reduced at a relative low cost. Another conclusion, was that to implement the traffic management in cooperative driving scenarios, the pose and speed of vehicles must be accurately determined.

Therefore a short study of the state of the art on vehicle localization, was performed, and revealed that a single sensor cannot provide all information required for this type of application, as these systems require not only accurate positioning but also the computation of the integrity risk, guaranteeing a safety margin between vehicles, therefore the fusion of multiple sensory information plays a key role in cooperative driving.

Chapter 3

Material and methods

Contents

The relevant material and methods pertaining to the thesis are presented in this chapter.

3.1 Sensor Modelling

The recent technology developments bring to the ITS new and more powerful tools and the latest advances in sensors miniaturization makes possible the integration of complex technological systems in a vehicle discreetly. Sensor modeling is both important for simulation as for experimental test, in this section the relevant simulated sensors are in detail described and characterized.

3.1.1 Global Navigation Satellite System (GNSS) positioning

Global Navigation Satellite System began as a military project in the United States on the 70's. The Global Positioning System (GPS) is fully functional since 1978 and available for civilian use since 1985. Today, several nations own this technology, namely Russia with GLObal NAvigation Satellite System (GLONASS), China with BeiDou Satellite Navigation System (BeiDou), while the European Galileo is expected to be fully functional in 2020. The focus will be the GPS system without significant loss of generality to other systems.

The GPS consists of three parts, or segments: the space segment, the control segment and the user segment. The space segment consists of 32 satellites with solar power, orbiting in six orbital planes at about 20000Km altitude. Each plane is inclined 55 degrees to the equatorial plane. This arrangement ensures visibility of at least four satellites at any land point with appropriate weather conditions. The control segment consists of several monitoring stations around the world and a central station, located in *Colorado Springs*. They monitor the satellites, estimate their position (ephemerides), calibrate the atomic clocks and make an update to the navigation information transmitted by the satellites. The user segment comprises all receivers used to make signal acquisition from the satellites.

The GNSS positioning methods (namely GPS) is based on the estimation of the distance between the receiver (*rover*) and the satellites (called *pseudorange*) through the signal propagation time. The signal propagation time is obtained by correlation of the received signal with a copy generated by the receiver. Knowing the distance from satellites to *rover* and the position of these satellites (*ephemerides*) one can determine the position of the receiver by trilateration.

The satellites transmit two main signals, the L1 carrier and L2 carrier. In order to distinguish the different signals from the two satellites carriers are modulated by Pseudo-Random Noise (PRN) sequences different for each satellite.

This modulation is divided in two binary sequences the coarse/acquisition (C/A) code with a chipping rate of 1.023 MHz, and precision (P) code with a chipping rate of 10.23 MHz. The (C/A) code is part of a code family usually designated has *golden codes*, characterized by having a low correlation among family members, making it particularly useful to distinguish signals from different satellites. The *L1* carrier phase at 1575.42 MHz, is modulated by both C/A and P codes. L2 carrier phase at 1227.60 MHz is modulated by P code. In addition to the *golden codes*, each carrier also transports the navigation message (D), containing information regarding *ephemerides* (satellite position), system status and GNSS satellite clock drift.

At the user segment, the receivers can be classified into two categories, those who receive and decode the two carriers (L1 and L2), are extremely accurate but quite expensive (thousands to tens of thousands of euros) and are mainly used in a professional context or research. The ones that only decode the L1 carrier have a lower cost tens of euros to a few hundred euros. The two carriers receivers can have an centimetric accuracy, while the ones capable of receiving only one carrier have a lower accuracy, ranging from a meters to tens of meters.

The satellite-receiver distance, from the satellite s to the receiver r, can be measured using *L1* carrier modulated by C/A binary sequence. This distance, designated by pseudorange, can be obtained by using either the C/A code or the carrier phase.

Automotive positioning systems need high accuracy. Two distinct ways to achieve the higher accuracy in real time using multi-antenna are, Differential GPS (DGPS) and RTK-GPS. In DGPS, a fixed base station antenna, sends differential corrections and the *rover* receiver correct its position based on the received corrections. In RTK-GPS,a fixed base station antenna, send their pseudoranges observations and the *rover* receiver performs pseudoranges differentiation process using both their own observations and base station antenna observations.

The GPSoft [74] software was used to emulate the GNSS system, namely the USA GPS constellation [75]. The GPSoft Toolbox emulates not only satellites and receivers but also the propagation channels. Error sources such as thermal noise, multipath, atmospheric delays and Selective Availability are modeled as an integral part of the pseudorange and integrated Doppler emulation. Furthermore, the errors are emulated such that the proper temporal and spatial correlation effects are observed in the measurements. This allows for realistic modeling of both *code* DGPS and *carrier-phase* RTK-GPS in addition to usual stand-alone positioning algorithms. GPSoft also enables emulation of Galileo, GEOs, GPS and GPS Modernization (C/A-code on L1, L2 and L5) as well as dual-frequency P-code measurements. The user can emulate signals on additional carrier frequencies defined by the user. The satellite constellation emulator supports GPS and Glonass as well as user-defined constellations. In addition, YUMA-format broadcast almanacs can be used. The emulation of C/A and P-code pseudorange and integrated Doppler with user definable civil and military carrier frequencies is available including characteristics such as thermal noise, ionospheric delay, tropospheric delay and diffuse multipath [74, 76].

Coordinate systems

In navigation, guidance, and control, there are several coordinate systems used in order to ease both position computations as well as graphical interpretation. In this section it is introduced the three major used coordinate systems for GNSS land vehicles position estimation, the Earth Centered Inertial (ECI), the Earth Centered Earth Fixed (ECEF) and the East North Up (ENU).

The vehicle coordinate system has its origin in the rear axle, the x axis (*Roll*) points towards the front of the vehicle, the y axis (*Pitch*) points to the left of the vehicle while the z axis (*Yaw*) point upwards. The ECI is the most used in land navigation, usually the Inertial Measurement Unit (IMU), return speed and angular accelerations observations using this coordinate system. The origin of this coordinate system is at the center of the earth, the x axis is permanently fixed in a direction relative to the celestial sphere, the z axis is aligned with the earth rotation axis and the y axis complete the system, according to the right-hand rule.

The ECEF coordinate system is geocentric, z axis points North, the axis x coincides with the intersection point of the prime meridian and the equator (latitude, longitude) point (0,0) and the axis y complete the system, according to the right-hand rule.

Closely related to the ECEF system is the geodetic coordinate system defined by World

Geodetic System (WGS) in 1984 (WGS84). The position is defined in terms of measured latitude and longitude angles to a reference ellipsoid. The ellipsoid which rotates around its minor axis is an approximation of the Earth's geoid. Although ECEF or ECI global coordinate systems provide correct positional description of any body in the terrestrial sphere, local systems such as ENU are more intuitive and practical. The ENU system is formed by a plane tangent to the Earth's surface at a specific position with the axis x pointing East and the the y axis pointing North.

GNSS error sources

There are several sources of error that may influence the performance of the position estimation system, either intrinsic or caused by the surrounding environment. One of the intrinsic sources of error, is the position of satellites on the sky dome, *i.e.* its spatial geometry. The estimation of the distance receiver-satellite is imprecise, the effect of this imprecision is magnified if the satellites are spatially close rather than spread on the sky dome which affects the trilateration process. This effect is quantified and denominated by Geometric Dilution of Precision (GDOP). GDOP quantifies the influence of satellite geometry on the quality of the receiver position estimation. Each satellite broadcast *Ephemeris*, the *Ephemeris* contain information about the position of satellites in space. The uncertainties on the exact position of each satellite, is also a common source of errors. Another intrinsic error is the clock errors of satellites, although very accurate small variations (*drift*) means a large positioning error.

The pseudorange measurements are affected by different physical phenomena that cause a delay in the propagation time of the signal. The signal propagation speed is influenced by the propagation medium, as the medium is not empty, signals travel at lower speeds than the speed of light, and suffers bias hard to model. The receiver independent error sources are: the ionospheric I_r^s and tropospheric T_r^s biases along the signal path; the satellite orbit estimation (or ephemeris) error E^s and the satellite clock offset dt_s. Both the level of ionisation of the ionosphere (which varies spatially and temporally) and the density of the gas molecules and moisture in the air in the troposphere, cause delays in the signal propagation. The receiver dependent error sources that affect the pseudorange measurement are: the pseudorange measurement noise and the receiver clock offset dt_r. I_r^s is generated as the signal passes through the upper layer of the atmosphere. The gases therein are ionized by solar radiation, resulting in a increase of the propagation time of the signal. The error introduced can be up to 50 m for low elevation satellites. T_r^s is generated as the signal passes through the lower layer of the atmosphere. This delay is mainly caused water vapor, and it ranges from 2.5 m at the zenith to 15 m for low satellite elevations. All other error sources, such as relativistic errors, multipath and thermal noise are lumped in ν_r^s.

Attenuation and/or reflections caused by trees, buildings and terrain irregularities (steep slopes near the receiver, mountains etc.) are also sources of errors, the reception of reflections is termed as *multipath.* The effect of *multipath* can cause large errors, because the receiver may be receiving a signal with a travel time higher. This effect, together with the poor quality of signal reception, are the main problems that arise, when estimating the position in urban areas, due to a high density of tall buildings.

Pseudorange code positioning: Iterative least squares GNSS Positioning algorithm (IGP)

The pseudorange measurement based on the *C/A* code has typical errors on the meter scale and it is obtained by the following: a PRN sequence modulated on the carrier, is sent by all satellites, each with its own pattern; when the receiver receives the signal, it is correlated with an internally generated replica and computes the time delay between the received signal and the replica.

The signal travel time from satellite to *rover* ($\Delta T_r^s(t)$), observed at the time t is given by [77]:

$$\Delta T_r^s(t) = t_{r(r)} - t^{s(s)} \tag{3.1}$$

where $t_{r(r)}$ is the *rover* signal reception epoch, measured using the rover clock and $t^{s(s)}$ is the satellite epoch of the transmitted signal, measured using the satellite clock. The clocks are not perfectly synchronized with GNSS time reference t_{GPS}, therefore one take into account the transmission and reception epochs taking into account the each clock *offset*:

$$t_{r(r)} = t_r + dt_r(t) \tag{3.2}$$

$$t^{s(s)} = t^s + dt^s(t) \tag{3.3}$$

where dt^s and dt_r are given by:

$$dt^s = t^{(s)} - t_{GPS} \tag{3.4}$$

$$dt_r = t_{(r)} - t_{GPS} \tag{3.5}$$

Taking into account the *offsets*, the signal travel time equation (3.1) is given by:

$$\Delta T_r^s(t) = t_r - t^s + dt_r(t) - dt^s(t) = \tau_r^s(t) + dt_r(t) - dt^s(t) \tag{3.6}$$

where $\tau_r^s(t)$ is the real propagation time between satellite and *rover*. By multiplying

$\Delta T_r^s(t)$ with the speed of propagation of light in the vacuum (c) gives the satellite to *rover* code observation equation [77], *i.e. pseudorange*:

$$P_r^s(t) = c\Delta T_r^s(t) = c\tau_r^s(t) + c(dt_r(t) - dt^s(t)) \tag{3.7}$$

Taking into account the GNSS error sources the code observation equation is given by:

$$P_r^s(t) = \rho_r^s(t) + c(dt_r(t) - dt^s(t)) + T_r^s(t) + I_r^s(t) + E^s(t) + \nu_r^s(t) \tag{3.8}$$

Without loss of generality and for the sake of readability the time t is removed in the following equations. The geometric range ρ_r^s from receiver r to satellite s is given by:

$$\rho_r^s = \sqrt{(x^s - x_r)^2 + (y^s - y_r)^2 + (z^s - z_r)^2} = \|\mathbf{x^s} - \mathbf{x_r}\| \tag{3.9}$$

where $\mathbf{x} = (x, y, z)$. Converting the receiver clock offset to distance units (co_r):

$$co_r = c \cdot dt_r \tag{3.10}$$

The satellite-specific GNSS errors sse^s; such as satellite clock offset, ephemeris, ionosphere and troposphere errors; are common to all receivers in a local area. These errors are spatially correlated errors for a particular satellite and therefore can be grouped:

$$sse^s = c \cdot (-t^s) + I_r^s + T_r^s + E^s \tag{3.11}$$

and lumping sse^s and ν_r^s errors:

$$v_r^s = sse^s + \nu_r^s \tag{3.12}$$

from (3.9), (3.10) and (3.12) one gets:

$$P_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r + v_r^s \tag{3.13}$$

By iteratively applying the following steps one can compute the receiver position: first use a priori estimate of state, then predict the pseudoranges you would get with that state. Based on the difference between the actual and predicted pseudorange measurements, update the a priori state. Linearizing the pseudorange observations P_r^s, using the first order Taylor expansion, one get the following:

$$\begin{aligned}
&P_r^s(x_r, y_r, z_r, co_r) \cong P_r^s(x_{r0}, y_{r0}, z_{r0}, co_{r0}) + (x_r - x_{r0})\tfrac{\delta P_r^s}{\delta x_r} + (y_r - y_{r0})\tfrac{\delta P_r^s}{\delta y_r} + (z_r - z_{r0})\tfrac{\delta P_r^s}{\delta z_r} \\
&+(co_r - co_{r0})\tfrac{\delta P_r^s}{\delta co_r} \\
&\Leftrightarrow \\
&P_r^s(x_r, y_r, z_r, co_r) - P_r^s(x_{r0}, y_{r0}, z_{r0}, co_{r0}) = \tfrac{\delta P_r^s}{\delta x_r}\Delta x_r + \tfrac{\delta P_r^s}{\delta y_r}\Delta y_r + \tfrac{\delta P_r^s}{\delta z_r}\Delta z_r + \tfrac{\delta P_r^s}{\delta co_r}\Delta co_r
\end{aligned} \tag{3.14}$$

and converting into an iterative notation:

$$\Delta P_r^s = P_r^s(k+1) - P_r^s(k) = \tfrac{\delta P_r^s}{\delta x_r}\Delta x_r + \tfrac{\delta P_r^s}{\delta y_r}\Delta y_r + \tfrac{\delta P_r^s}{\delta z_r}\Delta z_r + \tfrac{\delta P_r^s}{\delta co_r}\Delta co_r + v_r^s \tag{3.15}$$

where k denotes the iterations processed in the current epoch until the solution converges. Rewriting (3.15) in the matrix form:

$$\begin{bmatrix} \Delta P_r^1 \\ \vdots \\ \Delta P_r^m \end{bmatrix} = \begin{bmatrix} \frac{\delta P_r^1}{\delta x_r} & \frac{\delta P_r^1}{\delta y_r} & \frac{\delta P_r^1}{\delta z_r} & \frac{\delta P_r^1}{\delta co_r} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\delta P_r^m}{\delta x_r} & \frac{\delta P_r^m}{\delta y_r} & \frac{\delta P_r^m}{\delta z_r} & \frac{\delta P_r^m}{\delta co_r} \end{bmatrix} \begin{bmatrix} \Delta x_r \\ \Delta y_r \\ \Delta z_r \\ \Delta co_r \end{bmatrix} + \begin{bmatrix} v_r^1 \\ \vdots \\ v_r^m \end{bmatrix} \tag{3.16}$$

from (3.16) one can write the equation for a single satellite f, the pseudorange residual is given by:

$$\Delta P_r^f = -e_r^f \cdot \Delta \mathbf{x_r} + \Delta co_r + v_r^f \tag{3.17}$$

where e_r^f is the pointing vector from the user r to satellite f.

$$\Delta e_r^f = \begin{bmatrix} \frac{(x^s - x_r)}{\|\mathbf{x^s} - \mathbf{x_r}\|} & \frac{(y^s - y_r)}{\|\mathbf{x^s} - \mathbf{x_r}\|} & \frac{(z^s - z_r)}{\|\mathbf{x^s} - \mathbf{x_r}\|} \end{bmatrix} \tag{3.18}$$

If the receivers are assumed to be in close proximity, the pointing vector is the same for all users and e_r^f can be substituted by e^f. Equation (3.18) can be written in the following form:

$$\Delta P_r = \mathbf{H} \begin{bmatrix} \Delta \mathbf{x_r} \\ \Delta co_r \end{bmatrix} + v_r^s \tag{3.19}$$

where $\mathbf{H}$ is the *observation matrix*:

$$\mathbf{H} = \begin{bmatrix} -e^1 & 1 \\ \vdots & \vdots \\ -e^m & 1 \end{bmatrix} \tag{3.20}$$

and ΔP_r it is given by:

$$\Delta P_r = \begin{bmatrix} \Delta P_r^1 & \dots & \Delta P_r^m \end{bmatrix}^T \tag{3.21}$$

Using a single epoch, the receiver positioning can only be achieved if we have at least four observations (*i*.e. receiving signals from at least four satellites), due to the presence of 4 unknowns (x_r, y_r, z_r and co_r). Solving equation (3.19) using common least squares, one obtains:

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}_\mathbf{r} \\ \Delta\hat{co}_r \end{bmatrix} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta P_r \tag{3.22}$$

Thus by using satellite positions and measured pseudoranges one can compute an estimate for both the receiver position $\hat{\mathbf{x}}_\mathbf{r}$ and the receiver clock offset $\hat{co}_r$, via Iterative Least Squares GNSS positioning (IGP) method [75].

The iterative least squares solution $[\hat{\mathbf{x}}_\mathbf{r}, \hat{co}_r]^T$ can now be used to estimate the measurements $\hat{P}_r$:

$$\hat{P}_r = \mathbf{H} \begin{bmatrix} \hat{\mathbf{x}}_\mathbf{r} \\ \hat{co}_r \end{bmatrix} \tag{3.23}$$

Using the measurements and their estimate it is possible to compute residuals vector W_{res}:

$$W_{res} = P_r - \hat{P}_r \tag{3.24}$$

Pseudorange phase positioning

Pseudorange phase positioning is similar to the code positioning but in this case the carrier offset is used to compute the pseudorange. The phase observations compare the received signal with a replica generated in the receiver and the carrier offset is computed.

The observation equation is defined in (3.25) where $\Phi_r^s(t)$ is the measured phase shift (*offset*), $\Phi_r(t)$ is the replica generated in the receiver and $\Phi_r(^s_r)(t)$ is the phase signal sent by the satellite s received by receiver r at time t [77].

$$\Phi^s_r(t) = \Phi_r(t) - \Phi_r(^s_r)(t) \tag{3.25}$$

Assuming that there is no phase change over the signal route from sattelite to *rover*, then 3.25 can be rewritten:

$$\Phi_r(^s_r)(t) = \Phi^s(t - \tau^s_r) \tag{3.26}$$

where τ^s_r is the signal propagation time. Equation 3.26 can be written as an expansion in Taylor series of the first order:

$$\Phi^s(t - \tau^s_r) = \Phi^s(t) - \frac{d\Phi^s}{dt}\tau^s_r = \Phi^s(t) - f\tau^s_r + N^s_r(t) \tag{3.27}$$

where $f = \frac{d\Phi}{dt}$ is the frequency of a stable oscillator, and $N^s_r(t)$ is the integer carrier phase cycle ambiguity between satellite s and receiver r, which cannot be directly observed. Using equations 3.26 and 3.27 equation 3.25 can be rewritten as follows:

$$\Phi^s_r(t) = \Phi_r(t) - \Phi^s(t) + f\tau^s_r - N^s_r(t) \tag{3.28}$$

Similar to the code observations, both *rover* and satellite clocks have offsets, and are not perfectly in phase with the phase of an ideal oscillator synchronized with the GNSS time reference Φ_{GPS}:

$$\Phi_r(t) = \Phi_{GPS}(t) + f dt_r(t) \tag{3.29}$$

$$\Phi^s(t) = \Phi_{GPS}(t) + f dt^s(t) \tag{3.30}$$

Taking into account the phase clock *offsets*, equation 3.28 is rewritten as follows:

$$\Phi^s_r(t) = f\tau^s_r + f(dt_r(t) - dt^s(t)) - N^s_r(t) \tag{3.31}$$

Taking into account the GNSS error sources the phase observation, and given that $f = \frac{c}{\lambda}$, the equation of the carrier phase pseudoranges is given by:

$$\lambda\Phi^s_r(t) = \rho^s_r(t) + c(dt_r(t) - dt^s(t)) - I^s_r(t) + T^s_r(t) + E^s_r(t) - \lambda N^s_r(t) + \eta^s_r(t) \tag{3.32}$$

where η^s_r is the phase measurement noise. Compared to the solution of the code pseudoranges, the carrier phase pseudoranges, given by equation 3.32, adds one unknown variable per satellite $N_r^s(t)$. This initial phase ambiguity has to be estimated for each satellite, and therefore it is necessary to increase the number of observation epochs to solve the system and to obtain the coordinates of the *rover* r.

The carrier phase pseudoranges errors are centimetric, which are significantly lower than the code pseudoranges, therefore carrier phase pseudoranges positioning should produce better estimation position results. On the other hand, it needs better quality receivers (antennas, oscillators, etc.), additionally the solution to the carrier phase pseudoranges, given by equation (3.32), requires highly complex and heavy computing algorithms to solve the integer ambiguities N_r^s, meaning that low-cost devices only provide code observations as the positioning method.

Single differencing positioning

As mentioned previously, pseudoranges observations are affected by a variety of errors, some of them can be reduced or totally removed by applying the relative positioning between two receivers.

Single differentiation is the simplest method of differentiation, it involves only two receivers observing simultaneously the same satellite. This method can be applied for both code and carrier phase pseudoranges [77].

Considering two receivers r_i and m, where i is the number of rovers (without loss of generality i is considered to be $i = 1$) and m a reference station antenna, observing the same satellite s at the time t, then the single differentiation equation for code observations is given by equation (3.33):

$$\begin{aligned} P_r^s(t) &= P_{r_i}^s(t) - P_m^s(t) \\ &= \rho_{r_im}^s(t) + c(dt_{r_i}(t) - dt_m(t)) + I_{r_im}^s(t) + T_{r_im}^s(t) + \nu_{r_im}(t) \end{aligned} \tag{3.33}$$

The single differentiation equation for phase observations is similar and given by (3.34):

$$\begin{aligned} \lambda\Phi_{r_i}^s(t) &= \lambda\Phi_{r_i}^s(t) - \lambda\Phi_m^s(t) \\ &= \rho_{r_im}^s(t) + c(dt_{r_i}(t) - dt_m(t)) + I_{r_im}^s(t) + T_{r_im}^s(t) \\ &+ \lambda N_{r_im}^s(t) + \eta_{r_im}^s(t) \end{aligned} \tag{3.34}$$

The satellite clock bias $(dt^s(t))$ is eliminated by using single differentiation, nevertheless both receiver clock biases $(dt_{r_i}(t)$ and $dt_m(t))$ are still present.

The components of the troposphere and ionosphere errors are common to both both code and carrier phase single differentiation observations:

$$I^{s}_{r_im}(t) = I^{s}_{r_i}(t) - I^{s}_{m}(t) \tag{3.35}$$

$$T^{s}_{r_im}(t) = T^{s}_{r_i}(t) - T^{s}_{m}(t) \tag{3.36}$$

If the receivers are spatially close, so that the atmospheric conditions in the two receivers are the same, then the contribution of $I^{s}_{r_im}(t)$ and $T^{s}_{r_im}(t)$ to the observation tend to be very small, or even zero. This is another advantage of single differentiation, errors of ionospheric and troposphere propagation are mitigated when the two receivers are spatially close, meaning that both ionospheric and tropospheric atmospheric noises are negligible.

Double differencing positioning

When using single differenced observation equations for two receivers r_i $(i = 1)$ (*rover*) and m (*master*), if it is possible to observe an additional satellite p besides the satellite s, then a *double differentiation* method can be applied to the observations, i.e. when two receivers r_i $(i = 1)$ and m observe two satellites s and p at the same time t, the observation equation can be differentiated again [77]. This procedure so called *double differencing* can be applied for both *single differenced* pseudorange code (3.37) and *single differenced* pseudorange carrier phase (3.38) :

$$\begin{aligned} P^{ps}_{r_im}(t) &= P^{p}_{r_im}(t) - P^{s}_{r_im}(t) \\ &= \rho^{ps}_{r_im}(t) + I^{ps}_{r_im}(t) + T^{ps}_{r_im}(t) + \nu^{ps}_{r_im}(t) \end{aligned} \tag{3.37}$$

$$\begin{aligned} \lambda\Phi^{ps}_{r_im}(t) &= \lambda\Phi^{p}_{r_im}(t) - \lambda\Phi^{s}_{r_im}(t) \\ &= \rho^{ps}_{r_im}(t) + I^{ps}_{r_im}(t) + T^{ps}_{r_im}(t) \\ &+ \lambda N^{p}_{r_im}s(t) + \eta^{ps}_{r_im}(t) \end{aligned} \tag{3.38}$$

Assuming that the observations are from the same epoch and that clock *drifts* between epoch is small, the double differentiation eliminates receiver clock errors.

According to equations (3.37) and (3.38) the double differentiation process, eliminates all clocks related errors (receivers and satellites), being sensible to measurement noise and propagation errors modeled by $I^{ps}_{r_im}(t)$ and $T^{ps}_{r_im}(t)$. As in single differentiation, the unmodeled atmospheric errors of both ionosphere and troposphere, are greatly reduced. By reducing or eliminating a large part of of the typical GNSS, the double differentiation technique delivers a very accurate positioning solution.

3.1.2 Magnetic Sensing System setup and model (MSS)

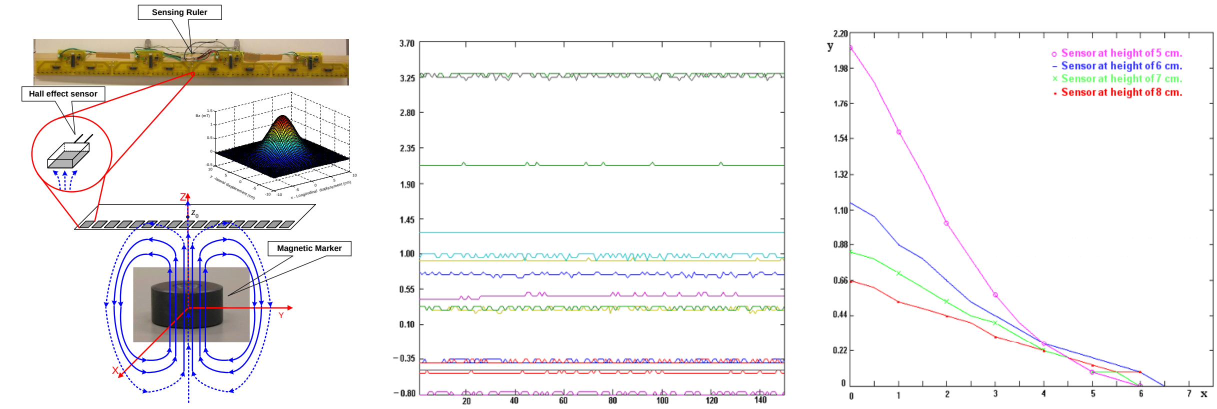
The MSS is composed by Hall sensors that are triggered when they cross over magnetic markers, which are placed in the ground defining center points of the path to be followed by the vehicles, and by Hall effect sensors mounted on the vehicle, sensing the magnetic field of the magnetic markers.
The MSS was used in both simulations and in real experiments.

Regarding the real setup, the Hall effect sensors are placed in line adjacently creating a sensor ruler, as shown in Figure 3.1(a). Each ruler is composed by sub-modules, each one with a microcontroller that acquires data from a set of Hall effect sensors. All the information from the microcontroller sub-modules is conveyed to a local microcontroller that computes the lateral deviation relatively to the center of the magnetic marker. All the connections among the microcontrollers as well as the communication to the vehicle host computer is done using Controller Area Network Bus (CAN-BUS). Each all effect sensor is sampled at rate of 1.3 KHz. Figure 3.1(b) illustrates the response of a set of Hall effect sensors having no magnetic marker in the sensed field. As can be observed there is an offset, which is different for each sensor. Figure 3.1(c) shows the sensed vertical magnetic component with the sensor, in different locations, relatively to the cylindric marker; the sensor was positioned along the vertical axis of the center point and moved towards out the center. In Figure 3.1(c) it is shown four scans at different heights.

The environment of an autonomous car has several sources of magnetic noise namely: earth magnetic field, vehicle power transmission lines, and electrical fields inherent to the vehicle. These sources influences the magnetic measurements. This undesirable effect is reduced by estimating and eliminating the offsets that characterize each sensor.

For the detection of the marker (i.e. its position) it is used the knowledge of the sensor response model (see Figure 3.2(a) and Figure 3.2(b)). When a marker detection is validated, after data processing and test procedures, this information is sent via CAN-BUS to the computer in charge of the control of the vehicle.

Position measurement using magnetic markers has been proven to be one promising technology for ground vehicle autonomous guidance and control. No physical contact is needed to produce measures, the magnetic markers reliability is independent of weather conditions, and maintenance requirements are low since the system uses passive rather than active markers.
This motivated the development of a position reference system based on magnetic markers. However there are some disadvantages too, magnetic sensors also measure the magnetic fields from the earth, the vehicle, and possibly from other sources, which combined define an undesirable magnetic field.



(a) (b) (c)

Figure 3.1: Magnetic Sensing System (MSS): (a) Diagram of the position measurement system using magnet markers. (b)[Real] Hall effect sensors responses (x-axis - sample number, y-axis - sensed data in [mT]); (c) [Real] Vertical component of Magnetic Field at different heights (x-axis - distance relatively to the cylindric marker in [cm], y-axis - sensed data in [mT]).

Getting a representative model of the magnetic marker behavior, is important for the design of a reliable processing algorithm. The researchers from PATH have chosen to model the magnetic marker as a magnetic dipole [78]. This is a very simple representation and easily understandable. The PATH researchers had proved that exist a strong correlation between the model predictions and the empirical tests. With a good model we can distinguish noise disturbance from the magnetic field radiated from the magnetic marker.

In real environments the detection of the magnets doesn't return the exact center of the magnet, but rather a coordinate close to its z-axis center, so in order to have measures similar to the real experiments a representative model of the magnetic field radiated by the magnetic marker was simulated.

Under the assumption that a magnetic marker can be modeled like a magnetic dipole, the magnetic field, $B(x, y, z)$, at an arbitrary point $P(x, y, z)$, can be given by[1]:

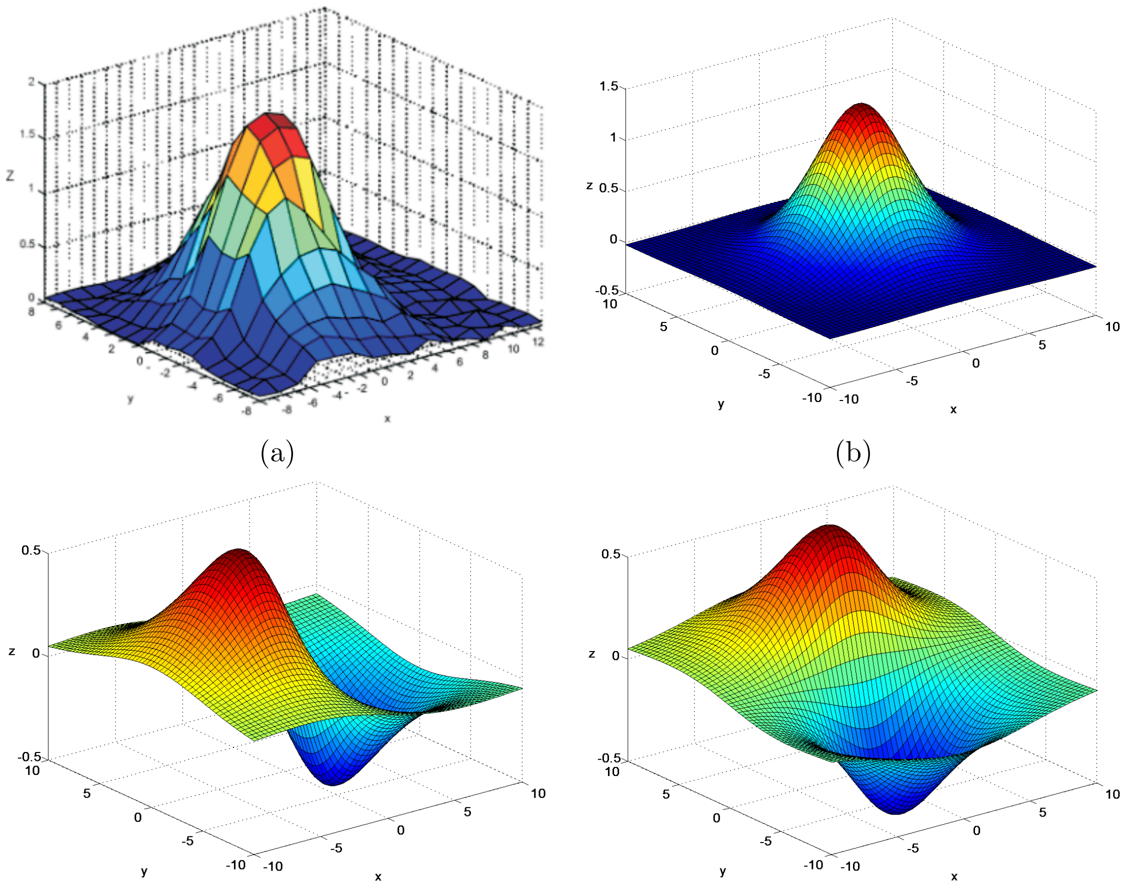
$$B = \frac{\mu_0 M}{4\pi r^5}(3xz\hat{i} + 3yz\hat{j} + (2z^2 - x^2 - y^2)\hat{k}) \tag{3.39}$$

with: $r = \sqrt{x^2 + y^2 + z^2}$

where M is the magnetic moment of the magnetic marker, and the coordinate system is chosen with x-axis ($\hat{i}$) being the longitudinal axis that corresponds to the the direction of vehicle travel, y-axis ($\hat{j}$) being the transversal axle that corresponds to the lateral deviation, and z-axis ($\hat{k}$) that corresponds to the height relative to the marker center (as illustrated in Figure 3.1(b)).

Figure 3.2(a) shows real data of a field emanated from a cylindrical magnet and sensed

[1] In cgs units. Conversion factor: $T = 10^4 G$



(a) (b) (c) (d)

Figure 3.2: Magnetic marker model: [Real] (a) Vertical component of the magnetic field emanated by a cylindrical magnet sensed by a Hall effect transducer at the height of 10cm ; [Simulated] (b) Vertical (c) Longitudinal and (d) Lateral component of the magnetic field (x-axis and y-axis in [cm] and z-axis in [mT], with sensor at height of 6.75cm).

by a Hall effect transducer aligned with the z-axis at the height of $10cm$ (detection range up to $25cm$). From the analysis of the vertical field component illustrated in Figure 3.2(b), computed using equation (3.39) it is possible to observe that the vertical field reaches the maximum when the sensor is exactly above of the magnetic marker center, this is the strongest component of the magnetic field. There is a strong correlation between the model predictions (equation 3.39) and the empirical tests as it can be observed when compared Figure 3.2(a) and Figure 3.2(b).

The longitudinal and lateral components makes a steep transition near the marker as it changes its sign, this could be meaningful in interpreting the point at which a sensor passes through the marker (see Figure 3.2(c) and Figure 3.2(d)).

3.1.3 Laser Range-bearing

A laser range-bearing, also known as LIDAR, is a device used to scan the distances of surrounding objects. The most common techniques employed by this type of devices are the 'time of fly' (where the distance is obtained by measuring the time of travel of a light pulse) or the 'phase-shift' (technique applied to a laser beam with sinusoidal modulated optical power to determine the distance). The matlab high-level programming environment caused serious restrictions (large processing times) on the LIDAR simulation, making impracticable to simulate large maps with centimetric resolution. Therefore the LIDAR module was implemented in C++ language embedded in matlab environment. To simulate the LIDAR, the entire map as divided in cells, with configurable granularity. For increased performance, the entire LIDAR simulation operation is performed at once for all simulated vehicles. The simulation of the each laser beam is based on a computer graphics algorithm denominated *Bresenham's line algorithm* (see algorithm 1) [79]. This algorithm is used to color pixels in order to draw a line independently of pixel granularity, and represent a good solution to follow a line segment in a map divided in cells.

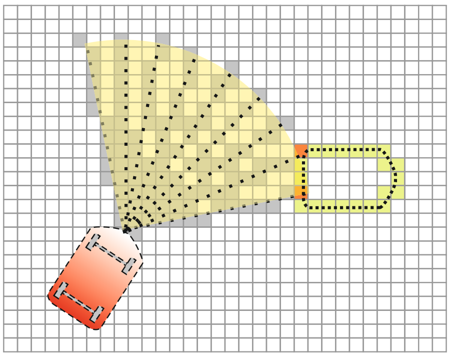


Figure 3.3: Operation of LIDAR module with neighbour vehicle detected.

```
**Require:** $x_0, y_0, x_1, y_1$
  Calculate difference in X-axis and Y-axis
  Decide main axis and secondary axis
  Set increment/decrement for both axis
  while The point $(x_1, y_1)$ is not reached do
    Calculate axis difference
    if Nearest pixel is on next secondary increment then
      Increment secondary axis
    end if
    Increment main axis
    Analyze pixel defined for $(x, y)$
  end while
```

Algorithm 1: Bresenham's line algorithm generalization to all quadrants to analyze cells that color a line segment defined for (x_0, y_0) (x_1, y_1)

When the algorithm finds a occupied cell (see Figure 3.3) a collision is detected and the distance between LIDAR origin and the collision place is calculated and stored. If no collision is detected the distance assigned to the beam is the maximum range of LIDAR. The group of distances measured by the LIDAR forms the output of the module. An important feature of this module, beyond configurable map granularity, is the possibility to adjust some LIDAR features such as: Maximum range, scan angle, angular resolution and scan rate.

3.2 Kinematics and Odometry Vehicle Modelling of Double Steered Vehicles

The configuration of a car-like vehicle can be described without ambiguity by (x, y, θ) (see Figure 3.4):

- x and y are the coordinates of the rear axle center with respect to the ${}^{W}\Sigma$ coordinates
- θ is the vehicle heading with respect to the ${}^{W}\Sigma$ coordinates

The vehicle kinematic equations are derived according to pure rolling, non-slipping and rigid body assumptions. Therefore a linear velocity vector and instantaneous rotation center exists at the reference frame located at the midpoint of the rear axle ${}^{R}\Sigma$ and the velocity is directed along the vehicle axle. Kinematics models have the property of keeping the steering and velocity of the vehicle completely decoupled, making the kinematics based control design easier.

Table 3.1: Vehicle Model Geometric Parameters

L	car length (distance between axles)	e	half-track (half car width)
Ψ_R	steering angle from the front right wheel	Ψ_L	steering angle from the front left wheel
Ψ	steering angle of the virtual front wheel	$F = (x_F; y_F)$	coordinate system of the front axle center
V_F	instantaneous speed at the front axle center	D_F	radius curvature from virtual front wheel
D_{FL}	radius curvature from front left wheel	D_{FR}	radius curvature from front right wheel
χ_R	steering angle from the rear right wheel	χ_L	steering angle from the rear left wheel
χ	steering angle of the virtual rear wheel	$R = (x_R; y_R)$	coordinate system of the rear axle center
V_R	instantaneous speed at the rear axle center	D_R	radius curvature from virtual rear wheel
D_{RL}	radius curvature from rear left wheel	D_{RR}	radius curvature from rear right wheel
ICC_V	orthogonal projection of the ICC onto	ICC	instantaneous curvature center
	the longitudinal axis of the vehicle	ρ	Length between ICC and ICC_V

The simulated vehicles include Two Wheel Steering (2WS) and Four Wheel Steering (4WS), the later has the ability to steer both the rear and the front pair of wheels [80]. In this section it is presented double steered model , the front steered model is described in the Appendix B. The classical model was used, which considers an imaginary wheel at the midpoint of the wheels axles, so that it is oriented in the direction of the steering command. The vehicle geometrical configuration parameters are illustrated in Figure 3.4 and summarized in Table 3.1.

3.2.1 Kinematics model

To deduce the kinematics equation of the double steered model, let us consider a point ICC_V which is defined as the orthogonal projection of the instantaneous curvature center (ICC) onto the longitudinal axis of the vehicle, see Figure 3.4 [81].

The distance between ICC and ICC_V can also be written in the following form,

$$\overline{(ICC)(ICC_V)} = \frac{\overline{(R)(ICC_V)}}{\tan(\chi)} = \frac{\overline{(ICC_V)(F)}}{\tan(\Psi)} \tag{3.40}$$

form (3.40) one can derive

$$\overline{(R)(ICC_V)} = \overline{(ICC_V)(F)} \cdot \frac{\tan(\chi)}{\tan(\Psi)} \tag{3.41}$$

where

$$\overline{(R)(ICC_V)} + \overline{(ICC_V)(F)} = L \tag{3.42}$$

now one can obtain $\overline{(R)(ICC_V)}$ and $\overline{(ICC_V)(F)}$ function of Ψ, χ and L,

$$\begin{cases} \overline{(R)(ICC_V)} = \frac{L}{1+\frac{\tan(\Psi)}{\tan(\chi)}} & = \quad L \cdot \frac{\cos(\Psi)\cdot\sin(\chi)}{\sin(\Psi+\chi)} \\ \overline{(ICC_V)(F)} = \frac{L}{1+\frac{\tan(\chi)}{\tan(\Psi)}} & = \quad L \cdot \frac{\cos(\chi)\cdot\sin(\Psi)}{\sin(\Psi+\chi)} \end{cases} \tag{3.43}$$

where the curvature radius D_R and D_F are given by the following equation,

$$\begin{cases} D_R = \frac{\overline{(R)(ICC_V)}}{|\sin(\chi)|} & = \quad L \cdot \frac{\cos(\Psi)}{|\sin(\Psi+\chi)|} \\ D_F = \frac{\overline{(ICC_V)(F)}}{|\sin(\Psi)|} & = \quad L \cdot \frac{\cos(\chi)}{|\sin(\Psi+\chi)|} \end{cases} \tag{3.44}$$

The instantaneous rotation speed can be written as follows,

$$\dot{\theta} = \frac{V_R}{D_R} = \frac{V_F}{D_F} \tag{3.45}$$

From equations (3.44) and (3.45), one can derive the following equation,

$$V_R = V_F \cdot \frac{D_R}{D_F} = V_F \cdot \frac{\cos(\Psi)}{\cos(\chi)} \tag{3.46}$$

therefore the equations of movement for the rear axle and front axle are given by equations (3.47) and (3.48) respectively.

$$\begin{cases} \dot{x_R} = V_R \cdot \cos(\theta+\chi) \\ \dot{y_R} = V_R \cdot \sin(\theta+\chi) \\ \dot{\theta} = V_R \cdot \frac{\sin(\Psi+\chi)}{L\cdot\cos(\Psi)} \end{cases} \tag{3.47}$$

$$\begin{cases} \dot{x_F} = V_F \cdot \cos(\theta+\Psi) \\ \dot{y_F} = V_F \cdot \sin(\theta+\Psi) \\ \dot{\theta} = V_F \cdot \frac{\sin(\Psi+\chi)}{L\cdot\cos(\chi)} \end{cases} \tag{3.48}$$

one can also express the equations of movement of the rear axle function of the front instantaneous velocity V_F and the front axle function of the rear instantaneous velocity V_R; equations (3.49) and (3.50) respectively.

$$\begin{cases} \dot{x_R} = V_F \cdot \frac{\cos(\Psi)}{\cos(\chi)} \cdot \cos(\theta + \chi) \\ \dot{y_R} = V_F \cdot \frac{\cos(\Psi)}{\cos(\chi)} \cdot \sin(\theta + \chi) \\ \dot{\theta} = V_F \cdot \frac{\sin(\Psi + \chi)}{L \cdot \cos(\chi)} \end{cases} \quad (3.49) \qquad \begin{cases} \dot{x_F} = V_R \cdot \frac{\cos(\chi)}{\cos(\Psi)} \cdot \cos(\theta + \Psi) \\ \dot{y_F} = V_R \cdot \frac{\cos(\chi)}{\cos(\Psi)} \cdot \sin(\theta + \Psi) \\ \dot{\theta} = V_R \cdot \frac{\sin(\Psi + \chi)}{L \cdot \cos(\Psi)} \end{cases} \quad (3.50)$$

note that if $\chi = 0$ we have the equation obtained in section (B). Then the kinematic model of the vehicle, assuming wheels rolling without slipping, for a reference frame located at point ICC_V, is given in the matrix form by the following equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\Psi} \\ \dot{\chi} \end{bmatrix} = \begin{bmatrix} \cos(\theta + \chi) \\ \sin(\theta + \chi) \\ \frac{\sin(\Psi + \chi)}{L \cos(\chi)} \\ 0 \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} v_2 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_3 \qquad (3.51)$$

Where v_1 represents the linear velocity of the vehicle, v_2 is the angular velocity of the front steering wheel, v_3 is the angular velocity of the rear steering wheel, L is the distance between the rear and front axles, χ is the rear steering angle, Ψ is the front steering angle and θ is the vehicle orientation in the world coordinate system.

3.2.2 Odometry model

A good odometry model for the bi-steerable vehicle is defined by the following equation [82], given that a local circular trajectory assumption was made [81]:

$$\begin{cases} x_{k+1} = x_k + \Delta_R \cdot \cos(\theta_k + \chi_k + \omega_R/2) \\ y_{k+1} = y_k + \Delta_R \cdot \sin(\theta_k + \chi_k + \omega_R/2) \\ \theta_{k+1} = \theta_k + \omega_R \end{cases} \qquad (3.52)$$

where Δ_R is the arc length and ω_R the elementary rotation. Assuming that there is no wheel slippage and using only data from the rear wheels encoders and rear steering encoder, then

$$\Delta_R = \frac{\Delta_{RR} + \Delta_{RL}}{2} \quad , \omega_R = \frac{\Delta_{RR} - \Delta_{RL}}{2 \cdot e \cdot \cos(\chi)} \qquad (3.53)$$

where e is the half distance between wheels, and Δ_{RR} and Δ_{RL} are calculated using the right and left wheel encoders measurements, respectively.

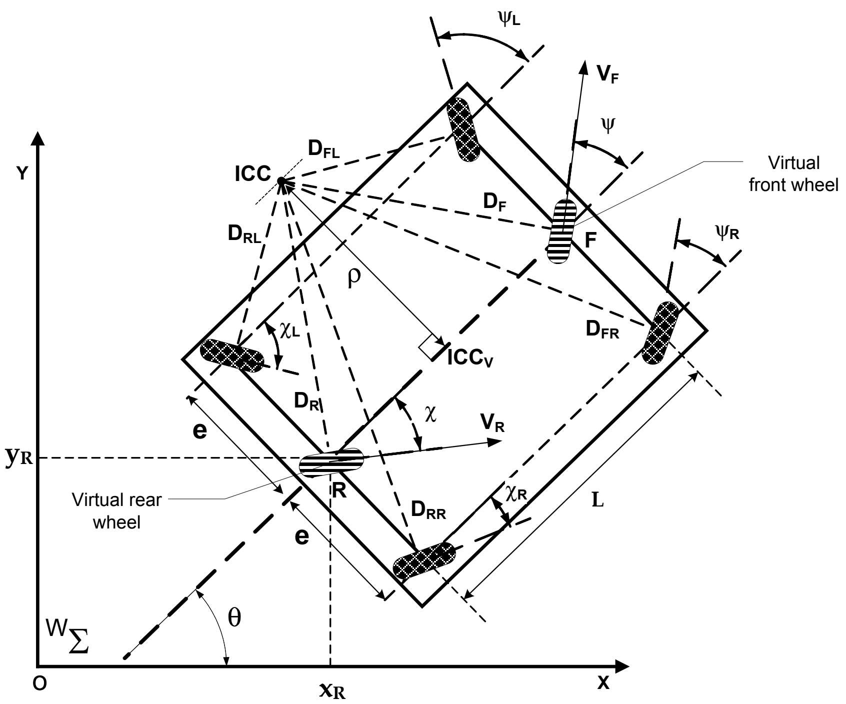


Figure 3.4: Double steered Vehicle geometrical configuration. ${}^{W}\Sigma$ and ${}^{R}\Sigma$ represent respectively the world coordinate system and the vehicle local coordinate system with its origin at the midpoint of the rear axle and its x-axis aligned with the longitudinal axis of the vehicle

3.2.3 Odometry disturbances

One of the most common methods used to estimate a vehicle pose (position and orientation) is odometry, and therefore its inclusion in simulations is paramount. This method uses encoders, coupled to vehicles wheels, to measure the wheel displacement and the measures are integrated to determinate an estimated pose. Odometry is prone to error, therefore to emulate disturbances more realistically, is it necessary to integrate systematic and non-systematic errors. The displacement of a wheel i, $\Delta S_i \in [\Delta_{RR}, \Delta_{RL}, \Delta_{FR}, \Delta_{FL}]$ is given by:

$$\Delta S_i = \frac{2\pi \times R_i \times N_i}{N_{rev}} \qquad [Meters] \tag{3.54}$$

where R_i is the wheel radius, N_i the measured pulses and N_{rev} are the pulses per revolution. The measured pulses are simulated by:

$$N_i = \Delta W_i \, \frac{N_{rev}}{2\pi} \qquad [Pulses] \tag{3.55}$$

where ΔW_i is the angular displacement of a wheel i.

Several disturbances where considered on the simulations related with odometry. To each simulated wheel, an independent variance Gaussian noise was added to the its radius and to the encoders besides the quantization error, an integer Gaussian noise was added.

The simulated steering wheel encoder also suffer from quantization error and an integer Gaussian noise was added. Gaussian noise was added to the lock to lock front wheel steering angle variable. The odometry was also contaminated with two independent Gaussian noise on the wheel to wheel distance $2e$ and on the front to rear axle distance L.

To add more realism to the simulator, it is possible to insert slippy areas on roads. In these areas the vehicles wheels will slip, and when this occurs, the generated pulses for the wheel that slips will be above the expected values with noise.

3.3 Interval analysis and Set inversion

Interval analysis [83] involves intervals and their multidimensional extension, interval vectors (or boxes). A box is a subset of $\mathbb{R}^n$ defined as the cartesian product of n intervals $[\mathbf{x}]$. The set of real intervals is denoted $\mathbb{IR}$, and the set of n-dimensional boxes is $\mathbb{IR}^n$ Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a given function, the interval function $[\mathbf{f}]$: $\mathbb{IR}^n \to \mathbb{IR}^m$ is an *inclusion function* for $\mathbf{f}$ if:

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]) \tag{3.56}$$

To approximate compact sets in a guaranteed way, subpavings are used. A subpaving of a box $[\mathbf{x}]$ is the union of nonempty and non-overlapping subboxes of $[\mathbf{x}]$. A guaranteed approximation of a compact set $\mathbb{X}$ can be made by bracketing it between an inner subpaving $\underline{\mathbb{X}}$ and an outer subpaving $\overline{\mathbb{X}}$ such as $\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}}$ (see Figure 3.5).

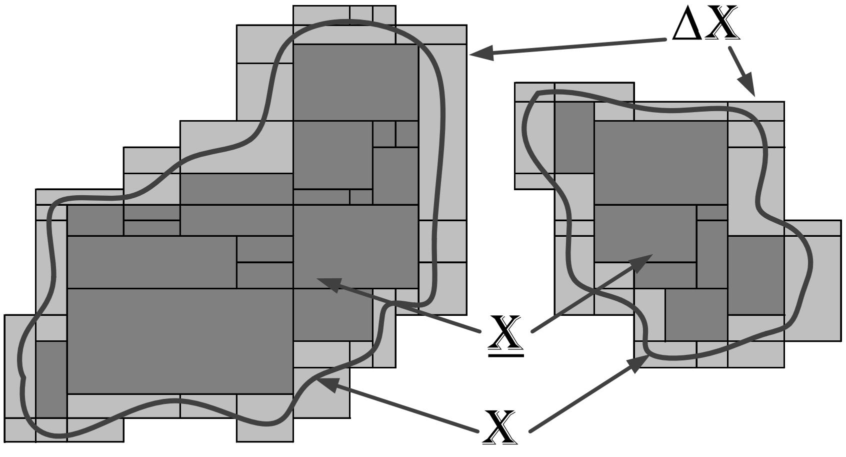


Figure 3.5: Compact set $\mathbb{X}$: bracketed between an inner subpaving $\underline{\mathbb{X}}$ and an outer subpaving $\overline{\mathbb{X}}$, where $\overline{\mathbb{X}}$ is given by $\overline{\mathbb{X}} = \underline{\mathbb{X}} + \Delta\mathbb{X}$.

3.3.1 Set inversion

Given a known interval vector $\mathbb{Y}$ of m measurements, the set inversion problem consists in determining the set $\mathbb{X}$ such that $\mathbf{f}(\mathbb{X}) = \mathbb{Y}$, where $\mathbb{Y} \subset \mathbb{R}^m$. The inverse evaluation problem is the characterized by:

$$\mathbb{X} = \mathbf{f}^{-1}(\mathbb{X}) = \{\mathbf{x} \in \mathbb{R}^n | \exists \mathbf{y} \in \mathbb{Y}, \mathbf{f}(\mathbf{x}) = \mathbf{y}\} \tag{3.57}$$

Given a arbitrarily large superset $\mathbb{X}_0$ where the solution set is known to belong an outer approximation $\overline{\mathbb{X}}$ can be computed using the Set Inversion Via Interval Analysis (SIVIA) algorithm [84] . Feasible boxes are added to the inner subpaving $\underline{\mathbb{X}}$ of solutions, a box $[\mathbf{x}]$ is feasible if $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$. Unfeasible boxes are discarded, since they contain no solution, a box $[\mathbf{x}]$ is unfeasible if $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$. Indeterminate boxes are bisected into two subboxes waiting to be examined, a box $[\mathbf{x}]$ is indeterminate if $[\mathbf{f}]([\mathbf{x}])$ intersects but is not included in $\mathbb{Y}$. Algorithm termination is ensured by adding indeterminate boxes whose width is less than ε to the subpaving $\Delta\mathbb{X}$ of indeterminate boxes. Since we are seeking to characterize the positioning confidence domain, we only need to compute the outer subpaving $\overline{\mathbb{X}}$ of the set that fulfills positioning constraints. Thus the outer subpaving is $\overline{\mathbb{X}} = \underline{\mathbb{X}} + \Delta\mathbb{X}$ (see Figure 3.5).

3.3.2 Measurement error bounds

In bounded-error models the integrity risk of the solution set not to include the ground truth is taken when the measurement bounds are chosen [72][73]. Let r be the risk associated with each pseudorange measurement:

$$r = Pr(\rho_r^s \notin [\rho_r^s]), i \in 1 \cdots m \tag{3.58}$$

The probability of having exactly i good pseudoranges out of m is given by a binomial distribution:

$$Pr(N_{ok} = i) = \frac{m!}{i!(m-i)!} \cdot (1-r)^i \cdot (r)^{m-i} \tag{3.59}$$

where N_{ok} is the number of pseudorange intervals that are consistent with the truth, and q is the number of the faulty measurements tolerated. Using (3.59) the probability of having at least $m - q$ good measurements is obtained:

$$Pr(N_{ok} \geq m-q) = \sum_{i=m-q}^{m} \frac{m!}{i!(m-i)!} \cdot (1-r)^i \cdot (r)^{m-i} \tag{3.60}$$

If there are no spurious pseudoranges $q = 0$ the SIVIA algorithm can compute an outer approximation $\overline{\mathbb{X}}$ of the solution set $\mathbb{X}$ which is guaranteed to be consistent with the true position x. The maximum risk r that can be assumed on each pseudorange interval, for a global risk $\mathbf{R}$, is given by:

$$\begin{array}{ccc} Pr(x \in \overline{\mathbb{X}}) & \geq & Pr(N_{ok} \geq m-q) \\ Pr(x \notin \overline{\mathbb{X}}) & \leq & 1 - Pr(N_{ok} \geq m-q) \\ \mathbf{R} & \leq & 1 - \sum_{i=m-q}^{m} \frac{m!}{i!(m-i)!} \cdot (1-r)^{i} \cdot (r)^{m-i} \end{array} \tag{3.61}$$

Once the maximum risk r of each measurement interval, to not contain the actual value is computed, the measurement error bounds can be set to meet this requirement. A centered Gaussian distribution with a variance σ^2, is used as an error measurement model for GNSS positioning, to set the error bounds on each pseudorange measurement:

$$\begin{cases} [\rho_r^s] = [\rho_r^s - \alpha\sigma, \rho_r^s + \alpha\sigma] \\ \alpha = -\Phi^{-1}(\frac{r}{2}) \end{cases} \tag{3.62}$$

where Φ is the cumulative distribution function of the standard normal distribution. Using this method leads to the same amount of risk taken on each tail of the Gaussian distribution.

Chapter 4

Simulator Framework

Contents

4.1 Simulator architecture

By definition, a Multi-Agent System (MAS) is composed by a group of individual agents that can interact with surrounding environment and between them [85]. The traffic motion inside of a city can be defined as a multiagent system because the vehicles are individual entities that interact with surround environment, and the traffic flow is a result of a large group of interactions and behaviors based on driver's perception, traffic laws and signaling systems. This approach to the problem brings a new paradigm to conceptualize, design and implement a traffic simulator, in this particular case, the MAS approach was the adopted solution. To achieve a complete simulator with the level of detail required, aspects like drivers field of view, drivers behavior, different type of vehicles and others must be considered to reach more accurate realistic simulator.

The ISR-TRAFSIM is composed by a group of modules, were each module is responsible for a self contained specific task. This modular type of architecture enables the inclusion of future functionalities in the simulator. The ISR-TRAFSIM was developed with broad set of modules, several complex systems are simulated, including GNSS and MSS. The V2V and V2I was also simulated to obtain a more realistic simulation. The simulator's traffic flow has two operation modes, the first mode operates without ITM, where drivers, respecting traffic laws and interacting with surrounding vehicles, follow their own will,

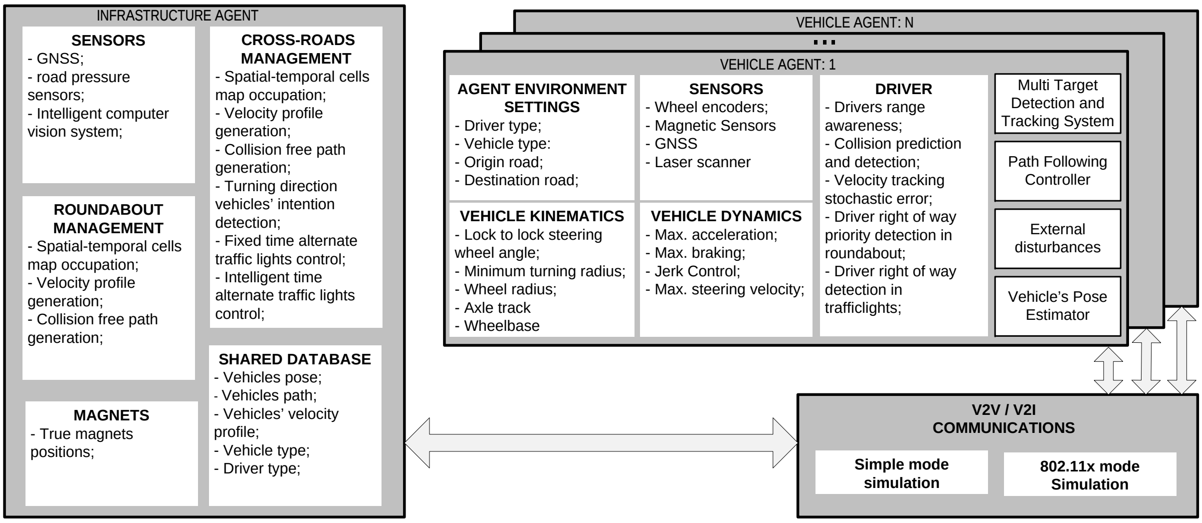

INFRASTRUCTURE AGENT
SENSORS
- GNSS;
- road pressure sensors;
- Intelligent computer vision system;
CROSS-ROADS MANAGEMENT
- Spatial-temporal cells map occupation;
- Velocity profile generation;
- Collision free path generation;
- Turning direction vehicles' intention detection;
- Fixed time alternate traffic lights control;
- Intelligent time alternate traffic lights control;
ROUNDABOUT MANAGEMENT
- Spatial-temporal cells map occupation;
- Velocity profile generation;
- Collision free path generation;
MAGNETS
- True magnets positions;
SHARED DATABASE
- Vehicles pose;
- Vehicles path;
- Vehicles' velocity profile;
- Vehicle type;
- Driver type;
VEHICLE AGENT: N
...
VEHICLE AGENT: 1
AGENT ENVIRONMENT SETTINGS
- Driver type;
- Vehicle type:
- Origin road;
- Destination road;
SENSORS
- Wheel encoders;
- Magnetic Sensors
- GNSS
- Laser scanner
DRIVER
- Drivers range awareness;
- Collision prediction and detection;
- Velocity tracking stochastic error;
- Driver right of way priority detection in roundabout;
- Driver right of way detection in trafficlights;
VEHICLE KINEMATICS
- Lock to lock steering wheel angle;
- Minimum turning radius;
- Wheel radius;
- Axle track
- Wheelbase
VEHICLE DYNAMICS
- Max. acceleration;
- Max. braking;
- Jerk Control;
- Max. steering velocity;
Multi Target Detection and Tracking System
Path Following Controller
External disturbances
Vehicle's Pose Estimator
V2V / V2I COMMUNICATIONS
Simple mode simulation
802.11x mode Simulation

Figure 4.1: Simulator architecture. Composed by three main modules: vehicle agents, infrastructure agent and V2V/V2I communications module.

and the second mode where each (RI and CI) individual intersection ITM or both in a cooperative manner, generate speed profiles along predefined paths which are to be followed by the vehicles.

The architecture of the ISR-TRAFSIM is shown in Figure 4.1. The ISR-TRAFSIM is composed by three main subsystems, designated by Infrastucture Agent, V2I/V2V Communications and Vehicle Agents.

Infrastructure Agent

The *Infrastructure Agent* is responsible for the selection of which algorithm is managing the traffic at each intersection. If an ITM algorithm is chosen, the speed profile and navigation directions of each vehicle are computed (see section 5.1.1) and sent by the *Infrastructure Agent* to the *Vehicle Agents*. The vehicles' pose, obtained using a Vehicle Pose Estimator (VPE) module (see chapter 6), are updated on a spatio-temporal matrix, according to the routes and speeds profile computed by the selected management algorithm.

Accurate driving cycles with acceleration and deceleration rates are important input values for the estimation of emissions and fuel consumption, as well as for traffic management, therefore these are sent to *Infrastructure Agent* through the communications module. Each intersection runs its own traffic management algorithm independently, and therefore the performance can be evaluated separately.

Vehicle Agents

The *Vehicle Agents* use the infrastructure road network and origin-destination road to determine its route. The vehicle real speed and acceleration are used by the *Environment*

Impact module in every simulation cycle for the computation of the fuel consumption and CO_2 emissions. The emissions are computed using the thermal engine model described in section 5.2. Accurate driving cycles with acceleration and deceleration rates, cruising rates are important input values to estimate emissions and fuel consumption and for the traffic management, therefore these are sent to Infrastucture Agent trough the simulated wireless V2I CS. The vehicles' pose is estimated using a VPE module (described in chapter 6), *i.e.* by fusing the information gathered by several simulated sources, namely GNSS and odometry . The simulated GNSS pseudoranges are affected by simulated sources of noise, namely: ionospheric delay, tropospheric delay, multipath error and thermal noise. Vehicles crossing the simulated intersections may be:

- autonomous vehicles;
- human driven vehicles;
- vehicles equipped V2X communications;
- vehicles not equipped or with faulty V2X communications.

When a new vehicle's agent is inserted into the simulation environment, several general characteristics defining the vehicle/driver are selected according to a predefined distribution.

Agent environment settings: The vehicle may differ in its dimensions whether it is defined as a *car* or a *truck*. The type of vehicle driver will enable a minor or greater acceleration variance whether it is defined has a calm or aggressive driver.

Vehicle kinematics: By default, each vehicle movement is simulated using the Ackerman steering kinematic model (2WS), its pose is updated every sampling time. In order to increase the stability and response of the vehicles at high speeds, automotive manufacturers are increasingly integrating 4WS technology in their vehicles. Due to this tendency the vehicle agent typology can be 2WS or 4WS. The kinematics variables, such as lock to lock steering wheel angle, minimum turning radius, wheel radius, axle track and wheelbase, are fully customized. They were adjusted for a typical hatchback European car and for a European light truck or minivan.

Vehicle dynamics: The vehicle dynamics constraints define the maximum acceleration, maximum deceleration and maximum steering velocity for each type of vehicle and driver. An additional constraint on the jerk variation was inserted enabling a comfortable vehicle ride (see section A). The vehicle's speed is derived from the vehicle environment and from traffic laws. When on straight lanes, if the sensors detects a car in front, the reference speed will be taken from the Intelligent Driver Model (IDM) car-following model [86], were the acceleration of follower vehicle, dv/dt, is given by:

$$\frac{dv}{dt} = a\left[1 - \left(\tfrac{v}{v_p}\right) - \left(\tfrac{s^*}{s}\right)^2\right] \tag{4.1}$$

where

$$s^* = s_0 + \left(v.T + \frac{v.\Delta v}{2.\sqrt{a.b}}\right) \tag{4.2}$$

This model specifies the acceleration of a vehicle, function of a breaking term given by the relation of the current speed, v and the maximum speed of road v_p and for the relation between the current gap s and the preferred distance, s^*. The headway time is $T = 1s$, the acceleration capacity for vehicles, a, is set to half of breaking capacity, $b = 9.81m/s^2$, and the distance limit is set to $s_0 = 2m$.

When not in straight lane, a different methodology is used to generate the speed command. For each simulated vehicle a collision time is computed to all nearby vehicles in the field of view using the Multi-Target Detection and Tracking (MTDT) module (see section A). The vision field can be circular, ellipsoidal or rectangular. The current velocity of vehicles is used to compute the estimated arrival time at a potential collision point. If there is a chance of collision, traffic rules are applied and one of the vehicles must break.

Simulated sensors: Several type of sensors are simulated, the information provided by each sensor is updated according to its own predefined sampling rate. Simulated sensors include:

encoders: four wheel encoders, front and rear axle steering encoder;

magnetic sensing system (MSS): composed by a set of magnetic markers detectors [87]. The simulated sensors are triggered when they cross over a magnetic landmark.

GNSS: the GPS constellation is simulated as well as the tropospheric delay, ionospheric delay, multipath noise, thermal noise and clock biases. The GPS simulated receiver provides both the code and carrier pseudoranges for all visible satellites

Laser Range Finder (LRF): this sensor, also referred as LIDAR, provides all range-bearing data to the MTDT and the average range-bearing of a preceding detected vehicle;

inductive loop: road drive-through inductive loop sensors in each outgoing lane capable of detecting the vehicles leaving the intersection

intelligent vision system: this system detect the approaching vehicles and check the status of the turning light signals of each vehicle

Noise and other disturbances may be applied to the sensors' readings.

Driver awareness: The driver's perception of the surrounding environment is updated and the decision of accelerate or brake is taken according the traffic rules and the neighboring vehicles inferred intention behavior. When in a human drive mode, each driver is only aware of surrounding vehicles if they are inside a "drivers range awareness", this range can be configured to a square area, circle area or a ellipsoidal area. The ellipsoidal area resembles more accurately the human awareness behavior but it is less computing efficient. Using the perceived velocity of the vehicle inside the awareness range, a collision prediction and detection is performed. If a future collision ahead is detected then the driver takes an action, braking while outside intersections or giving right of way if entering intersections. One must notice that while on human driver mode the information used by the agent is similar to the one a human driver in real life would have.

V2V/V2I Communications

ITM algorithms require a secure channel for information exchange between vehicles and infrastructure. The current CS technology, namely wireless, already provides reliability in communications. In the ISR-TRAFSIM a communication system is simulated with two operation modes: simulated-Fast and simulated-Real.

In the simulated-Fast mode the CS delivers messages instantly and the only constraint taken into account is to verify whether each agent is within, or not, the range of other agents' antennas.

Based on the work of Saikat Ray [88] we have developed a two-dimensional wireless network discrete event simulator, to approximate the simulated CS to a real CS system.

The *V2V and V2I communications module* provide a secure channel for information exchange among vehicles and infrastructure. Both intersections types have V2I modules, enabling each ITM to communicate with the vehicles on the communication range. Communication between RI and CI is also available, enabling information handshaking of a vehicle crossing both zones.

This operation mode has a variety of configurable parameters, the most important are:

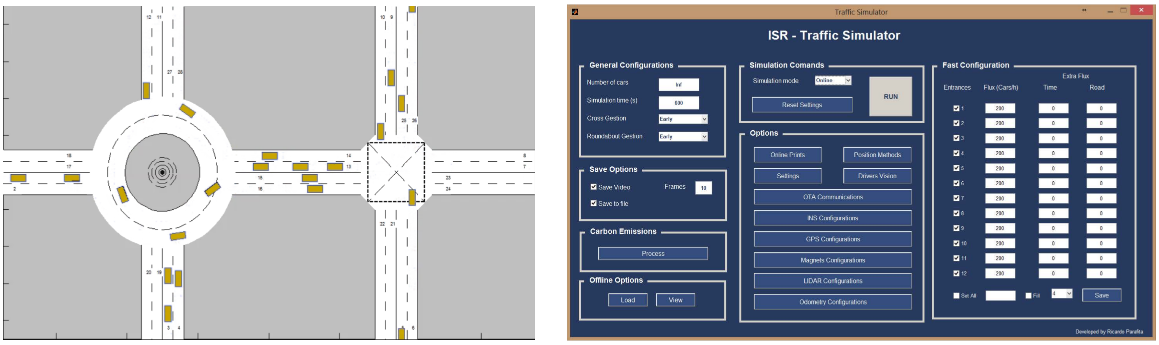
i Standards: IEEE 802.11 b\g\a;
ii Propagation model: Friis, two-ray ground and log shadow:
iii Maximum number of RREQ retries.

We assume that all transmissions experience the same path loss versus distance profile, that every node transmits with the same power in the same channel, that the propagation delay is negligible and that each node has the same antenna gain and receiver sensitivity. In this simulated-Real mode it may happen that two communicating nodes can communicate with a third station, but cannot directly communicate with each other due to physical or spatial limitations, this is the so-called hidden node problem, meaning that wireless nodes often cannot hear each other. Since hidden nodes cause packet collisions, their presence can severely affect the performance of the V2X communications . CS is implemented using four layers: application layer (APP), network layer (NET), medium access control layer (MAC) and physical layer (PHY). A receiver decodes a packet if and only if the packet does not overlap with any other packet transmitted by a node within its range. Every agent senses a busy channel immediately after the transmission begins, if a collision is detected the agent's CS will backoff for a random amount of time. A successful communication between a node 1 and a node 2, involves a packet going trough the following layers $APP1 \Rightarrow NET1 \Rightarrow MAC1 \Rightarrow PHY1 \Rightarrow PHY2 \Rightarrow MAC2 \Rightarrow NET2 \Rightarrow APP2$. These simulation settings enable the user to quantify the impact of packet delay, masked nodes, channel fading, and mobility on packet collisions on a large IEEE 802.11 ad-hoc wireless communications.

4.2 Simulator setups

Simulator environment

The ISR-TRAFSIM road network layout includes two common solutions to regulate intersections traffic, roundabout and crossroads as depicted in Figure 4.2 (a). The simulator has a friendly GUI, where users can parameterize most relevant settings without having programming skills, and making it possible to change settings and perform simulations with different settings (see Figure 4.2 (b)).



General Configurations

Number of cars: Inf
Simulation time (s): 600
Cross Gestion: Early
Roundabout Gestion: Early

Save Options

☑ Save Video Frames: 10
☑ Save to file

Carbon Emissions

Process

Offline Options

Load View

Simulation Comands

Simulation mode: Online
Reset Settings
RUN

Options

Online Prints
Position Methods
Settings
Drivers Vision
OTA Communications
INS Configurations
GPS Configurations
Magnets Configurations
LIDAR Configurations
Odometry Configurations

Fast Configuration

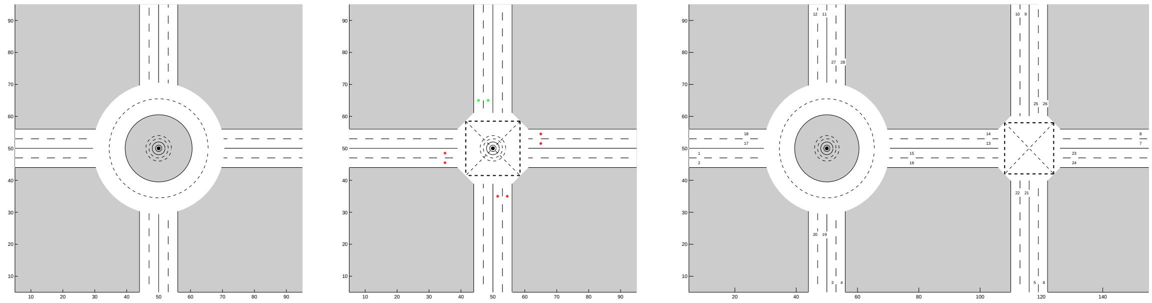
		Extra Flux	
Entrances	Flux (Cars/h)	Time	Road
☑ 1	200	0	0
☑ 2	200	0	0
☑ 3	200	0	0
☑ 4	200	0	0
☑ 5	200	0	0
☑ 6	200	0	0
☑ 7	200	0	0
☑ 8	200	0	0
☑ 9	200	0	0
☑ 10	200	0	0
☑ 11	200	0	0
☑ 12	200	0	0

□ Set All □ Fill 4 Save

(a) (b)

Figure 4.2: Simulator main interfaces: (a) scenario interface populated by car like vehicles, (b) configuration interface.

The traffic is right-handed, *i.e.* vehicles on the RI travel anticlockwise, and no overtaking maneuvers are considered in this study. The RI traffic inflow/outflow arrives from four directions. Each direction is composed of two roads of inflow/outflow and each road has two lanes; the same setup is also applied to the CI. Both RI and CI can be controlled by traffic lights. Inside the CI box junction, vehicles are not allowed to stop, *i.e.* a vehicle must not enter the box junction if there is no space for the entire vehicle to exit in the other side of the CI. No overtaking maneuvers are allowed and the speed limit in entire scenario is set to 40Km/h, except while driving in the RI and CI, where the speed limit is 30 Km/h. The lanes width is 3m on the straight roads and 5m on the RI lanes.



(a) (b) (c)

Figure 4.3: Simulator scenarios: (a) roundabout, (b) crossroads, (c) roundabout and crossroads interconnected

Three road networks environment scenarios can be tested independently, a single RI a single CI and a scenario with both a RI and CI interconnected (see Figure 4.3)

Data management

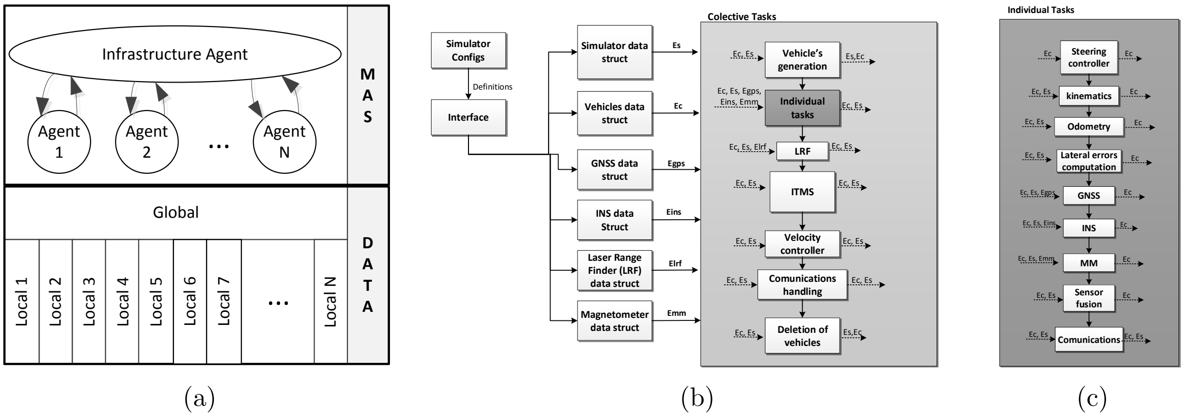
Data is divided according to its type: global simulator and local simulator data. The global data describe the scenario conditions such as antenna localizations, semaphores switching times, simulation options, radio propagation models, etc. Local data is stored in a linked list where each node corresponds to a specific set of driver and its vehicle. Each agent stores its data, that can be read and modified during simulation. Due to the high number of vehicles which may be present in the simulator environment, local simulator data is not allocated for all vehicles at once, *i.e.* the local simulator data is managed in order to add an entity node when a vehicle enters the simulation scenario or to delete an entity each time a vehicle leaves the simulated environment (see Figure 4.4).

The methodology used for the storage of data makes use of a small number of structures (Ec, Es, EGPs, Eins, Emm, Elrf) which store parameters for the simulator settings and each parameter available for vehicles. The structure storing the data for each vehicle (Ec) is formed by sub-structures, where each sub-structure stores data from a single vehicle, forming a single linked list available for read and write (see Figure 4.4(b-c)).

This data management enabled the development of analysis tools to publish graphics and performance values automatically, allowing a quick analysis of the results.

Traffic flow

The simulation starts with an empty vehicle scenario. For each lane, the vehicles agents are created according to the defined *Traffic Flow Profile*. In general, rush hours are the



(a) (b) (c)

Figure 4.4: Simulator data flow: (a) multi-agent system and data storage design, (b) overall agents data flow, (c) individual agents specific data flow.

Table 4.1: Vehicle origin-destination probability distribution.

Destination Origin	R_1	...	R_2	...	R_m	
R_1	P_{11}	...	P_{1j}	...	P_{1m}	$\sum_{k=1}^{m}(P_{1k}) = 1$
...	...	...	...	...	...	...
R_i	P_{i1}	...	P_{ij}	...	P_{im}	$\sum_{k=1}^{m}(P_{ik}) = 1$
...	...	...	...	...	...	...
R_n	P_{n1}	...	P_{nj}	...	P_{nm}	$\sum_{k=1}^{m}(P_{nk}) = 1$
	$\sum_{l=1}^{n}(P_{l1}) \leq n$	...	$\sum_{l=1}^{n}(P_{lj}) \leq n$	...	$\sum_{l=1}^{n}(P_{lm}) \leq n$	

most critical time to intersection, because of the high influx of vehicles in a short period of time, but during the rest of the day a road intersection may have low influx, therefore it experiments different levels of traffic density. Traffic flow (or traffic volume) and traffic density (or traffic concentration) represent different concepts. The traffic flow is described by the following equation [89]:

$$Flow = \frac{n_{F,vehicles}}{T} \qquad [Vehicles/hour] \tag{4.3}$$

where $n_{F,vehicles}$ represent the number of vehicles crossing a section of road in a determined amount of time, T. Traffic density is defined as:

$$Density = \frac{n_{D,vehicles}}{X} \qquad [Vehicles/meter] \tag{4.4}$$

where $n_{D,vehicles}$ represent the number of vehicles that are present in a section of road of length, X. To adjust the traffic density in the simulated area, the flow of each road is configurable, this adds more versatility to the simulator, because any time of day can be simulated in the developed scenario. Each vehicle origin-destination lane setup, follows a

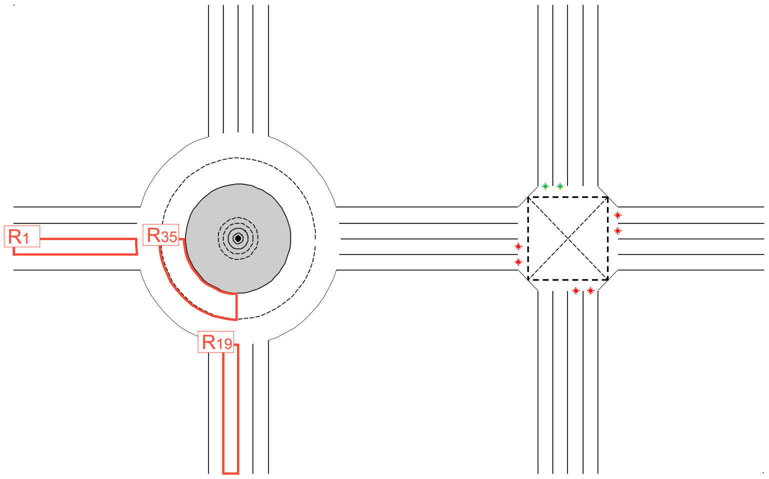


Figure 4.5: Road segments selection for a vehicle entering on lane 1 and exiting on lane 19.

Out In	…	R_{19}	…
⋮	⋮	⋮	⋮
R_1	…	R_{35}	…
⋮	⋮	⋮	⋮
R_{35}	…	R_{19}	…
⋮	⋮	⋮	⋮

Table 4.2: Routing table for a vehicle entering on lane 1 and exiting on lane 19.

semi-random model, as presented in Table 4.1, where P_{ij} represents the probability of a vehicle from an origin lane i to have its destination on lane j. It is possible for a road to have to have more outflux than influx, but its outflux probability must be lower than n, *e.g.* assuming the influx probability distribution from all n lanes to the outflux lane j to be 1, then $\sum_{l=1}^{n}(P_{lj}) = n$.

Path generation

It is necessary to generate the entire trajectory to be followed by a vehicle agent every time a vehicle agent is created. Based on the origin and the desired destination, a path is computed by a trajectory planner, this path consists on a list of path coordinates points and its associated curvature, obtained using cubic spline sampling.

The simulator is designed to allow the construction of diverse scenarios, therefore no predefined path is computed in advance. Hence each road segment is defined by its length; starting and end point; orientation and to which road segments it is connected. From the vehicle origin and the desired destination, the road segments necessary to traverse in order to complete the intended journey, are selected using a routing table. In Figure 4.5 and Table 4.2 it is presented the road segment selection and a routing table for a vehicle entering on lane 1 and exiting on lane 19. The first column represent the road entrance segments and the first line represent the desired exiting road segment.

For a vehicle that enters in segment R_1 and wants to go to segment R_{19}, the first path segment is therefore R_1, followed by the intersection value of row R_1 and column R_{19}, i.e. R_{35}. The next segment is selected at the intersection of the row of this newly selected segment R_{35} and column R_{19}, i.e. R_{19}. The road segment selection ends when the next segment is the exiting one, as in this last selection. Therefore the full road segment selection is $R_1 \rightarrow R_{35} \rightarrow R_{19}$.

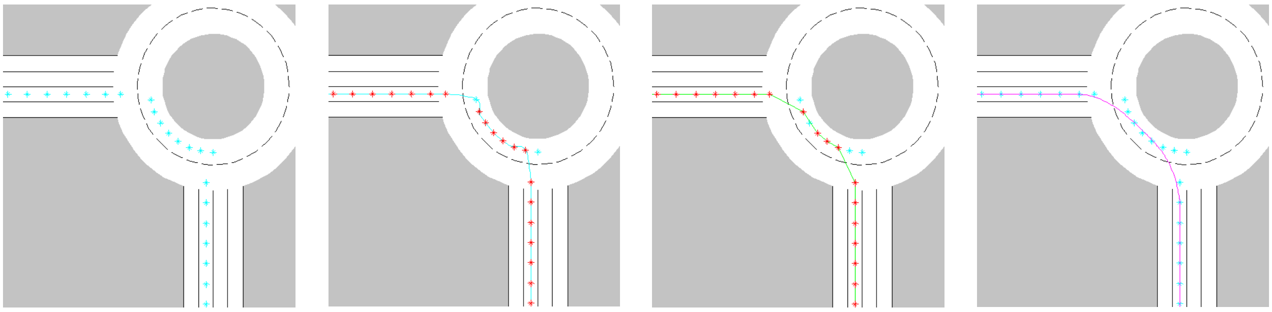


Figure 4.6: Smooth path generation procedure: 1^{st} insert reference points from the selected road segments; 2^{rd} spline using reference points; 3^{rd} remove reference points with high curvature nearby; 4^{th} spline using filtered points; 5^{th} iterate 3^{rd} and 4^{th} steps until the path has a a smooth and feasible curvature for the type of vehicle.

The procedure to generate path coordinates points and its associated curvature, using the list of road segments, starts by generating reference points of the all road segments. Each road segment is used to generate a set of reference points with all intermediary points are concatenated to generate an unrefined trajectory. The last stage of path generation is to smooth the trajectory generated by removing points with low curvature radius. The path generation stages is depited Figure 4.6.

4.3 Path Following Controller

The vehicle when moving from one starting point to an end point can execute a point-to-point stabilization, a path following or trajectory tracking. Point-to-point stabilization requires that the vehicle moves from point A to point B with no restrictions on its movement between these two points. When with path following, the vehicle must move along a geometric path. For trajectory tracking, the vehicle must move along a geometric path at a given speed. The controller here presented is of the path following type.

In order to follow the trajectory, the actual vehicle's pose, the next desired position and the reference velocity are feeded as inputs to the Lateral Controller (LC). The errors between the desired pose and the actual pose (see Figure 4.7) are used as inputs to the LC module (see Figure 4.8) that converts them to desired commands. The time-to-collision computed in the MTDT module (see section A) is also provided to the LC.

Collision avoidance is achieved by controlling the vehicle's reference velocity by reducing or even stopping the vehicle in situations of eminent danger. The main goal of the path-following controller is to ensure that the vehicle follows the predefined reference path with appropriate orientation.

This section addresses the path following problem using fuzzy-logic.

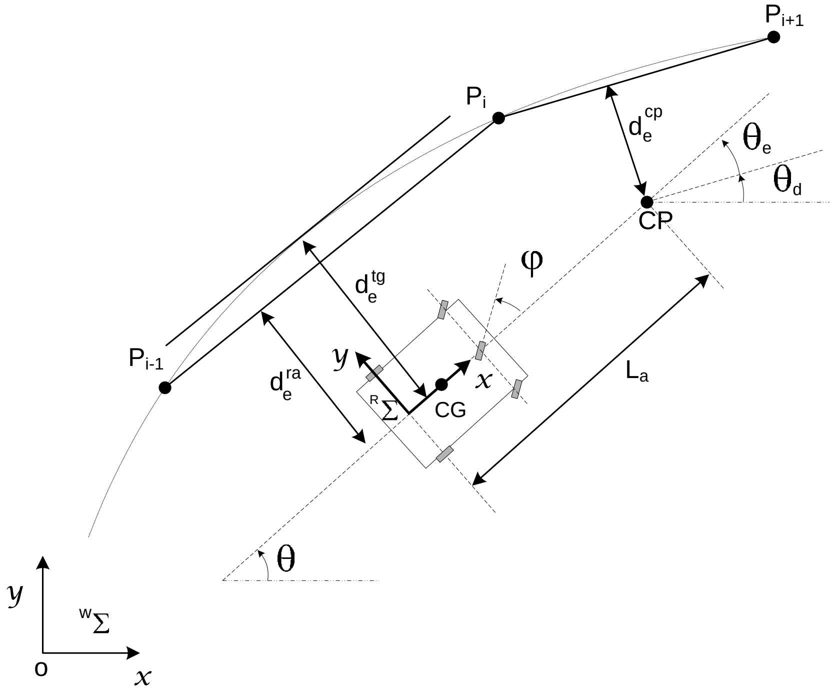


Figure 4.7: Illustrative construction of lateral and heading errors (in real situation the discrete path points P_i are much closer between each other, when compared with the vehicle dimensions). P_i denote the points that define the reference trajectory; d_e^{cp} and d_e^{ra} are the lateral errors at CP and rear axle, respectively; d_e^{tg} is the perpendicular distance between the rear axle midpoint and the current tangent to the path; θ, θ_e^{cp} and θ_d are the orientation of the vehicle, heading error and desired heading, respectively. $^W\Sigma$ and $^R\Sigma$ represent respectively the world coordinate system and the vehicle local coordinate system with its origin at the midpoint of the rear axle and its x-axis aligned with the longitudinal axis of the vehicle

Fuzzy-Logic path following Controller (FLC) for a 4WS Kinematic Vehicle

It was chosen to develop a fuzzy-logic based controller, so that non-skilled programmers could tweak or test new controlling setups. The fuzzy-logic based PFC was firstly evaluated on a simple kinematic model (*i.e.* a differential drive vehicle), and further development is built upon the first iteration on a rather complex kinematic model, (*i.e.* a single and double steered car-like vehicle). A detailed description can be found in Appendix A.

The fuzzy-logic based PFC is made up of two main modules: Velocity Planner (VP) and LC. The LC as the following set of inputs:

$$u_{FL} = [\theta_e^{cp}, d_e^{cp}, \Delta\theta_e^{cp}, \Delta d_e^{cp}, t_{imp}, d_{ile}, c(s), v] \tag{4.5}$$

where v denotes the linear reference velocity, $c(s)$ is the curvature along the path, d_{ile} is the inlinelateral error (see description of the *Rear Steering Switch module*), t_{imp} is the time-to-collision computed in the MTDT and the differential errors Δd_e^{cp} and $\Delta\theta_e^{cp}$, at the

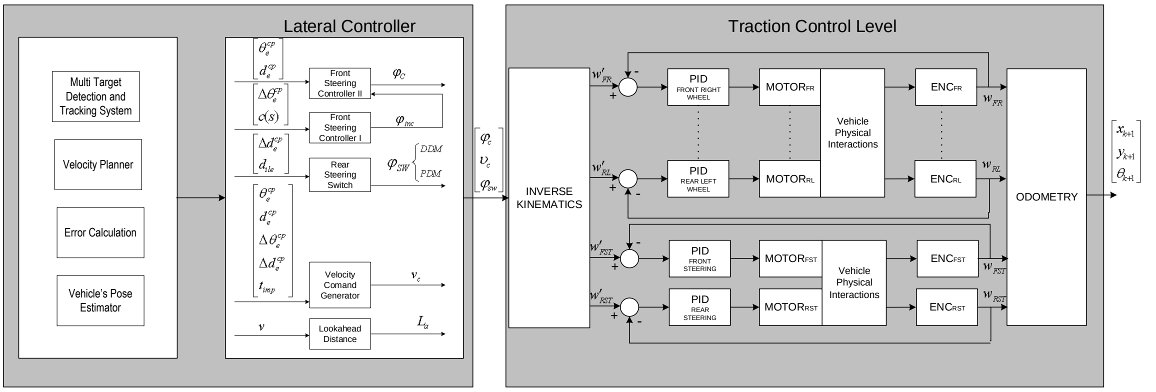


Figure 4.8: Controller architecture: FLC modules of the LC and low lever controller for a 4WS with 4WD

Control Point (CP), are given by

$$\Delta d_e^{cp} = d_e^{cp}(k) - d_e^{cp}(k-1) \tag{4.6}$$

and

$$\Delta \theta_e^{cp} = \theta_e^{cp}(k) - \theta_e^{cp}(k-1) \tag{4.7}$$

The main goal of the path-following controller is to ensure that the vehicle follows the predefined reference path with appropriate orientation. For the fuzzy-logic controller this can be understood as a task of minimizing the vehicle lateral and heading errors $(d_e^{cp}, \theta_e^{cp})$ with respect to the reference path, at a given CP located at a distance L_a denoted by lookahead distance, as illustrated in Figure 4.7.

Figure 4.8 presents the controller architecture. The architecture is divided in high and low level controller. The high level controller LC computes the control commands for the low level controller, some of the high level modules are also presented has they provide inputs to the LC. The LC is based on fuzzy-logic, it is composed by four independent modules: front steering controller, rear steering switch, velocity command generator and lookahead distance computation (see Figure 4.8). The low level controller is responsible for the traction control and besides regular vehicles it also enables the control of electrical Four Wheel Drive (4WD) and 4WS vehicles.

The controller provide a control vector ($[\varphi_c, v_c, \varphi_{sw}]$) to the traction control level,where φ_c (in degrees) is the steering angle, $v_c[ms^{-1}]$ is the velocity command, and φ_{sw} is the rear steering switch that controls the two possible driving modes: Dual Driving Mode (DDM) and Park Driving Mode (PDM). In DDM the rear axle steers in opposite direction of the front axle, while in PDM the rear and front axle steers in the same direction, both modes are 4WS. If the vehicle is 2WS then the φ_{sw} as no effect on the low level traction controller.

In order to properly avoid collisions with obstacles the time-to-impact t_{imp} (also referred here as time-to-collision), provided by the MTDT, is integrated in the velocity command generator. All modules are fuzzy-logic based. Figure 4.8 shows the LC identifying the fuzzy-logic inference flow from the input variables to the output variables. The fuzzy controller is characterized in Appendix A.

Front steering module: The front steering module (composed by controller 1 and 2) computes the steering command φ_c. The purpose is to minimize both the orientation error θ_e^{cp}, and the lateral error d_e^{cp}. A steering increment fuzzy variable (φ_{inc}) is computed in order to achieve a faster recovery from an undesirable pose. φ_{inc} is the output of a fuzzy module which has as inputs $c(s)$ and $\Delta\theta_e^{cp}$. This module ensures a geometrical convergence towards the path to be followed.

Rear steering switch module: The Rear Steering Switch module decides whether the rear axle should steer to opposite direction of the front axle DDM or in the same direction PDM. The inputs of this module are Δd_e^{cp}, and the inline lateral error d_{ile} given by

$$d_{ile} = |\frac{d_e^{cp}}{d_e^{cg}}| + |\theta_e^{cp}| \tag{4.8}$$

If Δd_e^{cp} is decreasing and d_{ile} is small, this module steers the rear wheels in the same direction as the front wheels; the result is a decreasing of the vehicle's yaw motion. The yaw motion is necessary for executing a manoeuvre but is not desired from the point of view of the vehicle's stability control [90]. This module was only implemented and tested in simulations.

Velocity command generator module: The inputs of this module are Δd_e^{cp}, d_e^{cp}, $\Delta\theta_e^{cp}, \theta_e^{cp}$ and the t_{imp}. This module computes a weight factor assigning a level of significance to the reference velocity, i.e. if the errors have a high magnitude or the time-to-collision has a low magnitude then the velocity must be decreased, otherwise the reference velocity is applied. This module is of extreme importance since collision avoidance is decided here, i.e. if the t_{imp} is small, then the vehicle velocity is reduced or the vehicle is even stopped; if t_{imp} is high the vehicle velocity (weight factor) is not affected.

Lookahead distance computation module: This module computes the lookahead distance, L_a, which is a function of the vehicle velocity, v. If the velocity increases, the damping factor of the closed loop system gets worse and is improved by increasing the lookahead distance. The lookahead distance provides a prediction behaviour to the controller, since it enables the control point to be far ahead of the Center of Gravity (CG) of the vehicle, see Figure 4.7.

Velocity planner

The VP module calculates the linear reference velocity, as well as determines the local reference trajectory points. One main objective taken into account was to make the trip as comfortable as possible, i.e. to give the system the capability of fully controlling the smoothness of the acceleration profile either lateral or longitudinal.

A Canadian study [91] used a highway testing ground to test speed and lateral acceleration on both wet and dry pavement on horizontal curves. They found that *"comfortable lateral acceleration"* and *"speed environment"* limited the driver's speed, while pavement surface conditions (dry or wet) and the driver's gender did not. Drivers adjusted their comfortable speed according to their comfortable lateral acceleration tolerance, approximately between $0.35g$ and $0.40g$. Another study [92] revealed the comfortable longitudinal acceleration, i.e. steady deceleration under expected-stop conditions; drivers generally exert an average steady braking force of $-0.35g$. This amount of braking force seems comfortable for most drivers.

Multi-Target Detection and Tracking (MTDT)

The information regarding the dynamics of the surrounding obstacles, is pivotal to the estimation of the trajectories and behaviour of surrounding objects. The MTDT estimates the position and velocity of existing objects in the environment, thanks to a number of consecutive scans provided by a LIDAR, based on which the impact-time is computed.

The MTDT comprises the following modules: scan segmentation, object classification, multi-object tracking and time-to-collision computation.

Segmentation: The goal here is to identify the limits of possible existing objects detected by the LIDAR (see Figure A.9) and if so, to filter and provide additional information about the object in analysis.

Object Tracking: It identifies the segment-object pair by performing object tracking using a Kalman filter.

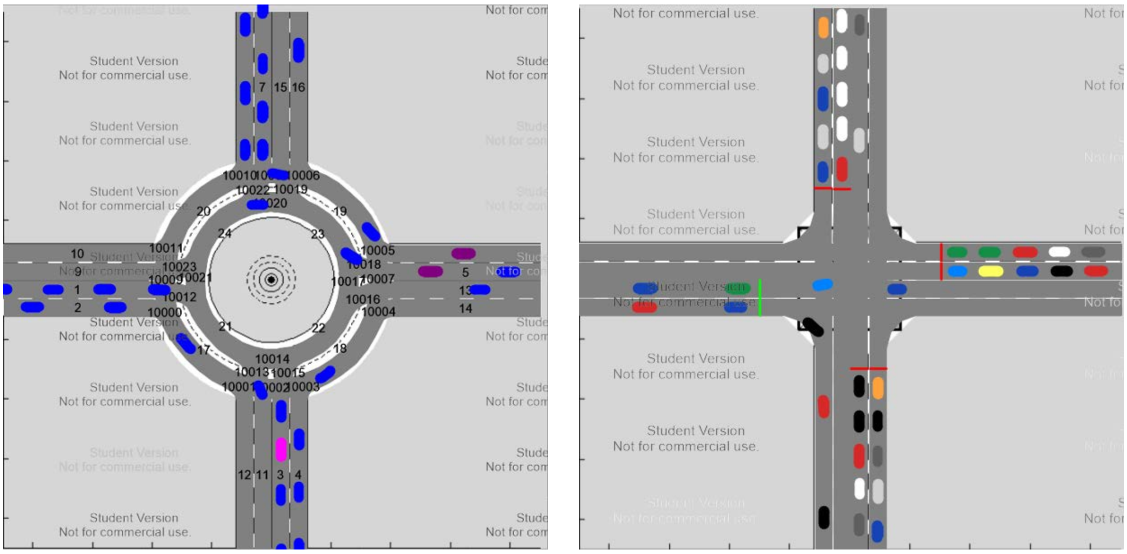
Impact-Time Computation: The Impact-Time Computation module uses the results of all the *Segmentation* and the *Object Tracking* to estimate the impact-time and position, for each one of all detected objects.

This system estimates the positions and velocities of existing objects in a selected area in the field of view of the LIDAR, based on which the times-to-collision are estimated.

4.4 Results

4.4.1 Traffic Management Simulator Assessment

This section presents a verification of ISR-TRAFSIM *w.r.t.* the traffic management. This verification will be performed using a validated Simulator, VISSIM [93].



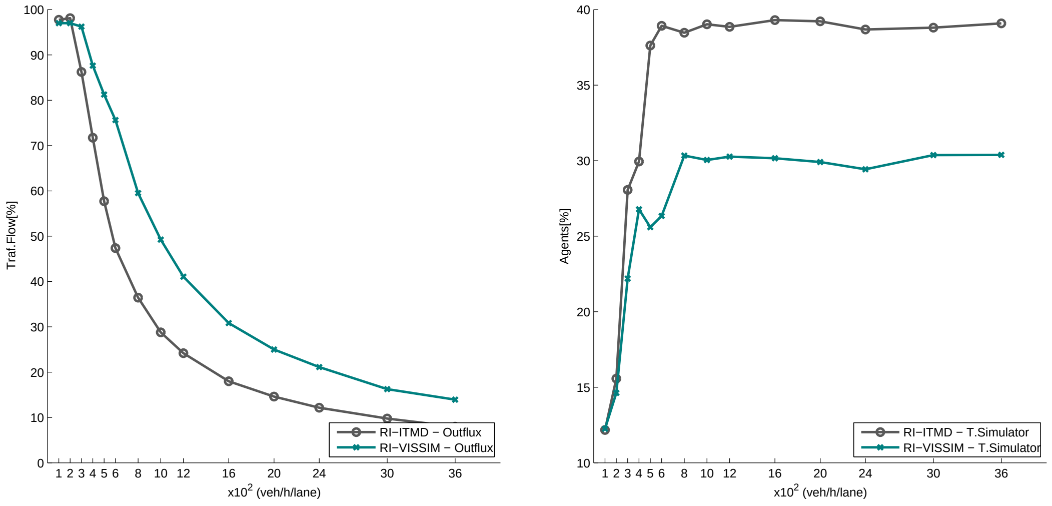
(a)

(b)

Figure 4.9: VISSIM test scenario: (a) roudabout, (b) crossroads.

In order to verificate ISR-TRAFSIM traffic management results, it was used traditional traffic management techniques, namely right of way inside roundabouts and fixed traffic light time switching for the crossroads. Each ISR-TRAFSIM intersection result was tested individually against VISSIM results. VISSIM simulator does not have an open architecture, and therefore there are some limitations when creating simulations environments, hence simulations comparisons with ISR-TRAFSIM may be inaccurate. Nevertheless the student version of VISSIM Simulator was used due to the fact that it is a state of the art simulator, it is already validated [94], it is freely available and it as a widespread use. In order to make the ISR-TRAFSIM results verification as accurate as possible, the intersection scenario, both the roundabout and the crossroads, was constructed with the same size measurements, (see Figure 4.9), and the parameters of both simulators was set as similar as possible. As stated earlier VISSIM simulator is not an open architecture, and therefore some of the inner workings are unknown to the user and some customization details are unavailable to the user. One of the limitation is the vehicles dimensions specification, *i.e.* it is not possible to define a specific value, since it is only allowed the definition of the vehicle type. For each vehicle type there is a preset of values for vehicle dimensions, which are chosen randomly. The definition of vehicles per lane influx and outflux can not be made on a per-vehicle basis. It is not possible to configure an acceleration profile for the vehicles, which is another limitation. Influx streams ranging from 100 vehicles per hour to 3600 vehicles per hour for each of the 8 input lanes, were simulated on ISR-TRAFSIM and VISSIM simulators for both RI and CI. Each influx test as a simulated time of 10 minutes. Test results are presented in Figures 4.10 and 4.11.

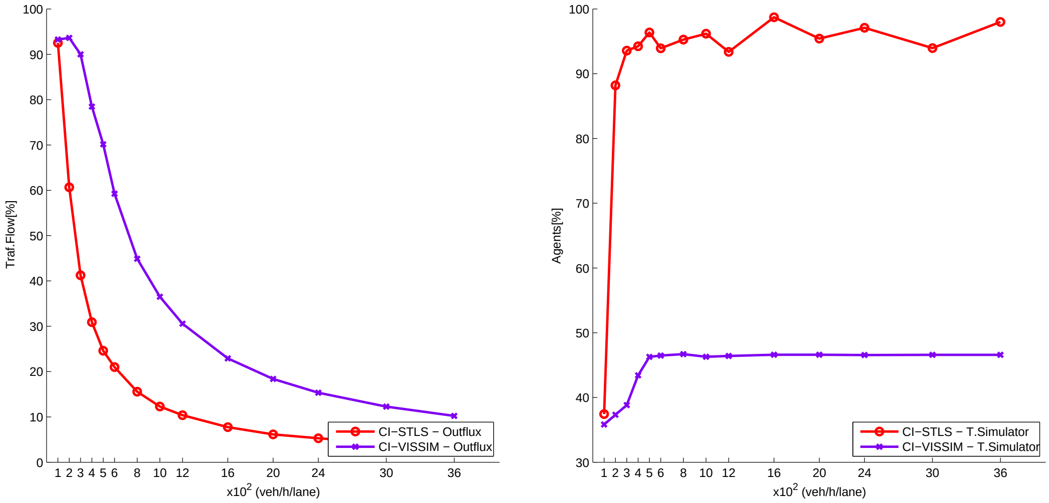
The ISR-TRAFSIM roundabout verification scenario, was set to run the Intelligent Traffic Management Deactivated (ITMD) algorithm (see section 5.1.2) and the VISSIM simulator was set accordingly. In this test vehicles entering the roundabout have to give right of way to vehicles already travelling inside the roundabout. Figure 4.10 presents the output flow and average time spent by vehicles traversing the intersection, for both simulators when using the roundabout scenario. The outflux average error is 12.49%, ranging between 0.75% for a 200 veh/h influx to 26.25% for a 500 veh/h influx, as the influx grows the error tends to zero. Similarly, average time a vehicle takes to traverse the intersection as an average error of 6.79 seconds, settling its value around 7.5 seconds after 800 veh/h influx. After a careful analysis of the previous data we can conclude that both simulators have similar results when simulating a roundabout scenario. These results suggest, that a intersection management performance improvement achieved using ITM algorithms, tested using ISR-TRAFSIM simulator on a roundabout scenario, is also expected an improvement in the real world scenarios.



(a) (b)

Figure 4.10: VISSIM test results for 8 lanes influx roundabout intersection: (a) Vehicle outflux function of influx, (b) Agents time traversing intersection function of influx.

The ISR-TRAFSIM crossroads verification scenario, was set to run the Standard Traffic Light System (STLS) algorithm (see section 5.1.2) and the VISSIM simulator was set to approximate these specifications. In this test vehicles can only enter the crossroad box while on green light. Figure 4.11 presents the output flow and average time spent by vehicles traversing the intersection, for both simulators when using the roundabout scenario.



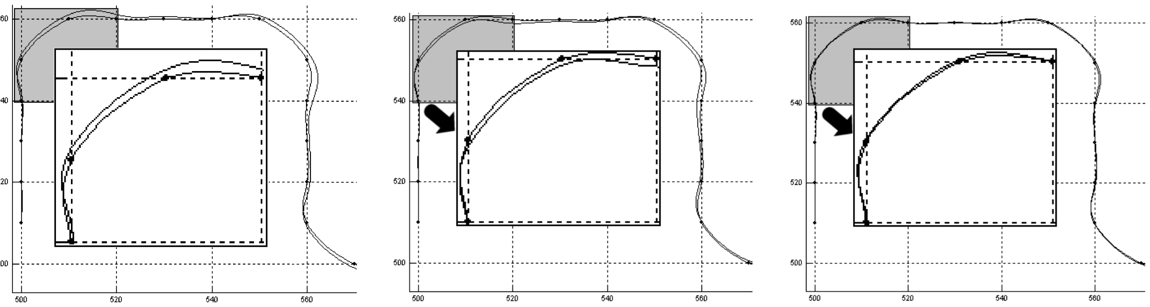
(a) (b)

Figure 4.11: VISSIM test results for 8 lanes influx crossroads intersection: (a) Vehicle outflux function of influx, (b) Average time traversing intersection function of influx.

The outflux average error is 24.28%, it is a high value but as the influx grows the error tends zero. The average time a vehicle takes to traverse the intersection as an average error of 46.53 seconds, alhough it is a high error it is stable error value. Part of the errors obtained can be attributed to the fact that the VISSIM simulator is not an open architecture, and therefore not allowing to define the acceleration profile, not being able to define precisely the vehicle dimensions as well as not being able to define a origin-destination routing table that could be fitted to the settings of ISR-TRAFSIM. It is not possible to set the parameters of ISR-TRAFSIM parameters to match the VISSIM as their models are not available. Contrary to the roundabout intersection the tests conducted for the crossroads intersection have fairly significant error in the comparison between the two simulators. Therefore, unlike the roundabout scenario, any performance improvement, of ITM algorithms applied to crossroads, when in ISR-TRAFSIM do not suggest that the same will succeed in real life.

4.4.2 Fuzzy-logic controller (FLC)

The results presented in this section were obtained using the "*double steered mode*", the effectiveness of the controller revealed to be similar for the "*front steered mode*", the main difference was that the in "*double steered*" the minimum curvature radius was inferior and therefore the "*reference path*" could had more narrow curves.



(a) (b) (c)

Figure 4.12: Test track result of the lateral controller: (a) control point at CG; (b) control point at a constant value lookahead distance $L_a > L/2$; (c) control point at a lookahead distance function of the vehicle velocity L_a. The dots above the solid line denote the "*reference path*" (x-axis and y-axis in meters).

The control point can be defined at the center of gravity of the vehicle, but it is also possible to define it in front of the vehicle, with help of a virtual point (section 4.4.2).

The control point is then defined at a lookahead distance L_a in front of the vehicle as defined in Figure 4.7. For $L_a = L/2$ the CP is located at the CG of the vehicle. The simulations here reported concerns with a bi-steerable vehicle without inducing any type of odometry errors.

Figure 4.12(a) shows a simulation result, where a response with a damping behaviour can be observed. Despite having a good performance, it has some drawbacks. When the vehicle approaches the curve it has a low lateral error d_e^{cp} and a high orientation error θ_e and when it leaves the curve it has a high lateral error d_e^{cp} and a low orientation error θ_e, which means a slower recovery from narrow curves.

Lateral controller with the CP at a lookahead distance L_a

The control point is chosen at a distance $L_a > L/2$ in front of the vehicle. L_a is a function of the vehicle velocity (section A), which must be kept within certain limits. If L_a is too small the vehicle might reach the target point between two computations, or oscillations might appear. If L_a is too high the vehicle might cut corners.

Figure 4.12(c) shows the behaviour for a velocity dependent L_a and Figure 4.12(b) for the case of a constant L_a. In the simulation of Figure 4.12(b) the constant value selected for L_a was too high and the vehicle when following the trajectory cut the curves by its interior side.

Path following controllers comparison

Although the Fuzzy-Logic based path following Controller (FLC) fulfills the requirement of being easy to be tweaked and adjusted by a non-skilled programmer, its ability to fulfill the requirements of cybercars scenarios is yet to be analyzed. This motivated a comparative study of the FLC with a canonical controller used in cybercars scenarios. A Chained Form based path following Controller (CFC) (described in Appendix B) was chosen as a canonical controller. Part of the following results are focused on the performance of the two path-following controllers, which are implemented using two different approaches, the first using fuzzy-logic and the second using chained systems theory. The control effort and the errors magnitude along the path are evaluated in a comparative way. The CFC parameters of equation (B.32) were used as in [95]: $k_1 = \lambda^3$, $k_2 = 3\lambda^2$, and $k_3 = 3\lambda$ with $\lambda = 5$. The value of λ was obtained iteratively starting from an initial guess $\lambda = 8$.

From Figures 4.13 to 4.16 one can observe the effectiveness of both controllers in guiding the car along a predefined path shown in Figure 4.16. From the analysis of Figure 4.13 and Figure 4.14 it is possible to observe that the fuzzy controller is generally better in coping with the angle error (Figure4.13) and the lateral error (Figure 4.14), in both errors the chained form has an higher overshoot than the fuzzy controller. It is also clear that the chained form controller attempts to reduce the errors with a faster response, but the reduction is only partially achieved, afterwards the errors rise again. The previous errors dynamics reveals a two lobes shape when analyzed over time, which does not occur with the fuzzy controller. Although the fuzzy-logic has a better performance in convergence with a pre-defined path it has also some drawbacks, the steering command is not as smoother (Figure 4.15) than the steering command of the chained form controller, and the control effort is higher for the fuzzy-logic. Figure 4.16 shows the path followed by both controllers:

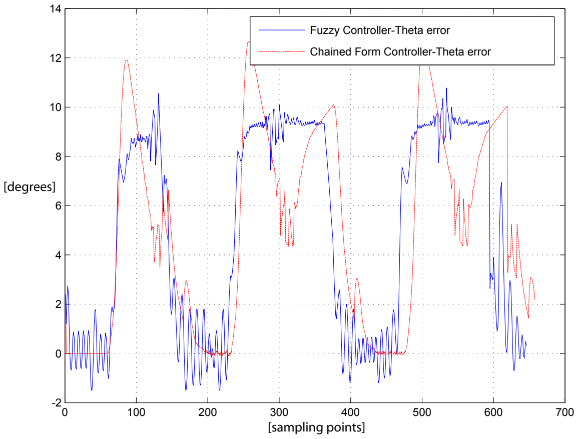


Figure 4.13: θ_e^{tg} and θ_e^{cp}: heading error using the chained form controller (dashed line) and heading error using the fuzzy-logic controller (solid line) in degrees.

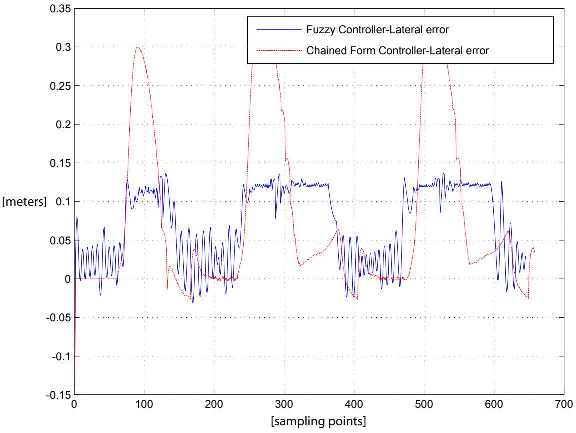


Figure 4.14: $d^t g_e$ and $d^c p_e$ respectively lateral error using the chained form controller (dashed line) and lateral error using the fuzzy-logic controller (solid line) in meters.

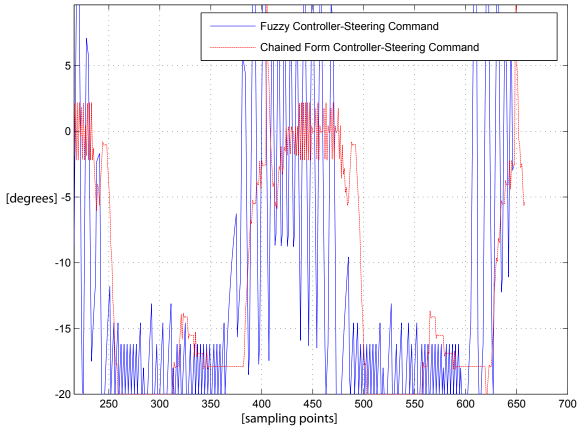


Figure 4.15: ϕ_c: steering command using the chained form controller (dashed line) and steering command using the fuzzy-logic controller (solid line).

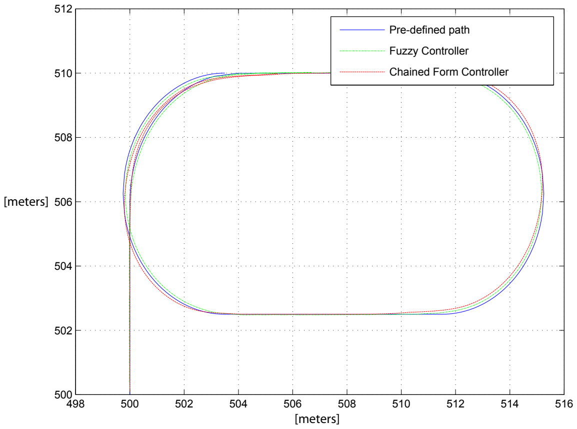


Figure 4.16: Path following simulation results, assuming no odometry errors (neither measurement noise nor cumulative errors).

the solid line is the predefined path, the dashed dotted line is the path followed by the vehicle when using the fuzzy controller and the dashed line is the path followed when using the chained form controller. The fuzzy controller behaves better on the curves than the chained form controller but it is worst when the path is a straight line. The chained form here analyzed does not embodies the same prediction behaviour when using the lookahead distance L_a; has the fuzzy controller, which may be one of the reasons of its inefficiency in curve.

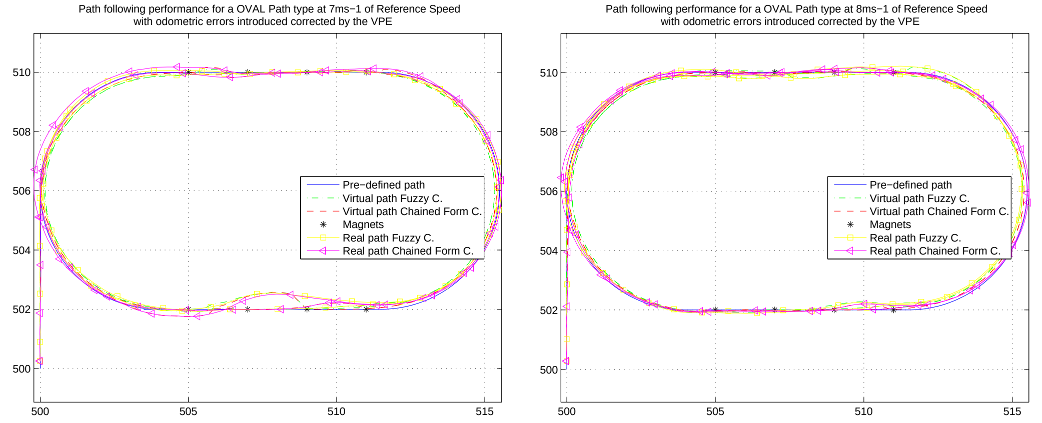
Extensive simulations and error analysis concerning path following assuming that there are no odometry errors introduced, are presented in [96];

Path Following Controller sensitivity to error corrections

Using VPE (see chapter 6) to correct the pose error corrections, it was possible to analyze the controllers sensitivity to vehicle pose corrections. The following simulation results assume the control point at the CG for both Fuzzy and Chained form controllers, the errors introduced are described in chapter 6.

From Figure 4.19(a-h) one can observe that the CFC errors profile and commands profile is similar for both $7ms^1$ and $8ms^{-1}$. The same observation can be made for the FLC. The histograms presented denote the values distribution when the reference speed is $8ms^{-1}$. Among the controllers the angle error profile (i.e. theta error profile) is similar but the angle error Root Mean Square (RMS) is higher in the CFC than in the FLC, see Figure 4.19(i). For the lateral error profile the error analysis revealed a better performance for the FLC, and 0 centered distribution of this error type , see Figure 4.19(j).

The steering command was not smooth in both controllers (Figure 4.19(k)) the histograms reflected this with a sparse distribution, the right steering turn $(-20[Degrees]$



(a)

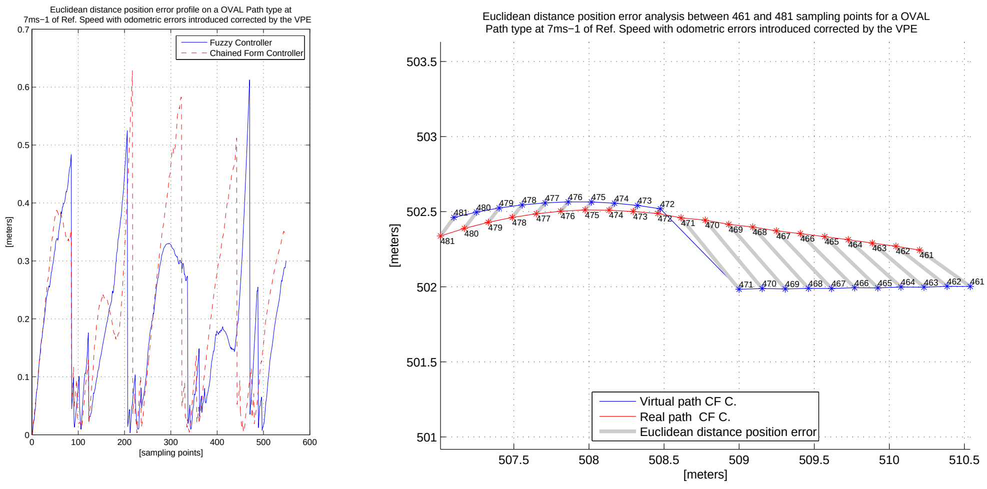
(b)

Figure 4.17: Path following simulation results with odometric errors introduced corrected by the VPE: (a) 7ms-1 (b) 8ms-1:

turn) was more prevalent since the predefined path is oval, i.e. all curves are to the right side. The main difference among the controllers was the velocity command Figure 4.19(l), i.e. for the same reference speed the velocity command was higher for the FLC, this mean that the predefined path loop took more time for the CFC. The path followed by both controller is presented in Figure 4.17(a-b). The performance of the controller is similar, and the recalibration is accomplish with success. The angle and lateral errors feeded as inputs are not continuous in time, they show significant values change, as result of a recalibration, for which both controllers were able to cope with, see Figure 4.19(a-h). The recalibration procedure occur every time a magnet is detected, this doesn't mean that every time the vehicle passes over a magnet a recalibration occur. The speed may be very high and the magnet is crossed over in between two time samples, therefore no recalibration is made. If the sampling time was chosen to be very small then all magnets cross over were detected, but that would be an unrealistic scenario.

Figure 4.18(b), shows the performance of the CFC when systematic errors are introduced to real arc length Δ_R, as in equation 6.60. The arc length Δ_R is multiplied by K_{se} factor, and since $K_{se} > 1$ (i.e. $K_{se} = 1.03$) the real velocity is higher than the virtual velocity that the vehicle controller is having as input, therefore the vehicle might have a real zero lateral and angle error and a virtual zero lateral and angle error but the real position is far ahead from the virtual position, this type of error is presented in Figure 4.18(a-b), this type of error is computed as an euclidean distance from real position to the virtual position.

In Figure 4.18(a) it is depicted the euclidean distance error for both controllers, it is also possible the observe where the recalibration procedure has been occurred. Every



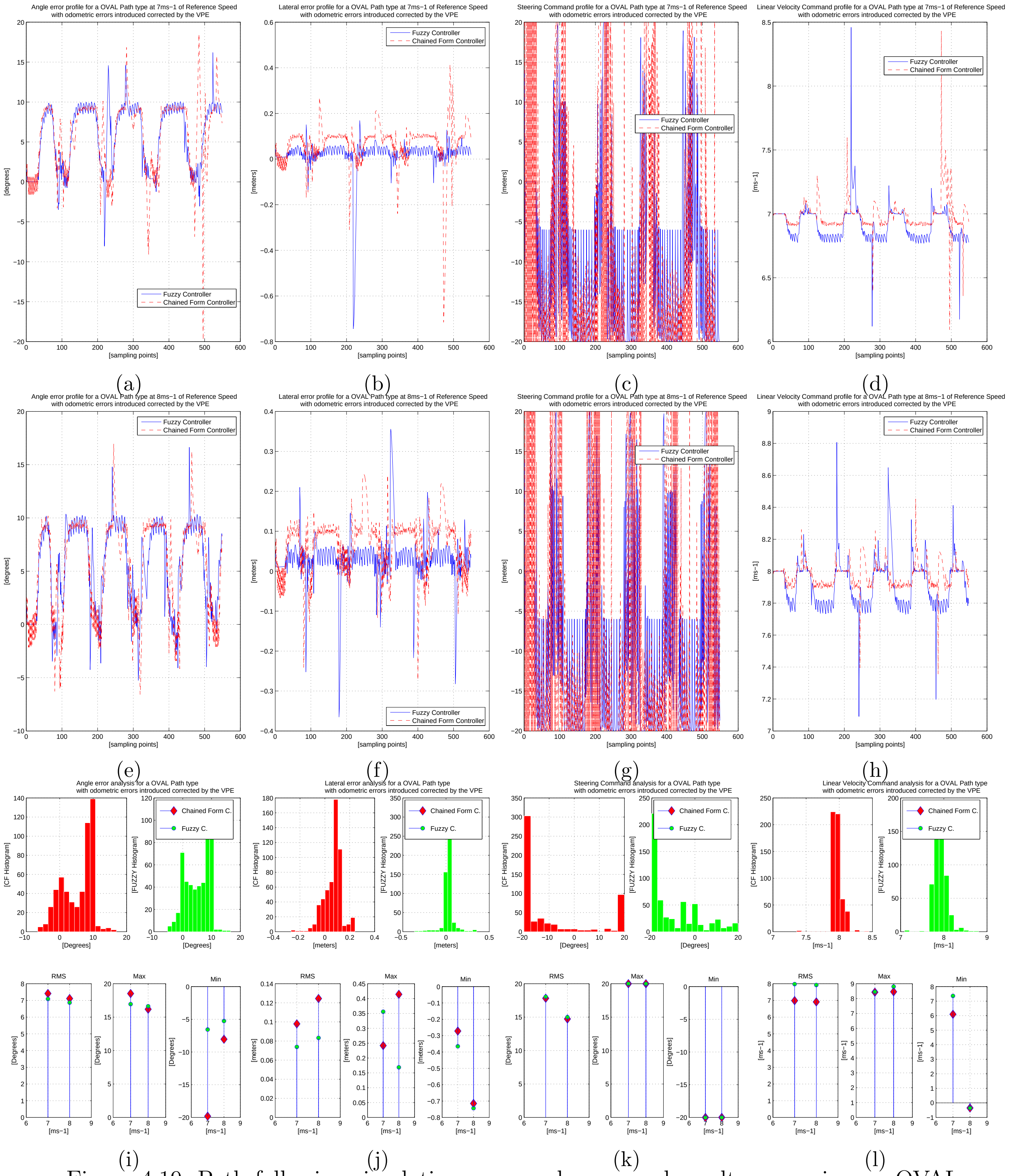
(a) (b)

Figure 4.18: Path following simulation results, assuming an OVAL path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE: (a)Euclidean distance position error profile for both CF C. and FUZZY C. (b) Euclidean distance error for the CF C. between a the 461 and 481 sampling points.

recalibration corresponds to a significative reduction in the euclidean error, i.e. around the sampling points number 100, 225, 350 and 370. This magnitude of error is acceptable since the controllers developed are of path following type, rather than of path tracking type. It is important to notice that the position in which the controller actuates is the virtual (Virtual Path) since it has no means of knowing about the accumulated errors.

The recalibration procedure and consecutively the euclidean distance error reduction are better visualized in Figure 4.18(b), in this scenario besides having a considerable lateral error, the vehicle is also ahead from the virtual position. When a recalibration occurs; at the 472 sampling point; the error in position is corrected and the real position is almost coincident with the virtual position. Before the recalibration; at the 472 sampling point; the euclidean error is very high and after the recalibration the euclidean error is almost null.

Figure 4.20(a-f) represent the euclidean (a)(b), lateral (c)(d) and angle errors (e)(f) respectively. There are ten magnets used for recalibration, five in each straight line, each one marked as a gray strip. The $z - axis$ of Figure 4.20(a-f) represent the error plus the time passed to reach that position. On these figures one can observe has time goes by the errors are accumulated and every time it detects a magnet the virtual position is updated to the real position (Real Path) due to the good estimate about the real position made by the VPE. Although the pose corrections are significant both controllers have a good behaviour handling these sudden path changes.

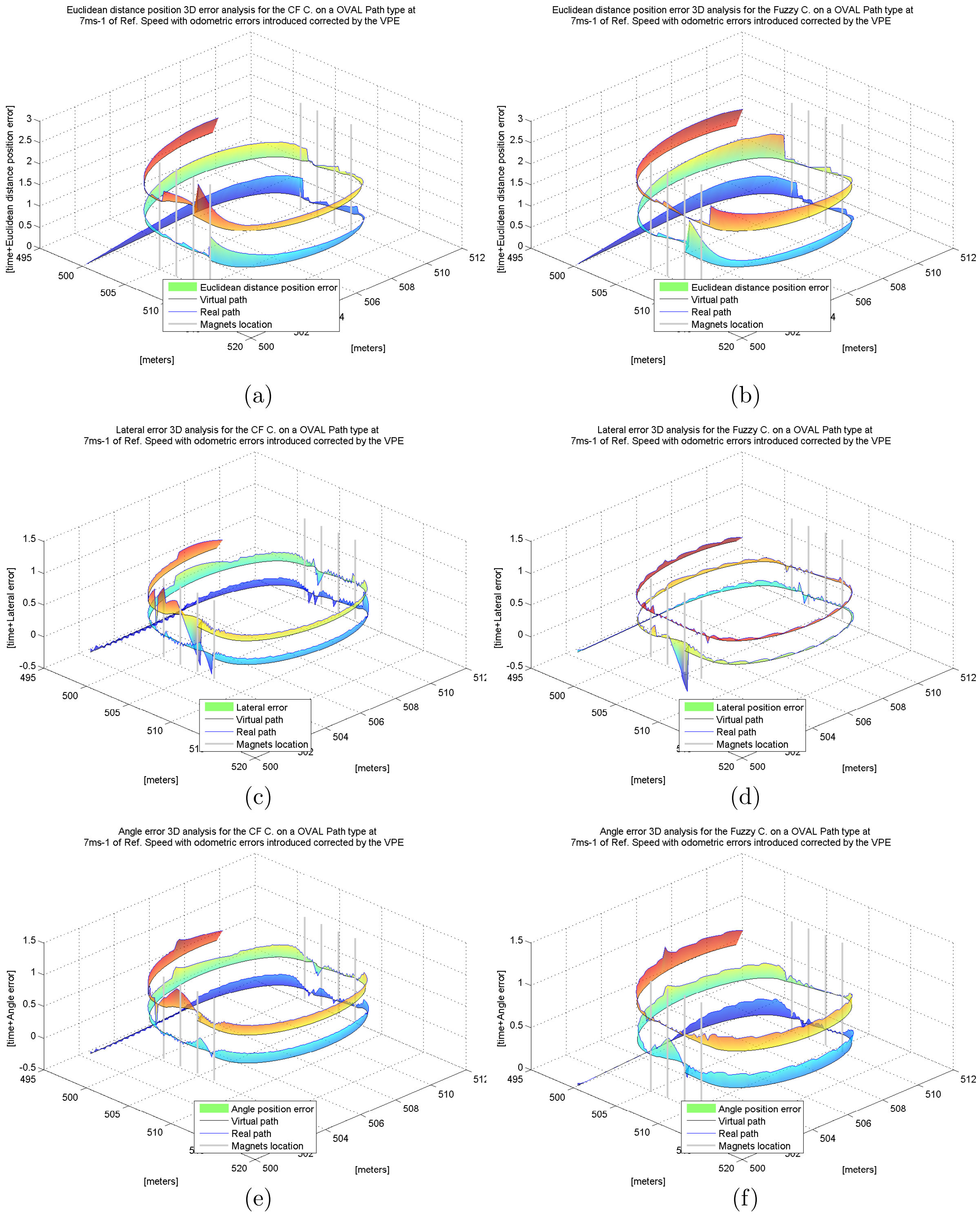


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.19: Path following simulation error and command results, assuming an OVAL path type at 7ms-1 and 8ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE: (a)(e)(i) Theta error (b)(f)(j) Lateral error (c)(g)(k) Steering Command (d)(h)(l) Linear Velocity Command: with odometric errors introduced corrected by the VPE



Euclidean distance position 3D error analysis for the CF C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(a)

Euclidean distance position error 3D analysis for the Fuzzy C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(b)

Lateral error 3D analysis for the CF C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(c)

Lateral error 3D analysis for the Fuzzy C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(d)

Angle error 3D analysis for the CF C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(e)

Angle error 3D analysis for the Fuzzy C. on a OVAL Path type at 7ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE

(f)

Figure 4.20: Path following simulation error results, assuming an OVAL path type at 7ms-1 and 8ms-1 of Ref. Speed with odometric errors introduced corrected by the VPE: (a) Path following at 7ms-1 (b) Path following at 8ms-1: with odometric errors introduced corrected by the VPE

Extensive simulations regarding controllers performance and error analysis when the the vehicle is subjected to odometry errors, are presented in [96];

4.5 Final Remarks

A highly parameterizable simulator ISR-TRAFSIM in the category of microscopic simulators and operating on discrete events, was developed. With this parameterizations, we are able to simulate complex vehicle/traffic situations; and thus one can evaluate the impacts of different ITM algorithms. As with the previously mentioned simulators it has its strengths and its weaknesses. The main advantages are the easiness to develop and integrate new modules; implemented in Matlab enabling a non-skilled programmer to customize code. The simulator has already implemented highly complex sensors and it has a strong feature-rich GUI. The main disadvantages are the poor computing efficiency and simple V2I/V2V communication model.

Extensive simulations and error analysis concerning path following performance of a easy to tune and adapt fuzzy-logic based controller, were conducted. The fuzzy-logic based controller was compared with a canonical chained form controller and the results are presented in this chapter and in [96]. The simulations comprise scenarios where odometry errors are introduced and situations where the vehicles'pose is corrected by the VPE. The simulation results provide sufficient information to conclude that the controllers have generally the same behaviour and similar performances, but the CFC is more exigent regarding its development and tuning cost.

Chapter 5

Intelligent Traffic Management

Contents

This chapter describes the implemented intelligent traffic management algorithms applied to automatic regulation of traffic at intersections, and the emissions and fuel consumption computation module. An assessment of the algorithms simulations results and their performance metrics is also presented in this chapter.

5.1 Intelligent Traffic Management

5.1.1 Spatio-temporal matrix

Using the vehicle's VPE estimated position, the intersection management system computes the distance from the vehicle to the intersection. When the vehicle enters in a predetermined control radius, the infrastructure agent runs the selected ITM algorithm. For each vehicle request, the selected ITM algorithm, generates a collision-free path to be followed by the vehicle, while traversing the intersection. The collision-free path is decomposed in a 2D path and an associated speed profile. The 2D path and speed profile are used in a three-dimensional matrix reservation procedure. The three-dimensional matrix is composed by a set of layers, one layer per each time instant, where δt is the sampling period. Each layer is divided in cells, where each cell represents a two-dimensional space in the scenario map. Each cell can be in one of the following status: occupied or free. The three-dimensional matrix integrates space and time and therefore it is henceforth designated *spatio-temporal matrix.* The space used by a vehicle when following the collision-free path, is allocated by the active ITM in the spatio-temporal matrix, see Figure 5.1. The cells areas reserved in the spatio-temporal matrix are wider than the vehicle dimensions; this safety margin A_{SM} ($A_{SM} = [wide; long][m]$) is used to avoid accidents in case a vehicle does not follow exactly the generated speed profile. Adjusting this safety margin also takes into account the type of vehicle: autonomous or driven by human. The parameter A_{SM} also accounts for the error-prone positions, *i.e.* it must be increased for less accurate vehicle pose estimations. The generated speed profile and navigation directions is used by the PFC [97] or by a human driver assisted by this information delivered through an Human Computer Interaction (HCI).

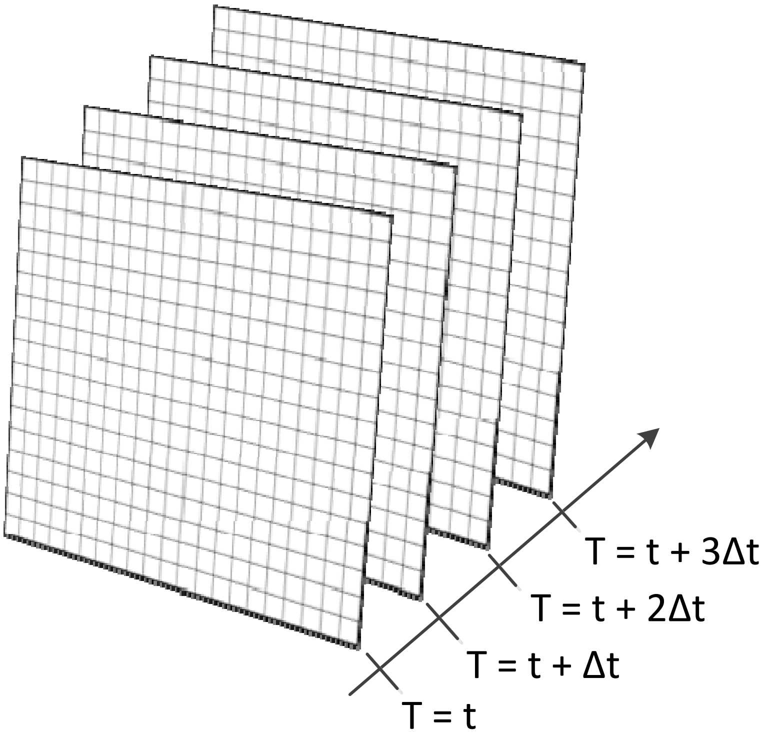


Figure 5.1: Three-dimensional time-space matrix. Each layer represents a time instant and each cell on each layer represents a physical space on the scenario.

5.1.2 Traditional intersection management techniques

Roundabouts and crossroads are usually operated using traditional techniques for traffic management. Intelligent traffic management algorithms are evaluated againts these traditional techniques:

Intelligent Traffic Management Deactivated (ITMD) While on ITMD, vehicles respecting traffic laws and interacting with surrounding vehicles, follow their own will and desired speed profile. There is no predefined speed profile and the speed is determined at each time interval using a longitudinal inter-vehicle gap speed model and a transversal time to collision algorithm. Consequently, vehicles are "manually" controlled, *i.e.* vehicles are driven as an unassisted human driver would drive. In this mode vehicles crossing the intersection do not require a V2X CS for proper operation.

Standard Traffic Light System (STLS) In STLS mode of operation it was considered a fixed-time control strategy, with a phase serving each direction at a time. Regardless of changes in traffic volumes, the STLS uses the same preset in every cycle. A full cycle length for one direction is composed of $T_{dir} = T_{green} + T_{yellow} + T_{red}$, during T_{dir} all other directions are in red light status. A full intersection cycle is composed by $T_{inter} = 4 \cdot T_{dir}$. During T_{red} all intersection traffic lights are red, enabling vehicles to clear the intersection safely. The STLS system does not require a V2X CS, neither the vehicles crossing the intersection, the vehicles respecting traffic laws and interacting with surrounding vehicles, follow their own desired will and speed.

Fixed-time signals assign the green light, yellow light and red light to the each travel direction of an intersection for a predefined amount of time:

West⇒East	South⇒North	East⇒West	North⇒South
T_{Green}	Red	Red	Red
T_{Yellow}	Red	Red	Red
T_{Red}	Red	Red	Red
Red	T_{Green}	Red	Red
Red	T_{Yellow}	Red	Red
Red	T_{Red}	Red	Red
⋮	⋮	⋮	⋮

Table 5.1: Fixed time intervals.

In this research work the standard traffic light system was developed for crossroad intersections (CI-STLS).

5.1.3 Car Intention Intelligent Traffic Management (CIITM)

Using the infrastructure sensors to detect incoming and outgoing traffic, the Car Intention Intelligent Traffic Management (CIITM) manages the intersection using a space allocation algorithm. In addition to the traffic lights, the intersection is assumed to be equipped with an intelligent vision system and road drive-through inductive loop sensors. The intelligent vision system, assumed to be present on the infrastructure, must detect the approaching vehicles and check the status of the turning light signals of each vehicle. A set of drive-through inductive loop sensors in each outgoing lane are capable of detecting the vehicles leaving the intersection.

The management process occurs in three stages. On the first stage the CIITM detects an approaching vehicle and the status of the turning signal, if the turning signal is off the

Algorithm 2 CIITM

```
Require: spatio-temporal matrix, vehicle origin, vehicle destination, inward sensors, outward
  sensors

  Use inward sensors to determine destination and compute trajectory path
  if destination and trajectory path free then
    Reserve cell(x,y) in the spatial matrix
    while vehicle not crossed traffic light do
      Set green light
    end while
    Set red light
  else
    Added to DATABASE QUEUE of cars
  end if
  if outward sensors detection then
    Release cell(x,y) in the spatial matrix
  end if
```

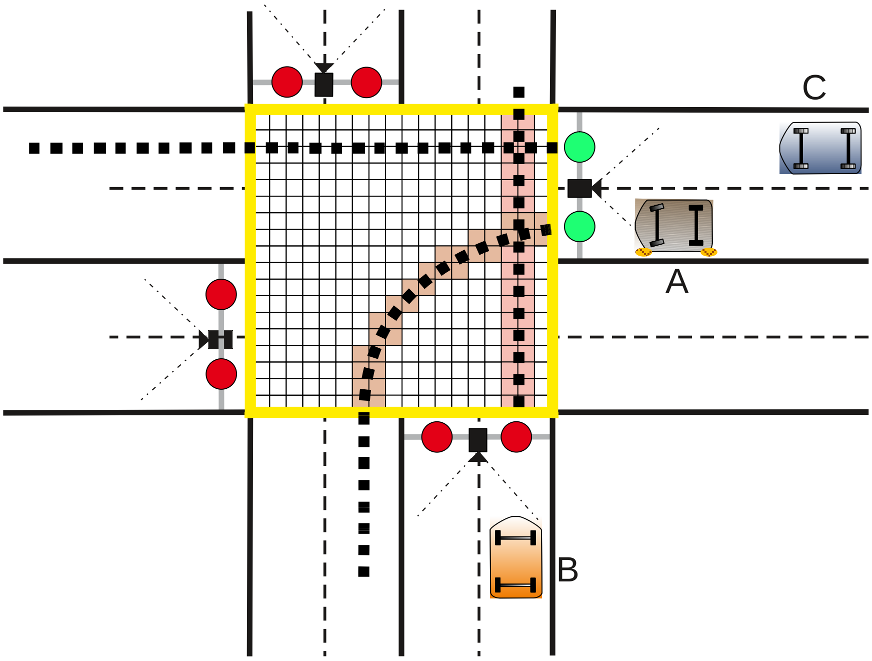


Figure 5.2: CIITM procedure at the CI. Vehicle A has left turning signal ”on”, vehicle B wants to go straight forward as well as vehicle C.

system assumes that the vehicle wants to go straight forward, instead of turning left or right. On next stage the system verifies if the desired outgoing lane is free, if the free space exists, the system makes a space reservation and switches the corresponding traffic light to green, authorizing the vehicle to cross. If the desired outgoing lane is not available the light remains red. On the last stage the passage of vehicle at the output lane is detected and the allocated space is released. In Figure 5.2 the space reservation is shown, where the turning signal for vehicle A is detected to be on (right light signal) and the space required to cross the CI is unallocated, i.e. the cells are free, therefore the traffic light becomes green. However, for the vehicle B (arriving at a later time) no turning signal is detected (the system assumes that the driver wants go straight ahead) and part of the space required to cross the CI is already allocated, with no free path found the traffic light remains in red state. This disallowance to enter the CI box junction, occurs because vehicle B needs to reserve space already reserved for vehicle A. The vehicle B should pass when the vehicle A left the junction area, but to maximize the CI area used, the CIITM system can allow the passage of a vehicle that arrives latter than vehicle B if the desired space is not in conflict with the space already reserved. This situation occur with vehicle C, because it has no turning signal is detected and the system assumes that the driver wants go straight forward, since its desired path does not conflict with previous allocated space, it receives permission to pass before vehicle B can pass. The passage will only be granted to vehicle B after vehicles A and C have gone through the drive-through inductive loops sensors and release the allocated space.

The three stages can be resumed as follows:

1. detection of intended turning behaviour;
2. verification of the status (free/occupied) of the path and outgoing lane, assigning green or red traffic light to the corresponding lane accordingly;
3. detection of the vehicle passage at the output lane and release of the allocated space for usage of following vehicles.

This algorithm does not require a V2X CS, relying solely on infrastructure sensors to detect incoming, outgoing traffic and intended turning behaviour. In this research work the car intention intelligent traffic management system was developed for crossroad intersections (CI-CIITM).

The pseudo-code of the CIITM is presented in Algorithm 2.

5.1.4 Waiting Method Intelligent Traffic Management (WMITM)

The WMITM reservation process starts with a trajectory generation based on the information sent by the vehicle with the position and destination lane, after receiving this information a trajectory is generated. This algorithm relies on a V2X CS to receive the vehicle's current estimated pose and destination lane and send the speed profile and the navigational directions.

The first step after trajectory generation, is to determine the occupied cells by the vehicle in layer T = t (Figure 5.1). The cells area, reserved in each layer, is wider than the vehicle dimensions, this is a safety measure, used to avoid accidents in case the driver

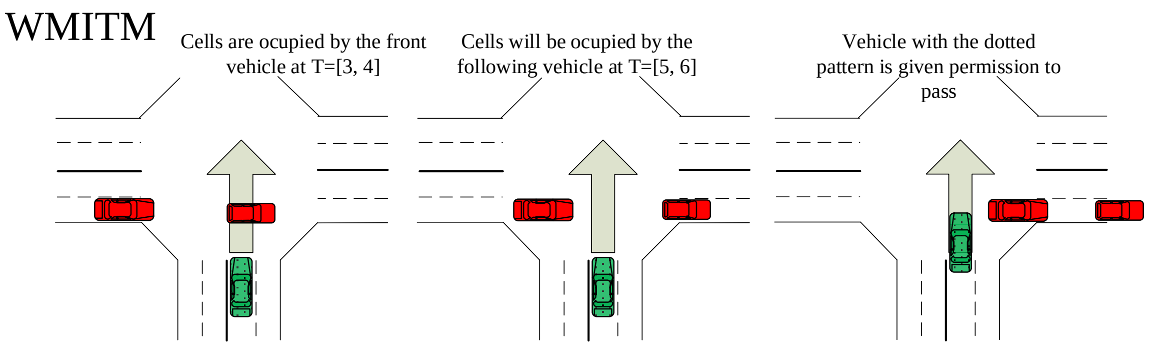

WMITM
Cells are ocupied by the front vehicle at T=[3, 4]
Cells will be ocupied by the following vehicle at T=[5, 6]
Vehicle with the dotted pattern is given permission to pass

Figure 5.3: Cell selection using *w*aiting algorithm. The WMITM algorithm determines that a vehicle must wait until the space required is no longer used.

Algorithm 3 WMITM

Require: spatio-temporal matrix, vehicle V2I/V2V communication, vehicle origin, vehicle destination

```
Use origin and destination to generate trajectory path
if destination free then
   Set obstacles list to empty
   while unsuccessful reservation do
      Use trajectory and obstacles list to generate the speed profile
      Use speed profile and spatio-temporal matrix to detect collisions
      if collisions then
         Find layer(i) and cell(x,y) of the spatio-temporal matrix with collision
         Set aux ← last layer
         while no solution do
            if layer(aux) is occupied then
               Save aux-1 and update obstacles list
               solution found
            end if
            Decrease aux
         end while
         Compute speed profile delaying the arrival time to cell(x,y) until layer(i)
      else
         Update spatio-temporal matrix
         successful reservation
      end if
   end while
   Send speed profile to vehicle V2I/V2V address
else
   Send waiting signal to vehicle V2I/V2V address
end if
```

does not follow exactly the generated speed profile. On the next step, the position of vehicle in the layer labeled with T $= t + \Delta t$ will be determined using its speed, car dynamic restrictions and estimated trajectory. If the vehicle is trying to reserve a space in a layer already reserved, the algorithm instruct the current vehicle to slow down before it reaches the occupied zone. The previous speed profile setpoints will also be modified so that during braking, the deceleration does not exceed the *comfortable acceleration* defined in section A, making the ride comfortable and giving time to reduce the speed or stop slowly. Therefore this algorithm imposes that the vehicle wait until the space it is no longer reserved on any upcomming layer. Although this method will always achieve a solution, it has a poor efficiency, since it does not use free spaces of the desired path, *e.g.* if the algorithm attempts to occupy a cell at $T = 3s$ and the cell was occupied between the following time intervals $T = [3; 4] \bigcup [5; 6]$, the speed profile generated will slow down the vehicle (or even stop it) until the moment just after $T = 6s$, as shown in Figure 5.3. This method is described in the pseudo-code Algorithm 3. In this research work the waiting method intelligent traffic management was developed for both roundabout (RI-WMITM) and crossroad intersections (CI-WMITM).

5.1.5 Early Method Intelligent Traffic Management (EMITM)

If a vehicle is inside a predetermined intersection control radius, the EMITM algorithm generates a collision-free path. The collision-free path is decomposed into a speed profile along with navigational directions. The speed profile and navigational directions are sent only once, therefore the success of this algorithm relies on the important premise that the vehicle follows the recommendations precisely. This method is based on a spatio-temporal matrix reservation scheme. The reservation process starts with a path generation based on the vehicle's current estimated pose and destination lane. Using the preferred velocity defined by the *agent environmental settings* and the determined path a speed profile is computed. The generated path and speed profile is used in the reservation procedure applied to the spatio-temporal matrix. If a reservation conflict should exist then the speed profile is recomputed so that the vehicle waits until the first free space is found. This collision avoidance reservation process is repeated until an origin-to-destination collision-free path is obtained. Once the collision-free path is obtained the final speed profile along with navigational directions are sent to the respective vehicle.

This algorithm relies on a V2X CS to receive the vehicle's current estimated pose and destination lane and to send the speed profile and the navigational directions. The *EMITM* pseudo-code is presented in Algorithm 4. The EMITM algorithm has roughly the same operation as a previously developed algorithm (WMITM) [98]. The main difference is that in conflicting situations the *EMITM* will try to reserve the first free space, while the *WMITM* will try to reserve the immediate free space after the last reserved space, *e.g.* if the algorithm attempts to reserve a cell at $T = 3s$ and the cell was occupied between the following time intervals $T = [3; 4] \bigcup [5; 6]$, the speed profile generated by *EMITM* will determine that the vehicle has to slow down until the moment just after $T = 4s$ while

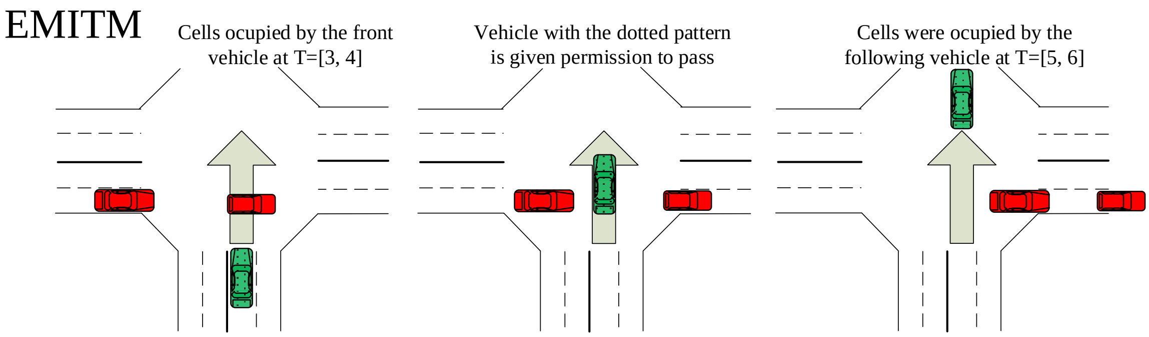

EMITM
Cells ocupied by the front vehicle at T=[3, 4]
Vehicle with the dotted pattern is given permission to pass
Cells were ocupied by the following vehicle at T=[5, 6]

Figure 5.4: Cell selection using *e*arly algorithm. The EMITM algorithm reserves the space as soon as it is available.

Algorithm 4 EMITM

Require: spatio-temporal matrix, vehicle V2I/V2V communication, `vehicle` origin, `vehicle` destination

```
Use origin and destination to generate trajectory path
if destination free then
   Set obstacles list to empty
   while unsuccessful reservation do
      Use trajectory and obstacles list to generate the speed profile
      Use speed profile and spatio-temporal matrix to detect collisions
      if collisions then
         Find layer(i) and cell(x,y) of the spatio-temporal matrix with collision
         for i ← layer to last layer do
            if cell in layer(i) is free then
               Update obstacles list
               Exit for loop
            end if
         end for
         Compute speed profile delaying the arrival time to cell(x,y) until layer(i)
      else
         Update spatio-temporal matrix
         successful reservation
      end if
   end while
   Send speed profile to vehicle V2I/V2V address
else
   Send waiting signal to vehicle V2I/V2V address
end if
```

the *WMITM* will determine to slow the vehicle down until the moment just after $T = 6s$, see Figure 5.4. This is the major difference between both algorithms. In this research work the early method intelligent traffic management was developed for both roundabouts (RI-WMITM) and crossroad intersections (CI-WMITM).

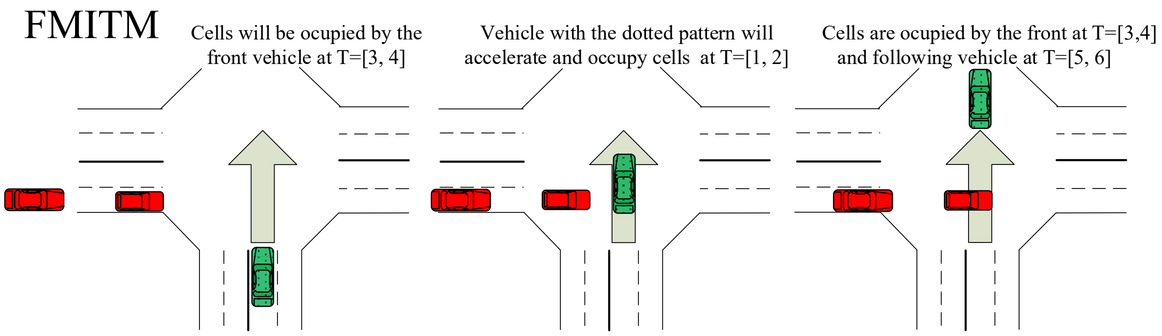

FMITM
Cells will be ocupied by the front vehicle at T=[3, 4]
Vehicle with the dotted pattern will accelerate and occupy cells at T=[1, 2]
Cells are ocupied by the front at T=[3,4] and following vehicle at T=[5, 6]

Figure 5.5: Cell selection using $forward$ algorithm. The FMITM algorithm determines that a vehicle must wait until the space required is no longer used.

5.1.6 Forward Method for Intelligent Traffic Management (FMITM)

The WMITM and EMITM traffic management systems, when generating a vehicle speed profile, if a potential collision with a vehicle is detected, they will try to avoid the collision by decelerating until it is possible to reserve a free path in the spatio-temporal matrix. The FMITM accelerates the vehicle in order to pass at the collision point ahead of the colliding vehicle. In the search for a solution to the desired trajectory the FMITM enable vehicles to travel over the legal speed limit that is set for the roads and intersections. The over the legal speed is constrained to a maximum speed V_{MOLS}. The ride over the legal speed can only be used when resolving the collision events, being restored the preset speed after the collision avoidance. The range of communication had to the extended, due to the higher speed that vehicles can achieve in order to resolve collisions, therefore vehicles are being tracked and properly managed earlier than previous ITM algorithms.

If the algorithm attempts to occupy a cell at $T = 3s$ and the cell was occupied between the following time intervals $T = [3; 4] \bigcup [5; 6]$, the speed profile generated will accelerate the vehicle until it is able to pass on a moment just before $T = 3s$. As shown in Figure 5.5 the vehicle with the dotted pattern will accelerate and occupy cells at T=[1, 2]. This method is described in the pseudo-code Algorithm 5. This algorithm relies on a V2X CS to receive the vehicle's current estimated pose and destination lane and to send the speed profile and the navigational directions. Although this method may achieve a higher traffic efficiency, not always achieves a solution, since it is not possible to infinitely accelerate a vehicle to find free spaces of the desired path. When it is not possible to find a solution with this algorithm, the computation of the path falls back to the EMITM algorithm, to set the appropriate speed profile.

Algorithm 5 FMITM

```
Require: spatio-temporal matrix, vehicle V2I/V2V communication, vehicle origin, vehicle destination,
  vehicle MaxTemporarySpeed

  Use origin and destination to generate trajectory path
  if destination free then
    Set obstacles list to empty
    while unsuccessful reservation do
      Use trajectory and obstacles list to generate the speed profile
      Use speed profile and spatio-temporal matrix to detect collisions
      if collisions then
        Find acceleration to arrive earlier at cell(x,y) of the spatio-temporal matrix with
        collision
        Compute speed profile anticipate the arrival time to cell(x,y)
      else
        Update spatio-temporal matrix
        successful reservation
      end if
    end while
    Send speed profile to vehicle V2I/V2V address
  else
    Send waiting signal to vehicle V2I/V2V address
  end if
```

5.1.7 Legacy Early Method for Intelligent Traffic Management (LEMITM)

In a previous sections, we have presented spatio-temporal reservation scheme of traffic at intersections for vehicles equipped with V2V and V2I CS. Here, a legacy algorithm is proposed, which enables a low percentage of vehicles, not equipped or with faulty V2V and V2I CS, to traverse the intelligent intersection using the spatio-temporal reservation scheme.

Using the infrastructure sensors to detect incoming and outgoing traffic, the LEMITM algorithm manages the intersection using a spatio-temporal reservation scheme. When using the LEMITM algorithm, vehicles do not have to follow an imposed speed profile, this fact enables the use of the infrastructure by vehicles not equipped or with faulty V2V and V2I CS. If the vehicle is not equipped with a communication system, it is not possible to know in advance to which destination and at what speed the vehicle will travel. Therefore the LEMITM algorithm will attempt to reserve the cells of all possible trajectories, during a time interval sufficient for any type of vehicle (car or truck) at different speeds, to traverse the intersection completely, see Figs. 5.6 and 5.7. A speed safety margin V_{SM} ($V_{SM} = [min\ speed; max\ speed][km/h]$) was set to cope with vehicle speed variations. The management process occurs in four steps. The first step is to detect a vehicle at an entrance lane. On the second step, all possible trajectories are reserved on

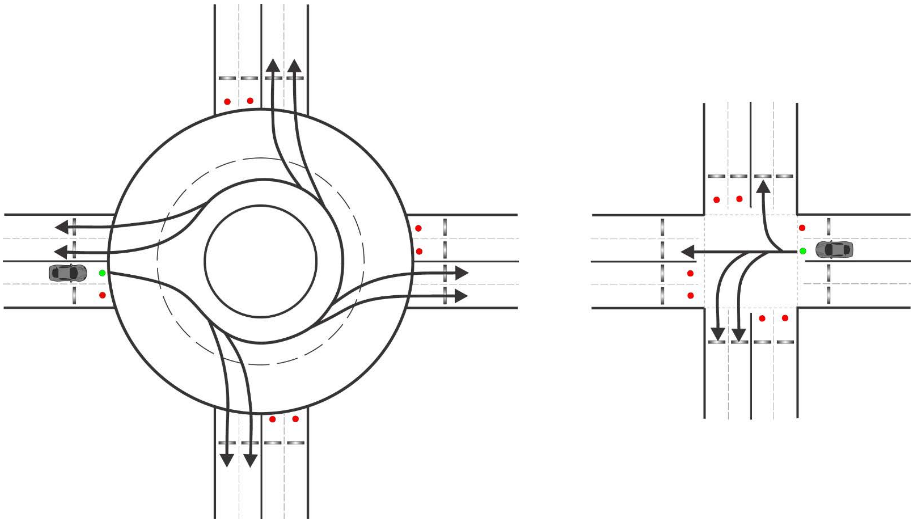


Figure 5.6: Cell reservation using LEMITM algorithm. The LEMITM algorithm reserves the space for all possible trajectories.

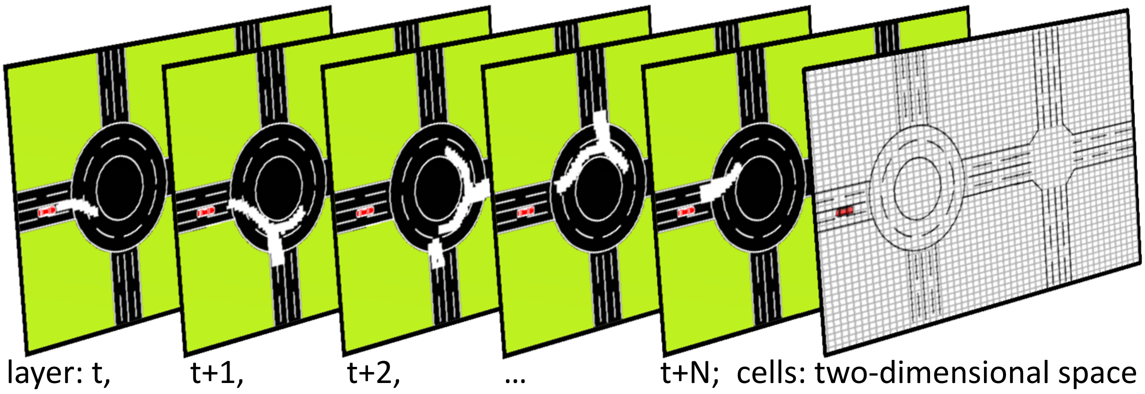


Figure 5.7: Spatio-temporal matrix sequence using LEMITM algorithm.

the spatio-temporal matrix, for a time sufficient for the vehicle to cross the intersection completely. After successfully completed the reservation, the green traffic light of the corresponding lane is turned on until the vehicle enters the intersection. The next step is to set on the red traffic light of the corresponding lane. The reserved space is sequentially freed as time passes, after permission was given to enter the intersection. The release of the previously reserved space will enable upcoming reservations. The pseudo-code of the *LEMITM* is presented in Algorithm 6. In order to use this algorithm the intersection must be signalized.

Algorithm 6 LEMITM

```
Input:
$stm(l, x, y)$: spatio-temporal matrix
$veh_{el}$: vehicle entrance lane
$apd(k)$: all possible destinations
$V_{SM} = [min\ speed; max\ speed]$: speed safety margin
Output:
$apdcr$: all possible destinations cells reservation
$ttsGlctl$: time to set GREEN light of the corresponding traffic light
Internal variables:
$aot$: algorithm offset time
$ReSt(k)$: reservation status (1 successful, 0 unsuccessful)
$CoSt$: collision status (1 collision, 0 no collision) $tSp$: test speed

$aot = 0$
while $\exists i \in [1, k] : \neg ReSt(i) == 1$ do
    :restart search
    for i $\leftarrow apd(1)$ to $apd(k)$ do
        for $tSp \leftarrow V_{SM}(min)$ to $V_{SM}(max)$ do
            $CoSt$=test_collision($tSp$,$stm(l, x, y)$)
            if $CoSt == 1$ then
                Find $layer(j)$ and $cell(x, y)$ of the $stm(l, x, y)$ with $CoSt == 1$
                for j $\leftarrow layer(1)$ to $layer(l)$ do
                    if $cell(x, y)$ in $layer(j)$ is free then
                        $aot = j$
                        $\forall i \in [1, k] : ReSt(i) == 0$
                        Exit for loop goto :restart search
                    end if
                end for
            else
                $ReSt(i) = 1$
            end if
        end for
    end for
end while
Reserve cells in the $stm(l, x, y)$
$ttsGlctl = j$
```

5.2 Environment Impact: emissions and fuel consumption computation

ISR-TRAFSIM includes a newly developed module that computes CO_2 emissions as well as the fuel consumption of the vehicles traversing the intersections, based on an engine thermal model and the vehicle longitudinal dynamics.

The engine power output and speed from physical interrelationships are computed as follows: the force required to accelerate the vehicle through a time step is calculated directly from the required linear speed which is translated into a rotational speed. To compute the fuel needed to meet speed requirements, the required force is translated into the torque

Algorithm 7 Emissions and Fuel Consumption

```
Require: Vehicle Mechanical specifications, Engine Specifications, Fuel Characteristics, vehicle Speed
  ↪Use vehicle Speed and Vehicle Mechanical Specifications to compute the Required Traction Force
  $F_{trac} = m_{veh} \cdot g \cdot C_R + 1/2 \cdot \rho_{air} \cdot A_f \cdot C_D \cdot \bar{v}_{veh}^2 + m_{veh} \cdot \dot{v}_{veh} + m_{veh} \cdot g \cdot \sin(\alpha)\ [N]$
  if $F_{trac} > 0$ then
    ↪Use vehicle Speed and Gear Ratio to obtain Rotational Speed: $\omega_{eng} = \frac{\bar{v}_{veh}}{r_{wheel}} \cdot \gamma\ [rad/s]$
    ↪Use the Required Traction Force, the Rotational Speed and the vehicle Speed to obtain Required Torque: $T_{eng} = \frac{F_{trac}}{\omega_{eng}/\bar{v}_{veh}}\ [Nm]$
    ↪Use the Required Traction Force and the engine displacement to obtain brake mean effective pressure: $\rho_{me} = \frac{T_{eng} \cdot N \cdot \pi}{V_d}\ [bar]$
    Applying the engine friction model compute the engine friction losses: $\rho_{me0f} = k_1 \cdot (k_2 + k_3 \cdot S^2 \cdot \omega_{eng}^2)\Pi_{max} \cdot \sqrt{\frac{k_4}{B}}\ [bar]$
    ↪Use friction losses and global engine losses to obtain global engine losses: $\rho_{me0} = \rho_{me0g} + \rho_{me0f}\ [bar]$
    ↪Use the brake mean effective pressure, the global engine losses and the indicated engine efficiency to obtain fuel mean effective pressure: $\rho_{mf} = \frac{\rho_{me} + \rho_{me0}}{\varepsilon}\ [bar]$
    switch (Fuel Type)
    case Gasoline:
      $H_l^{fuel} = H_l^{C_8H_{18}}\ [MJ/kg]$
      $\rho_{fuel} = \rho_{C_8H_{18}}\ [kg/l]$
      $N_{molecules}^{CO2} = 8$
      $M_{fuel} = M_{C_8H_{18}}$
      $WTT_{fuel} = WTT_{C_8H_{18}}^{CO_2}$
    case Ethanol:
      $H_l^{fuel} = H_l^{C_2H_5}\ [MJ/kg]$
      $\rho_{fuel} = \rho_{C_2H_5}\ [kg/l]$
      $N_{molecules}^{CO2} = 2$
      $M_{fuel} = M_{C_2H_5}$
      $WTT_{fuel} = WTT_{C_2H_5}^{CO_2}$
    end switch
    ↪Use the fuel mean effective pressure, the engine displacement and the fuel lower heating value to obtain fuel mass per engine cycle: $m_f = \frac{\rho_{mf} \cdot V_d}{H_l^{fuel}}\ [kg]$
    ↪Use vehicle fuel mass per engine cycle and Rotational Speed to obtain fuel mass flow: $\dot{m}_f = m_f \cdot \frac{\omega_{eng}}{N \cdot \pi}\ [kg/s]$
    ↪Use vehicle fuel mass flow, the time interval, the fuel density and travelled distance to obtain fuel consumption: $FC_{100km} = \frac{\frac{\dot{m}_f \cdot \Delta t}{\rho_{fuel} \cdot 100}}{\frac{\Delta s}{1000}}\ [l/100km]$
    ↪Use vehicle fuel consumption, the fuel density, the number of $CO_2$ molecules produced on the combustion process, the $CO_2$ molar mass and fuel molar mass to obtain TTW fuel $CO_2$ emissions: $TTW_{fuel}^{CO_2} = FC_{100km} \cdot \rho_{fuel} \cdot N_{molecules}^{CO2} \cdot \frac{\frac{M_{CO_2}}{M_{fuel}}}{100} \cdot 1000\ [gCO_2/km]$
    ↪Use vehicle TTW fuel $CO_2$ emissions, the fuel consumption, the fuel density, the fuel lower heating value and the $TTW_{fuel}^{CO_2}$ to obtain WTW $CO_2$ emissions: $WTW_{fuel}^{CO_2} = TTW_{fuel}^{CO_2} + FC_{100km} \cdot \rho_{fuel} \cdot H_l^{fuel} \cdot \frac{WTT_{fuel}}{100}[gCO_2/km]$
  else
    Set $FC_{100km} = 0$ , $TTW_{fuel}^{CO_2} = 0$ and $WTW_{fuel}^{CO_2} = 0$
  end if
```

that must be provided, when a rotational force is applied to a shaft. This approach is called *quasistatic* or *backward-facing* where, assuming that the vehicle follows a required speed, we compute how each component must perform [99]. Component by component, this calculation approach flows backward through the drivetrain, against the tractive power flow direction, until the required fuel use to meet the speed profile is computed. The backward-facing approach can compute the emissions and fuel consumption of any speed profile, and has the advantage of low computing requirements. The implemented backward-facing approach is presented in Algorithm 7.

5.2.1 Vehicle Mechanical Model

The model of the vehicle is based on the mechanical forces acting on the vehicle. In order to compute the vehicle fuel consumption the required power must be computed taking into account its physical dependencies [100]. Considering a vehicle as a mass-point, its equilibrium equation can be written using equation 5.1:

$$F_{trac} = F_{roll} + F_{aero} + F_{inertia} + F_{grade} \tag{5.1}$$

where F_{trac} is the resulting tractive force at the wheels to move the vehicle forward, F_{roll} is the rolling resistance, F_{aero} is the force due to aerodynamic drag, $F_{inertia}$ is the inertial force and F_{grade} is the force due to road slope. Rolling resistance is associated with the friction due to tire deformation as they rotate. The equation for rolling resistance is given by:

$$F_{roll} = m_{veh} \cdot g \cdot C_R \tag{5.2}$$

where m_{veh} is the total vehicle mass, g is the gravity acceleration, and C_r is the coefficient of rolling resistance. The friction generated by the vehicle with the atmosphere as it is traveling through, is referred to as the aerodynamic drag. The drag force is given by:

$$F_{aero} = 1/2 \cdot \rho_{air} \cdot A_f \cdot C_D \cdot \bar{v}_{veh}^2 \tag{5.3}$$

where ρ_{air} is the air density, A_f is the vehicle frontal area, C_D is the vehicle aerodynamic drag coefficient and v_{veh} is the vehicle velocity. The drag force has to be calculated with the mean velocity in every time step interval, *i.e.* $\bar{v}_{veh}$. The inertial force is the load due to the acceleration of the vehicle's mass and it is given by:

$$F_{inertial} = m_{veh} \cdot \dot{v}_{veh} \tag{5.4}$$

The grade force is the component of the vehicle weight opposing the motion of the vehicle and it is given by:

$$F_{grade} = m_{veh} \cdot g \cdot \sin(\alpha) \tag{5.5}$$

where α is the angle of the road from horizontal. In this work the road is assumed to be flat, therefore α is always zero.

5.2.2 Thermal Engine Model

The traction force F_{trac} has to match the total resistance force for every time step in the discrete speed profile [101]. Once obtained the F_{trac}, the first step to compute the necessary engine torque is to calculate the engine rotational speed ω_{eng} using the wheel radius r_{wheel} and the gear ratio:

$$\omega_{eng} = \frac{\bar{v}_{veh}}{r_{wheel}} \cdot \gamma \tag{5.6}$$

where the gear ratio γ is function of $\bar{v}_{veh}$. In order to comply with the determined vehicle speed, the engine must apply the following torque T_{eng} to the crankshaft:

$$T_{eng} = \frac{F_{trac}}{\omega_{eng}/\bar{v}_{veh}} \tag{5.7}$$

The Willans approximation [100] for the engine's torque and efficiency characteristics was used:

$$\rho_{me} = \varepsilon \cdot \rho_{mf} - \rho_{me0} \tag{5.8}$$

where ρ_{me} is the brake mean effective pressure, ε is the indicated engine efficiency, ρ_{mf} is the fuel mean effective pressure and ρ_{me0} represents all mechanical friction and pumping losses. The Willans approximation was chosen because it provides a low computation model while it models very well real engine behavior. The brake mean effective pressure ρ_{me}, is the pressure that has to act on the piston during one full expansion stroke to produce the same amount of work as the real engine does in two engine revolutions:

$$\rho_{me} = \frac{T_{eng} \cdot N \cdot \pi}{V_d} \tag{5.9}$$

where V_d is the engine's displacement and $N = 4$ for a four-stroke engine. The fuel mean effective pressure ρ_{mf}, is the brake mean effective pressure, that an engine with an efficiency of 100% would produce, by burning the fuel mass m_f per engine cycle, with a fuel lower heating value of H_l ($H_l^{C_8H_{18}}$ is the lower heating of the gasoline and $H_l^{C_2H_5}$ is the lower heating of the ethanol):

$$\rho_{mf} = \frac{m_f \cdot H_l}{V_d} \tag{5.10}$$

The ρ_{me0} represents the engines losses due to friction ρ_{me0f} and gas exchange ρ_{me0g}:

$$\rho_{me0} = \rho_{me0g} + \rho_{me0f} \tag{5.11}$$

The ρ_{me0f} follows a friction model described in [101]:

$$\rho_{me0f} = k_1 \cdot (k_2 + k_3 \cdot S^2 \cdot \omega_{eng}^2)\Pi_{max} \cdot \sqrt{\frac{k_4}{B}} \tag{5.12}$$

where S is the stroke, B is the bore and $k_i \wedge i \in \{1, 2, 3, 4\}$ are constants of the friction model defined in [101]. The maximum boost ratio, Π_{max}, is 1 for naturally aspirated engines.

5.2.3 Fuel Consumption and Emissions Computation

From (5.8-5.12), one can compute the fuel mass flow:

$$\dot{m}_f = m_f \cdot \frac{\omega_{eng}}{N \cdot \pi} \tag{5.13}$$

The total fuel consumed is the summation of the fuel consumption at each time interval Δt, as long as the required F_{trac} is positive. Two types of fuel consumption were analyzed in this work: gasoline and ethanol. The chemical equation for the combustion [102] of gasoline is:

$$C_8H_{18} + 12.5O_2 \longrightarrow 8CO_2 + 9H_2O \tag{5.14}$$

and for ethanol is:

$$C_2H_5OH + 3O_2 \longrightarrow 2CO_2 + 3H_2O \tag{5.15}$$

The burning process of the gasoline and ethanol results in emissions of 8 and 2 molecules of CO_2, respectively. Given the number of CO_2 molecules produced by the combustion and the molar mass of the gasoline ($M_{C_8H_{18}}$), ethanol ($M_{C_2H_5}$) and CO_2 (M_{CO_2}), one can compute the tank to wheel (TTW) CO_2 grams emission for gasoline [103]:

$$TTW^{CO_2}_{C_8H_{18}} = \sum (\dot{m}_f \cdot \Delta t) \cdot \rho_{C_8H_{18}} \cdot 8 \cdot \tfrac{M_{CO_2}}{M_{C_8H_{18}}} \tag{5.16}$$

and for ethanol:

$$TTW^{CO_2}_{C_2H_5} = \sum (\dot{m}_f \cdot \Delta t) \cdot \rho_{C_2H_5} \cdot 2 \cdot \tfrac{M_{CO_2}}{M_{C_2H_5}} \tag{5.17}$$

where $\rho_{C_8H_{18}}$ and $\rho_{C_2H_5}$ are respectively the gasoline and ethanol densities. Using the well-to-tank CO_2 emissions (WTT^{CO_2}) and equations (5.16) and (5.17) one can compute the well-to-wheel CO_2 emissions for gasoline [104, 105]:

$$\begin{aligned} &WTW^{CO_2}_{C_8H_{18}} = TTW^{CO_2}_{C_8H_{18}} + \\ &\sum (\dot{m}_f \cdot \Delta t) \cdot \rho_{C_8H_{18}} \cdot H_l^{C_8H_{18}} \cdot WTT^{CO_2}_{C_8H_{18}} \end{aligned} \tag{5.18}$$

and for ethanol:

$$\begin{aligned} &WTW^{CO_2}_{C_2H_5} = TTW^{CO_2}_{C_2H_5} + \\ &\sum (\dot{m}_f \cdot \Delta t) \cdot \rho_{C_2H_5} \cdot H_l^{C_2H_5} \cdot WTT^{CO_2}_{C_2H_5} \end{aligned} \tag{5.19}$$

The developed algorithm for total fuel consumed and CO_2 emissions is presented in Algorithm 7.

5.3 Results

The developed traffic simulator is highly customizable making it suitable not only for the analysis of ITM algorithms, fuel consumption and CO_2 emissions but as well for other ITS studies (*e.g.* path following controllers evaluation, sensor fusion, etc).

The simulated road layout under study is composed by a roundabout and a crossroad intersection. The traffic is right-handed, *i.e.* vehicles on the RI travel anticlockwise, and no overtaking maneuvers are considered in this study. The RI traffic inflow/outflow arrives from four directions. Each direction is composed of two roads of inflow/outflow and each road has two lanes; the same setup is also applied to the CI. The CI can be controlled by traffic lights, and inside the CI box junction, vehicles are not allowed to stop, *i.e.* a vehicle must not enter the box junction if there is no space for the entire vehicle to exit in the other side of the CI. The simulated vehicles run on gasoline or ethanol and are equipped with a four-stroke otto cycle engine. The modelled engine is equipped with the following two systems: Deceleration Fuel Cut Off (DFCO) and STart-StoP (ST-SP). The DFCO detects if the car is coasting and then cuts fuel to the engine and allows the wheels to keep the engine running, which means that the car doesn't consume any fuel, when the power at the wheels is smaller or equal to zero. The ST-SP system automatically shuts down the engine when the vehicle stops and restarts it when power is required, this system reduces the engine idling time, and thereby reduces the fuel consumption and emissions. The results presented here refer to a 10 min time interval. The **Influx**, represents the vehicle agents to be created according to the Traffic Flow Profile (TFP), *e.g.* in TFP7 a total of $VA_{\text{TFP}} = 1600$ vehicle agents should be created during 10min on a scenario composed of a RI + CI (12 lanes input):

$$\text{TFP7} = 800[v/h/lane] \longmapsto VA_{\text{TFP}} = 800[v/h/lane] \cdot \frac{10[min] \cdot 60[s]}{3600[s]} \cdot 12[lanes] = 1600[vehicles] \quad (5.20)$$

TFP ranging from 100 to 3600 vehicles/hour/lane were used in simulations (see Table 5.2):

TFP# (veh/h/lane)	TFP1 100	TFP2 200	TFP3 300	TFP4 400	TFP5 500	TFP6 600	TFP7 800	TFP8 1000	TFP9 1200	TFP10 1600	TFP11 2000	TFP12 2400	TFP13 3000	TFP14 3600
RI or CI (8 input lanes)														
VA_{TFP}	134	267	400	534	667	800	1067	1334	1600	2134	2667	3200	4000	4800
RI + CI (12 input lanes)														
VA_{TFP}	200	400	600	800	1000	1200	1600	2000	2400	3200	4000	4800	6000	7200

Table 5.2: Number of vehicle agents (VA_{TFP}) that should be created during 10min for each of the TFP and each of the tested scenarios.

Regarding the communication system, the IEEE 802.11a protocol was chosen along with the friis radio propagation model. We assume that all transmissions experience the

Table 5.3: Communication system parameters.

IEEE 802.11a	
RXTX Turnaround= 2 (μ s)	MAC header = 34 (Byte)
SlotTime = 9(μ s)	Max. payload = 2312 (Byte)
CCA = 4(μ s)	RTS = 20 (Byte)
SIFS = 16 (μ s)	CTS = ACK = 14 (Byte)
DIFS = 24 (μ s)	CW$_{min}$ = 15
RX SINR threshold = 10 (dB)	CW$_{max}$ = 1023
TX power = 0.2 (W)	RREQ = RREP = 22 (Byte)
Receiver antenna gain = 1	RREQ timeout = 0.2 (s)
Transmiter antenna gain = 1	RREQ max retries = 3
Frequency = 5(GHz)	Time To Live (TTL) = 7

Table 5.4: Vehicle mechanical specifications.

Parameter		Value	
Vehicle mass	m_{veh}	1500	$[kg]$
Frontal area	A_f	2.0	$[m^2]$
Air drag coeffi.	C_D	0.3	$[-]$
Air density	ρ_{air}	1.2	$[kg/m^3]$
Rolling frict. coeffi.	C_R	0.01	$[-]$
Maximum boost ratio	Π_{max}	1	$[m]$
Gear ratios	γ	(13.8; 7.7; 5.3; 3.9; 3)	$[-]$

Table 5.5: Otto cycle engine specifications.

Parameter		Value	
Displacement	V_d	$1.5 \cdot 10^{-3}$	$[m^3]$
Bore	B	$79.5 \cdot 10^{-3}$	$[m]$
Stroke	S	$80.5 \cdot 10^{-3}$	$[m]$
Friction model	k_1	$1.44 \cdot 10^5$	$[Pa]$
	k_2	0.46	$[-]$
	k_3	$9.1 \cdot 10^{-4}$	$[s^2/m^2]$
	k_4	0.075	$[m]$
Gas exchange losses	ρ_{me0g}	1	$[Pa]$
Engine efficiency	ε	0.4	$[-]$
Deceleration fuel cut off	DFCO	ON	$[-]$
Start-Stop system	ST-SP	ON	$[-]$

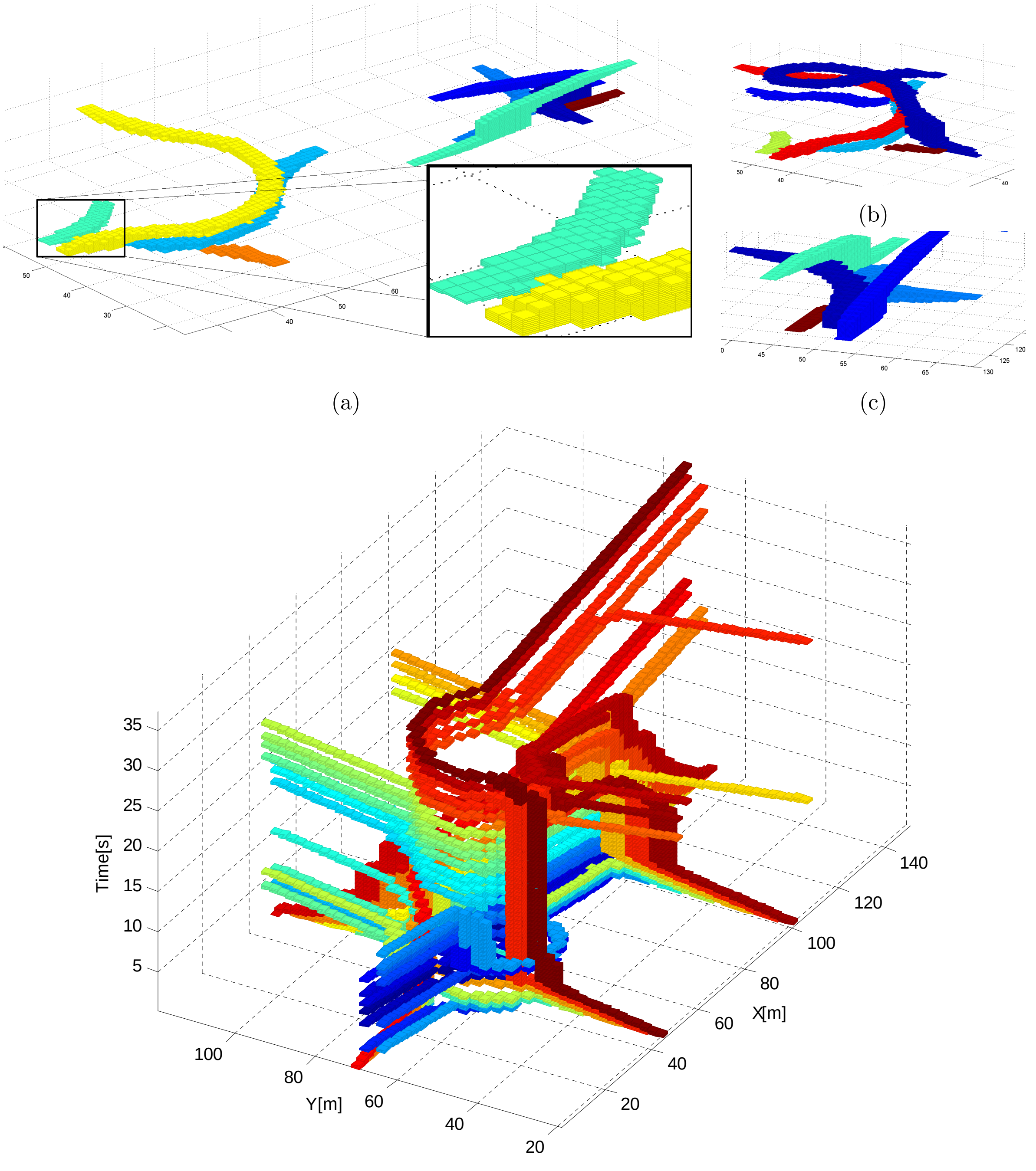
Table 5.6: Fuel Characteristics.

Gasoline			
Parameter		Value	
Density	$\rho_{C_8H_{18}}$	0.75	$[kg/l]$
Lower eating value	$H_l^{C_8H_{18}}$	44.4	$[MJ/kg]$
Molar mass	$M_{C_8H_{18}}$	114.22	$[g/mol]$
Well to tank	$WTT_{C_8H_{18}}^{CO_2}$	12.5	$[gCO_2/MJ]$
Ethanol			
Parameter		Value	
Density	$\rho_{C_2H_5}$	0.79	$[kg/l]$
Lower heating value	$H_l^{C_2H_5}$	28.86	$[MJ/kg]$
Molar mass	$M_{C_2H_5}$	46.07	$[g/mol]$
Well to tank	$WTT_{C_2H_5}^{CO_2}$	48.5	$[gCO_2/MJ]$
Carbon dioxide			
Parameter		Value	
Molar mass	M_{CO_2}	44.01	$[g/mol]$

same path loss versus distance profile, that every node transmits with the same power in the same channel, that the propagation delay is negligible and that each node has the same antenna gain and receiver sensitivity. The CS parameters used in simulation are presented in Table 5.3. The initial CS simulation settings had to be changed, and the TTL (see Table 5.3) had to be reduced otherwise to much requests would overflow the network.

While on STLS mode the fixed time traffic lights are set to: $T_{green} = 15[s]$, $T_{yellow} = 5[s]$ and $T_{red} = 2[s]$. The extra reserved area for safety purposes was set to $A_{safety} = [1.5; 3][m]$. A_{safety} can be increased or decreased, function of the vehicle positioning accuracy (see chapter 6) and of the autonomous path following controller efficiency (see section 4.4.2) or if the vehicle is driven by an human, *i.e.* it is less likely a constraint infringement by a *autonomous vehicle controller* [97] than by a human driver following a speed profile and navigational directions provided by an HCI.

Various simulation runs were conducted in order to quantify the potential benefits in fuel economy of the proposed ITM algorithms. The vehicle model is based on a mid-sized vehicle with mechanical specifications presented in Table 5.4 [100]. The fuel consumption and CO_2 emission results were obtained by simulating a naturally aspirated 1.5L otto



(a)

(b)

(c)

(d)

Figure 5.8: 3D spatio-temporal matrix reservation scheme for the *EMITM* applied to: (a) all scenario both RI and CI, each running its own management system; (b) the roundabout; (c) the crossroad; (d) applied to all scenario both RI and CI, including the link between intersections.

cycle engine equipped with a 5-speed manual transmission. In this work the gear-shift model efficiency and the engine simulation at cold-start were not taken into account. The engine specifications and fuel characteristics are respectively presented in Tables 5.5 and 5.6

In Figure 5.8(a-d) the spatio-temporal matrix reservation for a multiple vehicle cells resource competition is presented. Figure 5.8(b) and (c) show respectively the result of the application of the *EMITM* for the RI and CI while Figure 5.8(a) shows the result of both RI + CI, each intersection using their own independent *EMITM.* Figure 5.8(d) is presented as pictorial view of the EMITM if it was applied to all scenario, including the link between intersections, *i.e.* RI + RI2CI + CI.

The following performance metrics were used to assess ITM performance:

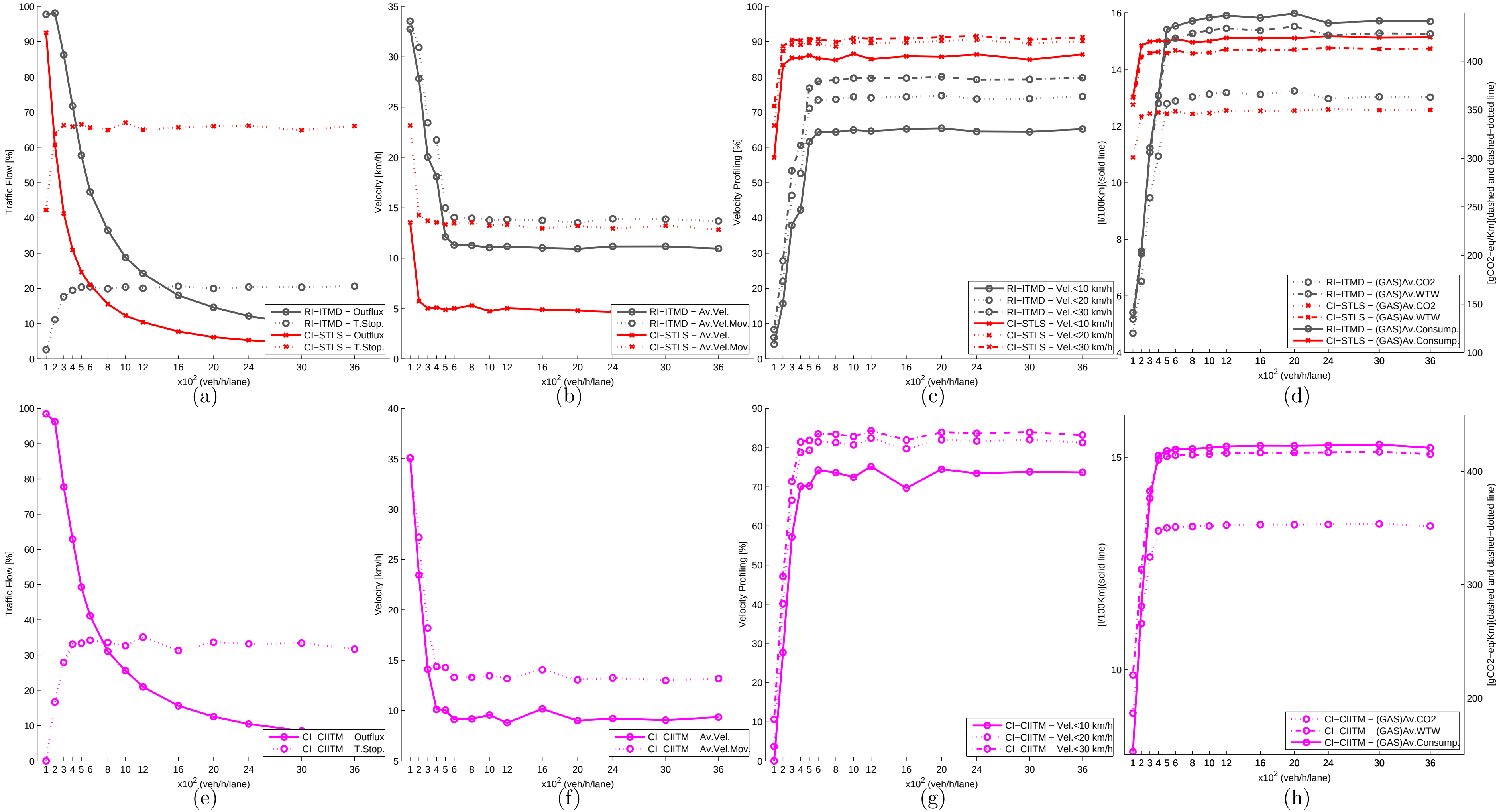
- **Created** : is the number of vehicle agents created during the 10min test;
- **Waiting** : is the number of vehicle agents waiting to be created at the end of 10min;
- **Outflux**[%] : is the % of the total VA_{TFP} vehicles that fully traverse the intersection;
- **T**.**Stop**.[%] : is the average % of time stopped of all vehicles during the traverse of the intersection;
- **T**.**Mov**.[%] : is the average % of time non-stopped of all vehicles during the traverse of the intersection;
- **Av**.**Vel**.[**km/h**] : is the average velocity of all vehicles during the traverse of the intersection;
- **Av**.**Vel**.**Mov**.[**km/h**] : is the average velocity of all vehicles during the traverse of the intersection excluding the time when vehicles are stopped;
- **Vel**. < [**km/h**] : average % of time vehicles spent travelling under 10, 20 and 30$[km/h]$;
- (**GAS**)**Av**.**Consump**.[**l/100km**] and (**ETH**)**Av**.**Consump**.[**l/100km**] : and are the gasoline and ethanol average consumption $[l/100km]$ of all vehicles respectively;
- (**GAS**)**Av**.**CO^2**[**gCO^2/Km**] and (**ETH**)**Av**.**CO^2**[**gCO^2/Km**] : are the average grams of $[CO^2/Km]$ emissions from tank-to-wheel for gasoline and ethanol fueled vehicles respectively;
- (**GAS**)**Av**.**WTW**[**gCO^{2}eq/Km**] and (**ETH**)**Av**.**WTW**[**gCO^{2}eq/Km**] : are the average grams of $[CO^2eq/Km]$ emissions from well-to-wheel for gasoline and ethanol fueled vehicles respectively
- **Conf**.**Solv**. : is the % of cell reservation conflicts, that were solved by the FMITM.

5.3.1 Individual RI and CI performance assessment of non-legacy and top speed limited ITM algorithms

This section presents the results of traditional, non-legacy and top speed limited ITM algorithms, *i.e.* RI-ITMD, CI-STLS, CI-CIITM, WMITM and EMITM, when applied individually to a RI (Figure 4.3(a)) or to a CI (Figure 4.3(b)). The simulation tests where RI is running the *RI-ITMD* and CI is running *CI-STLS* will serve as reference for the others algorithms performance evaluation. All algorithms have a maximum flux capacity also designated by algorithm intersection saturation point, from that point on, the **Outflux** reaches a nearly steady-state profile. The maximum flux capacity, varies for each algorithm and can be easily observed in Figures 5.9 and 5.10. The same applies to all others analyzed metrics.

From Figures 5.9-5.10 it is possible to observe that the CI intersections have better performance in almost all measured metrics, when using spatio-temporal ITM's, but when traditional method are applied to regulate the intersections the RI have better performance. The **Outflux** of the traditional method CI-STLS falls to less than 50% at TFP2 while at the RI-ITMD this only happens on TFP6 (see Figures 5.9(a)). The **Av**.**Vel**. is considerably lower at the CI-STLS than on the RI-ITMD, but the **Av**.**Vel**.**Mov**. has similar values among methods, this fact can be better understood by analysing the **T**.**Stop**.(see Figures 5.9(b-c)). The **T**.**Stop**. is much higher in CI-STLS, due to the time vehicles are held on red traffic light, leading to significant difference on **Av**.**Vel**. of the two methods, but not so relevant on **Av**.**Vel**.**Mov**. where the time stopped is not accounted for the average. The time stopped has direct influence on the % of time that **Vel**. is below each analyzed velocity threshold, namely 10, 20 or 30$[km/h]$, meaning that CI-STLS is most of the time below 10$[km/h]$, 85% of the time compared to around 60% for RI-ITDM. Regarding the fuel consumption (**GAS**)**Av**.**Consump**. and CO^2 emissions (**GAS**)**Av**.**WTW** the CI-STLS is slightly better than RI-ITMD, this contrast with all other analyzed metrics, and the reason for this out-of-trend can be explained by the fact that it was considered that the vehicles are equipped with the *start-stop system* (ST-SP), and therefore when vehicles are stopped no consumption $[l/100km]$ or emission is made $[gCO^2eq/Km]$. Since vehicle in CI-STLS controlled intersections spend considerable **T**.**Stop**. their consumption or emission is therefore lower, otherwise if no ST-SP were considered, it would be higher than the RI-ITMD (see Figure 5.9(d)).

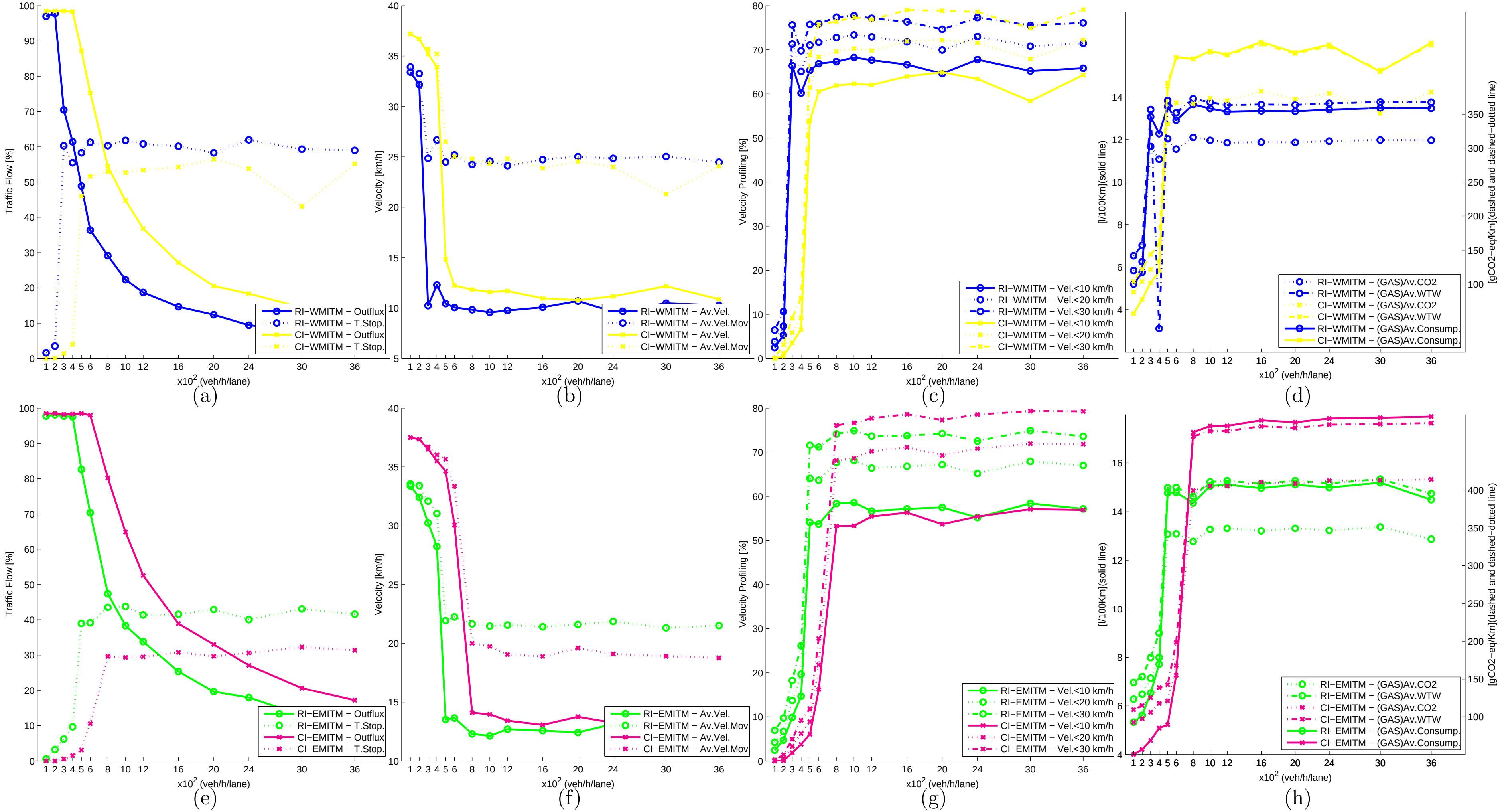
The results of algorithms using spatio-temporal ITM's are presented in Figure 5.9(e-h) and in Figure 5.10(a-h). In almost all measured metrics, when using spatio-temporal ITM's the CI is either better or equal to the RI.



(a) (b) (c) (d)

(e) (f) (g) (h)

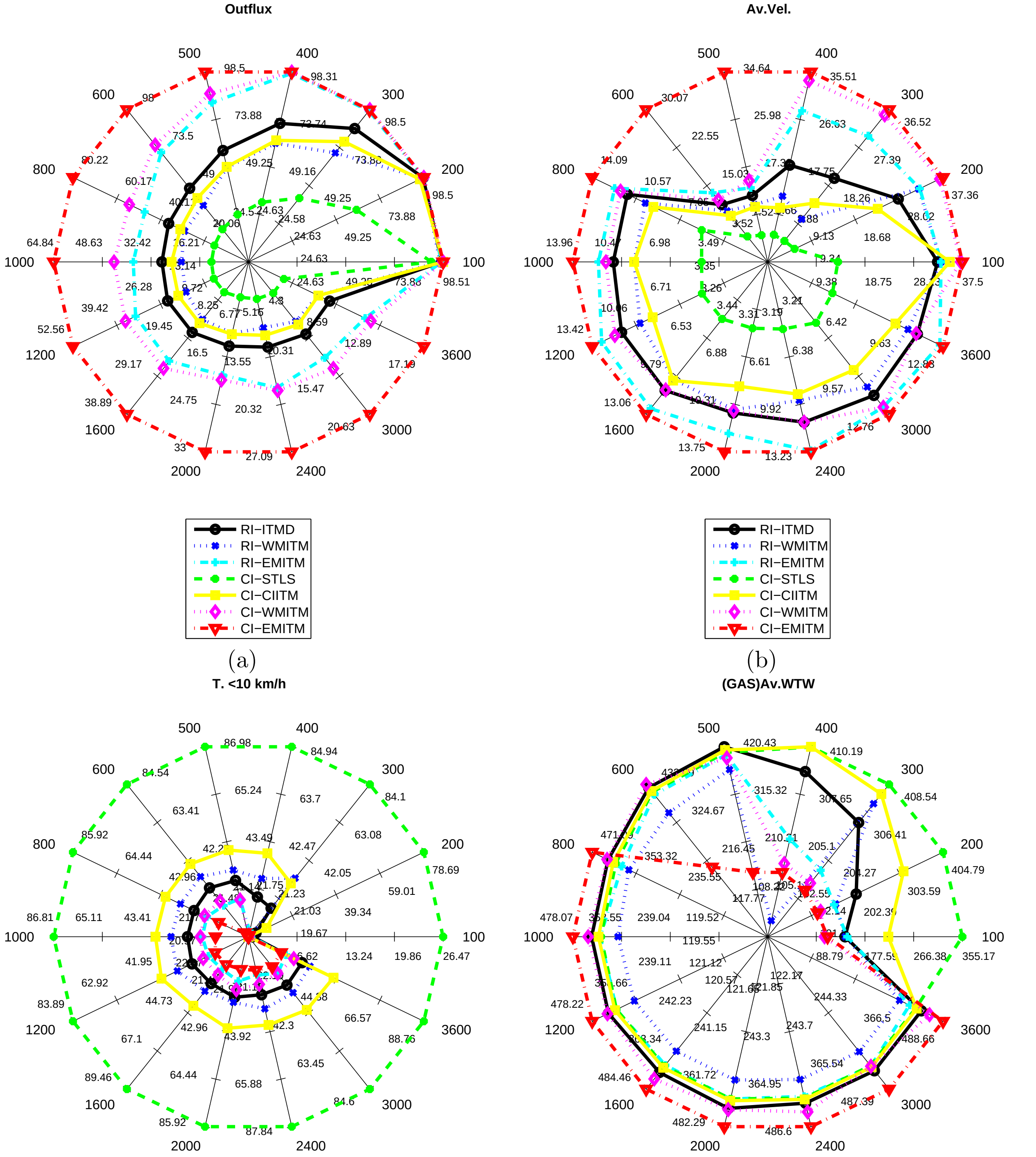
Figure 5.9: RI-ITMD, CI-STLS and CI-CIITM results for all TFP's applied to an intersection composed of a RI or a CI: (a-d) RI-ITMD and CI-STLS; (e-h) CI-CIITM. The abscissa axis represent TFP ranging 100 to 3600 vehicles/hour/lane.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.10: WMITM and EMITM results for all TFP's applied to an intersection composed of a RI or a CI: (a-d) RI-WMITM and CI-WMITM; (e-h) RI-EMITM and CI-EMITM. The abscissa axis represent TFP ranging 100 to 3600 vehicles/hour/lane.



(a) (b) (c) (d)

Figure 5.11: Non-legacy and top speed limited ITM results for an intersection composed of a RI or a CI: (a) $Outflux[\%]$; (b) $Av.Vel.[km/h]$; (c) $T. < 10km/h[s]$; (d) $(GAS)Av.WTW[gCO^2eq/Km]$. Each axis line (starting in the center and extending to the periphery of the circle) represent TFP ranging 100 to 3600 vehicles/hour/lane. For better graph readability each axis line has different quantitative scales, *e.g.* on subplot (a) the TFP1 (100veh/h/lane) scale, ranges from 0% to 98.51%, while for the TFP14 (3600veh/h/lane) scale the range is from 0% to 17.19%

Figure 5.9(e-h) presents the results of the CI-CIITM, where a small improvement from the traditional CI traffic regulating method (CI-STLS) can be observed. Although the use of the CI-CIITM represented an improvement to the CI-STLS it still falls behind the performance of the traditional RI traffic regulating method (RI-ITMD). The RI-ITMD compared to CI-CIITM as another advantage regarding the investment on infrastructure equipment, *i.e.* the CI-CIITM requires a computer vision sytem and induction loops to be equipped on installed on the infrastructure while the RI-ITMD it does not require any special equipment.

Figure 5.10(a-d) presents the results of the WMITM when applied to the RI (RI-WMITM) and to the CI (CI-WMITM), whereas Figure 5.10(e-h) presents the results of the EMITM when applied to the RI (RI-EMITM) and to the CI (CI-EMITM). Both WMITM and EMITM achieve a considerable improvement in all metrics except in fuel consumption $(\mathbf{GAS})\mathbf{Av}.\mathbf{Consump}.$ and CO^2 emissions $(\mathbf{GAS})\mathbf{Av}.\mathbf{WTW}$.

The overall comparative evaluation of all traditional methods and non-legacy and top speed limited ITM algorithms can be better accessed in Figure 5.11. Figure 5.11 presents a subset of relevant analysis metrics, namely the **Outflux** (Figure 5.11(a)), the $\mathbf{AV}.\mathbf{Vel}.$ (Figure 5.11(b)), $\mathbf{Vel}. < \mathbf{10[km/h]}$ (Figure 5.11(c)) and $(\mathbf{GAS})\mathbf{Av}.\mathbf{WTW}$ (Figure 5.11(d)). Each axis line, starting in the center and extending to the periphery of the circle, represent a TFP, ranging 100 to 3600 vehicles/hour/lane. For better graph readability each axis line has different quantitative scales, and the scale is dynamically adjusted so that the max scale correspond to the max displayed value, *e.g.* on subplot (a) of Figure 5.11 the TFP1 (100veh/h/lane) scale, ranges from 0% to 98.51%, while for the TFP14 (3600veh/h/lane) the scale range is from 0% to 17.19%.

The best performing algorithm on the **Outflux** metric is the CI-EMITM (see Figure 5.11(a)), from TFP7 (800veh/h/lane) its performance is around 1/4 better than the second best algorithm (CI-WMITM). The second best algorithm is not from RI-EMITM but again from a CI, *i.e.* the CI-WMITM, but not by a large margin. The RI-EMITM is the third best and their results are similar to the CI-WMITM. Surprisingly the traditional method for traffic regulating inside (RI-ITMD) is the forth best method in-front of algorithms using spatio-temporal ITM's, such as RI-WMITM and CI-CIITM. The RI-ITMD compared to spatio-temporal ITM's as an advantage of not requiring additional infrastructure investment. Although RI-WMITM and CI-CIITM have identical performances, the RI-WMITM requires additional investment in both infrastructure and vehicles.

Regarding the performance in the $\mathbf{AV}.\mathbf{Vel}.$ (Figure 5.11(b)) the profile analysis follows the same pattern as in the **Outflux**, except that the second best algorithm is switched from CI-WMITM to RI-EMITM. The RI-EMITM in steady state achieves a better result than the CI-WMITM. The differences among algorithms are not so evident as in the **Outflux**, except for the traditional CI traffic regulating method (CI-STLS). The failure to provide a reasonable performance in the **AV**.**Vel**. by the CI-STLS is well depicted in Figure 5.11(c). The CI-STLS spends most of the time with a **Vel**. $<$ **10**[**km**/**h**], thus making it difficult to analyze all other algorithms, due to the huge gap between this algorithm and all the other. Nevertheless a careful analysis reveal the same pattern as in **AV**.**Vel**..

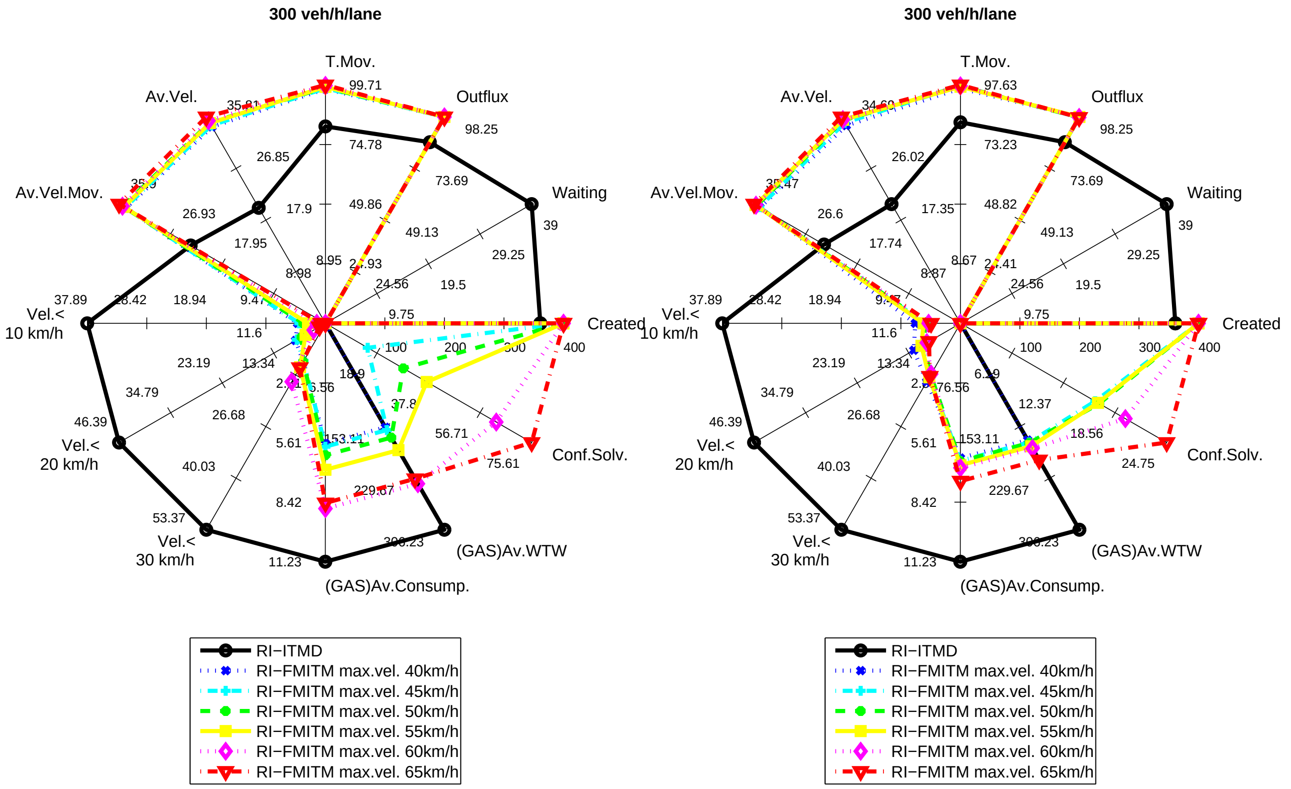
The CO^2 emissions (**GAS**)**Av**.**WTW** (Figure 5.11(d)), reveal an out of trend behaviour, meaning that:

- the CI-EMITM is the worst performing algorithm from TFP7 (800veh/h/lane);
- the CI-WMITM is the worst performing algorithm at TFP6 (600veh/h/lane) and the second worst from TFP7;
- the RI-EMITM is the third best performing algorithm from TFP5 (500veh/h/lane) and the second best from TFP7;
- the RI-WMITM is the second best performing algorithm from TFP5 (500veh/h/lane) and the best from TFP7;
- the CI-CIITM is performing worst than the CI-STLS from TFP4 (400veh/h/lane);
- the RI-ITMD is performing worst than the CI-STLS from TFP5 (500veh/h/lane);
- the CI-STLS is no longer the worst performing algorithm in all TFP's, and it is CI best performing CI algorithm from TFP6.

From TFP7 (800veh/h/lane) the best performing CI algorithm on the **Outflux** metric (CI-EMITM) is 1/6 to 1/5 more pollutant than the best RI algorithm (RI-EMITM). It is worth noting that, for each algorithm, before the **Outflux** starts to fall steeply the CO^2 emissions (**GAS**)**Av**.**WTW** follow the same grading pattern as with the **Outflux**.

5.3.2 Forward Method for Intelligent Traffic Management (FMITM)

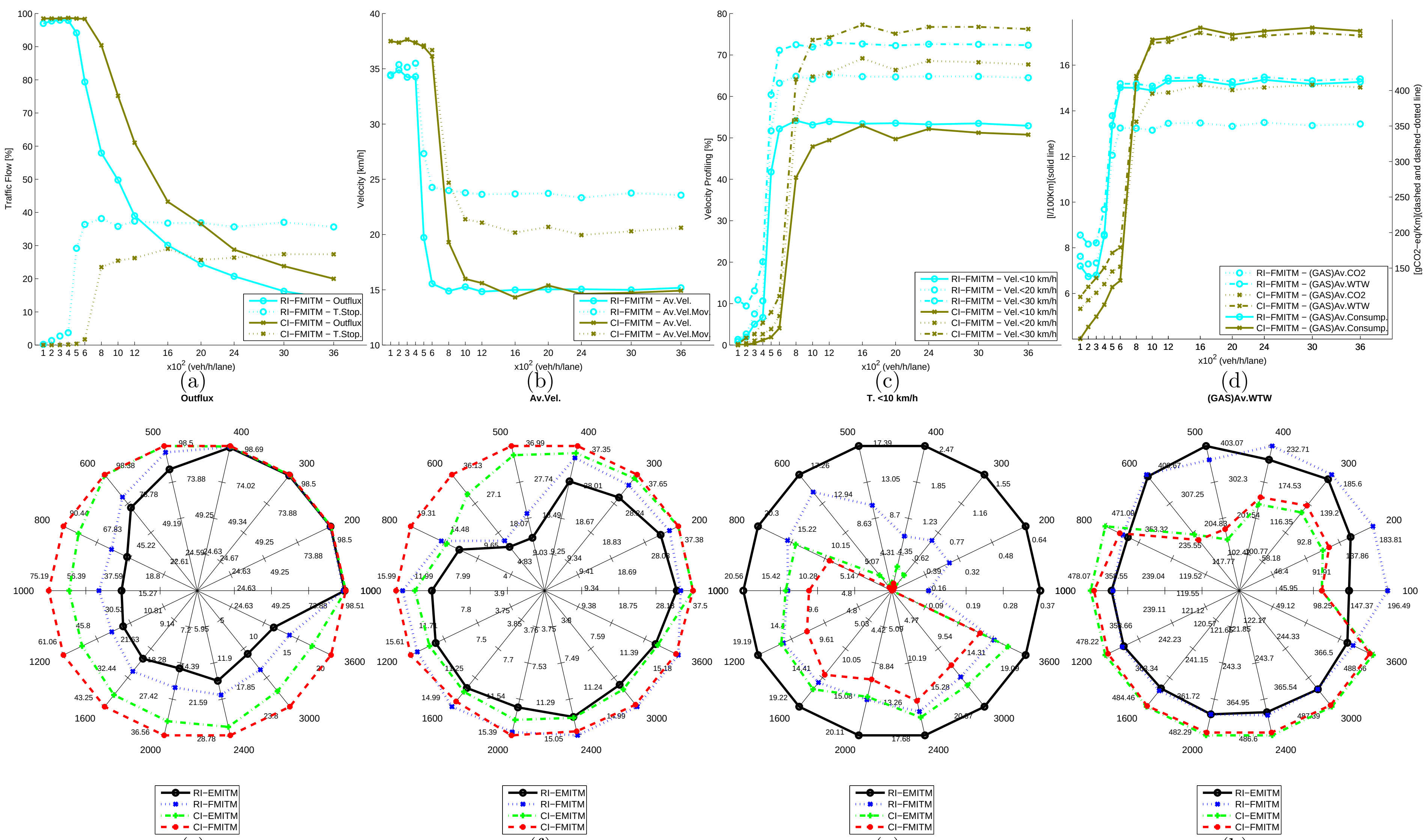
This section presents the results of the FMITM, compared to traditional methods and to the globally best performing algorithms in both CI and RI. When reserving the *spatio-temporal matrix* using the vehicle origin-destination setup and the preferred speed, each cell is tested to verify if it is reserved or free. If the cell is already reserved it is signalized a conflict situation. The number of conflict situation solved (**Conf**.**Solv**.) is used as a performance metric for parameter selection of the FMITM. The parametrization variables are the safety margin when booking the space-time matrix (A_{SM}) and the maximum speed that a vehicle can reach to avoid the collision (V_{MOLS}).



(a) (b)

Figure 5.12: Maximum speed allowed and safety margin performance evaluation of FMITM for an intersection composed of a RI: (a) safety margin $A_{SM} = [1; 2.5][m]$; (b) safety margin $A_{SM} = [1; 5][m]$. The tests above the scenario setup legal maximum velocity, comprise $V_{MOLS} = \{40, 45, 50, 55, 60, 65\}[km/h]$. For better graph readability each axis line has different quantitative scales.

A set of tests using different values of both A_{SM} and V_{MOLS} were performed, in order to access the suitable over the legal limit maximum speed V_{MOLS} and a proper safety margin A_{SM} for this new maximum speed. The cells area, reserved in the spatio-temporal matrix, are wider than the vehicle dimensions, this safety margin A_{SM} ($A_{SM} = [wide; long][m]$) is used to avoid accidents and should be increased as V_{MOLS} increases. The A_{SM} was tested using two setups: $A_{SM} = [1; 2.5][m]$ and $A_{SM} = [1; 5][m]$. The V_{MOLS} was tested with speed ranging from $V_{MOLS} = 40km/h$ to $V_{MOLS} = 65km/h$.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.13: Results of FMITM: (a-d) performance analysis on a intersection composed of a RI or a CI; FMITM comparison with eMITM for all TFP's applied to an intersection composed of a RI or a CI: (e) $Outflux[\%]$; (f) $Av.Vel.[km/h]$; (g) $T. < 10[km/h]$; (h) $(GAS)Av.WTW[gCO^2eq/Km]$.

Figure 5.12 presents a subset of the results of the multiple simulations, used to fine tune A_{SM} and V_{MOLS}, in order to obtain the maximum performance from the FMITM algorithm. The simulations presented were performed using a TFP3 for an RI intersection. As expected when increasing the V_{MOLS} the number of **Conf**.**Solv**. is greater, the **Av**.**Vel**. is increased and the vehicles spend less time travelling with a **Vel**. < **10**[**km**/**h**] and with a **Vel**. < **20**[**km**/**h**]. The increase of the V_{MOLS} leads to a increase of CO^2 emissions, being more significant with a $V_{MOLS} = \{60, 65\}[km/h]$. When reserving a small area $A_{SM} = [1; 2.5][m]$ (Figure 5.12(a)) the number of **Conf**.**Solv**. is greater but it represents an added collision risk. If A_{SM} is small, the true area occupied by the vehicle is less likely to be bounded inside the reserved A_{SM} area, when an inaccurate vehicle pose estimation or a less efficient *path following controller* event occurs.

When being more conservative and reserving a bigger area $A_{SM} = [1; 5][m]$ (Figure 5.12(b)), the number of **Conf**.**Solv**., *w.r.t.* $A_{SM} = [1; 2.5][m]$, is significantly lower but it represents a lower CO^2 emissions and reduced collision risk.

As a algorithm designer option, it was chosen that the V_{MOLS} should not exceed two times the specified legal velocity limits inside intersections and that there should not occur a significant subjective collision risk increase *w.r.t.* other implemented algorithms. Therefore the maximum speed that a vehicle can reach to avoid the collision was set to $V_{MOLS} = 60[km/h]$ and the safety margin when booking the space-time matrix was set to $A_{SM} = [1; 5][m]$, this choice also fulfills one requirement of not increasing significantly the the CO^2 emissions.

Figure 5.13 presents a comparative analysis between the FMITM and EMITM algorithm for both RI and CI intersections. As expected, for both CI and RI, the FMITM has a better record in most of the analyzed metrics, losing ground to the EMITM on the CO^2 emissions, but only when it is reached the intersection saturation for the FMITM algorithm. Regarding the performance of the CI-FMITM *w.r.t.* RI-FMITM, the CI-FMITM is globally better, except when it is reached the intersection saturation for the CI-FMITM algorithm, from that point on the **AV**.**Vel**. and the (**GAS**)**Av**.**WTW** have better results using the RI-FMITM.

5.3.3 RI and CI combined performance assessment of non-legacy and speed limited compliant ITM algorithms

The ITM algorithms were extensively simulated trough a diverse combination of simulation tests on the scenario composed of a RI and a CI, as depicted in Figure 4.3(c). A subset of the tests performed with the ITMD, STLS, WMITM and EMITM, are characterized in Table 5.7 and presented in Figure 5.15. This section does not include non-legacy

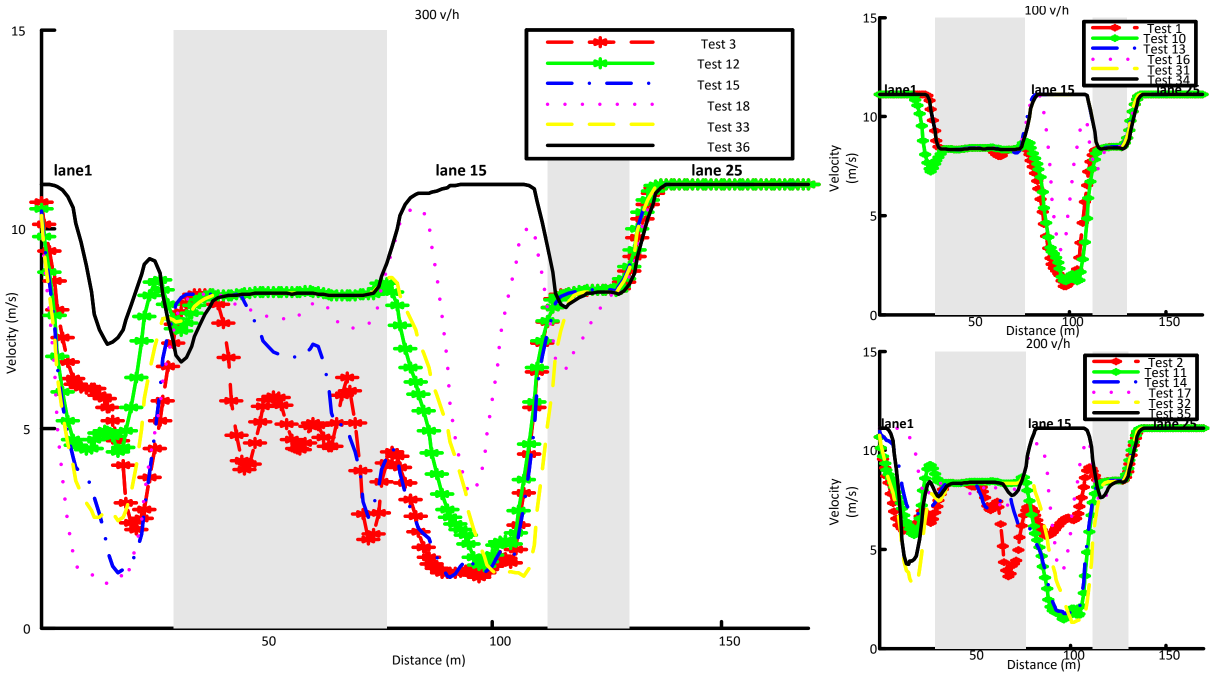


Figure 5.14: Mean speed profile vs distance travelled of vehicles entering on lane 1 and exiting on lane 25 (the path between lane 1 and 25 passing through 15 is marked on Figure 5.16). The marked gray area at the distance travelled [30-70] and [110-130] represent the RI and CI respectively. Mean speed profile for both TFP1, TFP2 and TFP3 is respectively on the left, center and right sub-plot. The ITM used for Test i, where $i = 1..18$, are presented on Table 5.7.

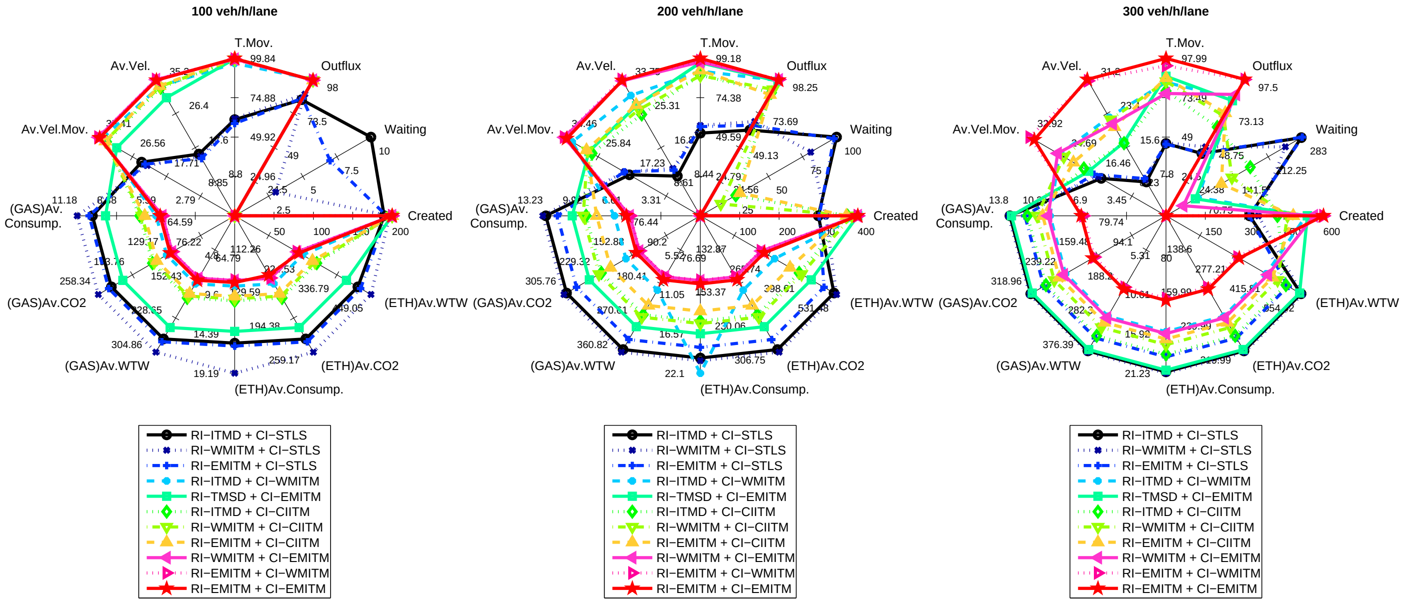
algorithm (LEMITM) and non-speed limit compliant algorithm (FMITM) due to their specific nature. The most important results, observed in each simulation experiment of 10 min, are summarized in Figure 5.15.

Table 5.7: Simulations tests catalog for each ITM, according to the traffic inflow for each of the 12 input lanes (RI and CI).

TFP#	Case-study (RI Algorithm + CI algorithm)											
RI + CI	ITMD + STLS	WMITM + STLS	ITMD + WMITM	EMITM + STLS	ITMD + CIITM	ITMD + EMITM	WMITM + CIITM	WMITM + WMITM	WMITM + EMITM	EMITM + WMITM	EMITM + CIITM	EMITM + EMITM
TFP1	test 1	test 4	test 7	test 10	test 13	test 16	test 19	test 22	test 25	test 28	test 31	test 34
TFP2	test 2	test 5	test 8	test 11	test 14	test 17	test 20	test 23	test 26	test 29	test 32	test 35
TFP3	test 3	test 6	test 9	test 12	test 15	test 18	test 21	test 24	test 27	test 30	test 33	test 36

The speed limit on the whole scenario was set at $11.1m/s$ ($40km/h$), except while driving inside the RI and CI, where the speed limit was set at $8.3m/s$ ($30km/h$). These limits are well depicted in Figure 5.14, where inside intersections (gray areas) the speed of vehicles falls to at least $8.3m/s$, while outside intersections the vehicle speed may rise to $11.1m/s$.

The **Outflux** results of tests 1-6 and tests 10-12, where STLS method was used at the CI, are significantly lower than results of tests 7-9 and tests 13-27, where an ITM was used at the CI. The results show that the maximum number of vehicles traversing two-intersections scenario is limited by the performance of the CI. The previous hampering traffic effect is easily seen in Figure 5.14, where the speed profile for tests 1-6 is severely affected, even when a low traffic density is applied.

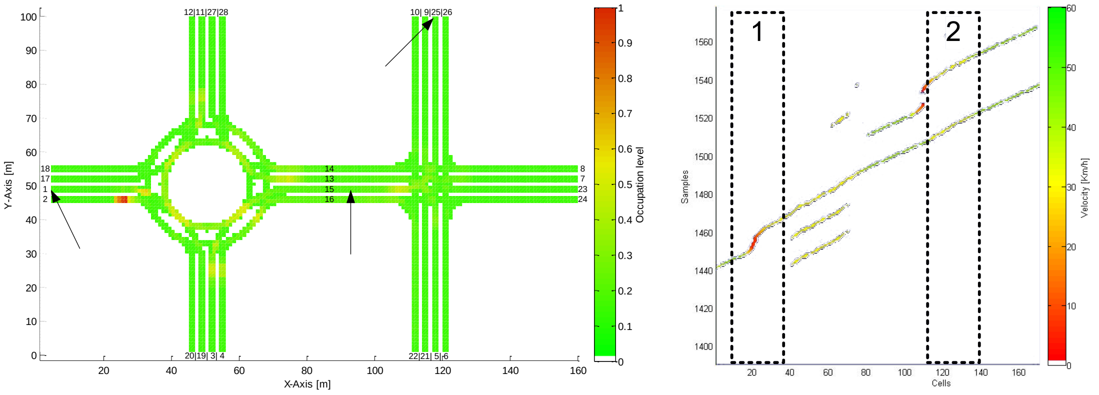


(a) (b) (c)

Figure 5.15: Non-legacy and speed limited compliant ITM results, applied to two intersections composed of a RI + CI (12 input lanes): (a) TFP1 (100/veh(h/lane) ; (b) TFP2 (300/veh(h/lane); (c) TFP3 (300/veh(h/lane).

The number of vehicles in waiting status increases with the increase of the traffic flow except when an ITM is applied in any one of the intersections; this effect can be easily seen on Figure 5.15 where the **Waiting** percentage is lower when using an ITM. The increase of the number of vehicles traversing the intersections occurs not only when the ITM systems are working together but as well if one of them is working exclusively. The performance improvement of any of the used ITM algorithms is more noticeable when the traffic flux per lane is higher.

The simultaneous use of the EMITM in both intersections revealed to be the best combination, in all analyzed metrics, when compared with other combinations of ITM algorithms. For example, this combination yields the largest number of vehicles crossing intersections and the lowest cumulative time stops. Figure 5.14 depicts very well this performance improvement, the speed profile is close to the speed limit for tests 34-36, consequently the *mean speed* is higher. Figure 5.16(a) shows the level of occupancy of cells concerning the simulation test 36, when both intersections are regulated by the EMITM. Although in this test the traffic inflow per lane is 300 vehicles/lane/h, the occupation level is mild and increases slightly only on approaching and inside of both intersections. This



(a)

(b)

Figure 5.16: Simulation scenario layout (RI and CI): (a) cell occupancy level for simulation Test 36. Higher occupancy levels occur mainly on approaching and inside intersections; (b) average speed of vehicles of path between lane 1 and 25 passing through lane 15, the marked area 1 and 2 represent the RI and CI respectively. This path is also used to study the vehicles speed profile on Figure 5.14.

effect is compliant with the information presented in Figure 5.14, whereas the speed profile is moderately reduced before entering the RI and has a low reduction before entering the CI.

Figure 5.16(b) represents the average speed of vehicles traveling path between lane 1 and 25 passing through lane 15 crossing the RI area (marked area 1) and the CI area (marked area 2). The vehicle entering on lane 1 slows down when arriving to the RI but it is little affected by the CI almost maintaining its speed. A vehicle coming from a lane other than the link between RI and CI is highly disturbed on its speed having to stop at the entrance of the CI, this disturbance favors overall vehicles flow through both intersections.

5.3.4 Legacy Early Method Intelligent Traffic Management (LEMITM)

Figure 5.18 presents the results of the tests for three TFP simulations and nine different % of vehicles that are not equipped or with faulty V2V and V2I CS. Each test has the same duration of 10 min. The reservation procedure was setup with a safety margin area of $A_{SM} = [2; 10][m]$, a speed safety margin of $V_{SM} = [25; 55][km/h]$ and a sampling period of $\delta t = 0.1[s]$.

The LEMITM algorithm reserves the space for all possible trajectories, activating the green light when the reservation is successful. The reserved space is sequentially freed as time passes, after permission was given to enter the intersection, i.e. green light is on. In Figure 5.17 it is possible to observe the *reservation of all trajectories*, for one vehicle at each

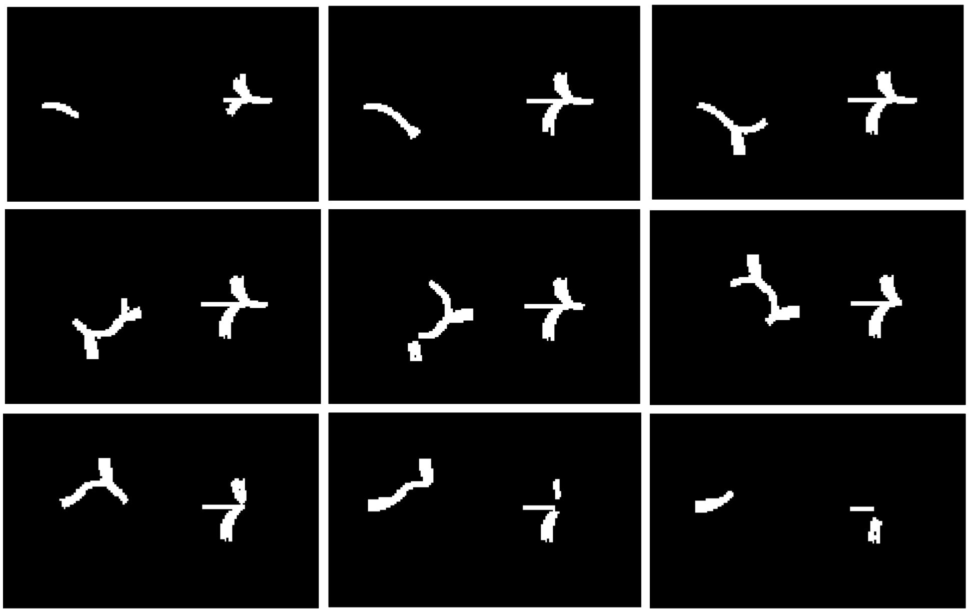
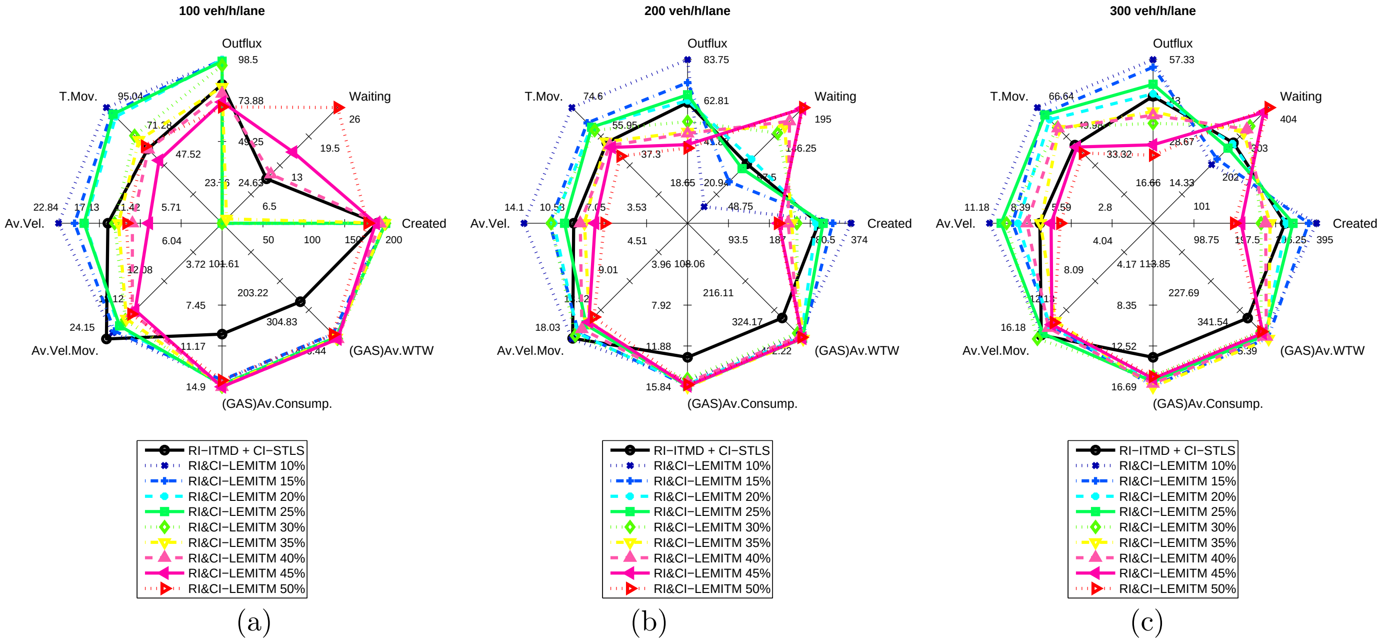


Figure 5.17: Spatio-temporal cell reservation using LEMITM algorithm for one vehicle at each intersection (see Figure 5.6). The LEMITM algorithm reserves the space for all possible trajectories, setting green light when the reservation is successful. The reserved space is sequentially freed as time passes after green light is on. Time increases from left-right top-bottom.



(a) (b) (c)

Figure 5.18: LEMITM results, applied to two intersections composed of a RI + CI (12 input lanes): (a) traffic flow profile 1 (100/veh(h/lane) ; (b) traffic flow profile 2 (300/veh(h/lane); (c) traffic flow profile 3 (300/veh(h/lane).

intersection. Results show that the LEMITM performance is lower than EMITM. The loss of efficiency was expected since at least a % of vehicles (vehicles without V2V and V2I CS) reserve all cells in all possible trajectories. This scheme leads to the reservation of space blocks that are not used; it is the payback of allowing vehicles without communications to go through the intersections. For the specific case in which % of vehicles that are not equipped or with faulty V2V and V2I CS is below 20%, although the reduction in efficiency occurs when the traffic flow increases, the LEMITM algorithm provides the best results, in almost all analyzed metrics, in comparison to other traditional methods of traffic control.

5.4 Final Remarks

In this chapter it was presented a spatio-temporal reservation based technique to reduce traffic congestion and to allow vehicles to traverse intelligent intersections. The spatio-temporal reservation scheme shows a better performance than the traditional control schemes: less time in crossing intersections with shorter stops or even no need of vehicle stops. Globally the CI has a better performance than the RI, but it is also more prone to accidents [5], which in the context of spatio-temporal reservation means that if vehicles do not follow the velocity profile an hazardous event is likely to occur and the risk of fatalities is higher.

The FMITM proved to be the best performing ITM algorithm in terms of traffic flow, but it was also the most pollutant method. The EMITM revealed to be a good method for ITM and it was observed a significant improvement on the RI traffic flow. Simulation results revealed that when using the EMITM algorithm, vehicles can traverse the intersections with short stops, or even without stoppages, and crossing time is significantly smaller in comparison with other methods.

As expected, the LEMITM has a lower efficiency than EMITM, but with its application is possible to reduce the time in crossing intersections, in comparison with the traditional techniques of traffic control. The results show that the LEMITM performance worsens with increasing percentage of vehicles that are not equipped or with faulty V2V and V2I CS. The degrading performance is due to the fact that these vehicles reserve cells in the *spatio-temporal matrix* for all possible paths, meaning that there are blocks in the *spatio-temporal matrix* that are never used. When the percentage of vehicles that are not equipped or with faulty V2V and V2I CS is low, the LEMITM performance is higher than the traditional method except for the CO^2 emissions.

Chapter 6

Vehicle Pose Estimator

Contents

The algorithms used in the Vehicle Pose Estimator (VPE) for autonomous vehicles navigation in semi-structured outdoor environments, are described in this chapter. The VPE provides estimates of both the position and orientation of car-like vehicles, using multiple sensors. The presented solution enable each vehicle to have different combination of sensors. The proposed sensor fusion algorithms, were tested using ISR-TRAFSIM simulator. As described in Chapter 4, the ISR-TRAFSIM is an agent based simulator, each agent incorporates a group of modules, were each module is responsible by a self-contained specific function (see Figure 6.1).

The vehicle agent is composed by a PFC and a Sensor Fusion and Reasoning (SFR) module. The PFC is made up of two main modules: the VP and the LC . The VP receives a reference path and speed profile from the traffic management system and according to the VPE estimated pose provides a reference velocity $[v_{ref}]$. A fuzzy-logic based lateral controller is compared with a canonical one and the second uses chained form theory, they provide both steering and velocity commands $[\varphi_c, v_c]$.

The SFR comprises the following sensors:

i four wheel encoders' measurements Δ_{RR}, Δ_{RL}, Δ_{FR}, Δ_{FL} and the steering wheel encoder γ. These sensors provide redundant data when computing the pose of an Ackermann steered vehicle;

ii MSS composed by a set of magnetic markers detectors. The simulated sensors are triggered when they cross over a magnetic landmark. This sensing system provides the distance d_m from the x axis of the vehicle to the triggered sensor along the y axis of the vehicle;

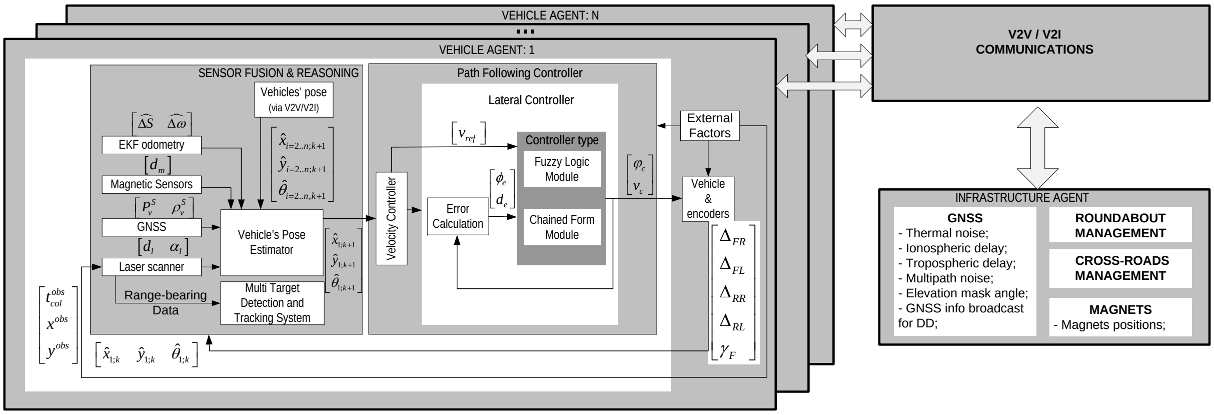


Figure 6.1: Vehicle Pose Estimator and Sensor Fusion & Reasoning modules integration in the overall Simulator architecture.

iii LIDAR this sensor provides all range-bearing data to the MTDT and the average range-bearing pair $[d_l, \lambda_l]$ of a preceding detected vehicle used as a landmark.

iv GNSS: the GPS constellation is simulated as well as disturbances, such as the tropospheric delay, ionospheric delay, multipath noise, thermal noise and clock biases.

- the VPE receives its own GPS simulated receiver code and carrier pseudoranges for all visible satellites $[P_v^S \rho_v^S]$;
- if in the range of the V2V, the third party vehicle (i=2..n) current (k) estimated pose $[\hat{x}_{i=2..n;k}, \hat{y}_{i=2..n;k}, \hat{\theta}_{i=2..n;k}]$ and its GPS simulated receiver code and carrier pseudoranges for all visible satellites $[P_v^S \rho_v^S]$ are also provided to the VPE.
- if in the range of the V2I, the Master Antenna Fixed Station (MAFS) position $[x_m, y_m]$ and its GPS simulated receiver code and carrier pseudoranges for all visible satellites $[P_m^S \rho_m^S]$ are also provided to the VPE.

The VPE comprises two main methodologies to compute the vehicle's pose:

- Enhanced odometry and absolute positioning data fusion in section (described in 6.1). In a initial fusion stage, data from the four wheel encoders and the steering encoder is fused by means of an EKF, providing robust odometric information, namely in face of undesirable effects of wheels slippage, *i.e.* the redundant four wheel encoders' measurements Δ_{RR}, Δ_{RL}, Δ_{FR}, Δ_{FL} and steering wheel encoder γ data, is used in an EKF to produce better estimates of $\widehat{\Delta_R}$ and $\widehat{\omega_R}$.

  Next, a second fusion stage is processed for the integration of the odometric estimates $([\widehat{\Delta_R}, \widehat{\omega_R}])$ and absolute positioning data provided by landmarks. Landmarks can either be provided by a LIDAR a front laser detected vehicle or natural feature, or by a MSS detected magnetic marker.

- Inter-vehicle pose estimation using laserscanner data and magnetic landmark via V2X (described in section 6.2). An initial approach uses set membership algorithm to compute the absolute vehicle position and an estimation of the satellite-specific errors by using raw GNSS pseudoranges, lane boundary measurements and a 2D georeferenced road map.

  Next, using the cross-track corrections in a cooperative estimation process enables the vehicles to improve their own estimates whatever the orientation of the roads.

  Finally it is applied a removal of NLOS multipath pseudoranges contaminated satellites, at the level of each individual vehicle, using a guaranteed risk integrity zone computation algorithms, for robust vehicle positioning, constrained by geo-referenced lane boundary measurements and GNSS pseudoranges.

6.1 Enhanced Odometry and absolute positioning data fusion

6.1.1 Kinematic sensors Fusion with Confidence Tests (KF-CT)

The majority of systematic errors and slippage (or high slippage) problem, associated to the odometry relying only on encoders, can be reduced by applying encoders confidence tests followed by an EKF estimation for the arc length Δ_R and the elementary rotation ω_R; this fusion method is designated by Kinematic sensors Fusion with Confidence Tests (KF-CT).

The KF-CT algorithm makes use of wheel encoders' measurements Δ_{RR}, Δ_{RL}, Δ_{FR}, Δ_{FL} and the steering wheel encoder γ to estimate the arc length $\widehat{\Delta_R}$ and the elementary rotation $\widehat{\omega_R}$ by means of a EKF and by applying encoders confidence tests.

For a 2WS vehicle, its odometry model is based only on the rear wheels encoders (see equations B.23 and B.24). Using also the front wheels encoders, redundant data become available, which can be used to produce better estimates of Δ_R and ω_R. The steering angle of left and right front wheels can be expressed by

$$\varphi_L = \arctan\left(\frac{\tan(\varphi).L}{L - e.\tan(\varphi)}\right) \tag{6.1}$$

$$\varphi_R = \arctan\left(\frac{\tan(\varphi).L}{L + e.\tan(\varphi)}\right) \tag{6.2}$$

where φ is the front steering angle of the virtual front wheel and L is the car length (distance between rear and front axles).

From (6.1) and (6.2) and knowing that $\Delta_R = \rho.\omega_R$ (where ρ is the curvature radius

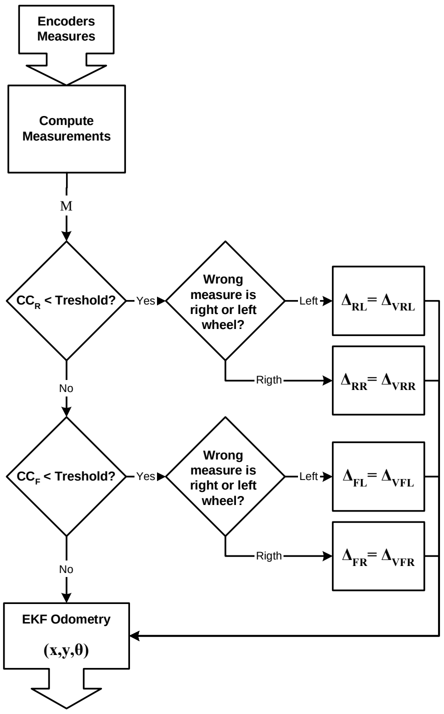


Figure 6.2: EKF-CT diagram block algorithm (M is the vector $(\varphi, \Delta_{RR}, \Delta_{RL}, \Delta_{FR}, \Delta_{FL})$).

of the rear axle center), a set of equations can be established which relates the encoders measurements (from each of the four wheels and steering) with the parameters Δ_R and ω_R [106]:

$$\begin{cases} \tan(\varphi) = L.\frac{\omega_R}{\Delta_R} \\ \Delta_{RL} = \Delta_R - e.\omega \\ \Delta_{RR} = \Delta_R + e.\omega \\ \Delta_{FL}.\cos(\varphi_L) = \Delta_R - e.\omega \\ \Delta_{FR}.\cos(\varphi_R) = \Delta_R + e.\omega \end{cases} \tag{6.3}$$

Let $\mathbf{x} = [\Delta_R, \omega_R]^T$ and $\mathbf{y} = [\tan(\varphi), \Delta_{RL}, \Delta_{RR}, \Delta_{FL}.\cos(\varphi_L), \Delta_{FR}.\cos(\varphi_R)]^T$ be the state vector and the measurement vector, respectively. Since the measures are related with the state vector in a non-linear way, an EKF is used to estimate the state vector.

1. *System Model:* is modeled by the stochastic process
   $\mathbf{x}_k = \mathbf{x}_{k-1} + \gamma_{k-1}$, where γ_{k-1} is a zero-mean white noise with covariance matrix $\mathbf{Q}$.

2. *Measurement Model:* from (6.3), the measurement model is defined by the non-linear equation
   $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k$
   where $\mathbf{v}_k$ is a zero-mean white noise with covariance matrix $\mathbf{R}$, known as measurement

noise.

3. *EKF Algorithm:* the future state of the system $\hat{\mathbf{x}}_k^-$, and the future state error covariance matrix, $\mathbf{P}_k^-$, are predicted using the time update equations:
$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_{k-1}\\ \mathbf{P}_k^- &= \mathbf{P}_{k-1} + \mathbf{Q}\end{aligned} \tag{6.4}$$
With the measurements $\mathbf{y}_k$ a correction stage is done. The EKF gain matrix $\mathbf{K}_k$, the state estimate $\hat{\mathbf{x}}_k$, and the state error covariance matrix for the updated state estimate, $\mathbf{P}_k$, are computed, as follows:
$$\begin{aligned}\mathbf{S}_k &= (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)\\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1}\\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-))\\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-\end{aligned} \tag{6.5}$$
where $\mathbf{I}$ is a identity matrix, and the measurement matrix ($\mathbf{H}$) is calculated as the following Jacobian of the measurement $\mathbf{h}(.)$ function:
$$\mathbf{H} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \vdots & \vdots \\ \frac{\partial h_5}{\partial x_1} & \frac{\partial h_5}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_5 \end{bmatrix} \tag{6.6}$$
where $\mathbf{H}_1 = \left[-\frac{L\omega}{(\Delta_R)^2} \; \frac{L}{\Delta_R} \right]$, $\mathbf{H}_2 = \mathbf{H}_4 = [1 \;\; -e]$, and $\mathbf{H}_3 = \mathbf{H}_5 = [1 \; e]$.

However this odometric model doesn't solve the problem inherent to slippages. If a big slippage occurs, the EKF will not eliminate its effects. This problem can be attenuated by pre-processing the redundant data before providing it to the KF-CT algorithm. We can compute an approximate motion of the rear wheels based on the motion performed by the front wheels and vice versa, applying the following equations:
$$\begin{aligned}\Delta_R &= \tfrac{\Delta_{RR}+\Delta_{RL}}{2} \;;\; \omega_R = \tfrac{\Delta_{RR}-\Delta_{RL}}{2e}\\ \Delta_F &= \tfrac{\Delta_{FR}+\Delta_{FL}}{2} \;;\; \omega_F = \tfrac{\Delta_{FR}-\Delta_{FL}}{2e}\end{aligned} \tag{6.7}$$
$$\Delta_{F/R} = \frac{\Delta_R}{\cos(\varphi)} \;;\; \Delta_{R/F} = \cos(\varphi).\Delta_F \tag{6.8}$$
$$\begin{aligned}\Delta_{VRL} = \Delta_{R/F} - e.\omega_F \;;\; \Delta_{VRR} = \Delta_{R/F} + e.\omega_F\\ \Delta_{VFL} = \Delta_{F/R} - e.\omega_R \;;\; \Delta_{VFR} = \Delta_{F/R} + e.\omega_R\end{aligned} \tag{6.9}$$
In (6.7) (Δ_R,ω_R) and (Δ_F,ω_F) are the parameters with respect to the midpoint of the rear axle and front axle, respectively. In (6.8) and (6.9) $\Delta_{i/j}$ means Δ_i computed based on measurements from j, with $i, j = \{F, R\}$. Equations (6.9) express the designated virtual

displacements for each wheel. Based on (6.9) we define the following confidence coefficients:

$$
\begin{aligned}
CC_R &= 1 - \tfrac{|\Delta_{VRL}-\Delta_{RL}|+|\Delta_{VRR}-\Delta_{RR}|}{|\Delta_{VRL}+\Delta_{RL}+\Delta_{VRR}+\Delta_{RR}|} \\
CC_F &= 1 - \tfrac{|\Delta_{VFL}-\Delta_{FL}|+|\Delta_{VFR}-\Delta_{FR}|}{|\Delta_{VFL}+\Delta_{FL}+\Delta_{VFR}+\Delta_{FR}|}
\end{aligned}
\qquad (6.10)
$$

The confidence coefficients are used to decide if a virtual measure (Δ_{VRR}, Δ_{VRL}, Δ_{VFR} and Δ_{VFL}) is used instead of the real measure. Figure 6.2 illustrates the algorithm implemented where the confidence coefficients are used to decide if a virtual measure is used instead the real measure.

6.1.2 Inter-vehicle pose estimation using laserscanner data and magnetic landmark

The effect of wheel slippage in the vehicle pose estimation are attenuated using algorithm KF-CT, but the cumulative errors, due to odometry relative and integrative nature, are not removed. These errors can almost be eliminated by applying a second cascaded EKF. If the cascaded EKF uses detected magnetic markers in the fusion method it is designated by Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection (KF-CT+MAG).

However this procedure by itself does not solve the problem of accurate positioning when the magnetic markers are not widely distributed. The magnitude of these errors may unable a correct speed profile following or speed profile assignment by the traffic management system, however this problem can be attenuated by using nearby V2V CS, LIDAR data and a RTK-GPS (see Figure 6.4). In the proposed method, the RTK-GPS positioning, is computed from raw GPS measurements using the double differencing method described in section 3.1.1. The RTK-GPS positioning is loosely coupled with the odometry/laser/magnetic sensors.

In this algorithm besides laser detected natural features and magnetic marker landmarks, a front laser detected vehicle can also be considered as a landmark. A front laser detected vehicle, is considered a landmark, if at a given time the front detected vehicle, it either detected a magnetic marker or it has been able to compute a RTK-GPS positioning solution (see Figures 6.3 and 6.4).

Without loss of generality, we assume that there are only two vehicles to simplify our notation, vehicle 1 is the following vehicle and vehicle 2 is the preceding vehicle. The two vehicles can have different combinations of sensors. In the scenario of Figure 6.3, both vehicles are able to perform V2X, vehicle 1 is only equipped with MSS, encoders and laserscanner while vehicle 2 is equipped with MSS, encoders, laserscanner and RTK-GPS receiver. Using laserscanner data and V2V, vehicle 1 can use vehicle 2 accurate RTK-GPS position to improve its own position; this method is designated by Kinematic sensors Fusion with Confidence Tests + Laser landmarks detection with Real Time Kinematic gps accuracy (KF-CT+LASER(RTK)). From laser scans we can calculate an average range-bearing pair (d_l, ϕ_l) using the range from ray intersection, $ri1$ to $ri4$ and its angular displacement. Vehicle 2 can therefore be considered has a landmark by vehicle 1, by using the average range-bearing pair and associating it with every vehicle 2 RTK-GPS position update. Even if vehicle 2 is not equipped with RTK-GPS the same principle may be applied when vehicle 2 detects a magnet and get its pose corrected. In this second scenario vehicle 2 informs vehicle 1 of its accurate position updated through magnet detection and then if vehicle 2 is in the laserscanner range of vehicle 1 its (vehicle 1) pose is also updated, this method is designated by Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection + Laser landmarks detection (KF-CT+MAG+LASER). When the update is made using both KF-CT+LASER(RTK) and KF-CT+LASER(RTK) it is is designated by Kinematic sensors Fusion with Confidence Tests + MAGnetic markers landmark detection + Laser landmarks detection with Real Time Kinematic gps accuracy (KF-CT+MAG+LASER(RTK)). The range-bearing measurements associated to each landmark are treated as measurements in the second cascaded EKF fusion filter.

Figure 6.4 presents this contribution of vehicle 2 to the vehicle 1 positioning, in the cascaded EKF position estimation.

In this algorithm, the RTK-GPS, when available at the vehicle, is loosely coupled with the odometry. The developed sensor fusion is supported by V2X CS, allowing the exchange of information, between vehicles and the infrastructure agent (see Figure 6.1).

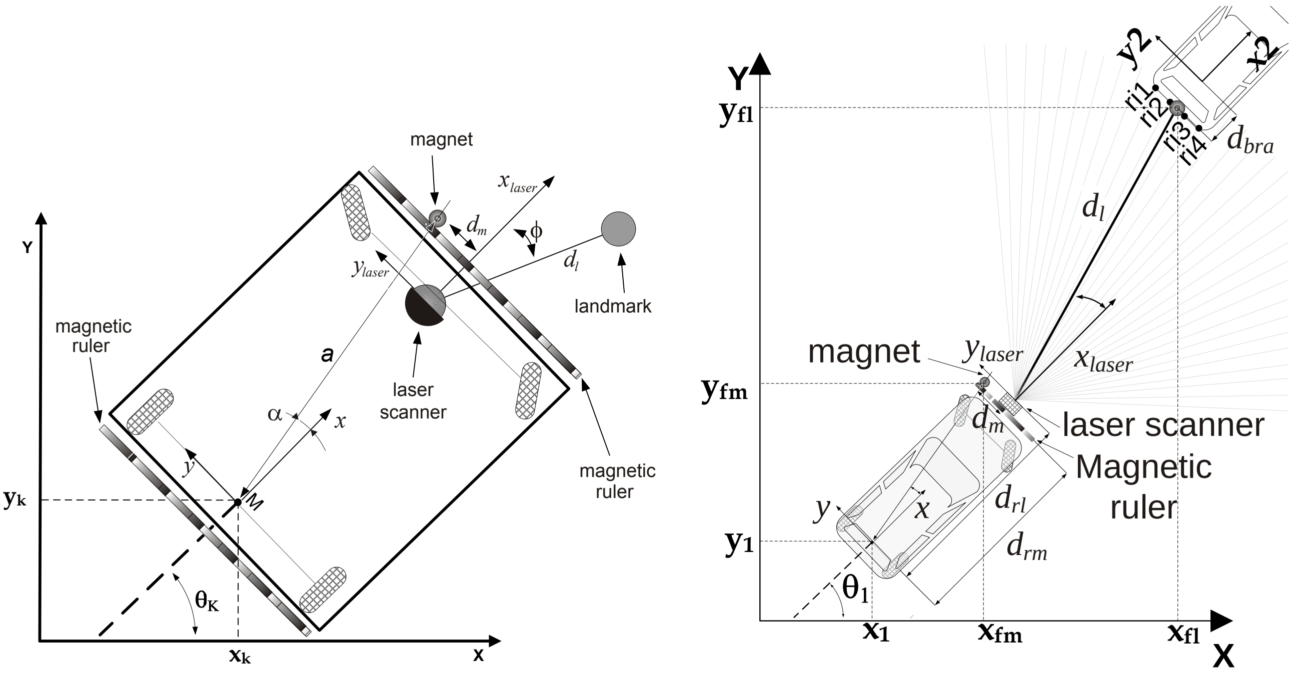
The vehicle's pose is defined by the Cartesian coordinates (x, y) and heading (θ), which are the state variables of a new EKF. The state variables of the KF-CT (section 6.1.1) are here treated as inputs to the EKF data fusion, i.e $\mathbf{u}_k = (\Delta_R, \omega_R)$ with an associated noise covariance matrix $\mathbf{\Gamma}_k$. The range-bearing measurements associated to each landmark are treated as measurements in the fusion process.

1) ***System Model:*** the system model is defined by the kinematic nonlinear equation (B.23), with state vector $\mathbf{x}_k = [x_k \ y_k \ \theta_k]^T$, and input $\mathbf{u}_k = [\Delta_{R,k} \ \omega_{R,k}]^T$, which can be written in the compact form (including noises):

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \gamma_k, \sigma_k) \tag{6.11}$$

where γ_k and σ_k denote the system and input noises, with associated matrices $\mathbf{Q}$ and $\mathbf{\Gamma}_k$.

2) ***Measurement Model*** *(example for the front magnetic ruler and laser-based detected landmarks):* let (a, α) be the range-bearing pair, associated to a detected landmark, defined in the local vehicle coordinate system (see Figure 6.3). Thus the following equations



(a) (b)

Figure 6.3: Measurement model variables: (a) (a, α) denotes range-bearing data associated to a magnet detection; (b) $[x_1, y_1, \theta_1]$ and $[x_2, y_2, \theta_2]$ are respectively the poses of the ahead vehicle and following vehicle; the $[x_{fm}, y_{fm}]$ and $[x_{fl}, y_{fl}]$ are respectively the magnet and laser detected landmark positions; the $[d_{rm}, d_{rl}]$ are respectively the distance from the rear axle to MSS and to the laserscanner axis; the d_{bra} is the distance from the rear bumper to the rear axle of the preceding vehicle; where d_m is the MSS measure which corresponds to the distance between the marker with known position and the MSS central point, and where (d_l, ϕ) is the average range-bearing data in the laserscanner coordinate system. It is assumed that both MSS and the laserscanner coordinate systems are aligned with the vehicle coordinate system.

yield:

$$\begin{gathered} a = \sqrt{(y_f - y_k)^2 + (x_f - x_k)^2} \\ \alpha = \arctan \tfrac{y_f - y_k}{x_f - x_k} - \theta_k \end{gathered} \tag{6.12}$$

where (x_f, y_f) represents the Cartesian position of the landmark. From (6.12) we can define the nonlinear measurement model

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{6.13}$$

where $\mathbf{h}(\mathbf{x}_k)$ is the nonlinear vector function

$$\mathbf{h}(\mathbf{x_k}) = \begin{bmatrix} \sqrt{(y_f - y_k)^2 + (x_f - x_k)^2} \\ \arctan \frac{y_f - y_k}{x_f - x_k} - \theta_k \end{bmatrix} \tag{6.14}$$

and $\mathbf{v}_k$ is the Gaussian sensor noise vector with covariance matrix $\mathbf{R_k}$. The range-bearing data (a, α) are the observation values entering the EKF, $\mathbf{z} = [a \ \alpha]^T$, which are calculated from sensor measures as follows:

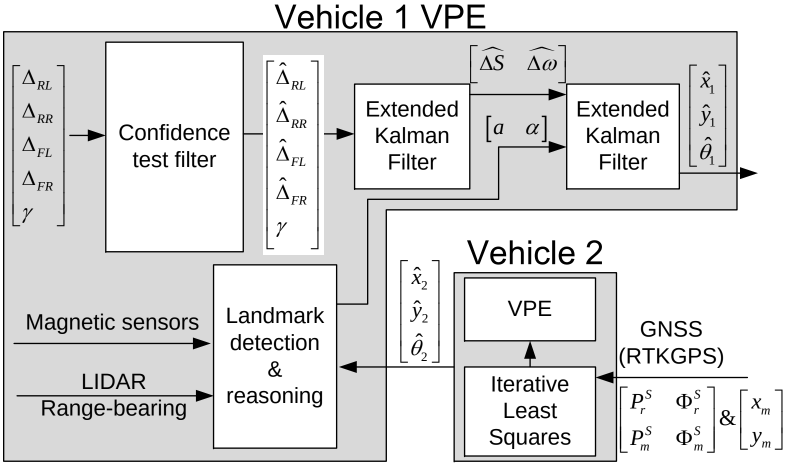

Vehicle 1 VPE
$\begin{bmatrix}\Delta_{RL}\\ \Delta_{RR}\\ \Delta_{FL}\\ \Delta_{FR}\\ \gamma\end{bmatrix}$
Confidence test filter
$\begin{bmatrix}\hat{\Delta}_{RL}\\ \hat{\Delta}_{RR}\\ \hat{\Delta}_{FL}\\ \hat{\Delta}_{FR}\\ \gamma\end{bmatrix}$
Extended Kalman Filter
$\begin{bmatrix}\widehat{\Delta S} & \widehat{\Delta\omega}\end{bmatrix}$
$\begin{bmatrix}a & \alpha\end{bmatrix}$
Extended Kalman Filter
$\begin{bmatrix}\hat{x}_1\\ \hat{y}_1\\ \hat{\theta}_1\end{bmatrix}$
Magnetic sensors
LIDAR Range-bearing
Landmark detection & reasoning
$\begin{bmatrix}\hat{x}_2\\ \hat{y}_2\\ \hat{\theta}_2\end{bmatrix}$
Vehicle 2
VPE
Iterative Least Squares
GNSS (RTKGPS)
$\begin{bmatrix}P_r^S & \Phi_r^S\\ P_m^S & \Phi_m^S\end{bmatrix}$ & $\begin{bmatrix}x_m\\ y_m\end{bmatrix}$

Figure 6.4: Inter-vehicle sensor fusion algorithm, where: $[\hat{x}_1, \hat{y}_1, \hat{\theta}_1]$ and $[\hat{x}_2, \hat{y}_2, \hat{\theta}_2]$ are respectively the following and preceding vehicle estimated pose; $[\hat{\Delta}_{RR}, \hat{\Delta}_{RL}, \hat{\Delta}_{FR}, \hat{\Delta}_{FL}, \gamma]$ are the estimated displacement wheel encoders and the steering encoder; $[\hat{\Delta}_R, \hat{\omega}_R]$ are respectively the estimated the arc length and estimated elementary rotation, $[a, \alpha]$ is the range-bearing pair, associated to a detected landmark, defined in the local vehicle coordinate system and where $[P_r^s, \Phi_r^s; P_m^s \Phi_m^s]$ and $[x_m, y_m]$ are respectively the pseudoranges and Master Station truth position.

a) *for the magnetic marker*

$$\begin{aligned} a &= \sqrt{d_m^2 + L_1^2} \\ \alpha &= \arctan \tfrac{d_m}{L_1} \end{aligned} \tag{6.15}$$

where d_m is the magnetic ruler measure which corresponds to the distance between the marker with known position (x_m, y_m) and the magnetic-sensing ruler central point, and L_1 is the distance between the front magnetic ruler and the vehicle rear axis (we are assuming that the ruler is perfectly parallel with the y-axis of vehicle coordinate system).

b) *for the laser-based detected landmark*

$$\begin{aligned} a &= \sqrt{d_l^2 + L_l^2 + 2 d_l L_l \cos(\phi)} \\ \alpha &= \arctan \tfrac{d_l \sin(\phi)}{L_l + d_l cos(\phi)} \end{aligned} \tag{6.16}$$

where (d_l, ϕ) are the range-bearing data described in the laser coordinate system. It is assumed that the laser coordinate system is aligned with the vehicle coordinate system, with a distance L_l, defined in the xy-plane, between them.

Another (non-standard) measurement model has been investigated and applied as described in [107], which consists on considering in model (6.13):

$$\begin{aligned} \mathbf{z} &= [x_f \ y_f]^T \\ \mathbf{h}(\mathbf{x}_k) &= \begin{bmatrix} x_k + a\cos(\theta_k + \alpha) \\ y_k + a\sin(\theta_k + \alpha) \end{bmatrix} \end{aligned} \tag{6.17}$$

3) ***EKF algorithm*:** it is composed by the following prediction and correction stages:

Prediction stage

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0, 0) \\ \mathbf{P}_k^- &= \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{B}_k \mathbf{\Gamma}_{k-1} \mathbf{B}_k^T + \mathbf{Q} \end{aligned} \tag{6.18}$$

where the system ($\mathbf{A}$) and input ($\mathbf{B}$) matrices are calculated as the following Jacobian of the system $\mathbf{f}(.)$ function:

$$\begin{aligned}
\mathbf{A}_k &= \begin{bmatrix} 1 & 0 & -\Delta_{R,k}\sin(\theta_k + \frac{\omega_{R,k}}{2}) \\ 0 & 1 & \Delta_{R,k}\cos(\theta_k + \frac{\omega_{R,k}}{2}) \\ 0 & 0 & 1 \end{bmatrix} \\
\mathbf{B}_k &= \begin{bmatrix} \cos(\theta_k + \frac{\omega_{R,k}}{2}) & -\frac{\Delta_{R,k}}{2}\sin(\theta_k + \frac{\omega_{R,k}}{2}) \\ \sin(\theta_k + \frac{\omega_k}{2}) & \frac{\Delta_{R,k}}{2}\cos(\theta_k + \frac{\omega_{R,k}}{2}) \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{6.19}$$

Correction stage

Once measurements (a, α) become available the following correction stage is done:

$$\begin{aligned}
\mathbf{S}_k &= (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \\
\hat{\mathbf{x}}_k &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^-
\end{aligned} \tag{6.20}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{H}_k$ is the Jacobian of the measurement $\mathbf{h}(.)$ function:

$$\mathbf{H}_k = \nabla_x \mathbf{h}(\mathbf{x}_k) \tag{6.21}$$

Data Association

Most techniques implement the data association process based on the innovation sequence and its predicted covariance. The innovation sequence υ_k relates observations $\mathbf{z}_k$ to the underlying predicted states $\hat{\mathbf{z}}_k$

$$\upsilon_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \tag{6.22}$$

Let define the normalised innovation distance as

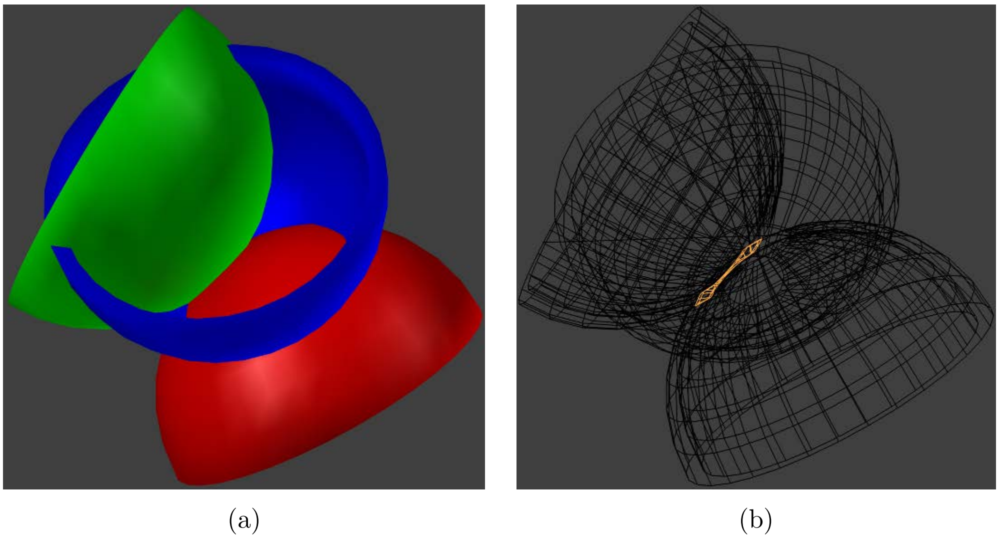
$$\bar{d_k} = {\upsilon_k}^T {\mathbf{S}_k}^{-1} \upsilon_k \tag{6.23}$$

where $\mathbf{S}_k$ is the innovation covariance matrix defined in (6.20). Note that if the innovation υ_k has a Gaussian distribution, then $\bar{d}$ is a random variable following the χ^2 distribution. The innovation sequence is the basis of the *gate validation* technique which accepts the observation that is inside a fixed region of a χ^2 distribution, and rejects the observation that make the innovation fall outside these bounds. This procedure is achieved by comparing the scalar obtained in (6.23) with a threshold value that is determined from the χ^2 distribution table. We use the *nearest neighbour* data association method. Among all the possible natural features or front laser detected vehicle, the one that is nearest to the observation is selected and used in the EKF correction stage (6.20).

6.2 Robust, Augmented and Collaborative GNSS positioning

6.2.1 Set membership GNSS Positioning (SGP)

The Set membership GNSS Positioning (*Set-membership GNSS Positioning (SGP)*) method [70] consists in finding a location zone given uncertainty on satellites observations and satellites positions. GNSS absolute positioning requires satellites observations ρ_r^s as well as their positions $\mathbf{x^s} = (x^s, y^s, z^s)$ at the time of transmission. Satellite positions are known with uncertainty. Therefore, for set-membership positioning, each satellite position is represented as a box $[\mathbf{x^s}] = ([x^s], [y^s], [z^s])$ whose bounds are chosen to contain the true satellite position at a given confidence level. As mentioned in Section 3.1.1, pseudorange measurements are inaccurate, therefore in the context of set-membership positioning, pseudorange measurements are modelled as intervals $[\rho_r^s]$ whose bounds are determined given a chosen risk [72].


(a)

(b)

Figure 6.5: Three satellite observations: (a) spherical shell for each satellite; (b) set inside the measurement and satellite position intervals for every satellite.

Assuming no time errors among devices, GNSS absolute positioning requires at least three satellite observations. Each observation constrains the GNSS receiver location inside a spherical shell, whose inner and outer radii are respectively the lower and upper bounds of the measurement interval, see Figure 6.5 (a).

The GNSS receiver is thus located within the set of locations for which a range and a satellite position can be found inside the measurement and satellite position intervals for every satellite, *i.e.* the highlighted area in Figure 6.5 (b).

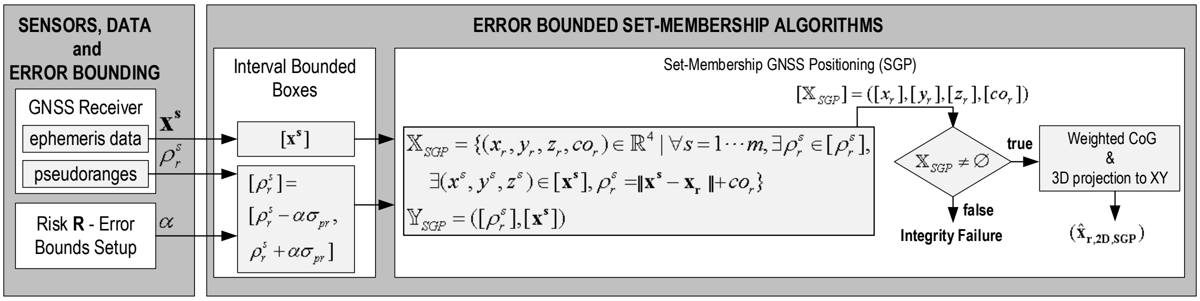

SENSORS, DATA and ERROR BOUNDING
GNSS Receiver
ephemeris data
pseudoranges
Risk R - Error Bounds Setup
$\mathbf{x^s}$
ρ_r^s
α
ERROR BOUNDED SET-MEMBERSHIP ALGORITHMS
Interval Bounded Boxes
$[\mathbf{x^s}]$
$[\rho_r^s] = [\rho_r^s - \alpha\sigma_{pr}, \rho_r^s + \alpha\sigma_{pr}]$
Set-Membership GNSS Positioning (SGP)
$[\mathbb{X}_{SGP}] = ([x_r],[y_r],[z_r],[co_r])$
$\mathbb{X}_{SGP} = \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 \mid \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r\}$
$\mathbb{Y}_{SGP} = ([\rho_r^s],[\mathbf{x^s}])$
$\mathbb{X}_{SGP} \neq \varnothing$
true
false
Integrity Failure
Weighted CoG & 3D projection to XY
$(\hat{\mathbf{x}}_{\mathbf{r,2D,SGP}})$

Figure 6.6: Dataflow and SGP algorithm architecture.

As there is a receiver clock offset, GNSS positioning is a four-dimensional problem and at least four satellites observations are necessary to estimate the GNSS receiver position. The GNSS receiver position zone computation consists in finding the set $\mathbb{X}$ of all locations compatible with the m available measurements and the corresponding satellite positions which are also manipulated as boxes (see Figure 6.6):

$$\begin{aligned} \mathbb{X}_{\text{SGP}} &= \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 | \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \\ &\qquad \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r\} \\ \mathbb{Y}_{\text{SGP}} &= ([\rho_r^s], [\mathbf{x^s}]) \end{aligned} \tag{6.24}$$

The subpaving $\mathbb{X}$ has a dual nature. It may be seen as a subset of $\mathbb{R}^4$ and it also can be viewed as a finite list of boxes $[X]$ [83], where $\#\mathbb{X}$ is the number of boxes belonging to the subset. In order to be useful, for instance to a path following controller, a punctual 2D position vector has to be estimated. The resulting set is not only composed of 3D boxes as it includes the clock offset co_r, making it a 4D set. Boxes in the subpaving do not have all the same size (see Figure 3.5). Therefore, a solution to provide punctual estimation with the 4D set is to compute for each of the 4 dimensions, the mean of the geometric center of all boxes weighted by the volume of each box. In the single-frequency GNSS navigation solution with raw observations, the co_r is dominant over atmospheric residuals and noise. A more accurate solution of the 3D punctual location estimate, is therefore, to compute the 3D the mean of the geometric center of all boxes weighted by two parameters. First, the contribution of a box to the final solution is weighted by its volume. Second, the weight is also a function of the co_r of each box. The weight of a box to the final solution decreases as its punctual estimate of the clock offset is far from the weighted average $\hat{co}_r$.

Given $([X] = ([x_r], [y_r], [z_r], [co_r]))$, the $\hat{co}_r$ is estimated by computing the center of gravity of the $X(4)$ component, for all n boxes:

$$\hat{co}_r = \left. \frac{\sum_{k=1}^{n} \left(\frac{\overline{X_k(4)} + \underline{X_k(4)}}{2} \cdot (\overline{X_k(4)} - \underline{X_k(4)}) \right)}{X_{all}(4)} \right| \tag{6.25}$$

$$\forall [X_k(4)] \in \mathbb{X} \wedge k = \#\mathbb{X}$$

$X_{all}(4)$ is the normalisation term.

The estimated 3D position vector $\mathbf{\hat{x}_r} = (\hat{x}_r, \hat{y}_r, \hat{z}_r)$ is obtained by computing the center of gravity of the sub-paving, weighted by the value of the estimated receiver clock offset $\hat{co}_r$ of each sub-paving:

$$\mathbf{\hat{x}_r} = \frac{\sum_{k=1}^{n} \left(\frac{\overline{X_k(i)} + \underline{X_k(i)}}{2} \cdot \left(\overline{X_k(i)} - \underline{X_k(i)} \right) \cdot C_{wf}(k) \right)}{X_{all}(i)} \tag{6.26}$$

$$\forall [X_k(i)] \in \mathbb{X} \wedge k = \#\mathbb{X} \wedge i = 1, \cdots, 3$$

and $X_{all}(i)$ is the sum of all boxes lengths along i axis:

$$X_{all}(i) = \sum_{k=1}^{n} (\overline{X_k(i)} - \underline{X_k(i)}) \tag{6.27}$$

$$\forall [X_k(i)] \in \mathbb{X} \wedge k = \#\mathbb{X} \wedge i = 1, \cdots, 4$$

where C_{wf} is the $\hat{co}_r$ weighting factor:

$$C_{wf}(k) = \frac{1 - \frac{\left| \left| \frac{\overline{X_k(4)} + \underline{X_k(4)}}{2} \right| - \hat{co}_r \right|}{\left| max\left(|\overline{X(4)}|, |\underline{X(4)}| \right) - \hat{co}_r \right|}}{\sum_{j=1}^{n} \left(1 - \frac{\left| \left| \frac{\overline{X_j(4)} + \underline{X_j(4)}}{2} \right| - \hat{co}_r \right|}{\left| max\left(|\overline{X(4)}|, |\underline{X(4)}| \right) - \hat{co}_r \right|} \right)} \tag{6.28}$$

$$\forall [X_j(4)] \in \mathbb{X} \wedge j = \#\mathbb{X}$$

The 2D position $[x_r, y_r]$ estimate is determined by projecting onto a flat plane the solution $\mathbf{\hat{x}_r}$.

6.2.2 Robust Lane Boundary Augmented Iterative least squares GNSS Positioning algorithm (*LB-RAIGP*)

Additional absolute measurements can be used to reduce the error in the estimation of the receiver position $\mathbf{\hat{x}_r}$ and the receiver clock offset $\hat{co}_r$. Geographic Information Systems (GIS)

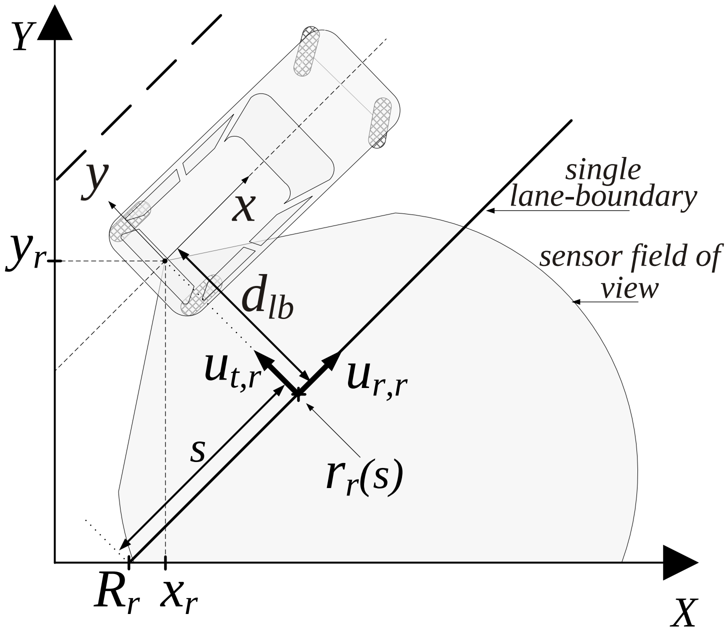


Figure 6.7: Lane-boundary distance measurement geometric layout

combined with lane-boundary sensors can provide the additional absolute measurements. Considering that the lane-boundary sensor is collocated with the GNSS receiver, the equation relating the vehicle position to the lane boundary is given by (see Figure 6.7):

$$r_r(s) = \mathbf{x_r} + d_{lb} \cdot \mathbf{u_{t,r}} \tag{6.29}$$

where s is the distance from the reference point R_r, $r_r(s)$ is the lane-boundary function, $\mathbf{u_{t,r}}$ is transverse unit vector and d_{lb} is the perpendicular distance from the lane-boundary sensor to the lane boundary. Lane-boundary sensors only provide the distance to a line, *i.e.* the location along the lane boundary line is undefined. If the observed road is composed by a single straight line segment, the transverse component of the distance from the lane-boundary sensor to the lane boundary it is unambiguous and can be expressed by:

$$d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) + \varepsilon_r^{lb} \tag{6.30}$$

where the added random noise ε_r^{lb}, models the lane-boundary sensor measurement errors.

Rewriting (3.19) with the augmented measurement vector:

$$\Delta\rho_{r,lb} = \mathbf{H_{lb}} \begin{bmatrix} \Delta\mathbf{x_r} \\ \Delta co_r \end{bmatrix} + v_r^s \tag{6.31}$$

where $\mathbf{H_{lb}}$ is given by:

$$\mathbf{H_{lb}} = \begin{bmatrix} \multicolumn{2}{c}{\mathbf{H}} \\ -\mathbf{u_{t,r}} & 0 \end{bmatrix} \tag{6.32}$$

$\mathbf{H}$ is the *observation matrix* from equation 6.38, and $\Delta\rho_{r,lb}$ is given by:

$$\Delta\rho_{r,lb} = \begin{bmatrix}\Delta\rho_r^1 & \dots & \Delta\rho_r^m & \Delta d_{lb}\end{bmatrix}^T \tag{6.33}$$

By adding the component of the satellite-specific errors, which causes the receiver position to move in a direction transverse to the lane boundary, as an additional state ($C_{t,cme}$), the linear least squares equation (6.31) is transformed into:

$$\Delta\rho_{r,lb} = \mathbf{H_{lbt}} \begin{bmatrix}\Delta\mathbf{x_r}\\ \Delta co_r \\ \Delta C_{t,cme}\end{bmatrix} + v_r^s \tag{6.34}$$

where $\mathbf{H_{lbt}}$ is given by:

$$\mathbf{H_{lbt}} = \begin{bmatrix}\mathbf{H} & t_{vsse}\\ [-\mathbf{u_{t,r}} \;\; 0] & 0\end{bmatrix} \tag{6.35}$$

and t_{vsse} vector represent the transverse components of the satellite-specific error for all satellites:

$$t_{vsse} = \mathbf{H}\begin{bmatrix}\mathbf{u_{t,r}}\\ 0\end{bmatrix} \tag{6.36}$$

The additional geometric diversity is provided by the augmented geometry matrix $\mathbf{H_{lbt}}$, this diversity further reduces the estimation errors. The method here denominated by Lane Boundary Augmented Iterative least squares GNSS Positioning (LB-AIGP) is fully described in [70].

This method is sensible to multipath, namely NLOS multipath , using a method described in section 6.2.6, the faulty satellites are detected and removed, considering a risk $\mathbf{R}$. The observation matrix $\mathbf{H_{lbt}}$, is therefore rewritten:

$$\mathbf{H_{lbt}} = \begin{bmatrix}\mathbf{H_{NLOS}} & t_{vsse}\\ [-\mathbf{u_{t,r}} \;\; 0] & 0\end{bmatrix} \tag{6.37}$$

where $\mathbf{H_{NLOS}}$ is the *observation matrix* :

$$\mathbf{H_{NLOS}} = drop_i\left(\begin{bmatrix}-e^1 & 1\\ \vdots & \vdots \\ -e^i & 1\\ \vdots & \vdots \\ -e^m & 1\end{bmatrix}\right) \wedge i = NLOS\ satellite \tag{6.38}$$

where the “drop” notation is used to represent the removing of a row from a matrix. Similarly the faulty pseudoranges are removed from the position computation of algorithm LB-RAIGP, hence $\Delta\rho_{r,lb,NLOS}$ is given by:

$$\Delta\rho_{r,lb,NLOS} = drop_i\left(\begin{bmatrix}\Delta\rho_r^1 & \dots & \Delta\rho_r^i & \dots & \Delta\rho_r^m & \Delta d_{lb}\end{bmatrix}^T\right) \wedge i = NLOS\ satellite \tag{6.39}$$

6.2.3 Lane Boundary Augmented set membership GNSS Positioning (LB-ASGP)

Lane-boundary constraints can further improve the performance of the set-membership GNSS location zone determination.

Algorithm 8 $[\mathbf{\hat{x}_{r,2D}}; C_{\mathbf{t,cme}}] =$
$ASGP\_LB\left([\mathbf{f_{SGP}}], [\mathbf{f_{LB-ASGP}}], \mathbb{Y}_{SGP}, \mathbb{Y}_{LB-ASGP}, \mathbb{X}_0\right)$

```
$\mathfrak{L} \leftarrow root(\mathbb{X}_0)$
$\mathbf{f}_{aux} = [\mathbf{f}_{\text{SGP}}; \mathbf{f}_{\text{LB-ASGP}}]$
$\mathbb{Y}_{aux} = [\mathbb{Y}_{\text{SGP}}, \mathbb{Y}_{\text{LB-ASGP}}]$
for jj=1 to 2 do
    $\mathbf{f} \leftarrow \mathbf{f}_{aux}(jj)$
    $\mathbb{Y} \leftarrow \mathbb{Y}_{aux}(jj)$
    $[\underline{\mathbb{X}}, \Delta\mathbb{X}] = \text{SIVIA}(\mathbf{f}, \mathbb{Y}, \mathfrak{L})$
    $\overline{\mathbb{X}} = \underline{\mathbb{X}} + \Delta\mathbb{X}$
    $\mathfrak{L} \leftarrow \overline{\mathbb{X}}$
    for i=1 to 4 do
        $X_{all}(i) \leftarrow$ equation (6.27)
    end for
    $\hat{co}_r \leftarrow$ equation (6.25)
    for k=1 to n do
        $C_{wf}(k) \leftarrow$ equation (6.28)
    end for
    for i=1 to 3 do
        $\mathbf{\hat{x}_{aux,r}}(\hat{x}_{aux,r}, \hat{y}_{aux,r}, \hat{z}_{aux,r}) \leftarrow$ equation (6.26)
    end for
    $\mathbf{\hat{x}_r}(\mathbf{jj}) = \mathbf{\hat{x}_{aux,r}}$
end for
$C_{\mathbf{t,cme}} = \mathbf{u_{t,r}} \times [\mathbf{\hat{x}_r}(2) - \mathbf{\hat{x}_r}(1)]$
$\mathbf{\hat{x}_{r,2D}} = [\hat{x}_{r,2D}, \hat{y}_{r,2D}] = \mathbf{\hat{x}_r}(2) \times \mathbf{P_{XY}}$
($\mathbf{P_{XY}}$: 3D projection to XY)
```

Algorithm 8 resumes the LB-ASGP proposed method where the functions **f** to invert are given by:

$$\mathbf{f}_{\text{SGP}} = \left\{\ \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r\right. \tag{6.40}$$

and

$$\mathbf{f}_{\text{LB-ASGP}} = \begin{cases} \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r \\ d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) \end{cases} \tag{6.41}$$

As observed in Figure 6.8, the **Algorithm 8** solves simultaneously two set inversion problems to compute an 2D estimate of the GNSS receiver $\mathbf{\hat{x}_{r,2D}}$, and an estimate of

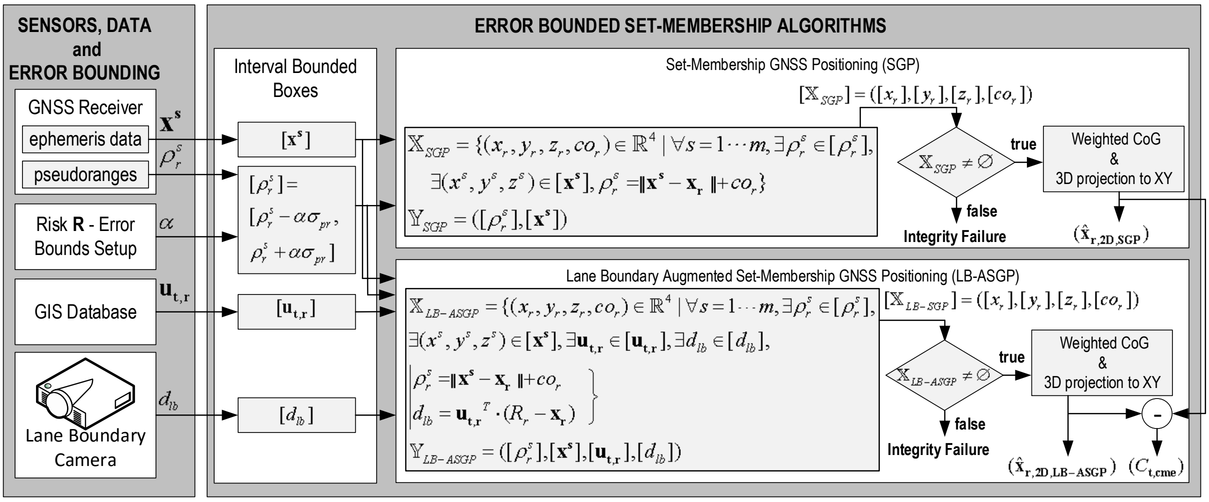

SENSORS, DATA and ERROR BOUNDING
GNSS Receiver
ephemeris data
pseudoranges
Risk R - Error Bounds Setup
GIS Database
Lane Boundary Camera
$\mathbf{x^s}$
ρ_r^s
α
$\mathbf{u_{t,r}}$
d_{lb}
ERROR BOUNDED SET-MEMBERSHIP ALGORITHMS
Interval Bounded Boxes
$[\mathbf{x^s}]$
$[\rho_r^s] = [\rho_r^s - \alpha\sigma_{pr}, \rho_r^s + \alpha\sigma_{pr}]$
$[\mathbf{u_{t,r}}]$
$[d_{lb}]$
Set-Membership GNSS Positioning (SGP)
$[\mathbb{X}_{SGP}] = ([x_r],[y_r],[z_r],[co_r])$
$\mathbb{X}_{SGP} = \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 \mid \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \rho_r^s = \| \mathbf{x^s} - \mathbf{x_r} \| + co_r\}$
$\mathbb{Y}_{SGP} = ([\rho_r^s],[\mathbf{x^s}])$
$\mathbb{X}_{SGP} \neq \emptyset$
true
false
Integrity Failure
Weighted CoG & 3D projection to XY
$(\hat{\mathbf{x}}_{\mathbf{r,2D,SGP}})$
Lane Boundary Augmented Set-Membership GNSS Positioning (LB-ASGP)
$[\mathbb{X}_{LB-SGP}] = ([x_r],[y_r],[z_r],[co_r])$
$\mathbb{X}_{LB-ASGP} = \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 \mid \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \exists \mathbf{u_{t,r}} \in [\mathbf{u_{t,r}}], \exists d_{lb} \in [d_{lb}], \left. \begin{array}{l} \rho_r^s = \| \mathbf{x^s} - \mathbf{x_r} \| + co_r \\ d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) \end{array} \right\}$
$\mathbb{Y}_{LB-ASGP} = ([\rho_r^s],[\mathbf{x^s}],[\mathbf{u_{t,r}}],[d_{lb}])$
$\mathbb{X}_{LB-ASGP} \neq \emptyset$
true
false
Integrity Failure
Weighted CoG & 3D projection to XY
-
$(\hat{\mathbf{x}}_{\mathbf{r,2D,LB-ASGP}})$
$(C_{\mathbf{t,cme}})$

Figure 6.8: Dataflow and LB-ASGP algorithm architecture.

the offset along the lateral direction of the road $C_{t,cme}$. From an initial searching volume $\mathbb{X}_0$, it first finds the set $\mathbb{X}_{\text{SGP}}$ of all locations compatible with the measurements $[\rho_r^s]$ and the satellite position intervals $[\mathbf{x^s}]$ using function $\mathbf{f}_{\text{SGP}}$. Starting with the $\mathbb{X}_{\text{SGP}}$ computed previously, using function $\mathbf{f}_{\text{LB-ASGP}}$ it computes the set $\mathbb{X}_{\text{LB-ASGP}}$ of all locations compatible with: the measurements $[\rho_r^s]$, the satellite position intervals $[\mathbf{x^s}]$, the transverse vector $[\mathbf{u_{t,r}}]$ and the lane boundary $[d_{lb}]$. Using sets $\mathbb{X}_{\text{SGP}}$ and $\mathbb{X}_{\text{LB-ASGP}}$, the 3D position estimate $\hat{\mathbf{x}}_\mathbf{r}$ is obtained by computing the center of gravity of the sub-paving, weighted by the value of the estimated receiver clock offset $\hat{co}_r$ of each sub-paving. The vehicle 2D position estimation $\hat{\mathbf{x}}_{\mathbf{r,2D}}$ is achieved by projecting the 3D position estimate $\hat{\mathbf{x}}_\mathbf{r}$ onto the XY plane.

The component of the satellite-specific errors which causes the receiver position estimate to suffer from an offset in the direction transverse to the lane boundary ($C_{t,cme}$) is also estimated in **Algorithm 8**. The $C_{t,cme}$ is given by equation (6.42), which is obtained by taking into account only the road transverse component that has shifted the estimated position obtained using the *SGP* algorithm to the estimated position obtained using *LB-ASGP* (see Figure 6.8).

$$C_{\mathbf{t,cme}} = \mathbf{u_{t,r}} \times [\hat{\mathbf{x}}_{\mathbf{LB-ASGP}} - \hat{\mathbf{x}}_{\mathbf{SGP}}] \tag{6.42}$$

6.2.4 Robust Lane Boundary Collaborative Augmented Iterative least squares GNSS Positioning algorithm (*LB-RCAIGP*)

A solution for collaborative GNSS augmented solution using lane-boundary sensors, is to fuse pseudoranges and lane-boundary sensor measurements from all vehicles, together into one large least-squares estimation problem, this method designated by LB-RCAIGP. Rewriting (3.19) with all vehicles one gets:

$$\Delta\rho_{r,cn} = \mathbf{H_{cn}} \cdot \Delta\mathbf{X_{cn}} + v_r^s \tag{6.43}$$

where the measurements vector $\Delta\rho_{r,cn}$ of all vehicles sensor data is given by:

$$\Delta\rho_{r,cn} = \begin{bmatrix} \Delta\rho_{r1,lb} \\ \vdots \\ \Delta\rho_{rn,lb} \end{bmatrix} \tag{6.44}$$

and $\mathbf{H_{cn}}$ is the multi-vehicle *augmented observation matrix*:

$$\mathbf{H_{cn}} = \left[\begin{array}{ccc|cc} \mathbf{H_{lb,1}} & \cdots & 0 & t_{vsse,1} & r_{vsse,1} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \mathbf{H_{lb,n}} & t_{vsse,n} & r_{vsse,n} \end{array}\right] \tag{6.45}$$

r_{vsse} vector represent the transverse components of the satellite-specific error for all satellites and $\mathbf{u_{r,r}}$ is the road-parallel unit vector:

$$r_{vsse} = \mathbf{H} \begin{bmatrix} \mathbf{u_{r,r}} \\ 0 \end{bmatrix} \tag{6.46}$$

When vehicles travel in different directions, the two ground-plane components (road-transverse and road-parallel components) of the satellite-specific error can be estimated. By adding these components of the common-mode GNSS errors as additional states (transverse $C_{t,cme}$ and parallel $C_{r,cme}$), the state vector $\mathbf{X_{cn}}$ of all vehicles position and the two components of the common-mode GNSS errors, is given by:

$$\mathbf{X_{cn}} = \left[\begin{array}{c} \Delta\mathbf{x_{r,1}} \\ \Delta co_{r,1} \\ \vdots \\ \Delta\mathbf{x_{r,n}} \\ \Delta co_{r,n} \\ \hline \Delta C_{t,cme} \\ \Delta C_{r,cme} \end{array}\right] \tag{6.47}$$

The estimated $C_{t,cme}$ and $C_{r,cme}$ can also be used by vehicles not equipped with lane-boundary sensors to cancel the effect of satellite specific errors. The same pre-defined pair of orthogonal unit vectors $\mathbf{\hat{u}_{r,r}}$ and $\mathbf{\hat{u}_{t,r}}$ is used as coordinate frame on the determination of $\Delta C_{t,cme}$ and $\Delta C_{r,cme}$ of each vehicle, i.e. direction does not change and it is used by all vehicles' to estimated $C_{t,cme}$ and $C_{r,cme}$. The method here denominated by Lane Boundary Collaborative Augmented Iterative least squares GNSS ositioning (LB-CAIGP) is fully described in [70].

This method is highly sensible to multipath, namely NLOS. If a single connected vehicle is subjected to multipath it results a in a erroneous contribution to all connected vehicles. The erroneous contribution, results in the connected vehicles solution (LB-CAIGP) of each vehicles worst than if the solution was achieved in a non-connected scenario (LB-AIGP).

Using the method described in section 6.2.6, the faulty satellites are detected and removed, considering a risk $\mathbf{R}$. The multi-vehicle *augmented observation matrix* $\mathbf{H_{cn}}$, is therefore rewritten:

$$\mathbf{H_{cn}} = \left[\begin{array}{ccc|cc} \mathbf{H_{lb,1,NLOS}} & \cdots & 0 & t_{vsse,1} & r_{vsse,1} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \mathbf{H_{lb,n,NLOS}} & t_{vsse,n} & r_{vsse,n} \end{array}\right] \tag{6.48}$$

where $\mathbf{H_{lb,n,NLOS}}$ is satellite fault free *observation matrix* of each vehicle $\mathbf{H_{NLOS}}$, as defined in section 6.2.2.

Similarly the faulty pseudoranges are removed from the position computation of algorithm LB-RAIGP, hence $\Delta\rho_{r,cn,NLOS}$ is given by:

$$\Delta\rho_{r,cn,NLOS} = \begin{bmatrix} \Delta\rho_{r1,lb,NLOS} \\ \vdots \\ \Delta\rho_{rn,lb,NLOS} \end{bmatrix} \tag{6.49}$$

where $\Delta\rho_{rn,lb,NLOS}$ is satellite fault free *pseudorange* of each vehicle $\Delta\rho_{lb,NLOS}$, as defined in section 6.2.2.

6.2.5 Lane Boundary Collaborative Augmented set membership GNSS Positioning (*LB-CASGP*)

Lane-boundary measurements can provide corrections to improve the position estimate of a single receiver using *LB-ASGP*, but an improvement by means of a cooperative vehicle positioning (*LB-CASGP*) can be achieved by sharing this corrections among vehicles.

With the assumption that the model and the measurement errors are bounded, both GNSS pseudoranges and lane-boundary data can be fused by using a set-inversion approach in such a way that all the results are guaranteed [108]. Given the road network information

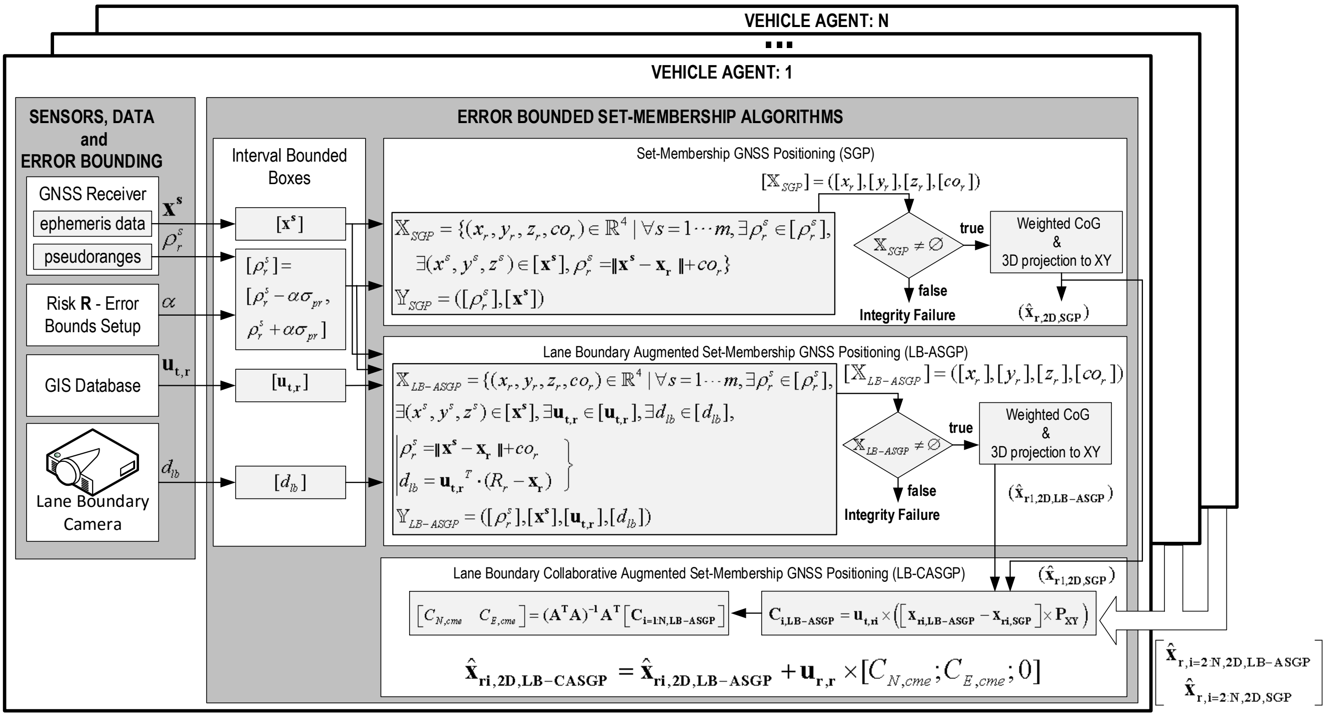

VEHICLE AGENT: N
VEHICLE AGENT: 1
SENSORS, DATA and ERROR BOUNDING
GNSS Receiver
ephemeris data
pseudoranges
$\mathbf{x^s}$
ρ_r^s
Risk **R** - Error Bounds Setup
α
GIS Database
$\mathbf{u_{t,r}}$
Lane Boundary Camera
d_{lb}
ERROR BOUNDED SET-MEMBERSHIP ALGORITHMS
Interval Bounded Boxes
$[\mathbf{x^s}]$
$[\rho_r^s] = [\rho_r^s - \alpha\sigma_{pr}, \rho_r^s + \alpha\sigma_{pr}]$
$[\mathbf{u_{t,r}}]$
$[d_{lb}]$
Set-Membership GNSS Positioning (SGP)
$[\mathbb{X}_{SGP}] = ([x_r],[y_r],[z_r],[co_r])$
$\mathbb{X}_{SGP} = \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 \mid \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r\}$
$\mathbb{Y}_{SGP} = ([\rho_r^s],[\mathbf{x^s}])$
$\mathbb{X}_{SGP} \neq \varnothing$
true
false
Integrity Failure
Weighted CoG & 3D projection to XY
$(\hat{\mathbf{x}}_{\mathbf{r,2D,SGP}})$
Lane Boundary Augmented Set-Membership GNSS Positioning (LB-ASGP)
$[\mathbb{X}_{LB-ASGP}] = ([x_r],[y_r],[z_r],[co_r])$
$\mathbb{X}_{LB-ASGP} = \{(x_r, y_r, z_r, co_r) \in \mathbb{R}^4 \mid \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \exists \mathbf{u_{t,r}} \in [\mathbf{u_{t,r}}], \exists d_{lb} \in [d_{lb}], \left| \begin{array}{l} \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r \\ d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) \end{array} \right\}$
$\mathbb{Y}_{LB-ASGP} = ([\rho_r^s],[\mathbf{x^s}],[\mathbf{u_{t,r}}],[d_{lb}])$
$\mathbb{X}_{LB-ASGP} \neq \varnothing$
true
false
Integrity Failure
Weighted CoG & 3D projection to XY
$(\hat{\mathbf{x}}_{\mathbf{r1,2D,LB-ASGP}})$
Lane Boundary Collaborative Augmented Set-Membership GNSS Positioning (LB-CASGP)
$(\hat{\mathbf{x}}_{\mathbf{r1,2D,SGP}})$
$\begin{bmatrix} C_{N,cme} & C_{E,cme} \end{bmatrix} = (\mathbf{A^T A})^{-1} \mathbf{A^T} \left[\mathbf{C_{i=1:N,LB-ASGP}}\right]$
$\mathbf{C_{i,LB-ASGP}} = \mathbf{u_{t,ri}} \times \left(\left[\mathbf{x_{ri,LB-ASGP}} - \mathbf{x_{ri,SGP}}\right] \times \mathbf{P_{XY}}\right)$
$\hat{\mathbf{x}}_{\mathbf{ri,2D,LB-CASGP}} = \hat{\mathbf{x}}_{\mathbf{ri,2D,LB-ASGP}} + \mathbf{u_{r,r}} \times [C_{N,cme}; C_{E,cme}; 0]$
$\begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{r,i=2:N,2D,LB-ASGP}} \\ \hat{\mathbf{x}}_{\mathbf{r,i=2:N,2D,SGP}} \end{bmatrix}$

Figure 6.9: Dataflow and LB-CASGP algorithm architecture.

provided by a GIS database, a constraint represented by the cross-track vector can be applied to a box $[\mathbf{x}]$. To apply this constraint, the road network information is first transformed into the coordinate system used in the GNSS, *i.e.* convert from rectangular local-level-tangent ENU coordinates to WGS-84 ECEF Cartesian coordinates.

The cross-track vector is represented as a box $[\mathbf{u_{t,r}}]$ whose bounds are chosen to contain the true cross-track vector. The measurement inaccuracy of the perpendicular distance given by the lane-boundary sensor with respect to the lane boundary is modelled as an interval $[d_{lb}]$ whose bounds are determined according to the lane-boundary sensor characteristics. Intervals are used to express the uncertainties of the information stored in the GIS database and measurement inaccuracies of the lane-boundary sensor. The cross-track vector is defined by $[\mathbf{u_{t,r}}]$ and the lane boundary is defined by $[d_{lb}]$. The search area is expanded on the horizontal plane defined by the road segment. The set-membership GNSS positioning location set $\mathbb{X}$ is then reduced after contraction which removes every location area not compatible with the lane-boundary sensor measurements:

$$\begin{aligned} \mathbb{X}_{\text{LB-ASGP}} = \{ & (x_r, y_r, z_r, co_r) \in \mathbb{R}^4 | \forall s = 1 \cdots m, \exists \rho_r^s \in [\rho_r^s], \\ & \exists (x^s, y^s, z^s) \in [\mathbf{x^s}], \exists \mathbf{u_{t,r}} \in [\mathbf{u_{t,r}}], \exists d_{lb} \in [d_{lb}], \\ & \left| \begin{array}{l} \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r \\ d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) \end{array} \right\} \\ \mathbb{Y}_{\text{LB-ASGP}} = (& [\rho_r^s], [\mathbf{x^s}], [\mathbf{u_{t,r}}], [d_{lb}]) \end{aligned} \tag{6.50}$$

Algorithm 9 resumes the proposed method (*LB-CASGP*), where the functions $\mathbf{f}$ to invert are given by:

$$\mathbf{f}_{\text{SGP}} = \left\{ \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r \right. \tag{6.51}$$

and

$$\mathbf{f}_{\text{LB-ASGP}} = \begin{cases} \rho_r^s = \|\mathbf{x^s} - \mathbf{x_r}\| + co_r \\ d_{lb} = \mathbf{u_{t,r}}^T \cdot (R_r - \mathbf{x_r}) \end{cases} \tag{6.52}$$

As observed in Figure 6.9, the LB-CASGP (**Algorithm 9**) solves simultaneously three problems, two set inversion problems (LB-ASGP and SGP) and the satellite specific corrections $[C_{cme}] = [C_{N,cme}; C_{E,cme}]$.

Lane-boundary constraints can further improve the performance of the set-membership GNSS location zone determination, if satellite-specific errors estimations obtained using lane-boundary are share among connected vehicles. *LB-CASGP* is a decentralized algorithm and it uses cross-track corrections shared among networked vehicles in order two improve the along-track vehicle position.

The component of the satellite-specific errors which causes the i^{th} receiver position estimate to suffer from an offset in the cross-track direction to the lane boundary ($\mathbf{C}_{\mathbf{i},\text{LB-ASGP}}$) is estimated in *LB-ASGP*. The $\mathbf{C}_{\mathbf{i},\text{LB-ASGP}}$ is obtained by taking into account only the cross-track component that has shifted the estimated position computed using the *SGP* algorithm to the estimated position computed using *LB-ASGP*. Each vehicle equipped with a lane-boundary sensor shares its correction to the networked vehicles. Given the position estimates of vehicle i given by the SGP algorithm $\mathbf{\hat{x}_{ri,SGP}} = [\hat{x}_{ri,SGP}, \hat{y}_{ri,SGP}, \hat{z}_{ri,SGP}]$ and by the *LB-ASGP* algorithm $\mathbf{\hat{x}}_{\mathbf{ri},\text{LB-ASGP}} = [\hat{x}_{ri,\text{LB-ASGP}}, \hat{y}_{ri,\text{LB-ASGP}}, \hat{z}_{ri,\text{LB-ASGP}}]$, the cross-track component error of a single vehicle is obtained by differencing both estimates and computing the vector along the cross-track direction:

$$\mathbf{C}_{\mathbf{i},\text{LB-ASGP}} = \mathbf{u_{t,ri}} \times \left(\left[\mathbf{x}_{\mathbf{ri},\text{LB-ASGP}} - \mathbf{x_{ri,SGP}}\right] \times \mathbf{P_{XY}}\right) \tag{6.53}$$

where $i = 1...n$ for n lane-boundary equipped vehicles.

The cooperative North (global Y-axis coordinates) and East (global X-axis coordinates) correction estimation *w.r.t.* to global coordinates, are respectively given by $C_{N,cme}$ and $C_{E,cme}$. They are computed using the least square estimation of all correction estimation vectors of each vehicle $\mathbf{C}_{\mathbf{i},\text{LB-ASGP}}$.

$$\begin{bmatrix} C_{N,cme} \\ C_{E,cme} \end{bmatrix} = (\mathbf{A^T A})^{-1}\mathbf{A^T} \begin{bmatrix} \mathbf{C}_{\mathbf{1},\text{LB-ASGP}} \\ \vdots \\ \mathbf{C}_{\mathbf{n},\text{LB-ASGP}} \end{bmatrix} \tag{6.54}$$

Algorithm 9 $[\hat{\mathbf{x}}_{\mathbf{r,2D,LB-CASGP}}; \mathbf{C}_{\mathbf{i,LB-ASGP}}] = CASGP\_LB$ $([\mathbf{f}_{\mathbf{SGP}}], [\mathbf{f}_{\mathbf{LB-ASGP}}], \mathbb{Y}_{SGP}, \mathbb{Y}_{LB-ASGP}, \mathbb{X}_0, \mathbf{C}_{\mathbf{net,LB-ASGP}})$

```
$\mathfrak{L} \leftarrow root(\mathbb{X}_0)$
if $\mathbf{f}_{\text{LB-ASGP}} \neq 0$ then
   $\mathbf{f}_{aux} = [\mathbf{f}_{\mathbf{SGP}}; \mathbf{f}_{\text{LB-ASGP}}]$
   $\mathbb{Y}_{aux} = [\mathbb{Y}_{SGP}, \mathbb{Y}_{\text{LB-ASGP}}]$
   $alg = 2$
else
   $\mathbf{f}_{aux} = [\mathbf{f}_{\mathbf{SGP}}]$
   $\mathbb{Y}_{aux} = [\mathbb{Y}_{SGP}]$
   $alg = 1$
end if
for jj=1 to alg do
   $\mathbf{f} \leftarrow \mathbf{f}_{aux}(jj)$
   $\mathbb{Y} \leftarrow \mathbb{Y}_{aux}(jj)$
   $[\underline{\mathbb{X}}, \Delta\mathbb{X}] = \mathrm{SIVIA}(\mathbf{f}, \mathbb{Y}, \mathfrak{L})$
   $\overline{\mathbb{X}} = \underline{\underline{\mathbb{X}}} + \Delta\mathbb{X}$
   $\mathfrak{L} \leftarrow \overline{\overline{\mathbb{X}}}$
   for i=1 to 4 do
      $X_{all}(i) \leftarrow$ equation (6.27)
   end for
   $\hat{co}_r \leftarrow$ equation (6.25)
   for k=1 to n do
      $C_{wf}(k) \leftarrow$ equation (6.28)
   end for
   for i=1 to 3 do
      $\hat{\mathbf{x}}_{\mathbf{aux,r}}(\hat{x}_{aux,r}, \hat{y}_{aux,r}, \hat{z}_{aux,r}) \leftarrow$ equation (6.26)
   end for
   $\hat{\mathbf{x}}_{\mathbf{r}}(\mathbf{jj}) = \hat{\mathbf{x}}_{\mathbf{aux,r}}$
end for
if $alg = 2$ then
   $\hat{\mathbf{x}}_{\mathbf{ri,2D},\text{LB-ASGP}} = [\hat{x}_{r,2D}, \hat{y}_{r,2D}] = \hat{\mathbf{x}}_{\mathbf{r}}(2) \times \mathbf{P}_{\mathbf{XY}}$
   $\mathbf{C}_{\mathbf{i},\text{LB-ASGP}} = \mathbf{u}_{\mathbf{t,ri}} \times ([\hat{\mathbf{x}}_{\mathbf{r}}(2) - \hat{\mathbf{x}}_{\mathbf{r}}(1)] \times \mathbf{P}_{\mathbf{XY}})$
else
   $\hat{\mathbf{x}}_{\mathbf{ri,2D,SGP}} = [\hat{x}_{r,2D}, \hat{y}_{r,2D}] = \hat{\mathbf{x}}_{\mathbf{r}}(1) \times \mathbf{P}_{\mathbf{XY}}$
end if
$[C_{cme}] = [C_{N,cme}; C_{E,cme}] \leftarrow$ equation (6.54)
if $alg = 2$ then
   $\hat{\mathbf{x}}_{\mathbf{ri,2D},\text{LB-CASGP}} = \hat{\mathbf{x}}_{\mathbf{ri,2D},\text{LB-ASGP}} + \mathbf{u}_{\mathbf{r,r}} \times [C_{N,cme}; C_{E,cme}; 0]$
else
   $\mathbf{x}_{\mathbf{ri,2D},\text{LB-CASGP}} = \mathbf{x}_{\mathbf{ri,2D,SGP}} + [C_{N,cme}; C_{E,cme}; 0]$
end if
($\mathbf{P}_{\mathbf{XY}}$: 3D projection to XY)
```

where $\mathbf{P_{XY}}$ is the projection matrix onto the XY plane, and $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} \mathbf{u_{t,r1}} \\ \vdots \\ \mathbf{u_{t,rn}} \end{bmatrix} \tag{6.55}$$

It was chosen to compute the $[C_{cme}]$ using least squares method, to avoid large amounts of data being transferred between vehicles, *i.e.* avoid the broadcast of all boxes belonging to solution set of each vehicle positioning using SGP and LB-ASGP, to further constraint each set. This decentralized solution requires minimal data transfer as it only requires the reception of $\mathbf{C_{i,\text{LB-ASGP}}}$ and $\mathbf{u_{t,ri}}$ from all lane-boundary sensor equipped vehicles.

Vehicles equipped with lane-boundary sensor have a good cross-track accuracy and a poor one in the along-track direction. Therefore, position estimates using *LB-CASGP* algorithm for vehicles equipped lane-boundary sensor, are only corrected (*w.r.t. LB-ASGP*) along the along-track component $\mathbf{u_{r,r}}$ using the shared corrections:

$$\begin{aligned} \mathbf{\hat{x}_{ri,2D}}{}_{,\text{LB-CASGP}} = \mathbf{\hat{x}_{ri,2D}}{}_{,\text{LB-ASGP}} + \\ \mathbf{u_{r,r}} \times [C_{N,cme}; C_{E,cme}; 0] \end{aligned} \tag{6.56}$$

For vehicles not equipped whit lane-boundary sensors, the application of the shared corrections into their own estimation process is done in both cross-track and along-track components:

$$\mathbf{\hat{x}_{ri,2D}}{}_{,\text{LB-CASGP}} = \mathbf{\hat{x}_{ri,2D,SGP}} + [C_{N,cme}; C_{E,cme}; 0] \tag{6.57}$$

When all vehicles are collinear no solution can be obtained since there is a singularity. In this situation, the position estimate of *LB-CASGP* is given by, $\mathbf{x_{ri,2D}}{}_{,\text{LB-CASGP}} = \mathbf{x_{ri,2D}}{}_{,\text{LB-ASGP}}$, *i.e.* the solution is given as if there was no cooperative information.

6.2.6 Lane Boundary Relaxed Set-membership Satellite NLOS Multipath Fault Detection and Exclusion (LB-RSSMFDE)

Any measure that does not meet the error assumptions taken initially are called aberrant. Such aberrant measure may be due to a malfunction of the measuring device or due to an exceptional disturbance of the model describing the process. Multipath disturbed signals are examples of aberrant measures, they have a strong effect on position estimates. NLOS multipath occurs when the direct signal is blocked and only a reflected signal is received (see Figure 6.10) . The ranging measurement errors that result from NLOS multipath reception are particularly common in cities where dense urban areas form urban canyons that block the signals. This type of disturbance is not corrected by most multipath mitigation techniques. Therefore, to improve positioning in urban areas it is necessary to detect and remove satellites responsible for the multipath [109].

In conventional external integrity methods (see chapter 4 of [75]), the level of uncertainty increases as the residues increase (see equation 3.24). On conventional methods position computation does not take into account the possible presence of a defect, it is assumed that an increased residue is related to the presence of an aberrant measure, and its impact on the solution is quantified [110].

The set-membership method has a completely different approach. More inconsistency between measurements increases, the volume of the whole solution is reduced, becoming empty in case of total inconsistency. In other words, in conventional methods more inconsistency between measurements, the greater the level of uncertainty increases, in set-membership method the set is reduced as the measures are not consistent.

In order to make set-membership method robust against aberrant measurements a method called constraint relaxation is applied. Relaxation means that instead of returning the set of solutions compatible with all measurements, it is considered the set of solutions compatible with all $m - q$ measurements [111], where m is the number of measurements and q is the number acceptable aberrant measurements (see Figure 6.11).

The method SIVIA was extended by V.Drevelle [110], enabling a robust estimation of sets in the presence of a q number of outliers among the m considered measurements (Robust Set Inversion Via Interval Analysis (RSIVIA)).

Considering m sets $\mathbb{X}_1, \ldots, \mathbb{X}_m$ on $\mathbb{R}^n$. It is possible to define their q-relaxed intersection, denoted $\bigcap^{\{q\}} \mathbb{X}_i$ as the set of points $x \in \mathbb{R}^n$ belonging to at least $m - q$ sets $\mathbb{X}_i$ (see Figure 6.11). Therefore if $q = 0$, the resulting set corresponds to the intersection of all sets, while if $q = m - 1$ then the resulting set corresponds to the union of all sets.

A robust method Guaranteed Minimum Outlier Number Estimator (GOMNE) [112] consists in adaptively relaxing a growing number of constraints, as long as the solution set is empty. In this work the GOMNE algorithm was adapted to detect NLOS multipath aberrant GNSS measurements, aided by lane boundary measurements and a 2D georeferenced road map, this method is entitled as LB-RSSMFDE.

The LB-RSSMFDE process start by assigning $Nsat_{removed} = 0$, and computing a solution set with all satellite measurements m minus $Nsat_{removed}$. If the previous step results in a empty set, then it is computed a solution set excluding one ($Nsat_{removed} = 1$) satellite measurement at a time. If no solution set is achieved after going through all possible combinations of satellites where only one satellite measurement is excluded at a time $Nsat_{removed} = 1$, then the number of satellites excluded at a time is increased $Nsat_{removed} = 2$, and a search for a solution set is done by going through all possible combinations of satellites where only two satellite measurement are excluded at a time.

The process is repeated until a non-empty solution is obtained, then the corresponding excluded satellites are considered aberrant measurements (*NLOS satellites*), and the

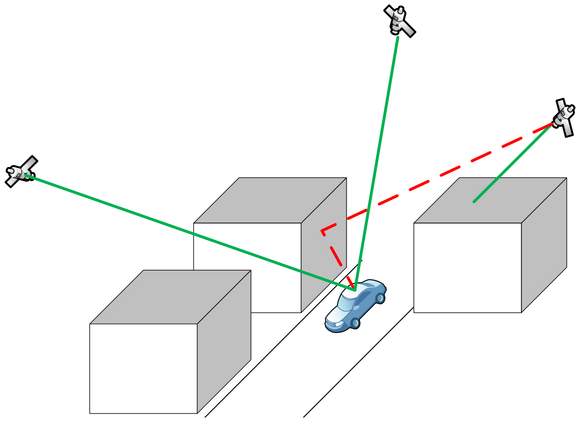


Figure 6.10: NLOS multipath scenario, where the direct signal (solid line) is blocked and the signal is received only via reflection (dashed line).

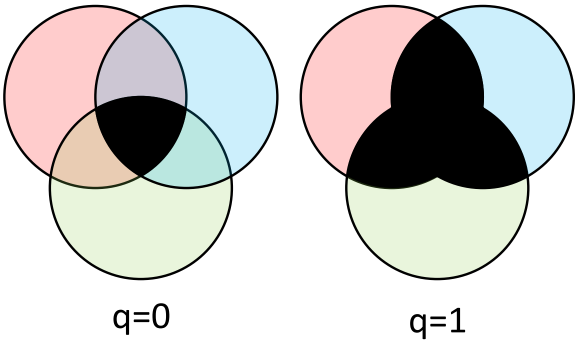


Figure 6.11: q-relaxed intersection with $m = 2$ and $q = \{0, 1\}$.

solution set is returned. This strategy avoids returning an empty set in the presence of aberrant measures, while not needing to overestimate the value of q.

By removing the aberrant measurements the satellite list $\mathbf{x^s_{NLOS}}$ is therefore rewritten:

$$\mathbf{x^s_{NLOS}} = drop_i\left(\begin{bmatrix}\mathbf{x^{s,1}_{NLOS}} & \dots & \mathbf{x^{s,i}_{NLOS}} & \dots & \mathbf{x^{s,m}_{NLOS}}\end{bmatrix}^T\right) \wedge i = NLOS\ satellite \quad (6.58)$$

where the "drop" notation is used to represent the removing of a row from a matrix. Similarly the faulty pseudoranges are removed from the position computation, hence $\rho_{r,NLOS}$ is given by:

$$\rho_{r,NLOS} = drop_i\left(\begin{bmatrix}\rho_r^1 & \dots & \rho_r^i & \dots & \rho_r^m\end{bmatrix}^T\right) \wedge i = NLOS\ satellite \quad (6.59)$$

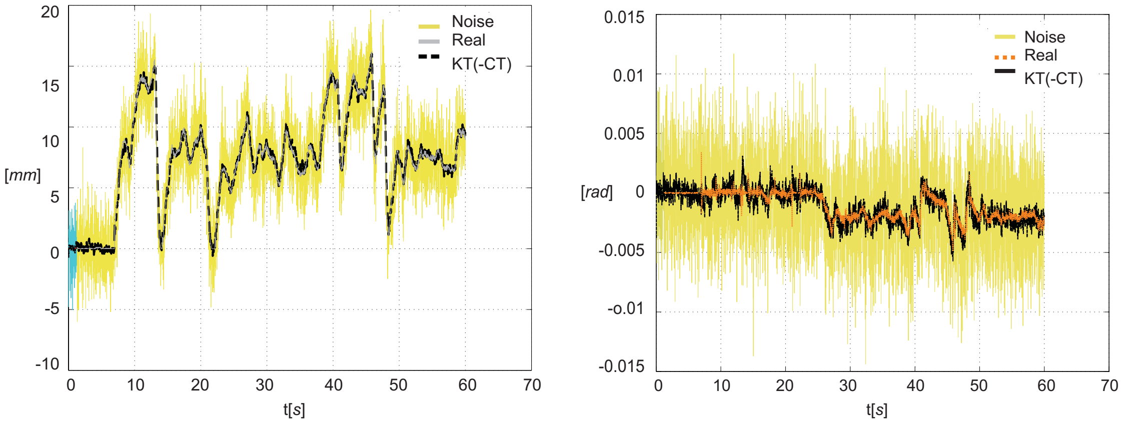
Using vectors, $\rho_{r,NLOS}$ and $\mathbf{x^s_{NLOS}}$, and applying RSIVIA instead of SIVIA in the LB-ASGP and LB-CASGP, provides a robustness characteristic to the implemented algorithms and therefore these are renamed to LB-RASGP and LB-RCASGP respectively. The LB-RASGP and LB-RCASGP algorithms were relaxed for $q = 1$ number of the faulty measurements tolerated in the set computation, *i.e.* 1-relaxed set-membership solution (see section 3.3.2).

The LB-RSSMFDE algorithm is used to determine the number $Nsat_{removed}$ aberrant measurements and which satellites are providing aberrant measures. Once remove the aberrant satellite measurements, the solution is computed using a 1-relaxed set-membership method in order to be robust with an undetected aberrant measure. This approach similar to the methods conventional external integrity methods (see chapter 4 of [75]), where the detection step and fault exclusion is repeated until no more fault is detected, then external integrity levels are calculated as assuming one undetected fault.

6.3 Results

6.3.1 Kinematic sensors Fusion with Confidence Tests (KF-CT)

The results presented in this section were used to assess the performance of KF-CT algorithm, described in section 6.1.1. A front steered vehicle is simulated and odometry errors were included.

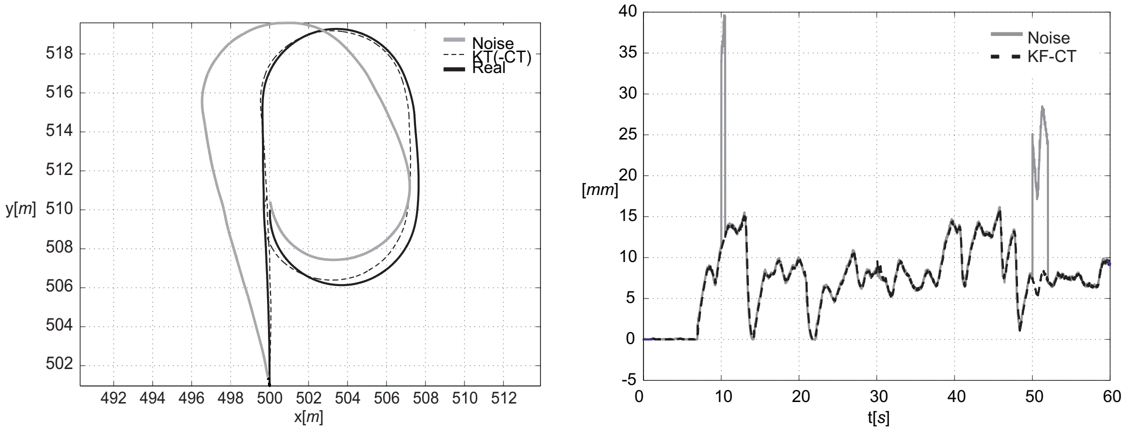


(a) (b)

Figure 6.12: Odometry based position estimation with 10dB of signal-to-noise ratio gaussian white noise on wheel encoders and steering wheel encoder: (a) Δ_R; (b) ω_R. Where: Real - encoders measures; Noise - measures with Gaussian white noise; KF(-CT) - estimation without using confidence tests.

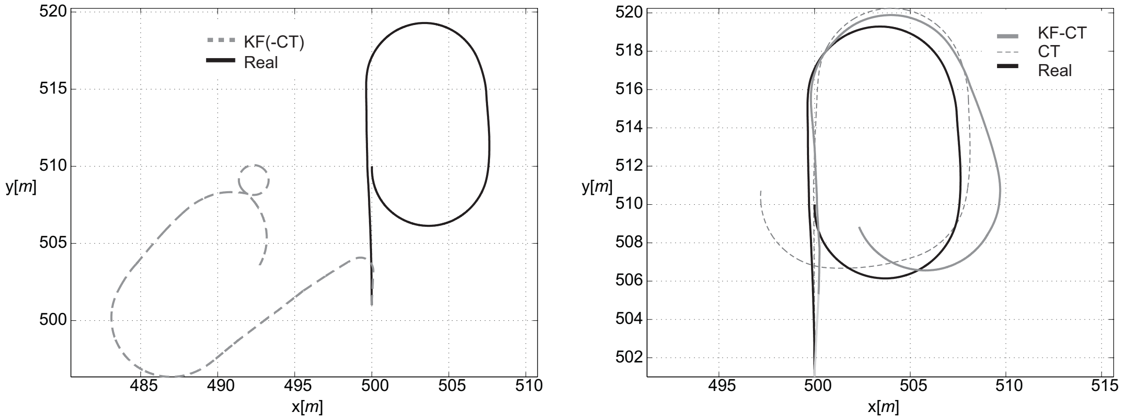
Real data measurements, gathered from vehicle encoders moving along a closed path, were used in the reported simulations (path depicted in Figure 6.13(a)). White Gaussian noise with a signal-to-noise ratio of 10dB, were added to the wheel encoders and steering wheel encoder measurements. The qualitative behavior of the Kinematic sensors Fusion without Confidence Tests (KF(-CT)) (*i.e.* KF-CT without the confidence tests) is very satisfactory in normal road conditions of adherence, *i.e.* the KF(-CT) efficiently coped with the added noise, see Figure 6.12. This performance is well illustrated in Figure 6.13(a), where the trajectory computed using the KF(-CT) is closely similar to the real trajectory, while the trajectory computed from the raw noisy measurements diverges significantly.

However this odometric model does not solve completely the problem inherent to the



(a) (b)

Figure 6.13: KF(-CT) and KF-CT performance evaluation: (a) KF(-CT) position estimation with 10dB of signal-to-noise ratio gaussian white noise on wheel encoders and steering wheel encoder; (b) KF-CT used for Δ_R estimation with simulated slippage injected at $t = 10s$ and $t = 50s$. Where: Real - encoders measures; Noise - measures with Gaussian white noise; KF(-CT) - estimation without using confidence tests; KF-CT - estimation using confidence tests.



(a) (b)

Figure 6.14: Odometry based position estimation with simulated slippage injected at $t = 10s$ and $t = 50s$: (a) position estimation using KF-CT without confidence tests; (b) position estimation using KF-CT; Where: Real - encoders measures; Noise - measures with Gaussian white noise; KF(-CT) - estimation without using confidence tests; CT - position estimation using only the confidence tests and excluding any contribution of the EKF; KF-CT - estimation using confidence tests.

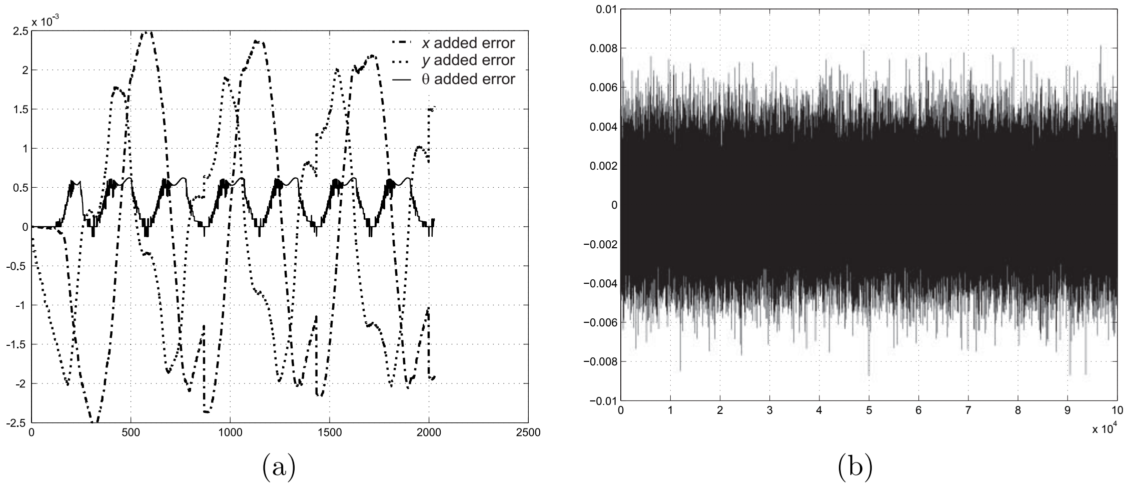
slippage. If a big slippage occurs, the KF(-CT) will not eliminate its effects, specially if more than one wheel slips. This effect can be easily seen in Figure 6.14(a), where the estimated position using the KF(-CT) is highly affected by both $t = 10s$ and $t = 50s$ slippage.

The previous disturbance can be attenuated by pre-processing the redundant data before provide it to the odometry module. The contribution to Δ_R of the slippage injected at $t = 10s$ and $t = 50s$, is mostly removed (see Figure 6.13(a)). As we can see from Figure 6.13(b), the EKF with this data pre-processing (*i.e.* KF-CT), will not follow the slippage. The KF-CT algorithm detects a disturbed encoder measurement and replaces it by a virtual encoder measurement, computed based on the non-disturbed encoder measurements.

As shown in Figure 6.14(b), the position estimation is greatly improved by using the KF-CT, being almost insensitive to both Gauussian white noise and slippage events. Even if the position estimation is using only the confidence tests (CT) and excluding any contribution of the EKF, its performance has a major improvement *w.r.t.* using the raw encoder measurements (see Figure 6.14).

6.3.2 Vehicle pose estimation using magnetic landmark

The results presented in this section show the effectiveness of vehicle pose estimation using magnetic landmark. In the reported simulations two types of disturbances are considered: systematic errors and Gaussian sensors measurement noise.



(a) (b)

Figure 6.15: KF-CT+MAG results: (a) systematic noise added to (x_{k+1}, y_{k+1}, θ_{k+1}), when Δ_R is multiplied by a noise factor K_{se}, i.e. $\Delta_{R,WithNoise} = \Delta_R \times K_{se}$, in meters; (b) Gaussian white noise (C_{onm}) added to the (x_{k+1}, y_{k+1}, θ_{k+1}), in meters for both x and y and in radians for θ;

These perturbations model several types of noises:

- Systematic errors - simulate incorrect size on the wheels radius (K_{se});
- Gaussian white noise error - emulate odometers readings noise (C_{onm}).

Regarding the systematic errors, they were applied in the process by multiplying arc length Δ_R with a K_{se} factor, this error simulates a erroneous wheel radius, *i.e.* the wheel is bigger:

$$\Delta_R = \frac{\Delta_{RR} + \Delta_{RL}}{2} \times K_{se} \tag{6.60}$$

where $K_{se} = 1.03$. Δ_R will therefore perturb the pose $(x_{k+1}, y_{k+1}, \theta_{k+1})$ obtained from the odometry computation, yielding

$$\begin{cases} x_{k+1} = x_k + \Delta_R \times K_{se} \times \cos(\theta_k + \omega/2) \\ y_{k+1} = y_k + \Delta_R \times K_{se} \times \sin(\theta_k + \omega/2) \\ \theta_{k+1} = \theta_k + \omega_R \end{cases} \tag{6.61}$$

The magnitude of the disturbance introduced by systematics errors *w.r.t.* the vehicle's pose (x, y, θ^{cp}) is displayed in Figure 6.15 (a).

Gaussian white noise, denoted by C_{onm}, was added to the odometry equations, this disturbance represent odometers noisy measurements (see Figure 6.15 (b)), resulting in:

$$\begin{cases} x_{k+1} = x_k + \Delta \times K_{se} \times \cos(\theta_k + \omega/2) + C_{onm} \\ y_{k+1} = y_k + \Delta \times K_{se} \times \sin(\theta_k + \omega/2) + C_{onm} \\ \theta_{k+1} = \theta_k + \omega + C_{onm} \end{cases} \tag{6.62}$$

In real environments the detection of the magnets doesn't return the exact center of the magnet, but rather a coordinate close to its z-axis center, therefore in order to have realistic simulated measures, a representative model of the magnetic field radiated by the magnetic marker was used in simulations (see Figure 6.16).

The process noise covariance matrix Q and the measurement noise covariance matrix R, used in the KF-CT+MAG are shown below:

$$\mathbf{Q} = \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^5 & 0 \\ 0 & 0 & 10^5 \end{bmatrix} \; ; \; \mathbf{R} = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \tag{6.63}$$

The simulation results of the pose estimation model while using KF-CT+MAG algorithm is presented in Figure 6.17 (b), and they are compared with pose estimation model using KF-CT presented in Figure 6.17 (a). The KF-CT+MAG algorithm achieves a good performance.

The controller drives the vehicle along a predefined path, as it moves towards its local goal it is unaware of the accumulating errors due to the disturbances introduced. Therefore

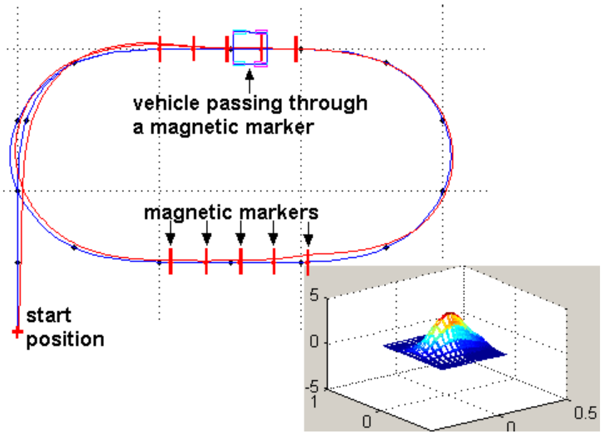
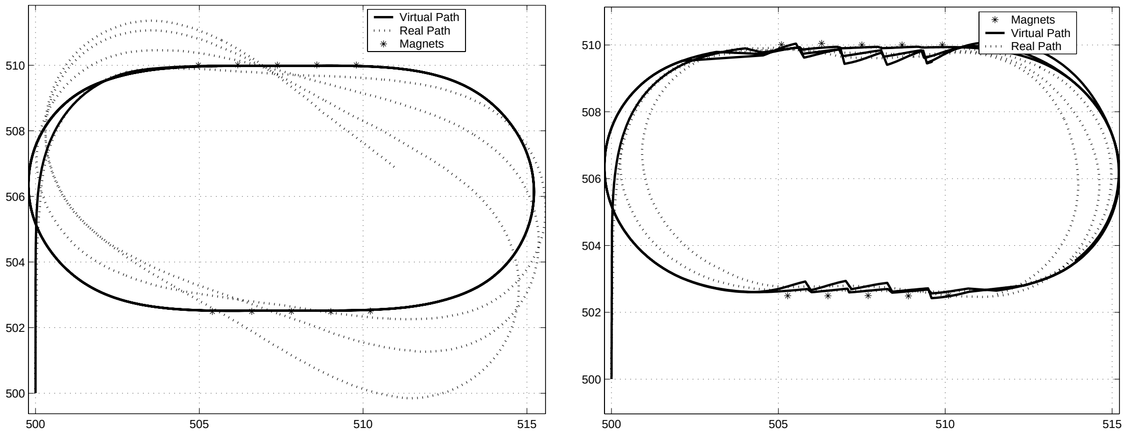


Figure 6.16: KF-CT+MAG simulated magnetic vertical field results: the lower right image displays the simulated magnetic vertical field component of a magnetic marker being detected by the vehicle front sensors array, ten magnetic markers were placed in the test loop



(a) (b)

Figure 6.17: Path following simulation with KF-CT+MAG position estimation: (a) Odometry results without using the magnetic markers as correction landmarks KF-CT; (b) Odometry results using the magnetic markers as correction landmarks KF-CT+MAG. Where: Real Path - real path followed by the vehicle; Virtual Path - path thought to be followed by the vehicle; (in meters).

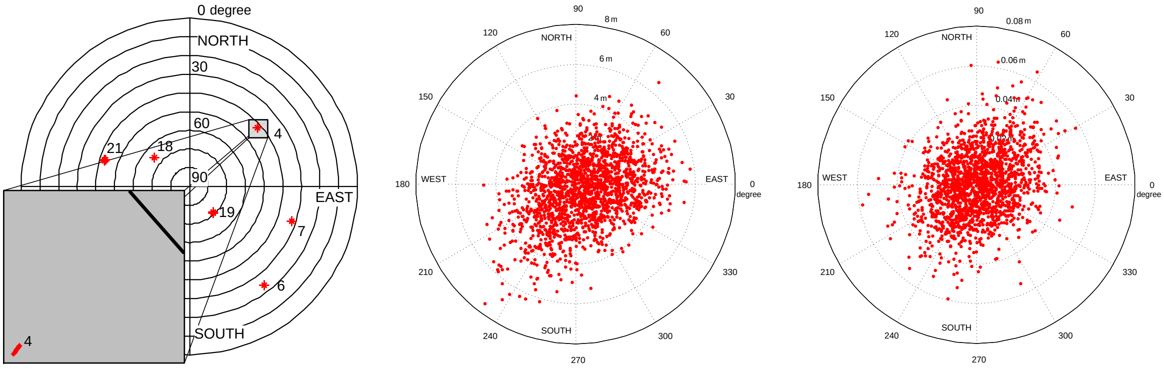
the vehicle continues as if it was following accurately the reference path. If while navigating, no correction is done, the path followed (Real Path) is significantly different, from the one (Virtual Path), on which the PFC is generating its commands . This effect is observed in Figure 6.17 (a)

In Figure 6.17 (b) the KF-CT+MAG handles disturbances by using the magnets located on the ground at the marked points on the figure. Although the errors are accumulated during the curves on the straight lines it recovers by using the detected magnets. The position in which the controller actuates is the virtual (Virtual Path) since it has no means of knowing about the accumulated errors. Every time it detects a magnet the virtual position is updated, moving towards the real position (Real Path), meaning that the KF-CT+MAG provides a good estimate. Therefore the virtual position is identical to the real position every time a magnet is found as it can be observed in Figure 6.17 (b) .

If false detection of magnets events occur, either coming from hardware anomaly or from incorrect positioned magnets, the KF-CT+MAG was also able to use discard them in the *Data Association* process of the KF-CT+MAG algorithm.

6.3.3 Inter-vehicle pose estimation using laserscanner data and magnetic landmark via V2X

This section presents the VPE results obtained using LIDAR, MSS and V2X, contributing altogether to a more insensitive to disturbance vehicle pose estimation. Regarding the GPS disturbances they were modelled and accounted in the simulation process. The daily behavior of the ionospheric delay was simulated using a half cosine function of the local time during daytime and by a constant level during nighttime, scaled by a satellite elevation factor. The average ionospheric injected error is 4 meters, no scintillation events were introduced and the daytime total electron content is bounded by $[4 \times 10^{17}; 1.6 \times 10^{18}]$. The simulated tropospheric delay, ranges from 3 meters for a satellite at zenith to 25 meters for a satellite at 5 degrees elevation. The multipath error was simulated for all satellites and receivers, the code pseudorange and carrier-phase pseudorange multipath errors have respectively 1.6 and 0.2 meters of standard deviation. The standard deviation of the thermal noise is 1 meter for code pseudorange and 1 centimeter for carrier-phase pseudorange.

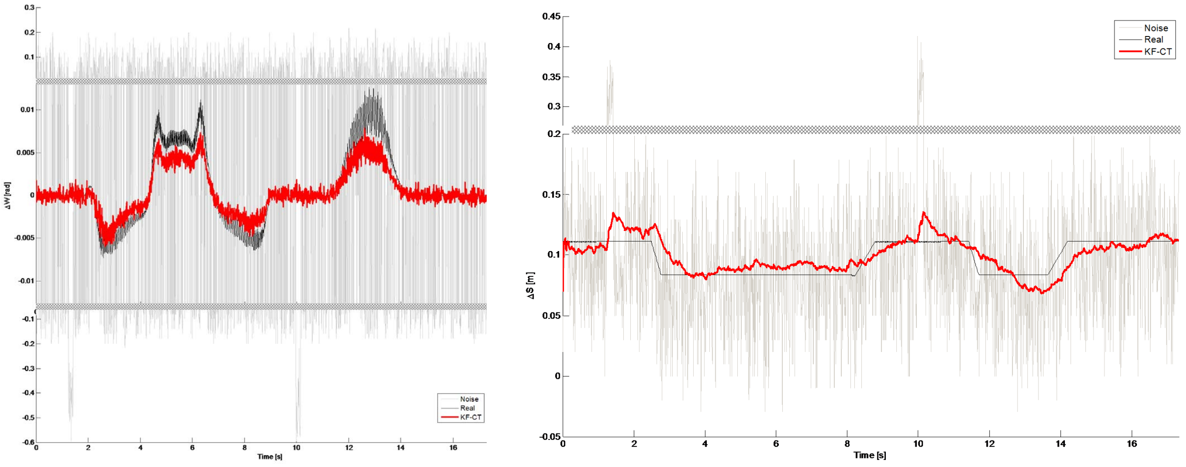


(a) (b) (c)

Figure 6.18: Satellite skyplot and positioning error: (a) satellite constellation (on the highlighted area it is depicted the sky path of satellite number 4 during simulation time interval); (b) pseudorange code positioning error; (c) double-differences carrier-phase pseudorange positioning error.

Figure 6.18 (a) shows the satellite constellation used during simulations, in the highlighted area one can observe the satellite number 4 path during the simulated time interval. Figure 6.18(b) and (c) present the positioning error of a static receiver using iterative least squares method for code pseudorange and double-differences carrier-phase pseudorange respectively. The double-differencing computation uses the highest elevation satellite for pivoting and it is assumed that the ambiguity resolution was already achieved and no cycle slip occurred. A satellite mask angle of 15 degrees was used and no smoothing of either code or carrier-phase was used. As expected, the accuracy of the double-differences positioning is centimetric while for the code pseudorange positioning is on the meter scale.

Several disturbances where considered on the simulations related with the fusion of odometry and landmarks. The simulated wheel radius is 0.32 meters and each of the four wheels is equipped with a 200 pulses per revolution encoder, to each wheel radius an independent 0.02 meters variance Gaussian white noise was added and to the encoders besides the quantization error, a 5 pulses variance integer Gaussian white noise was added. The simulated steering wheel is equipped with a 4096 pulses encoder and has 64 degrees from lock to lock front wheel steering angle, a 0.01 radians variance Gaussian white noise was added and to the encoder besides the quantization error, a 10 pulses variance integer Gaussian white noise was added. The odometry was also contaminated with two independent 0.01 meters variance Gaussian white noise on the wheel to wheel distance and on the front to rear axle distance.



(a)

(b)

Figure 6.19: KF-CT disturbances analysis: (a) and (b) respectively ω_R and Δ_R estimation with a simulated slippage (Noise - noisy encoders measures with simulated slippage at t=1.8s and t=10s; Real - ω_R and Δ_R; KF-CT - estimation with EKF and Confidence Tests).

The magnitude of the disturbance in the vehicle's pose is displayed in Figure 6.19 (a) and (b), where besides the magnitude of the introduced error it is possible to observe a relatively low difference between the real ω and Δ_R and the estimates $\widehat{\omega_R}$ and $\widehat{\Delta_R}$. The

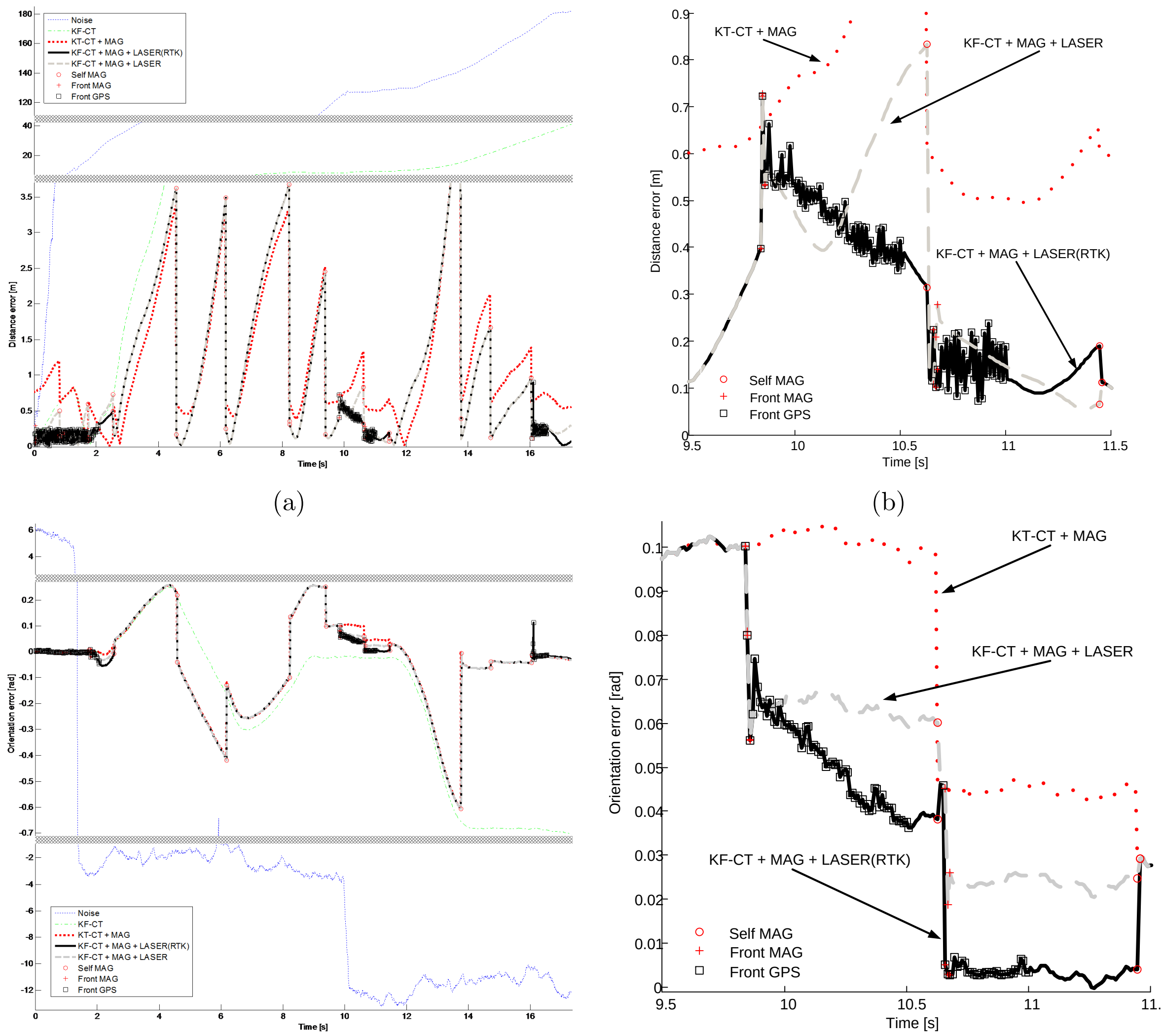
	Euclidean Error		Orientation Error	
	RMSE [m]	Average [10^3 m]	RMSE [°]	Average [°]
EKF-CT	10.54	18.26	13.85	419.93
EKF-CT+MAG	1.18	2.05	7.38	223.36
EKF-CT+MAG+LASER(RTK)	1.03	1.79	7.29	220.78
EKF-CT+MAG+LASER	1.06	1.83	7.36	222.87

Table 6.1: Root Mean Square Error RMSE and Average Error for the various pose estimation methods.

good estimation performance is due the Confidence Tests encoders slippage removal and to the EKF efficiency, enabling the method KF-CT to cope with both wheel slippage and Gaussian white noise. Both the MSS and LIDAR were simulated with quantization disturbances and 0.02 meters variance Gaussian white noise on both axes.

Results shown in Figures 6.20 and Table 6.1 testify the efficiency of the proposed algorithm, both Euclidian distance error and orientation error are generally smaller using the KF-CT+MAG+LASER(RTK) method. On the zoomed Figures 6.20(b) and (d) one can observe greater error reduction using the corrections made by the KF-CT+MAG+LASER(RTK) (*i.e* KF-CT+MAG and RTK-GPS fusion of LIDAR detected preceding vehicle using V2X communications), than the corrections made by KF-CT+MAG+LASER (*i.e* KF-CT+MAG fusion of LIDAR detected preceding vehicle using V2X communications). It was assumed that the LIDAR cannot provide preceding vehicle orientation, therefore the above mention corrections can only be made when the difference between vehicles' estimated orientation is very small. When the vehicles' orientation differ by a considerable amount, the previous restriction leads to both KF-CT+MAG+LASER(RTK) and KF-CT+MAG+LASER reaching the same level of accuracy as the KF-CT+MAG.

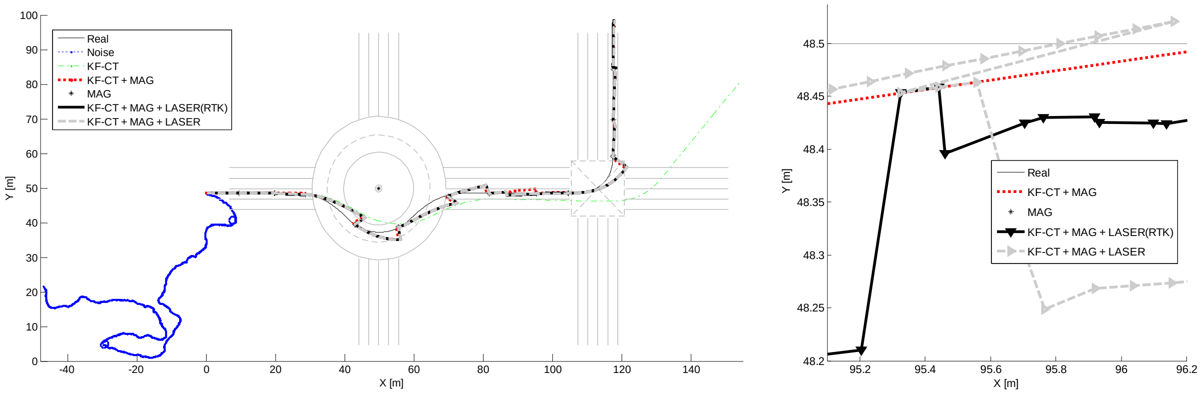
In Figure 6.21(a) the trajectory estimation is presented. It is easily observed that if no correction is done, the presumed position (Virtual Path) is much different from the real path followed. All fusion algorithms handle well the disturbances but the algorithms using absolute position based corrections; either by the vehicle detecting magnets or by using information of the preceding vehicle detected magnets or RTK-GPS; handle with them better. On the zoomed Figure 6.21(b) is appears that the KF-CT+MAG+LASER has a better performance than the KF-CT+MAG+LASER(RTK) due to being closer to the real followed path, but it is not the case since the estimated path is ahead in time and when a magnetic based correction is performed, the KF-CT+MAG+LASER updated position moves backwards by a greater amount than KF-CT+MAG+LASER(RTK) updates it position laterally (one should note that in Figure 6.21(b) the y-axis uses a different scale than the x-axis).



(a) (b)

(c) (d)

Figure 6.20: KF-CT + MAG + LASER(RTK) error analysis: (a) and (c) show, respectively, the Euclidian distance error and orientation error with a simulated slippage; (b) and (d) show, respectively, the Euclidian distance error and orientation error zoomed area between following time interval $[9.5, 11.5](s)$. Where: Noise - pose estimation using noisy encoders measures with a simulated slippage at t=1.8s and t=10s; KF-CT - pose estimation using KF-CT algorithm; KF-CT+MAG - pose estimation using using KF-CT+MAG and in vehicle magnets absolute position correction; KF-CT+MAG+LASER(RTK) - pose estimation using KF-CT and RTK-GPS fusion of LIDAR detected preceding vehicle using V2X communications; KF-CT+MAG+LASER - estimation with KF-CT+MAG fusion of LIDAR detected preceding vehicle using V2X communications communication and preceding vehicle magnet detection; Self MAG - it marks where a correction took place using its own MSS; Front MAG - it marks where a correction took place using preceding vehicle MSS; Front GPS - it marks where a correction took place using preceding vehicle RTK-GPS.



(a) (b)

Figure 6.21: Path following simulation using KF-CT+MAG+LASER(RTK) position estimation: (a) and (b) are respectively trajectory and zoomed trajectory followed with a simulated slippage. Where: Real - real pose; Noise - pose estimation using noisy encoders measures with a simulated slippage at t=1.8s and t=10s; KF-CT - pose estimation using KF-CT; KF-CT+MAG - pose estimation using KF-CT and in vehicle magnets absolute position correction; KF-CT+MAG+LASER(RTK) - pose estimation using KF-CT+MAG and RTK-GPS fusion of LIDAR detected preceding vehicle V2X communications; KF-CT+MAG+LASER - pose estimation using KF-CT+MAG fusion of LIDAR detected preceding vehicle using V2X communications communication and preceding vehicle magnet detection; MAG - magnet position;).

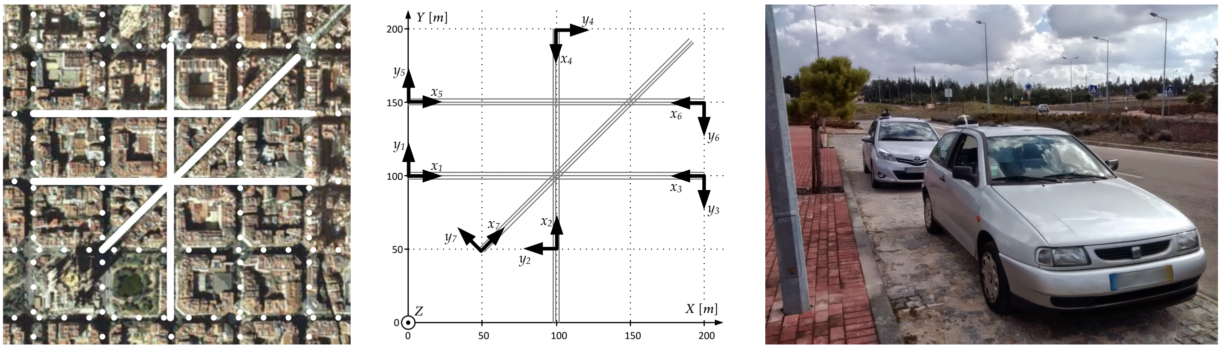
6.3.4 Robust and Set-membership based GNSS positioning algorithms

To analyze the performance of the proposed positioning algorithm, simulations and real experiments are reported.

Simulation Setup

Seven vehicles traveling on an urban road network have been simulated (see Figure 6.22-6.23). Each vehicle is marked with a pair of coordinate axes indicating local along-track and cross-track directions. All vehicles are equipped with V2V communications, but only vehicles 1-6 are equipped with lane-boundary sensor.

The GPSoft [76] software was used to emulate the GNSS system, namely the USA GPS constellation [75]. The GPSoft Toolbox emulates not only satellites and receivers but also the propagation channels. Error sources such as thermal noise, multipath, atmospheric delays and Selective Availability are modelled as an integral part of pseudorange and integrated Doppler emulation. Furthermore, the errors are emulated such that the proper temporal and spatial correlation effects are observed in the measurements. This allows for realistic modelling of both *code* DGPS and *carrier-phase* RTK-GPS in addition to usual stand-alone positioning algorithms. GPSoft also enables emulation of Galileo, GEOs, GPS and GPS Modernization (C/A-code on L1, L2 and L5) as well as dual-frequency P-code measurements. The user can emulate signals on additional carrier frequencies defined by



(a) (b) (c)

Figure 6.22: Set-membership based algorithms simulation and experimental setup: (a) simulated road network; (b) Simulation environment - 7 vehicles simulated road network with starting vehicle positions and headings, vehicle x-axis and y-axis represent along-track and cross-track directions respectively; (c) Real Experiment - instrumented vehicle used in the tests.

the user. The satellite constellation emulator supports GPS and Glonass as well as user-defined constellations. In addition, YUMA-format broadcast almanacs can be used. The emulation of C/A and P-code pseudorange and integrated Doppler with user definable civil and military carrier frequencies is available including characteristics such as thermal noise, ionospheric delay, tropospheric delay and diffuse multipath [76].

The signals coming from different satellites is made distinguishable using pseudo-random noise (PRN) sequences modulation of the carriers. This modulation is divided in two binary sequences: coarse/acquisition (C/A) code with a chipping rate of 1.023 MHz and precise (P) code with a chipping rate of 10.23 MHz. There are two main carriers:

- *L1* carrier phase at 1575.42 MHz, modulated by both C/A and P codes;
- *L2* carrier phase at 1227.60 MHz, modulated by P code.

In this work, the satellite-receiver distance from the satellite s to the receiver r is measured using *C/A-L1* code. As mentioned earlier, GNSS pseudoranges are affected by several types of error. The daily behavior of the ionospheric delay has been emulated using a half cosine function of the local time during daytime and by a constant level during nighttime, scaled by a satellite elevation factor. The average ionospheric injected error is 4 meters. No scintillation events have been introduced and the daytime total election content is bounded by $[4 \times 10^{17}; 1.6 \times 10^{18}]$. The emulate tropospheric delay ranges from 3 meters for a satellite at zenith to 25 meters for a satellite at 5 degrees elevation. A white noise is passed through a first-order Butterworth low-pass filter to generate the code diffuse multipath error of zero-elevation angle, which is scaled by the cosine of the true satellite elevation angle before it is applied to the range measurement. The standard deviation of pseudorange diffuse multipath errors at zero-elevation is 1.6 meters. A different

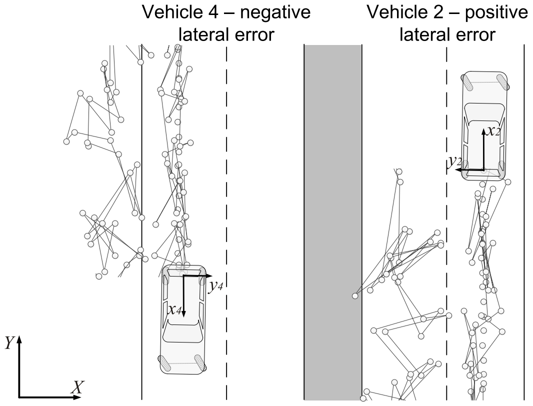

Vehicle 4 – negative lateral error
Vehicle 2 – positive lateral error
x_2
y_2
y_4
x_4
Y
X

Figure 6.23: Simulation - Satellite constellation configuration major error axis: Zoomed trajectory and position estimation for vehicles 2 and 4. Vehicles 2 and 4 travel in a road lane parallel to the *Y-axis* local coordinates system Σ. Vehicle 2 is rotated by $\pi/2$ with respect to Σ. Vehicle 4 is rotated by $3\pi/2$ with respect to Σ. The two vehicle's *y-axis* have opposite directions, leading to a positive lateral error for vehicle 2 and a negative lateral error for vehicle 4 in each vehicle local frame. This satellite constellation configuration generates an error with the major axis along the negative direction of the Σ *X-axis*.

Table 6.2: Pseudorange intervals risk r and α error bounds as a function of the number m of measurements

m	4	5	6	7	8	9	10	11
r	$0.25 \cdot 10^{-4}$	$0.20 \cdot 10^{-4}$	$0.17 \cdot 10^{-4}$	$0.14 \cdot 10^{-4}$	$0.13 \cdot 10^{-4}$	$0.11 \cdot 10^{-4}$	$0.10 \cdot 10^{-4}$	$0.09 \cdot 10^{-4}$
α	4.21	4.26	4.31	4.34	4.37	4.39	4.42	4.44

uncorrelated diffuse multipath error is generated for each satellite and receiver [75]. The standard deviation of the thermal noise is 1 meter.

For a global risk $\mathbf{R} = 10^{-4}$ and given the current number m of measurements, the pseudorange intervals risk r and the error bounds α are computed thanks to equation 3.58 and 3.62. The intervals $[\rho_r^s] = [\rho_r^s - \alpha\sigma, \rho_r^s + \alpha\sigma]$ are given in Table 6.2 for up to 11 satellites. The lane-boundary sensor error was assumed to have a standard deviation of $\sigma_{lb} = 0.25m$ and the lane-boundary interval was set to contain 95% of the measurements, *i.e.* $2 \times \sigma_{lb}$.

The satellite constellation setup produced a bias towards negative direction of the local coordinates *X-axis* (denoted by $---$) and a very small bias towards the positive direction of the local coordinates *Y-axis* (denoted by $+$). Therefore this satellite constellation configuration generates an error with the major axis along the negative direction of the local coordinates *X-axis*, see Figure 6.23.

Table 6.3 presents a qualitative evaluation of the satellite constellation setup bias on each vehicle coordinates, *e.g.* vehicle 1 axis is aligned with local coordinates, therefore the satellite constellation setup bias on vehicle 1 coordinates has the same direction and

Table 6.3: Satellite constellation bias on vehicle coordinates and vehicle orientation

Vehicle Coordinates	Vehicles						
	1	2	3	4	5	6	7
X-axis	−−−	+	+++	−	−−−	+++	−
Y-axis	+	+++	−	−−−	+	−	+++
θ_r	0	$\pi/2$	π	$3\pi/2$	0	π	$\pi/4$

signal as local coordinates.

Experimental Setup

Standard road vehicles equipped with the same experimental setup have been used on the test see Figure 6.22(c). The used test site allows to define a huge variety of paths. The surrounding environment is made of trees and buildings. Four *ublox LEA-6T* have been used as embedded GPS-receivers. This kind of receiver enables easy vehicle integration, have a standard communication interface and provides raw-data which is necessary for the pseudoranges processing. The ground-truth setup was a high performance RTK-GPS system *TOPCON HiperPro*, able to provide positioning solutions with centimeter accuracy. The MAFS used by the RTK-GPS used data obtained through a *SERVIR* project facility which consists on a military network of permanent reference GNSS stations capable of providing raw-data observations and corrections for real-time RTK-GPS or post-processed PPK. During the experiments, one of these stations (*SERVIR - Station 9*) gathered all the necessary conditions to be used as MAFS: short distance to the test site, no multipath or electromagnetic interference sources nearby and no signal obstruction caused by trees or higher buildings. The lane-boundary measurements were obtained using the RTK-GPS with a 25 cm additional white Gaussian white noise.Therefore, the global accuracy of the lane boundary measurement is in the order of the accuracy of a classical lane detection camera [113].

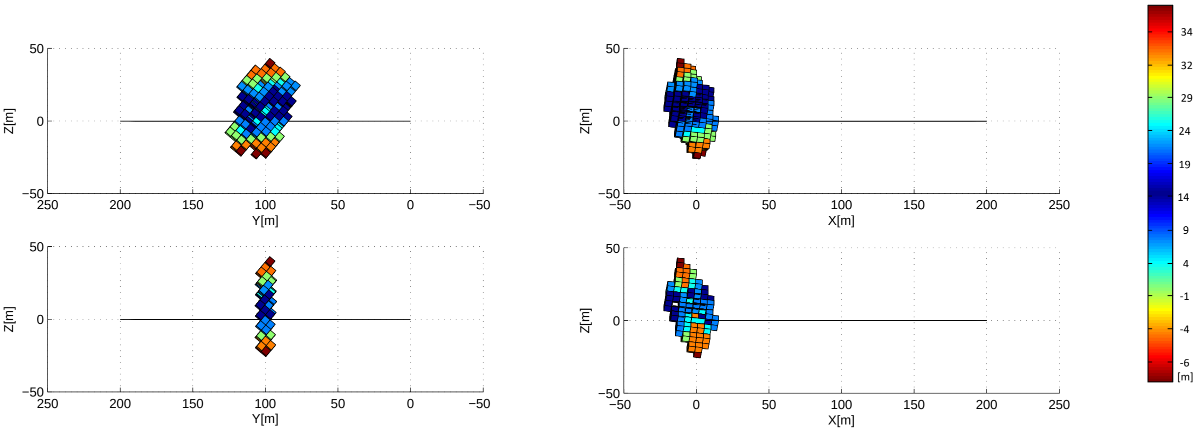
6.3.4.1 Set-membership GNSS positioning (SGP)

When using set-membership GNSS method, an important focus is on the characterization of domains which contain the solution rather than the search of a punctual result which might be misleading and with no associated confidence information. In this work, the unknown variables are (x_r, y_r, z_r, co_r). The initial searching volume was set to $27 \times 10^6 [m^3]$ which is an arbitrarily high value with little impact on the processing time.

The top row of Figures (6.24-6.25) present the enveloping box for the position of vehicle 1, located at $(x_{r1}, y_{r1}, z_{r1}) = (0, 100, 0)[m]$, when the constraints provided by the satellites constellation are used to reduce the search space of the initial box. If a box does not belong to the solution set, it is not explored anymore and discarded. The wide domains are therefore reduced to a small enclosing set of boxes. The set-membership GNSS positioning location set $\mathbb{X}$, presented on Figures (6.24-6.25), is made of 407 boxes.

The satellite constellation setup produced a very small bias towards the positive direction of the local coordinates *Y-axis*, *i.e.* a bias towards the left side of vehicle 1. This bias is shown on the top subplot of Figure 6.25(a) and can be more easily seen on Figure 6.24(a) where the 3D projection onto the YZ plane (lateral distribution) of the set-membership GNSS positioning location set $\mathbb{X}$ is presented. From the top subplot of Figure 6.25(a), it is possible to observe a small bias towards negative direction of the local coordinates *X-axis*, *i.e.* the satellite constellation produces a very small bias towards the rear side of vehicle 1. This bias can be more easily seen on Figure 6.24(b) where the 3D projection onto the XZ plane (longitudinal distribution) of the set-membership GNSS positioning location set $\mathbb{X}$ is presented. One can also observe from Figure 6.24 a positive bias along the local coordinates *Z-axis*.

Figures 6.26 and 6.28 present results of the lateral normalized positioning error, for simulation and real experiments respectively. The dashed line with '*' marker, represent the lateral position error, determined by solving the weighted center of gravity of the set-membership (*SGP*).

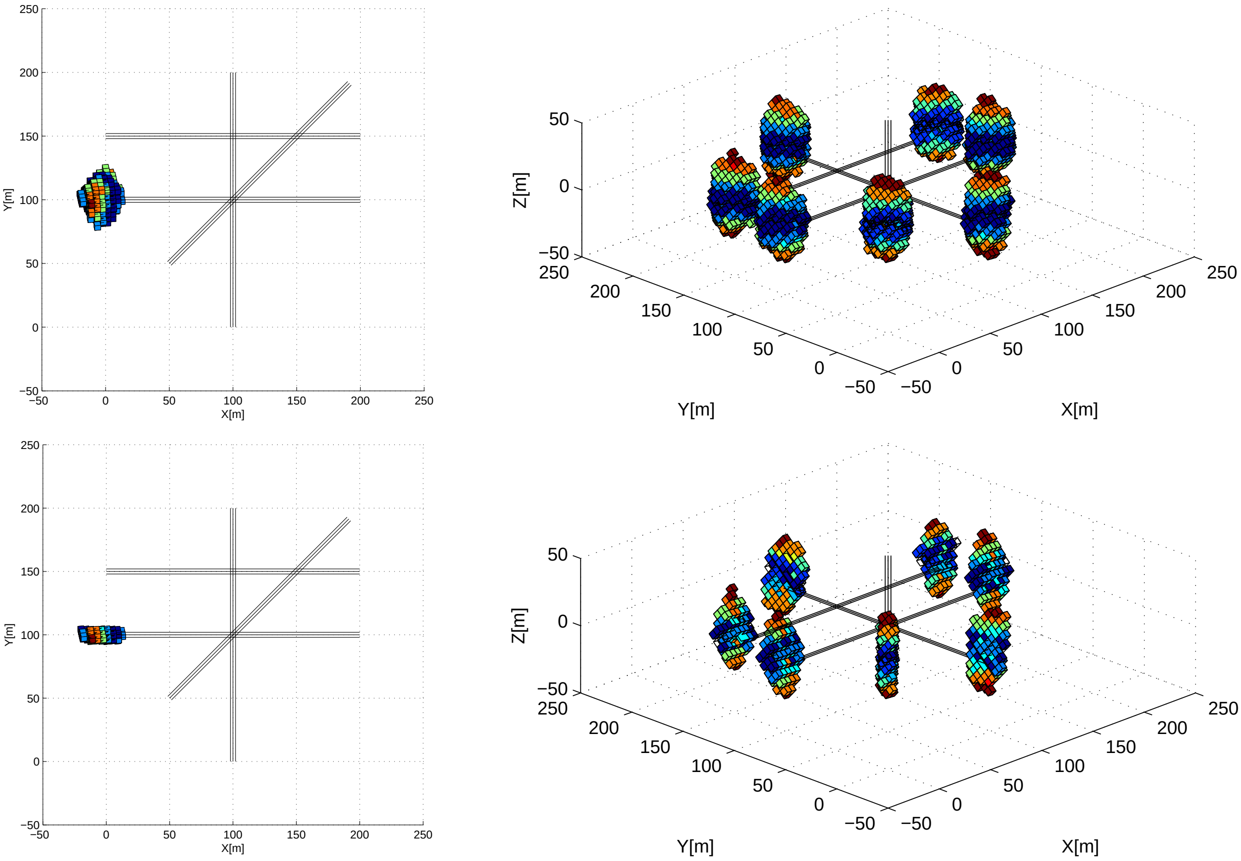


(a) (b) (c)

Figure 6.24: Simulation - Vehicle 1 SGP and LB-ASGP 2D½ solutions: 2D½ set-membership solution (top) and 2D½ lane-boundary constrained set-membership solution (bottom): (a) 3D projection onto the YZ plane (lat. distribution); (b) 3D projection onto the XZ plane (long. distribution). The clock offset co_r follows the gradient correspondence of (c)

From the simulation results depicted in Figure 6.26, it is possible to observe the bias along both *X-axis* and *Y-axis* local coordinates.

The negative bias along the local coordinates *X-axis* (*i.e.* this satellite constellation configuration generates along local coordinates *X-axis* a predominant negative error) can



(a)

(b)

Figure 6.25: Simulation - All vehicles SGP and LB-ASGP 2D$^{1}/_{2}$ solutions: (a) 3D projection onto the XY plane (top view distribution) of vehicle 1, 2D$^{1}/_{2}$ set-membership solution (top) and 2D$^{1}/_{2}$ lane-boundary constrained set-membership solution (bottom); (b) all vehicles set-membership solution, 3D$^{1}/_{2}$ set-membership solution (top) and 3D$^{1}/_{2}$ lane-boundary constrained set-membership solution (bottom). The clock offset co_r follows the gradient correspondence of Figure 6.24(c).

be easily seen through *SGP* lateral error of vehicles 2 and 4 and *SGP* longitudinal error of vehicles 1, 3, 5 and 6. The simulated vehicles 2 and 4 travel in a road lane parallel to the local coordinates *Y-axis*. The mean lateral error of vehicle 2 is positive while the mean lateral error of vehicle 4 is negative. This is because vehicle 2 heads upwards (its axis is rotated by $^{\pi}/_{2}$ with respect to the local coordinates) and vehicle 4 heads downwards (its axis is rotated by $^{3\pi}/_{2}$ with respect to the local coordinates), and therefore the vehicle *Y-axis* points toward the opposite direction (see Figure 6.23). The simulated vehicles 1, 3, 5 and 6 travel in a road lane parallel to the local coordinates *X-axis*. The mean Longitudinal error of vehicle 1 and 5 is negative while the mean lateral error of vehicle 3 and 6 is positive. This is due to the fact that vehicle 1 and 5 head towards left (their axis are aligned with the local coordinates) and vehicle 3 and 6 head towards right (their axis are rotated by π with respect to the local coordinates), and therefore the vehicle *axis* points toward the

Table 6.4: Error analysis for all vehicles.

Error [m]		Lateral					Longitudinal					Euclidean		
		$^\flat \sum_{i=1}^{N} \frac{\lVert\overline{LAT}\rVert}{N}$		mean	σ	3σ	$^\flat \sum_{i=1}^{N} \frac{\lVert\overline{LON}\rVert}{N}$		mean	σ	3σ	mean	σ	3σ
	vehicles	[1,3,5,6]	[2,4]	[1-7]	[1-7]	[1-7]	[1,3,5,6]	[2,4]	[1-7]	[1-7]	[1-7]	[1-7]	[1-7]	[1-7]
	SGP	0.372	1.558	0.204	1.27	3.300	1.494	0.349	-0.092	1.402	3.187	1.803	0.624	3.633
Sim.	LB-ASGP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	LB-CASGP	0.568	0.592	0.024	0.705	2.248	0.552	0.572	0.023	0.849	2.690	0.968	0.531	2.712
	vehicles	[1-4]		[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]	[1-4]
	SGP	0.702		0.701	4.367	13.45	0.301		-0.234	4.301	12.38	5.324	3.125	13.95
Real	LB-ASGP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	LB-CASGP	0.049		0.003	0.734	3.617	0.453		-0.332	2.109	10.30	1.760	1.447	10.33

$^\flat$ $\sum_{i=1}^{N} \frac{\lVert\overline{LAT}\rVert}{N}$ and $\sum_{i=1}^{N} \frac{\lVert\overline{LON}\rVert}{N}$ are the average of the absolute mean lateral and longitudinal error respectively.

opposite direction. Hence the absolute mean lateral error value along the local coordinates *X-axis* of vehicles 2 and 4 is high, as well as the absolute mean longitudinal error value along the local coordinates *Y-axis* of vehicles 1, 3, 5 and 6 (see Table 6.4)

The very small positive bias along the local coordinates *Y-axis* (*i.e.* this satellite constellation configuration generates a *Y-axis* predominant positive error along local coordinates) is not easily seen on Figure 6.26, hence the absolute mean lateral error value along the local coordinates *X-axis* of vehicles 1, 3, 5 and 6 is small, as well as the absolute mean longitudinal error value along the local coordinates *Y-axis* of vehicles 2 and 4 (see Table 6.4)

Therefore, this satellite constellation configuration generates an error with the major axis along the negative direction of the local coordinates *X-axis* as shown in Figure 6.23.

Regarding real experiments results, from Figure 6.28 it is not possible to observe a predominant axis error, since the trajectories followed by all vehicles have a closed loop shape. The absolute mean lateral and longitudinal error of *SGP* for all the vehicles is moderate, see Table 6.4.

Figures 6.27 and Figure 6.29 displays the normalized positioning lateral (top), longitudinal (center) and 2D euclidean (bottom) error distribution, as well as its cumulative distribution functions, for the simulated and real experiments respectively. The dashed line with '*' markers represents the position errors using *SGP*.

From the simulations results given in the top subplots of Figure 6.27 and Table 6.4, one can observe that the algorithm has a very small mean lateral and longitudinal error distribution and a high standard deviation σ. This distribution profile is due to the fact that vehicles traveling in different directions have opposite error signals. The cumulative distribution 3σ boundary of the lateral positioning error for *SGP* algorithm is very high meaning that the positioning method is rather inaccurate most of the time.

From center subplots of Figure 6.27, one can observe that the algorithm has nearly zero mean longitudinal bimodal error distribution and a high standard deviation σ. This distribution profile is due to the fact that vehicles traveling in different directions have opposite error signals, as in lateral error distribution. The cumulative distribution 3σ boundary of the longitudinal positioning error for *SGP* algorithm is very high.

The bottom subplots of Figures 6.27 present the 2D euclidean error distribution. The cumulative distribution 3σ boundary of the 2D euclidean error is very high.

The real results depicted in Figures 6.29 and Table 6.4 confirm the observations made in simulation. The lateral and longitudinal errors are therefore quite large and highly spread for the *SGP* algorithm.

6.3.4.2 Lane Boundary Augmented Set-membership GNSS positioning (LB-ASGP)

The simulation and experimental results presented in this section, make use of the lane-boundary measurements to improve the performance of the set-membership GNSS positioning method *LB-ASGP*. The experimental results presented in this section, are entitled *test experiment 1.*

The bottom subplots of Figures (6.24-6.25) present the enveloping box for the position of vehicle 1, located at $(x_{r1}, y_{r1}, z_{r1}) = (0, 100, 0)[m]$, when the constraints provided by the satellites constellation and lane-boundary constraints provided by GIS and lane-boundary sensor are used to reduce the search space of the initial box. The set-membership GNSS positioning location set $\mathbb{X}$ is reduced after a characterization of every zone compatible with the lane-boundary sensor measurements. The resulting lane-boundary constrained set-membership location set is composed by 137 boxes.

By analyzing the bottom subplot of both Figure 6.24(a) and Figure 6.25(a), it is possible to observe that the 3D projection onto the YZ plane (lateral distribution) of the set-membership GNSS positioning location set $\mathbb{X}$ is reduced after a search of all locations compatible with the lane-boundary sensor measurements. The impact of the positive bias along local coordinates *Y-axis* that this satellite constellation configuration generates on vehicle 1 is reduced. From the bottom subplot of both Figure 6.24(b) and Figure 6.25(a), it is possible to observe that the 3D projection onto the XZ plane (longitudinal distribution) of the set-membership GNSS positioning location set $\mathbb{X}$, maintains its shape, *i.e.* the impact of the predominant negative error along local coordinates *X-axis* generated by this satellite constellation configuration on vehicle 1 is not reduced.

After having applied the lane-boundary constrain on the estimated set $\mathbb{X}$, the position $[x_r, y_r]$ estimation is determined again by projecting the solution of the center of gravity weighted by the value of the receiver clock offset co_r of each sub-paving.

Figures 6.26 and 6.28 present results of the lateral normalized positioning error, for

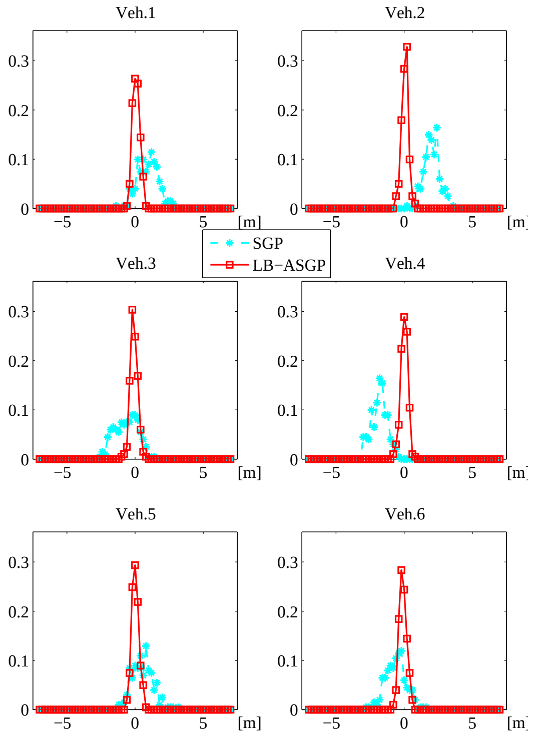


Figure 6.26: Simulation - Lateral normalized positioning error distribution, for all trajectory path for vehicles 1 to 6, using *SGP* and *LB-ASGP* estimation.

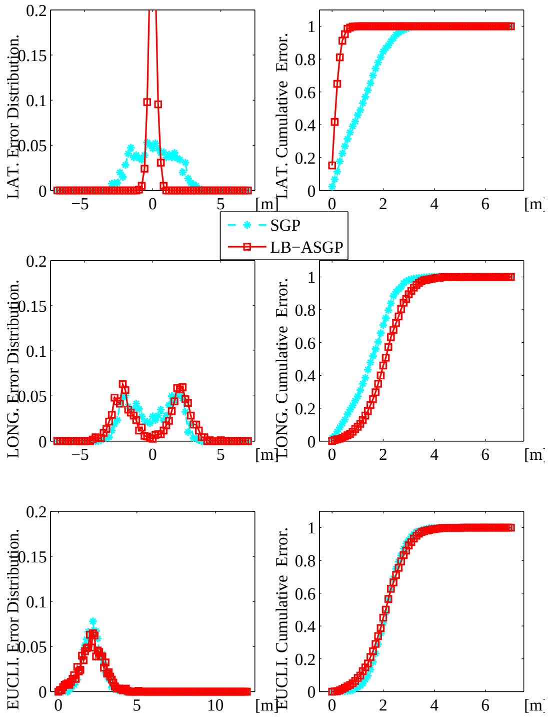


Figure 6.27: Simulation - All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom), using *SGP* and *LB-ASGP* estimation: (left column) normalized positioning errors distribution, (righ column) cumulative distribution functions.

simulation and real experiments respectively. The solid line with '□' marker, represent the lateral position error, determined by solving the *LB-ASGP* algorithm.

The *LB-ASGP* algorithm has considerably less mean absolute lateral error than *SGP*, with an error reduction of at least 90%.

Figures 6.27 and 6.29 present the normalized positioning lateral (top), longitudinal (center) and 2D euclidean (bottom) error distribution, as well as its cumulative distribution functions, for simulation and real experiments respectively. The solid line with '□' marker, represent the position errors, determined by solving the *LB-ASGP* algorithm.

From top subplots of Figure 6.27, 6.29 and Table 6.4 one can observe that the algorithm has nearly zero mean lateral error distribution and a low standard deviation σ, this distribution profile error is due to the inclusion of lane-boundary constraints provided by GIS and lane-boundary sensor, leading to a better position estimation.

The cumulative distribution 3σ boundary of the lateral positioning error for *LB-ASGP* algorithm is greatly reduced meaning that the positioning method has an accuracy better

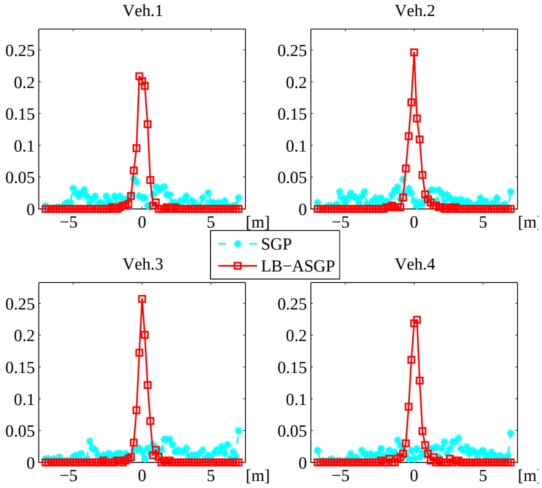


Figure 6.28: Real Experiments - Lateral normalized positioning error distribution, for all trajectory path of vehicles 1 to 4, using *SGP* and *LB-ASGP* estimation.

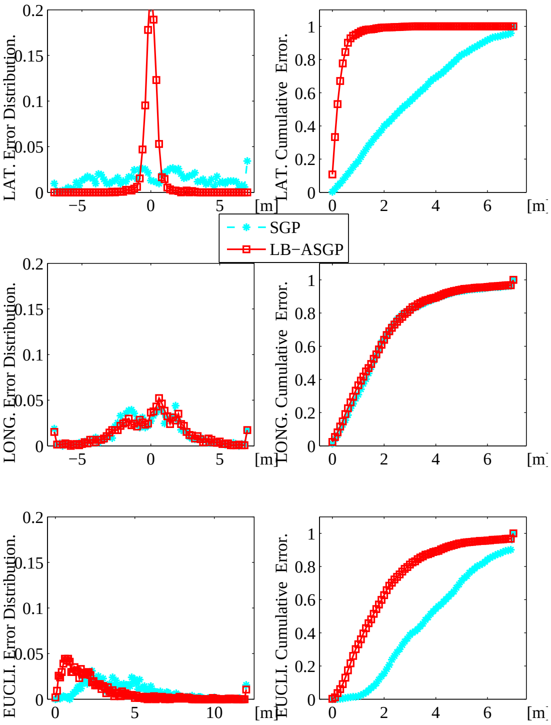
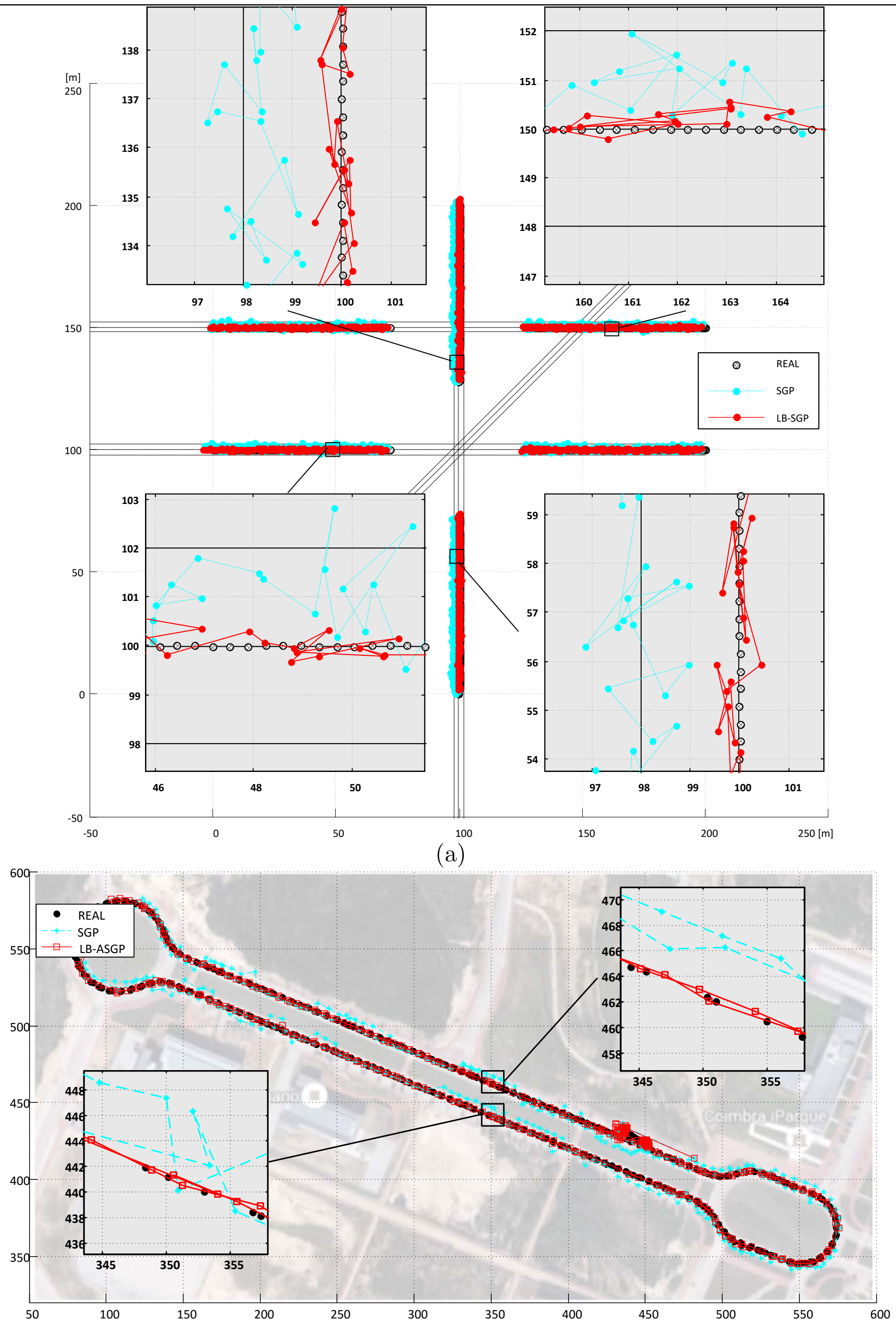


Figure 6.29: Real Experiments - All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom), using *SGP* and *LB-ASGP* estimation: (left column) normalized positioning errors distribution, (right column) cumulative distribution functions.

1 meter in simulation and 2.5 meters in real experiments, most of the time. The 3σ is reduced by least a factor of 4 when using the *LB-ASGP* algorithm. From center subplots of Figure 6.27 and 6.29, one can observe that algorithm has nearly zero mean longitudinal spread error distribution and a high standard deviation σ. As expected, this distribution profile is similar to the one obtained without the lane-boundary constraints, since the lane-boundary sensors only provide additional geometric diversity for the axis orthogonal to the road lane. The cumulative distribution 3σ boundary of the longitudinal positioning error for *LB-ASGP* algorithm maintains a very high value as expected.

The bottom subplots of Figures 6.27 and 6.29 present the 2D euclidean error distribution. For both simulations and real experiments, the mean and standard deviation σ error of the *LB-ASGP* algorithm are reduced, being more evident on the real experiment, therefore validating the simulations.

Figure 6.30(a) shows the trajectory followed by all simulated vehicles and position estimation for algorithms *SGP* and *LB-ASGP*. The subplots represent the zoomed trajec-



(a)

(b)

Figure 6.30: Trajectory and zoomed trajectory followed by each vehicle and SGP and LB-ASGP position estimation: (a) Simulation - vehicle 1 trajectory in the bottom-left zoom subplot; vehicle 2 trajectory in the bottom-right zoom subplot; vehicle 4 trajectory in the top-left zoom subplot and vehicle 6 trajectory in the top-right zoom. (b) Real Experiments

tory for vehicles 1, 2, 4 and 6 respectively bottom-left, bottom-right, top-left and top-right zoom subplot. As mentioned earlier, this satellite constellation configuration generates errors along both X and Y local coordinates axes, with the major error axis along the negative direction of the local coordinates *X-axis* (see position estimation of vehicles 2 and 4 using *SGP* in Figure 6.30(a). Therefore, the improvement of position estimation by using lane-boundary sensors is more noticeable for vehicles travelling along the local coordinates *Y-axis* and with the lane-boundary sensors measurements along the local coordinates *X-axis*, as with vehicles 2 and 4. The position estimation improvement of *LB-ASGP* algorithm for vehicles 2 and 4 is shown in subplots bottom-right and top-left respectively, where the position estimate is shifted right towards the real position. The position estimation improvement for vehicles 1 and 6 is less evident since these vehicles travel along the local coordinates *X-axis* and with the lane-boundary sensors measurements along the local coordinates *Y-axis* and the error along the local coordinates *Y-axis* is medium positive (see position estimation of vehicles 1 and 6 using *SGP* in Figure 6.30(a)). The estimates of vehicles 1 and 6 are shifted down towards the real position. This correction can be observed on subplot bottom-left and top-right respectively of Figure 6.30(a). This correction reflects a medium improvement of *LB-ASGP* algorithm on the position estimate.

Figure 6.30(b) shows the trajectory followed by all vehicles and position estimation for all presented algorithms. By analysing the subplots of Figure 6.30(b), it is possible to observe that the satellite constellation configuration, during the real experiments, generates an error with the major axis along the north-east direction (i.e towards the top-right corner of Figure 6.30(b)). This bias is successfully mitigated using the *LB-ASGP* algorithm.

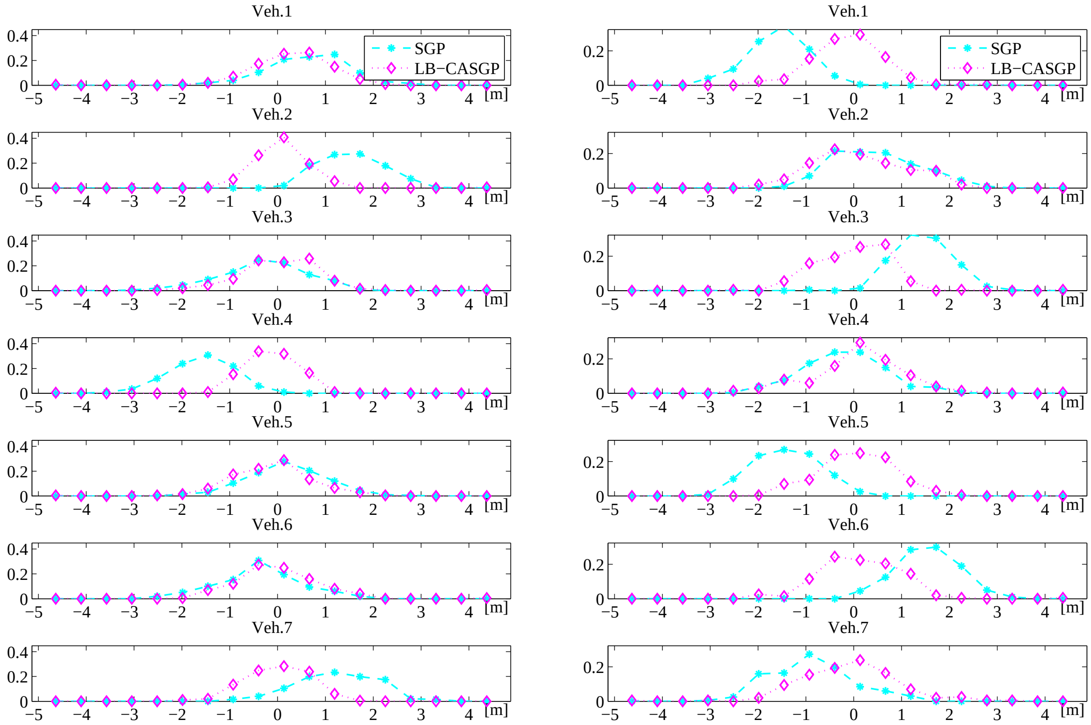
6.3.4.3 Collaborative Set-membership augmented GNSS positioning (*LB-CASGP*)

The simulation and experimental results presented in this section, make use of the lane-boundary measurements and it uses road transverse corrections shared among networked vehicles in order to improve the performance of the collaborative set-membership GNSS positioning method *LB-CASGP*. The experimental results presented in this section, are entitled *test experiment 2*.

After having applied the lane-boundary constrain on the estimated set $\mathbb{X}$, the position $[x_r, y_r]$ estimation is determined again by projecting the solution of the center of gravity weighted by the value of the receiver clock offset co_r of each sub-paving.

Figures 6.31 and 6.32 present results of the lateral and longitudinal normalized positioning error, for simulation and real experiments respectively. Figures 6.33 and 6.34 present the normalized positioning lateral (top), longitudinal (center) and 2D euclidean (bottom) error distribution, as well as its cumulative distribution functions, for simulation and real experiments respectively. The dotted line with '◇' marker, represent the position error, determined by solving the *LB-CASGP* algorithm.

Figures 6.31 and 6.32 reveal that the *LB-CASGP* concentrates the majority of both



(a)

(b)

Figure 6.31: Simulation - normalized positioning error distribution, for all trajectory path for vehicles 1 to 7, using *SGP* and *LB-CASGP* estimation: (a) lateral, (b) longitudinal.

lateral and longitudinal errors around 0, while in *SGP* the errors are more spread. This reveals that by sharing corrections, position estimates improves in both cross-track and along-track.

From top subplots of Figure 6.33, 6.34 and Table 6.4 one can observe that the algorithm has nearly zero mean lateral error distribution and a very low standard deviation σ, this distribution profile error is due to the inclusion of lane-boundary constraints provided by GIS and lane-boundary sensor, leading to a better position estimation.

The *LB-CASGP* algorithm has considerably less mean absolute lateral error than *SGP*, with an error reduction of at least 90%. The cumulative distribution σ boundary of the lateral positioning error for *LB-CASGP* algorithm is greatly reduced meaning that the positioning method has an accuracy better than 1 meter in simulation and in real experiments. The 3σ is reduced by least 60% when using the *LB-CASGP* algorithm, which mean a significant lateral error reduction during most of the time.

From center subplots of Figure 6.33, 6.34 and Table 6.4, one can observe that algorithm has nearly zero mean longitudinal spread error distribution and a low standard deviation σ. As expected, when using *LB-CASGP* algorithm, the σ is improved *w.r.t.* to the one obtained

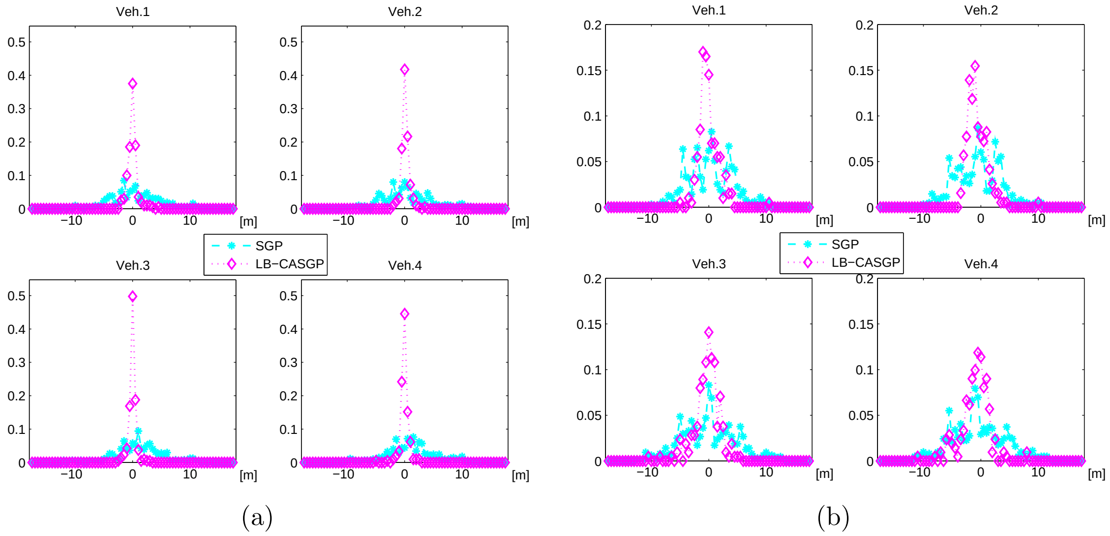


Figure 6.32: Real Experiments - normalized positioning error distribution, for all trajectory path for vehicles 1 to 4, using *SGP* and *LB-CASGP* estimation: (a) lateral, (b) longitudinal.

without the collaborative sharing of satellite specific errors, since the lane-boundary sensors not only provide additional geometric diversity for the axis orthogonal to the road lane of the carrying vehicle but as well for networked vehicles. The σ is reduced by at least 49% when using the *LB-CASGP* algorithm, which mean a significant longitudinal error reduction. The cumulative distribution 3σ boundary of the longitudinal positioning error is also reduced as expected.

The bottom subplots of Figures 6.33 and 6.34 present the 2D euclidean error distribution. For both simulations and real experiments, the mean and standard deviation σ error of the *LB-CASGP* algorithm are reduced, being more evident on the real experiment, therefore validating the simulations.

Figure 6.35 shows the trajectory followed by all vehicles and position estimation for all presented algorithms, tested in simulation Figure 6.35(a) and in real experiments Figure 6.35(b).

The subplots of Figure 6.35(a), represent the zoomed trajectory for vehicles 2, 6 and 7, respectively bottom-right, top-right and top-left zoom subplot. As mentioned earlier, this satellite constellation configuration generates errors along both X and Y local coordinates axes, with the major error axis along the negative direction of the local coordinates *X-axis* (see position estimation of vehicles 2 using *SGP* in bottom-right zoomed zoomed of Figure 6.35(a)). Therefore, the road-transverse improvement of position estimation by using *LB-CASGP* algorithm is more noticeable for vehicles travelling along the local coordinates *Y-axis* and with the lane-boundary sensors measurements along the local coordinates *X-axis*, as with vehicles 2 and 4. The road-transverse position estimation

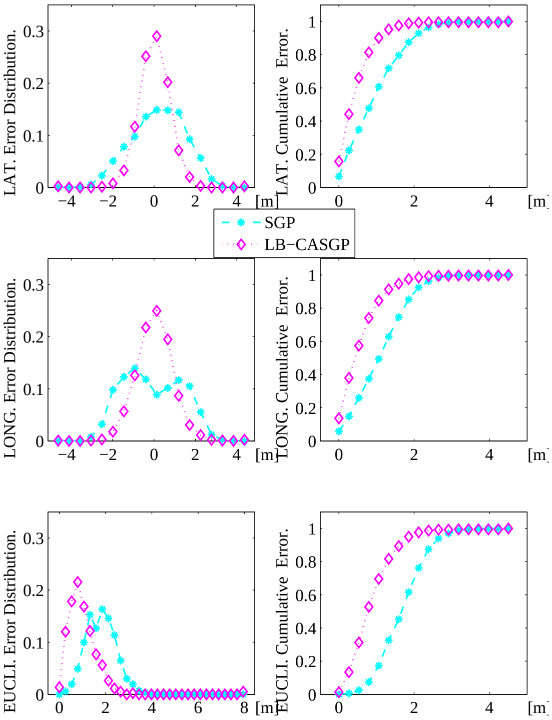


Figure 6.33: Simulation - All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom), using *SGP* and *LB-CASGP* estimation: (left) normalized positioning errors distribution; (right) cumulative distribution functions.

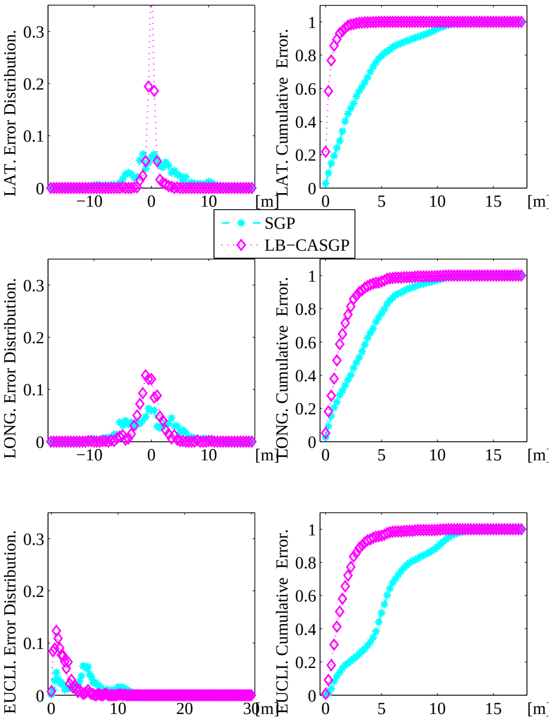
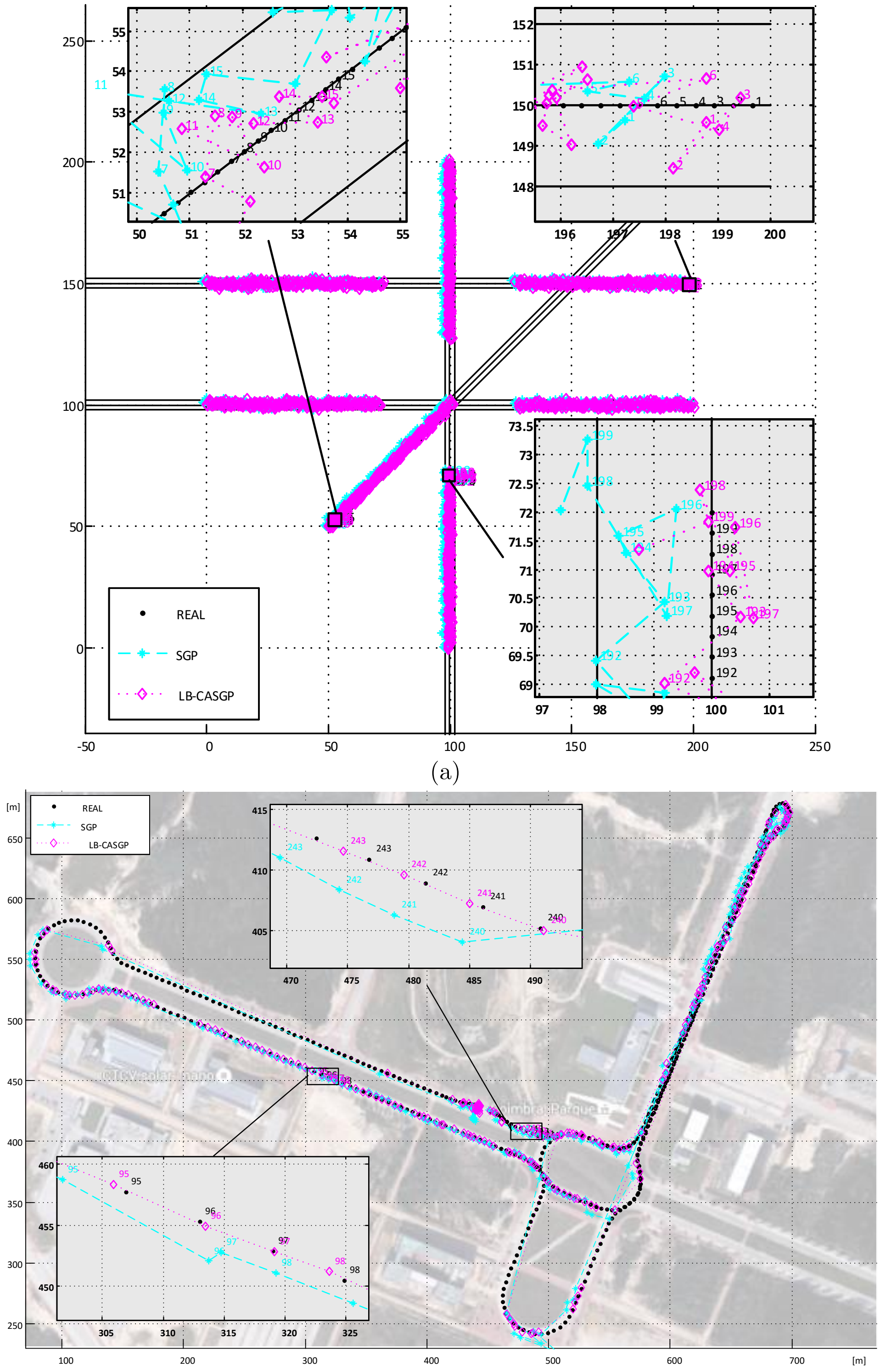


Figure 6.34: Real Experiments - All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom), using *SGP* and *LB-CASGP* estimation: (left) normalized positioning errors distribution; (right) cumulative distribution functions.

improvement of *LB-CASGP* algorithm for vehicle 2 is shown in subplot bottom-right, where the position estimate is shifted right towards the real position. The road-parallel improvement of position estimation by using *LB-CASGP* algorithm is more noticeable for vehicles travelling along the local coordinates *X-axis* and using the shared corrections along the local coordinates *X-axis*, as with vehicles 1, 3, 5 and 6. The road-transverse position estimation improvement of *LB-CASGP* algorithm for vehicle 6 is shown in subplot top-right, where the position estimate is shifted right towards the real position. The road-transverse improvement for vehicles 1, 3, 5 and 6 and the road-parallel improvement for vehicles 2 and 4 is less evident since this satellite constellation produces a very small positive bias along the local coordinates *Y-axis*.

The position estimation for vehicle 7 reflects a medium improvement of *LB-CASGP* algorithm on the position estimate as this vehicle is not equipped with lane-boundary sensor. Nevertheless it benefits from the corrections estimates broadcast by all lane-boundary sensor equipped vehicles, making its correction noticeable in both road-transverse and



(a)

(b)

Figure 6.35: Trajectory and zoomed trajectory followed by each vehicle and position estimation using SGP and LB-ASGP: (a) Simulation - vehicle 2 trajectory in the bottom-right zoom subplot; vehicle 6 trajectory in the top-right zoom subplot and vehicle 7 trajectory in the top-left zoom subplot (b) Real Experiments

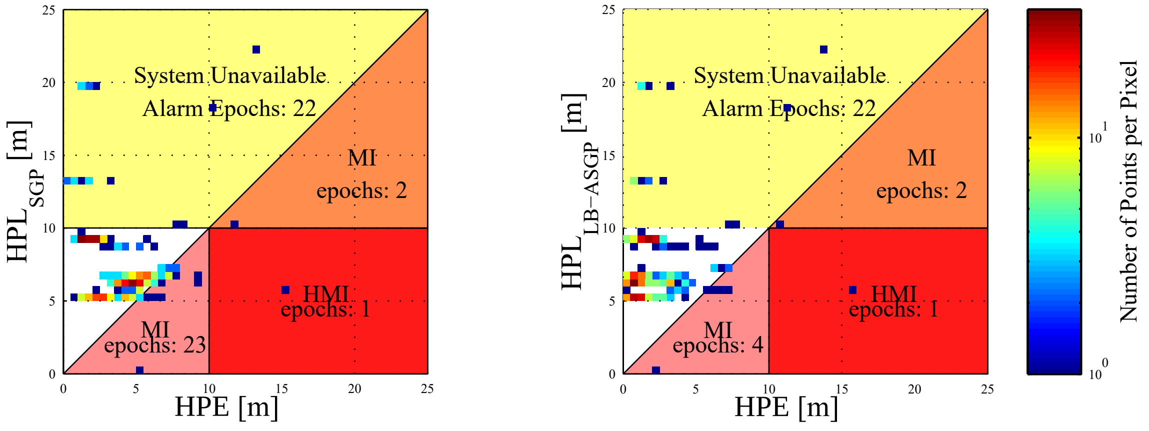
road-parallel. This correction can be observed on subplot top-left of Figure 6.35(a).

The trajectory followed by vehicle 3 and position estimation for all presented algorithms, under real experiments is shown in Figure 6.35(b). Vehicles were platooning and therefore most of the time they were nearly collinear, this poses a problem as *LB-CASGP* reaches a singularity when all networked vehicles are collinear. This event is higly unlikely to occurs in city scenarios with the presence of a high number of networked vehicles. When all networked vehicles are collinear, the position estimation is achieved using the non-collaborative position estimation method *LB-ASGP*. In Figure 6.35(b), only solutions, when the *LB-CASGP* is able provide a solution are shown, *i.e.* only epochs where the vehicles are non-collinear or not nearly collinear are presented in the figure. By analysing the subplots of Figure 6.35(b), it is possible to observe that the satellite constellation configuration, during the real experiments, generates an error along the south-west direction (*i.e.* towards the top-right corner of Figure 6.35(b)) and west (*i.e.* towards the right side of Figure 6.35(b)). These biases are successfully mitigated using the *LB-CASGP* algorithm. The road-transverse error is significantly removed as the position estimate is shifted towards north-east. The road-parallel error is corrected by shifting the position towards east. Subplots of Figure 6.35(b), where each position estimate is numbered, enables us to conclude that the position estimate using *LB-CASGP* is always better than using *SGP*, both in road-transverse and in road-parallel.

6.3.4.4 Integrity Analysis of SGP, LB-ASGP and LB-CASGP

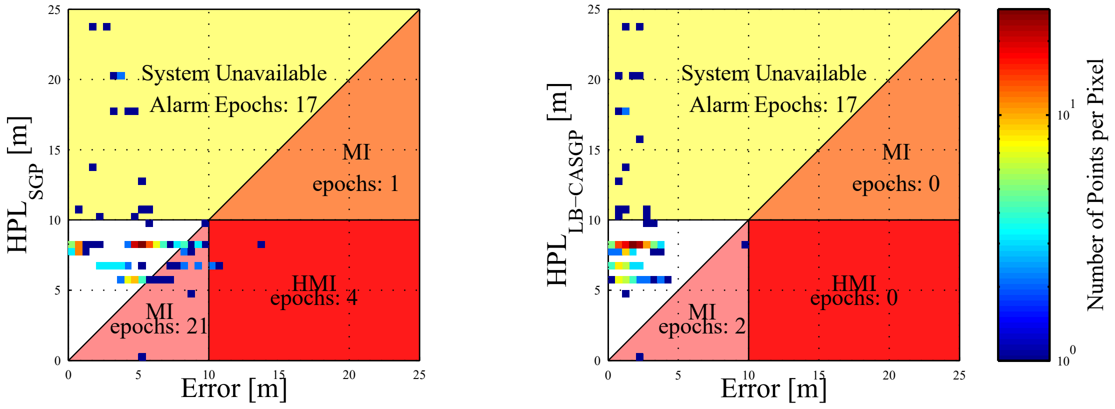
An integrity metric of major interest when dealing with integrity concerns is the Horizontal Protection Level (HPL) [114][115]. *Stanford diagrams* are widely used and consist of a histogram of positioning solutions in terms of actual error and protection level. The *Stanford diagrams* of both *SGP* and *LB-ASGP* obtained with the real data with test experiment 1 are shown in Figure 6.36. The *Stanford diagrams* of both *SGP* and *LB-CASGP* obtained with the real data with test experiment 2 are shown in Figure 6.37.

They illustrate the integrity performance achieved during the periods analyzed. For each algorithm, the navigation solution of all the vehicles has been merged in the same plot in order to show the total domain integrity per algorithm. For the HPL, it was considered a probability of missed detection of 0.001 and a false alarm tolerance of 3.33333×10^{-7}, the horizontal alarm limit (HAL) was set here to $10[m]$ but this value can be easily adapted to the requirements of different applications. Regarding the horizontal GNSS positioning system, its integrity risk is the probability that, at any moment, the horizontal position error (HPE) exceeds an HAL. The integrity system is declared unavailable when the HPL is greater than HAL. If the system is available and the HPE is not bounded by the HPL, the event is considered as a Misleading Information (MI), since the HPL is always supposed to be an upper bound of the HPE. Moreover, the event is declared as Hazardously Misleading Information (HMI) if the HPE exceeds the HAL [115].



(a) (b)

Figure 6.36: Real Experiments - HPL Stanford diagram of test experiment 1: (a) SGP; (b) LB-ASGP (misleading information (MI), hazardously misleading information (HMI))



(a) (b)

Figure 6.37: Real Experiments - HPL Stanford diagram of test experiment 2: (a) SGP; (b) LB-CASGP (misleading information (MI), hazardously misleading information (HMI))

The position error of test experiment 1, is not always bounded by the protection level, that is, there are several MI events in both *SGP* and *LB-ASGP* algorithms, although they are almost six times higher for the *SGP*. Regarding HMI, both algorithms have only one event. The real experiments results reveal that the *SGP* algorithm is 88% of the time under normal operation whereas the *LB-ASGP* is 93%, which is a significant improvement in terms of positioning availability.

Regarding the test experiment 2, there are also several MI events in both *SGP* and *LB-CASGP* algorithms, although they are almost ten times higher for the *SGP*. Regarding HMI, the *LB-CASGP* achieves none events while the *SGP* has four events. The real experiments results reveal that the *SGP* algorithm is 78% of the time under normal operation whereas the *LB-CASGP* is 90%, which is a significant improvement in terms of positioning availability.

6.3.4.5 Multipath satellite removal of Augmented and Collaborative GNSS positioning

The simulation results presented in this section, make use of the lane-boundary measurements to find aberrant satellite pseudoranges LB-RSSMFDE.

NLOS multipath error is always positive and, although typically tens of metres, is potentially unlimited. Signals received via distant tall buildings can exhibit errors of more than a kilometre [109].

In the simulation results presented in this section, vehicle 1 GPS receiver suffer a NLOS multipath disturbance of 50 meters in one satellite, this level of disturbance is consistent with observations made in urban environment. Olivier Le Marchand on a 500 seconds trip in urban environment, registered 80% of the satellites reached multipath values above 50 meters and that 30% of the satellites reached multipath values above 200 meters [116].

Iterative Least Squares based Positioning Methods

The robustness of the algorithms *LB-RAIGP* and *LB-RCAIGP* is obtained by means of the removal of the aberrant satellite pseudoranges from the computation of individual vehicle position estimation.

Figures 6.38 and 6.39 present the results obtained using *IGP*, *LB-AIGP*, *LB-CAIGP*, *LB-RAIGP* and *LB-RCAIGP*.

The overall performance of the robust algorithms (*LB-RAIGP* and *LB-RCAIGP*) is significantly higher than the non-robust algorithms (*IGP*, *LB-AIGP* and *LB-CAIGP*), see Figure 6.39(a). The traditional IGP positioning method can be even better than the non-robust collaborative positioning algorithm, the reason behind this poor performance of *LB-CAIGP* is due to the fact that the broadcasted corrections are biased by vehicle 1, leading to a worst position estimation.

Vehicle 1 The effect of NLOS multipath disturbance of 50 meters on vehicle 1, can be easily seen in top subplot of Figure 6.38, this effect is more evident in Figure 6.39(b). All non-robust algorithms, namely IGP, *LB-AIGP* and *LB-CAIGP*, have their estimated position highly affected. The *LB-AIGP* is the only non-robust algorithms that achieves a good performance but only on the lateral position. This due to the fact that the lane-boundary measurement, provides an additional road transverse information that superimposes to the NLOS multipath disturbance. Regarding the robust algorithms, all of them are able to provide good position estimates due to the removal of the aberrant satellite measurement using LB-RSSMFDE. As expected the *LB-RCAIGP* is able to

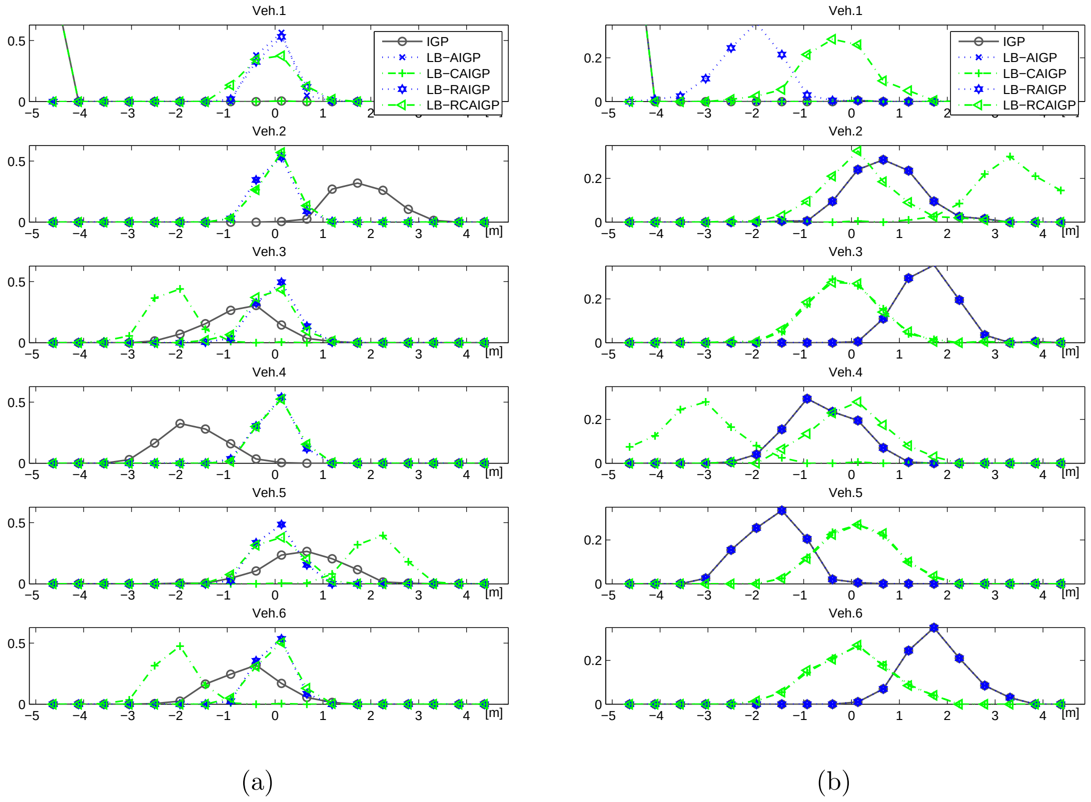
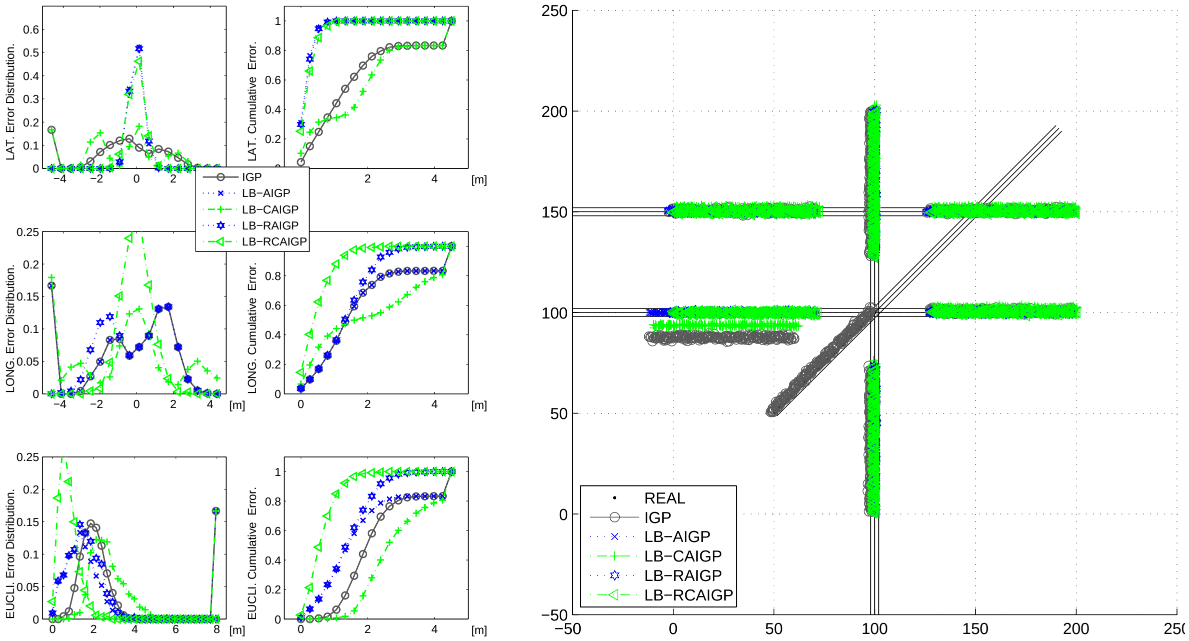


Figure 6.38: Simulation - normalized positioning error distribution, for all trajectory path for vehicles 1 to 7, using *IGP*, *LB-AIGP*, *LB-CAIGP*, *LB-RAIGP* and *LB-RCAIGP* estimation: (a) lateral, (b) longitudinal.

improve the position estimate in both lateral and longitudinal axis while the *LB-RAIGP* is only able to improve the position estimate along the lateral axis since it uses the lane-boundary measurement but it lacks the collaborative correction estimates to improve the longitudinal estimate.

Vehicle 2-6 As expected the effect of NLOS multipath disturbance of 50 meters of vehicle 1, on vehicles 1-6 should only occur on collaborative algorithms, namely *LB-CAIGP* and *LB-RCAIGP*. This third party effect can be easily observed in Figure 6.38 on *LB-CAIGP* algorithm, but it is mitigated on *LB-RCAIGP* due to the removal of the aberrant satellite measurement using LB-RSSMFDE. Vehicle 1 offset due to NLOS multipath disturbance leads to a wrongly correction estimation along its lateral axis. The sharing of this wrongly estimated correction using *LB-CAIGP*, leads to a lateral bias on the position estimation off vehicles travelling parallel to vehicle 1 (*i.e.* vehicle 3, 5 and 6) and a longitudinal bias on the position estimation off vehicles travelling orthogonal to vehicle 1 (*i.e.* vehicle 2 and 4). This means that in the presence of NLOS multipath disturbance, using collaborative positioning worsens the overall solution. As expected the



(a)

(b)

Figure 6.39: Simulation results for all trajectory path for vehicles 1 to 6, using *IGP*, *LB-AIGP*, *LB-CAIGP*, *LB-RAIGP* and *LB-RCAIGP* estimation: (a) All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom); (b) Trajectory followed by each vehicle.

LB-RCAIGP is able to improve the position estimate in both lateral and longitudinal axis of all connected vehicles.

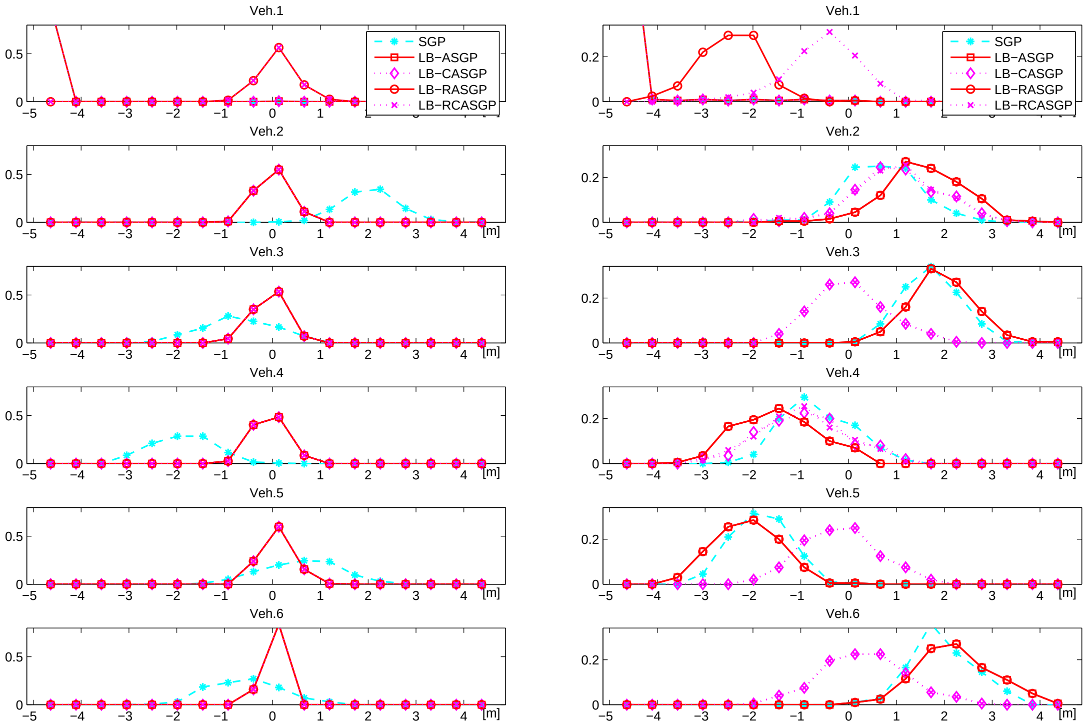
Set Membership based Positioning Methods

The *LB-RASGP* and *LB-RCASGP* besides benefitting from the removal of aberrant satellite pseudoranges using LB-RSSMFDE, it also includes a 1-relaxed set-membership method in order to be robust with an undetected aberrant measure.

Figures 6.40 and 6.41 present the results obtained using *SGP*, *LB-ASGP*, *LB-CASGP*, *LB-RASGP* and *LB-RCASGP*.

The overall performance of the robust set-membership based algorithms (*LB-RASGP* and *LB-RCASGP*) is higher than the non-robust algorithms (*SGP*, *LB-ASGP* and *LB-CASGP*), see Figure 6.41(a).

Vehicle 1 All non-robust algorithms, namely SGP, *LB-ASGP* and *LB-CASGP*, are unable to provide an estimate for the vehicle position as can be seen in top subplot of Figure



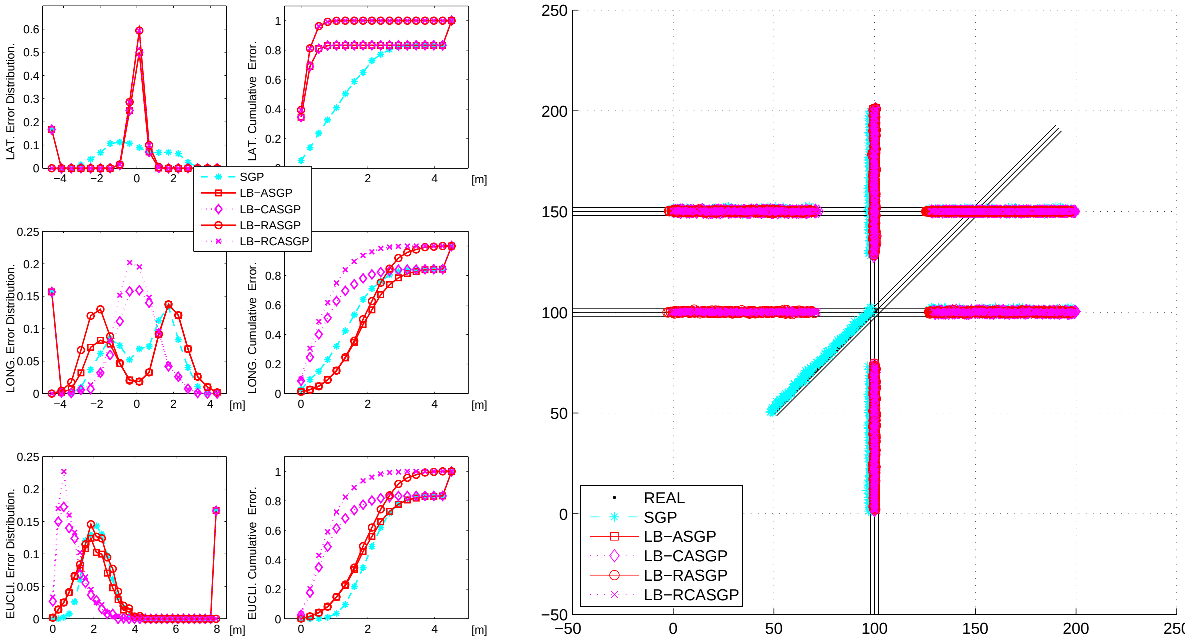
(a)

(b)

Figure 6.40: Simulation - normalized positioning error distribution, for all trajectory path for vehicles 1 to 7, using *SGP*, *LB-ASGP*, *LB-CASGP*, *LB-RASGP* and *LB-RCASGP* estimation: (a) lateral, (b) longitudinal.

6.40. The effect of NLOS multipath disturbance of 50 meters on vehicle 1, results in an empty set, and therefore no position estimate is possible to compute. Regarding the robust algorithms, all of them are able to find a location zone given the uncertainty on satellites observations and satellites positions due to the removal of the aberrant satellite measurement using LB-RSSMFDE. As expected the *LB-RCASGP* is able to improve the position estimate in both lateral and longitudinal axis while the *LB-RASGP* is only able to improve the position estimate along the lateral axis since uses the lane-boundary measurement but it lacks the collaborative correction estimates to improve the longitudinal estimate.

Vehicle 2-6 Regarding non-collaborative algorithms (*LB-ASGP* and *LB-RASGP*), a multipath disturbance on a third party vehicle would not hinder its own position estimate, whether the third party vehicle computes and empty set (lack of a solution) or a biased estimate. The effect of NLOS multipath disturbance of 50 meters off vehicle 1, leads to an empty set and a lack of a solution, this means that there is no correction estimation from vehicle 1, that could be shared with connected vehicles, meaning that the contamination effect from vehicle 1 on third party vehicles is, therefore inexistent, hence the *LB-CASGP* position estimation is similar to the *LB-RCASGP*.



(a)

(b)

Figure 6.41: Simulation results for all trajectory path for vehicles 1 to 6, using *SGP*, *LB-ASGP*, *LB-CASGP*, *LB-RASGP* and *LB-RCASGP* estimation: (a) All vehicles positioning errors, lateral (top), longitudinal (center) and 2D euclidean (bottom); (b) Trajectory followed by each vehicle.

6.4 Final Remarks

In this chapter simulations and field experiments were presented and they deeply characterize the performance of all algorithms composing the VPE data fusion module.

The odometry calibration with the MSS revealed good results. The majority of systematic errors associated to the odometry relying only on encoders, are by the magnetic markers based calibration, somehow eliminated. However that procedure by itself does not solve the wheel slippage problem, which was reduced by applying confidence tests. If the magnetic markers are not widely distributed the problem of accurate positioning is still present. Simulation results show that the V2X communication allied with sensor fusion, has the potential to lead to the improvement of the robustness and accuracy of vehicle-positioning estimation. The results show that although the errors accumulated, every time a landmark is detected the pose accuracy is significantly improved, where landmarks can either be provided by a front laser detected vehicle or natural feature, or by a MSS detected magnetic marker. The fusion method also handles false detections either coming from hardware anomaly, incorrect positioned magnets or false LIDAR detection, by discarding them on the innovation process of the filters. The overall performance including the RTK-GPS and MSS fusion with odometry revealed good results.

In an effort to move away from costly RTK-GPS receivers, several algorithms improving the performance of low cost standalone GNSS algorithms were presented. They improved the position estimate by augmenting the GNSS localization algorithm in environments where a lane-boundary sensor is capable of detecting the distance from vehicle to the lane and relate the detected lane boundary with a GIS database. When applying the collaborative algorithms, the satellite-specific errors are used in a cooperative framework enabling the reduction of its impact on networked vehicles in both cross-track and along-track components. This method can also improve position estimates of vehicles not equipped with lane-boundary sensor. The non-equipped vehicles use the cooperative estimation of the satellite-specific errors to improve its own position estimate.

The proposed set-inversion algorithms have the advantage of guaranteeing not to lose any solution in the computation process and are insensitive to local minimum convergence issues. They are based on constraints propagation on real intervals. They are therefore naturally very reliable by nature.

In this chapter, we have also included a performance analysis of several iterative least squares based GNSS localization algorithms: standard, a lane-boundary augmented and collaborative. The two later versions were implemented using two different approaches: robust and non-robust. When subjected to multipath disturbances the collaborative proposed algorithms can provide worst estimation than stand-alone solutions, *i.e.* the vehicle receiver affected by multipath is contaminating the estimation of the vehicles in the vicinity. Therefore, robust set-inversion based algorithms were also developed. They extend the guaranteed zone computation algorithm to take into account outliers, namely, by removing detected aberrant satellite measurements and using a relaxed set-membership approach. The repeatability and the errors magnitude of all algorithms were evaluated in a comparative way, and revealed that the proposed robust algorithms achieved a greater level of accuracy. The position estimation improvement at a low cost enables massification, which consequently provides sufficient data for an intelligent traffic management system to proper regulate traffic by being aware of every vehicle position with high accuracy.

Chapter 7

Final remarks and conclusion

The simulation framework is paramount to achieve the goals set for the assessment of novel ITM algorithms. One of the goals of the research work was to provide open-source tools, easy to use and to add new custom modules, for the research academic community. The study of vehicular road traffic simulators revealed a wide range of simulators available, nevertheless all existing traffic simulators lack key characteristics needed for the mentioned thesis work goals, namely being proprietary and lack of detailed sensors models. Therefore to achieve an accurate and realistic simulator, with the level of detail required, a new simulator ISR-TRAFSIM was developed in Matlab, with the disclosure of the source code. ISR-TRAFSIM was built based on a MAS architecture and it provides a vehicle simulation, up to the level of representation, required for the development of intersection traffic management algorithms for automated vehicles; a vehicle emission model that is comprehensive, computationally inexpensive and accurate for the assessment of fuel consumption and greenhouse gas emissions of CO_2; it integrates both V2X communications as well as different sensors/actuators models, as well as their behavior and associated noise characteristics for the automated vehicles pose estimation and control. The ISR-TRAFSIM is highly customizable allowing the analysis of ITM algorithms integrated with other ITS-related studies, such as, PFC performance evaluation and sensor fusion for accurate vehicle localization for both conventional Ackermann steering and 4WS vehicles. The integration of an instantaneous emission model, provides improved assessments of the environmental impact of traffic control strategies at intersections. The conversion of the simulator architecture to a socket-based multi-client intersection connected to a managing server is marked as future work, as well as the modeling of the electric and hybrid electric vehicles emissions, featuring batteries, supercapacitor and fuel cells.

The ITM algorithms are aimed to minimize accidents, traffic congestion and consequently the environmental costs of road traffic. The developed intelligent traffic management ITM algorithms, which are based on a spatio-temporal reservation scheme, ensure that vehicles proceed through the intersection without colliding with other vehicles while at the same time reducing the intersection delay and environmental impacts. Specifically, the spatio-temporal reservation scheme provides each vehicle a collision-free path that is decomposed into a speed profile along with navigational instructions. Three ITM algorithms for vehicles equipped with V2X communications are proposed aiming the reduction of the traffic environmental impacts in urban areas. The first algorithm (WMITM), decelerates the vehicle and reserve the upcoming layer cells after the last layer occupied cells of the spatio-temporal matrix. The second algorithm (EMITM), decelerates the vehicle and reserve the earliest layer free cells of the spatio-temporal matrix. The third algorithm (FMITM), accelerates the vehicle and reserve the layer free cells of the spatio-temporal matrix earlier than it would be if the vehicle followed a regular speed profile. The developed ITM algorithms are very well suited for autonomous vehicles, but it can also be used by human drivers if they follow accurately the proposed speed profile along the path. Compared with traditional traffic management techniques, the simulation results prove that the proposed ITM algorithms increased traffic output flux, that the traffic flow rate can be higher and the average time to cross intersections can be significantly reduced. The research also show that these improvements are more significant when the traffic flow increases. Regarding the CO_2 emissions they are significantly reduced until the ITM algorithm reaches its saturation point. One ITM algorithm for vehicles not equipped or with faulty V2X communications is proposed aiming to enable to accommodate a transitory period where both equipped and not equipped vehicles share the intersection. The conducted studies show that the inclusion of a low percentage of vehicles, not equipped or with faulty V2V and V2I communications, in intelligent intersections using the legacy algorithm, have a low impact on the traffic flow. This research demonstrates the potential of emission reduction of optimized traffic flow through the use of ITM techniques. Future work will focus on adapting the canonical potential fields and A^* algorithms, used in local path-planning, to a spatio-temporal reservation based technique. Moreover, the introduction of high priority emergency vehicles is also marked as future work.

To implement the traffic management in cooperative driving scenarios, the pose and speed of vehicles must be accurately determined. In the first stage of this research a vehicle pose estimator for autonomous vehicles navigation in semi-structured outdoor environments was developed. It integrates redundant encoders data and absolute positioning data provided by landmarks and artificial beacons. Natural features are localized using a laser range sensor, and magnetic sensing rulers were developed to detect magnetic markers buried in the ground. In the first fusion stage, data from four wheel encoders and one steering encoder are fused by means of an EKF, providing robust odometric information, namely in face of undesirable effects of wheels slippage. Next, a second fusion stage is processed for integrating odometric and absolute positioning data. Cooperation between road users trough V2X communication is a way to improve localization accuracy. The vehicle pose estimator was further extended to enable inter-vehicle sensor fusion for an higher vehicle positioning accuracy. This extension included augmenting GNSS localization with other sensory information.

Finally set-membership methodologies were used to compute GNSS guaranteed risk integrity zone. When vehicles localization systems involve standalone GNSS receivers, the resulting accuracy can be affected by satellite-specific errors of several meters. This work studied how road-features like lane marking detected by on-board cameras can be exploited to reduce absolute position errors of cooperative vehicles sharing information in real-time in a network. The algorithms considered in this work are based on a error bounded set membership strategy. In every vehicle, a set membership algorithm computes the absolute position and an estimation of the satellite-specific errors by using raw GNSS pseudoranges, lane boundary measurements and a 2D georeferenced road map which provides absolute geometric constraints. As lane-boundary measurements provide essentially cross-track corrections in the position estimation process, cooperation enables the vehicles to improve their own estimates whatever the orientation of the roads. Set-membership methods are very efficient to solve this problem since they do not involve any independence hypothesis of the errors and so, the same information can be used several times in the computation. Such class of algorithm provides a novel approach to improve position accuracy for connected vehicles guaranteeing the integrity of the computed solution which is pivoting for automated vehicles requiring guaranteed safety-critical solutions. Results from simulations and real experiments show that sharing position corrections reduces significantly satellite-specific GNSS errors effects in both cross-track and along-track components. Moreover, the lane-boundary measurements help reducing the estimation error in all networked vehicles even those which are not equipped with an embedded perception system.

A robust set-inversion based algorithm was also developed, to solve the problem of NLOS multipath and its error propagation when using collaborative algorithms, i.e. the vehicle receiver affected by multipath is contaminating the estimation of the vehicles in the vicinity. The robust algorithms extend the guaranteed zone computation algorithm to take into account outliers, namely, by removing detected aberrant satellite measurements and using a relaxed set-membership approach. The integration of a inertial measuring unit tightly coupled with multiple GNSS receivers per vehicle is set as future work.

Bibliography

[1] P. Goodwin, "The economic costs of road traffic congestion," *UCL (University College London), The Rail Freight Group*, 2004.

[2] C. Li and S. Shimamoto, "A real time traffic light control scheme for reducing vehicles co 2 emissions," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, pp. 855–859, IEEE, 2011.

[3] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, pp. 591–656, 2008.

[4] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 617–624, International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[5] OECD, "Road infrastructure, inclusive development and traffic safety in korea," 2014.

[6] X. Yang, X. Li, and K. Xue, "A new traffic-signal control for modern roundabouts: method and application," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 4, pp. 282–287, 2004.

[7] Y. Bai and K. Xue, "Association of signal-controlled method at roundabout and stop rate," in *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, vol. 3, pp. 37–42, IEEE, 2010.

[8] T. Tollazzi, M. Rencelj, and S. Turnsek, "Slovenian experiences with alternative types of roundabouts-"turbo" and "flower" roundabouts," in *8th International Conference May*, vol. 19, p. 20, 2011.

[9] A. Hassan, *VANET simulation.* Master's Thesis in Electrical Engineering, School of Information Science, Computer and Electrical Engineering, Halmstad University, 2009.

[10] A.-L. Beylot and H. Labiod, *Vehicular Networks: Models and Algorithms.* John Wiley & Sons, 2013.

[11] S. P. Hoogendoorn and P. H. Bovy, "State-of-the-art of vehicular traffic flow modelling," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, no. 4, pp. 283–303, 2001.

[12] S. K. Zegeye, *Model-based traffic control for sustainable mobility.* TU Delft, Delft University of Technology, 2011.

[13] B. Bartin, K. Ozbay, O. Yanmaz-Tuzel, and G. List, "Modeling and simulation of unconventional traffic circles," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1965, pp. 201–209, 2006.

[14] D. Gettman and L. Head, "Surrogate safety measures from traffic simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1840, pp. 104–115, 2003.

[15] E. Bernauer, *Review of micro-simulation models.* Institute for Transport Studies, 1997.

[16] J. L. Pereira and R. J. Rossetti, "An integrated architecture for autonomous vehicles simulation," in *Proceedings of the 27th annual ACM symposium on applied computing*, pp. 286–292, ACM, 2012.

[17] E. Juliussen and J. Carlson, *Emerging Technologies: Autonomous Cars-Not If, But When.* IHS Automotive Technology Research, 2014.

[18] H. Dia and S. Panwai, "Nanoscopic traffic simulation: enhanced models of driver behaviour for its and telematics simulations," in *International Symposium on Transport Simulation, 8th, 2008, Queensland, Australia*, 2008.

[19] A. Nagurney, "Congested urban transportation networks and emission paradoxes," *Transportation Research Part D: Transport and Environment*, vol. 5, no. 2, pp. 145–151, 2000.

[20] E. Demir, T. Bektaş, and G. Laporte, "A review of recent research on green road freight transportation," *European Journal of Operational Research*, vol. 237, no. 3, pp. 775–793, 2014.

[21] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak, "Vehicular networks for reduction of fuel consumption and co 2 emission," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, pp. 671–676, IEEE, 2010.

[22] C. Guo, B. Yang, C. S. Jensen, M. Kaul, Y. Ma, K. Torp, and O. Andersen, *Reducing Environmental Footprint based on Multi-Modal Fleet management Systems for Eco-Routing and Driver Behavior Adaptation - Deliverable D3.2.* Reduction project funded by the European Commission as FP7-ICT-2011-7 STREP project number:288254, 2012.

[23] P. Boulter, I. S. McCrae, and T. J. Barlow, *A review of instantaneous emission models for road vehicles.* TRL Limited, 2007.

[24] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 3, pp. 1018–1031, 2014.

[25] A. G. Simpson, *Parametric modelling of energy consumption in road vehicles.* Doctor of Philosophy in Engineering Thesis, University of Queensland, 2005.

[26] P. Melo, R. de Castro, and R. E. Araújo, *Evaluation of an Energy Loss-Minimization Algorithm for EVs Based on Induction Motor.* INTECH Open Access Publisher, 2012.

[27] X. Kang, *Vehicle-Infrastructure Integration (VII) enabled Plug-in Hybrid Electric Vehicles (PHEVS) for traffic and energy management.* Master of Science in Engineering Thesis, University of Clemson, 2009.

[28] C. Ongini, *Analysis and design of energy-oriented driving assistance systems.* PhD thesis, Italy, 2014.

[29] M. C. Figueiredo, R. J. Rossetti, R. A. Braga, and L. P. Reis, "An approach to simulate autonomous vehicles in urban traffic scenarios," in *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pp. 1–6, IEEE, 2009.

[30] C. Mehlführer, J. C. Ikuno, M. Simko, S. Schwarz, M. Wrulich, and M. Rupp, "The vienna lte simulators-enabling reproducibility in wireless communications research.," *EURASIP J. Adv. Sig. Proc.*, vol. 2011, p. 29, 2011.

[31] S. Ray, J. B. Carruthers, and D. Starobinski, "Evaluation of the masked node problem in ad hoc wireless lans," *Mobile Computing, IEEE Transactions on*, vol. 4, no. 5, pp. 430–442, 2005.

[32] A. Hassan and T. Larsson, "On the requirements on models and simulator design for integrated vanet simulation," in *8th International Workshop on Intelligent Transportation (WIT 2011), 22-23 March, 2011, Hotel Hafen Hamburg, Hamburg, Germany*, pp. 191–196, 2011.

[33] J. L. Pereira and R. J. Rossetti, "An integrated architecture for autonomous vehicles simulation," in *Proceedings of the 27th annual ACM symposium on applied computing*, pp. 286–292, ACM, 2012.

[34] A. Winder, M. Brackstone, P. D. Site, and M. Antognoli, "Traffic management for land transport: Research to increase the capacity, efficiency, sustainability and safety of road, rail and urban transport networks," 2009.

[35] H. Liimatainen, "Utilization of fuel consumption data in an ecodriving incentive system for heavy-duty vehicle drivers," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1087–1095, 2011.

[36] M. Munoz-Organero and V. C. Magana, "Validating the impact on reducing fuel consumption by using an ecodriving assistant based on traffic sign detection and optimal deceleration patterns," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 1023–1028, 2013.

[37] C. Suthaputchakun, Z. Sun, and M. Dianati, "Applications of vehicular communications for reducing fuel consumption and co 2 emission: the state of the art and research challenges," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 108–115, 2012.

[38] T. Bektaş and G. Laporte, "The pollution-routing problem," *Transportation Research Part B: Methodological*, vol. 45, no. 8, pp. 1232–1250, 2011.

[39] Y. Chen, M. G. Bell, and K. Bogenberger, "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 1, pp. 14–20, 2007.

[40] C. Sommer, R. Krul, R. German, and F. Dressler, "Emissions vs. travel time: simulative evaluation of the environmental impact of its," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pp. 1–5, IEEE, 2010.

[41] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1694–1704, 2012.

[42] M. Vasirani and S. Ossowski, "A computational market for distributed control of urban road traffic systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 2, pp. 313–321, 2011.

[43] K. Ahn and H. Rakha, "The effects of route choice decisions on vehicle energy consumption and emissions," *Transportation Research Part D: Transport and Environment*, vol. 13, no. 3, pp. 151–167, 2008.

[44] H. Rakha, K. Ahn, and K. Moran, "Integration framework for modeling eco-routing strategies: Logic and preliminary results," *International Journal of Transportation Science and Technology*, vol. 1, no. 3, pp. 259–274, 2012.

[45] K. Ahn and H. A. Rakha, "Network-wide impacts of eco-routing strategies: a large-scale case study," *Transportation Research Part D: Transport and Environment*, vol. 25, pp. 119–130, 2013.

[46] H. Yang, X. Wang, and Y. Yin, "The impact of speed limits on traffic equilibrium and system performance in networks," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1295–1307, 2012.

[47] J. Zhang and P. A. Ioannou, "Longitudinal control of heavy trucks in mixed traffic: environmental and fuel economy considerations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 92–104, 2006.

[48] O. Servin, K. Boriboonsomsin, and M. Barth, "An energy and emissions impact evaluation of intelligent speed adaptation," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pp. 1257–1262, IEEE, 2006.

[49] P. Allaby, B. Hellinga, and M. Bullock, "Variable speed limits: Safety and operational impacts of a candidate control strategy for freeway applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 4, pp. 671–680, 2007.

[50] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak, "Vehicular networks for reduction of fuel consumption and co 2 emission," in *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, pp. 671–676, IEEE, 2010.

[51] V. Khorani, F. Razavi, and V. R. Disfani, "A mathematical model for urban traffic and traffic optimization using a developed ica technique," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1024–1036, 2011.

[52] B. Liu, D. Ghosal, C.-N. Chuah, and H. M. Zhang, "Reducing greenhouse effects via fuel consumption-aware variable speed limit (fc-vsl)," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 111–122, 2012.

[53] M. Ferreira and P. M. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 284–295, 2012.

[54] L.-D. Chou, B.-T. Deng, D. C. Li, and K.-W. Kuo, "A passenger-based adaptive traffic signal control mechanism in intelligent transportation systems," in *ITS Telecommunications (ITST), 2012 12th International Conference on*, pp. 408–411, IEEE, 2012.

[55] A. A. Malikopoulos and J. P. Aguilar, "An optimization framework for driver feedback systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 955–964, 2013.

[56] R. Kishore Kamalanathsharma and H. A. Rakha, "Leveraging connected vehicle technology and telematics to enhance vehicle fuel efficiency in the vicinity of signalized intersections," *Journal of Intelligent Transportation Systems*, pp. 1–12, 2014.

[57] I. H. Zohdy and H. A. Rakha, "Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept," *Journal of Intelligent Transportation Systems*, pp. 1–16, 2014.

[58] I. Zohdy and H. Rakha, "Enhancing roundabout operations via vehicle connectivity," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2381, pp. 91–100, 2013.

[59] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, pp. 591–656, 2008.

[60] Q. Jin, G. Wu, K. Boriboonsomsin, and M. Barth, "Advanced intersection management for connected vehicles using a multi-agent systems approach," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 932–937, IEEE, 2012.

[61] S. Huang, A. W. Sadek, and Y. Zhao, "Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1201–1214, 2012.

[62] L. C. Bento, U. Nunes, F. Moita, and A. Surrecio, "Sensor fusion for precise autonomous vehicle navigation in outdoor semi-structured environments," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 245–250, IEEE, 2005.

[63] N. M. Drawil and O. Basir, "Intervehicle-communication-assisted localization," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 678–691, 2010.

[64] W. Li and H. Leung, "Simultaneous registration and fusion of multiple dissimilar sensors for cooperative driving," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, no. 2, pp. 84–98, 2004.

[65] M. J. Woo, J. W. Choi, and H. S. Han, "Carrier phase gps/millimeter-wave radar for vehicle platooning," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 3, pp. 1548–1552, IEEE, 2001.

[66] G. Challita, S. Mousset, F. Nashashibi, and A. Bensrhair, "An application of v2v communications: Cooperation of vehicles for a better car tracking using gps and vision systems," in *Vehicular Networking Conference (VNC), 2009 IEEE*, pp. 1–6, IEEE, 2009.

[67] W.-B. Zhang, R. E. Parsons, and T. West, "An intelligent roadway reference system for vehicle lateral guidance/control," in *American Control Conference, 1990*, pp. 281–286, IEEE, 1990.

[68] H. Li, F. Nashashibi, and G. Toulminet, "Localization for intelligent vehicle by fusing mono-camera, low-cost gps and map data," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1657–1662, IEEE, 2010.

[69] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 406–412, IEEE, 2013.

[70] J. Rife and X. Xiao, "Estimation of spatially correlated errors in vehicular collaborative navigation with shared gnss and road-boundary measurements," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pp. 1667–1677, 2001.

[71] J. Rife, "Collaborative vision-integrated pseudorange error removal: team-estimated differential gnss corrections with no stationary reference receiver," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 15–24, 2012.

[72] V. Drevelle and P. Bonnifait, “Integrity zone computation using interval analysis,” in *ENC-GNSS 2009 European Navigation Conference-Global Navigation Satellite Systems*, pp. Poster–Session, 2009.

[73] V. Drevelle and P. Bonnifait, “Localization confidence domains via set inversion on short-term trajectory,” *Robotics, IEEE Transactions on*, vol. 29, no. 5, pp. 1244–1256, 2013.

[74] *http://www.gpsoftnav.com/*.

[75] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[76] A. Tetewsky and A. Soltz, “Gps matlab toolbox review,” *GPS World*, vol. 9, no. 10, pp. 50–57, 1998.

[77] E. Realini, *goGPS: free and constrained relative kinematic positioning with low cost receivers*. PhD thesis, Politecnico di Milano, Feb. 2009.

[78] A. Andrews *et al.*, “Theoretical and empirical analysis of path magnetic lane tracking for the intelligent vehicle highway system,” *California Partners for Advanced Transit and Highways (PATH)*, 1992.

[79] L. Madan and K. A. B. Bhushan, “Bresenham’s lines algorithm,” vol. 4, 2014.

[80] S. Sekhavat and J. Hermosillo, “The cycab robot: a differentially flat system,” in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 1, pp. 312–317vol.1, 31 Oct.-5 Nov. 2000.

[81] G. Baille, P. Garnier, H. Mathieu, and R. Pissard-Gibollet, “Le cycab de l’INRIA Rhône-Alpes,” Research Report RT-0229, INRIA, 1999. Projet SERVICE ROBOTIQUE.

[82] J. Hermosillo, C. Pradalier, and S. Sekhavat, “Modeling odometry and uncertainty propagation for a bi-steerable car,” in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 401–406vol.2, 17-21 June 2002.

[83] L. Jaulin, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, vol. 1. Springer Science & Business Media, 2001.

[84] E. Walter and L. Jaulin, “Guaranteed characterization of stability domains via set inversion,” *Automatic Control, IEEE Transactions on*, vol. 39, no. 4, pp. 886–889, 1994.

[85] B. Chen and H. H. Cheng, “A review of the applications of agent technology in traffic and transportation systems,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 485–497, 2010.

[86] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.

[87] A. C. Lopes, F. Moita, U. Nunes, and R. Solea, “An outdoor guidepath navigation system for amrs based on robust detection of magnetic markers,” in *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pp. 989–996, IEEE, 2007.

[88] S. Ray, J. B. Carruthers, and D. Starobinski, “Evaluation of the masked node problem in ad hoc wireless lans,” *Mobile Computing, IEEE Transactions on*, vol. 4, no. 5, pp. 430–442, 2005.

[89] S. Suzuki, M. Tsunoda, M. Taniguchi, and T. Takagi, “Characterization of road traffic flow from measured data of speed and time-headway-relationship between density (k), flow rate (q) and speed (v)-,” in *SICE, 2007 Annual Conference*, pp. 1657–1661, IEEE, 2007.

[90] J. H. Sika, J., T. Bertram, J. P. Pauwlussen, and M. Hiller, “Test facility for lateral control of scaled vehicle in an automated highway system,” in *Proceedings of the 8th Mechatronics Forum - International Conference Mechatronics 2002, Twente/Netherlands*, pp. 142–150, 2002.

[91] E. Felipe and F. Navin., “Canadian researchers test driver response to horizontal curves.” Road Management & Engineering Journal TranSafety, Inc., 1998.

[92] TranSafety, “Simulated on-the-road emergencies used to test stopping sight distance assumptions.” Road Management & Engineering Journal TranSafety, Inc., 1997.

[93] M. P. Dixon and J. Birchman, “Validation of arterial travel-time estimation models using field data and simulation,” tech. rep., National Institute for Advanced Transportation Technology, University of Idaho, 2007.

[94] B. Park and J. Schneeberger, “Microscopic simulation model calibration and validation: case study of vissim simulation model for a coordinated actuated signal system,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1856, pp. 185–192, 2003.

[95] P. Mellodge, “Feedback control for a path following robotic car,” Master’s thesis, Faculty of the Virginia Polytechnic, Institute and State University, 2002.

[96] L. C. Bento, *FLC and CF path-following Controllers Comparison.* Technical Report, http://www2.isr.uc.pt/ conde/isr-trafsim/, 2011.

[97] U. Nunes and L. C. Bento, “Data fusion and path-following controllers comparison for autonomous vehicles,” *Nonlinear Dynamics*, vol. 49, no. 4, pp. 445–462, 2007.

[98] L. C. Bento, R. Parafita, and U. Nunes, “Intelligent traffic management at intersections supported by v2v and v2i communications,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 1495–1502, IEEE, 2012.

[99] K. B. Wipke, M. R. Cuddy, and S. D. Burch, "Advisor 2.1: A user-friendly advanced powertrain simulation using a combined backward/forward approach," *Vehicular Technology, IEEE Transactions on*, vol. 48, no. 6, pp. 1751–1761, 1999.

[100] L. Guzzella and A. Sciarretta, *Vehicle propulsion systems*, vol. 1. Springer, 2007.

[101] L. Guzzella and C. Onder, *Introduction to modeling and control of internal combustion engine systems.* Springer Science & Business Media, 2009.

[102] F. Pinkerton and J. Herbst, *Hydrogen-powered cars.* McGraw-Hill Yearbook of Science and Technology, 2010.

[103] K. M. Johnson, "A plug-in hybrid electric vehicle loss model to compare well-to-wheel energy use from multiple sources," 2008.

[104] R. Edwards, J.-F. Larivé, V. Mahieu, P. Rouveirolles, *et al.*, "Well-to-wheels analysis of future automotive fuels and powertrains in the european context," *CONCAWE, European Council for Automotive R&D, JRC Joint Research Centre of the European Commission, Version 2c*, 2007.

[105] A. García-Álvarez, P. Perez-Martinez, and I. González-Franco, "Energy consumption and carbon dioxide emissions in rail and road freight transport in spain: a case study of car carriers and bulk petrochemicals," *Journal of Intelligent Transportation Systems*, vol. 17, no. 3, pp. 233–244, 2013.

[106] P. 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 *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, ICRA 2001, May 21-26, 2001, Seoul, Korea*, pp. 1597–1602, IEEE, 2001.

[107] A. Surrecio, U. Nunes, and R. Araujo, "Fusion of odometry with magnetic sensors using kalman filters and augmented system models for mobile robot navigation," in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, vol. 4, pp. 1551–1556, June 20-23, 2005.

[108] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.

[109] P. D. Groves, Z. Jiang, M. Rudi, and P. Strode, "A portfolio approach to nlos and multipath mitigation in dense urban areas," 2013.

[110] V. Drevelle, *Study of robust set estimation methods for a high integrity multi-sensor localization. Application to navigation in urban areas.* Doctor of Philosophy in Engineering Thesis, Universite de Technologie de Compiegne, 2011.

[111] V. Drevelle and P. Bonnifait, "Integrity zone computation using interval analysis," in *ENC-GNSS 2009 European Navigation Conference-Global Navigation Satellite Systems*, pp. Poster–Session, 2009.

[112] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel, "Guaranteed robust nonlinear estimation with application to robot localization," *IEEE Transactions on systems, man and cybernetics; Part C Applications and Reviews*, vol. 32, no. 4, pp. 374–382, 2002.

[113] I. Gat, M. Benady, and A. Shashua, "A monocular vision advance warning system for the automotive aftermarket," in *SAE World Congress & Exhibition*, vol. 2005, 2005.

[114] J. Santa, B. Ubeda, R. Toledo, and A. F. Skarmeta, "Monitoring the position integrity in road transport localization based services," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pp. 1–5, IEEE, 2006.

[115] M. Tossaint, J. Samson, F. Toran, J. VENTURA-TRAVESET, M. HERNANDEZ-PAJARES, J. Juan, J. Sanz, and P. RAMOS-BOSCH, "The stanford–esa integrity diagram: A new tool for the user domain sbas integrity assessment," *Navigation*, vol. 54, no. 2, pp. 153–162, 2007.

[116] O. Le Marchand, P. Bonnifait, J. Ibanez-Guzmán, D. Betaille, and F. Peyret, "Characterization of gps multipath for passenger vehicles across urban environments," *ATTI dell'Istituto Italiano di Navigazione*, no. 189, pp. 77–88, 2009.

[117] U. Kiencke and L. Nielsen, *Automotive Control Systems, For Engine, Driveline, and Vehicle.* Springer Verlag, 2000.

[118] A. Fod, A. Howard, and M. Mataric, "A laser-based people tracker," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, pp. 3024–3029, 11-15 May 2002.

[119] P. Jensfelt and H. Christensen, "Laser based pose tracking," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4, pp. 2994–3000vol.4, 10-15 May 1999.

[120] K. C. Dietmayer, J. Sparbert, and D. Streller, "Model based object classification and object tracking in traffic scenes from range images," in *Proceedings of IV IEEE Intelligent Vehicles Symposium*, 2001.

[121] A. Mendes, L. Bento, and U. Nunes, "Multi-target detection and tracking with a laser scanner," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 796–801, 14-17 June 2004.

[122] A. D. Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Planning robot motion* (J.-P. Laumond, ed.), Berlin, DE: Springer-Verlag, 1998.

[123] J.-P. P. Laumond, *Robot Motion Planning and Control.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.

Part I

Appendices

Appendix A

Fuzzy-logic background and FLC for 4WS Kinematic Vehicle

WMR Fuzzy-Logic background

The fuzzification transforms numerical variables into fuzzy sets, which can be manipulated by the controller. A fuzzy set in a universe of discourse U is defined by μ_A, with values between 0 and 1 , i.e. $\mu_A : U \mapsto [0,1]$. A fuzzy set A in a universe of discourse U can then be expressed by u and its membership level $\mu_A(u)$:

$$A = \{(u, \mu_A(u)) : u \in U\} \tag{A.1}$$

A mathematical function that defines a fuzzy set on the universe of discourse is called membership function. The controller uses fuzzy functions to encode inputs and outputs. A membership function $\mu(u)$ assign a correspondence between values in the range of [0,1] to elements u in the universe of discourse. The membership functions used were typical membership functions in fuzzy expert systems, such has triangular and right and left shoulders.

Triangular membership function- is given by the following equation:

$$T(u;a,b,c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \le u \le b \\ \frac{c-u}{c-b} & c \le u \le d \\ 0 & u > c \end{cases} \tag{A.2}$$

left and right shoulder membership functions - are respectively given by the following equations:

$$T(u;a,b) = \begin{cases} 1 & u < a \\ \frac{b-u}{b-a} & a \le u \le b \\ 0 & u > b \end{cases} \tag{A.3}$$

$$T(u;a,b) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \le u \le b \\ 1 & u > b \end{cases} \tag{A.4}$$

Generally the smoothness of higher order membership functions is not directly reflected on the quality of the fuzzy-logic system output, therefore the triangular and right and left shoulders membership functions were chosen due to the easiness of implementation and to low computation resources used.

Fuzzy Set Theory defines Fuzzy Operators on Fuzzy Sets. Fuzzy-logic uses IF/THEN rules, these rules are conditional statements expressed in the form:

IF x is A THEN y is B

where x and y are linguistic variables, and A and B are linguistic values determined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively.

For the following rule the “IF $A(u)$” part is called the antecedent or premise, while the then-part of the rule “ THEN $B(v)$” is called the consequent or conclusion.

$$A \Rightarrow B \text{ or IF } A(u) \text{ THEN } B(v)$$

A fuzzy relation in a conditional statements is given by the cross product between A and B ($A \times B$), this represents all possible combinations between the elements of each universe of discourse, i.e.

$$A \times B = \int_{U \times V} \mu_A(u) t \mu_B(v)/(u,v) \tag{A.5}$$

The logical operator AND that joins together two antecedents in a production rule is called conjunction; the membership function $\mu_{A\cap B}(u)$ of the conjunction $A \cap B$ is defined by:

$$\mu_{A\cap B}(u) = \mu_A(u) t \mu_B(u), u \in U \tag{A.6}$$

where t is the t-norm, i.e. two-input function $[0,1] \times [0,1] \mapsto [0,1]$ that describes a superset of fuzzy intersection (AND) operators such as:

minimum

$$xty = x \wedge y = min(x,y) \tag{A.7}$$

algebraic product

$$xty = x * y = x.y \tag{A.8}$$

The logical operator OR that joins together two antecedents in a production rule is called disjunction, the membership function $\mu_{A\cup B}(u)$ of the disjunction $A \cup B$ is defined by:

$$\mu_{A\cup B}(u) = \mu_A(u) s \mu_B(u), u \in U \tag{A.9}$$

where s is the s-norm, i.e. two-input function $[0,1] \times [0,1] \mapsto [0,1]$ that describes a superset of fuzzy intersection (OR) operators such as:

maximum

$$xsy = x \vee y = max(x,y) \tag{A.10}$$

algebraic sum

$$xsy = x + y = x + y - xy \tag{A.11}$$

The process of reasoning based on fuzzy-logic is called “inference method”. Fuzzy inference includes four steps: fuzzification of the input variables, rule evaluation, aggregation of the rule outputs and defuzzification

The “inference method” used is described here (Mamdani max-min), it is the combination of “inference” and “composition”. The term “max-min inference” mean the combination of “max” composition and “min” inference. Consider the following rule type has reference for the knowledge base presented in A.1:

$$R^{(l)}\text{: } \underbrace{\text{IF } x_1 \text{ is } A_1^l \text{ AND } \cdots \text{ AND } x_n \text{ is } A_n^l}_{antecedent} \underbrace{\text{THEN } y \text{ is } B^l}_{consequent}$$

where:

- $x_1, ..., x_n \in U$ are the input linguistic variables and $y \in V$ is the output linguistic variable.
- $A_1^l (i = 1, ..., n)$ are fuzzy sets in U_i
- B^l are the output fuzzy set in V
- $l = 1, ..., M$, number of rules

Table A.1: Knowledge base

R1: IF **x** is A_1 AND **y** is B_1 THEN **z** is C_1 OR
R2: IF **x** is A_2 AND **y** is B_2 THEN **z** is C_2 OR
$\vdots$
Rn: IF **x** is A_n AND **y** is B_n THEN **z** is C_n

Under "inference", the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. In the "min" inferencing, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth (fuzzy-logic AND)

$$\mu_{Ci}(z) = min(min(\mu_{Ai}(x_0), \mu_{Bi}(y_0)), \mu_{Ci}(z)) \tag{A.12}$$

Under "composition", all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. In "max" composition, the combined output fuzzy subset is constructed by taking the pointwise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy-logic OR).

$$\mu_C(z) = max(\mu_{C1}(z), \cdots, \mu_{Cn}(z)) \tag{A.13}$$

The last step in fuzzy inference is called defuzzification; it is the process of converting a combined output of fuzzy rules into a crisp (numerical) value. The input for the defuzzification process is the aggregate set and the output is a single number

The *center of gravity* defuzzification method was used.

centre of gravity - also called the centroid, where a vertical line would slice the aggregate set into two equal masses:

$$u = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \tag{A.14}$$

where x_i is the degree of activation of the *ith* rule and $\mu(x_i)$ is the output membership function.

Because more than one output term can be evaluated as valid, the defuzzification method must be a compromise between different results. The center of gravity method was chosen because it takes into account, better than any other method, the distribution of the resultant fuzzy set. In this method the defuzzified value u is a weighted sum of the term membership.

Fuzzy-Logic path following Controller (FLC) for a 4WS Kinematic Vehicle

This section addresses the path following problem using fuzzy-logic. It was chosen to develop a fuzzy-logic based controller, so that non-skilled programmers could tweak or test new controlling setups. The fuzzy-logic based PFC was firstly evaluated on a simple kinematic model (*i.e.* a differential drive vehicle) (found in Appendix A), and further development is built upon the first iteration on a rather complex kinematic model, (*i.e.* a single and double steered car-like vehicle).

The fuzzy-logic based PFC is made up of two main modules: VP and LC. The LC as the following set of inputs:

$$u_{FL} = [\theta_e^{cp}, d_e^{cp}, \Delta\theta_e^{cp}, \Delta d_e^{cp}, t_{imp}, d_{ile}, c(s), v] \tag{A.15}$$

where v denotes the linear reference velocity, t_{imp} is the time-to-collision computed in the MTDT, d_{ile} is the inline lateral error (see section A) and the differential errors Δd_e^{cp} and $\Delta\theta_e^{cp}$, at the CP, are given by

$$\Delta d_e^{cp} = d_e^{cp}(k) - d_e^{cp}(k-1) \tag{A.16}$$

and

$$\Delta\theta_e^{cp} = \theta_e^{cp}(k) - \theta_e^{cp}(k-1) \tag{A.17}$$

Collision avoidance is achieved by controlling the vehicle's reference velocity, reducing or even stopping the vehicle in situations of eminent danger. The main goal of the path-following controller is to ensure that the vehicle follows the predefined reference path with appropriate orientation. For the fuzzy-logic controller this can be understood as a task of minimizing the vehicle lateral and heading errors $(d_e^{cp}, \theta_e^{cp})$ with respect to the reference path, at a given CP located at a distance L_a denoted by lookahead distance, as illustrated in Figure 4.7.

$$\theta_e = v_1 \frac{\tan\varphi}{L} - v_1 \frac{c(s)\cos(\theta_e)}{1 - d_e^{tg} c(s)} \tag{A.18}$$

$$c(s)\left[v_1\cos(\theta_e) + \frac{v_1 d\tan\varphi}{L} - \dot{\theta}_e d\right] = \frac{v_1 d\tan\varphi}{L} - \dot{\theta}_e \tag{A.19}$$

For both controllers the curvature along the path $c(s)$ is estimated as described in [95]. From the third row of (B.22) one can obtain a linearly parameterizable system in $c(s)$ written by:

$$y = wa \tag{A.20}$$

where

$$y = \frac{v_1 d_e^{tg}\tan\varphi}{L} - \dot{\theta}_e^{tg} \tag{A.21}$$

$$w = v_1\cos(\theta_e^{tg}) + \frac{v_1 d_e^{tg}\tan\varphi}{L} - \dot{\theta}_e^{tg} d_e^{tg} \tag{A.22}$$

$$a = c(s) \tag{A.23}$$

Knowing the w and y an estimate of a, i.e. $\hat{a}$, is obtained using the least squares estimator:

$$J = \int_0^t (y - w\hat{a})^2\, d\tau \tag{A.24}$$

Solving for $\hat{a}$ so as to minimize J, the following update equation for $\hat{a}$ is obtained:

$$\dot{\hat{a}} = P(wy - w^2\hat{a}) \tag{A.25}$$

where P and its update equation $\dot{P}$ are given by:

$$P = \frac{1}{\int_0^t w^2 d\tau} \tag{A.26}$$

$$\dot{P} = -P^2 w^2 \tag{A.27}$$

Figure 4.8 presents the controller architecture. The architecture is divided in high and low level controller. The high level controller LC computes the control commands for the low level controller, some of the high level modules are also presented has they provide inputs to the LC. The LC is based on fuzzy-logic, it is composed by four independent modules: front steering controller, rear steering switch, velocity command generator and lookahead distance computation (see Figure 4.8). The low level controller is responsible for the traction control and besides regular vehicles it also enables the control of electrical 4WD and 4WS vehicles.

The controller provide a control vector ($[\varphi_c, v_c, \varphi_{sw}]$) to the traction control level,where φ_c (in degrees) is the steering angle, $v_c[ms^{-1}]$ is the velocity command, and φ_{sw} is the rear steering switch that controls the two possible driving modes: DDM and PDM (see Figure A.1). In DDM the rear axle steers in opposite direction of the front axle, while in PDM the rear and front axle steers in the same direction, both modes are 4WS. If the vehicle is 2WS then the φ_{sw} as no effect on the low level traction controller.

Fuzzy System	Structure
input variables	8
output variables	4
intermediate variables	1
rule blocks	5
rules	615
membership functions	48

Table A.2: Overall Fuzzy-Logic Controller characteristics

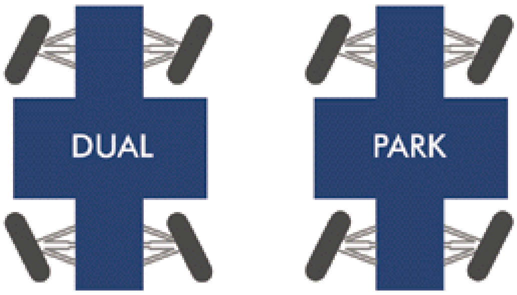


Figure A.1: Driving Modes: Park Driving Mode (DDM) and Park Driving Mode (PDM).

The fuzzy LC is composed by four independent modules: front steering controller, rear steering switch, velocity command generator and lookahead distance computation (see Figure 4.8). In order to properly avoid collisions with obstacles the time-to-impact t_{imp} (also referred here as time-to-collision), provided by the MTDT, is integrated in the velocity command generator. All modules are fuzzy-logic based. Figure 4.8 shows the LC identifying the fuzzy-logic inference flow from the input variables to the output variables. The fuzzy controller is characterized in Table A.2. The knowledge base of the LC expresses how the system should react. Part of the the fuzzy knowledge base, the input sets, the output sets and some of the membership functions are presented in the following paragraphs.

Front steering module

The front steering module (composed by controller 1 and 2) computes the steering command φ_c. The purpose is to minimize both the orientation error θ_e^{cp}, and the lateral error d_e^{cp}. A steering increment fuzzy variable (φ_{inc}) is computed in order to achieve a

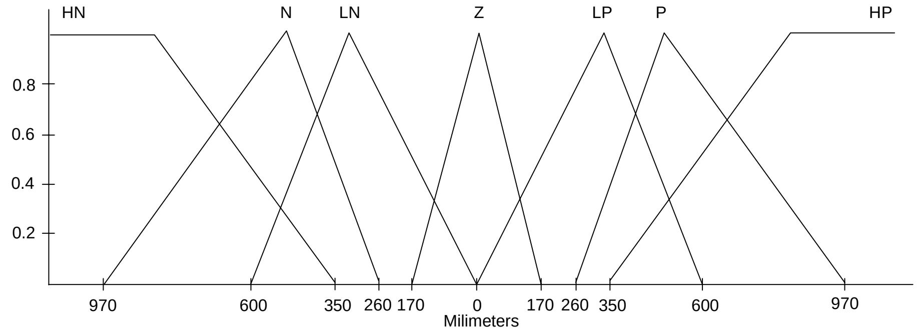


Figure A.2: Lateral error (d_e^{cp}) membership functions distribution

d_e^{cp}; θ_e^{cp}	Linguistic terms
HN	High Negative
N	Negative
LN	Low Negative
Z	Zero
LP	Low Positive
P	Positive
HP	High Positive

Table A.3: Linguistic terms of d_e^{cp} and θ_e^{cp}.

Δd_e^{cp}; $\Delta\theta_e^{cp}$	Linguistic terms
D	Decreasing
C	Constant
I	Increasing

Table A.4: Linguistic terms of Δd_e^{cp} and $\Delta\theta_e^{cp}$

faster recovery from an undesirable pose. φ_{inc} is the output of a fuzzy module which has as inputs $c(s)$ and $\Delta\theta_e^{cp}$. This module ensures a geometrical convergence towards the path to be followed.

Rear steering switch module

The Rear Steering Switch module decides whether the rear axle should steer to opposite direction of the front axle DDM or in the same direction PDM. The inputs of this module are Δd_e^{cp}, and the inline lateral error d_{ile} given by

$$d_{ile} = |\frac{d_e^{cp}}{d_e^{cg}}| + |\theta_e^{cp}| \tag{A.28}$$

If Δd_e^{cp} is decreasing and d_{ile} is small, this module steers the rear wheels in the same direction as the front wheels; the result is a decreasing of the vehicle's yaw motion. The yaw motion is necessary for executing a manoeuvre but is not desired from the point of view of the vehicle's stability control [90]. This module was only implemented and tested in simulations.

Velocity command generator module

$c(s)$; d_{ile}; v; φ_{inc}	Linguistic terms
Z	Zero
M	Medium
H	High

Table A.5: Linguistic terms of κ, d_{ile}, v and φ_{inc}

t_{imp}	Linguistic terms
L	Low
M	Medium
H	High

Table A.6: Linguistic terms of t_{imp}

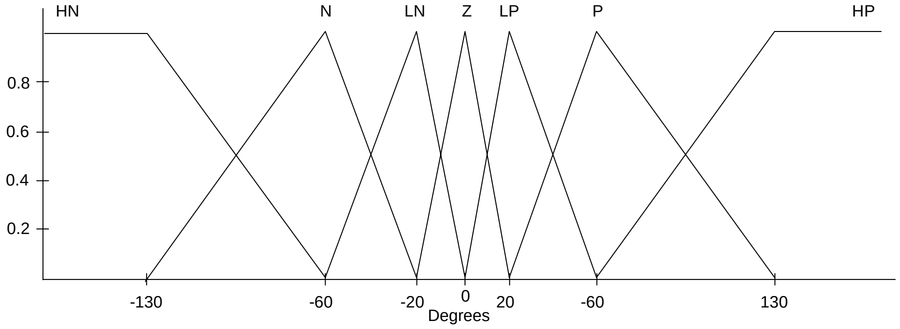


Figure A.3: Orientation error (θ_e) membership functions distribution

The inputs of this module are Δd_e^{cp}, d_e^{cp}, $\Delta\theta_e^{cp}$, θ_e^{cp} and the t_{imp}. This module computes a weight factor assigning a level of significance to the reference velocity, i.e. if the errors have a high magnitude or the time-to-collision has a low magnitude then the velocity must be decreased, otherwise the reference velocity is applied. This module is of extreme importance since collision avoidance is decided here, i.e. if the t_{imp} is small, then the vehicle velocity is reduced or the vehicle is even stopped; if t_{imp} is high the vehicle velocity (weight factor) is not affected.

Lookahead distance computation module

This module computes the lookahead distance, L_a, which is a function of the vehicle velocity, v. If the velocity increases, the damping factor of the closed loop system gets worse and is improved by increasing the lookahead distance. The lookahead distance provides a prediction behaviour to the controller, since it enables the control point to be far ahead of the CG of the vehicle, see Figure 4.7.

Fuzzy input sets

The lateral error d_e^{cp} and the orientation error θ_e^{cp} membership functions distributions are each one composed by a fifth-pronged triangular membership function and two shoulder membership functions (see Figure A.2 and Figure A.3). Their linguistic terms are shown in Table A.3.

The differential d_e^{cp} (Δd_e^{cp}) and the differential θ_e ($\Delta\theta_e$) membership functions distributions are each composed by a triangular membership function and two shoulder membership functions. Their linguistic terms are shown in Table A.4.

The curvature $c(s)$, inline lateral error d_{ile}, vehicle velocity v and the steering increment fuzzy variable φ_{inc} membership functions distributions are each one composed by a triangular membership function and two shoulder membership functions. Their linguistic terms are shown in Table A.5.

The time-to-collision (t_{imp}) membership functions distribution is similar to the previous one. Its linguistic terms are shown in Table A.6.

Fuzzy output sets

The steering command φ_c membership functions distribution is a fifth-pronged triangular membership function and two shoulder membership functions (see Figure A.4). Its linguistic terms are shown in Table A.7.

The rear steering switch φ_{sw} membership functions distribution is a two non-overlapping shoulder membership function, corresponding to the two possible driving modes. Its linguistic terms are shown in Table A.8.

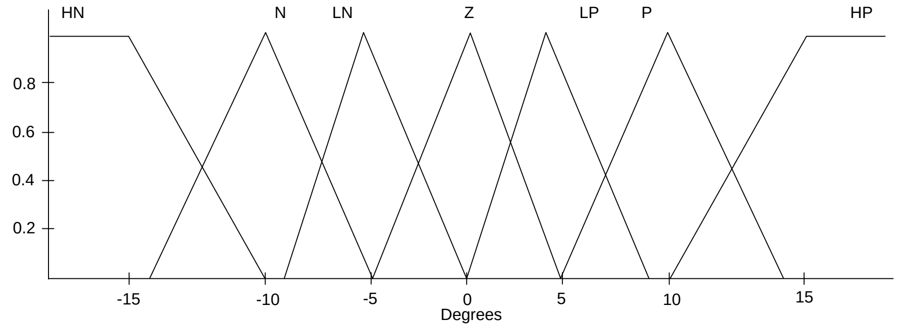


Figure A.4: Steering command (φ_c) membership functions distribution

φ_c	Linguistic terms
HN	High Negative
N	Negative
LN	Low Negative
Z	Zero
LP	Low Positive
P	Positive
HP	High Positive

Table A.7: Linguistic terms of φ_c

φ_{sw}	Linguistic terms
DDM	Dual Driving Mode
PDM	Park Driving Mode

Table A.8: Linguistic terms of φ_{sw}

v_c; L_a	Linguistic terms
Z	Zero
M	Medium
H	High

Table A.9: Linguistic terms of v_c and L_a

The velocity v_c and the lookahead distance L_a membership functions distributions are each one composed by a triangular membership function and two shoulder membership functions. Its linguistic terms are shown in Table A.9:

Fuzzy knowledge base

Tables A.10-A.13 show the fuzzy rules that constitute part of the knowledge base of the LC and express how the system should react.

Front steering module II
1-If d_e^{cp} is HN and θ_e^{cp} is HN and φ_{inc} is H then φ_c is HP;
2-If d_e^{cp} is LN and θ_e^{cp} is LP and φ_{inc} is Z then φ_c is LN;
3-If d_e^{cp} is LN and θ_e^{cp} is HN and φ_{inc} is H then φ_c is P;
4-If d_e^{cp} is MN and θ_e^{cp} is Z and φ_{inc} is Z then φ_c is MP.

Table A.10: Fuzzy rules subset of the front steering controller II

Velocity planner

The VP module calculates the linear reference velocity, as well as determines the local reference trajectory points. One main objective taken into account was to make the trip as comfortable as possible, i.e. to give the system the capability of fully controlling the smoothness of the acceleration profile either lateral or longitudinal.

A Canadian study [91] used a highway testing ground to test speed and lateral acceleration on both wet and dry pavement on horizontal curves. They found that *"comfortable*

Rear steering switch module
1-If Δd_e^{cp} is I and d_{ile} is Z then ϕ_{sw} is PDM;
2-If Δd_e^{cp} is I and d_{ile} is M then ϕ_{sw} is DDM;
3-If Δd_e^{cp} is I and d_{ile} is H then ϕ_{sw} is DDM;
4-If Δd_e^{cp} is D and d_{ile} is H then ϕ_{sw} is DDM.

Table A.11: Fuzzy rules of the rear steering switch

Velocity control module
1-If d_e^{cp} is HN and θ_e is Z and Δd_e^{cp} is I and $\Delta\theta_e$ is C and t_{imp} is H then v_c is M ;
2-If d_e^{cp} is Z and θ_e is Z and Δd_e^{cp} is I and $\Delta\theta_e$ is I and t_{imp} is H then v_c is M;
3-If d_e^{cp} is Z and θ_e is Z and Δd_e^{cp} is C and $\Delta\theta_e$ is C and t_{imp} is H then v_c is H;
4-If d_e^{cp} is HN and θ_e is HN and Δd_e^{cp} is C and $\Delta\theta_e$ is C and t_{imp} is L then v_c is Z;

Table A.12: Fuzzy rules of the velocity control module

Lookahead module
1-If v is Z then L_a is Z;
2-If v is M then L_a is M;
3-If v is H then L_a is H.

Table A.13: Fuzzy rules of the lookahead module

lateral acceleration" and *"speed environment"* limited the driver's speed, while pavement surface conditions (dry or wet) and the driver's gender did not. Drivers adjusted their comfortable speed according to their comfortable lateral acceleration tolerance, approximately between $0.35g$ and $0.40g$. Another study [92] revealed the comfortable longitudinal acceleration, i.e. steady deceleration under expected-stop conditions; drivers generally exert an average steady braking force of $-0.35g$. This amount of braking force seems comfortable for most drivers.

The previous acceleration limits were used to set up the maximum comfort acceleration a_{mc} and maximum comfort velocity v_{mc}. The maximum acceleration without slipping a_{mws} and maximum velocity without slipping v_{mws} still had to be computed to cope with unexpected situations. To estimate v_{mws}, it is necessary to know the forces that actuate on the vehicle, which are basically the horizontal forces, the wheel ground contact forces, the force that the vehicle exerts on the ground and the wind force over the vehicle (air resistance). In this study we consider a plane road, and no wind force effects are taken into account.

The friction force is proportional to the normal reaction, where the proportionality factor is the friction coefficient (static or dynamic):

$$F_{fric} = \mu F_z \tag{A.29}$$

The force that the vehicle exerts on the ground (weight P) is proportional to its mass m,

$$P = m \times g \tag{A.30}$$

where g is the gravity acceleration. Furthermore there is the centrifugal force that actuates on the vehicle when it describes a curvilinear trajectory,

$$F_c = m\frac{v^2}{r} \tag{A.31}$$

where v is the vehicle velocity and r the curvature radius. The vehicle, when describing a curvilinear trajectory, should not slide (either on the inner or the outer side of the curve).

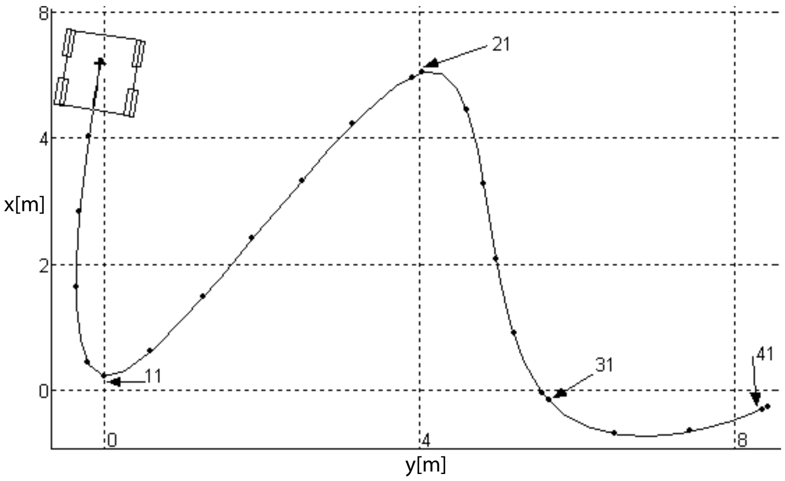


Figure A.5: Vehicle following a given path. In the figures some sampling points are marked (11, 21, 31 and 41) (x-axis and y-axis are in meters)

So, in order to avoid sliding, the sum of the forces, along y-axis (lateral direction) must be null:

$$F_c - F_{fric} - P\sin(\psi) = 0 \tag{A.32}$$

where ψ denotes the *roll* angle of the vehicle.

The friction force is proportional to the normal reaction, where the proportionality factor is the friction coefficient μ (static or dynamic).

Taking into account the previous assumptions one can derive the maximum velocity without slipping

$$v_{mws} = \sqrt{rg(\mu\cos(\psi) + \sin(\psi))} \tag{A.33}$$

where g is the gravity acceleration, r the curvature radius and ψ denotes the *roll* angle of the vehicle.

The main difficulty in the calculation of the reference velocity is the friction coefficient, which must be estimated with a significant precision. An estimate of μ is obtained by the following equation:

$$\mu(S) = (c_1(1 - e^{-c_2 S}) - c_3 S)e^{-c_4 S v}(1 - c_5 F_z^2) \tag{A.34}$$

where S is the resultant slip, and the constants c_i $(i = 1..5)$ are characteristic parameters of various types of road [117]. The velocity v_{mc} determines the intended vehicle velocity used in the vehicle motion. The v_{mws} has a more ruggedness profile, as can be observed in Figure A.6, which shows the velocity profiles correspondent to the example of a vehicle following the path depicted in Figure A.5. In order to fulfill the a_{mws}, or the a_{mc} constraints, the vehicle should start braking in advance being more restrictive for the a_{mc} profile, see Figure A.7. The profit of being more restrictive is a smoother variation on the a_{mc} profile.

Additionally the VP outputs the instantaneous curvature $\kappa(t)$ of the trajectory being described by the vehicle

$$\kappa(t) = \frac{\sin(\varphi(t) + \phi(t))}{L\cos(\varphi(t))} = \frac{d\theta(t)}{ds} \tag{A.35}$$

where L is the distance between the rear and front axles, φ is the front steering angle, ϕ

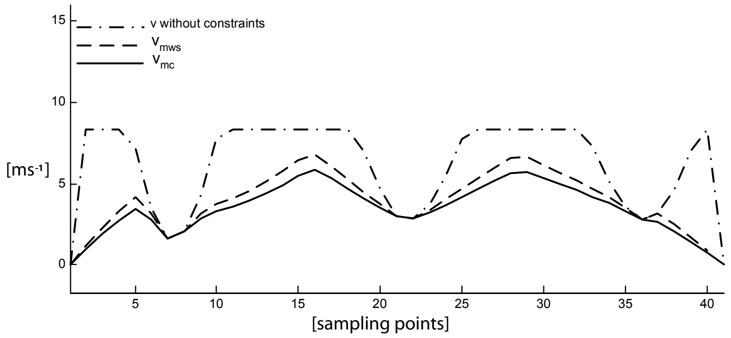


Figure A.6: Velocity profiles for the example of Figure A.5 (x-axis denotes the sampling points, and y-axis is in m/s)

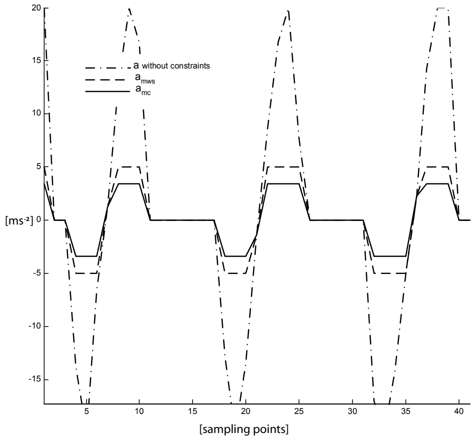


Figure A.7: Acceleration profiles for the example of Figure A.5 (x-axis denotes the sampling points, and y-axis is in m/s^2)

is the rear steering angle and θ is the vehicle orientation in the world coordinate system, as depicted in Figure 4.7.

Multi-Target Detection and Tracking (MTDT)

Although the author of this research contributed to the implementation of the MTDT System, the main contribution is from Institute of Systems and Robotics (ISR) researcher Abel Mendes. An overview of this module is presented here in order to fully understand the overall architecture. The block diagram in Figure A.8 describes the dataflow between the modules, that constitute the anti-collision system: Segmentation, Object Tracking, Obstacle Classification and Impact-time Computation.

- **Segmentation**: The goal here is to identify the limits of possible existing objects detected by the LIDAR (see Figure A.9) and if so, to filter and provide additional information about the object in analysis. The readings are subdivided into small sets of neighbour points (segments), taking into account the proximity between two consecutive points of the scan [118] [119]. A segment is, hence, a set of measurement values (points of the scan) close enough to each other, which due to their proximity, probably belong to the same object. The segmentation criterion is based on [120]. After the subdivision of each scan in segments, a selection of several points of the segment and the computation of the visible dimensions of the object take place.

- **Object Tracking**: It is necessary to identify the segment-object pair in an unquestionable fashion. For that purpose, the reference point of the detected segments is compared with the reference point estimated with a Kalman filter for each tracked object. For each detected segment, an interest region is defined, in which the search of existing objects is performed [121]. Since the LIDAR is carried by vehicles it is necessary to consider the movement of the LIDAR in the tracking of the objects. When performing the Kalman filtering update cycle all the variables involved (including actual and previous state) are expressed in the same reference frame [121].

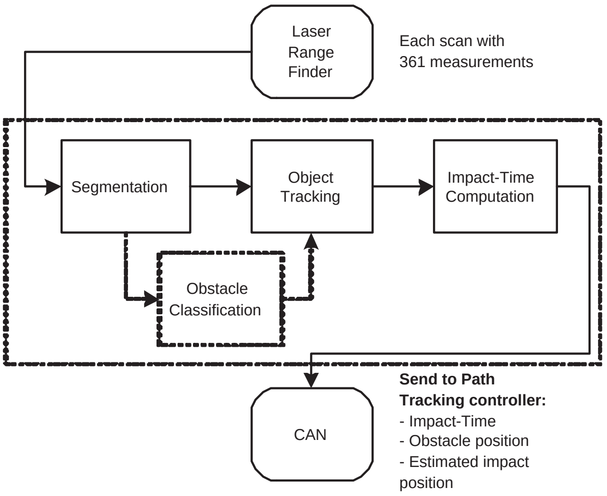


Figure A.8: Anti-collision system architecture.

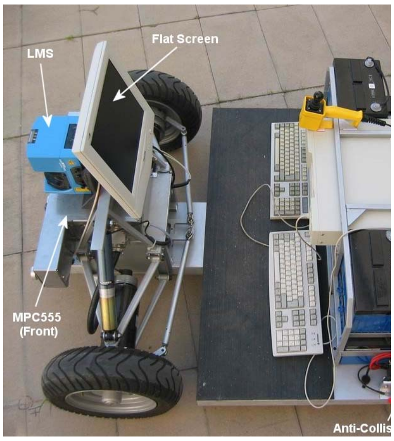


Figure A.9: Robucar prototype

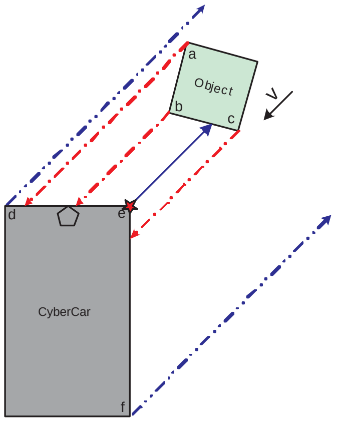


Figure A.10: Geometrical method of impact computation. (vector V represents the obstacle velocity relatively to the CyberCar; edge e will be the impact point)

- **Impact-Time Computation**: The Impact-Time Computation module uses the results of all the *Segmentation* and the *Object Tracking* to estimate the impact-time and position, for each one of all detected objects. The method used is based on the projection of all possible points of impact in the direction of the object's velocity, and for each instant it is assumed a constant object velocity relative to the vehicle. As we can see in Figure A.10, these points are the edges of the car (d, e, f) and of the object (a, b, c). So, from the projection lines starting on the object, defined by the starting points and velocity vector, we select the shortest line that intercepts a line segment of the boundary of the vehicle. Applying the same method for the projection lines that start on the car, we finish the process and achieve the colliding shortest distance. Note that the object velocity is assumed to be constant (V).

Appendix B

Chained Form Path Following Controller for front steered vehicles

Front steered vehicles Kinematic model

Consider the car-like robot moving on a plane, as shown in **Figure B.1**. For simplicity, we assume that the two wheels on each axis (front and rear) collapse into a single wheel located at the midpoint of the axis (bicycle model). The front wheel can be steered while the rear wheel orientation is fixed. Based on the center of rear wheel axle, the system is subject to four constraints, i.e. pure rolling and non-slipping constraints.

Non-slipping constraints

Rear virtual Wheel:

$$\dot{x}_R \sin\theta - \dot{y}_R \cos\theta = 0 \tag{B.1}$$

Front virtual Wheel:

$$\dot{x}_R \sin(\theta+\Psi) - \dot{y}_R \cos(\theta+\Psi) - L\dot{\theta}\cos\Psi = 0 \tag{B.2}$$

Pure rolling constraints

Rear virtual Wheel:

$$\dot{x}_R \cos\theta + \dot{y}_R \sin\theta = V_R \tag{B.3}$$

Front virtual Wheel:

$$\dot{x}_R \cos(\theta+\phi) + \dot{y}_R \sin(\theta+\phi) + L\dot{\theta}\sin\phi = V_F \tag{B.4}$$

Then the robot configuration is given by the generalized coordinate vector q.

$$q = [x_R, y_R, \theta, \Psi, V_R, V_F]^T \tag{B.5}$$

The constraint matrix is given as follows $A(q)\dot{q} = 0$, where $A(q)$ is obtained using **eq. B.1-B.5** and is given by:

$$A(q) = \begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 & 0 & 0 \\ \sin(\theta+\Psi) & -\cos(\theta+\Psi) & -L\cos\Psi & 0 & 0 & 0 \\ \cos\theta & \sin\theta & 0 & 0 & -1 & 0 \\ \cos(\theta+\Psi) & \sin(\theta+\Psi) & L\sin\Psi & 0 & 0 & -1 \end{bmatrix} \tag{B.6}$$

Choosing $\Omega(q)$ so as to satisfy the following condition $A(q)\Omega(q) = 0$, the model for rear-wheel drive can be derived:

$$\begin{cases} \dot{x}_R = u_1 \cos(\theta) \\ \dot{y}_R = u_1 \sin(\theta) \\ \dot{\theta} = u_1 \frac{\tan(\Psi)}{L} \\ \dot{\Psi} = u_2 \end{cases} \tag{B.7}$$

Where $u_1 = V_R$ is the driving velocity of the rear wheel, and u_2 is the steering velocity input of the front wheel. There is a control singularity at $\Psi = \pm\pi/2$, where the vector w_1 is discontinuous. This corresponds to the rear wheel drive car becoming jammed when

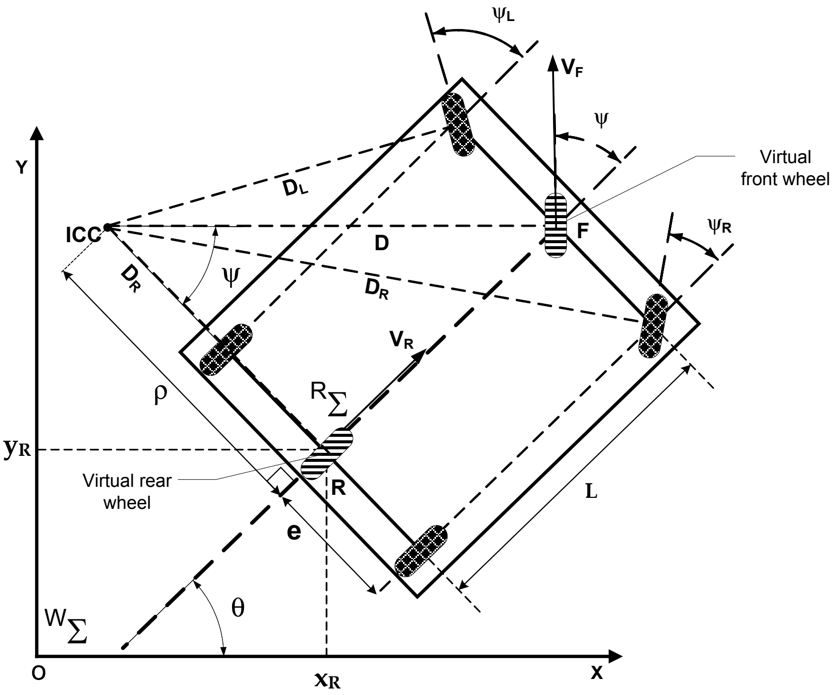


Figure B.1: Front steered Vehicle geometrical configuration. ${}^{W}\Sigma$ and ${}^{R}\Sigma$ represent respectively the world coordinate system and the vehicle local coordinate system with its origin at the midpoint of the rear axle and its x-axis aligned with the longitudinal axis of the vehicle

the front wheel is normal to the longitudinal axis L of the car body, the relevance of this singularity is limited, due to the restricted range of the steering angle Ψ in most of the vehicles.

A different way of obtaining the kinematic equations is by using geometrical analysis, as follows. Since there exists a linear velocity vector and instantaneous rotation center at the reference frame located at the midpoint of the rear axle ${}^{R}\Sigma$ due to to pure rolling, non-slipping and rigid body assumptions, one can also derive the kinematics equation has follows. The instantaneous curvature center when driving in front steering mode can be written:

$$\dot{\theta} = \frac{V_R}{D_R} = \frac{V_F}{D_F} \tag{B.8}$$

where

$$D_R = \frac{L}{|\tan(\Psi)|} = D_F \cdot \cos(\Psi) \tag{B.9}$$

the combination of (B.8) and (B.9) gives

$$V_R = V_R \cdot \cos(\Psi) \tag{B.10}$$

therefore the equations of movement for the rear axle and front axle are given by equations (B.11) and (B.12) respectively.

$$\begin{cases} \dot{x_R} = V_R \cdot \cos(\theta) \\ \dot{y_R} = V_R \cdot \sin(\theta) \\ \dot{\theta} = V_R \cdot \frac{\tan(\Psi)}{L} \end{cases} \tag{B.11}$$

$$\begin{cases} \dot{x_F} = V_F \cdot \cos(\theta + \Psi) \\ \dot{y_F} = V_F \cdot \sin(\theta + \Psi) \\ \dot{\theta} = V_F \cdot \frac{\sin(\Psi)}{L} \end{cases} \tag{B.12}$$

one can also express the equations of movement of the rear axle function of the front instantaneous velocity V_F and the front axle function of the rear instantaneous velocity V_R; equations (B.13) and (B.14) respectively.

$$\begin{cases} \dot{x_R} = V_F \cdot \cos(\Psi) \cdot \cos(\theta) \\ \dot{y_R} = V_R \cdot \cos(\Psi) \cdot \sin(\theta) \\ \dot{\theta} = V_F \cdot \frac{\tan(\Psi)}{L} \end{cases} \tag{B.13}$$

$$\begin{cases} \dot{x_F} = V_R \cdot \frac{\cos(\theta+\Psi)}{\cos(\Psi)} \\ \dot{y_F} = V_R \cdot \frac{\sin(\theta+\Psi)}{\cos(\Psi)} \\ \dot{\theta} = V_R \cdot \frac{\tan(\Psi)}{L} \end{cases} \tag{B.14}$$

Using equation (B.11) the kinematic model of the vehicle, for a reference frame located at the midpoint of the rear axle ${}^{R}\Sigma$, is given in the matrix form by the following equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \frac{\tan\Psi}{L} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{B.15}$$

Where $v_1 = V_R$ represents the linear velocity of the vehicle, v_2 is the angular velocity of the steering wheels, L is the distance between the rear and front axles, φ is the front steering angle and θ is the vehicle orientation in the world coordinate system, as depicted in **Figure B.1**. As expected equations **B.15** and **B.7** are equivalent.

A different point of view and more useful in terms of path following is the one that describes the vehicle behaviour in terms of the the path coordinates [122]. Assuming that has vehicle has to follow a path defined by its arc length, one can define the following variables, d_e^{tg} is the perpendicular distance between the rear axle midpoint and the current tangent to the path and s is the corresponding value of the path parameter. θ_e^{tg} is the angle between the x axis of the vehicle and the current tangent to the path, i.e. $\theta_e^{tg} = \theta_R - \theta_P$, where θ_R is the vehicle heading angle and the θ_P is the path tangent angle.

Therefore one can derive the following equations:

$$\dot{s} = v_1 \cdot cos(\theta_e^{tg}) + \dot{\theta}_P \cdot d_e^{tg} \tag{B.16}$$

$$\dot{d_e^{tg}} = v_1 \cdot sin(\theta_e^{tg}) \tag{B.17}$$

$$c(s) = \frac{\partial \theta_P}{\partial s} \Rightarrow \dot{\theta}_P = c(s) \cdot \dot{s} \tag{B.18}$$

Where $c(s)$ is the curvature of the path.

Replacing equation **B.18** in **B.16** one obtain $\dot{s}$.

$$\begin{array}{c} \dot{s} - \dot{\theta}_P \cdot d_e^{tg} = v_1 \cdot cos(\theta_e^{tg}) \\ \Leftrightarrow \\ \dot{s} - \dot{s} \cdot c(s) \cdot d_e^{tg} = v_1 \cdot cos(\theta_e^{tg}) \\ \Leftrightarrow \\ \dot{s} = \frac{v_1 \cdot cos(\theta_e^{tg})}{1 - c(s) \cdot d_e^{tg}} \end{array} \tag{B.19}$$

for $\dot{\theta}_P$ the following expression is obtained

$$\begin{array}{c} \dot{s} - \dot{\theta}_P \cdot d_e^{tg} = v_1 \cdot cos(\theta_e^{tg}) \\ \Leftrightarrow \\ \frac{\dot{\theta}_P}{c(s)} - \dot{\theta}_P \cdot d_e^{tg} = v_1 \cdot cos(\theta_e^{tg}) \\ \Leftrightarrow \\ \dot{\theta}_P \left(\frac{1}{c(s)} - d_e^{tg} \right) = v_1 \cdot cos(\theta_e^{tg}) \\ \Leftrightarrow \\ \dot{\theta}_P = \frac{v_1 \cdot cos(\theta_e^{tg}) \cdot c(s)}{1 - d_e^{tg} \cdot c(s)} \end{array} \tag{B.20}$$

From equations **B.7**, **B.16** and **B.20**

$$\begin{cases} \theta_e^{tg} = \theta_R - \theta_P \\ \dot{\theta}_P = \frac{v_1 \cdot cos(\theta_e^{tg}) \cdot c(s)}{1 - d_e^{tg} \cdot c(s)} \\ \dot{\theta}_R = v_1 \cdot \frac{\tan(\Psi)}{L} \end{cases} \Rightarrow \dot{\theta}_e^{tg} = v_1 \cdot \left(\frac{\tan(\Psi)}{L} - \frac{cos(\theta_e^{tg}) \cdot c(s)}{1 - d_e^{tg} \cdot c(s)} \right) \tag{B.21}$$

Using the previous transformations one can write in the matrix form the kinematic model in terms of the path coordinates:

$$\begin{bmatrix} \dot{s} \\ \dot{d}_e^{tg} \\ \dot{\theta}_e^{tg} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta_e^{tg})}{1 - c(s) d_e^{tg}} \\ \sin(\theta_e^{tg}) \\ \frac{\tan \Psi}{L} - \frac{c(s) \cos(\theta_e^{tg})}{1 - d_e^{tg} c(s)} \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{B.22}$$

Odometry model

Let the vehicle position be represented by the middle point rear axle with Cartesian coordinates (x_k, y_k) at time t_k. The vehicle local coordinate system is defined as having origin M and its x-axis aligned with the longitudinal axis of the car. θ_k is the vehicle heading angle at time t_k. Assuming that the vehicle's motion is locally circular, its position and orientation at time t_k is given by

$$\begin{cases} x_{k+1} = x_k + \Delta_R \cos(\theta_k + \omega/2) \\ y_{k+1} = y_k + \Delta_R \sin(\theta_k + \omega/2) \\ \theta_{k+1} = \theta_k + \omega_R \end{cases} \tag{B.23}$$

where Δ_R is the arc length and ω_R the elementary rotation of the rear axle. Assuming that there is no wheel slippage and using only data from the rear wheels encoders, then

$$\Delta_R = \frac{\Delta_{RR} + \Delta_{RL}}{2} \quad , \; \omega_R = \frac{\Delta_{RR} - \Delta_{RL}}{2e} \tag{B.24}$$

where e is the half distance between wheels, and Δ_{RR} and Δ_{RL} are calculated using the right and left wheel encoders measurements, respectively.

Chained form background

Consider a driftless two-input control system of the form

$$\begin{array}{rcl} \dot{q}_1 & = & u_1 \\ \dot{q}_2 & = & u_2 \\ \dot{q}_3 & = & q_2u_1 \\ \dot{q}_4 & = & q_3u_1 \\ & \vdots & \\ & \dot{q}_n = q_{n-1^{u}1} & \end{array} \quad \text{or} \quad \dot{q} = g_1(q)u_{2,}$$

with

$$g_1 = \begin{array}{c} 1 \\ 0 \\ q_2 \\ q_3 \\ \vdots \\ q_{n-1} \end{array}, \quad g_2 = \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array}$$

System (7.45) is a special case of chained form called one-chain system. If we set $u_1 = 1$, the system becomes linear and behaves like a chain of integrators from $q_n$to q_n, driven by the input u_2. Note that the two-input case is sufficiently broad to cover most of the kinematic models of practical wheeled mobile robots. For the case $m \geq 3$ and for the corresponding multi-chain forms, the reader is referred to [54], where these canonical forms were originally introduced.

The structure of system (7.45) is particularly interesting, in that controllability is achieved via $g_{1,}g_2$ and (repeated) Lie brackets of the form $ad^k_{g1}g_2$, where

$$ad_{g1}g_2 = ad^1_{g1}g_2 = [g_1, g_2], \quad ad^k_{g1}g_2 = \left[g_{1,}ad^{k-1}_{g1}g_2\right].$$

In fact, a simple computation shows that

$$ad^k_{g1}g_2 = \begin{bmatrix} o \\ \vdots \\ (-1)^k \\ \vdots \\ o \end{bmatrix}, \quad 1 \leq k \leq n-2,$$

where the nonzero element is the $(k+2) - nd$ entry. As a consequence, the n vector fields

$$\left\{g_1, g_{2,}ad^1_{g1,}g_{2,\dots,}ad^{n-2}_{g1}g_2\right\}$$

are linearly independent everywhere, and the controllability rank condition is satisfied.

Equivalently, we may say that system (7.45)is completely nonholonomic, with degree of nonholonomy $k = n-1$, growth vector $r = (2, 3, 4, ..., n)$, and relative growth vector $\sigma = (2, 1, 1, ..., 1)$. In connection with the foregoing calculation, $u_1$is called the generating input, while q_1 and $q_2$are often referred to as base variables. Note also that, for $k > n-2$,the repeated Lie brackets are identically zero ; this property of the system is called nilpotency.

It is natural to ask how general the chained form (7.45) is. In particular, we are interested in conditions for converting the driftless control system (7.44) into chained form by means of the invertible input transformation $v = \beta(q)u$ and the change of coordinates $z = \phi(q)$. If these conditions hold, then we can design the controller for the chained form and apply a precompensator to the system, that performs the input and state transformations.

Recently, Murray[55] established a set of necessary and sufficient conditions for the conversion of a two-input system into chained form. Let $\triangle$= *span* $\{g_1, g_2\}$ and define the two filtrations:

$$\begin{array}{cc} E_1 = \Delta & F_1 = \Delta \\ E_2 = E_1 + [E_1, E_1] & F_2 = F_1 + [F_1, F_1] \\ \vdots & \vdots \\ E_i + 1 = E_i + [E_1, E_1], & F_i + 1 = F_i + [F_i, F_i]. \end{array}$$

Then, system(7.44), with $m = 2$,can be cast in chained form if and only if

$\dim E_i = \dim F_i = i + 1, \quad i = 1, ..., n - 1.$

By applying this condition, one can show that completely nonholonomic systems with two inputs and relative growth vector $(2, 1)$ (obtained for $n = 3$) or $(2, 1, 1)$ (obtained for $n = 4$) can always be put in chained form.

The proof of this result relies on the theory of exterior differential systems , and in particular on the Goursat normal form, which is the dual of the chained form . The constructive procedure given in the proof has general validity, and has been used for example to find local transformations which convert the N-trailer system into chained form (*see* [41, 56]).

However, there is a simpler constructive algorithm based on the following sufficient condition for local conversion of a two-input system to chained form. Define the distributions

$\Delta_0 = span\left\{g_1, g_2, ad_{g_1} g_2, \cdots, ad_{g1}^{n-2} g_2\right\}$

$\Delta_1 = span\left\{g_2, ad_{g_1}, g_2, \ldots, ad_{g_1}^{n-2} g_2\right\}$

$\Delta_2 = span\left\{g_2, ad_{g_1} g_2, \ldots, , ad_{g_1}^{n-3} g_2\right\}$

If for some open set $U, \dim \Delta_0 = n$ and $\Delta_{1,} \Delta_2$ are involutive, and there exists a smooth function $h_1 : U \to IR^n$such that

$dh_1 . \Delta_1 = 0 \qquad \text{and} \qquad dh_1 . g_1 = 1,$

then there exists a local feedback transformation and change of coordinates that transform the system into chained form.

In particular, the change of coordinates $z = \phi(q)$ is given by

$z_1 = h_1$

$z_2 = L_{g1}^{n-2} h_2$

$\vdots$

$z_{n-1} = L_{g1} h_2$

$z_n = h_{2,}$

with $h_2$independent from $h_1$and such that

$dh_2 . \Delta_2 = L_{g1} h_2 = 0.$

The existence of independent h_1 and $h_2$with the above properties is guaranteed by Frobenius theorem, since Δ_1 and Δ_2 are involutive. Using the invertible input transformation

$v_1 = u_1$

$v_2 = (L_{g1}^{n-1} h_2) u_1 + (L_{g2} L_{g1}^{n-2} h_2) u_2$

results in the transformed system

$$\begin{aligned}
\dot{z}_1 &= v_1 \\
\dot{z}_2 &= v_2 \\
\dot{z}_3 &= z_2 v_1 \\
\dot{z}_4 &= z_3 v_1 \\
&\vdots \\
\dot{z}_n &= z_{n-1} v_1 .
\end{aligned}$$

To apply this procedure, we must solve two sets of partial differential equations, namely, eqs.(7.46) – (7.47) If g_1 and g_2 have the special form

$$g_1 = \begin{bmatrix} 1 \\ g_{12}(q) \\ \vdots \\ g_{1n}(q) \end{bmatrix}, \, g_2 = \begin{bmatrix} 0 \\ g_{22}(q) \\ \vdots \\ g_{2n}(q) \end{bmatrix}$$

with arbitrary g_{ij}'s , then it is easy to verify that Δ_1 is always involutive and we can choose $h_1 = q_1$. In this case, we only have to verify that Δ_2 is involutive, and solve the associated partial differential equation (7.47) . To this end, one may in general use the constructive procedure given in the proof of Frobenius theorem (*see* [4, *p*.26]). Note that it is always possible to cast g_1 and g_2 in above special form, by reordering variables and by virtue of the independence assumption on the input vector fields .

Chained form based lateral controller

The control law designed here, based upon the kinematics model uses the chained systems theory [123]. Although mobile robot models cannot be linearized, it has been proved that one can convert the nonlinear system in an almost linear system, termed as chained form.

Scaling the input v_1 so that it enters directly into $\dot{s}$. Denoting by $\tilde{v}_1$ the new input and setting and $\tilde{v}_2 = v_2$ the kinematic equations model of the front-steered car-like vehicle, in terms of the the path coordinates given by **eq. B.22**, are written in the following form:

$$\begin{bmatrix} \dot{s} \\ \dot{d}_e^{tg} \\ \dot{\theta}_e^{tg} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1-c(s)d_e^{tg}}{\cos(\theta_e^{tg})} \cdot \sin(\theta_e^{tg}) \\ \frac{1-c(s)d_e^{tg}}{\cos(\theta_e^{tg})} \cdot \left(\frac{\tan\Psi}{L} - \frac{c(s)\cos(\theta_e^{tg})}{1-d_e^{tg}c(s)} \right) \\ 0 \end{bmatrix} \tilde{v}_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tilde{v}_2 = \begin{bmatrix} 1 \\ \left(1 - c(s)d_e^{tg}\right)\tan\left(\theta_e^{tg}\right) \\ \frac{\left(1-c(s)d_e^{tg}\right)\tan(\Psi)}{L\cos\left(\theta_e^{tg}\right)} - c(s) \\ 0 \end{bmatrix} \tilde{v}_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tilde{v}_2 \tag{B.25}$$

then g_1 g_2 are given by:

$$g_1 = \begin{bmatrix} 1 \\ (1 - c(s)d_e^{tg})\tan(\theta_e^{tg}) \\ \frac{\left(1-c(s)d_e^{tg}\right)\tan(\Psi)}{L\cos\left(\theta_e^{tg}\right)} - c(s) \\ 0 \end{bmatrix}; \; g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{B.26}$$

The vectors g_1 and g_2 have the special structure described above, then $h_1 = s$ and $h_2 = d_e^{tg}$. The resulting change of coordinates is as following:

$$\begin{array}{l} z_1 = h_1 \\ z_2 = \mathcal{L}_{g_1}^2 h_2 \\ z_3 = \mathcal{L}_{g1} h_2 \\ z_4 = h_2 \end{array} \tag{B.27}$$

where $\mathcal{L}$ denotes the Lie derivative operator.

From **eq. B.26** and **eq. B.27** one can obtain z_3:

$$\begin{aligned}
z_3 \quad &= \mathcal{L}_{g_1} h_2 = \textstyle\sum_{i=1}^{n} \frac{\partial h_2}{\partial q} \cdot g_{1_i}(q) \ \wedge \ n = 4 \\
&= \tfrac{\partial h_2}{\partial s} \cdot g_{1_1}(q) + \tfrac{\partial h_2}{\partial d_e^{tg}} \cdot g_{1_2}(q) + \tfrac{\partial h_2}{\partial \theta_e^{tg}} \cdot g_{1_3}(q) + \tfrac{\partial h_2}{\partial \varphi} \cdot g_{1_4}(q) \\
&= \tfrac{\partial d_e^{tg}}{\partial s} \cdot 1 + \tfrac{\partial d_e^{tg}}{\partial d_e^{tg}} \cdot (1 - c(s) d_e^{tg}) \tan(\theta_e^{tg}) \\
&+ \tfrac{\partial d_e^{tg}}{\partial \theta_e^{tg}} \cdot \left(\tfrac{\left(1 - c(s) d_e^{tg}\right) \tan(\Psi)}{L \cos\left(\theta_e^{tg}\right)} - c(s) \right) + \tfrac{\partial d_e^{tg}}{\partial \varphi} \cdot 0 \\
&= (1 - c(s) d_e^{tg}) \tan(\theta_e^{tg})
\end{aligned} \tag{B.28}$$

From **eq. B.26**, **eq. B.27** and **eq. B.28** one can obtain z_2:

$$\begin{aligned}
z_2 \quad &= \mathcal{L}_{g_1}^2 h_2 = \frac{\partial\left(\mathcal{L}_{g_1} h_2\right)}{\partial q} \cdot g_1(q) \\
&= \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial q} \cdot g_1(q) \\
&= \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial s} \cdot g_{1_1}(q) + \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial d_e^{tg}} \cdot g_{1_2}(q) \\
&+ \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial \theta_e^{tg}} \cdot g_{1_3}(q) + \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial \varphi} \cdot g_{1_4}(q) \\
&= \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial s} \cdot 1 + \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial d_e^{tg}} \cdot (1 - c(s) d_e^{tg}) \tan(\theta_e^{tg}) \\
&+ \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan(\Psi)\right)}{\partial \theta_e^{tg}} \cdot \left(\frac{\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)}{L \cos\left(\theta_e^{tg}\right)} - c(s) \right) + \frac{\partial\left(\left(1 - c(s) d_e^{tg}\right) \tan\left(\theta_e^{tg}\right)\right)}{\partial \varphi} \cdot 0 \\
&= -d_e^{tg} \cdot c'(s) \tan(\theta_e^{tg}) + \frac{c(s) \sin\left(\theta_e^{tg}\right)^2 \left(d_e^{tg} \cdot c(s) - 1\right) + c(s)\left(d_e^{tg} \cdot c(s) - 1\right)}{\cos\left(\theta_e^{tg}\right)^2} + \frac{\tan(\Psi)\left(d_e^{tg} \cdot c(s) - 1\right)^2}{L \cos\left(\theta_e^{tg}\right)^3}
\end{aligned} \tag{B.29}$$

Using the invertible input transformation

$$\begin{array}{l} u_1 = \tilde{v}_1 \\ u_2 = (\mathcal{L}_{g1}^3 h_2) \tilde{v}_1 + (\mathcal{L}_{g2} \mathcal{L}_{g1}^2 h_2) \tilde{v}_2 \end{array} \tag{B.30}$$

where $\tilde{v}_1$ is:

$$\tilde{v}_1 = \tfrac{1 - c(s) d_e^{tg}}{\cos(\theta_e^{tg})} v_1 \tag{B.31}$$

$\mathcal{L}_{g1}^{3}h_2$ is:

$$
\begin{aligned}
\mathcal{L}_{g1}^{3}h_2 \quad & = \frac{\partial\left(\mathcal{L}_{g1}^{2}h_2\right)}{\partial q}\cdot g_1(q) \\
= \quad & \frac{\partial}{\partial s}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 1 \\
+ \quad & \frac{\partial}{\partial d_e^{tg}}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right) \\
\cdot \quad & \left(1-c(s)d_e^{tg}\right)\tan\left(\theta_e^{tg}\right) \\
+ \quad & \frac{\partial}{\partial \theta_e^{tg}}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right) \\
\cdot \quad & \frac{\left(1-c(s)d_e^{tg}\right)\tan\left(\Psi\right)}{L\cos\left(\theta_e^{tg}\right)}-c(s) \\
+ \quad & \frac{\partial}{\partial \Psi}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 0 \\
= \quad & -\tan\left(\theta_e^{tg}\right)\left(d_e^{tg}\mathrm{c}''(s)+2\left(c(s)\right)^2\left(d_e^{tg}c(s)-1\right)\right) \\
+ \quad & \frac{\mathrm{c}'(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(3d_e^{tg}c(s)-2\right)-\frac{2c(s)\sin\left(\theta_e^{tg}\right)\tan\left(\phi\right)\left(d_e^{tg}c(s)-1\right)^2}{L}+\mathrm{c}'(s)\left(3d_e^{tg}c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2} \\
- \quad & \frac{3(c(s))^2\left(\sin\left(\theta_e^{tg}\right)\right)^3\left(d_e^{tg}c(s)-1\right)+3(c(s))^2\sin\left(\theta_e^{tg}\right)\left(d_e^{tg}c(s)-1\right)+\frac{3d_e^{tg}\mathrm{c}'(s)\tan\left(\phi\right)\left(1-d_e^{tg}c(s)\right)}{L}}{\left(\cos\left(\theta_e^{tg}\right)\right)^3} \\
- \quad & \frac{\frac{2c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^3\tan\left(\phi\right)\left(d_e^{tg}c(s)-1\right)^2}{L}+\frac{7c(s)\sin\left(\theta_e^{tg}\right)\tan\left(\phi\right)\left(d_e^{tg}c(s)-1\right)^2}{L}}{\left(\cos\left(\theta_e^{tg}\right)\right)^4}+\frac{3\sin\left(\theta_e^{tg}\right)(\tan\left(\phi\right))^2\left(1-d_e^{tg}c(s)\right)^3}{L^2\left(\cos\left(\theta_e^{tg}\right)\right)^5}
\end{aligned}
\tag{B.32}
$$

and $\mathcal{L}_{g2}\mathcal{L}_{g1}^{2}h_2$ is:

$$
\begin{aligned}
& \mathcal{L}_{g2}\mathcal{L}_{g1}^{2}h_2 = \mathcal{L}_{g2} \\
\cdot \quad & \left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right) \\
= \quad & \frac{\partial}{\partial s}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 0 \\
+ \quad & \frac{\partial}{\partial d_e^{tg}}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 0 \\
+ \quad & \frac{\partial}{\partial \theta_e^{tg}}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 0 \\
+ \quad & \frac{\partial}{\partial \Psi}\left(-d_e^{tg}\cdot \mathrm{c}'(s)\tan\left(\theta_e^{tg}\right)+\frac{c(s)\left(\sin\left(\theta_e^{tg}\right)\right)^2\left(d_e^{tg}\cdot c(s)-1\right)+c(s)\left(d_e^{tg}\cdot c(s)-1\right)}{\left(\cos\left(\theta_e^{tg}\right)\right)^2}+\frac{\tan\left(\Psi\right)\left(d_e^{tg}c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3}\right)\cdot 1 \\
= \quad & \frac{\left(d_e^{tg}\cdot c(s)-1\right)^2}{L\left(\cos\left(\theta_e^{tg}\right)\right)^3\left(\cos\left(\Psi\right)\right)^2}
\end{aligned}
\tag{B.33}
$$

The transformed system becomes two inputs and four states chain form system $(2, 4)$

and has the following structure:

$$\begin{aligned}\dot{z}_1 &= u_1\\ \dot{z}_2 &= u_2\\ \dot{z}_3 &= x_2 \cdot u_1\\ \dot{z}_4 &= x_3 \cdot u_1\end{aligned} \tag{B.34}$$

the above change of coordinates (and thus,the obtained chained form) is only locally defined.

Although the system has two inputs, u_1 and u_2, this model can be considered single input if u_1 is known *a priori.* Then the objective of the control law is to achieve path following under the assumption that the vehicle linear velocity u_1 is constant.

The controller was made using the smooth time-varying feedback stabilization method described in [122], where control is either smooth or at least continuous with respect to the robot state.

As a first step, the variables of the chained form are redefined

$$\chi = (\chi_1, \chi_2, \chi_3, \chi_4) = (x_1, x_4, x_3, x_2) \tag{B.35}$$

resulting the chained form system

$$\begin{aligned}\dot{\chi_1} &= u_1\\ \dot{\chi_2} &= \chi_3 u_1\\ \dot{\chi_3} &= \chi_4 u_1\\ \dot{\chi_4} &= u_2\end{aligned} \tag{B.36}$$

The above reordering is simply an exchange between the second and fourth coordinates. Path following is achieved via input scaling, which requires zeroing the χ_2, χ_3 and χ_4 variables, independently from χ_1. The system (B.30) is controllable if u_1 is a piecewise continuous, bounded, and strictly positive (or negative) function, as stated in [123]. Therefore u_2 is the only input to the system as long has u_1 is known *a priori*:

$$u_2(\chi_2, \chi_3, \chi_4, t) = -k_1|u_1(t)|\chi_2 - k_2 u_1(t)\chi_3 - k_3|u_1(t)|\chi_4 \tag{B.37}$$

The complete deducing of the controller and its background theory are described in [123].