

UNIVERSITÉ DE SHERBROOKE  
Faculté de génie  
Département de génie électrique et de génie informatique

**SIMULATEUR TEMPS-RÉEL DE VÉHICULES  
INTELLIGENTS ET AUTONOMES EN CONDUITE  
COLLABORATIVE**

Thèse de doctorat  
Spécialité : génie électrique et informatique

par

Farid Bounini

Sherbrooke (Québec) Canada

23 avril 2018

## **Membres du jury**

Denis GINGRAS

Professeur à l'université de Sherbrooke et directeur du laboratoire sur l'intelligence  
véhiculaire (LIV) de la faculté de génie, Université de Sherbrooke  
Directeur

Hervé POLLART

Directeur du développement des affaires de l'équipe de véhicules intelligents chez OPAL-RT  
Technologies inc.  
Évaluateur

Dominique GRUYER

Directeur de recherche, responsable du laboratoire LIVIC, département COSYS, IFSTTAR.  
Évaluateur

Alexandre GIRARD

Professeur adjoint à l'Université de Sherbrooke  
Rapporteur

# Sommaire

Durant ces dernières décennies, l'apparition des systèmes d'aide à la conduite a essentiellement été favorisée par le développement des différentes technologies ainsi que par celui des outils mathématiques associés. Cela a profondément affecté les systèmes de transport et a donné naissance au domaine des systèmes de transport intelligents (STI). Nous assistons de nos jours au développement du marché des véhicules intelligents dotés de systèmes d'aide à la conduite et de moyens de communication inter-véhiculaire. Les véhicules et les infrastructures intelligents changeront le mode de conduite sur les routes. Ils pourront résoudre une grande partie des problèmes engendrés par le trafic routier comme les accidents, les embouteillages, la pollution, etc.

Cependant, le bon fonctionnement et la fiabilité des nouvelles générations des systèmes de transport nécessitent une parfaite maîtrise des différents processus de leur conception, en particulier en ce qui concerne les systèmes embarqués. Il est clair que l'identification et la correction des défauts des systèmes embarqués sont deux tâches primordiales à la fois pour la sauvegarde de la vie humaine, à la fois pour la préservation de l'intégrité des véhicules et des infrastructures urbaines. Pour ce faire, la simulation numérique en temps réel est la démarche la plus adéquate pour tester et valider les systèmes de conduite et les véhicules intelligents. Elle présente de nombreux avantages qui la rendent incontournable pour la conception des systèmes embarqués.

Par conséquent, dans ce projet, nous présentons une nouvelle plateforme de simulation temps-réel des véhicules intelligents et autonomes en conduite collaborative. Le projet se base sur deux principaux composants. Le premier étant les produits d'OPAL-RT Technologies notamment le logiciel RT-LAB « *en : Real Time LABoratory* », l'application Orchestra et les machines de simulation dédiées à la simulation en temps réel et aux calculs parallèles, le second composant est Pro-SiVIC pour la simulation de la dynamique des véhicules, du comportement des capteurs embarqués et de l'infrastructure. Cette nouvelle plateforme (Pro-SiVIC/RT-LAB) permettra notamment de tester les systèmes embarqués (capteurs, actionneurs, algorithmes), ainsi que les moyens de communication inter-véhiculaire. Elle permettra aussi d'identifier et de corriger les problèmes et les erreurs logicielles, et enfin de valider les systèmes embarqués avant même le prototypage.

**Mots-clés :** Systèmes d'aide à la conduite ; conduite automatisée ; véhicules autonomes, intelligents et connectés ; systèmes de transports intelligents ; systèmes embarqués ; calcul parallèle ; simulation temps-réel ; architectures de fusion de données multi-capteurs : centralisées, hiérarchiques et distribuées.

# Abstract

During the last decades, Advanced Driver Assistance Systems (ADAS) have been mainly driven by the development of embedded technologies as well as by the associated mathematical tools. Thus, ADAS deeply affected transportation systems and gave birth to the Intelligent Transportation Systems (ITS). Today, we are witnessing the development of the intelligent vehicles market that make use of ADAS and inter-vehicular communications. Intelligent vehicles and infrastructure will change drivers' habits on the roads; they will be able to solve many of the issues that are caused by road traffic such as accidents, traffic jams, pollution, etc.

However, the proper functioning and reliability of the new generations of transport systems require a perfect mastery of the various processes of their design, especially with regard to embedded systems. It is clear that the identification and correction of defects in embedded systems are essential for safeguarding human life, and the integrity of the vehicle, and the infrastructure. Thus, real-time simulation is the most appropriate approach for testing and validating ADAS and intelligent vehicles. It has many advantages that make it an essential framework for designing embedded systems.

To that end, we present in this project a new platform for real-time simulation of collaborative and autonomous vehicles. The project is based on two main components: the first being the OPAL-RT Technologies' platforms, including RT-LAB "Real Time LABoratory", Orchestra and the OPAL-RT simulators dedicated to real-time simulation and parallel computing; the second component is the Pro-SiVIC platform for real-time simulation of vehicle dynamics, embedded sensor, and infrastructure. This new platform (Pro-SiVIC / RT-LAB) will test and validate embedded systems (sensors, actuators, algorithms), as well as inter-vehicular communication tools. It will also identify and fix problems with the embedded systems and help to debug embedded software and validate embedded systems (ADAS), even before prototyping.

**Keywords:** Advanced driver assistance systems (ADAS) ; automated driving ; autonomous, intelligent and connected vehicles ; intelligent transport systems ; embedded systems ; parallel computing ; real-time simulation ; multi-sensors data fusion architectures: centralized, hierarchical and distributed.

# Remerciements

Je tiens à remercier particulièrement mon superviseur, le professeur Denis Gingras, pour ses précieux conseils et sa compréhension, et pour avoir orienté et encouragé mes travaux de recherche.

Je voudrais également remercier les membres de mon comité, le directeur du développement des affaires de l'équipe de véhicules intelligents Hervé Pollart, le professeur Dominique Gruyer et le professeur Alexandre Girard pour leurs précieux commentaires et suggestions.

Je tiens également à remercier tous les membres du laboratoire de l'intelligence véhiculaire (LIV) de la faculté de génie de l'université de Sherbrooke, qui sont de bons collègues et de bon(ne)s ami(e)s.

Ce travail fait partie de COOPERCOM, un projet de recherche international de trois ans (Canada-France). Je tiens à remercier nos deux partenaires industriels ESI-CIVITEC en France et OPAL-RT Technologies au "Canada, le programme de financement « MITACs Accelerate » du Canada, le Conseil national de recherches en sciences et en génie (CRSNG) du Canada et l'agence nationale de la recherche (ANR) en France pour avoir soutenu mon projet de recherche.

Je voudrais aussi remercier mes ami(e)s et ma famille pour leur soutien. Surtout, je voudrais remercier mes parents pour leur soutien et leur amour non conditionnels et indéfectibles.

*“ Taqvaylit/Tamaziyt : Ad snimrey imddukkal, timeddukkal, iselmaden, tiselmadin d twaculin iw sumata i wafud d tevyest i rran u snarenuyen deg-i. Tanemmirt u tajmilt timeqqrarin i Yemma d Azizu yef lelmala, lehnana, ttrevga, tigzi d usevyes deg iyi-d-sekkrent u deg iyi-d-sneggant.”*

Je voudrais également remercier les nombreuses personnes qui ont contribué à l'élaboration de ce travail.

# Table des matières

Sommaire.....	iii
Abstract .....	iv
Remerciements .....	v
Table des matières .....	vi
Liste des abréviations .....	xii
Liste des symboles .....	xv
Liste des tableaux .....	xvii
Liste des figures .....	xviii
Chapitre 1 Introduction générale .....	21
1.1    Niveaux de la conduite automatisée .....	22
1.2    Problématique .....	23
1.3    Objectif principal .....	24
1.4    Objectifs intermédiaires .....	24
1.4.1    Interfaçage des deux plateformes de simulation Pro-SiVIC et RT-LAB .....	24
1.4.2    Développement et simulation des systèmes d'aide à la conduite .....	26
1.5    Contribution et originalité du projet de recherche .....	26
1.6    Structure de la thèse .....	29
Chapitre 2 État de l'art .....	32
2.1    ADAS équipant un véhicule intelligent .....	32
2.1.1    ADAS dédiés au contrôle du véhicule.....	32
2.1.2    Synthèse de la combinaison de plusieurs ADAS .....	34
2.1.3    Communication véhiculaire.....	35
2.1.4    Systèmes d'information géographique et données satellitaires .....	36
2.2    État de l'art sur véhicules intelligents .....	38
2.2.1    Navette Arma de NAVYA .....	38
2.2.2    Véhicules de Volvo .....	38

2.2.3	Uber et Lyft .....	39
2.2.4	Pilote automatique de Tesla.....	39
2.2.5	Voiture de Waymo (Google) .....	39
2.3	Plateformes de simulation.....	40
2.3.1	Simulateurs de la dynamique des véhicules et des capteurs embarqués .....	43
2.3.2	Simulateurs de réseau et de mobilité .....	46
2.3.3	Plateformes temps réel et calcul parallèle .....	48
2.4	Comparaison des plateformes .....	50
2.5	Conclusion .....	52
	 Chapitre 3 Capteurs embarqués .....	54
3.1	Introduction.....	54
3.2	Type de capteurs embarqués .....	55
3.2.1	Capteurs proprioceptifs.....	55
3.2.2	Capteurs extéroceptifs .....	58
3.3	Capteurs embarqués et zones à couvrir autour d'un véhicule.....	60
3.4	Conclusion .....	62
	 Chapitre 4 Fusion de données des capteurs embarqués .....	64
4.1	Introduction.....	64
4.2	Observateur d'état des systèmes dynamiques.....	65
4.2.1	Définition de la condition d'observabilité d'un système linéaire.....	66
4.2.2	Observateur de Luenberger .....	66
4.2.3	Filtre de Kalman .....	68
4.2.4	Filtre de Kalman étendu .....	69
4.2.5	Filtre de Kalman non parfumé.....	71
4.2.6	Filtre d'information .....	74
4.3	Comparaison des trois filtres (EKF, UKF et EIF) .....	75
4.4	Types d'architectures de fusion de données et simulation temps-réel.....	77
4.5	Intérêts et enjeux des architectures distribuées .....	80
4.6	Conclusion .....	81
	 Chapitre 5 Real-time simulator of collaborative and autonomous vehicles .....	83

5.1	Avant-propos .....	83
5.1.1	Auteurs et affiliation.....	83
5.1.2	Date de soumission.....	83
5.1.3	État de l'acceptation .....	83
5.1.4	Conférence.....	83
5.1.5	Référence .....	84
5.1.6	Titre français.....	84
5.1.7	Contribution au document .....	84
5.1.8	Résumé français.....	84
5.1.9	Addendum .....	85
5.2	Abstract.....	88
5.3	Introduction.....	88
5.4	Proposed strategy .....	89
5.5	Real-time simulator architecture for multi-vehicular scenarios.....	91
5.6	Real-time laboratory platform.....	91
5.7	Implementation on OPAL-RT software (RT-LAB) .....	92
5.8	OP 5600 simulator presentation.....	93
5.9	Pro-SiVIC simulator .....	93
5.10	Proportional and integral controller .....	96
5.11	Simulation of Adaptive Cruise Control.....	97
5.12	Conclusion.....	99
	Chapitre 6 Autonomous Vehicle and Real Time Road Lane Detection and Tracking.....	100

6.1	Avant-propos .....	100
6.1.1	Auteurs et affiliation.....	100
6.1.2	Date de soumission.....	100
6.1.3	État de l'acceptation .....	100
6.1.4	Conférence.....	100
6.1.5	Titre français.....	101
6.1.6	Référence .....	101
6.1.7	Contribution au document .....	101

6.1.8	Résumé français.....	101
6.2	Abstract.....	103
6.3	Introduction.....	103
6.4	Simulation scenario.....	104
6.5	Selection of regions of interest corresponding to the road's section in the image ...	105
6.6	Road-lanes detection.....	106
6.6.1	Edge detection .....	106
6.6.2	Edges and painted line colors .....	107
6.7	Hough transform and lines' detection.....	107
6.8	Adaptive region of interest and polynomial approximation .....	108
6.9	Least-square method .....	109
6.10	Kalman filter road lanes prediction and tracking.....	110
6.10.1	Prediction step .....	111
6.10.2	Update step .....	111
6.11	Preliminary results.....	112
6.12	Conclusion.....	114
	 Chapitre 7 Real time cooperative localization for autonomous vehicles.....	115
7.1	Avant-propos .....	115
7.1.1	Auteurs et affiliation.....	115
7.1.2	Date de soumission.....	115
7.1.3	État de l'acceptation .....	115
7.1.4	Conférence.....	115
7.1.5	Référence .....	116
7.1.6	Titre français.....	116
7.1.7	Contribution au document .....	116
7.1.8	Résumé français.....	116
7.1.9	Modification .....	117
7.2	Abstract.....	118
7.3	Introduction.....	118
7.4	Vehicle non-linear model.....	120
7.5	Vehicle neighbors position estimate .....	122

7.6	Extended Kalman filter and state estimate .....	124
7.7	Inter-vehicular communication and cooperative localization.....	125
7.8	Simulation results .....	127
7.9	Conclusion .....	130
	<b>Chapitre 8 From simultaneous localization and mapping to simultaneous collaborative localization and mapping for intelligent vehicles .....</b>	<b>131</b>
8.1	Avant-propos .....	131
8.1.1	Auteurs et affiliation.....	131
8.1.2	Date de soumission.....	131
8.1.3	Etat de l'acceptation .....	131
8.1.4	Revue.....	132
8.1.5	Titre français.....	132
8.1.6	Contribution au document .....	132
8.1.7	Résumé français.....	132
8.2	Abstract.....	134
8.3	Introduction.....	134
8.4	Related works .....	136
8.5	Vehicle localization statement with an extended Kalman filter .....	138
8.6	Simultaneous localization and mapping with EKF .....	140
8.7	Canonical representation and extended information filter for vehicle localization..	143
8.8	Simultaneous localization and mapping with extended information filter .....	144
8.9	Vehicle neighbors state estimate statement .....	148
8.10	Cooperative localization.....	149
8.10.1	Centralized data fusion architecture .....	149
8.10.2	Decentralized data fusion architecture .....	150
8.10.3	Distributed data fusion architecture.....	150
8.11	Fast suboptimal covariance intersection .....	151
8.12	Information matrix fusion .....	153
8.13	Simulation results .....	154
8.14	Experimental scenario .....	160
8.15	Conclusion.....	165

8.16	Appendix A .....	166
Chapitre 9	Online trajectory planning with a modified potential field method on distributed architectures for autonomous vehicles.....	167
9.1	Avant-propos .....	167
9.1.1	Auteurs et affiliation.....	167
9.1.2	Date de soumission.....	167
9.1.3	État de l'acceptation .....	167
9.1.4	Conférence.....	168
9.1.5	Référence .....	168
9.1.6	Titre français.....	168
9.1.7	Contribution au document .....	168
9.1.8	Résumé français.....	168
9.2	Abstract:.....	170
9.3	Introduction.....	170
9.4	Trajectory planning methods .....	172
9.4.1	A-Star (A*) .....	172
9.4.2	Artificial potential field method .....	172
9.5	Trajectory planning technique with APF approach .....	174
9.6	Artificial potential field method and local minima.....	175
9.7	Comparison of MAPF and A* methods .....	176
9.8	Adapted MAPF to vehicle trajectory planning .....	177
9.9	Path planning on distributed architectures .....	178
9.10	Simulation scenario .....	180
9.11	Conclusion.....	182
Chapitre 10	Conclusion générale .....	183
10.1	Bilan .....	183
10.2	Perspectives .....	185
Références bibliographique.....		188

# Liste des abréviations

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AEB	Automatic Emergency Braking
APF	Artificial Potential Field
APFM	Artificial Potential Field Method
AUV	Autonomous Underwater Vehicles
CAN	Controller Area Network
CC	Cruise Control
CCD	Charge-Coupled Device Complementarity
CI	Covariance Intersection
CIV	Communication Inter-Véhiculaire
CL	Cooperative Localization
CMAL	Concurrent Mapping And Localization
CMOS	Metal-Oxide-Semiconductor
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
DDS	Data Distributed Store
DIL	Driver In the Loop
DMIA	Decentralized Minimum Information Algorithm
DSRC	Dedicated Short Range Communication
DTC	Dispositif à Transfert de Charge
ECU	Electronic Control Unit
EES	Equivalent Energy Speed
EIF	Extended Information Filter
EKF	Extended Kalman Filter
ESC/P	Electronic Stability Control/Program
FMI	Functional Mock-up Interface
FPGA	Field-Programmable Gate Array

GFN	Global Fusion Node
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HIL	Hardware In the Loop
I/O	Input/Output
ICSL	Intelligent Control Systems Laboratory
IF	Information Filter
IMF	Information Matrix Filter
INS/IMU	Inertial Navigation/Measurement System/Unit
ISO	International Organization for Standardization
KF	Kalman Filter
Laser	Light Amplification by Stimulated Emission of Radiation
LFN	Local Fusion Node
Lidar/Ladar	Light/LAser Detection And Ranging
LIN	Local Interconnect Network
LIVIC	Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs
LS	Least-Square
LTI	Linear Time Invariant
LTV	Linear Time Varying
MAPFM	Modified Artificial Potential Field Method
Matlab	Matrix Laboratory
MEMs	Micro-Electro-Mechanical systems
MIL	Model In the Loop (Model Based Design)
MTI	modèle linéaire à temps invariant
NA	Naïve Algorithm
NOAA	National Oceanic and Atmospheric Administration
NP	Non-deterministic Polynomial
ODA	Optimal Distributed Algorithm

OEMs	Original Equipment Manufacturers
PC	Personal Computer
PGDA	Potential Gradient Descent Algorithm
PIN	Positive Intrinsic Negative
RADAR	RAdio Detection And Ranging
RAPF	Repulsive Artificial Potential Field
RCP	Rapid Control Prototyping
RGB	Red, Green, Blue
RT-LAB	Real Time LABoratory
RTMaps	Real-Time Multimodal Applications
RVB	Rouge, Vert, Bleu
SIA	Société des Ingénieurs de l'Automobile
SIG	Systèmes d'Information Géographiques
SIL	Software In the Loop
SiVIC	Simulateur Véhicule-Infrastructure-Capteur
SLAM	Simultaneous Localization And Mapping
SOCI	Sub-Optimal Covariance Intersection
SONAR	SOund Navigation And Ranging
T2TF	Track-To-Track Fusion
TCP/IP	Transmission Control Protocol / Internet Protocol
TH	Transformée de Hough
UAVs	Unmanned Aerial Vehicles
UDP	User Datagram Protocol
UKF	Unscented Kalman filter
USGS	United States Geological Survey
UT	Unscented Transform
V2I	Vehicle To Infrastructure
V2V	Vehicle To Vehicle
V2X	Vehicle To X (Vehicle, Infrastructure, Pedestrian, etc.)
VANET	Vehicular Ad-hoc Network
XHP	eXtra-High Performance

# Liste des symboles

$x$	Coordonnée d'un point sur l'axe des abscisses x
$y$	Coordonnée d'un point sur l'axe des ordonnées y
$z$	Coordonnée d'un point sur l'axe des cotes z
$\theta$	Angle de camp d'un objet dans un repère à deux dimensions et orthonormé
$\omega$	Vitesse angulaire d'un objet dans un repère à deux dimensions et orthonormé
$x$	Vecteur d'état d'un modèle d'état
$z$	Vecteur d'observation de mesure d'un modèle d'état
$u$	Vecteur de commande d'un modèle d'état
$w_m$	Vecteur de bruit d'état (perturbations exogènes)
$w_s$	Vecteur de bruit de mesure des modèles capteurs
$\hat{x}$	Vecteur d'estimation (moyenne) du vecteur d'état $x$
$\bar{x}$	Vecteur real de la moyenne du vecteur d'état $x$
$P$	Matrice de covariance
$\Omega$	Matrice d'information
$\xi$	Vecteur d'information
$\chi$	Vecteur de point sigmas
$\omega_i$	Poids de pondération / Points sigma
$(O, \overrightarrow{OX}, \overrightarrow{OY})$	Repère orthonormé à deux dimensions
$t$	Variable temporelle (temps)
$t_0$	Instant initial d'un processus
$\dot{x}$	Première dérivée du vecteur $x$ par rapport au temps
$\ddot{x}$	Deuxième dérivée du vecteur $x$ par rapport au temps
$f_e$	Fréquence d'échantillonnage (de discréétisation)
$T$	Période d'échantillonnage d'un processus ou d'acquisition des capteurs
$f_{NS}$	Fréquence de Nyquist-Shannon

$c$	Célérité des ondes électromagnétiques
$\Delta\phi$	Décalage en phase d'une onde
$A$	Matrice d'état d'un modèle d'état linéaire dans le domaine continu
$B$	Matrice de commande d'un modèle d'état linéaire dans le domaine continu
$C$	Matrice d'observation d'un modèle d'état linéaire dans le domaine continu
$F$	Matrice d'état d'un modèle d'état linéaire dans le domaine discret
$G$	Matrice de commande d'un modèle d'état linéaire dans le domaine discret
$H$	Matrice d'observation d'un modèle d'état linéaire dans le domaine discret
$M$	Matrice d'interaction du bruit d'état sur le modèle
$Q$	Matrice de variance du bruit d'état
$R$	Matrice de variance de mesure capteurs
$K$	Matrice de gain d'observation
$\alpha$	Angle d'inclinaison d'une droite ou d'un plan
$f(*)$	Fonction / transformation non linéaire
$h(*)$	Fonction non linéaire

# Liste des tableaux

Tableau 2-1 : Performances des simulateurs de la dynamique des véhicules, des capteurs embarqués et de l'infrastructure .....	51
Tableau 2-2 : Performances des plateformes de simulation de réseau et de mobilité .....	52
Tableau 4-1 : Paramètres de la simulation.....	77
Tableau 4-2 : Avantages et enjeux des architectures distribuées [87].....	81
Table 7-1 : Sensors' variance matrices.....	130
Table 8-1: Computation complexity order of EKF and EIF.....	144
Table 8-2 : GPS receivers' covariance matrices.....	160
Table 8-3 : Real GPS receivers' and VANET covariance matrices .....	164
Table 9-1: Simulation parameters.....	181

# Liste des figures

Figure 1-1 : Niveaux de la conduite automatisée [10].....	23
Figure 1-2 : Architecture du simulateur RT-LAB/Pro-SiVIC.....	25
Figure 2-1 : Modèle du canal de communication [7] .....	36
Figure 2-2 : Architecture générique d'un véhicule intelligent [8].....	37
Figure 3-1 : Schéma de fonctionnement d'un télémètre optique [67].....	59
Figure 3-2 : Zones de couverture pour les différentes applications automobiles [70]. .....	62
Figure 4-1 : Schéma fonctionnel d'un système linéaire en boucle ouverte.....	65
Figure 4-2 : Schéma fonctionnel du modèle et de l'observateur d'état dans le domaine continu .....	67
Figure 4-3 : Schéma fonctionnel de l'observateur d'état sous forme matricielle dans le domaine continu .....	67
Figure 4-4 : Algorithme du filtre de Kalman pour les systèmes linéaires à temps invariant [76]. .....	68
Figure 4-5 : Algorithme du filtre de Kalman étendu pour l'estimation du vecteur d'état des modèles non linéaires [76].....	70
Figure 4-6 : Schéma fonctionnel de la transformée non parfumée [80].....	72
Figure 4-7 : Algorithme du filtre de Kalman non parfumé pour l'estimation du vecteur d'état des modèles fortement non linéaires .....	73
Figure 4-8 : Algorithme du filtre d'information étendu .....	74
Figure 4-9 : Évolution des éléments des matrices de covariance des trois filtres (EKF, UKF et EIF) reliés à la position du véhicule .....	76
Figure 4-10 : Évolution de l'erreur quadratique moyenne des trois filtres (EKF, UKF et EIF)	76
Figure 4-11 : Types d'architectures de fusion de données [85] .....	78
Figure 4-12 : Graphe d'information d'une architecture de fusion centralisée [85].....	79
Figure 4-13 : Graphe d'information d'une architecture de fusion décentralisée [85].....	79
Figure 5-1 : Version actuelle de la plateforme de simulation de véhicules intelligents et connectés en conduite collaborative .....	85

Figure 5-2 : Prochaine version de la plateforme de simulation de véhicules intelligents et connectés en conduite collaborative .....	86
Figure 5-3: OPAL-RT platform and cores attribution for parallel processing.....	92
Figure 5-4: OP5600 simulator [95] .....	93
Figure 5-5: Exteroceptive sensors in Pro-SiVIC.....	95
Figure 5-6: Closed loop scheme with PI and state feedback controllers.....	96
Figure 5-7: Simulink model of ACC with RT-LAB's standard configurations .....	97
Figure 5-8: Inter-vehicular distance with PI controllers.....	98
Figure 5-9: Vehicles' coordinates evolution .....	98
Figure 5-10: Vehicles' trajectories (Satory test track).....	99
Figure 6-1: Regions of interest .....	106
Figure 6-2: Canny edges detector results .....	107
Figure 6-3: Lines representation in Hough space.....	108
Figure 6-4: Road lanes detection and tracking diagram .....	112
Figure 6-5: Real-time road line detection and road lanes center tracking.....	113
Figure 7-1: Vehicle's model in 2D .....	123
Figure 7-2: Inter-vehicular distance and bearing angle .....	123
Figure 7-3: Cooperative localization with three vehicles .....	128
Figure 7-4: Vehicles position estimate in cooperative localization scenario. ....	128
Figure 7-5: Absolute error on the vehicle positioning on X and Y axis.....	129
Figure 8-1: Vehicle dynamic in 2D .....	140
Figure 8-2: Inter-vehicular distance and bearing angle .....	140
Figure 8-3: Centralized data fusion architecture .....	149
Figure 8-4: Decentralized data fusion architecture.....	150
Figure 8-5: Distributed data fusion architecture .....	150
Figure 8-6: Vehicle trajectory estimation using three GPS receivers' measurements, real position of the vehicle (green line), the other lines are its estimates according to: 1- LFNs with EKFs of each GPS (dashed black, red and blue lines), and 2- GFN of the LFNs with CI, SOCI (dashed magenta and doted lines), and with ODA and CEKF (solid cyan and black lines).....	155

Figure 8-7: RMSE of LFNs with EKFs of each GPS (black, red and blue dashed lines), and of the GFN of the LFNs with CI, SOCI (dashed and dotted magenta lines), and with CEKF and ODA (solid black and dashed cyan lines).....	155
Figure 8-8: Position covariance elements of the local and the global fusion nodes evolution of the GPS receivers data fusion of the LFNs with EKFs of each GPS (dashed black, red and blue lines), and GFNs of the LFNs with CI, SOCI (magenta dashed and dotted lines), and with ODA and CEKF (solid cyan and black lines) .....	156
Figure 8-9: Vehicles RMSE position estimate in cooperative localization scenario .....	157
Figure 8-10: Position covariance elements of the local and the global fusion nodes evolution of the three vehicles in cooperative localization scenario .....	159
Figure 8-11: Blue vehicle trajectory estimation according to the local and to the global fusion nodes: $LFN_{EIF_1}$ related to the first vehicle (red), $LFN_{EIF_2}$ related to the second vehicle (blue), $LFN_{EIF_3}$ related to the third vehicle (green), $GFn_{SOCI}$ (magenta), $GFn_{ODA}$ (cyan) and $GFn_{CEIF}$ (black) .....	161
Figure 8-12: Root mean square error of the local fusion nodes and according to the central optimal EIF .....	162
Figure 8-13: LFNs and GFNs covariance matrix elements evolution.....	163
Figure 9-1: Modified potential field shape and local minima removing .....	176
Figure 9-2: Path planning with MAPF and A* method .....	176
Figure 9-3: Distributed MAPF on Pro-SiVIC and RT-LAB platform .....	181
Figure 9-4: Trajectory planning for intelligent vehicles.....	182

# Chapitre 1

## Introduction générale

La diversification et l'évolution des systèmes d'aide à la conduite « *en : Advanced Driver Assistance Systems (ADAS)* » sont essentiellement propulsées par le développement rapide de deux éléments ; l'intelligence artificielle et les technologies de l'information et des capteurs [7]. Leur principal rôle vise à éviter l'apparition de toute situation dangereuse et à améliorer le confort de conduite. Ainsi, un véhicule intelligent peut être doté d'un ou de plusieurs ADAS. Ces derniers lui permettent d'apprendre, de comprendre et d'interagir avec son environnement à travers les capteurs proprioceptifs et extéroceptifs ainsi que les actionneurs ([69] et [119]-[121], [142]).

De nos jours, plusieurs ADAS sont déjà déployés et opérationnels sur des véhicules commercialisés [69] et [121]. Dans l'ensemble, les systèmes d'aide à la conduite sont répartis en deux catégories :

- Actifs : Cette catégorie s'étend aux fonctions et aux manœuvres qui sont directement effectuées par le véhicule, notamment à des fins d'évitement ou de minimisation d'impact d'un éventuel accident
- Passifs : Cette catégorie se résume aux systèmes intervenant après que l'accident ait eu lieu. Ces derniers avertissent les autorités concernées (pompiers, police, etc.) en leur transmettant les coordonnées de l'accident, ainsi que des informations sur les dégâts matériels et humains

Un autre aspect de l'intelligence, qui caractérise un véhicule intelligent, est sa capacité à se déplacer dans un environnement dynamique en toute sécurité et sans l'intervention d'une tierce entité. La plupart des décisions sont intégralement prises de façon autonome. Exemples : l'itinéraire, la détection d'obstacles, le contrôle de la dynamique du véhicule et la communication, de véhicule à véhicule et de véhicule à infrastructure, sont toutes des tâches que le véhicule doit pouvoir effectuer en complète autonomie.

## 1.1 Niveaux de la conduite automatisée

Dans l’optique de normaliser la conduite automatisée, le ministère des transports des États-Unis a adopté – dans son rapport de septembre 2016 concernant sa politique fédérale sur les véhicules automatisés – la norme J3016 comme référence pour la taxonomie et les définitions des différents niveaux de la conduite automatisée. Cette norme est publiée par la société internationale des ingénieurs de l’automobile « *en : Society Of Automotive Engineers (SAE)* » [2], [29]. Ainsi, six niveaux de conduite automatisée ont été définis. Chaque niveau est alors déterminé en considérant : le type de conducteur, le genre et l’instant de l’action menée [2]. La définition des différents niveaux de conduite automatisée, tels que donnée dans [2], [10] et [29], est comme suit : «

- *Niveau 0 : le conducteur humain accomplit toutes les tâches de la conduite et surveille l’environnement de son véhicule ;*
- *Niveau 1 : Un système d’aide à la conduite peut aider le conducteur à accomplir une partie des tâches de la conduite, comme contrôler la dynamique, latérale OU longitudinale, du véhicule. Cependant, le conducteur humain surveille l’environnement de son véhicule et accomplit le reste des tâches liées à la conduite ;*
- *Niveau 2 : Un ou plusieurs systèmes d’aide à la conduite peuvent accomplir certaines parties de la conduite comme contrôler à la fois les dynamiques latérale ET longitudinale du véhicule, tandis que le conducteur humain surveille l’environnement de son véhicule et accomplit le reste des tâches de la conduite ;*
- *Niveau 3 : le système automatisé peut à la fois effectuer toutes les tâches de la conduite et surveiller son environnement dans certaines conditions. Toutefois, le conducteur doit être prêt à reprendre le volant en cas d’incapacité du système automatisé à accomplir les différentes tâches de la conduite ou de la surveillance de son environnement ;*
- *Niveau 4 : le système automatisé peut à la fois effectuer toutes les tâches liées à la conduite et à la surveillance de son environnement sans avoir à solliciter le conducteur humain. Cependant, le système automatisé peut fonctionner seulement dans certains environnements et sous certaines conditions ;*

- *Niveau 5 : le système automatisé peut effectuer toutes les tâches liées à la conduite et à la surveillance de son environnement dans les mêmes conditions que n'importe quel conducteur humain. »*

La figure 1-1 résume les différents niveaux de la conduite automatisée selon la capacité du système embarqué à surveiller l'environnement du véhicule et à contrôler la dynamique de ce dernier.

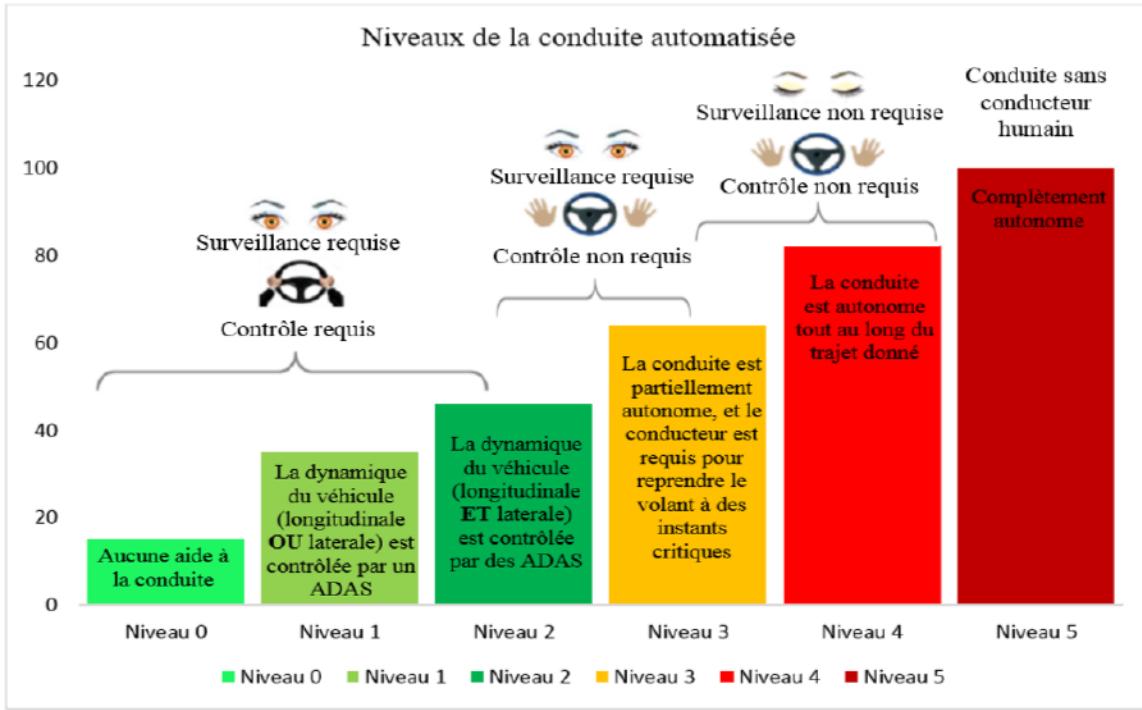


Figure 1-1 : Niveaux de la conduite automatisée [10]

## 1.2 Problématique

Le test et la validation des systèmes d'aide à la conduite, associés aux cinq derniers niveaux (1 à 5), nécessitent un grand nombre de kilomètres à parcourir sur les routes [10]-[11]. Ces tests requièrent souvent beaucoup de temps et des équipements exorbitants qui sont, dans la majorité des cas, expérimentaux. Ainsi, la validation des nombreux scénarios multivéhiculaires sur piste n'est pas une option envisageable (efficace). Par conséquent, il serait intéressant d'avoir recours aux environnements virtuels pour faire passer, à chaque niveau de la conduite automatisée, les tests de validation standards. Cependant, il est nécessaire que ces tests de validation considèrent la complexité des différentes situations qui peuvent se présenter sur la route. C'est pourquoi

notre projet s'inscrit dans l'optique de fournir un moyen d'étude des véhicules intelligents, de leur développement et des tests de validation des nouveaux ADAS coopératifs qu'ils embarquent. Ainsi, l'idée de concevoir une plateforme de simulation de systèmes d'aide à la conduite va dans le sens de nos efforts visant à accélérer le développement et la validation des ADAS.

Cette nouvelle plateforme permettra de générer des scénarios mono et multivéhiculaires avec lesquels il sera possible de simuler le comportement des ADAS et d'identifier leurs limitations. Ainsi, les fabricants automobiles pourront tester et améliorer les performances de leurs systèmes, ceci à travers un travail d'itérations rendu possible par la simulation. Le but sera alors d'analyser les situations critiques engendrées par les limitations des ADAS. Par la suite, il s'agira de passer aux tests réels en reproduisant les pires cas (situations critiques) à des fins d'étude et d'évaluation du comportement de l'ADAS dans la réalité. La plateforme permettra ainsi de réduire le nombre, la durée et le coût de ces tests de validation.

### **1.3 Objectif principal**

L'objectif principal de notre projet de recherche est donc de mettre en place un simulateur temps-réel de véhicules intelligents et autonomes en conduite collaborative. Ce dernier permettra de générer des scénarios complexes (mono et multi-véhiculaires) et d'améliorer la robustesse des algorithmes embarqués (ADAS), avant de passer à la phase de prototypage puis aux tests physiques.

### **1.4 Objectifs intermédiaires**

Cet objectif principal (1.3) repose sur la réalisation d'objectifs intermédiaires importants. Ces derniers, posés dans le cadre de nos travaux, se déclinent en deux volets présentés dans ce qui suit :

#### **1.4.1 Interfaçage des deux plateformes de simulation Pro-SiVIC et RT-LAB**

La nouvelle plateforme de simulation des véhicules intelligents et connectés se basera principalement sur deux plateformes complémentaires :

- **Pro-SiVIC** est la version commerciale de SiVIC « *Simulateur Véhicule-Infrastructure-Capteur* » commercialisée par la société ESI-CIVITEC. Cette plateforme de simulation a été initialement développée par le laboratoire français LIVIC « *Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs* ». Pro-SiVIC est une plateforme de simulation de la dynamique des véhicules, des capteurs embarqués et de l'infrastructure ;
- **RT-LAB** « *en : Real Time LABoratory* » est une plateforme de simulation temps-réel distribuée [94]. C'est le premier logiciel de simulation et de contrôle doté d'un algorithme de séparation des modèles mathématiques, ceci pour l'exécution en parallèle sur un réseau d'ordinateurs de bureau standards [94]. La modularité des architectures parallèles des plateformes d'OPAL-RT et leurs modules d'entrées et de sorties permettent à RT-LAB de communiquer avec son environnement extérieur et de simuler des systèmes complexes.

Ces deux plateformes sont complémentaires et fournissent à la fois un environnement de création de scénarios multivéhiculaires, et une puissance de calcul requise pour la simulation temps-réel sur des architectures parallèles. Plus de détails sur les deux plateformes sont présentés dans les deux chapitres 2 et 5. Ces derniers présentent et listent les différentes caractéristiques des deux plateformes. La figure 1-2 présente l'architecture du simulateur RT-LAB/Pro-SiVIC.

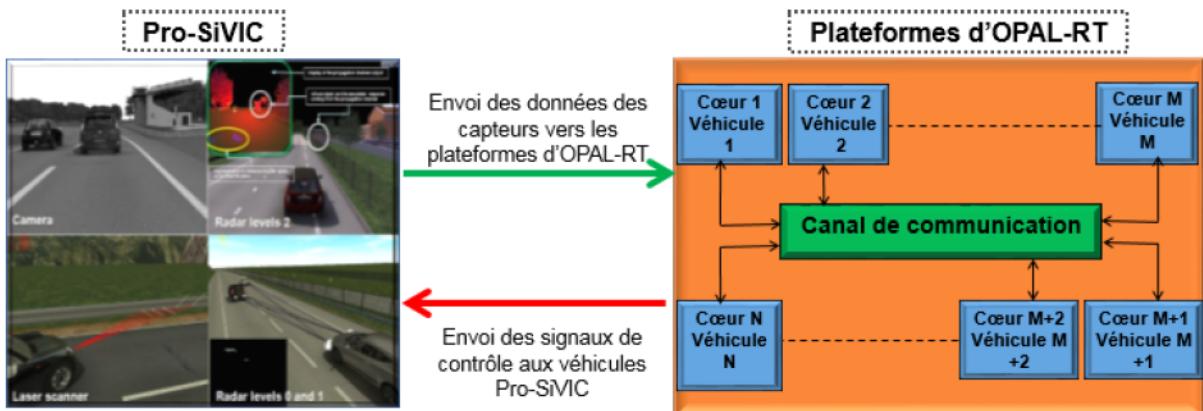


Figure 1-2 : Architecture du simulateur RT-LAB/Pro-SiVIC

### **1.4.2 Développement et simulation des systèmes d'aide à la conduite**

Dans cette partie, nous concentrerons nos efforts sur le développement et la simulation des ADAS sur la nouvelle plateforme RT-LAB/Pro-SiVIC. Nous nous intéresserons notamment à la détection des lignes de marquage au sol, à la localisation et la cartographie simultanées (individuelles et coopératives), et à la planification de la trajectoire du véhicule. Pour ce faire, nous comptons développer et simuler principalement trois scénarios portant sur :

- La fusion des données de deux capteurs proprioceptifs : le premier est à référence globale, et le second est à référence locale. La combinaison des mesures de ces deux capteurs est nécessaire pour surmonter les lacunes de chacun d'entre eux et permettre au véhicule de se localiser dans son environnement ;
- La détection des lignes de marquage au sol et le maintien du véhicule dans une voie de circulation : l'objectif est de garder le véhicule sur la même voie de circulation. Et cela, malgré les incertitudes de localisation dues aux défauts de fabrication des capteurs proprioceptifs et aux bruits de mesure des capteurs extéroceptifs ;
- La planification de la trajectoire du véhicule : cette partie consiste à reconstruire l'environnement du véhicule en 2D à partir des observations des capteurs extéroceptifs. Ceci, afin de détecter, de mesurer et de suivre la position relative des objets qui s'y trouvent dans le but de planifier la trajectoire du véhicule.

## **1.5 Contribution et originalité du projet de recherche**

Les contributions de ce projet résident dans :

- La livraison d'une plateforme de simulation des systèmes d'aide à la conduite et des véhicules intelligents à des fins de développement, de test et de validation des ADAS. Ceci en incluant les configurations monovéhiculaires et multivéhiculaires (Chapitre 5) ;
- Le développement et l'implémentation d'une nouvelle méthode de détection des lignes de marquage au sol sur des architectures distribuées. Ceci pour accélérer le traitement d'image pour des applications en temps réel. Contrairement aux méthodes qui utilisent le filtre à particules et la transformée de Hough (TH) — très gourmandes en matière de

puissance de calcul —, la méthode proposée utilise la TH uniquement pour initialiser la méthode, ou pour la réinitialiser dans le cas de perte du suivi des lignes de marquage au sol (Chapitre 6, section 6.10) ;

- L'utilisation de la nouvelle plateforme pour le développement, l'implémentation et les tests des algorithmes de fusion de données des capteurs embarqués :
  - Proprioceptifs lors de la localisation des véhicules, cas de la localisation individuelle (Chapitre 7, section 7.6, Chapitre 8, section 8.5) ;
  - Extéroceptifs lors de la numérisation de l'environnement du véhicule ; mesure des distances et des angles relatifs entre le véhicule et le reste des objets se trouvant dans son environnement, notamment à des fins de localisation collaborative (Chapitre 7, section 7.5, Chapitre 8, section 8.6).
- La proposition d'une nouvelle approche de calcul exact des matrices d'information et de covariances pour l'estimation des deux premiers moments :
  - des vecteurs d'information ;
  - des vecteurs d'état des véhicules et de leur environnement.

Ceci est réalisé lors des étapes de prédiction et de mise à jour du filtre d'information. Cette méthode permet ainsi d'obtenir les deux estimées qui sont exploitées par des algorithmes de fusion de données de haut niveau, sur des architectures de fusion hiérarchiques et distribuées. De cette façon, le choix est donné à l'utilisateur d'exploiter les performances des différents algorithmes, et de passer d'une représentation à l'autre sans avoir à inverser ces matrices de taille importante (Chapitre 8, section 0) ;

- La résolution du problème des minimums locaux de la méthode des champs de potentiels artificiels. La stratégie proposée consiste à submerger les minimums locaux rencontrés par le véhicule jusqu'à ce que ce dernier atteigne sa destination. Alors, une deuxième itération est effectuée dans le nouveau champ pour trouver une trajectoire viable (Chapitre 9, section 9.8).

L'originalité de notre travail réside dans :

- La mise à l'échelle « *en : scalable* » de la nouvelle plateforme de simulation des systèmes d'aide à la conduite et de véhicules intelligents et coopératifs. Cette plateforme permet de connecter plusieurs simulateurs de la dynamique des véhicules, des capteurs embarqués, de l'infrastructure, de la communication, etc. Ceci à des fins de co-simulation. Les simulations peuvent être à temps réel ou différé (accéléré). La puissance de calcul de la plateforme est aussi évolutive et ne dépend essentiellement que du nombre de plateformes d'OPAL-RT utilisées. Ainsi, la nouvelle plateforme (RTLAB/ProSiVIC) permet de minimiser le nombre des scénarios de tests et de validation des ADAS sur piste. Cela se fait en identifiant les scénarios critiques et en ne reproduisant que les pires cas. Par conséquent, la plateforme permet de minimiser la durée, le coût du développement et de validation de l'ADAS en question (Chapitre 5, section 5.4) ;
- L'atout de la nouvelle stratégie de détection des lignes sur la route, comparativement aux autres, réside dans sa capacité :
  - à réduire et à adapter la taille des zones d'intérêt associées aux lignes de marquage au sol ;
  - à traiter ces zones sur des cœurs différents (Chapitre 6, section 6.8).
- La capacité de prédiction des différents paramètres associés à chaque marquage au sol, permettant ainsi le suivi « *en : tracking* » des paramètres de ces marquages à chaque instant « *t* », même en absence de mesures. Et cela pendant une certaine durée avant que la méthode ne commence à diverger, à cause notamment de l'accumulation des erreurs lors de l'étape de prédiction (Chapitre 6, section 6.10) ;
- L'adaptation de la méthode de localisation et de la cartographie simultanées aux scénarios multivéhiculaires. Contrairement aux autres méthodes qui ont recours à l'optimisation ou à l'approximation d'une partie de la matrice de covariance dans l'étape de prédiction du filtre d'information, l'algorithme proposé permet d'avoir les valeurs exactes des deux matrices d'information et de covariance, et cela sans avoir à inverser directement la matrice d'information (Chapitre 8, section 0) ;

- Le calcul de l'inverse des matrices d'information et de covariance : en ayant recours au lemme de l'inverse de la somme des matrices. On transforme ainsi la complexité du calcul du problème initial de  $O(n^3)$  à  $O(q^3)$ , où  $n$  et  $q$  sont respectivement les tailles des vecteurs d'état et d'observation. Dans le cas de la localisation et de la cartographie simultanées,  $q$  est souvent très inférieur à  $n$  (Chapitre 8, section 0) ;
- L'exploitation de l'aspect creux «en : *sparse*» des matrices d'information et de covariance dans les deux étapes de prédiction et de mise à jour du filtre d'information pour la localisation collaborative (Chapitre 8, section 8.9) ;
- La décorrélation des estimées des véhicules formant des architectures distribuées. Le principe de base repose sur la fusion des estimées locales de tous les véhicules à l'aide du filtre de la matrice d'information. Ce dernier est souvent utilisé à des fins de fusion des données multi-capteur (Chapitre 8, section 8.12) ;
- L'adaptation de la méthode des champs de potentiels artificiels pour la planification de la trajectoire des véhicules. La nouvelle stratégie exploite la distance et la vitesse relatives de chaque objet présent dans l'environnement du véhicule, afin de modifier en ligne les différents paramètres des champs de potentiels répulsifs associés à chaque obstacle (Chapitre 9, section 9.8).

## 1.6 Structure de la thèse

La présente thèse contient dix chapitres. Cette partie fournit un résumé de l'ensemble des chapitres composant ce document.

Ce chapitre montre la nécessité d'avoir une plateforme de simulation des systèmes d'aide à la conduite et des véhicules intelligents et autonomes en conduite collaborative. Ceci à des fins de test et de validation des ADAS. Il donne également un premier aperçu de la problématique du projet de recherche et décrit les objectifs, l'originalité et les contributions de ce projet.

Le deuxième chapitre présente et compare les différents niveaux d'autonomie des véhicules intelligents, les plateformes de simulation de la dynamique des véhicules, les capteurs embarqués, les moyens de la communication, etc.

Le troisième chapitre donne une idée générale sur le principe de fonctionnement des différents capteurs (proprioceptifs et extéroceptifs) et la zone de couverture nécessaire autour d'un véhicule intelligent.

Le quatrième chapitre présente certains outils mathématiques pour l'estimation du vecteur d'état des modèles linéaires et non linéaires. Les différentes méthodes de fusion de données qui y sont présentées sont le filtre de Kalman et ses dérivées. Ces méthodes sont utilisées pour exploiter les mesures de différents capteurs. Ceci en raison de leur performance, et leur faible complexité de calcul. Le chapitre décrit également les différentes architectures de fusion de données.

Le cinquième chapitre présente la première version de la plateforme de simulation des véhicules intelligents en conduite collaborative. Il décrit en détail les composantes et les capacités de la nouvelle plateforme de simulation RT-LAB/Pro-SiVIC.

Le sixième chapitre décrit une nouvelle méthode de détection des lignes sur la route pour des applications en temps réel. L'idée consiste à choisir adéquatement l'emplacement et la taille des zones d'intérêt pour réduire la durée de traitement. Pour ce faire, un filtre de Kalman est mis en œuvre pour prédire les différents paramètres de chaque zone d'intérêt. Cette méthode permet également de paralléliser le traitement de ces zones.

Le septième chapitre présente notre première tentative abordant le problème de la localisation coopérative à base d'architectures distribuées. Il décrit le problème de divergence des filtres centralisés, lors de la localisation, dans le cas de fusion des estimées corrélées. La solution proposée se base sur la création de deux noeuds de fusion de données : le noeud local fusionne les mesures de capteurs indépendantes à l'aide d'un filtre de Kalman étendu, et le noeud global fusionne les estimées corrélées des noeuds locaux à l'aide de la méthode des intersections des covariances.

Le huitième chapitre est une extension du précédent chapitre. Il exploite le principe de la localisation et de la cartographie simultanées et le lemme d'inversion des sommes des matrices pour déterminer les valeurs exactes des matrices d'information et de covariance. La méthode réduit la complexité du calcul et permet au groupe de véhicules intelligents et connectés de se

suivre et de localiser simultanément les différents obstacles mobiles et statiques présents dans leur environnement.

Le neuvième chapitre décrit une solution du problème de la méthode des champs de potentiels artificiels pour la planification de la trajectoire. La reformulation de la problématique de la méthode a permis d'adapter cette dernière pour des applications en temps réel. La méthode est alors adaptée à la fois aux scénarios véhiculaires, en exploitant la distance et la vitesse relatives des objets, ainsi qu'aux plateformes hiérarchiques en distribuant les tâches de calcul des champs de potentiels sur plusieurs cœurs.

Le dixième et dernier chapitre est la conclusion générale. Elle présente une synthèse de notre travail et rappelle nos contributions. On y retrouve également les atouts et les limites des méthodes proposées permettant ainsi de faire des projections sur des travaux futurs.

# Chapitre 2

## État de l'art

Ce chapitre présente les différents travaux, menés sur les véhicules intelligents et sur les plateformes de simulation des systèmes d'aide à la conduite, qui nous semblent les plus pertinents au projet. Le chapitre est composé de trois parties : la première partie décrit les ADAS équipant un véhicule intelligent, tel qu'il est présenté dans le précédent chapitre — niveaux 2 et 3 de la conduite automatisée —. La deuxième partie présente un certain nombre de travaux réalisés sur les véhicules intelligents dans les milieux académique et industriel. La troisième et dernière partie présente et compare les différentes plateformes de simulation dédiées au développement des ADAS et à la simulation des véhicules intelligents et connectés.

### 2.1 ADAS équipant un véhicule intelligent

Dès les années soixante, Gordon Moore — cofondateur d'Intel — avait déjà prédit que, d'une année à l'autre, le nombre de transistors dans une puce électronique se multiplierait par deux. Par la suite, la durée de ce doublement a été rectifiée à deux années [12]. Ainsi, les progrès technologiques en matière de puissance de calcul, d'espace mémoire et de miniaturisation des systèmes électromécaniques « en : *Micro-Electro-Mechanical systems (MEMs)* » ont largement contribué au développement des systèmes d'aide à la conduite. La partie suivante dresse une liste non exhaustive des ADAS équipant un véhicule intelligent et connecté.

#### 2.1.1 ADAS dédiés au contrôle du véhicule

Le niveau de la conduite automatisée, comme présenté dans le premier chapitre, est déterminé par la capacité du véhicule à accomplir un certain nombre de tâches sans recourir au conducteur humain. De nos jours, de nombreux ADAS, équipant les véhicules conventionnels, prennent le dessus dans des situations dangereuses pour minimiser l'ampleur de l'accident [5]. Cependant, il est clair que malgré l'apparition des applications couplant la perception au contrôle longitudinal et/ou latéral du véhicule — correspondant à la conduite automatisée de niveau 2 —

le chemin est encore long pour atteindre la conduite automatisée de niveau 5. Dans ce qui suit, une description des ADAS équipant des véhicules intelligent et mis en vente sur le marché nord-américain [13].

### **2.1.1.1 Régulateur de vitesse intelligent**

Le régulateur de vitesse intelligent « *en : Adaptive Cruise Control (ACC)* » permet de garder une distance de sécurité, entre deux véhicules sur la même voie de circulation, évitant ainsi d'éventuelles collisions entre les deux véhicules [112] et [126]. Cette distance de sécurité est paramétrée par la distance et la vitesse relatives du véhicule suivi. Ces mesures sont fournies par les capteurs extéroceptifs du véhicule hôte, suiveur. En l'absence d'autres véhicules, l'ACC agit comme un simple régulateur de vitesse « *en : Cruise Control (CC)* » [112] régulant ainsi la vitesse du véhicule hôte à une consigne désirée.

### **2.1.1.2 Système de contrôle latéral**

Le correcteur électronique de trajectoire « *en : Electronic Stability Control/Program (ESC/P)* » est le premier ADAS permettant de redresser et de stabiliser la dynamique latérale d'un véhicule [14]. Le système de contrôle latéral permet non seulement de stabiliser le véhicule, mais aussi de le forcer à suivre une trajectoire désirée [15]. Bien que de nombreuses versions, plus performantes, soient développées et adaptées aux besoins des véhicules intelligents, les dynamiques latérale et longitudinale du véhicule sont souvent considérées comme étant découplées ; hypothèse simplificatrice lors de la conception des contrôleurs de ces dynamiques.

### **2.1.1.3 Systèmes de maintien et de changement de voies de circulation**

Dans le milieu urbain, il est important que chaque véhicule puisse garder une voie de circulation et la changer au besoin. Pour ce faire, deux systèmes d'aide à la conduite sont mis en œuvre :

- Le système de maintien de voie de circulation existe sous deux formes : passive et active.
  - En mode passif, l'application détecte les bordures et les marquages de la chaussée, au moyen d'une caméra. Dès que le véhicule s'approche des limites de la voie de circulation, le conducteur est alors alerté afin de redresser son véhicule ;
  - En mode actif, le système redresse le véhicule et le force à maintenir sa trajectoire au milieu de la voie de circulation, ceci en contrôlant la vitesse et la direction du

véhicule — contrôle longitudinal et latéral du véhicule —, tel qu’expérimenté dans [16].

- Lorsque le changement de voie de circulation est requis, le système estime la faisabilité de la manœuvre [18]. Si le changement de voie est réalisable, alors le véhicule maintient la même vitesse et se dirige vers la file à prendre en ayant recours au système de contrôle latéral, sinon il garde sa voie de circulation et active l’ACC [18].

### 2.1.2 Synthèse de la combinaison de plusieurs ADAS

La complexité des ADAS et leurs utilisations, combinées avec d’autres systèmes, pourraient entraîner des conséquences dangereuses pour la sécurité des occupants des véhicules. À titre d’exemple, l’association de l’ACC et du système de maintien de voie de circulation, qui a été expérimentée par la société Honda [5] et [16], a révélé que :

- le conducteur est susceptible à tout moment de faire des manœuvres dans des situations inhabituelles, où il n’est pas conscient ni des dangers environnants ni des conditions routières ;
- Le conducteur risque de ne pas se concentrer sur la route, voire de s’endormir, alors qu’il doit absolument reprendre les commandes.

Un autre accident a eu lieu en mai de l’année 2016, impliquant un véhicule de Tesla, équipée d’un pilote automatique. Celui-ci est entré en collision avec la remorque d’un camion routier traversant une intersection non contrôlée en Floride, États-Unis [17]. L’enquête menée par l’administration nationale de la sécurité routière, reliée au département des transports des États-Unis, a révélé que [17] : «

- *Le pilote automatique fonctionnait au moment de l’accident ;*
- *Le système de freinage d’urgence automatique n’a fourni aucun avertissement, aucune action n’a donc été menée pour éviter la collision ;*
- *Aucune action n’a été entreprise par le conducteur ;*
- *La dernière action enregistrée du conducteur remontait à moins de deux minutes avant l’impact. Cette action consistait à augmenter la vitesse du régulateur de vitesse à 74 milles par heure (~ 120 km/h). »*

L'enquête en elle-même a déterminé que le véhicule de Tesla aurait dû détecter la remorque sept secondes avant que la collision n'ait eu lieu. Cependant, le rapport indique qu'aucun défaut du pilote automatique, concernant la sécurité, n'a été identifié [17].

À la lumière de ces rapports, il est évident que le conducteur est souvent sollicité par son véhicule intelligent durant les situations inhabituelles, et ceci malgré les performances des ADAS formant les trois premiers niveaux de la conduite automatisée. Cependant, il est prévu que la conduite automatisée de niveau 4 résolve ce genre de problèmes sur des itinéraires prédéfinis et contrôlés (circuits fermés, autoroutes, etc.). Ainsi, il faudra attendre le niveau 5 pour que les véhicules intelligents prennent des décisions et agissent en conséquence, en fonction des différents scénarios et situations inhabituelles qui pourront être rencontrés [13].

### 2.1.3 Communication véhiculaire

La communication dans le domaine véhiculaire est principalement utilisée pour diffuser des informations pertinentes concernant l'état et les intentions des véhicules communicants. Elle peut se faire de véhicule à véhicule « *en : Vehicle To Vehicle (V2V)* » ou de véhicule à infrastructure « *en : Vehicle To Infrastructure (V2I)* » [1] [5] [6] et [126]. Afin d'assurer le bon fonctionnement et la sécurité des véhicules autonomes, il est primordial que ces transmissions soient fiables, rapides et adaptées aux contraintes du milieu véhiculaire. D'après G. Silberg et al. [1], la mise en pratique de telles applications nécessite impérativement la satisfaction des points suivants : «

- *Rapidité d'acquisition du réseau et latence faible : le délai de transmission d'un ensemble de données, messages, paquets de données entre deux véhicules ou plus devrait être minimisé le plus possible [6] ;*
- *Haute fiabilité : le système de communication doit surmonter les différents obstacles et/ou pannes qui peuvent survenir d'un moment à l'autre (une panne d'un nœud du réseau ne doit pas bloquer le fonctionnement des autres nœuds) ;*
- *Priorité des messages/questions de sécurité : l'échange d'information doit se faire selon l'impact et l'importance des messages à échanger sur la sécurité ;*
- *Interopérabilité : la capacité d'un ensemble de systèmes à échanger des données et à les interpréter automatiquement et adéquatement ;*

- Sécurité : la capacité de garder l'ensemble des données intactes et hors de portée de toute personne non autorisée à les consulter, modifier, etc. ;
- Confidentialité : les données sont dédiées seulement aux individus autorisés. »

Le DSRC « en : Dedicated Short Range Communication » est une technologie de télécommunication du type ad hoc [1] [5], basée sur une communication sans fil à courte et moyenne distance. Il est dédié aux échanges V2V et V2I et on y définit une structure spécifique de messages à envoyer [1] et [6]. Il permet une communication efficace dans un milieu hautement mobile et où les entités se déplacent très rapidement.

La figure suivante montre les principaux composants participants à l'envoi et à la réception des données à travers le canal de communication.

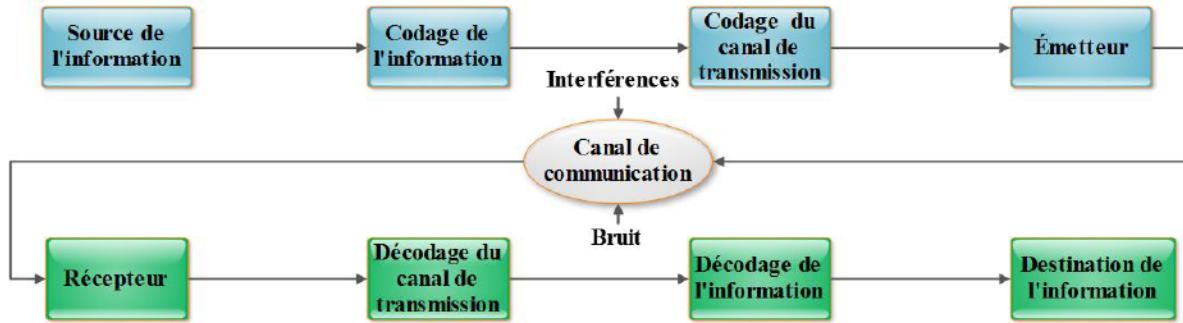


Figure 2-1 : Modèle du canal de communication [7].

#### 2.1.4 Systèmes d'information géographique et données satellitaires

La navigation d'un véhicule intelligent, partiellement ou complètement automatisé, requiert la localisation de ce dernier dans son environnement. Pour cela, des cartographies numériques de son entourage, connues sous le nom des Systèmes d'Information Géographiques (SIG), sont exploitées par le véhicule. Ces SIG sont souvent fournis par des services gouvernementaux ou privés. On retrouve notamment :

- L'administration nationale des océans et de l'atmosphère « en : National Oceanic and Atmospheric Administration (NOAA) » [3] ;
- L'institut d'études géologiques des États-Unis « en : United States Geological Survey (USGS) » [4] ;
- Et le plus célèbre : « Google Earth ».

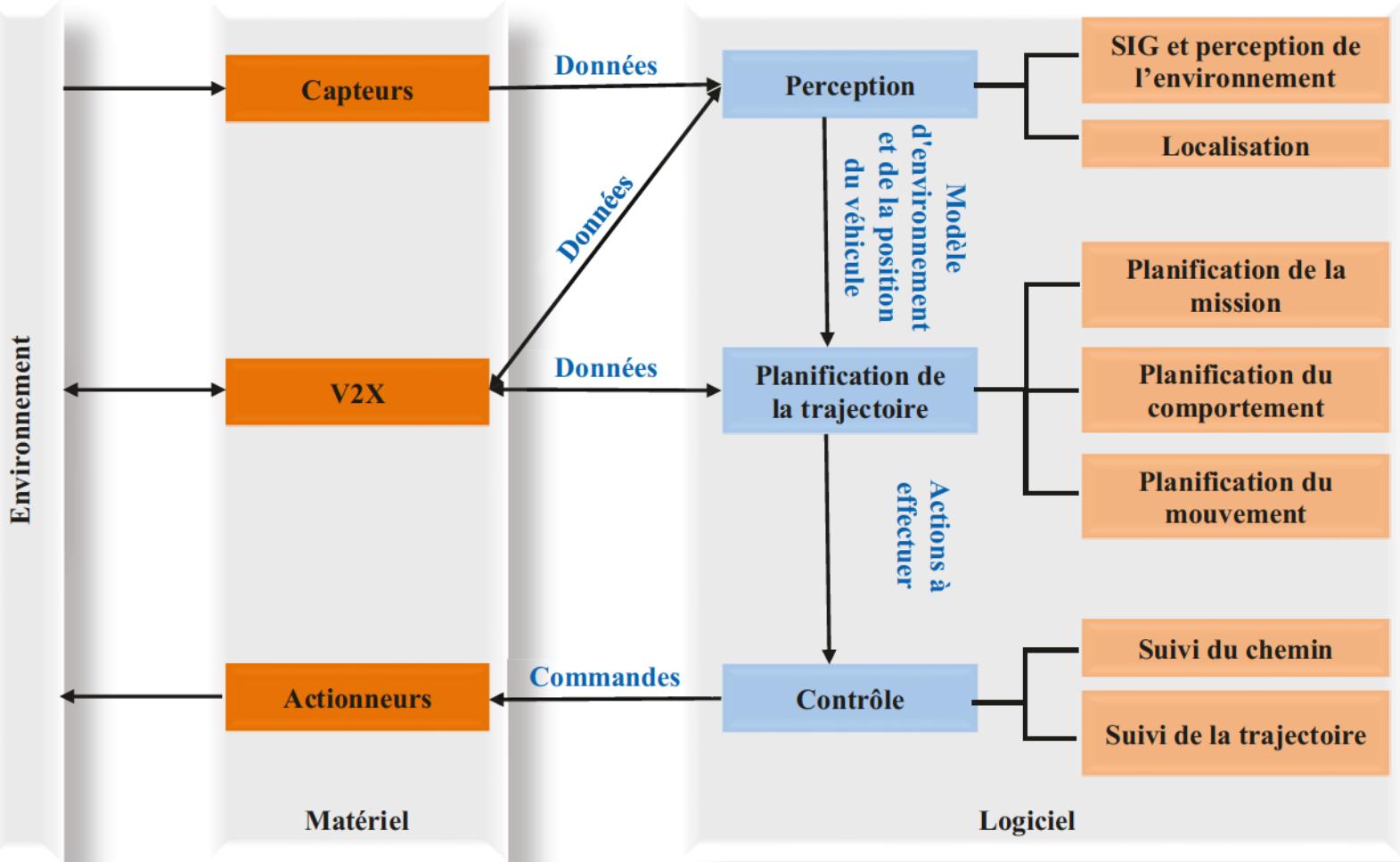


Figure 2-2 : Architecture générique d'un véhicule intelligent [8]

Il est aussi possible de créer des SIG précis en dotant un ou plusieurs véhicules d'un ensemble de capteurs embarqués — proprioceptifs et extéroceptifs — et des algorithmes de fusion de données appropriés. Ce type d'algorithmes est connu dans la littérature sous le nom de la localisation et de la cartographie simultanées « *en : Simultaneous Localization And Mapping (SLAM)* » [139]. Le chapitre 8 aborde en détail ce genre de problèmes. Ces systèmes sont souvent utilisés comme références géographiques locales dans le domaine de la robotique et de l'automobile.

## 2.2 État de l'art sur véhicules intelligents

Après l'aperçu général effectué sur les systèmes d'aide à la conduite, on trouvera dans cette section une brève présentation de certains travaux prometteurs. Dans la plupart de ces travaux, la perception et la localisation représentent les deux principaux aspects d'un véhicule intelligent. Ceux-ci y sont omniprésents, on les retrouve dans tous les véhicules ayant participé aux défis de DARPA « *en : Defense Advanced Research Projects Agency* », comme : « *Grand Challenge* » et « *Urban Challenge* » [18] et [88]. À la lumière de ces défis, de nombreux projets de développement de véhicule autonomes ont vu le jour.

### 2.2.1 Navette Arma de NAVYA

En septembre 2015, la société lyonnaise NAVYA — en France —, spécialisée dans le domaine de la mobilité intelligente [20], a lancé sa première navette autonome Arma (minibus) à Paris [21]. Durant le congrès mondial des systèmes de transport intelligents, qui a eu lieu en octobre de la même année à Bordeaux, la navette Arma a été mise à l'épreuve sur une route ouverte et dans des conditions de trafic réelles [22]. Bien que les navettes Arma ne soient autorisées à emprunter que des trajets prédéfinis, ces derniers s'étendent de plus en plus. En effet, les navettes sont mises à la disposition du grand public dans plusieurs villes, notamment : Lyon en France, Perth en Australie, Nanyang à Singapour, etc. [23].

### 2.2.2 Véhicules de Volvo

Volvo est une société suédoise qui s'est spécialisée, entre autres, dans la construction et le développement des véhicules poids-lourds. Le groupe de Volvo a équipé une de ses flottes de camions avec des ADAS, exploitant les données des capteurs embarqués et les moyens de

communication inter-véhiculaire, ceci à des fins de formation de pelotons véhiculaires [24]. De cette façon, ces camions sont aptes à se suivre entre eux au rythme du camion de tête et en réduisant la distance inter-véhiculaire, et donc en minimisant la traînée aérodynamique des suiveurs [24]. En outre, Volvo développe et adapte ses camions semi-autonomes pour remplir d'autres tâches. On retrouve ainsi des camions à ordures ménagères capables de suivre l'ouvrier dans des environnements fortement dynamiques [25], et des camions de mine roulant sous terre en toute autonomie [24]. En 2013, Volvo a lancé un nouveau projet de voiture autonome sous le nom de « *Drive Me* » [26]. Ce programme permet d'une part au grand public d'essayer les voitures autonomes de Volvo et d'autre part à l'entreprise de récupérer le retour des usagers et les données du trajet parcouru pour améliorer ses véhicules intelligents, notamment en matière du confort et de l'intelligence embarquée.

### **2.2.3 Uber et Lyft**

Dans le cadre du développement des véhicules autonomes, Uber et Volvo se sont associés pour utiliser le système de la conduite automatisée d'Uber à bord des véhicules de Volvo [27]. À cet effet, Uber a engagé une cinquantaine de chercheurs, de l'université Carnegie Mellon, en Pennsylvanie aux États-Unis, pour développer son propre système de conduite automatisée [28] visant le niveau 4 [29]. En effet, Uber planifie de remplacer ses taxis à conducteur humain par des véhicules taxis complètement autonomes [28]. Lyft, un des principaux concurrents d'Uber aux États-Unis, s'est également associé à Ford pour développer et déployer des voitures autonomes d'ici 2021 [30]. De plus, l'entreprise (Lyft) prévoit d'étendre son champ d'action en dehors des États-Unis, en faisant de la ville de Toronto son premier marché international [31].

### **2.2.4 Pilote automatique de Tesla**

Depuis 2015, Tesla a doté certains de ses véhicules de son système de « *pilote automatique* » [28]. Ce dernier comprend principalement trois ADAS [32] : un ACC ; un système de contrôle de la direction ayant comme fonction de garder le véhicule sur la même voie de circulation — en exploitant les marquages sur la route —; et un système de changement de voie de circulation.

### **2.2.5 Voiture de Waymo (Google)**

S'intéressant de plus en plus aux véhicules intelligents, Google se lance dans leur développement dès 2009. Les premières versions de ces véhicules sont dotées du même type de

capteurs que les véhicules participant aux défis DARPA [33]. Deux ans plus tard, en 2011, Google fait évoluer sa vision et propose de mettre au point, dans les cinq années à venir, des véhicules intelligents en mesure de circuler sur les mêmes infrastructures que les véhicules conventionnels [33]. Cependant, Chris Urmson, directeur de projet, a relaté qu'il est plus facile de faire fonctionner les véhicules dans de bonnes conditions climatiques et sur des axes routiers principaux [33], ce qui contribue ainsi à limiter le déploiement des véhicules à certaines zones géographiques, notamment la Californie. En 2016, Google crée une filiale spécialisée dans le développement des véhicules autonomes nommée Waymo [34]. Cette dernière, à travers le programme « *Early rider* », permet aux habitants de Phoenix en Arizona, de participer aux essais publics de ces véhicules autonomes pour une utilisation spécifique (aller au travail, à l'école, etc.) [34].

Il existe encore beaucoup d'autres projets portant sur les véhicules autonomes dont l'application dépend essentiellement d'un domaine à un autre — civil et militaire — tel le projet Talos [18] [35], Valeo [36], NURO [19], etc. Cependant, il s'est avéré qu'il est très important de disposer d'un environnement de simulation permettant d'évaluer et d'améliorer les systèmes d'aide à la conduite [126]. Ou encore, une plateforme de simulation de la conduite automatisée évoluant dans des scénarios mixtes : conduite en mode manuel, semi-automatisé, et/ou entièrement automatisé et collaboratif [37]. Ainsi, le reste du chapitre est consacré à la description et la comparaison de certaines plateformes dédiées à la simulation des ADAS, des moyens de communication et de mobilité.

## 2.3 Plateformes de simulation

L'article [122] présente une nouvelle plateforme de simulation et de débogage des algorithmes embarqués d'un véhicule intelligent. Le véhicule est doté d'un capteur extéroceptif du type CCD/CMOS « *en : Charge-Coupled Device / Complementary Metal Oxide Semiconductor* » pour la détection et le suivi des lignes de marquage au sol. La plateforme comprend : OpenGL « *en : Open Graphics Libraries* » pour construire l'environnement du véhicule et la route en 2D ou 3D ; et la librairie ODE « *en : Open Dynamics Engine* » pour définir la dynamique des différents composants du véhicule et leurs associations.

Les deux apports principaux de cette plateforme sont : l'élimination des erreurs des programmes du microcontrôleur sur la nouvelle plateforme ; et la simulation du traitement des données des capteurs sur un microcontrôleur virtuel. Ainsi, il est possible de tester et de valider, à priori, les algorithmes embarqués.

D'autres travaux portent sur la construction et le développement des simulateurs de conduite en mode HIL « *en : Hardware In the Loop* ». Ils s'appuient sur des outils préalablement expérimentés comme dans l'article [38] où le simulateur est construit à base de deux plateformes :

- ROS « *en : Robot Operating System* » : environnement de création et de simulation des robots offrant un ensemble d'outils et de librairies sous Linux et à code source ouvert ;
- Gazebo : logiciel de simulation réaliste permettant de représenter la dynamique des objets, le comportement des capteurs et l'interaction entre les objets d'une manière réaliste (cause-effet).

Ce genre de simulateur permet de réaliser des tests de contrôle du véhicule, d'entrainer les conducteurs et d'évaluer leurs performances. Grâce aux mécanismes du HIL et de la téléopération, les conducteurs interagissent et sentent l'impact de leurs actions dans l'environnement virtuel. L'analyse des performances de la plateforme « *ROS/Gazebo* » a révélé qu'elle respecte la plupart des conditions de réalisme :

- Importation de l'environnement réel à partir de Google Earth ;
- Interfaçage des capteurs et des actionneurs réels avec le simulateur au moyen des modules d'entrées/sorties et du module HIL ;
- Mise à jour de la scène de simulation à travers les mesures des capteurs et des actions du conducteur ;
- Réalisme d'interaction entre le véhicule et le conducteur grâce à la téléopération ;
- Latence d'interaction acceptable entre le conducteur et le simulateur (cas à un seul conducteur).

Cependant, la simulation des scénarios multiconducteurs augmente la latence du système même en ayant recours à plusieurs ordinateurs.

Dans le même cadre, les mécanismes du conducteur dans la boucle « *en : Driver In the Loop (DIL)* », Kevin S. Swanson et al. [38] fournissent également une courte description de la plateforme de simulation de conduite de Toyota se trouvant à son centre de technique à Higashifuji au Japon. La plateforme est constituée d'une capsule de simulation, d'une vue de 360 degrés et d'un logiciel de simulation de la dynamique des véhicules « *CarSim* ».

Dominique Gruyer et al., [37] et [126], décrivent une nouvelle plateforme de simulation d'une panoplie de systèmes d'aide à la conduite. Celle-ci est basée sur deux logiciels :

- SiVIC (Simulateur Véhicule-Infrastructure-Capteur) : comme son acronyme l'indique, SiVIC est un environnement de simulation de la dynamique des véhicules, des capteurs embarqués et de l'infrastructure ;
- RTMaps « *en : Real-Time Multimodal Applications* » : environnement de traitement des flux de données venant des autres logiciels et des capteurs : caméras, laser, système de localisation globaux et locaux, etc.

La combinaison des deux plateformes (SiVIC et RTMaps) a fourni un moyen assez performant pour le test des ADAS. À titre d'exemple, l'évaluation de l'ensemble des procédures du système de stationnement était d'abord faite sur la plateforme SiVIC/RTMaps, avant que ce système ne soit implémenté sur le véhicule Cycube du laboratoire INRIA [37].

Cette même plateforme fournit un modèle réaliste de communication inter-véhiculaire (CIV). Ce modèle est exploité durant la simulation des scénarios avec plusieurs véhicules en mode coopératif [126]. Dans cet article, on retrouve notamment un scénario à cinq véhicules dont le premier est doté d'un émetteur et les quatre autres de récepteurs. Le protocole de communication inter-véhiculaire utilisé est « *IEEE 802.11p* », dont les paramètres sont déterminés d'une manière empirique après plusieurs expériences, menées sur le site d'essai de Versailles Satory, en France.

Les nombreux tests, selon le scénario défini sur SiVIC, notamment l'ACC en mode coopératif et dont les données sont traitées sur RTMaps, ont montré que la communication inter-véhiculaire engendre de nouveaux défis à relever. En effet, le taux d'accidents diminue au fur et à mesure que les véhicules sont équipés de moyens CIV, et cela jusqu'à l'élimination des accidents sur la

route. Cependant, l'analyse de la quantité d'énergie cinétique « *en : Equivalent Energy Speed (EES)* » échangée a révélé que l'impact des accidents sur le corps des conducteurs est important et souvent fatal. Ceci aboutit au dilemme suivant :

« *Nombre d'accidents/sévérité d'impact : La minimisation du nombre d'accidents à l'aide de la communication inter-véhiculaire augmente la sévérité d'impacts des accidents sur les conducteurs. Et la réduction de la gravité de l'impact des accidents augmente le nombre d'accidents.* »

Cet article présente deux apports principaux pour notre projet, qui sont les suivants :

- L'introduction d'un modèle de communication inter-véhiculaire IEEE 802.11p d'une manière empirique basé sur les données récupérées sur le terrain (circuit Satory) ;
- L'amélioration de la loi de contrôle des véhicules intelligents, en intégrant les différentes anomalies et les comportements humains pour mieux reproduire le comportement du conducteur dans des scénarios multivéhiculaires.

Cependant, le modèle de la communication inter-véhiculaire, implémenté dans la version actuelle du simulateur Pro-SiVIC (produit commercial du simulateur SiVIC), est très basique et ne fonctionne pas en mode DDS « *en : Distributed Data Store* ». En l'occurrence, une application tierce ne peut pas récupérer les données de communication.

### **2.3.1 Simulateurs de la dynamique des véhicules et des capteurs embarqués**

L'utilité des plateformes virtuelles réside dans leur capacité à créer et à simuler des scénarios multivéhiculaires, avec beaucoup de réalisme ; ceci en contrôlant les différents facteurs de l'environnement (la lumière, le brouillard, la pluie, etc.), des conditions de trafic (la densité, la vitesse, etc.), des capteurs embarqués, de la dynamique des véhicules, etc. En effet, la simulation permet de tester et de valider la robustesse des algorithmes de traitement de données et du contrôle avant leur implémentation sur les premiers prototypes expérimentaux.

Dans ce qui suit, nous présentons quelques simulateurs prometteurs — conduite et dynamique des véhicules — pouvant servir à des fins de validation des ADAS et au développement des scénarios multivéhiculaires.

### **2.3.1.1 Pro-SiVIC**

Pro-SiVIC, comme introduit précédemment, est la version commerciale du simulateur SiVIC. Il a été commercialisé par la société CIVITEC, puis par la société ESI-Group après que celle-ci a rachetée CIVITEC [40]. Cette plateforme de simulation a été initialement développée par le laboratoire LIVIC «*Laboratoire sur les Interactions Véhicules-Infrastructure-Conducteurs*» [118]. Pro-SiVIC est une plateforme de simulation et de prototypage virtuel. Elle permet de réaliser ou de reproduire des simulations 3D d'environnements et de capteurs avec beaucoup de réalisme [39] [118]. La panoplie de capteurs disponibles dans la bibliothèque de Pro-SiVIC : caméras, télémètres, centrales inertielles, odomètres, etc., permet de générer des mesures sur l'état des véhicules et de leur environnement. Ces mesures sont alors soit enregistrées sur le disque dur pour des post-traitements, soit transmises à d'autres applications telles que : RTMaps, Matlab, etc. Le mode DDS permet aux véhicules de recevoir les signaux de contrôle de la dynamique des véhicules simulés venant des applications tierces permettant ainsi de lancer des simulations en mode : DIL, HIL et SIL.

### **2.3.1.2 PreScan**

PreScan est une autre plateforme de simulation des ADAS, développée par la société «*TASS International*». La plateforme permet de créer des scénarios multivéhiculaires. Les limitations de la plateforme se résument principalement dans la liste très restreinte des capteurs embarqués. En outre, le comportement de ces capteurs n'est pas réaliste. Les données des capteurs sont directement extraites du vecteur d'état des objets, préalablement identifiés lors de la création d'un scénario. Cependant, il est prévu que «*TASS International*» remédie à ce problème en développant des modèles capteurs plus réalistes qui seront inclus dans les prochaines versions [41].

PreScan a la capacité d'importer des réseaux routiers à partir d'un fichier externe. Ainsi, cela permettra à l'utilisateur de reproduire et de visualiser tous ses tests expérimentaux sur la plateforme. Il pourra ensuite développer des stratégies de traitement et de contrôle à partir des mesures expérimentales. Toutefois, le logiciel ne supporte que les cartes routières au format «*OpenStreetMap*» [41]. PreScan est également doté de modules de télécommunication — émetteurs et récepteurs — idéaux, utilisés pour la conception et l'évaluation des applications de communication V2V et V2I, cas des scénarios multivéhiculaires. L'exportation des scénarios

de PreScan vers Matlab/Simulink facilite son utilisation pour la conception de différents systèmes de contrôle et de test : HIL, MIL « *en : Model In the Loop* » et SIL.

### **2.3.1.3 CarMaker**

CarMaker est l'un des produits commerciaux de la société « *IPG Automotive* ». La plateforme est principalement dédiée à la simulation des véhicules légers (des voitures). Elle permet de simuler la dynamique des véhicules, des systèmes d'aide à la conduite et des modèles de routes avec beaucoup de réalisme [42]. Le simulateur offre également la possibilité de spécifier des modèles (maneuvres) de conducteurs et de les intégrer facilement dans le scénario simulé. Cependant, le comportement des capteurs embarqués est semblable à celui de PreScan. En outre, le nombre de véhicules intelligents, véhicules équipés, simulés est limité à un. Ce qui est un vrai obstacle pour la simulation des scénarios multivéhiculaires en mode coopératif.

Grâce aux nombreuses API « *en : Application Programming Interface* » de CarMaker et l'exportation de ses modèles sur Matlab/Simulink, le simulateur permet de tester et de valider des ADAS en mode : HIL, MIL et SIL.

### **2.3.1.4 VehicleSim**

Cette plateforme inclut trois simulateurs de voitures, des camions et des véhicules à deux roues (CarSim, TruckSim et BikeSim). Ces produits sont développés et commercialisés par la société « *Mechanical Simulation* ». Chacun de ces produits simule la dynamique et le comportement du véhicule dédié, les interactions entre les véhicules, le conducteur et les routes [43]. Les principaux atouts de la plateforme résident dans : la précision des modèles mathématiques de la dynamique des différents véhicules simulés, et dans sa capacité d'interfaçage à de nombreuses plateformes. À titre d'exemple, CarSim est déjà interfacé à Matlab/Simulink, aux plateformes d'OPAL-RT, de dSpace, etc.[44]. Par conséquent, ce simulateur est fonctionnel dans les modes : DIL, HIL, MIL et SIL.

Cependant, cette concentration particulière sur la modélisation précise de la dynamique des véhicules semble être au détriment du développement des capteurs embarqués. Et comme présenté précédemment, il est nécessaire que les véhicules intelligents soient dotés d'un ensemble de capteurs, proprioceptifs et extéroceptifs, et des moyens de communication réalistes pour la simulation de scénarios multivéhiculaires.

### **2.3.1.5 SCANeR Studio**

OKTAL est une firme française proposant des solutions logicielles dans différents domaines : aéronautique, automobile, ferroviaire, etc.[45]. Parmi ses nombreux produits, on retrouve notamment le simulateur « *SCANeR Studio* » de la dynamique de véhicules, des capteurs et de l'infrastructure. Ce dernier est l'une des solutions que l'entreprise propose pour le développement et la simulation des scénarios multivéhiculaires [46]. Cette plateforme permet de lancer des co-simulations avec plusieurs autres produits tels que : Matlab/Simulink, VI-CarRealTime, etc.

Grâce à ses différentes APIs, SCANeR Studio peut également fonctionner en modes SIL et HIL (cas de SCANeR DT « *en : SCANeR Driver Training* ») [46]. De plus, le logiciel est aussi doté de moyens de communication UDP/IP qui lui permettent d'échanger des données avec des plateformes tierces [47].

## **2.3.2 Simulateurs de réseau et de mobilité**

### **2.3.2.1 EXata**

EXata est une plateforme d'émulation et de simulation de réseaux de communication de la société « *Scalable Network Technologies* ». Elle permet de créer et de reproduire numériquement le comportement d'un réseau au complet, comprenant les différents composants tels que : les antennes, les différentes couches de protocole de communication (la pile protocolaire), etc. [48].

En mode émulateur, la plateforme émule les réseaux de communication avec beaucoup de réalisme. Sa capacité d'exécution en temps réel lui procure la possibilité de se connecter à des applications externes et au matériel physique. La plateforme peut donc fonctionner en mode SIL et HIL, étendant ainsi ses capacités à des fins de co-simulations [48]. En mode simulateur, la plateforme désactive toutes les fonctionnalités du mode émulateur pour exécuter les scénarios développés au-delà du temps réel, aussi vite que possible. Ce mode est souvent utilisé à des fins de développement et d'évaluation du comportement des réseaux avant de passer en mode d'émulation afin de tester et de valider les stratégies mises en œuvre en boucle fermée (logiciel ou matériel physique dans la boucle).

L’aspect évolutif « *en : scalability* » de la plateforme lui permet d’exploiter les ressources du matériel mis en œuvre, processeurs multicœurs, pour simuler et émuler de grands réseaux de 100 à 1000 appareils. En plus de la portabilité des modèles d’EXata et de ses modèles fidèles à la réalité (protocoles des standards, environnement sans fil, etc.), la plateforme permet également de simuler des cyberattaques offrant ainsi la possibilité d’évaluer les risques et les dégâts d’éventuelles attaques sur des véhicules connectés. Elle est notamment compatible sur les plateformes Windows et Linux.

### **2.3.2.2 SUMO/QualNet**

SUMO « *en : Simulation of Urban Mobility* » est un logiciel de simulation de trafic routier et à code source ouvert. Le simulateur comprend des modèles du trafic routier. Il permet de créer divers environnements de simulation de trafic réaliste incluant de nombreux types de véhicules et de réseaux [49]-[50]. Le simulateur inclut un ensemble d’applications facilitant la création de réseaux routiers, ainsi que l’importation et l’exportation d’environnements routiers. Il est compatible sous Windows et Linux.

QualNet est un autre produit commercial de « *Scalable Network Technologies* » [51]. Cette plateforme est très similaire à la version d’EXata en mode simulateur. Elle permet de créer et de simuler le comportement des réseaux de communication [51].

Bien que SUMO soit doté d’API telle que TraCI « *en : Traffic Control Interface* », des contraintes d’interfaçage de la plateforme à des fins de co-simulation restent encore non résolues et rares sont les versions (SUMO) qui sont dotées de ce genre d’API. Par conséquent, l’usage de SUMO/QualNet, à des fins de co-simulations, est limité par les API de cette plateforme.

### **2.3.2.3 SUMO/NS-2**

Une autre alternative à la solution commerciale de QualNet est NS-2 « *en : Network Simulator* » [50]. Ce dernier est très utilisé dans le domaine de la recherche, cela est principalement dû au fait que la plateforme est à code source ouvert et qu’elle est accompagnée d’une riche documentation et de nombreux exemples propres à lui.

Comme beaucoup de logiciels gratuits, certains modules de NS-2 ne sont pas maintenus à jour, cela limite leur utilisation [50]. Une autre contrainte handicapant le simulateur est sa gourmandise en matière d’espace mémoire limitant ainsi le nombre de nœuds simulés [50].

#### **2.3.2.4 NS3**

Contrairement à NS2, NS3 est très modulaire, ceci principalement grâce à son architecture se basant sur un ensemble de librairies qui lui permettent de créer et de simuler des scénarios complexes. La combinaison de ces librairies entre elles ainsi qu'avec d'autres librairies, propres à d'autres logiciels, se fait sous deux environnements de programmation nommément : « C++ » et Python [52]. Certains modules de NS2 ont été exportés vers NS3 et sont actuellement fonctionnels. La version actuelle (NS-3.28) fonctionne principalement sous Linux. Néanmoins, son utilisation sous l'environnement Windows est possible à travers Cygwin [53].

### **2.3.3 Plateformes temps réel et calcul parallèle**

Cette partie présente quatre principales plateformes pour la simulation temps-réel et le calcul parallèle. Les quatre firmes, développeuses et vendeuses de ces produits, sont des leaders de renommée internationale.

#### **2.3.3.1 Plateformes de dSpace**

dSpace est une firme allemande de Paderborn et l'un des leaders mondiaux de fabrication des outils d'ingénierie orientés vers le développement et les tests de systèmes de contrôle mécatroniques dans divers domaines académique et industriel : réseaux électriques, automobiles, aviation, etc. [54]. Parmi ses produits, on retrouve notamment « *ModelDesk* » portant sur la simulation des systèmes d'aide à la conduite. Cette plateforme offre un ensemble d'outils de simulation des applications automobiles ASMs « *en : Automotive Simulation Models* » comprenant des modèles : de moteurs à combustion (essence et diesel) ou électriques ; de la dynamique du véhicule ; de composants électriques ; de l'environnement routier [55].

Ce simulateur a beaucoup de points communs avec la plateforme PreScan, à titre d'exemple : la limitation des capteurs embarqués, des moyens de communication de véhicule à véhicule et de véhicule à infrastructure (V2X), considérés idéaux, etc. Les modèles Matlab/Simulink sont également à code ouvert, permettant ainsi aux utilisateurs de concevoir des systèmes d'aide à la conduite à base de ces modèles et les tester en différent mode : RCP « *en : Rapid Control Prototyping* », MIL, SIL et HIL. L'atout principal de cette plateforme réside dans sa capacité à fonctionner en temps réel sur du matériel de dSpace (p. ex. DS1006) [56].

VEOS est une autre solution de dSpace qui porte sur la validation des algorithmes embarqués des ECU « *en : Electronic Control Unit* » [57]-[58]. Grâce aux différentes APIs de communication et aux modules FMI « *en : Functional Mock-up Interface* » implémentés sur VEOS, ce dernier est capable de simuler plusieurs modèles des systèmes embarqués sur PC « *en : Personal Computer* » [57]. Ces modules lui procurent une capacité de co-simulation importante avec plusieurs plateformes. À titre d'exemple, VEOS est déjà connecté à plusieurs environnements de simulation de la dynamique véhicules, des capteurs embarqués et de l'infrastructure notamment : Modelica, ModelDesk, AMS [...], et à des environnements de traitement et de fusion de données et de contrôle tels que RTMaps, Simulink, ControlDesk, etc.[57].

### **2.3.3.2 Plateformes de National Instruments**

National Instruments (NI) est une firme américaine et un autre leader mondial en matière de tests automatiques des équipements et de la simulation en temps réel. VeriStand est l'un de ses produits les plus aboutis. En effet, ce dernier (VeriStand) est un environnement de création et de configuration des équipements et des applications en temps réel, et cela sur des plateformes multicœurs. Il comprend plusieurs fonctionnalités en mode SIL/HIL et d'autres APIs (FMI) pour lancer des co-simulations et communiquer avec les autres environnements de simulation, de traitement de données et de contrôle. VeriStand est également doté de plusieurs bus de communication : CAN « *en : Controller Area Network* », LIN « *en : Local Interconnect Network* » et FlexRay [59]. Parmi les nombreux simulateurs de la dynamique des véhicules, des capteurs embarqués, de l'infrastructure et du comportement du conducteur, auxquels les plateformes de NI sont interfacées, on retrouve notamment : CarMaker de IPG, CarSim de « Mechanical Simulation », Dynacar de Tecnalia, VI-CarRealTime de VI-Grade, DYNA4 Framework de « TESIS DYNAAware », etc.[60]. Ces nombreux simulateurs permettent ainsi à VeriStand d'étendre ses capacités de développement, de co-simulation et de test des systèmes d'aide à la conduite.

### **2.3.3.3 Plateformes de Concurrent Real-Time**

Concurrent RT (Real-Time) est une autre firme américaine qui s'est spécialisée dans le développement du système d'exploitation temps-réel « *RedHawk Linux* ». Ainsi, il offre une panoplie de solutions temps réel dans différents domaines : véhiculaire, aérospatial, électrique,

etc. Parmi ses nombreux produits, on retrouve notamment SimWB « *en : SIMulation Workbench* » qui est un environnement de modélisation, de simulation et de test des systèmes aérospatiaux et automobiles [57][61]. Il comprend plusieurs fonctionnalités telles que : HIL, RCP, MIL « *en : Man In the Loop* ». En outre, SimWB prend en charge plusieurs environnements de modélisation et de simulation notamment : Matlab/Simulink de Mathworks, MapleSim de MapleSoft, Dymola de « Dassault Systèmes » [...], et des environnements de simulation de la dynamique des véhicules et des capteurs embarqués tels que : VI-grade, veDYNA de « TESIS DYNAware », CarSim de « Mechanical Simulation », etc. Son support du FMI lui procure également la capacité de communiquer avec le reste des environnements et équipements supportant ce même standard [61].

#### 2.3.3.4 Plateformes d'OPAL-RT

OPAL-RT est une firme canadienne et leader mondial du développement de simulateurs temps réel — basés sur des ordinateurs standards et sur des FPGA « *en : Field-Programmable Gate Array* » — et d'équipements de test (HIL et RCP) [94]. Les domaines d'applications de ses produits sont divers et presque identiques à ceux de dSpace et de NI notamment : les réseaux électriques, l'électronique de puissance, les motorisations, l'automobile, le ferroviaire, l'aéronautique, etc. L'un des produits le plus abouti d'OPAL-RT est RT-LAB « *en : Real-Time Laboratory* ». Ce dernier est une plateforme de simulation temps-réel distribuée [94]. Elle exploite un ensemble de modèles mathématiques et/ou de modèles dynamiques — conçus principalement sur : Simulink, C/C++, Python, etc. —, pour une simulation temps réel [94]. RT-LAB est le premier logiciel de simulation et de contrôle muni d'un algorithme de séparation des modèles mathématiques. Ainsi, il permet l'exécution parallèle de ces modèles sur différents types de plateformes, que ce soit des plateformes dédiées ou sur un réseau d'ordinateurs standards [95]. La modularité de ces architectures parallèles et de leurs modules d'entrées/sorties lui permet de communiquer avec l'environnement extérieur et de simuler des systèmes complexes.

## 2.4 Comparaison des plateformes

Cette partie présente une vue générale et une liste des différentes caractéristiques et performances des plateformes de simulation présentées ci-haut.

Tableau 2-1 : Performances des simulateurs de la dynamique des véhicules, des capteurs embarqués et de l'infrastructure

Caractéristiques Simulateurs	Fidélité de la dynamique des véhicules	Fidélité des capteurs embarqués	Scénarios multivéhicules intelligents	Communication V2X		Facteurs environ- nementaux et climatiques	Fidélité temps réel	S I L	H I L	Gr atu ité
				Idéale	Réaliste					
Pro-SiVIC	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗
PreScan	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗
CarMaker	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
SiVIC/RTMaps	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗
VehicleSim	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗
SCANeR Studio	✓	✓	✓	✗	✗	✓	✗	✓	✓	✗
VTD/OMNet++	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗
DYNACAR	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗
GT-SUITE	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗
ROS/Gazebo	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓
rFactor	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
TORCS	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
Racer	✓	✗	✓	✗	✗	✓	✓	✗	✗	✓

Tableau 2-2 : Performances des plateformes de simulation de réseau et de mobilité

Caractéristiques Simulateurs	Scenarios multi-véhicules	Communication V2X	Fidélité temps réel	SIL	HIL	Gratuité
EXata	✓	✓	✓	✓	✓	✗
SUMO/QualNet	✓	✓	✗	✓	✗	✗
SUMO/NS-2	✓	✓	✗	✓	✗	✓
NS3	✓	✓	✓	✓	✓	✓
iTetris	✓	✓	✓	✗	✗	✗
Aimsun Next	✓	✗	✓	✗	✗	✗

## 2.5 Conclusion

Plusieurs autres plateformes de simulation des systèmes d'aide à la conduite et des véhicules intelligents existent — en mode SIL [35] et [122] ou en mode HIL [38] —. Parmi elles, on estime que les plateformes SiVIC/RTMaps, VEOS, VeriStand et SimWB sont les plus prometteuses, les plus adaptables et les mieux équipées en termes de fonctionnalités pour la simulation de scénarios multivéhiculaires en conduite coopérative.

À la lumière de l'étude comparative des différentes plateformes, notre choix se porte sur la plateforme SiVIC/RTMaps. On utilisera cette dernière pour proposer une autre plateforme de simulation temps-réel avec une puissance de calcul nettement supérieure. La première version de cette nouvelle plateforme sera alors constituée des plateformes d'OPAL-RT pour le calcul parallèle et la simulation en temps réel d'une part, et d'autre part, de Pro-SiVIC pour la création et la simulation de scénarios mono et multivéhiculaires. Chaque scénario comprendra un certain nombre de véhicules intelligents et connectés équipés de capteurs embarqués, ceci à des fins de développement de nouveaux ADAS, de traitement de signal et de tests de validation des différents niveaux de la conduite automatisée.

Il est à noter que, dans le futur, cette plateforme évoluera pour inclure d'autres plateformes afin d'exploiter les performances de ces dernières d'une part, et d'autre part, remédier aux lacunes et apporter des améliorations à la plateforme de simulation de la dynamique des véhicules et des capteurs embarqués — Pro-SiVIC —. Parmi ces lacunes, on retrouve notamment la limitation des moyens de communication de véhicule à véhicule et de véhicule à infrastructure. L'évolution de la plateforme pourrait également concerner la puissance de calcul, et ceci en ayant recours à l'exportation de l'ensemble des modèles et des scénarios mono et multivéhiculaires sur une infrastructure informatique dématérialisée « *en : cloud computing* ». L'exploitation de l'ensemble de ces modèles et de ces scénarios sur le cloud permettra de lancer des simulations accélérées à des fins d'identification des limites des systèmes de la conduite automatisée sur plusieurs millions, voire plusieurs milliards de kilomètres — en fonction de chaque niveau de la conduite automatisée —.

# **Chapitre 3**

## **Capteurs embarqués**

### **3.1 Introduction**

Un véhicule intelligent ne peut être considéré comme tel, s'il ne peut pas à la fois déterminer sa position, scanner son environnement et s'y situer. Ce genre d'application nécessite deux types de capteurs : proprioceptifs et extéroceptifs. La nature, le nombre et la fidélité des capteurs mis en œuvre jouent tous un rôle important dans la qualité des informations transmises au véhicule. Néanmoins, réduire au minimum le coût et le poids de l'équipement embarqué dans le véhicule devient une nécessité. En effet, cette quête est imposée aux constructeurs automobiles qui favorisent toute nouvelle technologie leur permettant de maximiser leurs profits d'une part, et d'autre part de se hisser au rang des leaders du marché automobile international avec des produits modernes, innovants, précis, fiables et surtout compétitifs et accessibles au grand public. C'est pour toutes ces raisons que l'on retrouve une utilisation massive de capteurs bon marché, dont la plupart sont déjà installés dans les véhicules actuellement commercialisés. Ils sont notamment exploités dans les différentes méthodes de fusion de données. Ces dernières seront présentées dans le prochain chapitre.

Ce chapitre a pour objectif de présenter les capteurs embarqués utilisés dans ce projet, ainsi que leur principe de fonctionnement. Les modèles de ces capteurs font partie de la panoplie de modèles présents dans le simulateur Pro-SiVIC, ils sont donc prêts à être exploités. Ces capteurs ont souvent accès à des informations partielles du vecteur d'état du véhicule et de son environnement. Par conséquent, la reconstruction de ce vecteur, au complet, se fait à travers la fusion de données de l'ensemble des capteurs embarqués et des modèles d'état du système en étude.

## 3.2 Type de capteurs embarqués

Les capteurs embarqués peuvent se répartir en deux catégories. Cette répartition est faite selon la nature de l'information que les capteurs fournissent. Leurs mesures partielles peuvent soit être liées au vecteur d'état du véhicule, soit à celui de son environnement. Ainsi, on retrouve les capteurs proprioceptifs et les capteurs extéroceptifs.

### 3.2.1 Capteurs proprioceptifs

Les capteurs proprioceptifs fournissent des informations partielles sur le vecteur d'état de l'objet hôte. Ceux-ci se divisent en deux grandes catégories que nous allons aborder dans les sous-parties suivantes ; à références locale et globale.

#### 3.2.1.1 À référence locale

Les capteurs proprioceptifs à référence locale permettent aux véhicules de déterminer leur position par rapport à une référence donnée, souvent prise comme point de départ du véhicule. La prédiction et l'évolution de la dynamique du véhicule nécessitent de connaître à priori certaines informations [63], comme les mesures de certaines grandeurs, notamment : accélération, vitesse, orientation, etc. Ci-après, un descriptif d'un des capteurs proprioceptifs à référence locale (centrale inertielle) utilisé dans ce projet.

La centrale inertielle « *en : Inertial Navigation/Measurement System/Unit (INS/IMU)* » est placée au niveau du centre de gravité du véhicule. On retrouve sur chaque axe du repère orthonormé  $(O, \overrightarrow{OX}, \overrightarrow{OY}, \overrightarrow{OZ})$ , lié à la centrale inertielle, un accéléromètre et un gyroscope. Ces derniers mesurent les accélérations et les rotations subies par l'INS sur ses trois axes : roulis, tangage et lacet. Ainsi, l'INS permet de reconstruire l'ensemble des variables du vecteur d'état de la dynamique du véhicule, ceci à l'aide des observateurs et des filtres (linéaires ou non linéaires). En effet, en se basant sur les conditions initiales du véhicule, dans un référentiel local, il est possible de retrouver la vitesse et la position du véhicule en intégrant, en fonction du temps, les accélérations subies et mesurées par la centrale inertielle comme suit.

$$\begin{cases} x = x_{t_0} + \dot{x}t + \frac{1}{2}\ddot{x}t^2 \\ \dot{x} = \dot{x}_{t_0} + \ddot{x}t \end{cases} \quad (3.1)$$

Tels que:  $x = [x \ y \ z]^T$ ,  $\dot{x} = [\dot{x} \ \dot{y} \ \dot{z}]^T$  et  $\ddot{x} = [\ddot{x} \ \ddot{y} \ \ddot{z}]^T$  sont, respectivement, les vecteurs de position, de vitesses et d'accélération du véhicule sur ses trois axes. Le seul vecteur accessible, pour la centrale inertie, est le vecteur d'accélération  $\ddot{x}$ . Cet ensemble d'équations décrit, dans le domaine continu, l'évolution des deux premiers vecteurs d'état ; la position et la vitesse en fonction des accélérations et du temps. Cependant, le traitement numérique nécessite un passage au domaine discret en transformant l'équation (3.1) selon certaines conditions :

- Le temps d'intégration numérique doit respecter le théorème de Nyquist-Shannon  $T = \frac{1}{f_e} \leq \frac{1}{2f_{NS}}$ , où  $f_e$  est la fréquence de discréétisation du modèle et  $f_{NS}$  est la fréquence de Nyquist-Shannon ;
- La variation des grandeurs mesurées et estimées, durant la durée d'échantillonnage  $T$ , est considérée comme étant nulle.

$$\begin{cases} x_{k+1}^p = x_k^p + Tx_k^v + \frac{T^2}{2} x_k^a \\ x_{k+1}^v = x_k^v + Tx_k^a \end{cases} \quad (3.2)$$

Où  $x_k^p$ ,  $x_k^v$  et  $x_k^a$  sont, respectivement, les vecteurs de la position, de la vitesse et de l'accélération mesurées sur chacun des axes du véhicule à l'instant  $t=t_0+kT$ ; T est la période d'échantillonnage de la centrale inertie correspondant au temps nécessaire pour avoir une nouvelle acquisition (taux de rafraîchissement des données de la centrale),  $t_0$  est le temps initial et k est un nombre entier positif, correspondant à la k<sup>ème</sup> acquisition.

La représentation matricielle du modèle discret du système (3.2) peut être exprimée sous forme d'un modèle linéaire à temps invariant (MTI) « *en : Linear Time Invariant (LTI)* » comme suit :

$$\begin{cases} x_{k+1} = F_k x_k + G_k u_k + M_k w_{m,k} \\ z_k = H_k x_k + w_{s,k} \end{cases} \quad (3.3)$$

Où :

- $x_k$ ,  $z_k$ ,  $u_k$ ,  $w_{m,k}$  et  $w_{s,k}$  sont, respectivement, les vecteurs d'état du modèle, des mesures, de commande, des perturbations exogènes et des bruits des capteurs ;
- $F_k$ ,  $G_k$ ,  $M_k$  et  $H_k$  sont, respectivement, les matrices d'état (de transition), de commande, d'itération des perturbations externes et d'observations (des mesures).

### 3.2.1.2 À référence globale

La détermination de la position instantanée des objets en général, et des véhicules aériens terrestres et maritimes en particulier, sur la surface de la Terre est primordiale pour la navigation. Elle se fait à l'aide des systèmes de navigation globaux par satellites « *en : Global Navigation Satellite Systems (GNSS)* » [64]. À cet effet, on retrouve dans la littérature trois systèmes de navigation par satellites :

- GLONASS « *russe : Globalnaya Navigatsionnaya Sputnikovaya Sistema* » : un système de positionnement par satellites développé par la Fédération de Russie ;
- Galileo : projet européen de positionnement par satellites. Cependant, il est toujours en cours de développement ;
- GPS « *en : Global Positioning System* » : l'un des systèmes de positionnement par satellites le plus utilisé au monde. Ce système a été initialement développé à des fins militaires par les États-Unis d'Amérique [64]-[65]. Il est composé de trois systèmes :
  - Un système spatial, comprenant 24 satellites en orbites terrestres ;
  - Un système de réception, du côté des utilisateurs (récepteur GPS) ;
  - Un système de poursuite et de contrôle, constitué de stations terrestres.

Le GPS permet de déterminer la position des objets dotés d'un récepteur du signal GPS. Ce dernier capte les signaux d'au moins trois satellites, en théorie, et quatre satellites en pratique. Les signaux envoyés au récepteur terrestre sont alors datés, ceci pour calculer la pseudo-distance qui est proportionnelle à la vitesse de propagation des ondes électromagnétiques :

$$d_k^i = c\tau_k^i = \sqrt{(x_k^i - x_k^{GPS})^2 + (y_k^i - y_k^{GPS})^2 + (z_k^i - z_k^{GPS})^2} \quad i = 1, 2, 3, \dots, N \quad (3.4)$$

Où :

- $N$  : nombre de satellites vus par le récepteur GPS ;
- $c$  : célérité des ondes électromagnétiques, égale à la vitesse de la lumière ;
- $\tau_k^i$  : temps mis par le signal électromagnétique, venant de l'i<sup>ème</sup> satellite, jusqu'au récepteur GPS ;
- $x_k^i = [x_k^i \quad y_k^i \quad z_k^i]^T$  : coordonnées connues du i<sup>ème</sup> satellite ;

- $x_k^{GPS} = [x_k^{GPS} \quad y_k^{GPS} \quad z_k^{GPS}]^T$  : coordonnées du récepteur GPS.

La résolution d'un ensemble d'équations à trois inconnues, coordonnées de l'objet (x, y, z), nécessite au minimum trois équations (trois satellites). Cependant, le décalage temporel entre les satellites et le récepteur « *en : time offset* », nécessite un quatrième satellite. Pour plus de détails sur le principe de fonctionnement d'un GPS et la formulation mathématique de ce genre de problème, le lecteur peut se référer à [65]-[66].

### 3.2.2 Capteurs extéroceptifs

Ils constituent l'ensemble des capteurs embarqués permettant au véhicule de percevoir son environnement extérieur et de mesurer la position relative de chaque obstacle. Comme mentionné précédemment, tout véhicule intelligent ou autonome doit impérativement être doté de ce genre de capteurs, ceci pour se repérer et localiser les différents obstacles présents dans son environnement. Il existe trois principales catégories de capteurs extéroceptifs :

- Capteurs sonores et radios (ultrason, SONAR « *en : SOund Navigation And Ranging* », RADAR « *en : RAdio Detection And Ranging* », etc.) ;
- Capteurs optiques ; Lidar/Ladar « *en : Light/LAser Detection And Ranging* » ;
- Capteurs de stéréovision : caméras ; monoculaire, binoculaire, infrarouge.

Ci-après, une brève description des deux dernières catégories, regroupant les capteurs extéroceptifs utilisés dans le cadre de ce projet.

#### 3.2.2.1 Télémètre optique

Un télémètre optique est constitué de deux principales composantes : un émetteur laser « *en : Light Amplification by Stimulated Emission of Radiation* » ; et un dispositif à transfert de charge (DTC) du type *CCD* « *en : Charge-Coupled Device Complementarity* », ou *CMOS* « *Metal-Oxide-Semiconductor* ». Son fonctionnement consiste principalement à envoyer des signaux électromagnétiques datés, et à capter leurs échos afin de mesurer la distance séparant le télémètre optique de l'obstacle. Cela peut être réalisé avec une bonne résolution spatiale. Le principe de la télédétection est illustré dans la figure 3-2 :

La détermination de la distance «  $d$  » entre le télémètre optique et l'obstacle est faite en ayant recours à la théorie de la triangulation, ce qui donne :

$$d = a \left( 1 + \frac{b}{l \sin(\alpha)} \right) \quad (3.5)$$

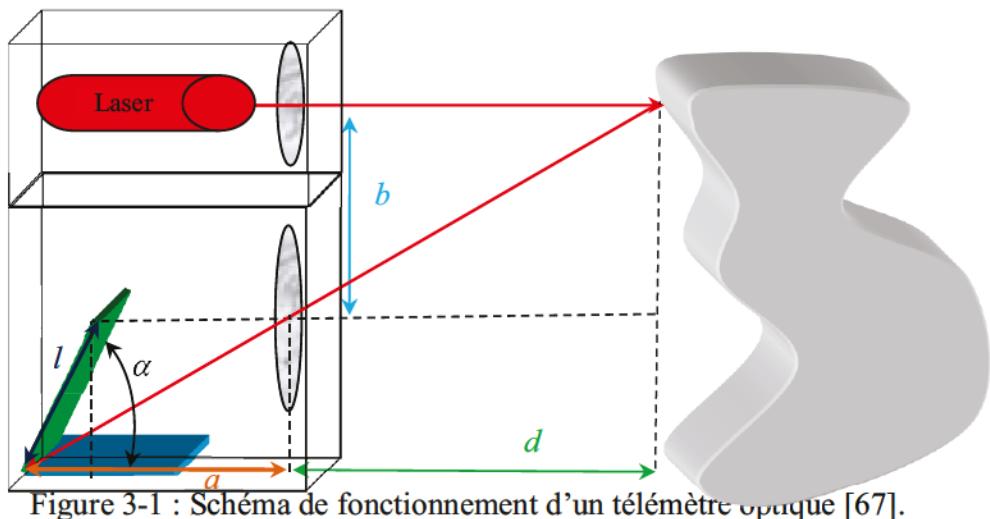


Figure 3-1 : Schéma de fonctionnement d'un télémètre optique [67].

### 3.2.2.2 Lidar

Le fonctionnement du lidar/ladar est un cas particulier de celui du télémètre optique. En effet, grâce à un émetteur — composé d'une ou plusieurs diodes laser — le lidar envoie, sur un obstacle, un signal lumineux très intense [68]. Une partie du signal réfléchi par l'obstacle est alors captée par le récepteur laser. Ce récepteur est composé d'une ou de plusieurs photodiodes sensibles aux longueurs d'ondes laser.

Après la réception du signal réfléchi, on détermine la distance «  $d$  » entre l'obstacle et le lidar au moyen de décalage en phase de l'écho comme suit :

$$d = \frac{c}{4\pi f} \Delta\varphi \quad (3.6)$$

Où :

- $\Delta\varphi$  : déphasage en phase de l'onde réfléchie ;
- $c$  : vitesse de la lumière dans le vide ;
- $f$  : fréquence de l'onde émise par le lidar (diode laser).

On peut aussi exprimer le déphasage de l'onde réfléchie comme suit :

$$\Delta\varphi = 2\pi f \left( \frac{2d}{c} \right) \quad (3.7)$$

Dans de bonnes conditions météorologiques, il est possible de mesurer l'emplacement des obstacles à moins de 200 m de l'objectif avec une précision centimétrique. Cependant, le faisceau lumineux est susceptible d'être absorbé, réfléchi et/ou dispersé. Par conséquent, la qualité de la réception du signal dépend essentiellement du milieu de propagation de l'onde lumineuse (les propriétés du milieu de propagation et les propriétés de la surface de l'obstacle).

### 3.2.2.3 Caméra

Les systèmes optiques utilisent différents types de caméras : standard RGB « *en : Red, Green, Blue* », infrarouge, etc. Ces caméras acquièrent des séquences d'images de l'environnement du véhicule afin de détecter les bordures et les lignes de marquage au sol, les panneaux de signalisation, les piétons, les plaques d'immatriculation des véhicules, etc. Cependant, elles présentent aussi certaines problématiques à surmonter comme le grand nombre de caméras à mettre dans le véhicule hôte pour que ce dernier aperçoive la totalité de son environnement. Cela induit un besoin important en matière de ressources de calcul, notamment nécessaires pour le traitement des images acquises par toutes ces caméras embarquées.

## 3.3 Capteurs embarqués et zones à couvrir autour d'un véhicule

Le type et le nombre des capteurs à utiliser déterminent les capacités, la précision, la robustesse et le taux d'autonomie du véhicule intelligent. Ainsi, pour répondre aux besoins primaires (la localisation et la navigation) d'un véhicule semi-autonome, il faut le doter au minimum :

- d'un système de positionnement à référence globale « GPS », pour déterminer la position absolue du véhicule dans un système d'information géographique (surface de la Terre) ;
- d'un système de localisation à référence locale (une centrale inertielle) pour localiser le véhicule à courtes durées, entre deux réceptions ou rafraîchissement du signal GPS, et cas de perte du signal GPS dans certains milieux urbains comme les tunnels ;
- d'au moins une caméra pour la détection des lignes de marquage au sol, des panneaux de signalisation, etc. ;
- d'un lidar pour la détection des différents obstacles, mobiles et immobiles, présents tout autour du véhicule, afin que ce dernier puisse reconstruire et comprendre son environnement, ceci pour éviter toutes collisions.

Cependant, la redondance des informations des capteurs est un facteur important, voire indispensable, pour améliorer la robustesse des algorithmes embarqués dans un véhicule intelligent. Cette redondance permettra ainsi au véhicule de surmonter des états de mauvais fonctionnement, causés notamment par la défaillance d'un ou de plusieurs capteurs. Néanmoins, assurer cette redondance peut se montrer onéreux sur plusieurs plans, notamment sur le plan financier. Cette situation revient à juger d'un compromis entre la fiabilité d'un côté, et le coût des équipements embarqués de l'autre.

*« Fiabilité et robustesse de fonctionnement du véhicule autonome VS coût des capteurs et des équipements embarqués ».*

La figure suivante montre les différentes zones associées aux différents ADAS, et donc à couvrir par les capteurs.

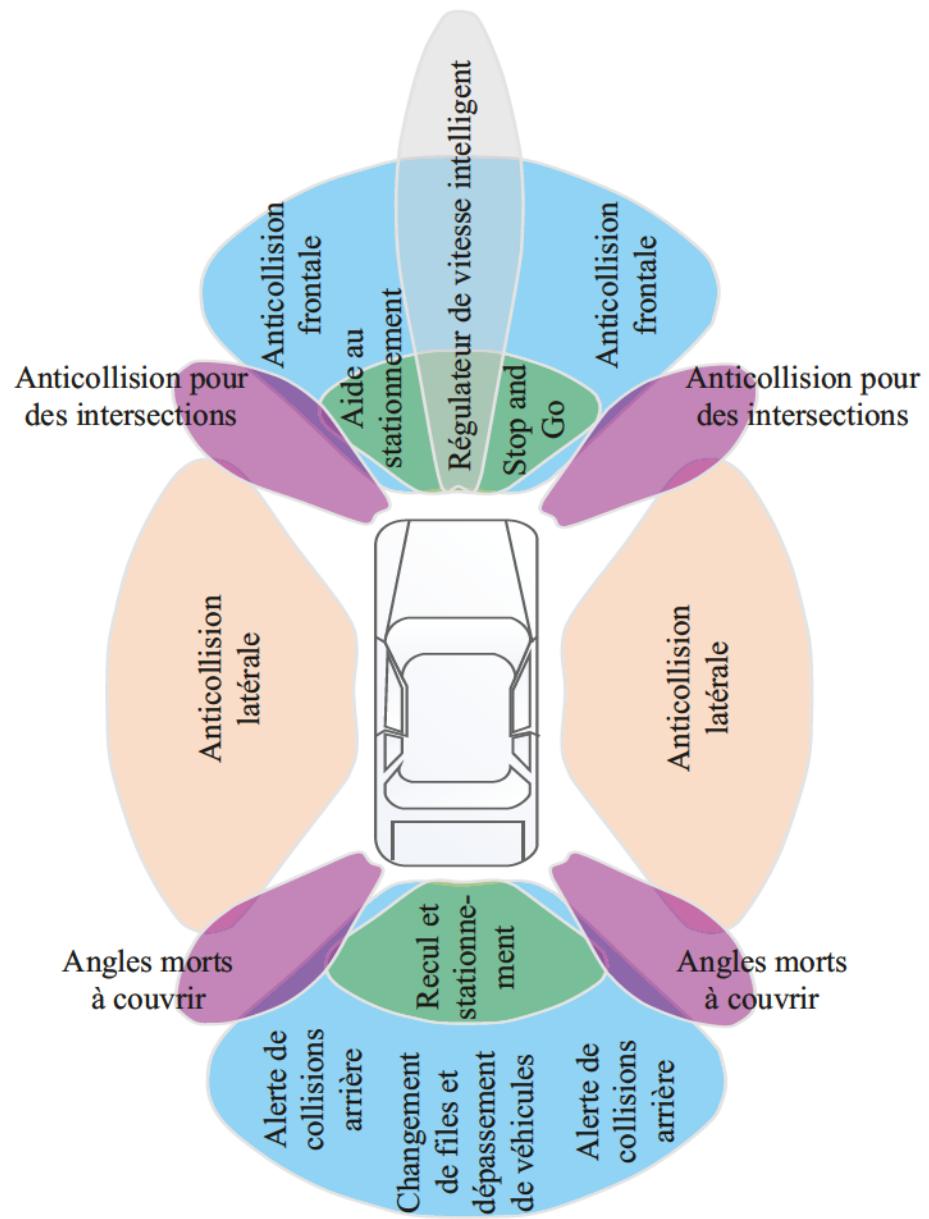


Figure 3-2 : Zones de couverture pour les différentes applications automobiles [70].

### 3.4 Conclusion

Dans ce chapitre, nous avons présenté, principalement et brièvement, les capteurs embarqués utilisés dans le cadre de ce projet, leur principe de fonctionnement et leurs limitations. L'usage de ces capteurs est indispensable dans le domaine véhiculaire. Chacun d'eux présente des

lacunes et des avantages plus ou moins importants les uns par rapport aux autres. Voici quelques exemples des limitations possibles de ces capteurs :

- La centrale inertielle présente un bon fonctionnement à courtes distances et à des fréquences d'acquisitions élevées. Cependant, les erreurs systématiques (déviations) propres aux accéléromètres et aux gyromètres engendrent une erreur cumulative sur les grandeurs mesurées (les accélérations et les vitesses angulaires) qui fait diverger l'estimation de la position au fil du temps ;
- Les lacunes des centrales inertielles peuvent être surmontées par l'utilisation du GPS, et cela même avec un taux de rafraîchissement assez faible. Un des principaux inconvénients du GPS est le nombre de satellites nécessairement visibles au récepteur du signal GPS. Cela dégrade son fonctionnement surtout dans certains milieux urbains, tels que les tunnels, etc. Et cela contrairement aux centrales inertielles qui ne présentent pas ce genre de contraintes ;
- Lidar : ce capteur arrive à scanner son environnement lointain. Toutefois, il rencontre certaines difficultés : à faible portée à cause de la vitesse des ondes électromagnétiques qui requièrent souvent des horloges très précises ; à longue portée à cause du nombre limité de faisceaux laser par degré du champ de détection ;
- Caméra : l'usage des caméras dans des véhicules intelligents est évidemment incontournable. Cependant, la puissance de calcul nécessaire pour le traitement d'image en temps réel est l'une des contraintes à prendre en considération. Elle dépend essentiellement du type de l'application : détection des lignes, suivi des véhicules et des piétons, estimation de la distance séparant le système optique (plusieurs caméras) et les objets, etc.

Ainsi, la nouvelle plateforme RT-LAB/Pro-SiVIC pourra aider : à déterminer le meilleur emplacement des capteurs sur le véhicule ; à analyser le risque des zones d'ombre — zones non couvertes par les capteurs extéroceptifs — ; et à développer de nouvelles méthodes de fusion de données des capteurs embarqués.

# Chapitre 4

## Fusion de données des capteurs embarqués

### 4.1 Introduction

Les nouvelles technologies, présentes sur le marché, offrent une panoplie de capteurs embarqués, dont le rôle est, essentiellement, de mesurer des grandeurs physiques. La miniaturisation de ces derniers ainsi que leur prix relativement bas permettent de nos jours de déployer plusieurs types de capteurs pour servir une seule application [71]. Les bruits de mesure et les asymétries de capteurs, dues notamment aux procédés de fabrication, dégradent leur fonctionnement. Néanmoins, l'utilisation de plusieurs capteurs peut y remédier. Ainsi, la combinaison des différentes informations, venant de chaque capteur, est un autre défi qu'il faut relever et qui nécessite des algorithmes de traitement de signal récursifs, surtout pour des applications en temps réel. Ce genre d'algorithmes est connu sous le nom d'algorithmes de «*fusion de données*». L'origine de ces algorithmes remonte aux années 60 [72]. Leurs mises en œuvre et leurs applications en temps réel sont dues :

- À la puissance de calcul et à la mémoire des ordinateurs de nos jours : certains algorithmes de traitement de signal demandent énormément de puissance de calcul, surtout ceux qui nécessitent l'inverse d'une ou de plusieurs matrices carrées d'une grande dimension ;
- Aux méthodes de traitement de signal : de plus en plus nombreuses et complexes, elles permettent la reconstruction de l'ensemble des informations du système étudié. L'exploitation du modèle mathématique de ce dernier, combinée aux mesures de capteurs, permet de réaliser cette reconstruction.

La fusion de données offre un cadre très adéquat pour le traitement de signal des systèmes multi-capteur. Dans le reste de ce chapitre, le principe de la fusion et les méthodes les plus utilisées, avec des ordres de complexité acceptables pour le traitement en temps réel, sont décrits.

Le but principal de ce chapitre est de présenter un certain nombre d’algorithmes de traitement de signal, à des fins d’estimation du vecteur d’état des modèles linéaires et non linéaires dans l’espace d’état. Le chapitre présente également différentes architectures de fusion de données.

## 4.2 Observateur d’état des systèmes dynamiques

Un observateur estime le vecteur d’état d’un système dynamique, évoluant au cours du temps. L’estimation est réalisée à partir des mesures de capteurs fournissant des informations partielles sur le vecteur d’état du système. L’objectif de cette estimation est ainsi de pallier au manque d’informations sur le système. Des variantes de mise en œuvre existent dans les deux domaines d’espace d’état (continu et discret).

Soit un système linéaire à temps invariant, dont la représentation d’état en continu est la suivante :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ z(t) = Cx(t) \end{cases} \quad (4.1)$$

Où  $x(t)$ ,  $u(t)$  et  $z(t)$  sont, respectivement, les vecteurs d’état, de commande et d’observation du système. A, B et C sont, respectivement, les matrices d’état (de transition), de commande et d’observation du modèle.

Le schéma fonctionnel du système linéaire à temps invariant en boucle ouverte est alors donné par la figure suivante :

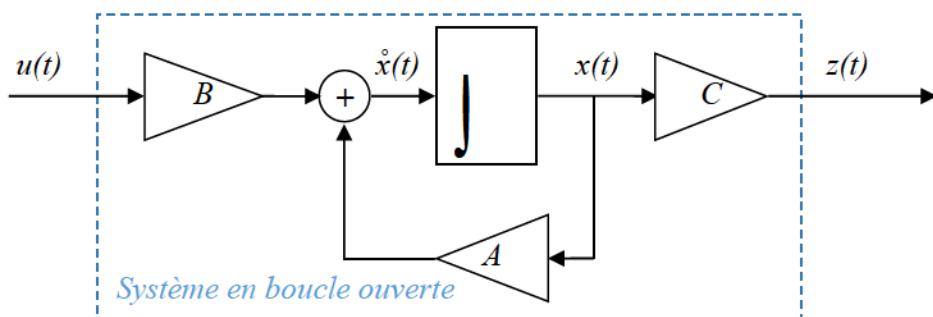


Figure 4-1 : Schéma fonctionnel d’un système linéaire en boucle ouverte

#### 4.2.1 Définition de la condition d'observabilité d'un système linéaire

Un système linéaire est dit observable, si pour tout état  $x_0 \in \mathbb{R}^n$  ( $\mathbb{R}$  est l'ensemble des nombres réels), il existe un temps fini «  $T_f$  » et une commande admissible «  $u(t) : [0, T_f] \rightarrow U$  » tel que la connaissance du vecteur d'observation à l'instant  $t \in [0, T_f]$  «  $z(t)$  » permet de déterminer l'état initial «  $x_0$  » [73]. Kalman a pu mettre au point des relations algébriques sur l'observabilité et la contrôlabilité des systèmes linéaires. Ci-après, le lemme de Kalman sur l'observabilité de ces derniers.

*Lemme de Kalman sur l'observabilité :* Un système linéaire est dit observable, si et seulement si la matrice d'observabilité de Kalman, notée «  $W_{obs}$  », est de rang plein. Autrement dit, son rang est égal à «  $n$  ». Alors, le couple  $(A, C)$  est dit observable tel que :

$$W_{obs} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.2)$$

Où : «  $n$  » est l'ordre du système, défini par le nombre de variables d'état du système (4.1).

#### 4.2.2 Observateur de Luenberger

L'observateur de Luenberger est un observateur d'état des systèmes linéaires, dont le rang de la matrice d'observabilité est égal à l'ordre du système. Sa forme générale est définie par (4.3) :

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(z(t) - \hat{z}(t)) \\ \hat{z}(t) = C\hat{x}(t) \end{cases} \quad (4.3)$$

Il existe une matrice de gain d'observation «  $K$  », de sorte que la matrice d'état de l'observateur est une matrice Hurwitz (les valeurs propres de la matrice Hurwitz sont à parties réelles négatives) [73].

$$\begin{cases} \dot{\hat{x}}(t) = (A - KC)\hat{x}(t) + [B \quad K] \begin{bmatrix} u(t) \\ z(t) \end{bmatrix} \\ \hat{z}(t) = C\hat{x}(t) \end{cases} \quad (4.4)$$

La dynamique de l'observateur est conditionnée par la matrice de gain d'observation «  $K$  ». Les valeurs de cette dernière peuvent être calculées par placement de pôles, en imposant une certaine dynamique désirée à l'observateur. Elles peuvent également être calculées de manière optimale, en minimisant l'erreur d'observation  $e(t) = x(t) - \hat{x}(t)$ .

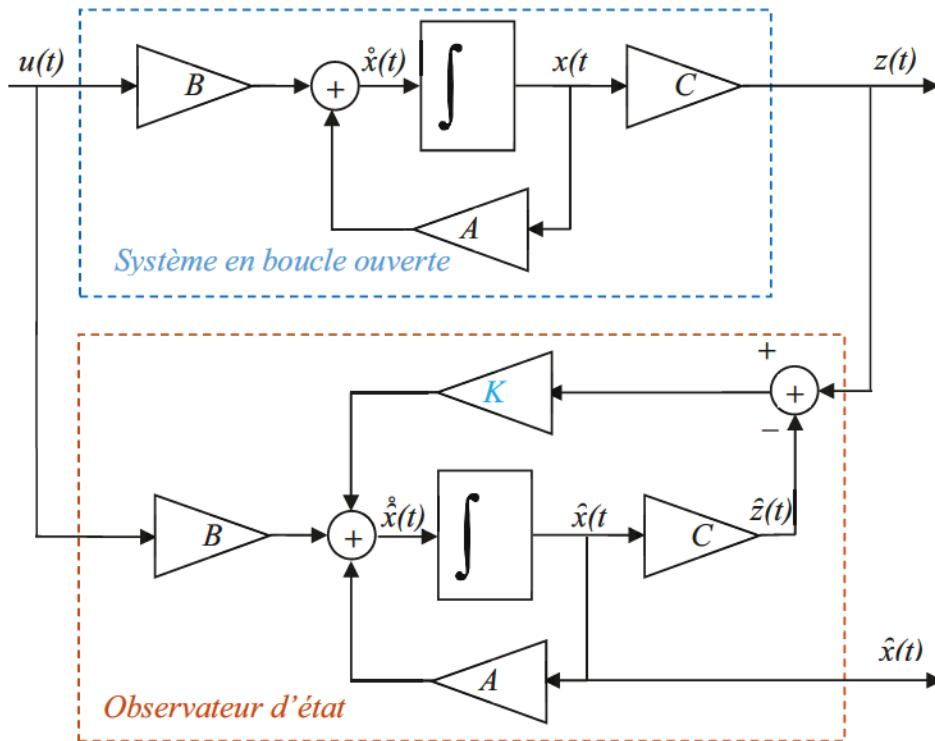


Figure 4-2 : Schéma fonctionnel du modèle et de l'observateur d'état dans le domaine continu

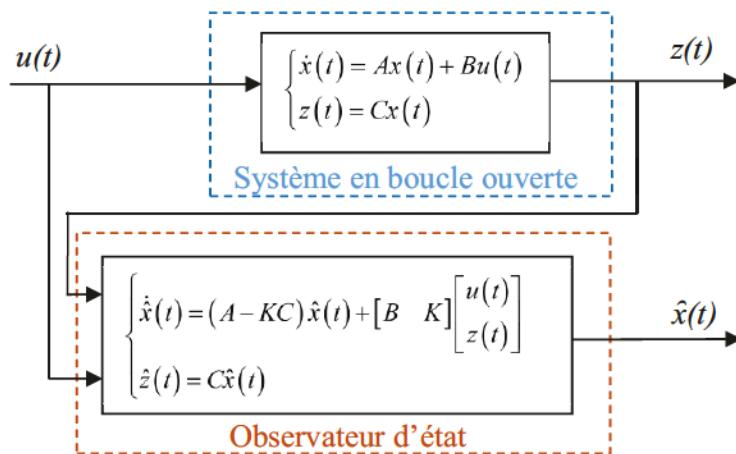


Figure 4-3 : Schéma fonctionnel de l'observateur d'état sous forme matricielle dans le domaine continu

### 4.2.3 Filtre de Kalman

Bien que la généralisation de l'observateur d'état soit faite par Luenberger pour les systèmes linéaires, son principe a été initialement introduit par Kalman [72]. Grâce à l'aspect optimal du filtre de Kalman (KF) et surtout à sa récursivité et à son faible coût de calcul, le KF reste indéniablement l'un des algorithmes de filtrage et de traitement de signal les plus utilisés pour la fusion de données [74]-[75]. Il est conçu pour filtrer les données entachées de bruits additifs gaussiens, cas des mesures des capteurs embarqués sur les véhicules intelligents, et de fusionner des informations asynchrones et non corrélées venant de plusieurs sources. Le schéma suivant résume le fonctionnement du filtre de Kalman [76].

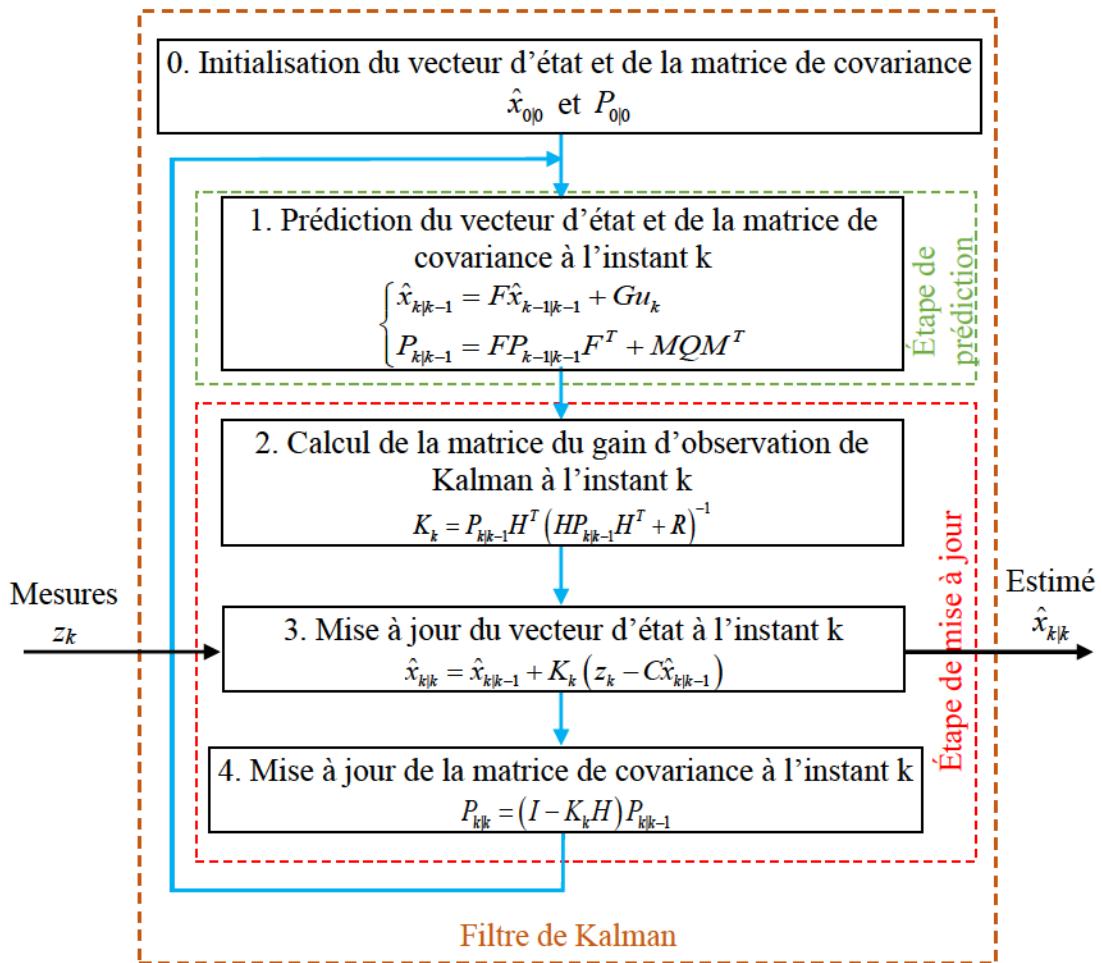


Figure 4-4 : Algorithme du filtre de Kalman pour les systèmes linéaires à temps invariant [76].

Où  $F$ ,  $G$ ,  $H$ ,  $M$  sont respectivement les matrices d'état, de commande, d'observation et d'interaction du bruit d'état sur le modèle.  $P$ ,  $Q$  et  $R$  sont respectivement les matrices de

variance de l'erreur d'estimation, du bruit d'état et du bruit de mesure des capteurs.  $\hat{x}$ ,  $u$  et  $z$  sont respectivement les vecteurs d'état de KF, de commande et d'observation d'état du modèle.

#### 4.2.4 Filtre de Kalman étendu

L'application du filtre de Kalman aux modèles des systèmes non linéaires nécessite la linéarisation de ces derniers autour d'un point de fonctionnement. Ceci en vue de construire, au préalable, un modèle linéaire autour de ce point à chaque instant  $k$ . Pour ensuite mettre à la disposition du nouveau KF, à chaque instant  $k$ , les différentes matrices (d'état, de commande et d'observation) nécessaires pour la mise à jour de ses deux matrices (de covariance et de gain d'observation), qui sont à leur tour requises pour la mise à jour du vecteur d'état. Ainsi, le KF devient l'EKF « *en : Extended Kalman Filter* » [74]-[78]. Soit le système non linéaire suivant :

$$\begin{cases} x_{k+1} = f(x_k, u_k, w_{m,k}) \\ z_k = h(x_k) + w_{s,k} \end{cases} \quad (4.5)$$

Tels que :  $x_k$ ,  $u_k$ ,  $z_k$ ,  $w_{m,k}$  et  $w_{s,k}$  sont, respectivement, les vecteurs d'état, de commande, d'observation, des perturbations exogènes agissant sur la dynamique du modèle et des bruits de mesure ;  $f(x_k, u_k, w_{m,k})$  et  $h(x_k)$  sont, respectivement, les fonctions d'état et d'observation.

L'approximation du premier ordre du modèle non linéaire en modèle linéaire se fait alors en calculant la Jacobienne de chaque fonction non linéaire autour du point de fonctionnement du modèle :

$$\begin{cases} x_{k+1} = \frac{\partial f(x_k, u_k, w_k)}{\partial x_k} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_k=\bar{w}_k}} x_k + \frac{\partial f(x_k, u_k, \bar{w}_{m,k})}{\partial u_k} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} u_k + \frac{\partial f(x_k, u_k, \bar{w}_k)}{\partial w_k} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k}} w_{m,k} \\ z_k = \frac{\partial h(x_k, w_{s,k})}{\partial x_k} \Bigg|_{\substack{x_k=x_{Pf} \\ w_{s,k}=\bar{w}_{s,k}}} x_k + w_{s,k} \end{cases} \quad (4.6)$$

Alors, les différentes matrices du modèle sont définies comme suit.

$$F \left( x_k, u_k, w_{m,k} \Big|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} \right) = \frac{\partial f(x_k, u_k, w_{m,k})}{\partial x_k} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}}, \quad G \left( x_k, u_k, w_{m,k} \Big|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} \right) = \frac{\partial f(x_k, u_k, w_{m,k})}{\partial u_k} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} \\ M \left( x_k, u_k, w_{m,k} \Big|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} \right) = \frac{\partial f(x_k, u_k, w_{m,k})}{\partial w_{m,k}} \Bigg|_{\substack{x_k=x_{Pf} \\ u_k=\bar{u}_k \\ w_{m,k}=\bar{w}_{m,k}}} \text{ et } H \left( x_k \Big|_{x_k=x_{Pf}} \right) = \frac{\partial h(x_k)}{\partial x_k} \Bigg|_{x_k=x_{Pf}}$$

Elles sont, respectivement, les matrices d'état, de commande, d'interaction des perturbations externes sur le système et d'observation autour du point de fonctionnement du modèle.

Maintenant que le problème de la non-linéarité est résolu, grâce à la linéarisation autour du point de fonctionnement du système, le filtre de Kalman peut être mis en œuvre en apportant les modifications nécessaires dans les deux étapes de prédiction et de mise à jour. Cette nouvelle version du KF, adaptée aux modèles non linéaires, est connue sous le nom d'EKF (filtre de Kalman étendu). Ce dernier utilise le modèle non linéaire pour la prédiction des vecteurs d'état et d'observation, et le modèle d'approximation pour la prédiction et la mise à jour des matrices de covariance et du gain d'observation, dont l'algorithme est donné ci-après.

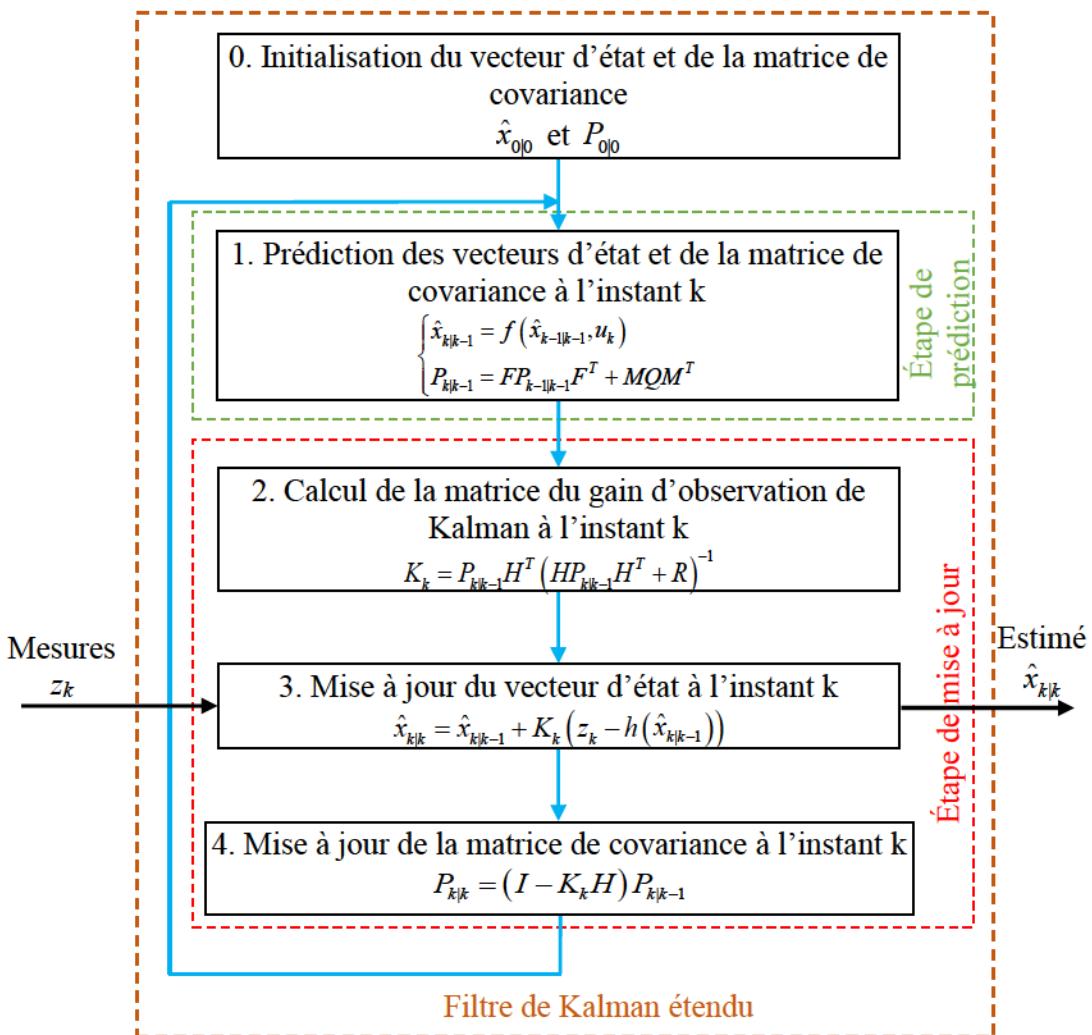


Figure 4-5 : Algorithme du filtre de Kalman étendu pour l'estimation du vecteur d'état des modèles non linéaires [76].

#### 4.2.5 Filtre de Kalman non parfumé

Le fait d'approximer le système non linéaire par un modèle linéaire, autour de son point de fonctionnement, induit parfois une dégradation du fonctionnement du filtre de Kalman étendu. Cette dégradation cause souvent l'instabilité du système, notamment dans le cas où l'estimée du vecteur d'état de l'EKF est utilisée à des fins de contrôle de la dynamique du système. La validité des estimées de l'EKF, dans le cadre de la commande des systèmes, est souvent conditionnée par le degré de sévérité des non-linéarités du système [79]-[81]. Cette limitation de l'EKF est due au fait que les moments stochastiques du système non linéaire supérieurs à un ne sont pas pris en compte. Pour y remédier, certains travaux, notamment [79]-[80], ont recours à la transformée non parfumée «en : *Unscented Transform (UT)*», pour mieux estimer les transformations non linéaires des systèmes fortement non linéaires.

##### 4.2.5.1 La transformation non parfumée

L'UT permet de calculer la moyenne et la covariance d'une variable aléatoire ayant subi une transformation non linéaire. Ce calcul se fait à l'aide d'un nombre fini d'échantillons, appelés les points sigma, avec beaucoup de précision et cela jusqu'au deuxième ordre de série de Taylor [79] et [85]. La détermination de ces points sigma se fait comme suit :

Soient  $X$  une variable aléatoire de dimension  $n$ , de moyenne  $x$  et de covariance  $P_x$ ,  $f(*)$  une fonction non linéaire, et  $Z$  l'image de  $X$  à travers  $f(*)$ . Le calcul des caractéristiques stochastiques de  $Z$ , résultat de la transformation non linéaire, se fait après la détermination des valeurs des points et des vecteurs sigmas ( $\omega_i$  et  $\chi_i$ ) qui sont définis comme suit :

$$\begin{cases} \chi_0 = x \\ \chi_i = x + \left( \sqrt{(n+\lambda)P_x} \right)_i, \quad i = 1, \dots, n \\ \chi_i = x - \left( \sqrt{(n+\lambda)P_x} \right)_i, \quad i = L+1, \dots, 2n \\ \omega_0 = \lambda / (n+\lambda) \\ \omega_i = (2(n+\lambda))^{-1}, \quad i = 1, \dots, 2n \end{cases} \quad (4.7)$$

Où :

- $\lambda = 3 - n$ , pour une distribution gaussienne ( $X$ );

- $\left(\sqrt{(n+\lambda)P_x}\right)_i$  est la  $i^{\text{ème}}$  colonne de la racine carrée de la matrice  $(n+\lambda)P_x$ .

La moyenne et la covariance de  $Z$  sont alors définies comme étant la moyenne et la covariance des échantillons pondérés (des points sigmas) [80] tels que :

$$\begin{cases} Z_i = f(\chi_i) \\ z \approx \sum_{i=0}^{2n} \omega_i Z_i \\ P_z \approx \sum_{i=0}^{2n} \omega_i (Z_i - z)(Z_i - z)^T \end{cases} \quad (4.8)$$

La figure suivante présente le diagramme de la transformée non parfumée donnant une vue générale et résumant l'ensemble de ses étapes.

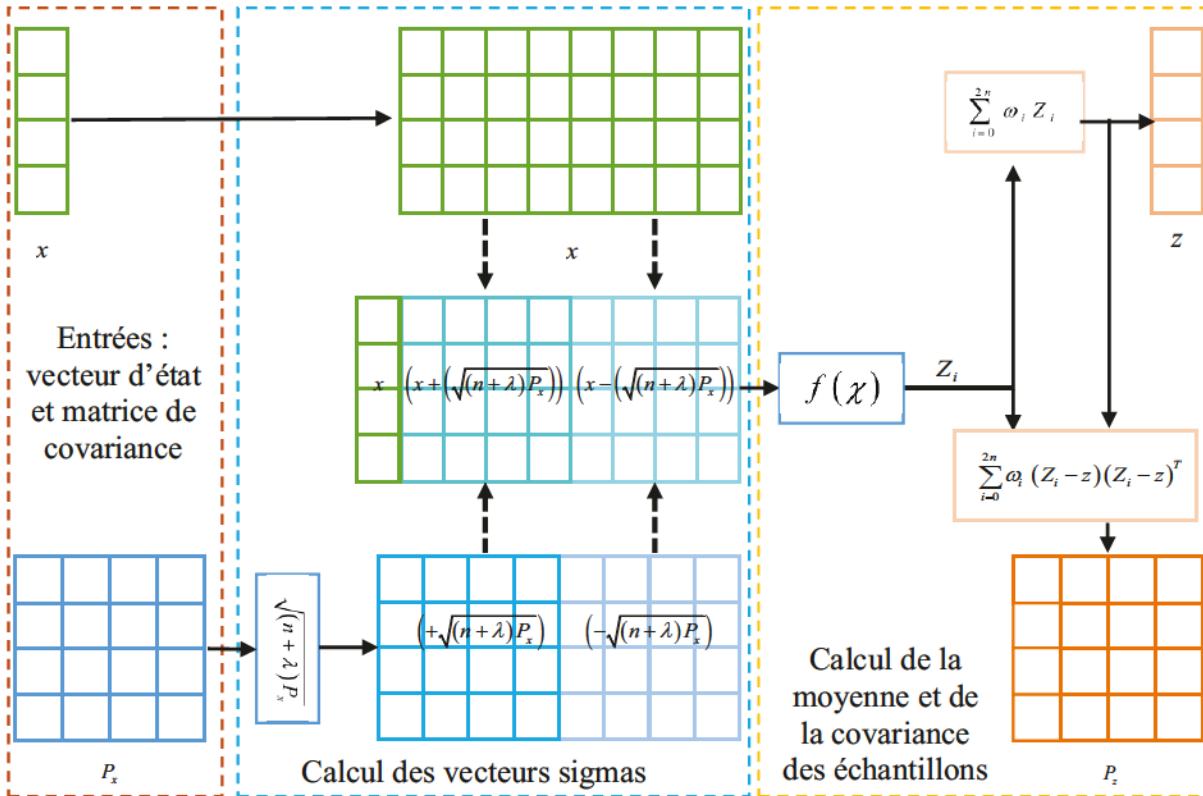


Figure 4-6 : Schéma fonctionnel de la transformée non parfumée [80]

Le résultat de la combinaison des deux algorithmes, KF et UT, est le filtre de Kalman non parfumé « *en* : *Unscented Kalman Filter (UKF)* ». Bien que la première version de l'UKF soit lente par rapport à celle de l'EKF qui requiert le calcul de la racine carrée de la matrice de covariance, une nouvelle méthode est proposée dans [81] permettant ainsi d'éviter la factorisation de Cholesky de la matrice de covariance, par la mise à jour de la racine carrée de la matrice de covariance à chaque instant  $k$ . La figure suivante résume l'algorithme de l'UKF.

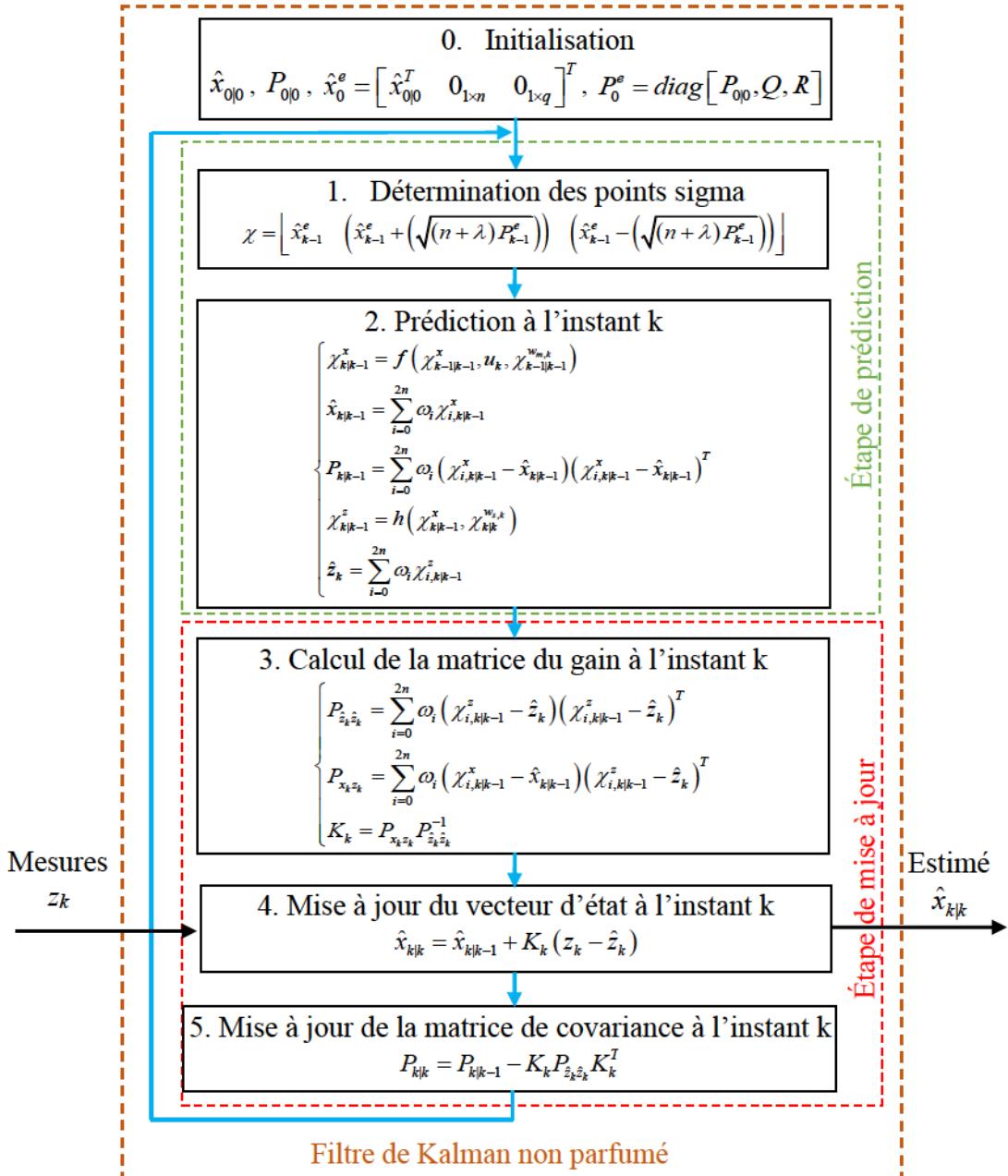


Figure 4-7 : Algorithme du filtre de Kalman non parfumé pour l'estimation du vecteur d'état des modèles fortement non linéaires

#### 4.2.6 Filtre d'information

La fusion de données des capteurs, entachées des bruits additifs gaussiens, est souvent effectuée par le filtre de Kalman ou une de ces dérivées qui permettent d'estimer les deux premiers moments stochastiques du vecteur d'état. Contrairement au KF, le filtre d'information « *en : information filter (IF)* » estime la matrice d'information  $\Omega$  à la place de la matrice de covariance  $P$  et le vecteur d'information  $\xi$  à la place du vecteur d'état  $x$ . La relation entre les deux estimés est donnée par la transformation bilatérale suivante.

$$\begin{cases} \Omega = P^{-1} \\ \xi = P^{-1}x \end{cases} \Leftrightarrow \begin{cases} P = \Omega^{-1} \\ x = \Omega^{-1}\xi \end{cases} \quad (4.9)$$

La figure suivante résume les deux étapes de prédiction et de mise à jour du filtre d'information étendu « *en : extended information filter (EIF)* ».

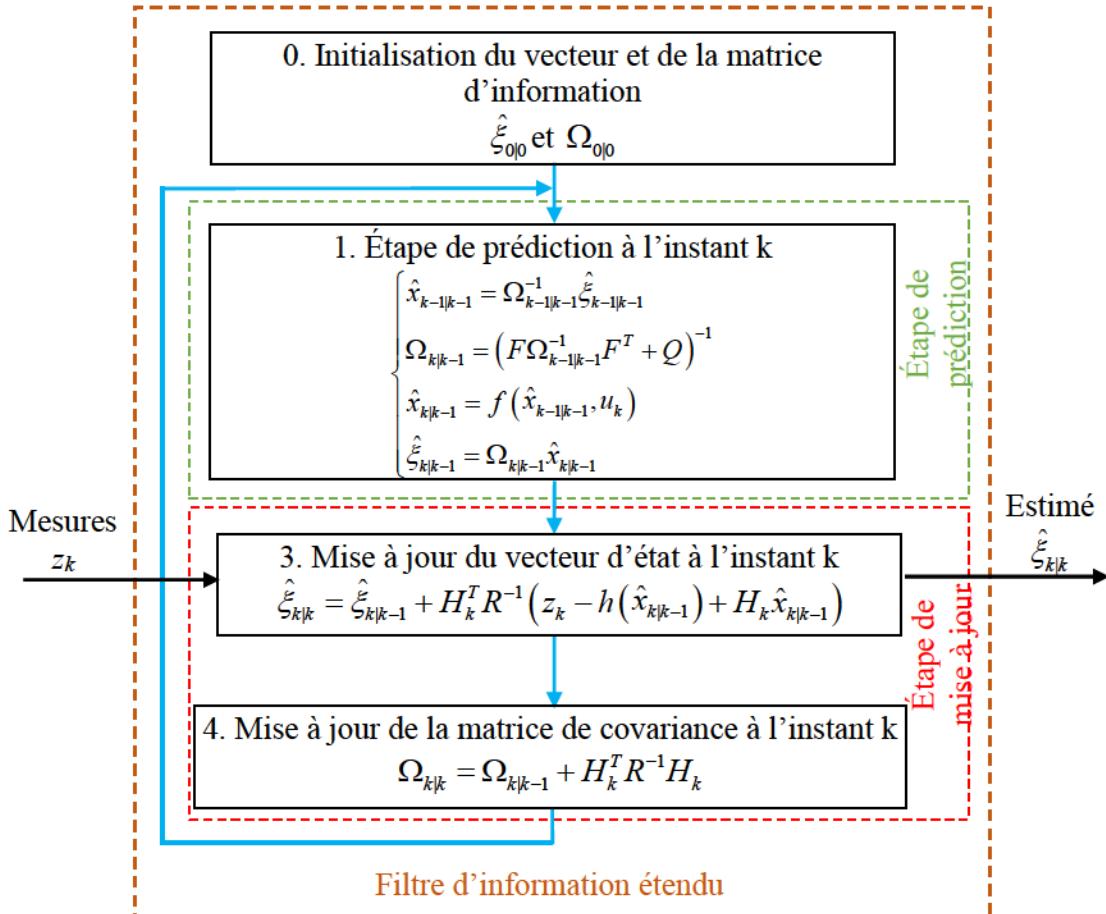


Figure 4-8 : Algorithme du filtre d'information étendu

Le choix d'utilisation de l'EIF ou de l'EKF est déterminé par le type d'application. La comparaison des deux filtres montre que les étapes de prédition et de mis à jour des deux filtres sont opposées en matière de complexité de calcul. Par exemple, l'étape de prédition de l'EKF ne nécessite aucune inversion matricielle, contrairement à celle de l'EIF qui requiert l'inverse de la matrice d'information pour la prédition du vecteur d'information. Inversement, l'étape de mise à jour de l'EIF est beaucoup plus simple que celle du l'EKF.

D'autres filtres plus gourmands, en matière de puissance de calcul, existent et sont conçus principalement pour traiter les bruits de mesure des capteurs, dont la forme de distribution est inconnue. Cas du filtre à particules qui utilise un nombre important de particules pour estimer les variables d'état du modèle.

### 4.3 Comparaison des trois filtres (EKF, UKF et EIF)

Dans le cadre de comparaison des trois dérivées du filtre de Kalman (EKF, UKF et EIF), appliqués aux systèmes non linéaires, nous avons créé un scénario monovéhiculaire sur la plateforme de simulation Pro-SiVIC. Le véhicule en question est doté d'un capteur proprioceptif à référence globale. Les mesures de ce dernier sont alors filtrées par un EKF, un UKF et un EIF à travers un modèle de mouvement d'objet en 2D, dont les vitesses longitudinale et angulaire sont constantes (Chapitre 7, section 7.4).

La figure 4-1 montre l'évolution des éléments des matrices de covariance des trois filtres, reliés à la position du véhicule. Les différents éléments de l'EKF, en rouge, sont égaux à ceux de l'EIF, en bleu. La figure 4-10 montre que même les erreurs quadratiques moyennes sur les axes des x et des y de ces deux filtres sont identiques. Ainsi, les deux figures confirment l'analogie des deux filtres (EKF et EIF) en matière d'estimation du vecteur d'état et de la matrice de covariance.

La figure 4-10 montre que l'erreur quadratique moyenne de l'UKF (sur les axes x et y) est bien inférieure à celles des de l'EKF et de l'EIF. La figure 4-9 montre également que les variances  $P(x)$  et  $P(y)$  de l'UKF, en noir, sont inférieures à celles de l'EKF et de l'EIF. La covariance  $P(x, y)$  illustre l'évolution de la corrélation existante entre les coordonnées du véhicule dans le temps. Elle montre ainsi que les coordonnées du véhicule sont fortement corrélées à chaque fois que ce

dernier se retrouve à se déplacer dans les deux axes (virages). L'UKF estime que cette corrélation est plus importante comparativement à l'EKF et à l'EIF. Par conséquent, les résultats de simulation confirment l'adaptabilité de l'UKF pour l'estimation d'état des systèmes non linéaires et donc sa supériorité étant donné la plus grande précision de ses résultats.

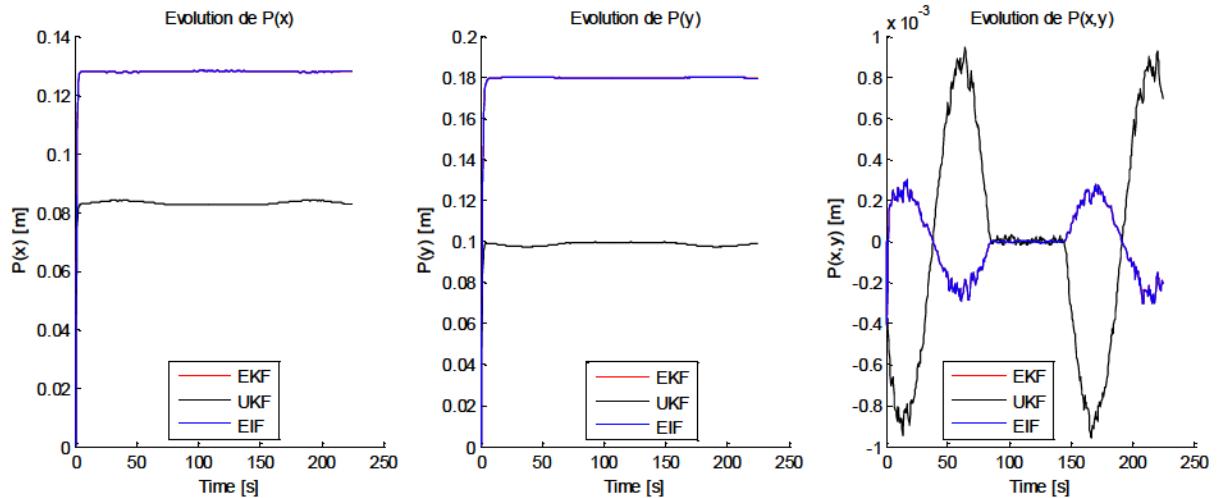


Figure 4-9 : Évolution des éléments des matrices de covariance des trois filtres (EKF, UKF et EIF) reliés à la position du véhicule

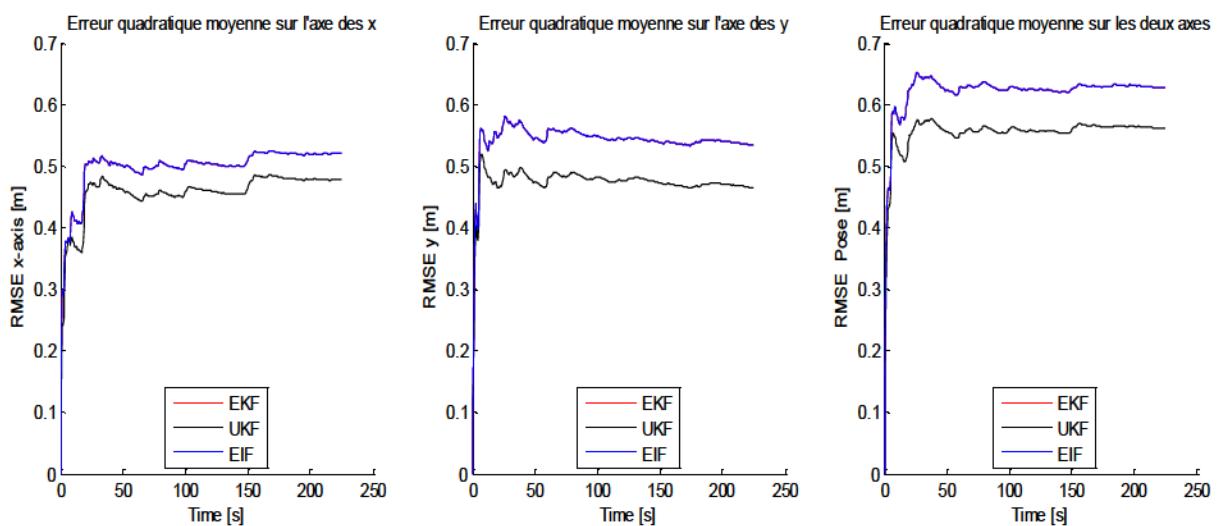


Figure 4-10 : Évolution de l'erreur quadratique moyenne des trois filtres (EKF, UKF et EIF)

Tableau 4-1 : Paramètres de la simulation

Paramètres de simulation	Valeurs
$n$	6
$Q$	$0.05^2 * \text{eye}(n)$
$R_{GPS} = \text{diag} [\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_v^2]$	$\text{diag} [2.5^2, 3.5^2, (3\pi/180)^2, (2/3.6)^2]$
$\lambda$	$3-n$
$f_{GPS}$	20 Hz

#### 4.4 Types d'architectures de fusion de données et simulation temps-réel

Comme indiqué ci-dessus, certains algorithmes de fusion de données demandent énormément d'espace mémoire et de puissance de calcul [82]-[84]. À cet effet, les architectures de fusion sont réparties sur trois catégories : centralisées, distribuées et hiérarchiques (hybride avec ou sans retour) [85]. Dans le cas des systèmes multi-capteur et multi-agents, cas des véhicules intelligents, le traitement des données se fait au niveau de chaque véhicule. Une fois que le traitement est terminé, les résultats sont communiqués aux autres véhicules. Ainsi, l'ensemble des nœuds de fusion des véhicules forme une architecture distribuée [85].

La configuration de ces architectures dépend essentiellement de la localisation des véhicules, de la méthode de communication et du type d'échange, envoi et/ou réception, des estimés locaux. Il est évident que les algorithmes de fusion, qui devraient être exploités au niveau des véhicules intelligents et connectés (en conduite collaborative), s'inscrivent dans la catégorie des architectures distribuées. Ces architectures sont constituées d'un ensemble de systèmes (micro-ordinateurs, super calculateurs, etc.) interconnectés et géographiquement dispersés [86]. Le choix des algorithmes de fusion se base essentiellement sur [85] :

- l'architecture du réseau : centralisée ou distribuée ;
- les contraintes de la communication (taux de communication, perte d'une partie ou de la totalité des paquets de données, capacité et les limites des nœuds de traitement, etc.) ;
- les méthodes de traitement et de contrôle implémentées dans chaque nœud.

Les figures suivantes illustrent les trois types d'architecture de fusion de données :

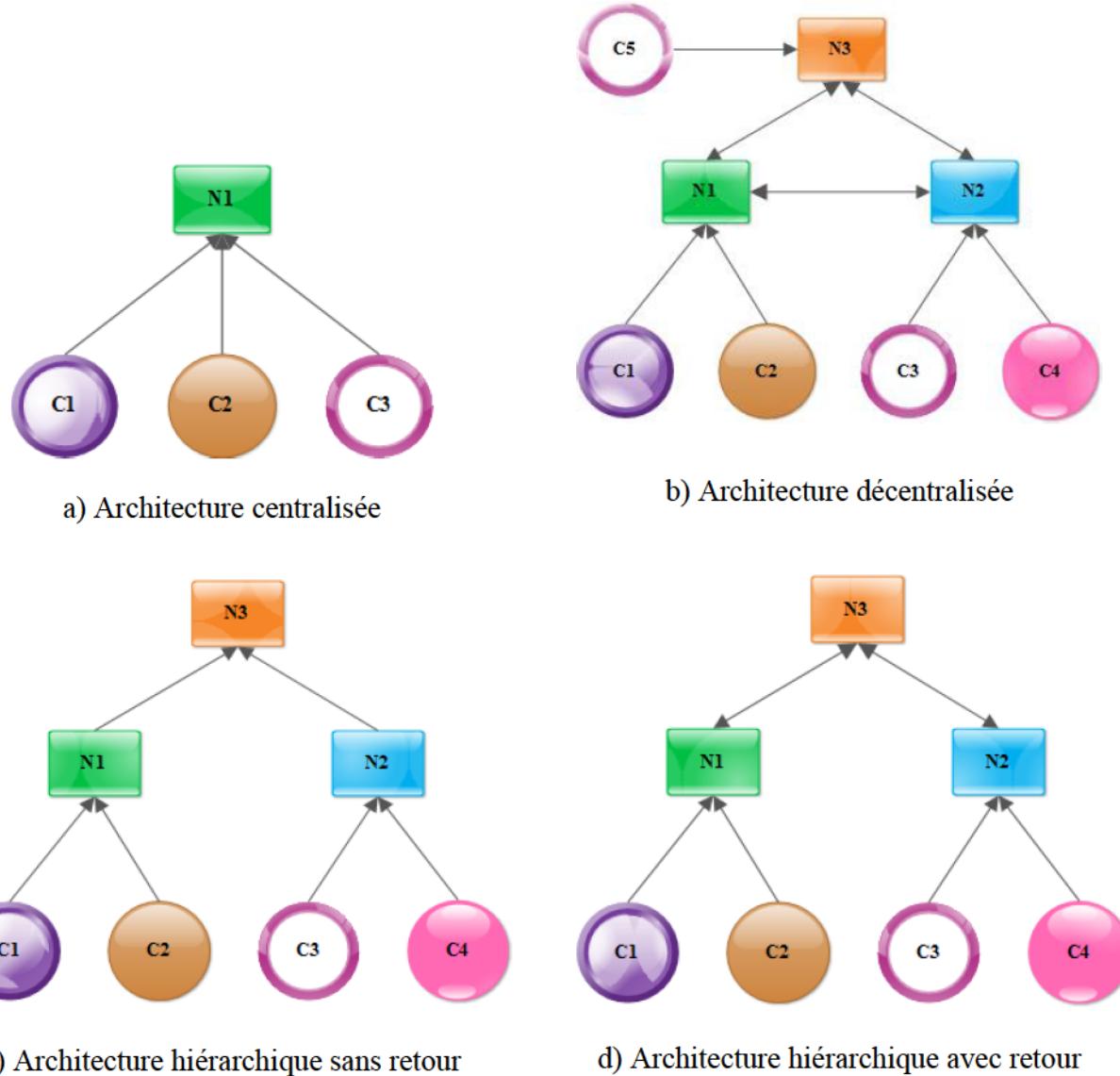


Figure 4-11 : Types d'architectures de fusion de données [85]

Où  $C_i$  et  $N_j$  sont respectivement le  $i^{\text{ème}}$  capteur et le  $j^{\text{ème}}$  nœud de fusion.

La structure et les besoins opérationnels (capteurs, algorithmes de fusion, taille des données, taux de communication des nœuds, etc.) du réseau créent des redondances d'informations. Ces dernières sont souvent affectées par des erreurs de poursuite en passant d'un nœud de fusion à un autre [85]. Pour remédier à ce problème, d'autres algorithmes de traitement des données sont développés sur le principe des graphes d'information et de la loi de Bayes [85]. Les deux figures ci-dessous présentent respectivement les graphes d'information des deux architectures centralisée et distribuée (parties « a » et « b » de la figure 4-11).

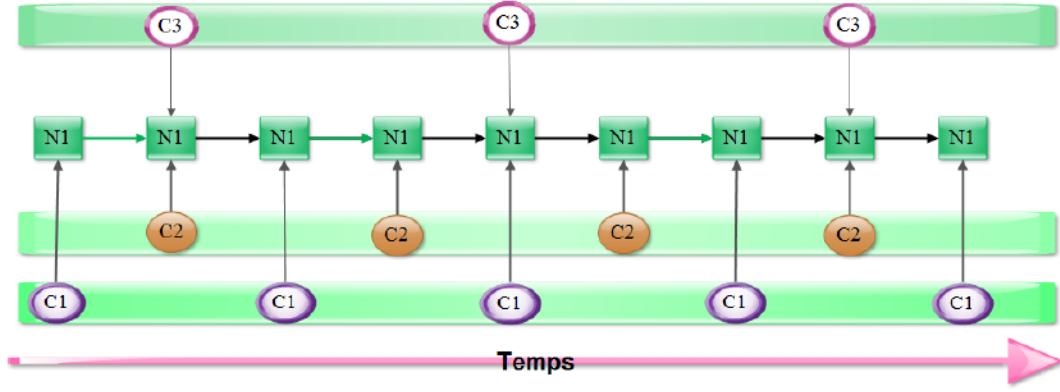


Figure 4-12 : Graphe d'information d'une architecture de fusion centralisée [85].

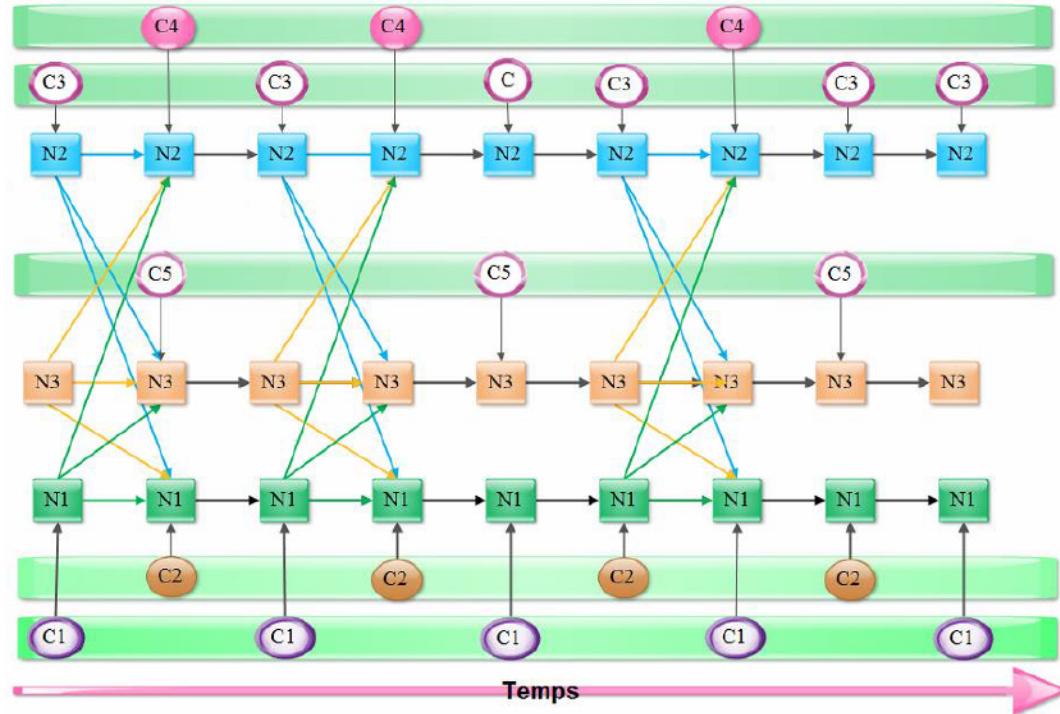


Figure 4-13 : Graphe d'information d'une architecture de fusion décentralisée [85].

Tels que :  $C_i$  et  $N_j$  sont respectivement le  $i^{\text{ème}}$  capteur et le  $j^{\text{ème}}$  nœud de fusion.

La représentation d'une architecture donnée à base des graphes d'information permet d'avoir une vue globale du processus de migration des estimations des nœuds dans le réseau. Ainsi, il est possible de localiser et d'identifier à chaque instant  $k$  les informations redondantes au niveau de chaque nœud [85].

La figure 4-10 représente le graphe d'information de l'architecture centralisée (partie « a » de la figure 4-9). Elle montre l'évolution de la procédure de fusion de données multi-capteurs dans le temps, sur une architecture centralisée. La non-corrélation des mesures des capteurs permet au nœud global (central) de fusionner et de garder une certaine consistance de ses estimations après la fusion de ces mesures. Cette fusion s'effectue à chaque fois que le nœud reçoit de nouvelles mesures des capteurs.

Cependant, la figure 4-11 représente le graphe d'information de l'architecture distribuée (partie « b » de la figure 4-9). Elle montre d'une part l'évolution de la procédure de fusion des capteurs dans chaque nœud de fusion du réseau, et d'autre part la migration des différentes estimations entre ces nœuds de fusion. Dans une première phase, chaque nœud fusionne les mesures de ses propres capteurs, qui sont non corrélées, puis transmet le résultat de la fusion au reste des nœuds. Dès la première transmission et la fusion des estimations des nœuds, ces derniers se retrouvent à partager les mêmes informations, comme l'illustre le graphe d'information (Figure 4-13). Les estimations des nœuds deviennent alors corrélées. La prochaine étape de fusion des estimations des nœuds (entre deux étapes successives; de nœud à nœud) nécessite le retrait des informations communes, qui se sont propagées durant les précédentes étapes de fusion. Si ces informations ne sont pas supprimées, les méthodes de fusion classiques, telles que le KF et ses dérivées qui sont développés pour la fusion des données non corrélées, tendent à diverger. À cet effet, de nouvelles méthodes de fusion de données sur des architectures distribuées sont développées et connues sous le nom des « *méthodes de fusion à haut niveau* ». On retrouve notamment la méthode de l'intersection des covariances et ses dérivées, la méthode des matrices d'information qui sont présentées en détail dans les chapitres 7 et 8.

## 4.5 Intérêts et enjeux des architectures distribuées

La complexité des applications de nos jours requiert des ressources de calcul importantes pour des applications en temps réel. Il est donc nécessaire d'avoir recours aux architectures distribuées/décentralisées. Cependant, bien que celles-ci offrent des avantages, elles présentent également un certain nombre d'enjeux à surmonter. Le tableau suivant résume les principaux avantages et enjeux des architectures distribuées.

Tableau 4-2 : Avantages et enjeux des architectures distribuées [87].

Avantages	Enjeux
<ul style="list-style-type: none"> <li>1. Réduction de la charge de traitement du nœud de fusion, en la répartissant sur plusieurs nœuds ;</li> <li>2. Étant donné que chaque nœud possède sa propre base de données, il n'est pas nécessaire de garder une grande base de données centralisée ;</li> <li>3. Communication : les informations circulant entre les nœuds se limitent aux estimées de ces derniers ;</li> <li>4. La réduction des charges de communication permet à l'ensemble des nœuds d'avoir accès rapidement aux résultats de la fusion ;</li> <li>5. Élimination des défaillances centrales (cas des architectures centralisées qui souffrent de ce mode de défaillance). Une architecture décentralisée est ainsi plus robuste ;</li> <li>6. Modularité : contrairement aux architectures centralisées, les architectures décentralisées sont très modulaires et très flexibles.</li> </ul>	<ul style="list-style-type: none"> <li>1. Architecture : partage de la responsabilité entre les nœuds de fusion ; accomplissent des objectifs pour lesquels chaque nœud est responsable ;</li> <li>2. Communication : connectivité et bande passante de chaque nœud du réseau, identification des sources des données et tri des résultats de fusion des nœuds locaux ;</li> <li>3. Algorithmes : choix des méthodes de fusion à utiliser par les nœuds pour fusionner efficacement les différentes informations, données brutes des capteurs et des estimées des nœuds, et pour sélectionner le mode de communication approprié (avec ou sans retour), associé au type de l'architecture choisie.</li> </ul>

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté le principe du filtrage et de la fusion des données à base du filtre de Kalman et ses dérivées. Chaque filtre présente certains avantages et certains

inconvénients. Cela conduit l'utilisateur à faire des compromis, selon le type d'application, entre la performance du filtre et la puissance de calcul (le taux de tolérance de l'erreur d'estimation, nature des bruits de mesures, etc.). Ainsi, ces méthodes de fusion de données permettent de remédier aux lacunes des capteurs embarqués, présentés dans le précédent chapitre, et de bien exploiter leurs performances.

La plupart de ces méthodes sont souvent gourmandes en matière de puissance de calcul, notamment dans le cas de la simulation de scénarios multivéhiculaires à des fins de planification et de conduite coopérative. La morphologie de ce genre de scénarios correspond à celle des architectures distribuées. Par conséquent, les algorithmes de fusion de données, qui seront mis en œuvre dans le cadre de ce projet, sont les algorithmes de fusion de données multi-capteurs qui reflètent le cas des scénarios multivéhiculaires à multi-capteurs tels que présentés les chapitres 7 et 8.

Le reste de la thèse comprend des articles publiés ou soumis. Ces derniers abordent principalement :

- La description de la nouvelle plateforme de création de simulation des scénarios mono et multivéhiculaires sur des plateformes distribuées. Ceci, à des fins de parallélisation et de la distribution des tâches de traitement sur plusieurs cœurs de calcul ;
- Les différents aspects caractérisant la conduite automatisée. Celle-ci nécessite des informations des trois acteurs de la scène routière : les marquages au sol, les obstacles statiques et mobiles présents dans la scène routière et l'ego-véhicule (véhicule équipé de systèmes embarqués : capteurs, actionneurs, algorithmes de traitement des données multi-capteurs, de planification de trajectoire, de contrôle, etc.) ;
- La localisation et la cartographie simultanées et coopératives. Cela, en exploitant les résultats de la fusion des données multi-capteurs des véhicules intelligents et les moyens de communication dans le but d'améliorer les performances, en matière de précision, de la localisation de ces véhicules intelligents et connectés ;
- Et enfin, l'utilisation de l'ensemble de ces informations pour la génération des trajectoires plausibles des véhicules.

# **Chapitre 5**

## **Real-time simulator of collaborative and autonomous vehicles**

### **5.1 Avant-propos**

#### **5.1.1 Auteurs et affiliation**

**Farid Bounini :** Étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Denis Gingras :** Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Vincent Lapointe :** Chef de produit, OPAL-RT Technologies inc. Montréal, Canada.

**Dominique Gruyer :** Chargé de recherche, IFSTTAR, CoSys – LIVIC, Versailles, France.

#### **5.1.2 Date de soumission**

2 juin 2014.

#### **5.1.3 État de l’acceptation**

Version finale publiée.

#### **5.1.4 Conférence**

International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014).

### **5.1.5 Référence**

BOUNINI F., GINGRAS D., LAPOINTE V., GRUYER D., "*Real-time simulator of collaborative autonomous vehicles*", IEEE Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI), Greater Noida, India. September 24<sup>th</sup> – 27<sup>th</sup>, 2014.

### **5.1.6 Titre français**

Simulateur temps-réel de véhicules autonomes en conduite collaborative.

### **5.1.7 Contribution au document**

Ce chapitre fournit un tableau des outils et des produits mis en œuvre durant le développement de la première version du simulateur ; dédié à la création et à la simulation de scénarios multivéhiculaires pour la conception et la validation numériques des systèmes d'aide à la conduite. Il décrit et met en avant les différentes fonctionnalités du simulateur de la dynamique des véhicules, des infrastructures et des capteurs embarqués (Pro-SiVIC), et la puissance du calcul en parallèle des produits d'OPAL-RT (RT-LAB et OP5600).

### **5.1.8 Résumé français**

Prochainement, les véhicules autonomes en conduite collaborative — véhicules intelligents et connectés — apparaîtront et apporteront des changements importants au niveau des systèmes de transport routier. En remédiant, en partie, à de nombreux problèmes tels que la sécurité routière, l'efficacité du trafic routier, etc. Tester et valider des scénarios complexes, impliquant un ensemble de véhicules intelligents et collaboratifs, est de nos jours le nouveau défi à relever. Chaque véhicule autonome simulé, agit d'une manière asynchrone, reçoit et traite en temps réel une quantité de données colossale, provenant à la fois de ses propres capteurs embarqués, de son environnement et des autres véhicules. La simulation de tels scénarios, en temps réel, nécessite des ressources de traitement numérique importantes. Ce chapitre présente un nouveau simulateur combinant les plateformes d'OPAL-RT, pour le traitement et le calcul parallèle en temps réel, et la plateforme Pro-SiVIC de la société ESI-CIVITEC, pour la simulation de la dynamique des véhicules, de l'infrastructure et des capteurs. Les deux plateformes sont complémentaires et leur combinaison nous permet de proposer un simulateur temps-réel de véhicules intelligents et autonomes en conduite collaborative.

## 5.1.9 Addendum

### 5.1.9.1 Versions de la plateforme

La figure suivante présente la nouvelle version de la plateforme. Elle comprend plusieurs plateformes de simulation de la dynamique de véhicules, des capteurs embarqués et des moyens de communication, notamment Pro-SiVIC, PreScan et EXata. Ces plateformes fournissent des données brutes, venant de capteurs embarqués et des moyens de communication inter-véhiculaire, aux plateformes d'OPAL-RT à travers le bus d'Orchestra. Cette dernière exploite les capacités de « Matlab/Simulink » pour traiter en parallèle les données reçues. RT-LAB sépare les modèles « Simulink » et en crée des exécutables. Ces derniers sont exécutés en parallèle et génèrent les signaux de commande de la dynamique des véhicules et les transmettent à travers Orchestra aux autres plateformes de simulation fermant ainsi la boucle du contrôle. Grâce aux capacités évolutives d'Orchestra, la plateforme est facilement adaptable et interfaçable à toute plateforme ayant des API « C/C++ » et « Python » et exportant leurs modèles de simulation sur Matlab/Simulink.

La figure suivante montre la version actuelle de la plateforme sur Windows, interfaçant les différentes plateformes de la dynamique de véhicules, des capteurs embarqués, et de la communication. La partie EXata est récemment développée, pour une démonstration avec les autres plateformes durant la conférence RT17. Cependant, la démonstration n'inclut pas la partie EXata, faute de manque de temps et de disponibilité de licence.

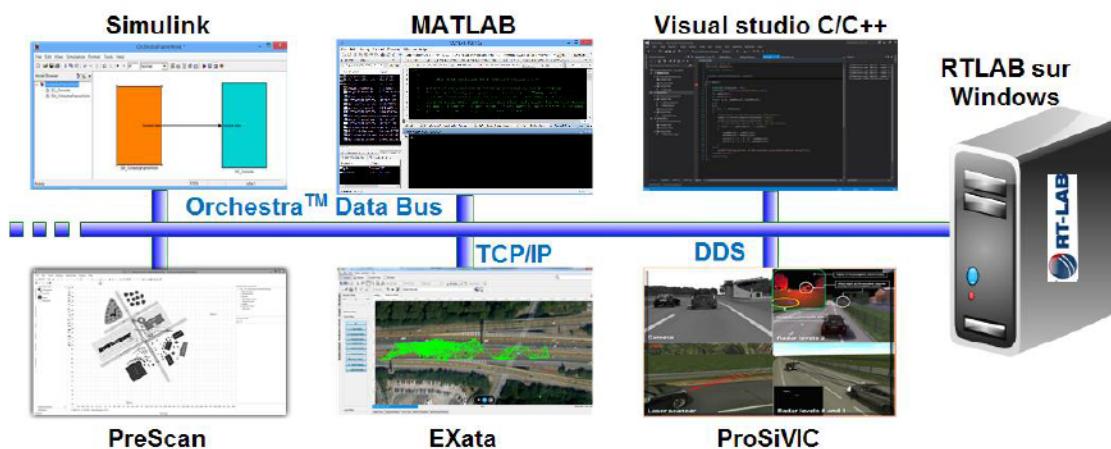


Figure 5-1 : Version actuelle de la plateforme de simulation de véhicules intelligents et connectés en conduite collaborative

Dans la future version, il est prévu que les différentes plateformes de la simulation de la dynamique des véhicules, des capteurs embarqués et de la communication [...], fonctionnant sur Windows, soient directement interfacées aux machines d'OPAL-RT à travers les ports TCP/IP « en : *Transmission Control Protocol/Internet Protocol* » ou UDP « en : *User Datagram Protocol* ». La figure suivante donne un aperçu général de la future version de la plateforme.

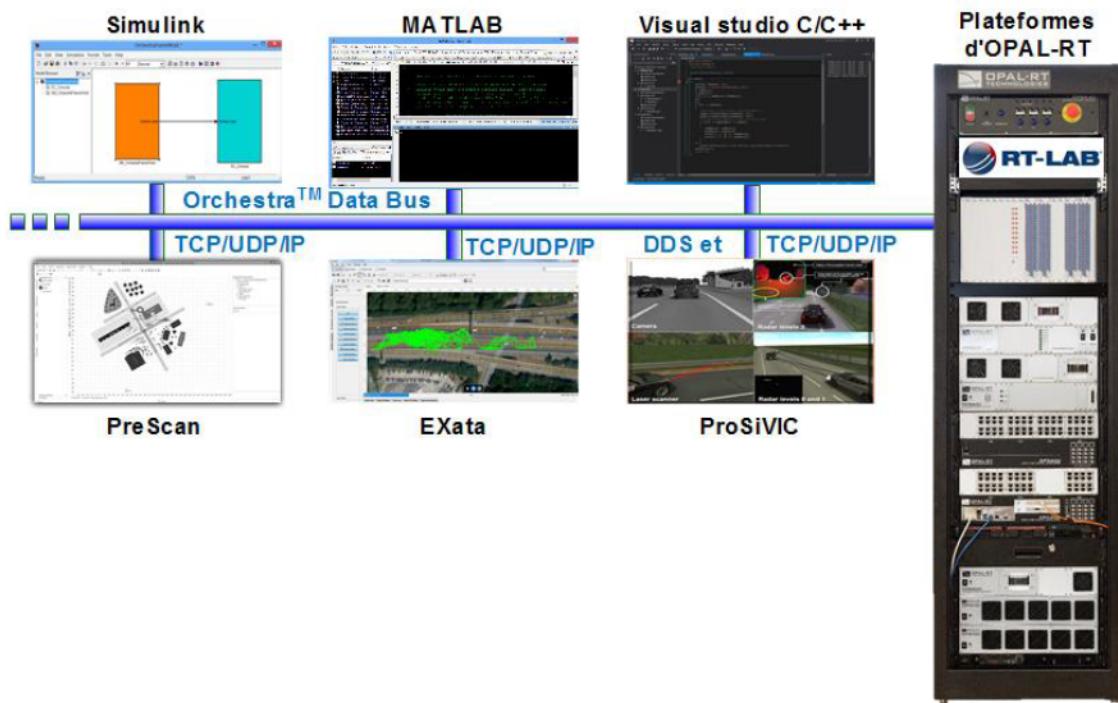


Figure 5-2 : Prochaine version de la plateforme de simulation de véhicules intelligents et connectés en conduite collaborative<sup>1</sup>

### 5.1.9.2 Verrous technologiques

Les principales difficultés rencontrées dans le cadre de ce projet se résument à l'indisponibilité et parfois à l'absence de certaines APIs d'interfaçage des différentes plateformes de simulation utilisées dans ce projet de recherche, à titre d'exemple :

---

<sup>1</sup> L'image de la plateforme EXata – dans les deux figures 5-1 et 5-2 – a été fournie par Mr. Lloyd Wihl de Scalable Networks et c'est sous son autorisation qu'elle a été intégrée à cette thèse de recherche.

- Pro-SiVIC : le rachat de la société CIVITEC, fournisseur du simulateur de la dynamique des véhicules et des capteurs, a engendré un retard important pour la mise en place de l'API DDS et son intégration à Orchestra ;
- RT-LAB : l'abandon de la fonctionnalité de communication TCP/IP par OPAL-RT Technologies, dans les nouvelles versions d'Orchestra, a engendré un certain nombre de limitations de fonctionnement de la nouvelle plateforme regroupant plusieurs systèmes fonctionnant sur le système d'exploitation Windows et sur PC ;
- L'utilisation de différentes versions des compilateurs « C/C++ », par nos partenaires industriels lors de la mise en place de leurs propres APIs, a engendré certaines incompatibilités des APIs d'interfaçage. Ainsi, d'autres défis relatifs à la compatibilité de ces différentes APIs ont dû être surmontés lors du développement de la nouvelle API « DDS/Orchestra » ;
- La limitation de la taille des variables d'Orchestra, variables d'entrée/sortie, réduit la capacité d'échange de l'API d'Orchestra. Ainsi, il est très difficile d'échanger des trames d'images d'un modèle à un autre ;
- Le fonctionnement des plateformes sur différents systèmes d'exploitation (Pro-SiVIC sur l'OS Windows et RTL LAB/OPxxx sur l'OS Redhat/Linux) était un autre problème auquel il fallait remédier. Ce dernier empêchait l'échange des données des capteurs et des contrôleurs d'une plateforme à une autre. Pour y remédier, une bibliothèque TCP/IP est développée, celle-ci est en cours de test. Pour un meilleur débit d'échange de données, il serait également intéressant d'avoir recours au mode d'échange UDP. Cependant, il faut, en premier lieu, analyser les performances du fonctionnement de la nouvelle plateforme utilisant ce mode d'échange de données et l'impact d'une perte de données échangées ;
- L'utilisation du « Simulink coder », par RTL LAB (lors de la génération du code « C/C++ » après la séparation des modèles Matlab/Simulink), limite le développeur à n'utiliser que les boîtes à outils « *en : Toolbox* » supportées par « Simulink coder ».

## 5.2 Abstract

Collaborative autonomous vehicles will appear in the near future and will transform deeply road transportation systems, addressing in part many issues such as safety, traffic efficiency, etc. Validation and testing of complex scenarios involving clusters of autonomous and collaborative vehicles are becoming an important challenge. Each vehicle in the cluster is autonomous and acts asynchronously, receiving and processing huge amount of data in real time, coming from its environment and other vehicles. Simulation of such scenarios, in real time, requires huge computing resources. This paper presents a simulation platform combining the real-time OPAL-RT Technologies for processing and parallel computing, and the Pro-SiVIC vehicular simulator from ESI-CIVITEC for realistic simulation of vehicles dynamic, road/environment, and sensors behaviors. The two platforms are complementary and their combining allow us to propose a real time simulator of collaborative and autonomous systems.

## 5.3 Introduction

The development of embedded sensors and communication technologies, during the last few decades, has influenced significantly the automotive sector, helping to overcome traffic problems and to achieve greater safety on roads and highways. Nowadays, on board active safety systems are essentially developed to avoid collisions rather than only reducing their impact on passengers as with passive systems. Furthermore, enhanced night vision, automated park assistance and navigation systems are functionalities that are already implemented on high-end vehicles [88].

Although, current Advanced Driver Assistance Systems (ADAS) are built as isolated subsystems that are typically not sharing information within or between vehicles. The future generations of ADAS will feature a much higher level of integration as well as extended communication capabilities allowing to exchange information between vehicles. This will lead to fully automated vehicles capable of sharing information and interacting with dynamic environment. Some research efforts are focusing on the cooperative and optimized behavior of nodes, and multi-hop networks [89], some specific aspects are introduced as friends/enemies, dominance, etc., in [90] to optimize the behavior of multi-hop network. On the other side, some

basic scenarios on collaborative autonomous vehicles have been experimented with three collaborative autonomous vehicles at the Griffith University’s Intelligent Control Systems Laboratory (ICSL) with its partners [91]. In these experiments, simple scenarios were considered such as overtaking, transversal intersection and lane platooning.

This paper presents a real-time simulator of collaborative and autonomous vehicles on parallel computing tools, and deals mainly with nine points: Section II is dedicated to embedded sensors technology and ADAS systems. Section III describes our strategy to build a real time simulator architecture of collaborative and autonomous vehicles. Section IV presents the OPAL-RT platforms (software: RT-LAB and hardware: e.g. OP5600) for real-time simulation and parallel computing in high data throughput situations. These platforms allow us to simulate the embedded information processing systems and the “brain” behind each individual autonomous vehicle, as well as the VANET (Vehicular Ad-hoc Network) to exchange relevant information between the vehicles in collaborative scenarios. Section V provides a brief presentation of the Pro-SiVIC simulator and explains our choice due to its high capability to simulate the other essential components of multi-vehicular collaborative scenarios such as the various sensors, the road environment and the dynamic of each vehicle in real-time. Section VI interconnects RT-LAB and Pro-SiVIC platforms to complete the required architecture to simulate collaborative autonomous vehicles scenarios in real-time. Section VII presents some simulation results of the ACC (Adaptive Cruise Control) scenario with inter-vehicular communication capabilities. As the main embedded processing/control algorithms; PI “*Proportional Integral*” controller, where the details are given in section VIII. Section XI analyzes and comments the simulation results, then it is followed by a conclusion.

## 5.4 Proposed strategy

As the system complexity increases, the design, prototyping and validation of autonomous vehicles in multi-vehicular scenarios will become a major challenge. The computer power, memory space and data communication requirements for real-time simulation of these complex scenarios will need dedicated hardware/software architectures. The goal of this paper is to describe some solution elements, which are based on; the RT-LAB platform produced by OPAL-RT, and the advanced vehicle simulator named Pro-SiVIC, provided by ESI-CIVITEC.

We consider several scenarios involving clusters of cooperative vehicles equipped with given sets of sensors, and on board processing capabilities, able to share some of their data dynamically (e.g. position, orientation, speed, acceleration, etc.). Pro-SiVIC simulator allows us to generate a realistic simulation of the vehicles dynamic, their embedded sensors and the infrastructure. The RT-LAB platform allows us to simulate the on-board real-time processing, to estimate the various parameters needed for generating the various control signals, and to be used by the Pro-SiVIC simulator, in order to control the dynamic of each vehicle involved in the scenario under study. The on-board processing of each vehicle may be quite different and rely on different computing resources. Typically, the set algorithms involved in each vehicle include some multi-sensors data fusion, detection and estimation processes as well as recognition, tracking and risk assessment analysis. Each vehicle has its own set of algorithms that may differ substantially from one vehicle to another, bringing further complexity to the multi-vehicular scenarios, which lead to intractable NP (Non-deterministic Polynomial) problems.

In order to assess the behavior of these complex systems, advanced simulation tools are required, which allow to randomize the various parameters in play and to sample the multidimensional operating space of the systems under study, either for design, testing or validation. Therefore, our goal is to develop a real-time simulator for collaborative and autonomous vehicles, which will permit to assess and improve the algorithms' behavior. They are usually used for pre/post processing of heterogeneous multi-sources data before moving to physical prototyping. Given the complexity of the new embedded technologies in the vehicles and the requirement of compliance with safety standards from transportation agencies and thanks to real time simulation, we can minimize the cost in the design phase and have a better assessment of complex systems while interacting with multi-vehicular scenarios. This new concept allows users, in numerical (virtual) environment, to deal with specific and dangerous cases that must be identified before real prototyping, in order to detect potential failures or defects before deployment on the market. Indeed, whenever a defect is detected a priori Original Equipment Manufacturers (OEMs) can earn hundreds of times of recalling and repairing costs as often happened in the automotive industry. Hence, extensive tests must be performed before large-scale manufacturing of complex technologies that are used in intelligent, autonomous and connected vehicles.

## 5.5 Real-time simulator architecture for multi-vehicular scenarios

To evaluate the performance of the data processing and decision-making algorithms that are implemented in autonomous vehicles, in case of multi-vehicular scenarios, we need to use a set of simulation tools that can operate in real time and emulate real conditions as closely as possible. In this paper, we focus on the development of parallel architectures for the real-time simulation of data fusion algorithms, and real time processing of multi-sensors data. We deal in particular with cooperative localization and perception issues, in the context of active safety for autonomous vehicles. Simulating these scenarios involves the detailed mathematical model of the vehicles dynamic, the road environment, the on-board sensors, the real-time processing, communication and control systems embedded in each vehicle. Each one processes its own data and broadcasts relevant information to others. They independently operate and make their own decisions in real time. To simulate the different tasks performed in each vehicle, we use a set of processing cores operating in parallel on the OPAL-RT platform (RT-LAB and OP5600). Once the parallel processing of all vehicles (cores) is completed, the vectors state of the individual vehicle will be communicated to others cores, in order to make further decision or accomplish other actions, such as improving their own ego position estimate.

## 5.6 Real-time laboratory platform

RT-LAB (Real-Time Laboratory) is a powerful real-time simulation platform, which allows users to implement complex dynamic mathematical models that are designed primarily on Matlab/Simulink for real-time integration with external equipment —Hardware In the Loop (HIL) or Rapid Prototyping Controller (RPC) modes— forming closed-loop systems. It offers the possibility to separate complex models into several simple sub-models, subsystems, and processes them on parallel architectures to perform complex tasks [92]. Its Input/output (I/O) modules allow the models to communicate with real environment during simulation, which provides the possibility to test multiple products in real time. The XHP (eXtra-High Performance) mode permits very fast computation of real-time models on the OPAL-RT's target, while using I/O modules at high frequency [93]. Its high computing speed allows designers to work on all types of real time distributed systems such as multi-vehicular scenarios, involving intelligent, autonomous and connected vehicles.

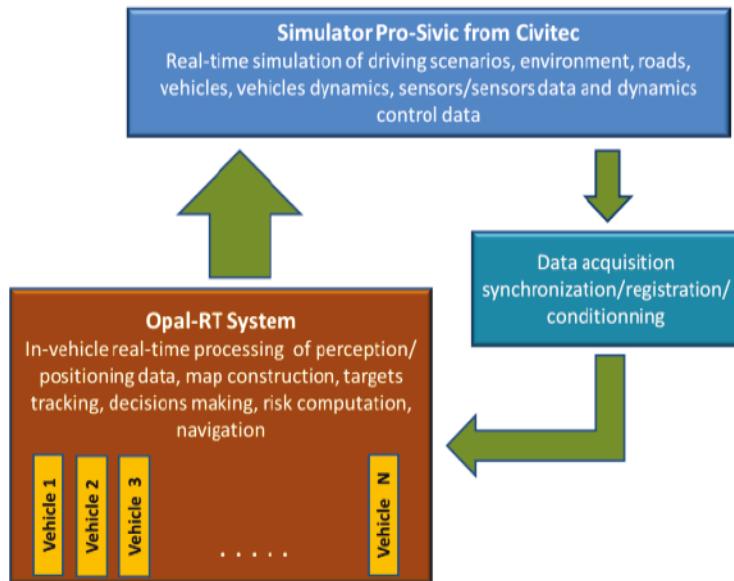


Figure 5-3: OPAL-RT platform and cores attribution for parallel processing.

## 5.7 Implementation on OPAL-RT software (RT-LAB)

The RT-LAB builder process allows generating several C/C++ executables from Simulink models by distributing the multi-vehicular scenario into several subsystems (vehicles). The code is optimized for real-time execution on LINUX-Redhat and QNX operating systems. Depending on available resources, OPAL-RT targets cores are either automatically assigned to each subsystem built in Simulink, or manually through assignment with the RT-LAB's graphical user interface (GUI).

Numerous capabilities of the platform allow on-line parameter adjustment, signal acquisition, data logging, and some other features are commonly used for making model development and testing easier. The platform includes data acquisition modules and signal processing software that is optimized for visualization and analysis of real-time results. It may be used to simultaneously load, view, and process data from applications such as Matlab/Simulink. It also possesses advanced mathematical post processing capabilities, and is able to save and load templates associated to customized sessions of similar simulations. RT-LAB allows several commercial software to be quickly interfaced with other simulators such as Pro-SiVIC from ESI-CIVITEC including modeling tools: Plecs, Matrixxx, Statemate, C/C++, FORTRAN, etc.

## 5.8 OP 5600 simulator presentation

To implement our multi-vehicular scenarios, we use the OPAL-RT's OP5600 simulator based on either Spartan 3 or Virtex 6 FPGA (Field-Programmable Gate Array) and CPU (Central Processing Unit) [94] that makes it an ultra-fast simulator, since it uses multiple cores for parallel computing. This simulator can be used as a desktop or traditional rack mount for high precision and high speed in real time simulation. The new improvements on Input and Output circuits allow the avoidance of I/O adaptors and use standard connectors (DB37, RJ45, etc.), which makes monitoring easier.

The OP5600 simulator has the following features [94] :

- Great computation power: Powerful real-time targets (12 CPU cores 3.46 GHz), real-time OS (Linux Redhat), and distributed parallel computation ;
- Important Inputs and Outputs capabilities: Up to 128 analog or 256 digital IO or mix of both, according to the configuration of the rear D-sub 37 connector for external devices, many chassis can be connected together to form larger IO tracks, etc. ;
- Connectivity: Embedded hard driver for real-time data logging, and various communication protocols are available and offered by RT-LAB to communicate between the subsystems (Dolphin, Ethernet, Firewire, etc.), and many others dedicated to automobile on-board data communication: Local Interconnect Network (LIN), Controller Area Network (CAN).

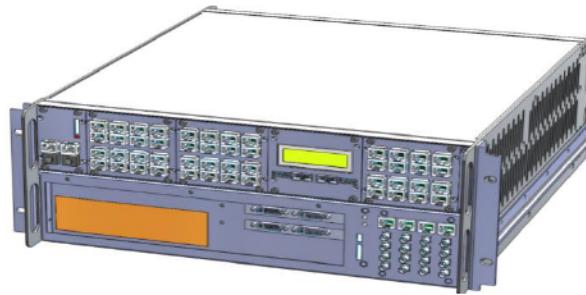


Figure 5-4: OP5600 simulator [95].

## 5.9 Pro-SiVIC simulator

Nowadays, most of developments aim to improve the safety of the road through advanced driving assistance systems (ADAS). These studies generally take into account vehicle's

perception and maneuvers (e.g. position, heading angle changing, braking and accelerating). Generally, ego perception is no longer sufficient. Additional information is required to eliminate or at least minimize accidents on roads and make them safer by making collaborative decisions. This additional information requires additional resources which are both pricey and time-consuming for data processing. Therefore, it becomes essential to have real-time tools allowing to prototype and to assess, at each stage, the performance and improvement of collaborative and autonomous vehicles before the design phase. Real time simulation platforms have to integrate physical models of vehicle environment, embedded sensors —both proprioceptive and exteroceptive— and communication devices (transmitters and receivers). Other mathematical models should be added like vehicle dynamics coupled with actuators (steering angle and wheels torque), etc.

Pro-SiVIC is a real time simulation tool and offers requirements for developing and prototyping collaborative and autonomous driving systems that are involved in multi-vehicular scenarios. Pro-SiVIC platform includes several categories of sensors and communication tools [96],[97]–[99]. The exteroceptive sensors are cameras, laser scanners, RADio Detection And Ranging (RADAR) and LIght Detection And Ranging (LIDAR). The proprioceptive sensors involve some local positioning sensors: odometers, Inertial Navigation Systems (INS) and Global Positioning Systems (GPS). The communication protocols, for collaborative and autonomous systems, include both IEEE 802.11p communication media and beacon. Pro-SiVIC provides the possibility to tune and set the time stamp in real time for each sensor, communication media, and change the intrinsic and extrinsic parameters as well. The user could activate several operation modes, which can be modified during the simulation:

- “Off” and ‘On’ to switch on or switch off a sensor ;
- “Record” in order to save the sensors data in a file ;
- “RTMaps” and “Matlab” to broadcast sensors data and actuators signals to the two named software ;
- and DDS (Data Distributed Store) for external applications data exchange. Figure 5-5 shows some scenarios with operating embedded exteroceptive sensors. RT-LAB/Pro-SiVIC : Interconnected platforms for real time simulation of collaborative and autonomous vehicles.

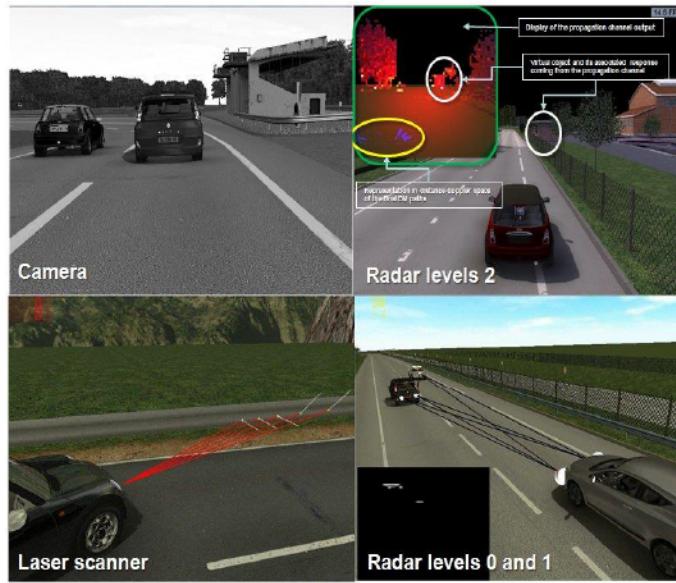


Figure 5-5: Exteroceptive sensors in Pro-SiVIC.

Pro-SiVIC simulates the virtual prototypes of vehicles with their proprioceptive and exteroceptive embedded sensors. OPAL-RT platform coupled to the Pro-SiVIC simulator allows to simulate in real-time the embedded data processing and the decision-making process of each vehicle that is involved in the scenario. Pro-SiVIC provides all the required models to simulate the vehicles accurately, the road environment, and the sensors embedded in each vehicle. In addition, Pro-SiVIC allows to replace real data by simulating the dynamic of multiple vehicles, allowing RT-LAB to operate in Software In the Loop (SIL) mode. Thus, the interconnected platforms provide us a full platform, for advanced real time data processing, inter-vehicular communication, prototyping, validation of the control and perception algorithms for real time simulation of collaborative autonomous vehicles. RT-LAB processes at high speed the virtual data coming from Pro-SiVIC's virtual vehicles and their respective virtual embedded sensors. The RT-LAB output results are used as input control signals back to the Pro-SiVIC simulator in order to control the vehicle dynamics. Hence, once virtual processing and decision-making are completed on the RT-LAB platform, control signals are sent in real time from the RT-LAB platform to Pro-SiVIC simulator to control the set of virtual autonomous vehicles. Thanks to RT-LAB/Pro-SiVIC platform all data processing and control algorithms, once tested and validated using extensive simulations, can be directly transferred and used in real hardware of intelligent and connected vehicles.

## 5.10 Proportional and integral controller

The Proportional and Integral (PI) controller eliminates static errors and rejects external perturbations [100]–[103]. To construct the PI controller, additional state variables are required for the open loop linear systems, like tracking error integral. As a result, the following augmented linear system is obtained:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ z(t) = Cx(t) \\ \dot{y}(t) = e(t) = z_{ref}(t) - z(t) \end{cases} \quad (5.1)$$

where:  $x$ ,  $u$  and  $z$  are the state, control and observation vectors of the open loop model with the following dimensions :  $n$ ,  $p$  and  $q$ , respectively.  $z_{ref}$  and  $y$  are the input reference and the error integral between the reference input and observation output vectors of the closed-loop system.  $A$ ,  $B$  and  $C$  are the state transition, control and observation matrices of the open loop model, respectively.

The extended model is written as follows:

$$\begin{cases} \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} A & 0_{n \times q} \\ -C & 0_{q \times q} \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} B \\ 0_{q \times p} \end{bmatrix} u(t) + \begin{bmatrix} 0_{n \times q} \\ I_{q \times q} \end{bmatrix} z_{ref}(t) \\ z(t) = \begin{bmatrix} C & 0_{q \times q} \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \end{cases} \quad (5.2)$$

We then place the poles of the extended system by choosing adequate feedback gain matrices ( $K_x$ ,  $K_y$ ) such that:

$$u(t) = -[K_x \ K_y] \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \quad (5.3)$$

The following figure summarizes both controller and state observer, with its specific observation gain matrix “ $L$ ”, in state space for closed loop system:

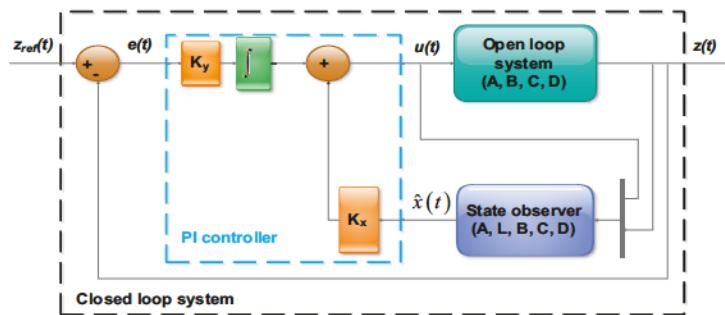


Figure 5-6: Closed loop scheme with PI and state feedback controllers.

## 5.11 Simulation of Adaptive Cruise Control

To simulate the two vehicles for Adaptive Cruise Control (ACC), the vehicles dynamic, data processing and control algorithms have been implemented on Matlab/Simulink. Each vehicle was implemented by a subsystem, according to the RT-LAB protocol. Figure 5-9 shows the closed loop configuration. Figure 5-10 shows that the inter-vehicle distance “ $D_{V1V2}(t)$ ” converges to the safe distance set herein to 30 meters, where  $t_d$  depends on drivers type/nature (human driver  $t_d=1.5s$ , autonomous vehicles  $t_d=10^{-3}s$ , etc.). The different peaks are caused by sensors noise of each vehicle during the simulation. The closed loop with PI controller offers the system robustness, tracking and precision, which rejected exogenous perturbations. The second vehicle is instructed to maintaining a specific inter-vehicle distance, at least equal to a safe distance (set here to 30 meters). Inter-vehicle safe distance and communication delay effects are visible on the Figure 5-9, where the second vehicle’s coordinates on both axis (OX, OY) are delayed in comparison to the first vehicle’s coordinates. Data exchanges are performed through the communication channel, which is modeled by a delay of 10 ms. Figure 5-10 shows the vehicles’ trajectory on Satory test track. The accuracy of the second vehicle depends not only on communication delay, but on the safe distance, the speed of the vehicle and the radius the turn. Therefore, if the safe distance and the rate-of-turn are important, the radius of turn is short then the accuracy becomes lower, and vice versa. To improve the precision of the system, road maps, curvilinear coordinates, and paths historic of vehicles must be shared by communication means, to determine the exact inter-vehicle distance as a function of the radius of the road and the appropriate rate-of-turns.

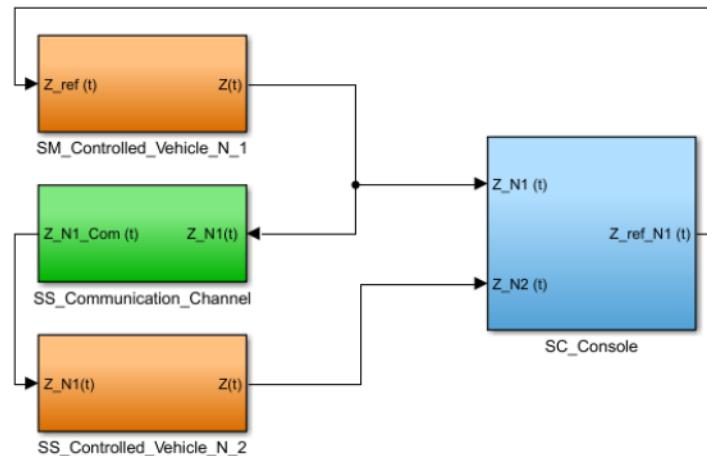


Figure 5-7: Simulink model of ACC with RT-LAB's standard configurations

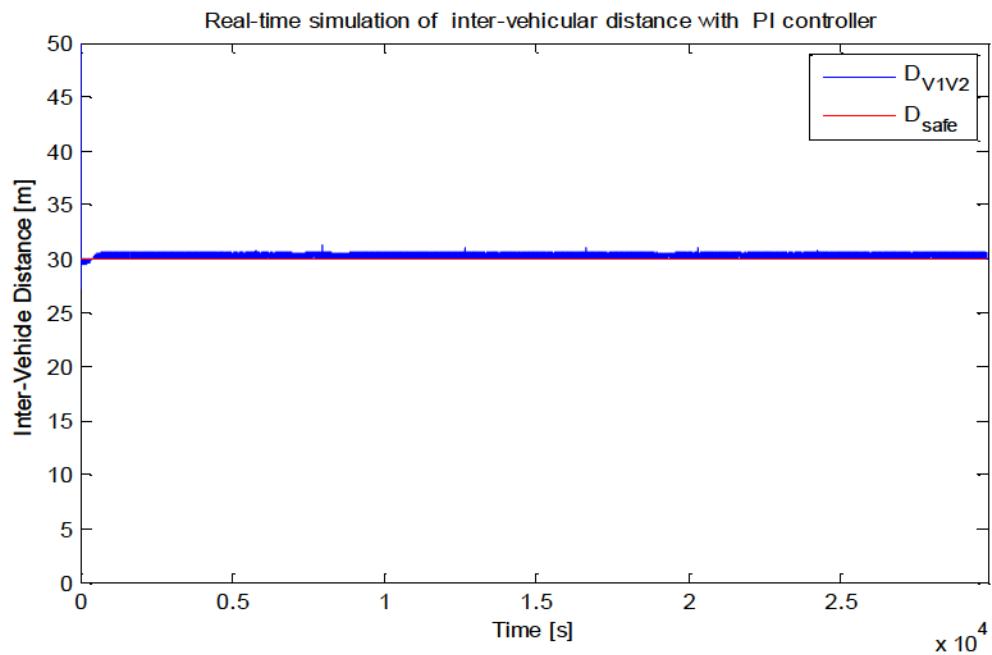


Figure 5-8: Inter-vehicular distance with PI controllers.

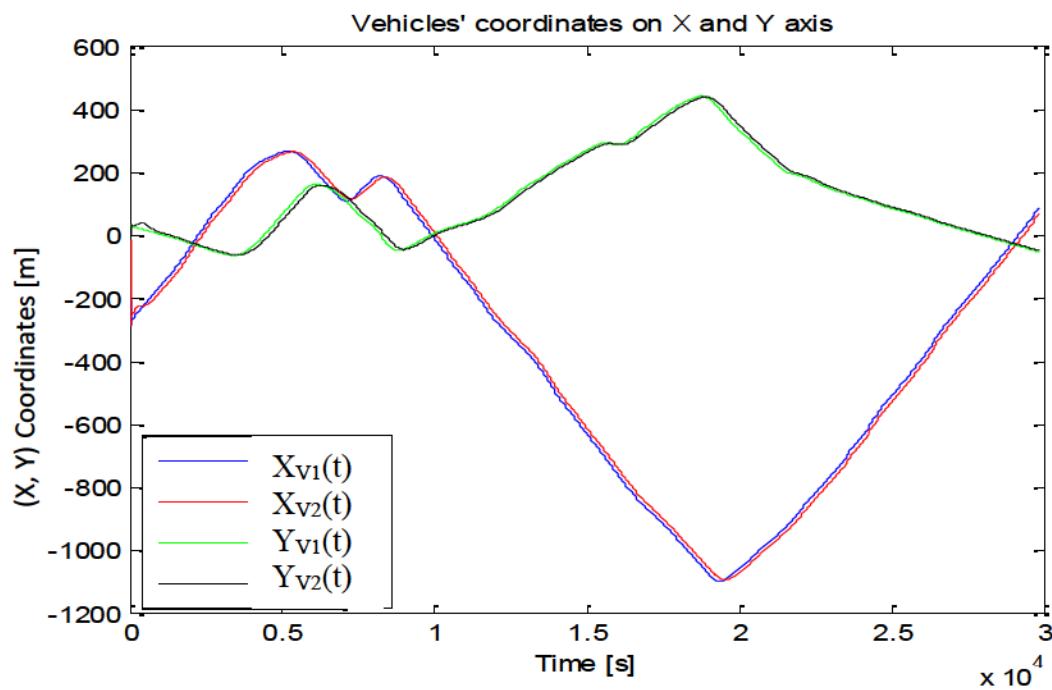


Figure 5-9: Vehicles' coordinates evolution

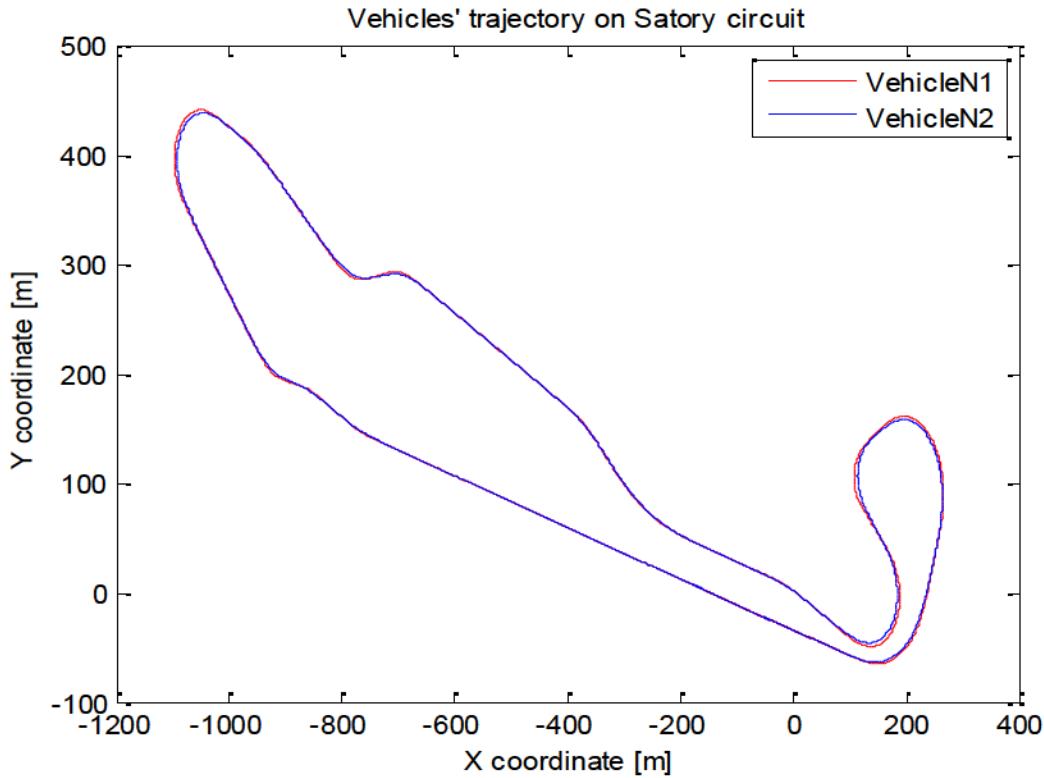


Figure 5-10: Vehicles' trajectories (Satory test track)

## 5.12 Conclusion

This paper presents the first steps to design a real time simulator of collaborative and autonomous vehicles based on two platforms: The first is OPAL-RT Technologies. It offers several architectures for parallel computing, providing high speed processing and high precision. The second is Pro-SiVIC simulator from CIVITEC. It is used to simulate the road environment and vehicles' dynamic as well as the embedded sensors behavior. The main objective is to interface the two platforms (RT-LAB/Pro-SiVIC) for real time simulation of clusters of intelligent, autonomous and connected vehicles, each one having its own sets of sensors and embedded processing and communication capabilities. This paper has shown some preliminary results with a basic scenario with two intelligent vehicles that are equipped with virtual communications means to simulate ACC behavior.

# **Chapitre 6**

## **Autonomous Vehicle and Real Time Road Lane Detection and Tracking**

### **6.1 Avant-propos**

#### **6.1.1 Auteurs et affiliation**

**Farid Bounini :** étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Denis Gingras :** Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Vincent Lapointe :** Chef de produit, OPAL-RT Technologies inc. Montréal, Canada.

**Hervé Pollart :** Responsable du développement commercial de la voiture intelligente chez OPAL-RT Technologies inc. Lyon, France.

#### **6.1.2 Date de soumission**

13 avril 2015.

#### **6.1.3 État de l’acceptation**

Version finale publiée.

#### **6.1.4 Conférence**

Vehicle Power and Propulsion Conference (VPPC, 2015).

### **6.1.5 Titre français**

Détection et suivi des voies de la route en temps réel pour les véhicules autonomes.

### **6.1.6 Référence**

BOUNINI F., GINGRAS D., LAPOINTE V., POLLART H., “Autonomous Vehicle and Real Time Road Lanes Detection and Tracking.” IEEE Vehicle Power and Propulsion Conference (VPPC), Montreal, Canada, October 19<sup>th</sup> – 22<sup>nd</sup>, 2015.

### **6.1.7 Contribution au document**

La contribution principale de ce chapitre réside dans la méthode proposée pour la détection et la poursuite des lignes et des bords de la route sur plusieurs coeurs de calcul. Elle vise à montrer comment exploiter les capacités du simulateur, présenté précédemment, en matière de :

- Simulation de la dynamique des véhicules ;
- Génération des données de capteurs extéroceptifs et proprioceptifs ;
- Calcul parallèle, parallélisation de la méthode de traitement d’image ;
- Simulation à logiciel dans la boucle « *Software-In-the-Loop (SIL)* ».

### **6.1.8 Résumé français**

Les systèmes d'aide à la conduite, les véhicules intelligents, autonomes et connectés sont des solutions prometteuses pour améliorer la sécurité routière et résoudre la plupart des problèmes de la circulation routière et du confort des passagers. Ces applications nécessitent des algorithmes complexes de vision par ordinateur, et exigent du matériel informatique de capacités de calcul et de traitement importantes. Le maintien des véhicules intelligents sur la route jusqu'à leur destination, dans certains cas, reste un grand défi, notamment lors de la conduite à grande vitesse. Par conséquent, la tâche principale de telles applications est d'assurer une navigation fiable, qui repose souvent sur des systèmes de vision afin d'acquérir des images en couleurs RVB (Rouge, Vert, Bleu) de la route pour des traitements poussés. La deuxième tâche est le contrôle de la dynamique du véhicule en fonction de sa position, de sa vitesse et de son orientation.

Ce chapitre présente un algorithme de détection des lignes de marquage au sol précis et efficace pour les véhicules intelligents. La stratégie combine : la transformée de Hough (TH) — pour initialiser et réinitialiser l'algorithme à chaque fois que cela est nécessaire —, le détecteur de bordures de Canny, la méthode des moindres carrés et le filtre de Kalman pour minimiser la taille des régions d'intérêt adaptatives et prédire les paramètres des lignes et des bordures de la route. Un scénario multivéhiculaire est créé et simulé sur la plateforme Pro-SiVIC fournie par ESI-CIVITEC, et sur les plateformes d'OPAL-RT dédiées au calcul parallèle et au traitement en temps réel.

## **6.2 Abstract**

Advanced Driving Assistant Systems, intelligent and autonomous vehicles are promising solutions to enhance road safety, traffic issues and passenger comfort. Such applications require advanced computer vision algorithms that require powerful computers with high-speed processing capabilities. Keeping intelligent vehicles on the road until their destination still a great challenge, particularly when driving at high speeds. The vehicle's first task is navigation, which is often based on system vision. The latter acquires RGB (Red, Green, Blue) images of the road for more advanced processing. The vehicle's second task is dynamic control according to its position, speed and direction. This paper presents an accurate and efficient road boundaries and painted lines' detection algorithm for intelligent and autonomous vehicles. It combines Hough Transform to initialize the algorithm each time it is needed, Canny edge detector, least-square method and Kalman filter (KF) to minimize the adaptive region of interest, predict the location of future road boundaries and lines' parameters. The scenarios are simulated on the Pro-SiVIC simulator provided by ESI-CIVITEC, which is a realistic simulator for vehicle dynamics, road infrastructure, and sensors behaviors, and OPAL-RT's products (simulators and software) dedicated to real time processing and parallel computing.

## **6.3 Introduction**

One of the technologies that deeply contributes to the proliferation of Advanced Driving Assistance System (ADAS) is the low-cost embedded Micro-Electro-Mechanical systems (MEMs), which are becoming cheaper and cheaper. Affordable memory and computing resources contribute to the development of ADAS and intelligent vehicles. These systems enhance road safety and resolve some of the traffic issues. One of the major challenges for self-driving vehicles is related to navigation issues in uncertain or non-static environments. Nevertheless, artificial intelligence and computer vision offer potential solutions for autonomous vehicle navigation in an unstructured environment: scene analysis, objects classification, etc. One of the solutions is a system vision that could be based, either on one monocular camera with morphological image processing operators [106]-[108], fusing road geometry and B-Snakes[109], or several cameras for advanced processing such as inter-objects distance estimation and 3D objects reconstruction [110]-[111]. However, road lane detection is

still a difficult task for intelligent vehicles. This is essentially due to the huge amount of data that should be processed in real time. The algorithms should recover the road state with some uncertainties: shadow, vehicle vibration, sensors noise, etc. To overcome such constraints, new methods should be fused with the traditional image processing algorithms to improve them and better target the region of interest and minimize the computational cost for the real-time applications.

This paper presents a simulation architecture for autonomous vehicles and real-time road lane detection and tracking. The paper involves five sections: the first section is dedicated to some related work on road boundaries and painted lines detection. In the second section, a description of the main scenario that was created on Pro-SiVIC [117]-[118] — a simulator of infrastructure, vehicle dynamics, and embedded sensors — is presented. The third section deals essentially with road lanes detection algorithms by applying Canny edge detector and Hough Transform to initialize the proposed method that is based on adaptive regions of interest, least-square method and Kalman filter to predict the next position of the road boundaries and lines' parameters. The fourth section describes the control strategy based on road lane center and fuzzy logic. This is followed by the fifth section that provides some simulation results on road lanes tracking, and then the paper is concluded.

## 6.4 Simulation scenario

The scenarios are mainly implemented on two different platforms in order to simulate the road lanes detection system. The first platform is called Pro-SiVIC, which is a real-time simulator of the road, environment (infrastructure), vehicle dynamic, and embedded sensors (Inertial Navigation System (INS), odometers, Light Detection And Ranging (LiDAR), camera, etc.), which is provided by ESI-CIVITEC. It allows users to develop multi-vehicle scenarios for real-time simulations. The second platform is the OPAL-RT products, RT-LAB (Real Time Laboratory) to separate, compile, and prepare the mathematical model — initially built on Matlab/Simulink, C/C++, etc. — for parallel computing. The combination of RT-LAB and Pro-SiVIC forms a powerful platform for ADAS, intelligent/autonomous and collaborative/connected vehicles simulation and validation. The reader can find more details on these platforms in [112]-[113].

The main scenario consists of generating a virtual environment, set here to the standard Pro-SiVIC trajectory called “HorseRing”. The circuit includes road, traffic barriers, landscape, etc. Two cars are used. They are both equipped with an INS connected to their chassis to measure their acceleration and orientation, and with a camera on their top, directed towards the road scene and take pictures each 40ms. The RGB images are converted to gray to detect road lanes. The Canny edge detector is applied to the gray frame to perform “Hough Transform” on the edge images to detect the lines that match road boundaries and painted lines. This initializes the suggested algorithm that consists of selecting the right regions of interest (ROI) — a rectangle with a center that corresponds to the means of each road boundaries, width W, and length H.— that fits with the last measurement and their Kalman filter (KF) parameters estimation.

## **6.5 Selection of regions of interest corresponding to the road’s section in the image**

The on-board cameras, placed at the front of a car and protected behind the windshield, take N frames per second. The frame itself contains an amount of data, which includes information about the road boundaries and the road lines. For this reason, a region of interest (ROI), which holds the road’s major information, is carefully selected to mitigate the computational cost, as shown in Figure 6-1, with the green rectangle. Another solution is to minimize the size of the ROI corresponding to each road boundary and line. This could be done after determining the road lines coordinates in the image, then a ROI is defined and associated to each line. This procedure allows the algorithm to better target the lines and minimize the image processing time. More details are given in the next sections. The next step, after choosing the ROI, is RGB image conversion to grayscale. The main objective is to generate an image with one layer rather than three, like RGB, without compromising the original road line information. The concept saves more computational power for further processing data like searching and tracking the adequate road lanes in the grayscale frame.

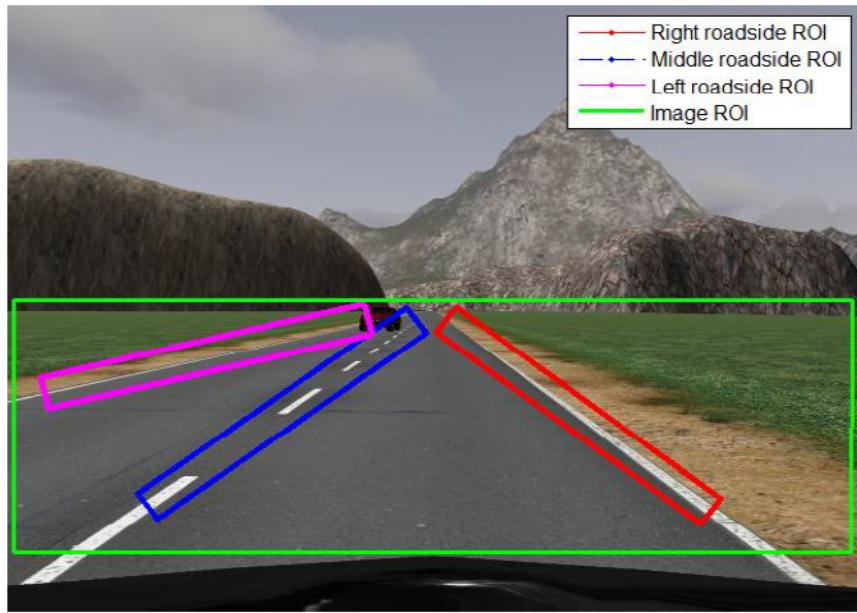


Figure 6-1: Regions of interest

## 6.6 Road-lanes detection

### 6.6.1 Edge detection

Road boundaries detection is the most difficult and important task that an intelligent vehicle should perform. They are often defined by the sharp contrast between the road bitumen and any non-pavement surface or painted lines [106]. The pixels, with high gradient values, are the image edges, where some of them correspond to road boundaries coordinates. To determine those edges, there are several edge detectors that could work, like Sobel, Prewitt, zero cross, Roberts, Laplacian of Gaussian, Canny, etc. Nevertheless, Canny edge detector is still the one that provides the best edged frames [106], and it is described by two steps [107].

- Smoothing the ROI by convolving a Gaussian mask with the image ;
- Getting the region of high contrast with a 2D first derivative operator.

In this work, a Canny edge detector is used to determine the edges of the frames, essentially the road boundaries and painted lines as shown in Figure 6-2. In order to keep the autonomous vehicle on the road, the frame's edges should be classified to hyperboles [106] or to straight lines [114] according to the vehicle's relative position and direction — to predict the future road lane parameters — and to the road geometry.

### 6.6.2 Edges and painted line colors

The color detection method is an additional but unnecessary step. It could be used to check the road lane edges and distinguish them from the image's background. It makes some classification, like determining whether the road lines edges are surrounded by bitumen or not, which is often dark, gray or black. The main problem of RGB image processing is the threshold determination that should be adaptive in order to handle the different RGB image contrast and brightness. One solution is to construct, for each frame, a histogram to calibrate the threshold to look for any white or yellow marks on the road.

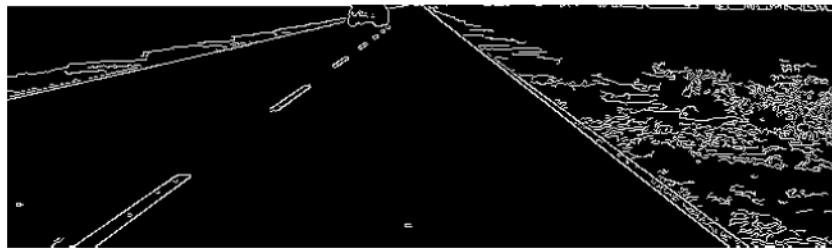


Figure 6-2: Canny edges detector results

## 6.7 Hough transform and lines' detection

High-speed process and computational resources improvement permit an important advance in computer vision, allowing to perform several complex operations on each image for further advanced applications like building a 3D model from 2D frames, inter-objects distance estimation, etc. These operations are conditioned by searching and classifying multiple lines in an edged image. Thus, to overcome the time-consuming issues, one should appeal powerful and efficient algorithms that detect different lines in an edged frame. One of the most effective methods is Hough transform (HT) [114], which is named after its inventor Paul Hough, who patented it in 1962 [115]. The main advantage of HT is its robustness to noise and occlusions, since HT is invariant to pixels' position in a frame. In other words, the algorithm is still valid to estimate noisy and dashed edges [116]. It transforms a set of frame pixels, points, in the Cartesian space to another space known as Hough space according to specific parameter space (HT) [116]. Some limitations, on the lines parameters, could be considered regarding the camera viewport, road geometry, or by modifying the HT as suggested in [106] by discarding any horizontal line, restricting lines' slope in the left and the right side at  $\pm\theta_{\max}$ .

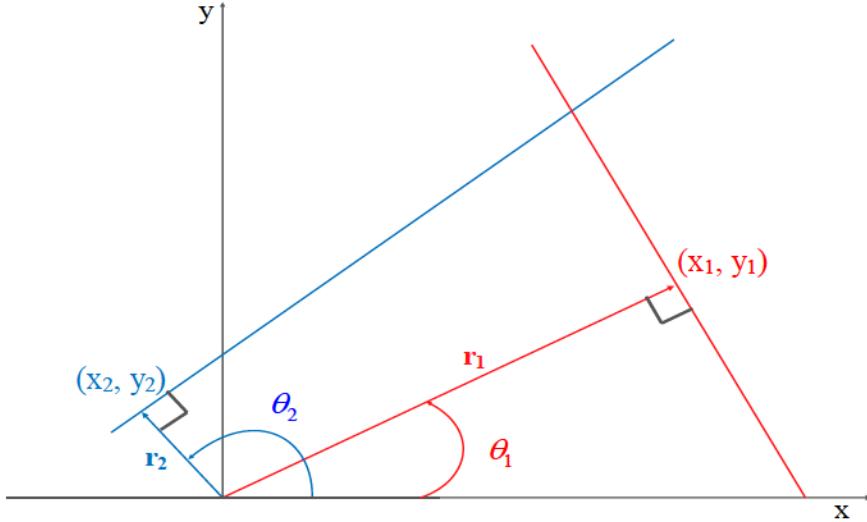


Figure 6-3: Lines representation in Hough space

Another option is to partition the ROI to N sections, where each section is tied to the road boundary or to a painted line. The HT is given as follows :

$$r_i = x \cos \theta_i + y \sin \theta_i \quad (6.1)$$

where  $x$  and  $y$  are the coordinates of the road lines that correspond to the rows and the columns of  $i^{\text{th}}$  the line.  $r_i$  is the smallest distance between the origin of the 2D plane and the  $i^{\text{th}}$  line  $r_i = \sqrt{x_i^2 + y_i^2}$  and forming the angle  $\theta_i = \text{atan2}(y_i, x_i)$  with the  $x$  axis and  $\theta_i \in [-\pi, \pi]$ .

## 6.8 Adaptive region of interest and polynomial approximation

Hough-Transform is computationally expensive. This is due to its lines clustering in the high resolution images that require an advanced numerical processing, which makes it (HT) not feasible for real-time simulation or embedded applications [114]. To meet real-time requirements and to overcome the HT gaps, vibrations and high-speed constraints, we suggest using HT to detect and reconstruct the road lanes' boundaries. This initializes our method, which is based on adaptive ROI dedicated to each road boundary and painted lines as shown in Figure 6-1 (red, blue and magenta rectangles). The second step is to find a mathematical model — polynomial model (6.2) — that fits the road boundary edges.

$$y = p_n x^n + p_{n-1} x^{n-1} + \dots + p_0 \quad (6.2)$$

This model consists of looking for a polynomial of the  $n^{\text{th}}$  order that better fits the road lines counters, which is represented by a linear system of the form  $X * p = Y$ .

$X$ ,  $Y$  and  $p$  are x and y counters' coordinates vectors and polynomial parameters vector respectively.

$$\begin{bmatrix} x^n & x^{n-1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_n & p_{n-1} & \cdots & p_1 \end{bmatrix}^T = y \quad (6.3)$$

In order to find  $p$  values, we need at least a set of  $n$  equations formed by each edges pixel, which gives:

$$\begin{bmatrix} x_1^n & x_1^{n-1} & \cdots & 1 \\ x_2^n & x_2^{n-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (6.4)$$

where the X matrix is known as the Vandermonde matrix.

## 6.9 Least-square method

The least-square (LS) is a method for fitting a set of measured points with a mathematical model of the type  $y = f(p, x)$ . Where  $\{(y_i, x_i), i = 1, \dots, n\}$  is a set of measured points that could be represented in the Cartesian space,  $p$  is a vector of parameters that minimizes the sum of the squares of the difference between the observed data and the fitted mathematical model as expressed by the equation (6.5); often called *residual* or *measurement error*. In order to solve and determine the vector of parameters  $p$  that better fits the mathematical model, one should ensure that the main constraint on the number of equations — which better matches the measured set of points — has to be greater than the number of unknown variables.

$$J = \min \sum_{i=1}^n \varepsilon_i^2 = \min \sum_{i=1}^n (y_i - f(p, x_i))^2 \quad (6.5)$$

In case of ordinary least-square and linear road lines representation, the polynomial model order is set to  $n = 1$  and the equation (6.2) becomes  $y = p_1 x + p_0 = a x + b$ . This model fits our work in terms of real-time road lanes detection, lines parameters determination. The problem is

reduced to estimating two parameters  $\hat{a}$  and  $\hat{b}$ , corresponding to lines' slope and their intersection with Y-coordinate, respectively.

$$\hat{a} = \left( \left( \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \right) / \sum_{i=1}^n (x_i - \bar{x}) \right) \text{ and } \hat{b} = \bar{y} - \hat{a} \bar{x} \quad (6.6)$$

where  $(\bar{x}, \bar{y}) = \left( \frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i \right)$  is the center of the gravity of the measured set of the points.

LS method could be adapted to estimate more complex models for hyperbole shapes that were usually used to model sharp corners.

## 6.10 Kalman filter road lanes prediction and tracking

Road irregularity causes vehicles' vibrations, and when these vibrations are associated to vehicles' high speed, they shift and blur the road boundaries on the acquired images. This creates a blurred array that compromises the edge detection methods based on the image textures. To overcome such constraints, a Kalman filter (KF) is used to filter the measured road lane edge parameters and predict their future values in order to better target the position of the regions of interest, on each frame, corresponding to each line on the road. The vector parameter of each line, for instance the  $i^{\text{th}}$  line, and its associated ROI is given as follows:

$$\hat{x}_k^{iT} = \begin{pmatrix} \hat{a}_k^i & \hat{b}_k^i & \hat{x}_k^i & \hat{y}_k^i \end{pmatrix}^T \quad (6.7)$$

This reduces the size of the ROIs and the number of images pre-processing operations. Where  $\hat{a}_k^i$  and  $\hat{b}_k^i$  are the  $i^{\text{th}}$  line parameters.  $\hat{x}_k^i$  and  $\hat{y}_k^i$  form the center of the region of interest corresponding to  $i^{\text{th}}$  line at step  $k$  (a rectangle: defined by its center  $(\hat{x}_k^i, \hat{y}_k^i)$ , its width  $W$ , its height  $H$ , and its declination parameter  $\theta_k^i \cong \hat{a}_k^i$ , which coincides with the  $i^{\text{th}}$  line declination as shown in Figure 6-1). The linear time invariant (LTI) model that is considered herein is given by.

$$\begin{cases} x_{k+1}^i = F^i x_k^i + w_{m,k}^i \\ y_k^i = H^i x_k^i + w_{s,k}^i \end{cases} \quad (6.8)$$

$F^i$  and  $H^i$  are transition and measurement matrices, respectively.  $w_{m,k}^i$  and  $w_{s,k}^i$  are independent Gaussian noises corresponding to unmodeled errors and exogenous perturbations, and measurement noise, respectively.

$$\text{where } w_{m,k}^i \approx (0, Q_k^i), w_{s,k}^i \approx (0, R_k^i): E[w_{m,k}^i w_{m,k}^{i,T}] = Q_k^i > 0, E[w_{s,k}^i w_{s,k}^{i,T}] = R_k^i > 0,$$

$E[w_{m,k}^i w_{s,k}^{i,T}] = E[w_{s,k}^i w_{m,k}^{i,T}] = 0$  and  $P_k^i = E[(x_k^i - \hat{x}_k^i)(x_k^i - \hat{x}_k^i)^T]$  are covariance matrices at the step  $k$ . As known, the Kalman filter algorithm relies on two steps:

### 6.10.1 Prediction step

To estimate the future state vector according to the mathematical model and the last update step results.

$$\begin{cases} \hat{x}_{k+1|k}^i = F^i \hat{x}_{k|k}^i \\ P_{k+1|k}^i = F^i P_{k|k}^i F^{iT} + Q_k^i \\ P_{k+1|k}^i = E[(x_k^i - \hat{x}_{k+1|k}^i)(x_k^i - \hat{x}_{k+1|k}^i)^T] \end{cases} \quad (6.9)$$

### 6.10.2 Update step

To take in account the available measurements and update the observation gain and the covariance matrices and the state vector estimate, as follows.

$$\begin{cases} \hat{x}_{k|k}^i = \hat{x}_{k|k-1}^i + K_k^i (z_k^i - H^i \hat{x}_{k|k-1}^i) \\ K_k^i = P_{k|k-1}^i H^{iT} (H^i P_{k|k-1}^i H^{iT} + R_k^i)^{-1} \\ P_{k|k}^i = P_{k|k-1}^i - K_k^i H^i P_{k|k-1}^i \\ P_{k|k}^i = E[(x_k^i - \hat{x}_{k|k}^i)(x_k^i - \hat{x}_{k|k}^i)^T] \end{cases} \quad (6.10)$$

where  $z_k^i$  is the output vector (measurements) of the  $i^{\text{th}}$  line.

The following diagram summarizes the main steps of the proposed method to meet the real-time applications.

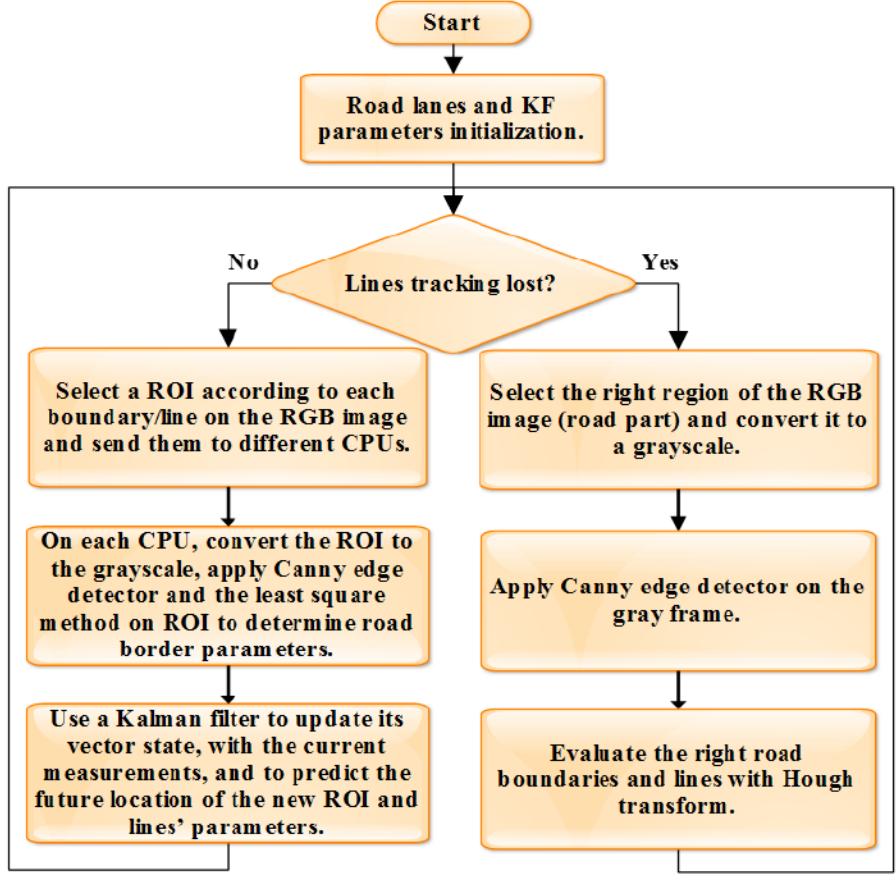


Figure 6-4: Road lanes detection and tracking diagram

## 6.11 Preliminary results

As described in section II, the road lane detection and tracking scenarios are designed on ProSiVIC. The environment and the test track are the “*Horse Ring*”. Two vehicles (red and blue) are created on the test track: the red one is equipped with a camera, INS, and programmed to follow the center of the right road lane at 70 km/h in the opposite direction of the black vehicle, which is also equipped by a vision system, like the red vehicle, which acquires 25 RGB frames per second of the size  $480 \times 640$  pixels. Due to the vehicles’ dynamic, some restrictions are imposed on the maximum steering angle, which is set to  $\delta_{\max} = 30^\circ$  degrees in each direction (left and right), and the maximum of torque of each wheel is set to  $\tau_{\max} = 350$  N.m.

To control the vehicle, we use three simple fuzzy logic laws that depend on the vehicle’s position versus the right road lane’s center. Once the road lanes are detected with the proposed method,

described just above (Figure 6-4), then the center of the correct road lane is estimated, represented by the dashed green line in Figure 6-5. According to the vehicle's position on the road, the fuzzy controller handles the vehicle's steering by generating and applying a steering reference angle to the vehicle's steering direction to keep and track the estimated road lane center. The same laws are carried out by the speed controller that compares the vehicle actual speed —thanks to the vehicles' model and the embedded inertial navigation system (INS) [112]— with the speed reference. Given the speed tracking error and the exogenous perturbations, the speed controller applies a torque control signal over the vehicle's wheels to control and set the vehicle's speed to the reference speed.

The simulation results confirm the efficiency and the robustness of the proposed method for real-time road line detection even at high speed and with discontinuous and dashed road lines.

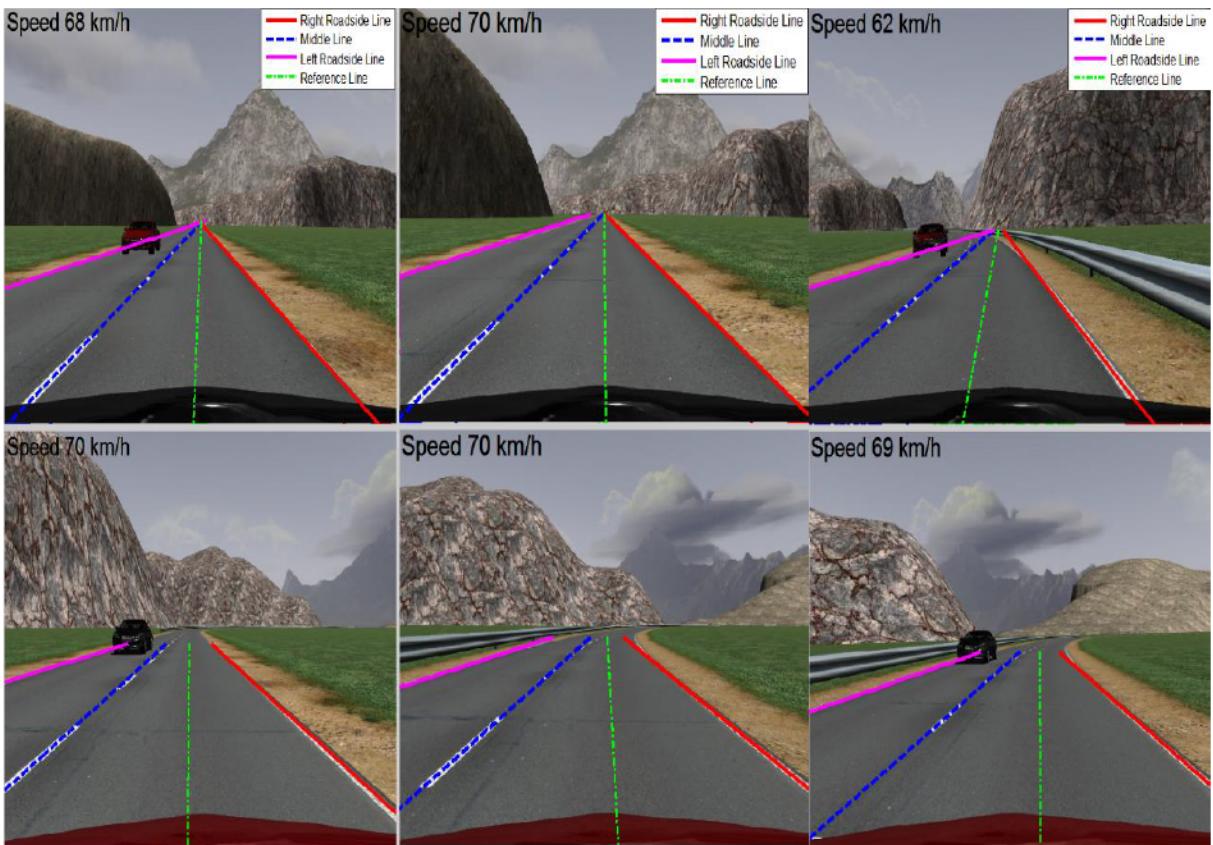


Figure 6-5: Real-time road line detection and road lanes center tracking

The black car runs along the test track where the right line gets thinner and thinner until it completely vanishes. Although the roadside line vanishes, the vehicle maintained and tracked the road lane center. The embedded algorithm successfully overcame the scenario constraints, dissolving lines, by distinguishing the different road textures. The steering fuzzy controller is limited in corners to 70 km/h: If the speed is higher, the autonomous vehicle is no longer able to keep driving over the road, particularly in the turns. This is due, mainly, to the approximated vehicle model errors, exogenous and unmodeled perturbations. To overcome such constraints, one should focus on the vehicle's mathematical model and lateral vehicle's controllers.

## 6.12 Conclusion

This work deals with road boundaries and painted lines detection for intelligent and autonomous vehicles. The purpose is to overcome the different constraints that could overwhelm the real-time road lanes detection and tracking. Those constraints are essentially due to the huge amount of data that should be processed especially: in case of high resolution images, vehicle's speed, vibrations. The preliminary results show that the proposed method is not only robust against exogenous perturbations and different constraints, but is good enough to control the vehicle with a couple of simple fuzzy logic laws. The vehicle's controller limitations arise for a maximum speed of 70 km/h in turns with some swaying in direction. Such issues will be overcome in the future work by suggesting a further controller. It will insure a better vehicle control, by fusing road lane parameters with the vehicle's dynamic to better predict the road boundaries, in particular cases: completely obscured boundaries and lines by other vehicles, and road intersections.

# **Chapitre 7**

## **Real time cooperative localization for autonomous vehicles**

### **7.1 Avant-propos**

#### **7.1.1 Auteurs et affiliation**

**Farid Bounini :** Étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Denis Gingras :** Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Hervé Pollart :** responsable du développement commercial de la voiture intelligente, OPAL-RT Technologies inc. Lyon, France.

**Dominique Gruyer :** Chargé de recherche, IFSTTAR, CoSys — LIVIC, Versailles, France.

#### **7.1.2 Date de soumission**

22 janvier 2016.

#### **7.1.3 État de l’acceptation**

Version finale publiée.

#### **7.1.4 Conférence**

IEEE 19<sup>th</sup> International Conference on Intelligent Transportation Systems (ITSC).

### **7.1.5 Référence**

BOUNINI F., GINGRAS D., POLLART H., GRUYER D, “Real Time Cooperative Localization for Autonomous Vehicles.” IEEE 19<sup>th</sup> International Conference on Intelligent Transportation Systems (ITSC 2016), Rio Brazil, November 1<sup>st</sup> – 4<sup>th</sup>, 2016.

### **7.1.6 Titre français**

Localisation coopérative en temps réel pour les véhicules autonomes.

### **7.1.7 Contribution au document**

Ce chapitre démontre la capacité de la plateforme de simulation des systèmes d'aide à la conduite et des véhicules intelligents, autonomes et connectés en conduite collaborative, à supporter et à gérer les scénarios multivéhiculaires en temps réel. Une nouvelle stratégie de localisation coopérative est aussi proposée et implémentée sur le simulateur. La méthode exploite les librairies et les outils Matlab/Simulink pour résoudre le problème d'optimisation de l'algorithme de l'intersection des covariances, qui sera défini dans ce présent chapitre, montrant ainsi l'adaptabilité de la plateforme à exploiter les ressources et les développements algorithmiques, faits à priori sur des logiciels riches et performants (Matlab/Simulink, Python, C/C++, etc.).

### **7.1.8 Résumé français**

Ce chapitre décrit une nouvelle stratégie de localisation coopérative en temps réel pour les véhicules autonomes. La stratégie vise à améliorer la précision de la localisation véhiculaire et à réduire l'impact du temps de calcul des algorithmes de fusion de données multi-capteur. Ceci en ayant recours à la communication inter-véhiculaire (véhicule à véhicule) sur des architectures parallèles. Ce travail vise à résoudre le problème de la localisation d'un groupe de véhicules autonomes, équipés de systèmes de navigation à faible coût et circulant dans un environnement inconnu. La solution se base sur l'une des dérivées du filtre de Kalman pour estimer le vecteur d'état, associé aux modèles non linéaires des véhicules, dans un nœud de fusion de données local. Au moyen de la communication inter-véhiculaire, les véhicules échangent les estimées de leurs modèles d'état. La fusion des estimées locales se fait dans un deuxième nœud de fusion, dit «nœud de fusion de données global », avec l'algorithme de l'intersection des covariances.

Cette méthode exploite simultanément un capteur proprioceptif, un système de positionnement global GPS, un capteur extéroceptif, un télémètre pour détecter les objets et mesurer la distance et l'angle relatifs entre les véhicules et leurs alentours, et finalement les moyens de communication, un émetteur et un récepteur, pour une localisation collaborative précise et fiable.

### **7.1.9 Modification**

Une erreur de dérivation de la fonction de transition s'est glissée lors du calcul de la matrice d'état «F» dans la version publiée de cet article (Chapitre 7, section 7.4). Toutefois, elle est corrigée dans ce chapitre.

## 7.2 Abstract

This paper describes a new strategy for real-time cooperative localization for autonomous vehicles. The strategy aims to improve the vehicles localization accuracy and reduce the impact of computing time of multi-sensor data fusion algorithms through vehicle-to-vehicle communication on parallel architectures. The method aims to solve localization issues in a cluster of autonomous vehicles, equipped with low-cost navigation systems in an unknown environment. It stands on multiple forms of the Kalman filter derivatives to estimate the vehicles' nonlinear model vector state in local fusion nodes. Then, the vehicles exchange their local state estimate. The Covariance Intersection algorithm merges the local vehicles' state estimates in a second node, named global data fusion node. This strategy simultaneously exploits the proprioceptive and sensors — a Global Positioning System, a Vehicle-To-Vehicle transmitter and receiver — and an exteroceptive sensor, rangefinder, to sense their surroundings for more accurate and reliable collaborative localization.

## 7.3 Introduction

The interest in Advanced Driving Assistance Systems (ADAS) has mainly been driven by the sheer increase in autonomy, cheap and complex sensors, computer science and power, and embedded technology. As a result, the automotive industry is expanding more and more all around the world. ADAS are mushrooming and enhancing vehicles' active safety. Localization issues, in unknown and in uncertain environments, are still a big challenge for ADAS, even with accurate and expensive embedded local positioning sensors. This is primarily due to navigation derivative issues, which are related to sensors' noise measurements. Thus, the error in the position estimate, based on dead reckoning, continuously grows [123] that is a serious challenge for autonomous vehicle navigation. If one relies on Global Navigation Satellite Systems (GNSSs) — such as the American GPS (Global Positioning System), the Russian GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema), or the European Galileo — to eliminate dead reckoning issues, it is necessary to periodically reset the local positioning sensors.

However, the GNSSs' localization accuracy, in active safety and in autonomous vehicles, still needs to be improved. As a result, the major aspect of ADAS for intelligent vehicles is still an

accurate and a reliable localization [124]. Cooperative navigation architecture schemes — in a cluster of “N” smart vehicles — are becoming an alternative solution for localization issues and a relevant application in autonomous vehicles. In cooperative scenarios, each vehicle is equipped with proprioceptive sensors (Inertial Navigation System (INS): gyros, accelerometers, odometers, etc.) and exteroceptive sensors (Radio Detection And Ranging (RADAR), LAser Detection And Ranging (LADAR), camera, GPS, etc.). According to data fusion of sensors measurements, it is possible to build a consistent estimate of both the numerical environment map and the vehicle trajectory. This technique is called Simultaneous Localization and Mapping (SLAM) [123]. For more accuracy, the vehicles broadcast their relevant information and their position, bearing angles, speed, attention, etc., and their environment information to their neighbors. They use the shared information to improve their own and their neighbors’ position estimate.

Extensive research has been done in the literature on target localization according to static references by creating a Long Baseline (LBL) positioning system [123], for robots’ indoor and underwater applications. For instance, in their project, Borges M et al. [123] were faced with some communication constraints of the underwater environment, such as GPS limitations in case if the Autonomous Underwater Vehicles (AUV) are submerged. A basic solution was to employ static beacons in well-known positions. This solution reduces the AUVs localization area to the beacons’ long range. To get around static beacon issues, they resorted to cooperative autonomous underwater vehicle localization. The strategy is based on a communicating cluster of AUVs with the following shortcomings: unreliable communication and low bandwidth. The AUVs considered each other as a moving landmark with an uncertainty associated to their position, which is susceptible to change in time.

For the outdoor ones, Rohani M and al. [124] developed a new decentralized Bayesian approach for cooperative vehicle localization based on data fusion of the GPS and Vehicular Ad-hoc Network (VANET) inter-vehicle distance measurement. The vehicles were able to ascertain the inter-vehicular distance independently from their measurements according to the VANET based method, in a cluster of five vehicles. The objective was to reduce the vehicles’ localization uncertainties by overcoming the GPS drawbacks. The data synchronization of the embedded sensors, which were involved in the simulation scenario, was performed using the GPS time as

a reference time. Each vehicle shared its relevant data with the other vehicles in the cluster. Morbidi F et al. [125] also proposed a new approach for a more accurate localization state estimate of a cluster of Unmanned Aerial Vehicles (UAVs) in 3D space. It is also worth emphasizing herein that it is very important to have a simulation platform to simulate ADAS, intelligent and autonomous vehicles behavior ad depicted in [112]-[113], [126].

This work moves toward real-time collaborative localization regarding specific multiple vehicles' scenarios on parallel architectures and to demonstrate its performance. It elaborates on [128]. The paper is partitioned into six sections. The first section presents some related work on an urban navigation system using GPS. The second section deals with a simple non-linear model of the vehicle in 2D. The third and fourth sections involve the vehicle self-position estimate, and its neighbors' position estimate, according to their respective pre-defined nonlinear model, to the proprioceptive and exteroceptive measurements and to an Extended Kalman Filter (EKF). The fifth section describes a solution for cooperative localization. It solves the over-convergence issue, which usually occurs when the merged data coming from different targets, are dependent on each other. In the penultimate sections — which talks about some preliminary results — specific multi-vehicular scenarios are built and executed on the Matlab/Pro-SiVIC, which could also be simulated on RT-LAB/Pro-SiVIC real time simulation platform. For more information about the platform the reader is referred to [113]-[114] and [127]. The paper is then concluded, and the future works are highlighted.

## 7.4 Vehicle non-linear model

The rudimentary purpose behind modeling vehicle dynamics is to exploit linear state estimate and control theories — command and observation — to design signal control and to govern the vehicle. Real systems are generally characterized by their non-linear behavior, as referred in (7.1). For the purpose of simplicity and not to encumber the paper with mathematical equations, the vehicle is considered as a moving object in limited 2D plan, as shown in Figure 7-1.

$$\begin{cases} x_{k+1}^v = f(x_k^v, u_k^v, w_{m,k}^v) \\ z_k^v = h(x_k^v) + w_{s,k}^v \end{cases} \quad (7.1)$$

where  $x^v = [x \ y \ \theta \ v \ a \ \omega]^T$  is the state vector of the vehicle:  $x$  and  $y$  are the vehicle coordinates in the 2D global or Cartesian plan.  $\theta$  and  $v$  are the heading angle and longitudinal speed of the vehicle,  $a$  and  $\omega$  are the vehicle longitudinal acceleration and yaw angular speed, respectively. The control input  $u^v$  holds the external forces that are assumed, for this paper, to be unknown.  $f(*)$  is a state/transition non-linear function. It defines the system's dynamic behavior, according to its previous state, the control inputs and the exogenous perturbations which include the unmodeled components of the system. The observation vector  $z^v$  incorporates the sensed quantities (position, bearing angle and speed) by the embedded sensor (GPS receiver).  $h(*)$  is the observation function. It details the relationship between the measured state variables.  $w_m^v = [x^w \ y^w \ \theta^w \ v^w \ a^w \ \omega^w]^T \approx N(0, Q)$  and  $w_s^v \approx N(0, R)$  are Gaussian random vectors, which consolidate the unmodeled component of the vehicle and nonlinear model linearization error, and measurement noise due to sensors with covariance  $Q$  and  $R$ , respectively.  $T$  is the step time (time lapse between two updates).

$$\begin{cases} f(x_k^v, u_k^v, w_k^v) = \begin{bmatrix} x_k + x_k^w + T \left( v_k + v_k^w + \frac{T}{2} (a_k + a_k^w) \right) \cos(\theta_k + \theta_k^w) \\ y_k + y_k^w + T \left( v_k + v_k^w + \frac{T}{2} (a_k + a_k^w) \right) \sin(\theta_k + \theta_k^w) \\ \theta_k + T (\omega_k + \omega_k^w) + \theta_k^w \\ v_k + T (a_k + a_k^w) + v_k^w \\ a_k + a_k^w \\ \omega_k + \omega_k^w \end{bmatrix} \\ h(x_k^v, u_k^v) = [x_k \ y_k \ \theta_k \ v_k]^T \end{cases} \quad (7.2)$$

In order to benefit from the advanced state estimate strategies (of continuous and discrete state space) of linear systems, the vehicle nonlinear model will be approximated by its equivalent linear model at each step time  $k$ . The Jacobian of the non-linear functions  $f(*)$  and  $h(*)$  are respectively computed. Then, an approximation of the non-linear model is computed at time  $k$ .

$$\begin{cases} x_{k+1}^v = F_k^v x_k^v + G_k^v u_k^v + M_k^v w_{m,k}^v \\ y_k^v = H_k^v x_k^v + w_{s,k}^v \end{cases} \quad (7.3)$$

where:

$$F_k^v = \frac{\partial f}{\partial x^v} \Big|_{w_k^v} = \begin{bmatrix} 1 & 0 & -T \left( v_k + \frac{T}{2} a_k \right) \sin(\theta_k) & T \cos(\theta_k) & \frac{T^2}{2} \cos(\theta_k) & 0 \\ 0 & 1 & T \left( v_k + \frac{T}{2} a_k \right) \cos(\theta_k) & T \sin(\theta_k) & \frac{T^2}{2} \sin(\theta_k) & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_k^v = \frac{\partial f}{\partial u^v} \Big|_{w_k^v} = 0_{6 \times 1}, M_k^v = \frac{\partial f}{\partial w^v} \Big|_{x_k^v, w_k^v=0} = F_k^v,$$

$$H_k^v = \frac{\partial h}{\partial x^v} \Big|_{x_k^v} = [I_{4 \times 4} \quad 0_{4 \times 2}],$$

$$Q_k^{v_m} = E \left[ (w_m^v) (w_m^v)^T \right] = \text{diag} \left[ \sigma_{x_{v_m}}^2 \quad \sigma_{y_{v_m}}^2 \quad \sigma_{\theta_{v_m}}^2 \quad \sigma_{v_{v_m}}^2 \quad \sigma_{a_{v_m}}^2 \quad \sigma_{\alpha_{v_m}}^2 \right],$$

$$R_k^{v_{GPS}} = E \left[ (w_s^v) (w_s^v)^T \right] = \text{diag} \left[ \sigma_{x_{v_s}}^2 \quad \sigma_{y_{v_s}}^2 \quad \sigma_{\theta_{v_s}}^2 \quad \sigma_{v_{v_s}}^2 \right]$$

## 7.5 Vehicle neighbors position estimate

Cooperative localization could be a good solution to sensors' gaps. It would allow automakers to equip their products with several inexpensive embedded proprioceptive and exteroceptive sensors. The idea is to equip each vehicle with a GPS, a telemeter (lidar "Light Detection and Ranging"), and with the means of inter-vehicular communication (a transmitter and a receiver). The proprioceptive sensors (odometers, INS and GPS) provide the vehicle its own ego vector state measurement, which will be exploited to predict and estimate their future values. The exteroceptive sensors allow the vehicle to sense its surrounding environment to plan its trajectory and avoid static and moving obstacles. Thus, the rangefinder provides the relative distance separating the two vehicles (inter-vehicular distance)  $d_k^{v_i}$ , and the relative bearing angle  $\theta_k^{v_i}$ . The state model is then defined by (7.4) according to Figure 7-2.

$$\begin{cases} x_{k+1}^{v_{j/i}} = f^{v_{j/i}} (x_k^{v_i}, x_k^{v_{j/i}}, w_{m,k}^{v_{j/i}}) = \begin{bmatrix} d_k^{v_i} + T \delta d_k^{v_i} \\ \theta_k^{v_i} + T \delta \theta_k^{v_i} \\ \delta d_k^{v_i} \\ \delta \theta_k^{v_i} \\ x_k^{v_i} + d_k^{v_i} \cos(\theta_k^{v_i} - \theta_k^{v_j}) \\ y_k^{v_i} + d_k^{v_i} \sin(\theta_k^{v_i} - \theta_k^{v_j}) \end{bmatrix} \\ z_k^{v_{j/i}} = h^{v_{j/i}} (x_k^{v_{j/i}}, w_{s,k}^{v_{j/i}}) = \begin{bmatrix} d_k^{v_i} & \theta_k^{v_i} \end{bmatrix}^T \end{cases} \quad (7.4)$$

where:

$x_{k+1}^{v_{j/i}} = \begin{bmatrix} d_k^{v_i} & \theta_k^{v_i} & \delta d_k^{v_i} & \delta \theta_k^{v_i} & x_k^{v_j} & y_k^{v_j} \end{bmatrix}^T$ ,  $d_k^{v_i} = \sqrt{(x_k^{v_j} - x_k^{v_i})^2 + (y_k^{v_j} - y_k^{v_i})^2}$  and  $\delta d_k^{v_i}$  are the relative distance and speed.  $\theta_k^{v_i}$  and  $\delta \theta_k^{v_i}$  are the relative bearing angle and its variation in time.

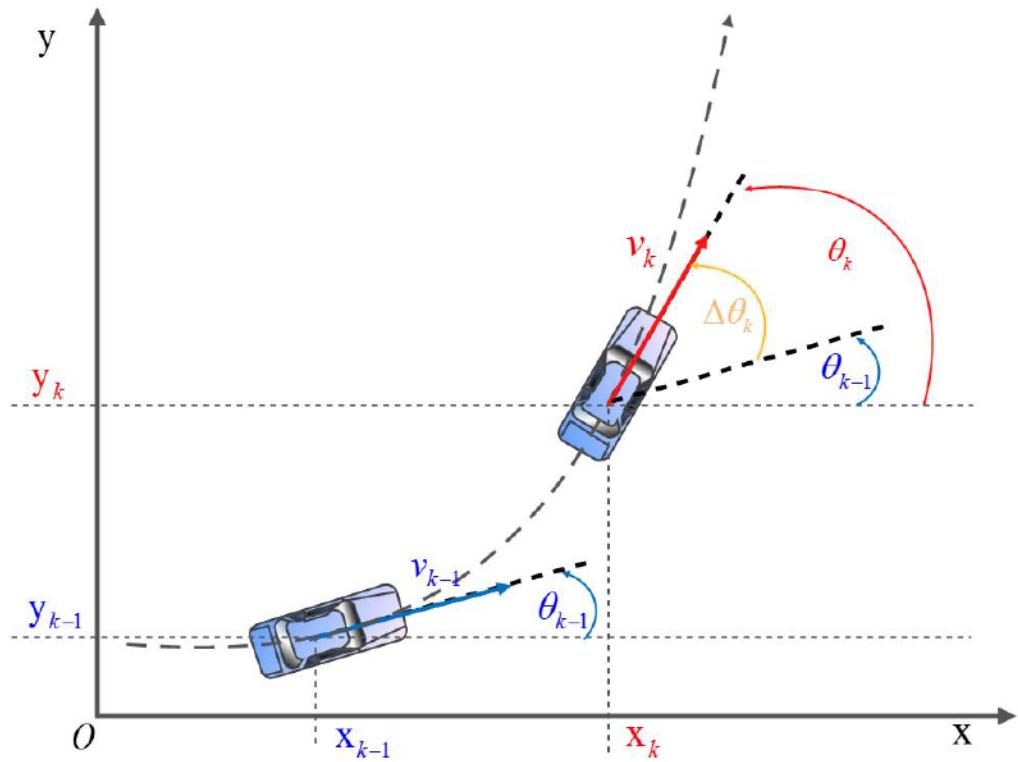


Figure 7-1: Vehicle's model in 2D

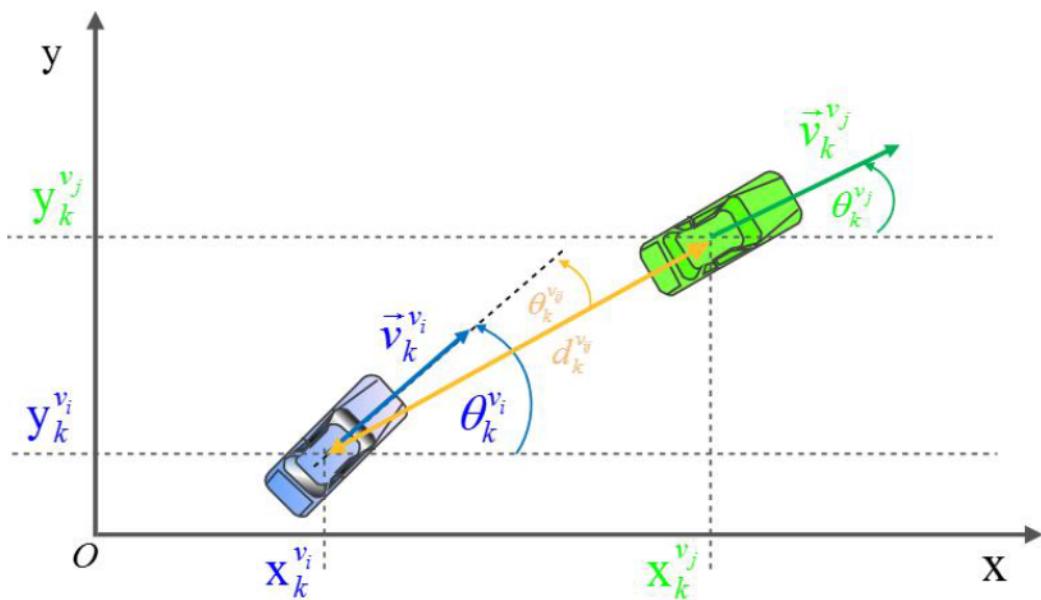


Figure 7-2: Inter-vehicular distance and bearing angle

$$\begin{aligned}
F_k^{v_{(j,i)}} &= \frac{\partial f^{v_{j/i}}(x_k^{v_{j/i}}, w_{m,k}^{v_{j/i}})}{\partial x_k^{v_i}} \Bigg|_{\substack{x_k^{v_i} \\ w_{m,k}^{v_{j/i}} = 0}} = \begin{bmatrix} 0_{4 \times 3} \\ I_{2 \times 2} \\ -d_k^{v_j} \sin(\theta_k^{v_i} - \theta_k^{v_j}) \\ d_k^{v_j} \cos(\theta_k^{v_i} - \theta_k^{v_j}) \end{bmatrix}_{0_{2 \times 3}}, \quad F_k^{v_{j/i,j/i}} = \frac{\partial f^{v_{j/i}}(x_k^{v_{j/i}}, w_{m,k}^{v_{j/i}})}{\partial x_k^{v_{j/i}}} \Bigg|_{\substack{x_k^{v_{j/i}} \\ w_{m,k}^{v_{j/i}} = 0}} = \begin{bmatrix} I_{2 \times 2} & II_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & I_{2 \times 2} & 0_{2 \times 2} \\ F_k^{v_{\bar{j}}} & 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix}, \\
H_k^{v_{j/i}} &= \frac{\partial h^{v_{j/i}}(x_k^{v_{j/i}}, w_{s,k}^{v_{j/i}})}{\partial x_k^{v_{j/i}}} \Bigg|_{\substack{x_k^{v_{j/i}} \\ w_{s,k}^{v_{j/i}} = 0}} = [I_{2 \times 2} \quad 0_{2 \times 4}], \quad F_k^{v_{\bar{j}}} = \begin{bmatrix} \cos(\theta_k^{v_i} - \theta_k^{v_{\bar{j}}}) & d_k^{v_{\bar{j}}} \sin(\theta_k^{v_i} - \theta_k^{v_{\bar{j}}}) \\ \sin(\theta_k^{v_i} - \theta_k^{v_{\bar{j}}}) & -d_k^{v_{\bar{j}}} \cos(\theta_k^{v_i} - \theta_k^{v_{\bar{j}}}) \end{bmatrix}
\end{aligned}$$

and  $\theta_k^{v_{\bar{j}}} = \text{atan}2((y_k^{v_j} - y_k^{v_i}), (x_k^{v_j} - x_k^{v_i}))$ .

To get a complete model, the two vehicles' mathematical models are concatenated and form together an extended model (7.5).

$$\begin{cases} x_{k+1}^{v_{ex}} = \begin{bmatrix} x_{k+1}^{v_i} & x_{k+1}^{v_{j/i} T} \end{bmatrix}^T = F_k^{v_{ex}} x_k^{v_{ex}} + G_k^{v_{ex}} u_k^{v_{ex}} + M_k^{v_{ex}} w_{m,k}^{v_{ex}} \\ z_k^{v_{ex}} = H^{v_{ex}} x_k^{v_{ex}} + w_{s,k}^{v_{ex}} \end{cases} \quad (7.5)$$

where:

$$F_k^{v_{ex}} = \begin{bmatrix} F_k^v & 0_{6 \times 6} \\ F_k^{v_{(\bar{j},j/i)}} & F_k^{v_{(j/i,j/i)}} \end{bmatrix}, \quad G_k^{v_{ex}} = \begin{bmatrix} G_k^v & 0_{6 \times 2} \\ 0_{6 \times 2} & G_k^{v_{j/i}} \end{bmatrix}, \quad H_k^{v_{ex}} = \begin{bmatrix} H_k^v & 0_{4 \times 6} \\ 0_{2 \times 6} & H_k^{v_{j/i}} \end{bmatrix}, \quad M_k^{v_{ex}} = \begin{bmatrix} M_k^v & 0_{6 \times 6} \\ 0_{6 \times 6} & M_k^{v_{j/i}} \end{bmatrix},$$

$$Q_k^{v_{ex}} = \begin{bmatrix} Q_k^{v_{w_m}} & 0_{6 \times 6} \\ 0_{6 \times 6} & Q_k^{v_{w_m}} \end{bmatrix}, \quad R_k^{v_{ex}} = \begin{bmatrix} R_k^{v_{INS/GPS}} & 0_{4 \times 2} \\ 0_{2 \times 4} & R_k^{v_{Lidar}} \end{bmatrix}, \quad R_k^{v_{Lidar}} = E \left[ (w_{Lidar}^v) (w_{Lidar}^v)^T \right] = \text{diag} \left[ d_{w_{Lidar}^v}^2 \quad \theta_{w_{Lidar}^v}^2 \right]$$

$$\text{and } w_{m,k}^{v_{ex}} = \begin{bmatrix} w_{m,k}^{v_i,T} & w_{m,k}^{v_{j/i},T} \end{bmatrix}^T \text{ and } w_{s,k}^{v_{ex}} = \begin{bmatrix} w_{INS/GPS,k}^{v_i,T} & w_{Lidar,k}^{v_i,T} \end{bmatrix}^T.$$

## 7.6 Extended Kalman filter and state estimate

Over the past half-century, Kalman filter has been one of the most popular data fusion algorithms in literature and in embedded applications [131]. This is especially due to its optimality, in the case of linear systems, and particularly to its recursive form and its low-cost computation resource requirements [74]-[78] and [132]. In the case of a nonlinear system, an Extended Kalman Filter (EKF) is often deployed. The EKF offers the ability to synchronize sensors' data, which are acquired asynchronously, while it receives the exteroceptive and proprioceptive sensors measurement (odometers, INS, GPS, telemeters). The data fusion algorithm (EKF) is fashioned by two steps.

- Prediction step: to estimate the future covariance matrix and state vector according to the mathematical model and the last update step results and to the control inputs ;

$$\begin{cases} \hat{x}_{k|k-1}^{v_{\alpha}} = F_{k-1}^{v_{\alpha}} \hat{x}_{k-1|k-1}^{v_{\alpha}} + G_k^{v_{\alpha}} u_k^{v_{\alpha}} \\ P_{k|k-1}^{v_{\alpha}} = F_{k-1}^{v_{\alpha}} P_{k-1|k-1}^{v_{\alpha}} F_{k-1}^{v_{\alpha} T} + M_k^{v_{\alpha}} Q_k^{v_{\alpha}} M_k^{v_{\alpha} T} \\ P_{k|k-1}^{v_{\alpha}} = E \left[ (x_k^{v_{\alpha}} - \hat{x}_{k|k-1}^{v_{\alpha}}) (x_k^{v_{\alpha}} - \hat{x}_{k|k-1}^{v_{\alpha}})^T \right] \end{cases} \quad (7.6)$$

- Update step: to include the available measurements and to update the predicted the covariance matrix and state vector during the prediction step.

$$\begin{cases} \hat{x}_{k|k}^{v_{\alpha}} = \hat{x}_{k|k-1}^{v_{\alpha}} + K_k^{v_{\alpha}} (z_k^{v_{\alpha}} - H_k^{v_{\alpha}} \hat{x}_{k|k-1}^{v_{\alpha}}) \\ K_k^{v_{\alpha}} = P_{k|k-1}^{v_{\alpha}} H_k^{v_{\alpha} T} (H_k^{v_{\alpha}} P_{k|k-1}^{v_{\alpha}} H_k^{v_{\alpha} T} + R_k^{v_{\alpha}})^{-1} \\ P_{k|k}^{v_{\alpha}} = L_k^{v_{\alpha}} P_{k|k-1}^{v_{\alpha}} L_k^{v_{\alpha} T} + K_k^{v_{\alpha}} R_k^{v_{\alpha}} K_k^{v_{\alpha} T} \\ L_k^{v_{\alpha}} = I - K_k^{v_{\alpha}} H_k^{v_{\alpha}} \\ P_{k|k}^{v_{\alpha}} = E \left[ (x_k^{v_{\alpha}} - \hat{x}_{k|k}^{v_{\alpha}}) (x_k^{v_{\alpha}} - \hat{x}_{k|k}^{v_{\alpha}})^T \right] \end{cases} \quad (7.7)$$

## 7.7 Inter-vehicular communication and cooperative localization

The inter-vehicular communication system allows the equipped vehicles to exchange their own and their environment's state estimate with their neighbors, as in Figure 7-3. The main purpose of such applications is to improve cooperative localization and accuracy. The communication delay, in this work, is assumed to be equal to zero. Considering two pieces of a signal;  $x_1$  and  $x_2$  are provided from two different sources of information; they could be two sensors, databases, intelligent vehicles, etc., and assume that these two pieces are noisy. It is necessary to fuse them for more accuracy, because they are not known precisely. If the two signals are independent, then  $P_{x1x2}=0$ , so the fusion is often performed with an EKF. However, the pieces are usually dependent. In other words, their cross-covariance differs from zero  $P_{x1x2} \neq 0$  and  $P_{x2x1} \neq 0$  [129]. Consequently, the only information, that could be used from the two signals, is their mean ( $\bar{x}_1$ ,  $\bar{x}_2$ ), and their covariance ( $P_{x1}$ ,  $P_{x2}$ ), respectively. The unidentified cross-covariance induce the EKF divergence [129]. The Covariance Intersection (CI) is specifically developed to overcome

such problems. The only constraint imposed on the assumed estimate is consistency [85] as expressed by the following equation (7.8).

$$P_{x_1} - P_{e_{x_1}} \geq 0, \quad P_{x_2} - P_{e_{x_2}} \geq 0 \quad \text{and} \quad P_x - P_{e_x} \geq 0 \quad (7.8)$$

where  $e_{x_1} = x_1 - \bar{x}_1$ ,  $e_{x_2} = x_2 - \bar{x}_2$ ,  $e_x = x - \bar{x}$ , and  $x$  is the fusion result of the two pieces of the original signal with a mean  $\bar{x}$  and covariance  $P_x$ .

The principle of covariance intersection algorithms is to find  $\omega$  that minimizes the trace or the determinant of  $P_x$  such as:

$$\begin{cases} P_x^{-1} = \omega P_{x_1}^{-1} + (1 - \omega) P_{x_2}^{-1} \\ P_x^{-1} x = \omega P_{x_1}^{-1} x_1 + (1 - \omega) P_{x_2}^{-1} x_2 \end{cases} \quad (7.9)$$

The general form of the covariance intersection algorithm, in this case, is given by the following equation:

$$\begin{cases} \hat{x}_{k/k}^{v_\alpha} = P_{k/k}^{v_\alpha} \left( \sum_{i=1}^n \omega_i \left( P_{k/k}^{v_{ex,i}} \right)^{-1} \hat{x}_{k/k}^{v_{ex,i}} \right) \\ P_{k/k}^{v_\alpha,-1} = \sum_{i=1}^n \omega_i \left( P_{k/k}^{v_{ex,i}} \right)^{-1} \\ \sum_{i=1}^n \omega_i = 1 \end{cases} \quad (7.10)$$

Several sources of common errors, in distributed architecture, need to be removed each time the fusion node receives a new update [130]. Otherwise, they will cause over-covariance (data inbreeding), which is another novel challenge to overcome in collaborative vehicles scenarios. It describes a situation where the analysis covariance errors are systematically underestimated after each observation [85] and [130]. After the second observation, the fusion nodes use almost the same inbreeding data. As a consequence, the data fusion is no longer efficient. To get around over-covariance issues, local and global fusion nodes are created. The local-one fuses the non-correlated data; which comes from the sensors, according to the Extended Kalman Filter. The global node fuses the exchanged data separately and independently and communicates the relevant information. Each vehicle sends the results of the local fusion node to its neighbors, and uses the results of the global nodes to accomplish its purposes: trajectory planning, collision assessment, vehicle dynamic control, etc.

## 7.8 Simulation results

To evaluate the efficiency of the new strategy, a multi-vehicular scenario — a cluster of three intelligent vehicles — is developed on the real-time simulator of collaborative and autonomous vehicles. It combines two platforms [127]. The first one is Pro-SiVIC, which is a real-time simulator of a vehicle's dynamic, embedded sensors and infrastructure. It allows the user to build multi-vehicular scenario and equip each one with different proprioceptive and exteroceptive embedded sensors to acquire data about the vehicle's dynamic and its environment. The second platform is RT-LAB, which processes the required data on distributed architectures and in real-time.

To produce sufficient data, each vehicle is equipped with a telemeter, an exteroceptive sensor, their respective covariance matrices are given in Table 7-1. They are also equipped with a V2V (Vehicle-To-Vehicle) communication device. The communication channels are assumed to be ideal, the transmission delay is equal to zero. An illustrative multi-vehicular configuration scenario is shown in Figure 7-3, where the red vehicle can detect and measure the inter-vehicular distances and the relative heading angles between itself and the other two vehicles. The blue vehicle can measure only the relative distance and heading angle that keeps it apart from the green one. However, the last one is not able to measure any relative distance. But, thanks to the V2V communication receiver, the green vehicle receives those relative distances and the other vehicles' position and their respective relative error estimate. The strategy was directly implemented on Matlab. The Pro-SiVIC data exchange mode is set to Matlab to exchange and process sensor data. The black ellipses, in Figure 7-3, correspond to the GPS position estimate of each vehicle with its own relative error position according the GPS data.

The EKF allows each vehicle to locate its visible neighbors with the same relative position error as in the result of the inboard sensors (GPS and telemeter) data fusion, as shown in the same figure with thin dashed red, blue and green ellipses according to each vehicle. The global data fusion, with inter-vehicular communication results, markedly, reduces the positioning uncertainty and makes the positioning error decrease. For instance, at the global node of the green vehicle, the rate of position error decreased until thirty and forty-five percent on the X and Y axis, respectively, as shown in Figure 7-5. By minimizing the trace of the three covariance matrices.

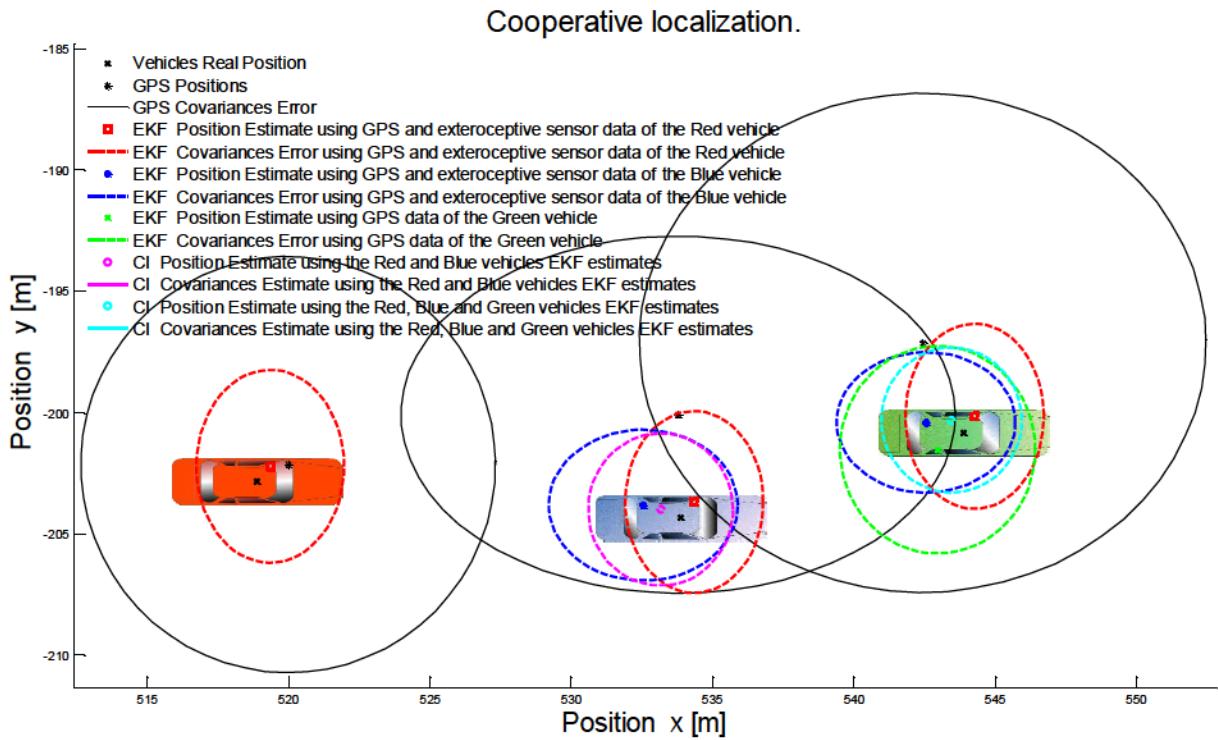


Figure 7-3: Cooperative localization with three vehicles

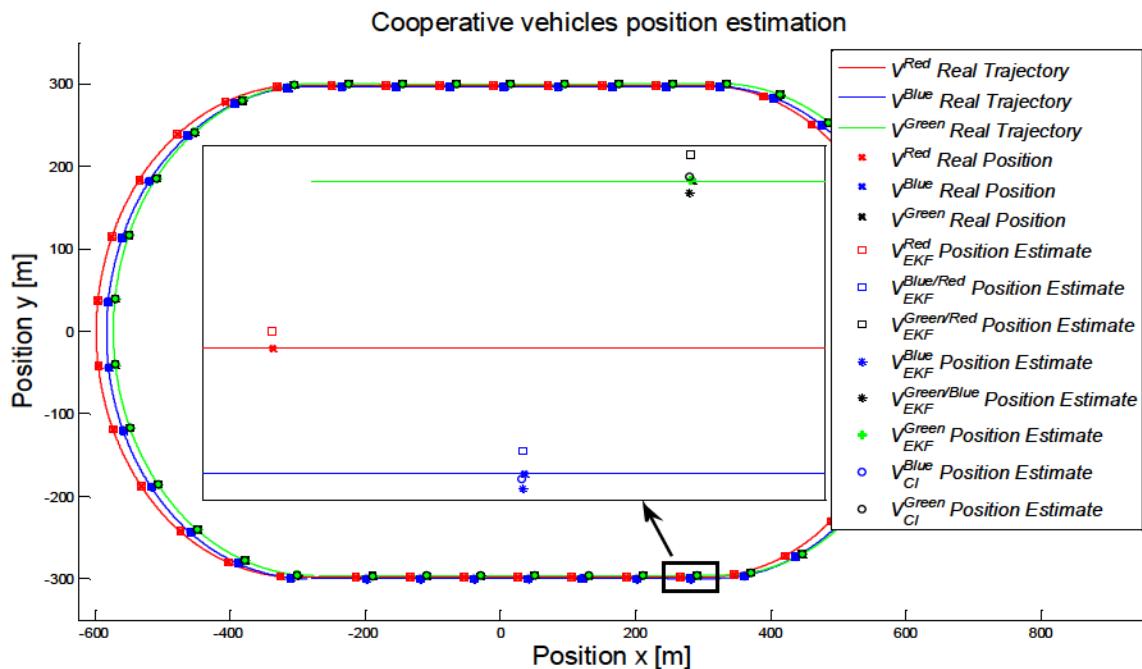


Figure 7-4: Vehicles position estimate in cooperative localization scenario.

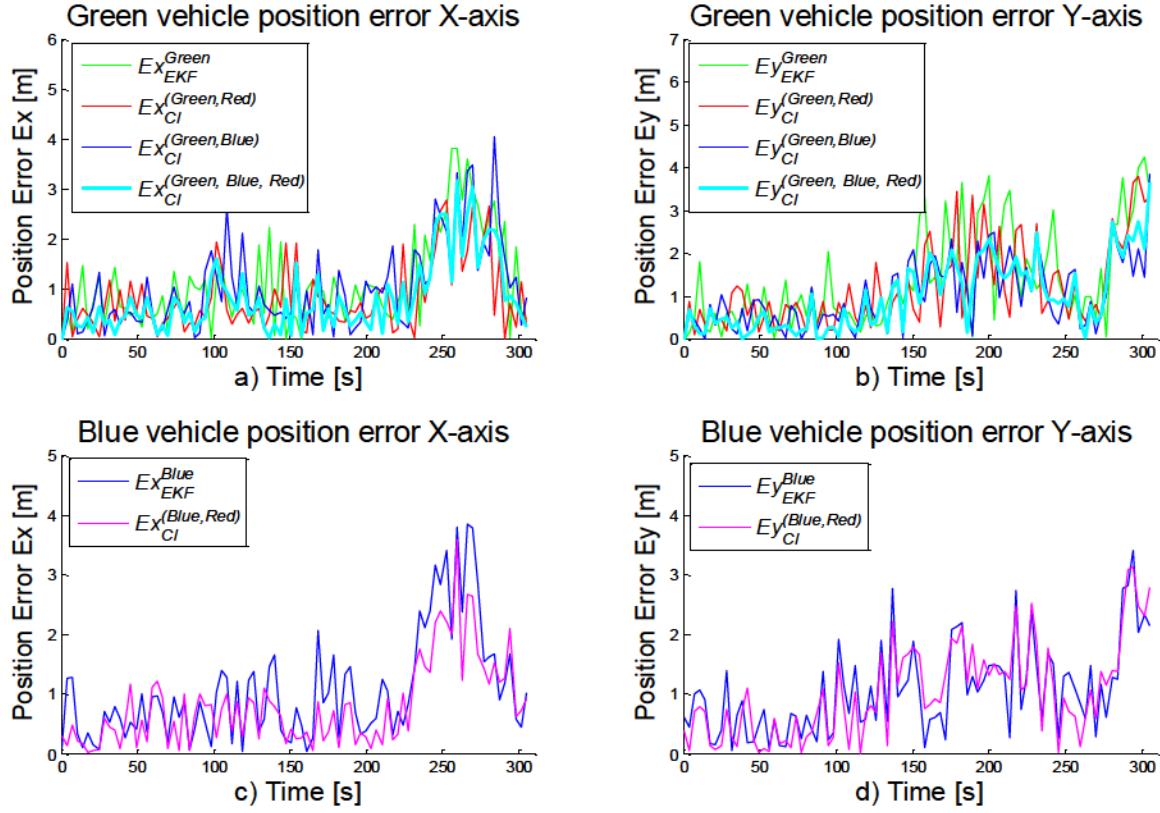


Figure 7-5: Absolute error on the vehicle positioning on X and Y axis

The CI result, as shown in Figure 7-3–5, reduces the position error in the two X and Y directions according to the optimal values of  $\omega_i$ .

Nevertheless, the ratio of the covariance error ( $\sigma_x^2/\sigma_y^2$ ) is decreasing when cooperative localization method is performed. In other words, the amount of variance reduction is greater in the X rather than in the Y direction. This is due mainly to vehicles moving direction, as it is shown in Figure 7-4, the vehicles move twice longer in the X rather than in the Y axis. Then it oscillates around that ratio according the vehicle motion in x or in y direction. Since the precision of red and blue cars' sensors are almost the same, the improvements on the blue vehicle position are not so notable in the Y axis and only increases about ten percent in the X-axis. As a consequence of the simulated scenario, the global nodes allowed the vehicles to benefit from their neighbors sensors precision, as shown in blue and particularly in the green vehicle global node. This performed by minimizing the trace of the matrix covariance error in the X and the Y axis according to the red and blue vehicles' sensors precision. The result of the

covariance intersection presented herein by the magenta and the cyan ellipse respectively. The simulation results showed that the CI covariance errors of each vehicle vary according to the number of vehicles that are involved in the simulated scenario and their sensors' accuracy. The simulation also revealed that, to increase the robustness and reliability, the method should be adaptable to any configuration of the cluster, and take into account the appearing and vanishing vehicles, especially at the local data fusion node, where the inter-vehicular distance is merged with proprioceptive data to estimate the relative position of the other vehicles.

## 7.9 Conclusion

This paper presented a real time cooperative localization method for cooperative and autonomous vehicles using global position system GPS, rangefinder and inter-vehicular communication. Each vehicle uses its GPS receiver data to estimate its own ego vector state, and it uses a rangefinder to measure the inter-vehicular distance, between itself and its neighbors, and their bearing angle. It uses V2V communication devices to transmit the relevant information to other vehicles. It also described the main issues with distributed data fusion architectures, such as unknown cross-covariance, which induces to over-covariance, due to fusion of correlated information. The preliminary simulation results showed the efficiency of the suggested data fusion algorithms. The strategy stands on two data fusion nodes: a local one to fuse the proprioceptive and exteroceptive non-correlated sensors' data according to an EKF, and a global one to fuse the broadcast information between vehicles. It revealed that the more vehicles there are in the cluster, the more accurate the vehicles' state estimate will be, as in the case of the blue and green vehicles. In future work, we will consider a more sophisticated non-linear model of the vehicle, and use dedicated filters, such as Unscented Kalman Filter (UKF), Particles Filter (PF), [...]. We will also focus on solutions to appearing and disappearing vehicles, and on some inter-vehicular communication, delay, issues.

Table 7-1 : Sensors' variance matrices

Sensors	R
GPS 1	$R^1_{GPS} = \text{diag} [ 3.5^2, 3.5^2, (\pi /180)^2, 0.1^2 ]$
GPS 2	$R^2_{GPS} = \text{diag} [ 4^2, 3^2, (\pi /180)^2, 0.2^2 ]$
GPS 3	$R^3_{GPS} = \text{diag} [ 4^2, 4^2, (\pi /180)^2, 0.2^2 ]$
Lidars	$R^1_{LIDAR} = \text{diag} [ 0.1^2, (\pi /180)^2 ]$

# **Chapitre 8**

## **From simultaneous localization and mapping to simultaneous collaborative localization and mapping for intelligent vehicles**

### **8.1 Avant-propos**

#### **8.1.1 Auteurs et affiliation**

**Farid Bounini :** Étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Denis Gingras :** Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Hervé Pollart :** Responsable du développement commercial de la voiture intelligente, OPAL-RT Technologies inc. Lyon, France.

**Dominique Gruyer :** Chargé de recherche, IFSTTAR, CoSys – LIVIC, Versailles, France.

#### **8.1.2 Date de soumission**

2 septembre 2017.

#### **8.1.3 Etat de l’acceptation**

Soumis.

### **8.1.4 Revue**

IEEE Transaction on Intelligent Vehicles - Special Issue IV'17.

### **8.1.5 Titre français**

De la localisation et la cartographie simultanées à la localisation et la cartographie collaboratives et simultanées pour les véhicules intelligents et connectés.

### **8.1.6 Contribution au document**

Ce chapitre est une extension du précédent chapitre, abordant la localisation collaborative d'un ensemble de véhicules intelligents et connectés ; équipés de capteurs proprioceptifs et extéroceptifs, peu couteux, et de systèmes de communication. La contribution principale de ce chapitre réside dans l'algorithme de fusion de données corrélées sur les architectures distribuées. La méthode permet de fusionner des informations corrélées et non alignées, ceci à des fins de localisation et de cartographie simultanées en mode coopératif dans un ensemble de véhicules intelligents et communicants. La méthode permet de réduire la complexité du calcul en diminuant l'ordre associé aux composantes liées au nombre de véhicules intelligents et connectés et à celles liées aux obstacles détectés : cet ordre passant du cubique au quadratique.

Ce chapitre fait partie de l'un des scénarios multivéhiculaires montrant la puissance et la capacité du simulateur à s'adapter aux besoins de simulation et de validation des algorithmes de traitement de données sur des plateformes distribuées.

### **8.1.7 Résumé français**

Ce chapitre présente une nouvelle stratégie de localisation coopérative en temps réel des véhicules intelligents et connectés. Elle combine la méthode de la localisation et de la cartographie simultanées avec la méthode de fusion de données « *Track-To-Track* ». La stratégie vise à améliorer la précision du positionnement et du suivi d'un ensemble de véhicules autonomes. Elle aspire aussi à surmonter les lacunes des capteurs proprioceptifs embarqués et à résoudre le problème de localisation dans un groupe de véhicules intelligents et connectés, équipés de systèmes de navigation bon marché et évoluant dans un environnement inconnu. Pour ce faire, deux nœuds indépendants de fusion de données – local et global – sont associés à chaque véhicule. Le premier nœud exploite plusieurs formes des dérivés du filtre de Kalman

pour la représentation des moments statistiques du vecteur d'état des véhicules, et son équivalent, le filtre de l'information, pour estimer respectivement les vecteurs d'état et d'information, et les matrices de covariance et d'information des véhicules. La deuxième étape de cette approche consiste à transmettre les résultats des nœuds de fusion locaux au reste du groupe. Ces résultats sont alors exploités pour résoudre le problème de fusion de données corrélées, cas de « *Track-To-Track* », à l'aide des méthodes de l'intersection des covariances et de fusion des matrices d'information, au niveau des nœuds globaux. Pour une meilleure localisation collaborative de ce groupe de véhicules, en matière de précision et de fiabilité, la stratégie exploite les mêmes sources de données que dans les travaux présentés dans le précédent chapitre, à savoir : un capteur proprioceptif, un système de positionnement global GPS, un capteur extéroceptif, un télémètre pour détecter les obstacles statiques et mobiles dans leur environnement et les moyens de communication de véhicule à véhicule. Les résultats de la simulation et du traitement des données expérimentales des scénarios multi-véhicules sur des architectures distribuées montrent et confirment l'efficacité des stratégies proposées.

## 8.2 Abstract

This paper describes a new method for real-time collaborative localization for intelligent vehicles based on a combination of an extension of simultaneous localization and mapping and track-to-track fusion methods. This strategy aims to overcome the shortcomings of the embedded proprioceptive sensors and solve the localization issues in a cluster of intelligent vehicles, which are equipped with low-cost navigation systems (global position systems, rangefinders, and vehicle to vehicle communication means) in an unknown environment. The strategy improves the vehicles' localization accuracy and tracks a cluster of vehicles for advanced driver assistance systems and vehicles self-driving. For real-time purposes, the computational complexity of standard solutions, which are based on Kalman filter derivatives, is transformed from cubic to quadratic with respect to the state vector of the cluster. Furthermore, the correlated data is another challenge that needs to be solved in cooperative localization scenarios. To this end, two independent, local and global, fusion nodes are associated to each vehicle. The local node exploits multiple forms of the extended Kalman filter and its dual extended information filter to estimate the vehicles' state, information vectors, covariance and information matrices in the moments and information forms, respectively. Then, the vehicles broadcast the result of their respective local fusion node to cluster. The information is used to solve the second problem, which is a track-to-track fusion on distributed architectures, with two methods: covariance intersection and information matrix fusion in the global fusion node. The simulation and the experimental results of the proposed methods on distributed architecture are compared to the optimal solution of a centralized data fusion that is based on an extended information filter in terms of root mean square errors and consistency.

## 8.3 Introduction

Localization accuracy is one of the most challenging aspects for Advanced Driving Assistance Systems (ADAS), which is one of the fastest-growing branches of electronic in automotive industry [133], and intelligent vehicles in general and for self-driving vehicles in particular. The interest behind accurate localization has mainly been driven by the increase in autonomy, cheap Micro-Electro-Mechanical Systems (MEMS), computer resources, embedded technologies, etc. However, localization systems still suffer from some lacks e.g., on the one hand, the local

positioning sensors experience the dead reckoning, measurement noises [123], on the other hand, global positioning systems (GPS) lose signal due to the configuration of the environment in which the GPS-receivers are evolving. This makes intelligent and self-driving vehicles' navigation a serious challenge.

For active safety applications, intelligent vehicles still need to locate themselves continuously and precisely. As a result, the major aspect of ADAS for intelligent vehicles is still an accurate and a reliable localization [124]. Cooperative localization is becoming an alternative solution for vehicles localization issues and a relevant application in autonomous vehicles. In collaborative scenarios, each vehicle is equipped with proprioceptive sensors and exteroceptive sensors and communication means forming to gather Vehicular Ad-hoc Network (VANET). The vehicles use their respective sensors to scan their environment, locate themselves in and track their neighbors. However, the number and the type of the objects to track increase the dimension of the model and call for powerful computational resources.

Subsequently, to reduce the computational complexity of the collaborative localization problem for real-time applications, the paper is structured as follows: Section II presents some related work on multi-sensors data fusion methods. Section III deals with vehicle localization statements according to a simple non-linear motion model in 2D plan. Section IV involves the vehicle simultaneous location and mapping (SLAM) statement according to its pre-defined motion model, the proprioceptive and exteroceptive measurements and an extended Kalman filter (EKF), which will be a basic framework and a starting point to explain its (EKF's) dual filters in information form that will be briefly discussed in sections V and VI. The EIF-SLAM is then adapted in section VII to include the vehicle's moving neighbors with their respective motion model to track them. Section VIII describes a solution for cooperative localization (CL) according to the adapted EIF-SLAM. The suggested real-time CL strategy is based on T2TF solution, suboptimal covariance intersection and on information matrix filters, on distributed architecture, that will be discussed in the sections IX and X. The approach solves the over-convergence issue, which usually occurs when the fused data sets, which come from different vehicles, are dependent on each other. The penultimate section presents some preliminary results for some specific, simulation and experimentation, multi-vehicular scenarios. Finally, the last section concludes the paper and highlights future works.

## 8.4 Related works

In vehicles' tracking scenarios, the relative state vectors estimate of the vehicles are obtained according to exteroceptive sensors measurements and their associated discrete state space models, which are often defined in 2D or 3D plan. To solve the vehicles' tracking problems, the multi-sensors measurements are merged to estimate some properties of the vehicles state vector—including for instance: position, heading, speed, acceleration, etc.—according to the hosted vehicle. Nevertheless, the exteroceptive sensors fusion problem can be divided into [133]:

- Data associations: where the measurements are associated to the vehicles, with their respective identification (LIDAR measurements are classified according to the detected vehicles and usually processed by filters. The latter could be an extended or unscented Kalman Filter (E&UKF, extended information filter (EIF), information filter (IF), etc.) in a local fusion node. Then, the results are sent to a global fusion node ;
- Vehicles state estimation to solve the target-tracking problem.

Generally, the state estimation and tracking problem is performed in decentralized architecture [133] and known as Track-To-Track Fusion (T2TF) (state vectors fusion) problem. Chang et al. in [135] focused on the analysis and performance evaluation of the T2TF problem of state vector fusion for two sensors according to the information matrix fusion, under the assumption that the sensors measurements are synchronized and independent. The study also described an analytical evaluation approach for comparing performance of several fusion methods with information matrix, in which two methods for the state estimation on hierarchical architecture—with full, partial and without feedback—are presented and compared. In case of full-rate communication, the fusion results of the hierarchical architecture with feedback are not optimal, and their incorporation in the local fusion nodes does not guarantee a better performance because of the approximation [133]. Thus, the study revealed that the quality of the information to be fused is more important than the quantity [133]. The study provided a performance trade-offs when designing a distributed fusion system. As the simulation results showed, the state vector estimates are very sensitive to the process noise, particularly those that are not directly observable. Thus, when the process noise is relatively small, the information matrix provides almost optimal performance with minimum communication requirements.

However, when it is large the performance of the optimal distributed algorithm (ODA) with full feedback degrades noticeably. It also revealed that increasing the communication rate, in case of important process noise, deteriorates more the performance of the ODA. Xiangdong et al. in [136] extend and investigate the stability and the performance of the information filter—primarily used to evaluate T2TF problem with two sensors in [135]—in distributed tracking system, in which ‘N’ sensors are tracking the same target with different feedback strategies (partial and full feedback). The paper showed that; 1- In case of full feedback, increasing communication rate or the number of sensors makes performance of the global fusion node worse, and 2- In the case of partial feedback and with half the data rate, the instability, which was observed with full feedback, is eliminated when the process noise is small. It performs as well as the centralized architecture. However, when the process noise gets larger, its performance degrades slightly compared to the optimal solution. The paper ended with the same conclusion as in [135]; “*The quality of the information to be fused is more important than the quantity.*”.

An evaluation of a decentralized information matrix fusion for advanced driver assistance systems in heavy-duty vehicles is presented in [133]. The objective was to compare multiple decentralized and distributed multi-sensors data fusion algorithms; 1- Naïve Algorithm (NA), which is a derivation of the weighted least square method, 2- Decentralized Minimum Information Algorithm (DMIA), and 3- Optimal Decentralized Algorithm (ODA). The last two algorithms are based on IMF without feedback (when the available communication bandwidth is limited, the information filter without feedback is very useful). Two embedded proprioceptive sensors, a camera and a radar, are used to detect and track several vehicles surrounding the hosted heavy-duty vehicle. The simulation results, conducted on overtaking scenarios, revealed that ODA without feedback is robust against sensor failures and is the most effective among the chosen algorithms, namely DMIA and NA. The main disadvantage of the ODA is memory space and a large bandwidth that it requires compared to the two others. Based on these high-level fusion methods, Rauch A. et al. also proposed in [137] a fusion architecture of cooperative perception systems. The main purpose of this architecture is to fuse the exchange data of the detected objects by the vehicles local exteroceptive sensors via wireless communication.

In this work, we adopt this concept to solve the data correlation problem that arises in distributed data fusion architectures for vehicles' collaborative localization. The main contributions of this work are:

- The reduction in the computational complexity of the localization methods using a standard extended information filter (EIF), which is an extension of the EIF-SLAM, presented in [139]. The strategy exploits also the sparseness of the collaborative localization model to compute the covariance matrix and its inverse without having to directly invert it ;
- The availability of the covariance and the information matrices at the prediction and the update steps allows the EIF to recover the exact value of its entire state vector, unlike the other methods that resort to matrix direct inversion or to optimization methods [139]. The latter needs several iterations to recover a segment of the state vector. This availability offers the possibility to take advantage of high-level data fusion methods on hierarchical and distributed architectures.

## 8.5 Vehicle localization statement with an extended Kalman filter

Real systems are generally characterized by their nonlinear behavior. For the sake of simplicity, and not to encumber the paper with mathematical equations, the vehicle is considered as a moving object in limited 2D plan, as shown in Figure 8-1. Then, a constant turn rate and acceleration motion model (CTRA) is associated to the vehicle. The exogenous disturbance and sensors measurements noise,  $w_m$  and  $w_s$ , are assumed to be additive to the process as follows.

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_{m,k}) \\ z_k = h(x_k, w_{s,k}) \end{cases} \quad (8.1)$$

where  $x^v = [x \ y \ \theta \ v \ a \ \omega]^T$  is the state vector of the vehicle:  $x$  and  $y$  are the vehicle coordinates in the 2D global plan.  $\theta$  and  $v$  are the heading angle and the speed of the vehicle,  $a$  and  $\omega$  are the vehicle longitudinal acceleration and heading speed, respectively.  $f^v(x^v_k)$  is a nonlinear transition function of the vehicle. The observation vector  $z^v$  incorporates the position, the heading angle and the longitudinal speed of the vehicle, which is provided, for instance, by the embedded GPS receiver.  $h^v(x^v_k)$  details the relationship between the measured state variables and the measurements vector  $z^v$ .  $w_{m,k} \sim N(0, Q)$  and  $w_{s,k} \sim N(0, R)$  are additive zero-mean Gaussian random vectors with covariance  $Q$  and  $R$ , respectively.  $T$  is the step time between two updates. To benefit

from the advanced state estimate strategies of linear systems, presented by discrete state space, the vehicle nonlinear model is approximated by its equivalent discrete linear time varying (LTV) model, according to the Taylor expansion with the first order at each time stamp  $k$ .

$$\begin{cases} \hat{x}_k^v = F_k^v \hat{x}_{k-1}^v + w_{m,k}^v \\ \hat{z}_k^v = H_k^v \hat{x}_k^v + w_{s,k}^v \end{cases} \quad (8.2)$$

where:  $F_k^v$  and  $H_k^v$  are the Jacobian of the nonlinear transition function  $f^v(x_k^v)$  and the nonlinear observation function  $h^v(x_k^v)$  according to the state vector, respectively. The observation function holds the first four components of the vehicle state vector, namely the vehicle's x and y coordinates, its longitudinal speed and its heading angle.

The estimation of the state vector is performed with an extended Kalman filter. The latter estimates, smooths and predicts the state vector  $x$ . Depending on the relationship between the two positive integers “ $l$ ” and “ $k$ ” in  $x_{k|l}$ , as given in [133], the KF or EKF could be:

- An estimator, in case if  $k$  is equal to  $l$  ;
- A smoother, in case if  $k$  is lower than  $l$  ;
- A predictor, in case if  $k$  is greater than  $l$ .

Due to the approximation of the state model and the observation matrices around its current state through Taylor series, the EKF is defined as a sub-optimal implementation of the recursive Bayesian estimator [138]. The EKF for localization is given in this case by the algorithm 1.

---

**Algorithm 1:** EKF for localization

---

```

EKF_Localization_PredictingStep( $\hat{x}_{k-1|k-1}^v, P_{k-1|k-1}^v, Q_k^v$ )
    1 :  $\hat{x}_{k|k-1}^v = f^v(\hat{x}_{k-1|k-1}^v)$ 
    2 :  $P_{k|k-1}^v = F_k^v P_{k-1|k-1}^v F_k^{vT} + Q_k^v$ 
    Return( $\hat{x}_{k|k-1}^v, P_{k|k-1}^v$ )
EKF_Localization_UpdatingStep( $\hat{x}_{k|k-1}^v, P_{k|k-1}^v, z_k^v, R_k^v$ )
    //GPS measurement
    1 :  $H_{k|k-1}^v = \nabla h^v(\hat{x}_{k|k-1}^v)$ 
    2 :  $K_k^v = P_{k|k-1}^v H_{k|k-1}^{vT} \left( H_{k|k-1}^v P_{k|k-1}^v H_{k|k-1}^{vT} + R_{GPS,k}^v \right)^{-1}$ 
    3 :  $\hat{x}_{k|k}^v = \hat{x}_{k|k-1}^v + K_k^v (z_k^v - h^v(\hat{x}_{k|k-1}^v))$ 
    4 :  $P_{k|k}^v = (I - K_k^v H_{k|k-1}^v) P_{k|k-1}^v$ 
    Return( $\hat{x}_{k|k}^v, P_{k|k}^v$ )

```

---

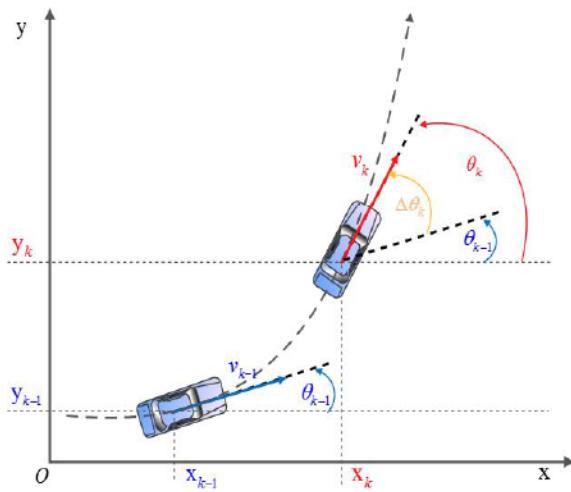


Figure 8-1: Vehicle dynamic in 2D

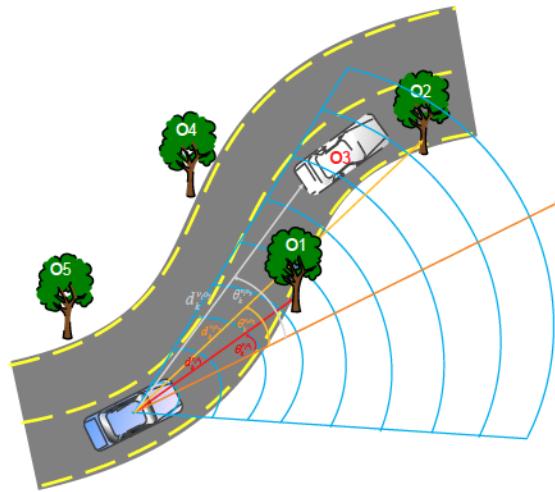


Figure 8-2: Inter-vehicular distance and bearing angle

where:  $\hat{x}_k$  is the state vector estimate of the vehicle.  $P^v_k$  is the error covariance estimate between the vehicle state vector and its estimate  $P^v_k = E[(x^v_k - \hat{x}^v_k)(x^v_k - \hat{x}^v_k)^T]$ .  $K^v_k$  is known as the KF observation gain matrix, which is defined optimally to minimize the error covariance of the state estimation during the updating step.

## 8.6 Simultaneous localization and mapping with EKF

Concurrent Mapping And Localization (CMAL) problem, which is also known as Simultaneous Localization And Mapping (SLAM), is the ability of robots and intelligent vehicles to build a map of their environment and locate simultaneously themselves in it for navigation purposes. To accomplish the two tasks, the hosted vehicle is basically equipped with two types of sensors; 1- proprioceptive sensors: Global Positioning System (GPS), Inertial Navigation System (INS), Inertial Measurement Unit (IMU), odometers, etc. These often provide the vehicle partial measurements on its own vector state, which will be exploited to estimate and predict the unmeasured states and their future values, as presented in the previous section, and 2- exteroceptive sensors: telemeter, LIDAR, RADAR, rangefinder, camera, etc. that sense their surroundings in order to be able to navigate in it. The exteroceptive sensors provide, for the majority, the relative distance between the hosted vehicle and the  $j^{\text{th}}$  obstacle  $d_k^j$ , and the relative bearing angle  $\theta_k^j$  at step time  $k$ . The discrete non-linear state model is then defined by (8.3).

$$\begin{cases} x_k = F_{k-1}x_{k-1} + w_{m,k} \\ z_k^o = h^o(x_k) + w_{s,k} \end{cases} \quad (8.3)$$

where:  $x_k = \begin{bmatrix} x_k^{vT} & \underbrace{x_k^{o_1T} \dots x_k^{o_NT}}_{x_k^{oT}} \end{bmatrix}^T$ ,  $x_k^v$ ,  $F_k = \begin{bmatrix} F_k^v & 0_{6 \times 2N} \\ 0_{2N \times 2} & I_{2N} \end{bmatrix}$  are the state vector of the extended model and the hosted vehicle and the state transition matrix, respectively.  $x_k^{o_j}$  contains the x and y coordinates of the  $j^{\text{th}}$  obstacle,  $h^o(x_k) = [h^{o_1}(x_k)^T \ h^{o_2}(x_k)^T \ \dots \ h^{o_N}(x_k)^T]^T$ ,  $R_{l,k}^v = E[(w_{Lidar,k}^v)(w_{Lidar,k}^v)^T] = \text{diag}\left(\begin{bmatrix} d_{w_{Lidar}^v}^2 & \theta_{w_{Lidar}^v}^2 \end{bmatrix}\right)$ ,  $R_k = \text{diag}\left(\begin{bmatrix} R_{GPS,k}^v & \underbrace{R_{l,k}^v \dots R_{l,k}^v}_{N\_times} \end{bmatrix}\right)$ ,  $w_{s,k}^v = [w_{GPS,k}^{vT} \ w_{Lidar,k}^{vT}]^T$ , and "N" is the number of detected obstacles by the hosted vehicle.

Thus, a new covariance matrix is defined according to the new state vector as follows.

$$P_k = \begin{bmatrix} P_k^v & P_k^{vo} \\ P_k^{ov} & P_k^o \end{bmatrix} \quad (8.4)$$

The measurement vector of the LIDAR is given as follows.

$$\begin{aligned} z_k^{o_j} &= h_k^{o_j}(x_k) = [d_k^{vo_j} \ \theta_k^{vo_j}]^T \\ &= \left[ \sqrt{(y_k^{o_j} - y_k^v)^2 + (x_k^{o_j} - x_k^v)^2} \right. \\ &\quad \left. + w_{s,k}^{vo_j} \right. \\ &\quad \left. \text{atan2}\left((y_k^{o_j} - y_k^v), (x_k^{o_j} - x_k^v)\right) - \theta_k^v \right] \end{aligned} \quad (8.5)$$

In the case of the EKF, the updating step requires an observation matrix, which is the result of the Jacobian of the observation nonlinear function according to the new extended state vector. The extended observation matrix is then given according to each obstacle, which is a combination of two observation matrices:

- The observation matrix according to the vehicle state ;

$$H_k^{vo_j} = \frac{\partial h_k^{o_j}(x_k)}{\partial x_k^v} \Big|_{x_k^v} = \frac{-1}{d_k^{o_j 2}} \begin{bmatrix} d_k^{o_j} \delta_k^x & d_k^{o_j} \delta_k^x & 0 & 0 & 0 & 0 \\ -\delta_k^y & \delta_k^x & d_k^{o_j 2} & 0 & 0 & 0 \end{bmatrix} \quad (8.6)$$

- The observation matrix according to the relative bearing angle and to the relative distance between the hosted vehicle and the  $j^{\text{th}}$  obstacle.

$$H_k^{o_j v_i} = \left. \frac{\partial h_k^{o_j}(x_k)}{\partial x_k^{o_j}} \right|_{x_k} = \frac{1}{d_k^{o_j 2}} \begin{bmatrix} d_k^{o_j} \delta_k^x & d_k^{o_j} \delta_k^y \\ -\delta_k^y & \delta_k^x \end{bmatrix} \quad (8.7)$$

where  $\delta_k^x = x_k^{o_j} - x_k^v$  and  $\delta_k^y = y_k^{o_j} - y_k^v$  are the difference between the x and y coordinates of the  $j^{\text{th}}$  obstacle and the hosted vehicle, respectively. Thus, the observation matrix according to the relative distance, the relative heading angle to the  $j^{\text{th}}$  obstacle and to the number of detected obstacles, denoted herein by  $N$ , is given as follows:

$$H_k^{o_j} = \begin{bmatrix} H_k^{v o_j} & 0_{2 \times 2(j-1)} & H_k^{o_j v_i} & 0_{2 \times 2(N-j)} \end{bmatrix} \quad (8.8)$$

$$H_k = \begin{bmatrix} H_k^{v z_{N_0}}^T & H_k^{o_1 T} & \cdots & H_k^{o_N T} \end{bmatrix}^T \quad (8.9)$$

where  $H_k^{v z_{N_0}} = [H_k^v \quad 0_{6 \times 2N}]$ .

The adapted EKF for simultaneous localization and mapping is given as follows:

---

**Algorithm 2: Adapted EKF SLAM**

---

*EKF\_SLAM\_PredictingStep*( $\hat{x}_{k|k-1}, P_{k|k-1}, Q_k$ )

- 1:  $\hat{x}_{k|k-1} = f(\hat{x}_{k|k-1})$
- 2:  $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$

*Return*( $\hat{x}_{k|k-1}, P_{k|k-1}$ )

*EKF\_SLAM\_UpdatingStep*( $\hat{x}_{k|k-1}, P_{k|k-1}, z_k, R_k$ )

// GPS and LIDAR measurement

- 1:  $H_{k|k-1} = \nabla h(\hat{x}_{k|k-1})$
- 2:  $K_k = P_{k|k-1} H_{k|k-1}^T (H_{k|k-1} P_{k|k-1} H_{k|k-1}^T + R_k)^{-1}$

// Update the state vector and the covariance matrix

- 3:  $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1}))$
- 4:  $P_{k|k} = (I - K_k H_{k|k-1}) P_{k|k-1}$

*Return*( $\hat{x}_{k|k}, P_{k|k}$ )

---

$N$  is the number of detected obstacles by the exteroceptive sensor (LIDAR). The computational complexity of the EKF is computed for the big order, and it is equal to  $O(q^{2.4}+n^2)$ , where  $q$  is the length of the observation vector, and  $n$  is the length of the state vector. It is evident that the computational complexity, according to the big order  $O(*)$ , of the EKF SLAM, in a crowded environment, is dominated by the observation vector.

## 8.7 Canonical representation and extended information filter for vehicle localization

The dual representation of the first two moments — mean and covariance —, in the canonical representation, of a multivariate Gaussian “ $X$ ” are the information vector and information matrix, respectively.

- The information matrix “ $\Omega$ ”, or the precision matrix, is defined as the inverse of the error covariance matrix “ $P$ ”;

$$\Omega = P^{-1} \quad (8.10)$$

- The information vector “ $\xi$ ” is defined as the product of the inverse of the error covariance matrix “ $P$ ” and the state mean “ $x$ ” vector.

$$\xi = P^{-1}x = \Omega x \quad (8.11)$$

The bilateral transformation from the canonical representation to the moment representation can be easily performed using the expression hereafter.

$$\begin{cases} \Omega = P^{-1} \\ \xi = P^{-1}x \end{cases} \Leftrightarrow \begin{cases} P = \Omega^{-1} \\ x = \Omega^{-1}\xi \end{cases} \quad (8.12)$$

From the bilateral transformation of the two representations, the information filter (IF) is defined as the dual of the Kalman filter [139] for (continuous and discrete) linear state models. By analogy, the extended information filter (EIF) is the dual EKF for nonlinear state models. The formulation of the EIF predicting and updating steps are obtained in accordance of the EKF localization algorithm depicted in previous sections, and the bilateral transformation of the representations (8.10), (8.11) and (8.12). For more details the reader is referred to [139]. The corresponding EIF for localization algorithms is given by the algorithm 3.

---

**Algorithm 3: EIF for localization**

---

```

EIF_Localization_PredictingStep $\left(\hat{\zeta}_{k-1|k-1}^v, \Omega_{k-1|k-1}^v\right)$ 
    1 :  $\hat{x}_{k-1|k-1}^v = \Omega_{k-1|k-1}^{v,-1} \hat{\zeta}_{k-1|k-1}^v$ 
    2 :  $\Omega_{k|k-1}^v = \left(F_k^v \Omega_{k-1|k-1}^{v,-1} F_k^{vT} + Q_k^v\right)^{-1}$ 
    3 :  $\hat{x}_{k|k-1}^v = f^v\left(\hat{x}_{k-1|k-1}^v\right)$ 
    4 :  $\hat{\zeta}_{k|k-1}^v = \Omega_{k|k-1}^v \hat{x}_{k|k-1}^v$ 
    Return $\left(\hat{x}_{k|k-1}^v, \hat{\zeta}_{k|k-1}^v, \Omega_{k|k-1}^v\right)$ 
EKF_SLAM_UpdatingStep $\left(\hat{x}_{k|k-1}^v, \hat{\zeta}_{k|k-1}^v, \Omega_{k|k-1}^v, R_k\right)$ 
    // GPS measurement
    1 :  $\Omega_{k|k}^v = \Omega_{k|k-1}^v + H_k^{vT} R_{GPS,k}^{v,-1} H_k^v$ 
    2 :  $\hat{\zeta}_{k|k}^v = \hat{\zeta}_{k|k-1}^v + H_k^{vT} R_{GPS,k}^{v,-1} \left(z_k^v - h^v\left(x_{k|k-1}^v\right) + H_k^v x_{k|k-1}^v\right)$ 
    Return $\left(\hat{\zeta}_{k|k}^v, \Omega_{k|k}^v\right)$ 

```

---

The comparison between the EKF and EIF filters, in the case of localization, shows that they are complementary in terms of computational complexity for the predicting and updating steps. Table 8-1 shows that the complexity order of the EKF predicting step is lower than the EIF's. However, the complexity order of the EKF updating step is higher than the EIF's. The total computational complexity of the two filters depends on the size of the state vector of the model and the size of the measurement vector. Thus, the choice of implementing EKF or EIF depends on the application.

Table 8-1: Computation complexity order of EKF and EIF

Algorithms	Computation complexity order O()		
	Predicting step	Updating step	Total
EKF	$O(n^2)$	$O(q^{2.373})$	$O(n^2 + q^{2.373})$
EIF	$O(n^{2.373})$	$O(q^2)$	$O(n^{2.373} + q^2)$

## 8.8 Simultaneous localization and mapping with extended information filter

In order to follow the same procedure as described by Sebastian Thrun et al. in [139], the model (8.2) is then rewritten as follows:

$$f^v\left(x_k^v\right) = x_k^v + g^v\left(x_k^v\right) \quad (8.13)$$

Then, the extended nonlinear transition function  $f(*)$  is rewritten as follows :

$$f(x_k) = x_k + F_x^T g^v(x_k^v) \quad (8.14)$$

where  $F_x$  is a mask matrix of dimension  $n \times (n+2N)$ , adapting the dimension of  $g^v(x_k^v)$  to  $f(x_k)$ .  
 $F_x = [I_{n \times n}, 0_{n \times 2N}]$ .

Now, let us adapt the information matrix expression from localization to SLAM.

$$\Omega_{k|k-1} = \left( \underbrace{F_k \Omega_{k-1|k-1}^{-1} F_k^T}_{\Phi_{k|k-1}^{-1}} + \underbrace{Q_k}_{F_x^T Q_k^v F_x} \right)^{-1} \quad (8.15)$$

The effective way to compute the inverse of the information matrix is to exploit its expression as a sum of products of matrices according to the following lemma.

**Lemma:** For invertible quadratic matrices A, B, U and V with appropriate dimensions.

$$(A + UBV)^{-1} = A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1} \quad (8.16)$$

and

$$(A^{-1} + UB^{-1}V)^{-1} = A - AU(B + VAU)^{-1}VA \quad (8.17)$$

The proof of the inversion lemma is given in the Appendix A, for invertible quadratic matrices A, B, U and V. For the general form of any matrices U and V with appropriate dimensions, it suffices, as shown in [139], to show that:

$$(A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1})(A + UBV) = I \quad (8.18)$$

By applying the matrix inversion lemma (8.17) on (8.15), the information matrix expression becomes:

$$\Omega_{k|k-1} = \Phi_{k|k-1} - \Phi_{k|k-1} F_x^T \underbrace{\left( Q_k^{v,-1} + F_x \Phi_{k|k-1} F_x^T \right)^{-1}}_{n \times n} F_x \Phi_{k|k-1} \quad (8.19)$$

where :  $\Phi_{k|k-1} = (F_k \Omega_{k-1|k-1}^{-1} F_k^T)^{-1} = (F_k^T)^{-1} \Omega_{k-1|k-1} F_k^{-1}$ ,

$$F_k^{-1} = \begin{bmatrix} F_k^v & 0_{n \times 2N} \\ 0_{2N \times n} & I_{2N} \end{bmatrix}^{-1} = \begin{bmatrix} F_k^{v,-1} & 0_{n \times 2N} \\ 0_{2N \times n} & I_{2N} \end{bmatrix} \text{ and } (F_k^T)^{-1} = \left( \begin{bmatrix} F_k^v & 0_{n \times 2N} \\ 0_{2N \times n} & I_{2N} \end{bmatrix}^T \right)^{-1} = \begin{bmatrix} (F_k^{vT})^{-1} & 0_{2N \times n} \\ 0_{n \times 2N} & I_{2N} \end{bmatrix}$$

Then we define the EIF-SLAM as given by the algorithm 4.

---

**Algorithm 4:** EIF-SLAM

---

*EIF\_SLAM\_PredictingStep* $\left(\hat{\xi}_{k-1|k-1}, \Omega_{k-1|k-1}\right)$

- 1 :  $\hat{x}_{k-1|k-1} = \Omega_{k-1|k-1}^{-1} \hat{\xi}_{k-1|k-1}$
- 2 :  $F_k^{-1} = \text{diag}\left[F_k^{v,-1}, I_{2N}\right], (F_k^T)^{-1} = \text{diag}\left[F_k^{v,T,-1}, I_{2N}\right]$
- 3 :  $\Phi_{k|k-1} = F_k^{T,-1} \Omega_{k-1|k-1} F_k^{-1}$
- 4 :  $\Omega_{k|k-1} = \Phi_{k|k-1} - \Phi_{k|k-1} F_x^T \left( Q_k^{v,-1} + F_x \Phi_{k|k-1} F_x^T \right)^{-1} F_x \Phi_{k|k-1}$
- 5 :  $\hat{x}_{k|k-1} = f^v\left(\hat{x}_{k-1|k-1}\right)$
- 6 :  $\hat{\xi}_{k|k-1} = \Omega_{k|k-1} \hat{x}_{k|k-1}$

*Return* $\left(\hat{x}_{k|k-1}^v, \hat{\xi}_{k|k-1}^v, \Omega_{k|k-1}^v\right)$

*EKF\_SLAM\_UpdatingStep* $\left(\hat{x}_{k|k-1}, \Omega_{k|k-1}, z_k, R_k\right)$

// GPS and LIDAR measurement

- 1 :  $H_k = \nabla h\left(\hat{x}_{k|k-1}\right)$
- 2 :  $\Omega_{k|k} = \Omega_{k|k-1} + H_k^T R_k^{-1} H_k$
- 3 :  $\hat{\xi}_{k|k} = \hat{\xi}_{k|k-1} + H_k^T R_k^{-1} \left( z_k - h\left(x_{k|k-1}\right) + H_k x_{k|k-1} \right)$

*Return* $\left(\hat{\xi}_{k|k}, \Omega_{k|k}\right)$

---

The EIF predicting step is faster, but its computation complexity order is still the same. This is due to the requirement of the vector state  $x$  during the prediction step. Some proposed to approximate the  $x$  vector by transforming the matrix inversion problem into an optimization problem [139] that provides a partial approximation of the state vector. This solution is valid for sparse information matrix, and its convergence requires several iterations at each timestamp. In order to construct the exact and the full vector state estimate, for T2TF problems, the bilateral transformation needs the covariance matrix as depicted in equation (8.12). Its computation in conical representation for the updating step, according to the exteroceptive sensor measurements and to the number of detected obstacles, is based on the information matrix expression. The effective way to compute the inverse of the information matrix is to exploit its expression as a sum of products of matrices. According to the matrix inversion lemma, the EIF general covariance matrices, as expressed in the localization formulation, can be rewritten as follows.

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H_k^T \underbrace{\left(R_k + H_k P_{k|k-1} H_k^T\right)^{-1}}_{q \times q} H_k P_{k|k-1} \quad (8.20)$$

This new formulation of the covariance matrix changes the computation order of the standard EIF prediction step from  $O(n^{2.4})$  to  $O(q^{2.4})$ , algorithm 5, e.g. in case of localization issue, the computational complexity of the predicting step goes from  $O((n=6)^{2.4})$  to  $O((q=4)^{2.4})$ . In case of spars matrices, it is easy to transform the computational complexity form cubic (matrix invers) to quadratic (matrix products) with respect to the size of the state vector.

It is worth noting that the proposed method for the EIF-SLAM does not need to directly inverse the information matrix during the predicting and updating steps. It provides, at each timestamp, the exact values of the first two estimates of the moments of the multivariate vector  $x$  in the moments and in the canonical representations, respectively. The data fusion algorithms for distributed architectures sometimes uses both representations. Another interest in having the values of the first two movements in the two representations is their use in Track-To-Track fusion processes. The global node can directly employ the information in both representations.

---

**Algorithm 5:** The new complete EIF-SLAM

---


$$\begin{aligned}
 & EIF\_SLAM\_PredictingStep\left(\hat{x}_{k-1|k-1}, P_{k-1|k-1}, \hat{\xi}_{k-1|k-1}, \Omega_{k-1|k-1}\right) \\
 1: & F_k^{-1} = \text{diag}\left[F_k^{v,-1}, I_{2N}\right], (F_k^T)^{-1} = \text{diag}\left[F_k^{v,T,-1}, I_{2N}\right] \\
 2: & P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \\
 3: & \hat{x}_{k|k-1} = f^v\left(\hat{x}_{k-1|k-1}\right) \\
 4: & \Phi_{k|k-1} = (F_k^T)^{-1} \Omega_{k-1|k-1} F_k^{-1} \\
 5: & \Omega_{k|k-1} = \Phi_{k|k-1} - \Phi_{k|k-1} F_x^T \left(Q_k^{v,-1} + F_x \Phi_{k|k-1} F_x^T\right)^{-1} F_x \Phi_{k|k-1} \\
 6: & \hat{\xi}_{k|k-1} = \Omega_{k|k-1} \hat{x}_{k|k-1} \\
 & \text{Return}\left(\hat{x}_{k|k-1}^v, P_{k|k-1}, \hat{\xi}_{k|k-1}^v, \Omega_{k|k-1}^v\right) \\
 & EIF\_SLAM\_UpdatingStep\left(\hat{x}_{k|k-1}, P_{k|k-1}, \hat{\xi}_{k|k-1}, \Omega_{k|k-1}, z_k, R_k\right) \\
 1: & H_k = \nabla h\left(\hat{x}_{k|k-1}\right) \\
 2: & \Omega_{k|k} = \Omega_{k|k-1} + H_k^T R_k^{-1} H_k \\
 3: & \hat{\xi}_{k|k} = \hat{\xi}_{k|k-1} + H_k^T R_k^{-1} \left(z_k - h\left(\hat{x}_{k|k-1}\right) + H_k \hat{x}_{k|k-1}\right) \\
 4: & P_{k|k} = P_{k|k-1} - P_{k|k-1} H_k^T \left(R_k + H_k P_{k|k-1} H_k^T\right)^{-1} H_k P_{k|k-1} \\
 5: & \hat{x}_{k|k} = P_{k|k} \hat{\xi}_{k|k} \\
 & \text{Return}\left(\hat{x}_{k|k}, P_{k|k}, \hat{\xi}_{k|k}, \Omega_{k|k}\right)
 \end{aligned}$$


---

## 8.9 Vehicle neighbors state estimate statement

Cooperative localization could be a good solution to overcome the sensors' gaps. It would allow automotive makers to equip their intelligent vehicles with several inexpensive embedded proprioceptive and exteroceptive sensors. The idea is to equip each vehicle with a GPS, a telemeter (Lidar, Radar, camera, etc.), and with the means of inter-vehicular communication (a transmitter and a receiver). The proprioceptive sensors (odometers, INS, GPS, etc.) provide the hosted vehicle with its own ego vector state measurement, which will be exploited to predict and estimate the vehicle future vector state. The exteroceptive sensors (telemeters) allow the vehicle to sense its surrounding environment to plan its trajectory and avoid static and moving obstacles. Thus, the rangefinder provides the relative distance separating the two vehicles (inter-vehicular distance), and the bearing angle, which is the case of SLAM. The mathematical model of hosted vehicles and their neighbors that will be involved during the next T2TF problem is defined as in the equation (8.2) according to Figure 8-2.

To get a general model for cooperative localization, let us consider a cluster of “ $M$ ” vehicles. The vehicles’ dynamics models are concatenated to form a new extended model.

$$\begin{cases} \boldsymbol{x}_k^{v_i^{\text{ex}}} = \left[ \underbrace{\boldsymbol{x}_k^{v_1 T} \cdots \boldsymbol{x}_k^{v_M T}}_{\boldsymbol{x}_k^{v_{\text{hs},T}}} \quad \underbrace{\boldsymbol{x}_k^{o_1 T} \cdots \boldsymbol{x}_k^{o_N T}}_{\boldsymbol{x}_k^{o_{\text{bs},T}}} \right]^T = F_k^{v_i^{\text{ex}}} \boldsymbol{x}_{k-1}^{v_i^{\text{ex}}} + \boldsymbol{w}_{m,k}^{v_i^{\text{ex}}} \\ \boldsymbol{z}_k^{v_i^{\text{ex}}} = H_k^{v_i^{\text{ex}}} \boldsymbol{x}_k^{v_i^{\text{ex}}} + \boldsymbol{w}_{s,k}^{v_i^{\text{ex}}} \end{cases} \quad (8.21)$$

$$\text{where: } F_k^{v_i^{\text{ex}}} = \text{diag} \left[ \underbrace{\boldsymbol{F}_k^{v_1} \quad \boldsymbol{F}_k^{v_2} \quad \cdots \quad \boldsymbol{F}_k^{v_M}}_{M \text{ times}} \quad \boldsymbol{I}_{2N} \right],$$

$$\boldsymbol{w}_{m,k}^{v_i^{\text{ex}}} = \left[ \underbrace{\boldsymbol{w}_{m,k}^{v_1 T} \cdots \boldsymbol{w}_{m,k}^{v_M T}}_{\boldsymbol{w}_{m,k}^{v_{\text{hs},T}}} \quad \underbrace{\boldsymbol{w}_{m,k}^{o_1 T} \cdots \boldsymbol{w}_{m,k}^{o_N T}}_{\boldsymbol{w}_{m,k}^{o_{\text{bs},T}}} \right]^T \text{ and}$$

$$\boldsymbol{Q}_{m,k}^{v_i^{\text{ex}}} = E \left[ \boldsymbol{w}_{m,k}^{v_i^{\text{ex}}} \boldsymbol{w}_{m,k}^{v_i^{\text{ex},T}} \right] = \text{diag} \left( \left[ \underbrace{\boldsymbol{Q}_k^v \quad \cdots \quad \boldsymbol{Q}_k^v}_{M \text{ times}} \quad \boldsymbol{0}_{2N} \right] \right)$$

The changes take place at the EIF-SLAM prediction step when computing the prediction of the information matrix  $\Omega_{k|k-1}$ ,  $F_x$  becomes  $F_x^i$  according to the  $i^{\text{th}}$  vehicle.

$$\Omega_{k|k-1} = \left( \Phi_{k|k-1}^{-1} + \sum_{i=1}^M F_x^i Q_k^{v,-1} F_x^{i,T} \right)^{-1} \quad (8.22)$$

Then, the inversion matrix lemma is applied according to the number of vehicles in the cluster and  $F_x^i$  is a selection mask of the state variables that are related to the  $i^{\text{th}}$  vehicle with appropriate

dimension. The exploitation of the sparseness of the SLAM and collaborative localization problems reduces the computational complexity of the algorithms from cubic to quadratic. A good implementation of the algorithms could reduce their respective computational complexity furthermore. On the other hand, the computation of the state and information vectors and of the covariance and the information matrices with the proposed method requires larger memory, twice larger than an EKF/EIF's, which is quadratic to the length of its state vector.

## 8.10 Cooperative localization

The inter-vehicular communication system allows the equipped vehicles to exchange their own and their environment's state estimate with their neighbors. The main purpose of such applications is to improve cooperative localization accuracy. The communication deadline, in this work, is assumed to be equal to zero second. The V2V communication allows the cluster of vehicles to form a Vehicular Ad Hoc Network (VANET) and measure the inter-vehicular distance to locate their neighbors. This measurement is taken into account during the updating step, to exchange their own and environment state estimate (mean and information vectors) and their respective covariance and information matrices. Thereby, the cluster of vehicles forms a data fusion architecture. There are mainly three data fusion architectures [130] and [133]:

### 8.10.1 Centralized data fusion architecture

The raw measurements of each sensor “S” are sent to the same global fusion node (GFN) that merges them. Thanks to the availability of non-correlated data (the raw sensors measurements), this architecture delivers optimal results. The following filters could be used according to the severity of the non-linearity in the models: Kalman filter for linear models, EKF for nonlinear models and unscented KF in the case of severer non-linearity.

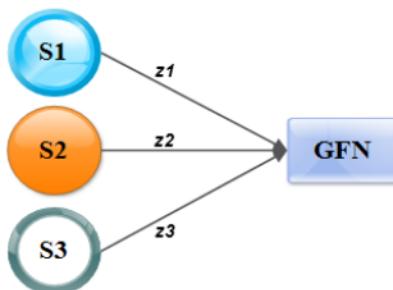


Figure 8-3: Centralized data fusion architecture

### 8.10.2 Decentralized data fusion architecture

Each sensor raw measurement is first sent to their respective local fusion node (LFN) that processes the raw data; filtering and estimating. Then, the result of the LFNs are sent to the global fusion node for a global estimation.

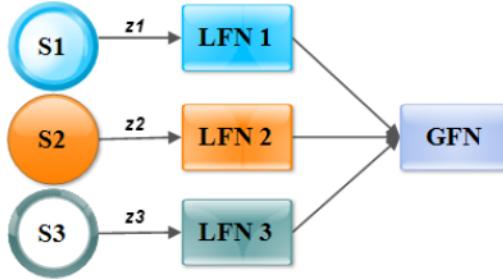


Figure 8-4: Decentralized data fusion architecture

### 8.10.3 Distributed data fusion architecture

There is no hierarchy between the fusion nodes. However, there are several fusion nodes communicating with each other and performing their own fusion of raw measurements and transmitted estimates from other fusion nodes.

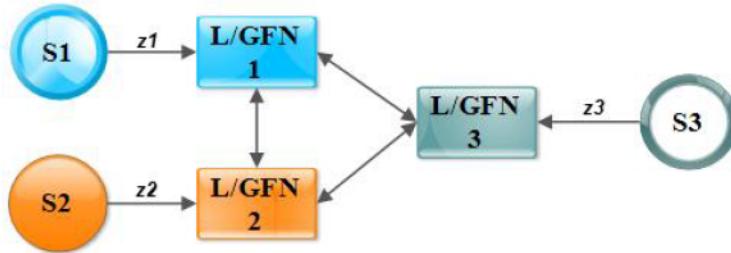


Figure 8-5: Distributed data fusion architecture

In order to perform collaborative localization, the distributed data fusion architecture is selected to solve the cooperative localization and track to track fusion problems. To do so, each vehicle is considered as two separate fusion nodes: a local and a global fusion nodes. The local fusion node fuses the vehicle sensors raw measurement according to the suggested EIF-SLAM algorithm, as presented previously, and then the result is broadcasted to its own global fusion node and to its neighbors'.

Several sources of common errors, in decentralized and distributed architecture, need to be removed each time the fusion node receives a new update [130]. Otherwise, they will cause

over-covariance, which is another novel challenge to overcome in T2TF problems that will be deployed in cooperative vehicles scenarios. It describes a situation where the analysis covariance errors are systematically underestimated after each observation [85]. After the second observation, the fusion nodes use almost the same correlated data. As a consequence, the usual data fusion algorithms (EKF, UKF, EIF, etc.) are no longer efficient.

To get around over-covariance issues, local and global fusion nodes (LFN and GFN) are created, as noted previously. The LFN fuses the non-correlated data; which comes from the sensors, according to the EIF for localization and EIF-SLAM for simultaneous localization and mapping. Then, each vehicle broadcasts the results of its LFN to its neighbors. The GFN merges the exchanged data separately and independently. Then several data fusion architectures could be adapted with partial, full, or without feedback. Thus, the vehicles use their own GFN to accomplish their own purposes: cooperative SLAM, trajectory planning, control, etc. To do so, two solutions for real-time applications of adapted T2TF are briefly discussed in the next two sections.

## 8.11 Fast suboptimal covariance intersection

Considering “ $x_i$ ” and “ $x_j$ ” are multivariate Gaussian estimates of the “ $x$ ”, state vector of the vehicle, provided by two different vehicles. The vector state estimates are usually dependent, their cross-covariance differs from zero  $P_{xij} \neq 0$  and  $P_{jxi} \neq 0$  [128]. Consequently, the only information, that could be exploited from the two signals, are their mean  $(\bar{x}_i, \bar{x}_j)$  and their error covariance matrices  $(P_{xi*}, P_{xj*})$ , respectively. The unidentified cross-covariance causes the conventional KF and its derivatives (EKF, UKF, IF, EIF) to diverge [128]. The Covariance Intersection (CI) is specifically developed to overcome such problems. As noted in [140], CI provides a general framework for information fusion with incomplete knowledge about the signal sources, since it yields to consistent estimates for any degree of cross correlation. The only constraint imposed on the assumed estimate is the consistency [85]. If CI filter is not over-confident, it is said to be consistent. However, it is made for the worst cases, where the source estimates are completely correlated. As a result, the CI neglects possible independent information during the fusion.

$$P_{x_i} - P_{e_{x_i}} \geq 0, P_{x_j} - P_{e_{x_j}} \geq 0 \text{ and } P_x - P_{e_x} \geq 0 \quad (8.23)$$

where  $e_{x_i} = x_i - \bar{x}_i$ ,  $e_{x_j} = x_j - \bar{x}_j$ ,  $e_x = x - \bar{x}$ , and  $x$  is the fusion result of the two pieces of the original signal with a mean  $\bar{x}$  and covariance  $P_x$ .

The principle of covariance intersection algorithms consists of finding the values of  $\omega_i$  that minimizes the trace or the determinant of the final covariance matrix  $P_k$ . The general form of the CI, for cooperative localization according to the bilateral representation, is given as follows:

$$\begin{cases} \Omega_{k/k}^{v_\alpha} = \sum_{i=1}^M \omega_i \Omega_{k/k}^{v_{\alpha,i}} \\ \xi_{k/k}^{v_\alpha} = \sum_{i=1}^M \omega_i \xi_{k/k}^{v_{\alpha,i}} \\ \sum_{i=1}^M \omega_i = 1 \end{cases} \Leftrightarrow \begin{cases} P_{k/k}^{v_\alpha} = \Omega_{k/k}^{v_\alpha - 1} \\ x_{k/k}^{v_\alpha} = P_{k/k}^{v_\alpha} \xi_{k/k}^{v_\alpha} \\ \sum_{i=1}^M \omega_i = 1 \end{cases} \quad (8.24)$$

The covariance intersection algorithm requires an optimization of a nonlinear cost function, minimization of the trace/determinant of the sum of weighted covariance matrices. This is a major gap with respect to computational complexity. To overcome CI drawback, Wolfgang Niehsen et al. [140] developed a suboptimal fast covariance intersection algorithm. First, they reformulated the covariance optimization problem, in the case of two estimates ( $P_i$  and  $P_j$ ) by assuming an addition constraint on the traces of the covariance matrices that provides an estimation uncertainty of any multivariate estimate such as  $\text{tr}(P_i) = \text{tr}(P_j)$ . Thus, given  $P_i$  and  $P_j$  are symmetric and semidefinite positive matrices that implies:  $\lambda_n = \text{tr}(P_n) > 0$  and  $\lambda_i + \lambda_j > 0$ .

Thus:

$$\begin{cases} \omega_i = \lambda_j / (\lambda_i + \lambda_j) \\ \omega_j = \lambda_i / (\lambda_i + \lambda_j) \end{cases} \quad (8.25)$$

The general expression of any  $\omega_i$  for this suboptimal covariance intersection, in case of “ $M$ ” estimates, is given as follows:

$$\omega_i = \lambda_i^{-1} \left( \sum_{j=1}^M \lambda_j^{-1} \right)^{-1} \quad (8.26)$$

This new expression of the weight factor  $\omega_i$  gets around the nonlinear optimization that is very expensive in terms of computational resources. Once the traces of the different estimates of the error covariance matrices are computed, then the final values of the weight factors are easily determined according the equation (8.26).

## 8.12 Information matrix fusion

The principal information matrix fusion (IMF) is decorrelating the exchanged estimates by identifying and removing the common information that are caused by prior information [136]. The IMF provides almost optimal performance with a minimum communication requirement [135]. It is mainly used in distributed architectures with multi-sensors data fusion nodes to solve the T2TF problem. The advantages of the IMF are its simplicity, robustness, modularity, and it does not require the cross-covariances among the local tracks [141]. The general form of the IMF at the global fusion node is given as follows:

- In moments representation ;

$$\begin{cases} P_{k/k}^{v_\alpha, -1} = P_{k/k-1}^{v_\alpha, -1} + \sum_{i=1}^M (P_{k/k}^{v_\alpha, i, -1} - P_{k/k-1}^{v_\alpha, i, -1}) \\ P_{k/k}^{v_\alpha, -1} x_{k/k}^{v_\alpha} = P_{k/k-1}^{v_\alpha, -1} x_{k/k-1}^{v_\alpha} + \sum_{i=1}^M (P_{k/k}^{v_\alpha, i, -1} x_{k/k}^{v_\alpha, i} - P_{k/k-1}^{v_\alpha, i, -1} x_{k/k-1}^{v_\alpha, i}) \end{cases} \quad (8.27)$$

- In canonical/ information representation ;

$$\begin{cases} \Omega_{k/k}^{v_\alpha} = \Omega_{k/k-1}^{v_\alpha} + \sum_{i=1}^M (\Omega_{k/k}^{v_\alpha, i} - \Omega_{k/k-1}^{v_\alpha, i}) \\ \xi_{k/k}^{v_\alpha} = \xi_{k/k-1}^{v_\alpha} + \sum_{i=1}^M (\xi_{k/k}^{v_\alpha, i} - \xi_{k/k-1}^{v_\alpha, i}) \end{cases} \quad (8.28)$$

- Bilateral transformation.

$$\begin{cases} \Omega_{k/k}^{v_\alpha} = \Omega_{k/k-1}^{v_\alpha} + \sum_{i=1}^M (\Omega_{k/k}^{v_\alpha, i} - \Omega_{k/k-1}^{v_\alpha, i}) \\ \xi_{k/k}^{v_\alpha} = \xi_{k/k-1}^{v_\alpha} + \sum_{i=1}^M (\xi_{k/k}^{v_\alpha, i} - \xi_{k/k-1}^{v_\alpha, i}) \\ P_{k/k}^{v_\alpha} = \Omega_{k/k}^{v_\alpha, -1} \\ x_{k/k}^{v_\alpha} = P_{k/k}^{v_\alpha} \xi_{k/k}^{v_\alpha} \end{cases} \quad (8.29)$$

Different strategies could be adapted to this hierarchical structure with partial and full feedback. When the available bandwidth is limited, the information matrix filter without the feedback is very useful to overcome this limitation. However, the global fusion node may send the latest fused information to the local fusion nodes when the communication bandwidth is large enough. The transmitted information of the GFN replaces the LFN's. In the case of full-rate communication, the fusion results of the hierarchical architecture with feedback are not optimal, and their incorporation in the local fusion nodes does not guarantee a better performance because of approximations [133]. As highlighted in [136]: increasing communication rates or

the number of LFNs (sensors) with complete feedback makes performance of T2TF worse. However, partial feedback eliminates the instability. When the process noise is small, the algorithm performs as well as the centralized architecture. When the process noise get larger, its performance degrades slightly compared to the optimal solution.

## 8.13 Simulation results

The two multi-vehicle scenarios are implemented on our real-time simulator of intelligent and connected vehicles; a description of the simulator is given in [112]. The first scenario consists of simulating and evaluating the performance of EKF localization for a single vehicle. The latter is equipped with three GPS receivers. Their respective covariance matrices  $R^1_{GPS}$ ,  $R^2_{GPS}$  and  $R^3_{GPS}$  involve the variances of the global position of the vehicle on x and y direction, its heading angle and its speed as follows:  $R^i_{GPS} = \text{diag} [\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_v^2]$ . To estimate the vehicle position, an EKF for localization algorithms is implemented in a LFN for each GPS receiver. Then, the fusion results are sent to the on-board GFN. This description corresponds to the distributed architecture fusion without feedback which, and it is similar to the decentralized architecture. As mentioned previously, the performance of central architectures is optimal. Thus, a central EKF (CEKF) is used to merge the measurements of the three GPS receivers in a central data fusion architecture, Figure 8-3. The CEKF result is then used as a reference to compare the simulation results of the different algorithms and their consistency.

Figure 8-6 shows the simulation results for vehicle trajectory estimations based on: the three LFNs using EKFs, and the GFNs using CI, suboptimal CI (SOCl), CEIF and the ODA, respectively. The simulation results show that the CI and SOCl both perform well. However, SOCl doesn't resort to the optimization methods as CI does. Thus, SOCl is faster than CI. The ODA performs as well as the central optimal filter CEKF. Their respective estimate of the vehicles trajectory (dashed cyan and solid black lines) overlap on each other.

Figure 8-7 shows the RMS errors of each node according to x and y axis and to the vehicle position. The RMSE of CEKF and ODA, in the x and y direction and in the total position. Their RMSEs, in the x direction, decrease faster than the other fusion nodes and have the minimum value. However, in the y direction, the RMEs of the GFNs with CI and SOCl decrease faster and perform slightly better. Their respective RMEs in both directions oscillate around the CEKF's and ODA's.

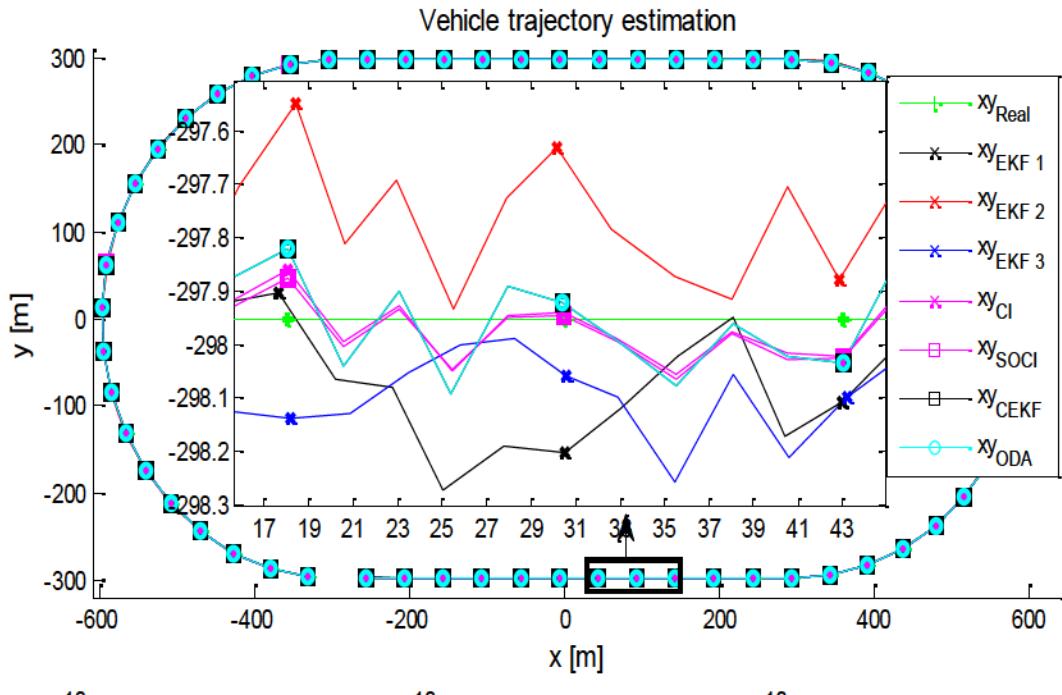


Figure 8-6: Vehicle trajectory estimation using three GPS receivers' measurements, real position of the vehicle (green line), the other lines are its estimates according to: 1- LFNs with EKFs of each GPS (dashed black, red and blue lines), and 2- GFN of the LFNs with CI, SOCI (dashed magenta and doted lines), and with ODA and CEKF (solid cyan and black lines)

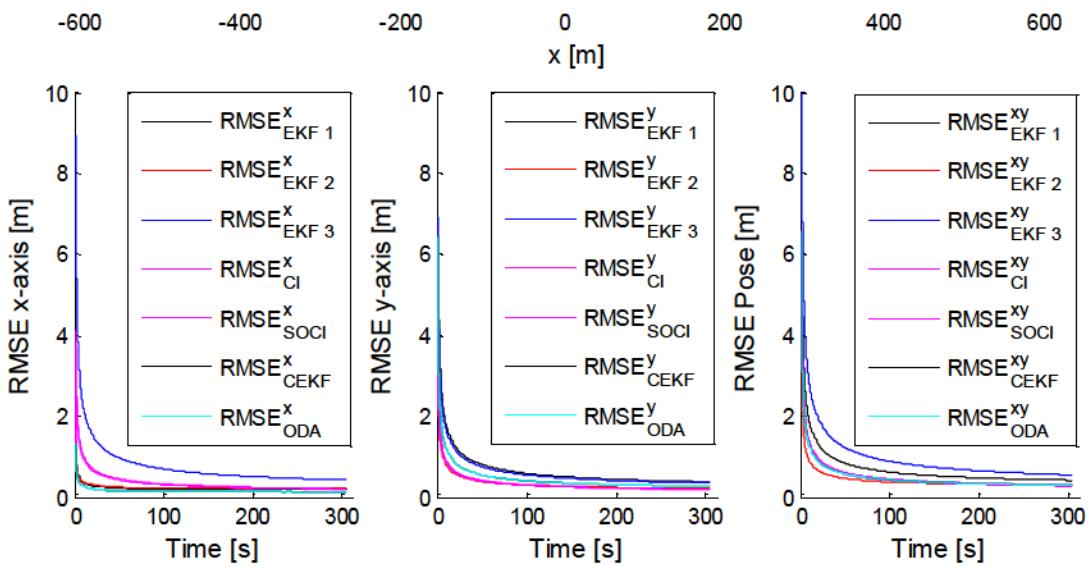


Figure 8-7: RMSE of LFNs with EKFs of each GPS (black, red and blue dashed lines), and of the GFN of the LFNs with CI, SOCI (dashed and doted magenta lines), and with CEKF and ODA (solid black and dashed cyan lines)

Figure 8-8 shows the evolution of the elements of the covariance matrix concerning the vehicle position, in x and y directions ( $P(x)=P(1,1)$ ,  $P(y)=P(2,2)$ ,  $P(x,y)= P(1,2)$ ). The elements of covariance matrices of CEKF and ODA are identical and the smallest. These results confirm the optimal criteria and the consistency of ODA for small process noise. It also shows the consistency of CI, SOCI, which are both designed for the worst case, in which the data are completely correlated. The CI and SOCI are both performing quite well in term of RMSE, compared to the LFNs, but they are limited in the covariance error minimization. Moreover, the two algorithms present some limitations. They both require synchronized information to perform the data fusion and estimate the vectors and matrix information. They are also memoryless, which makes them inefficient for predicting and tracking objects in the absence of new updates from the LFNs. Another version could be developed to include the vehicles motion model to predict the evolution of the moving objects and to synchronize the LFNs data. Nevertheless, the ODA has already the ability to predict and synchronize the broadcast data. Though, it loses its consistency in the case of significant process noise and requires twice as much memory as the CI. The computational complexity of the GFNs, involving CI, SOCI and ODA, is cubic, in the case where the LFNs use EKF for the localization, and quadratic, in the case of the use of EIF. However, the LFNs with EIF need to broadcast more information, twice as much as LFNs with EKF.

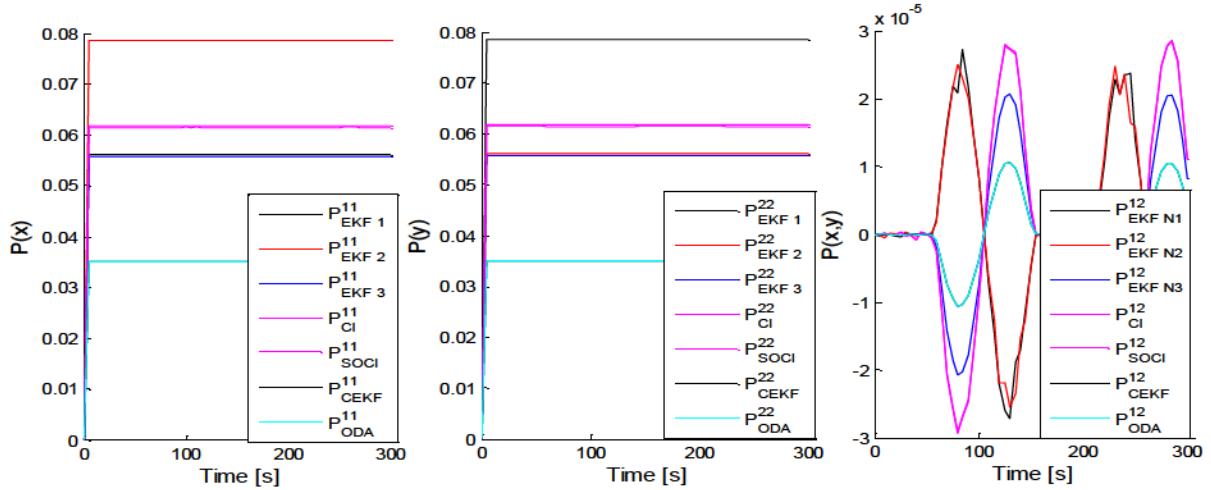


Figure 8-8: Position covariance elements of the local and the global fusion nodes evolution of the GPS receivers data fusion of the LFNs with EKFs of each GPS (dashed black, red and blue lines), and GFNs of the LFNs with CI, SOCI (magenta dashed and dotted lines), and with ODA and CEKF (solid cyan and black lines)

The second scenario involves a cluster of three vehicles (red, blue and green). The GPS receivers, which are used in the previous scenario (Table 8-2), are distributed on the cluster. The vehicles are equipped with identical rangefinder to measure the relative distance and bearing angle between the hosted vehicle and its neighbors in all directions, and with V2V communication means to broadcast their respective LFN fusion results. The communicating vehicles form together a VANET. The purpose of this network is to ensure that the vehicles are able to measure the inter-vehicular distance and to broadcast their respective estimate to their neighbors in order to perform the track-to-track fusion method. However, the communication channel is assumed to be ideal, thus the communication delay is assumed to be equal to zero. In this scenario, the proposed EIF for cooperative localization is deployed on the local fusion nodes to have access to the covariance and information matrices and the state and information vectors, which are sent to the global fusion nodes. A central EIF (CEIF) is also used to merge the raw measurement of the different sensors.

To reduce the complexity of the simulation scenario, the sensors measurements are assumed to be temporally and spatially aligned at the frequency of twenty Hertz. The same initial conditions (real values) are provided to all fusion nodes.

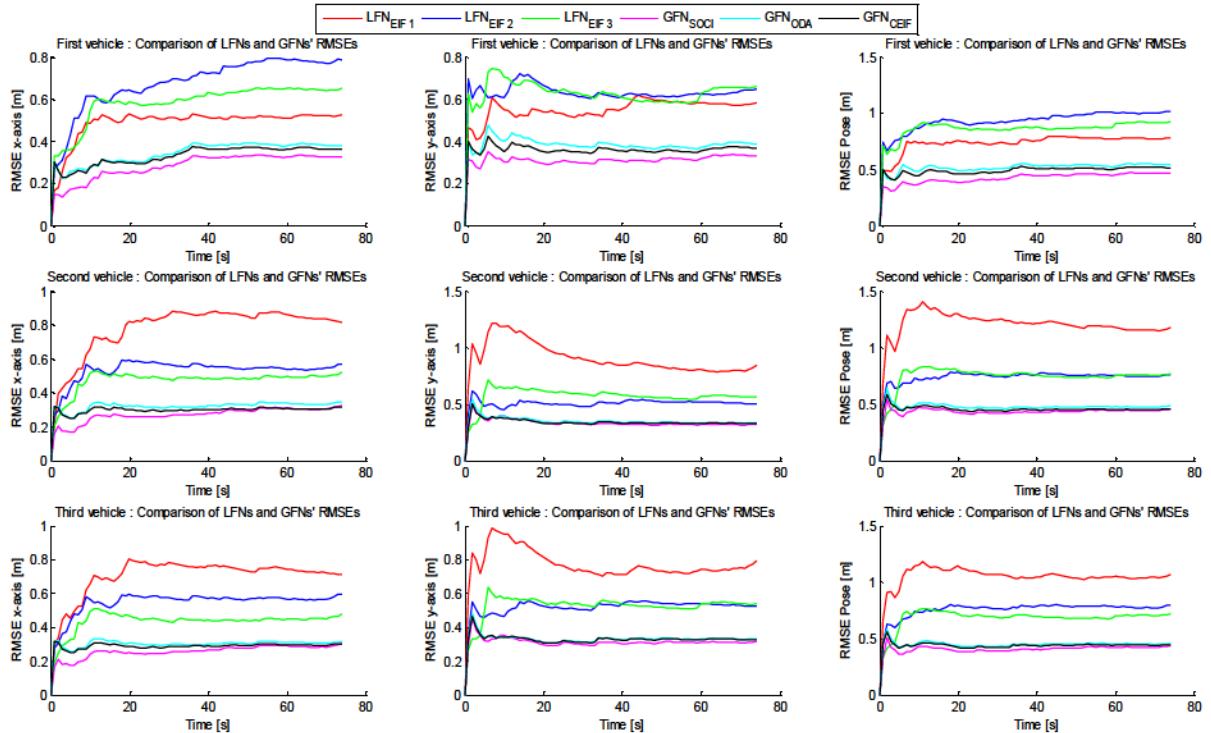


Figure 8-9: Vehicles RMSE position estimate in cooperative localization scenario

Figure 8-9 presents the evolution of the root mean square error of the vehicles positions compared to their respective real positions. Each line represents the RMSE evolution of the local and global fusion nodes tracking the same vehicle. The first two columns represent the RMSE position in x and y directions, respectively. The third column shows the evolution of the total RMSE position of the three vehicles. For instance, the first line shows the evolution of the different RMSEs on the position of the first vehicle that are estimated by its own LFN and GFN and by the LFNs of its neighbors.

The same figure shows also that the accuracy of the proprioceptive and exteroceptive sensors have an important effect on the fusion results. For instance, the first line shows that the RMSE of the LFN of the first vehicle (red) in x direction is smaller than the RMSE of the LFNs of its neighbors, by using only their own sensors measurement. This is due to the accuracy of the GPS receiver in the x direction, GPS 1, Table 8-2. However, because the same GPS is not as accurate in y direction, as it is in x direction, the RMSE of its LFN is more important to the x direction and to its corresponding estimates of its neighbors LFNs.

However, the LFN results of the first vehicle shows that the latter does not track its neighbors as well as they track it. This is true even in the x direction in which it is preconceived to perform quite well because of its GPS receiver accuracy in that specific direction. This is due to the configuration of the cluster; the first vehicle is far away from the other two vehicles that move in the scenario close to each other. The RMSEs of the local fusion nodes converge to some specific offset value that depends on the quality of the vehicles embedded proprioceptive and exteroceptive sensors. These bias errors also depend, in cooperative localization scenarios, on the number of the vehicles in the cluster, on the configuration of the cluster and on the type of the hierarchical fusion structure that is adopted; with/without partial or full feedback. The same convergence is observable in Figure 8-7, in the first scenario concerning the multi-sensors data fusion and adopting a hierarchical structure without feedback. Figure 8-9 also shows that in collaborative localization (CL) scenario, the GFN of SOCI performs better than the LFNs and GFNs of the cluster in terms of RMSEs. This is can be explained by its performance that depends on the traces of the covariance matrices of the LFNs that their respective inverse are used as weights to fuse the three estimates. If the latter are oscillating around the real position of the vehicles the SOCI fusion result is taken roughly as the mean of the three estimates when the traces are very close. However, when the process noise is important, the SOCI often

promotes the LFN of the third vehicle (green) because of the accuracy of its GPS receiver. The CL simulation results, on the GFN of each vehicle, show a slight difference between the ODA and the optimal results of CEIF. This could be due to the process noise that is associated to the model.

Figure 8-10 represents the evolution of the three elements of the LFNs and GFNs covariance matrices that are related to the vehicles position on the x and y axis. It shows that both SOCI and ODA algorithms are consistent, and confirms the optimality of the ODA in term of minimization of the covariance error. It (ODA) solves the track-to-track problem on distributed architectures by assuming that the process noise is minor. As mentioned in [133], when the process noise is important the ODA is no longer consistent. Nevertheless, the numerous simulations showed that RMSE increases progressively with increase of the process noise and the ODA loses its consistency. The SOCI does not perform as well as the ODA, but it performs better than the other LFNs, even the vehicle number 3, which is equipped with the most accurate GPS receiver.

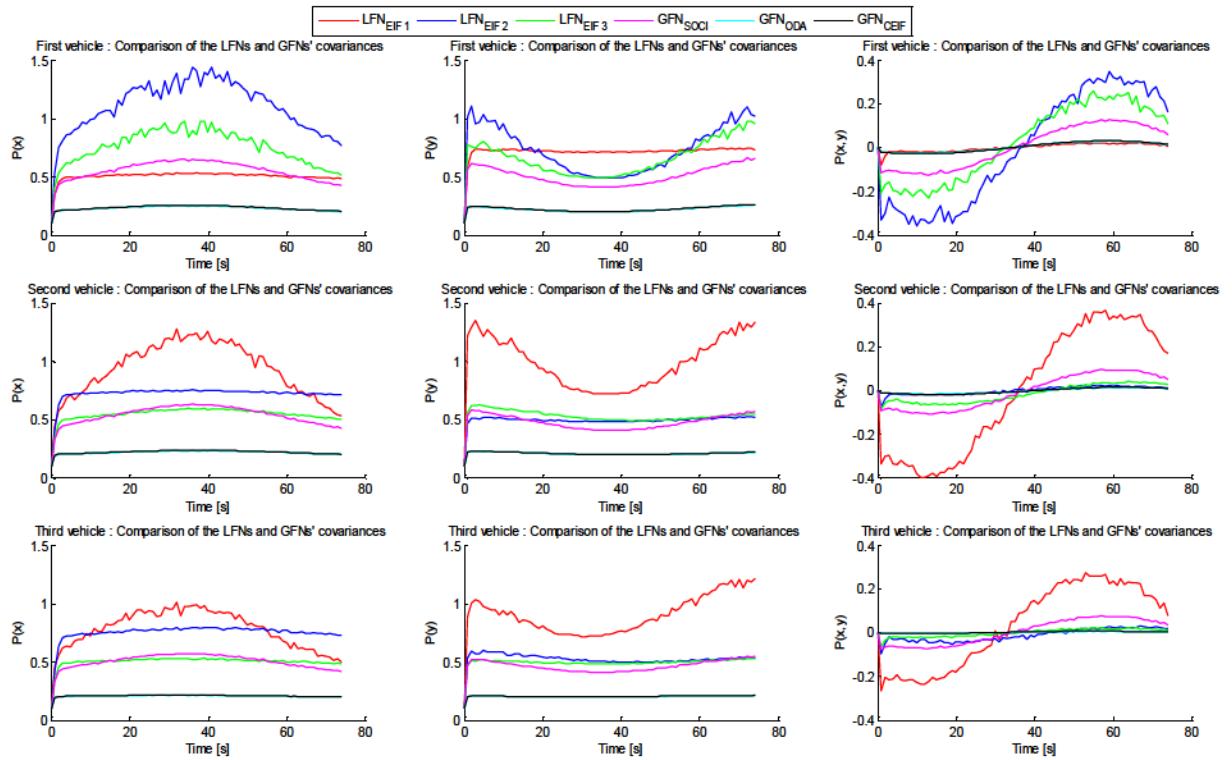


Figure 8-10: Position covariance elements of the local and the global fusion nodes evolution of the three vehicles in cooperative localization scenario

The CL scenario reveals that the increase in the covariance error of LFNs, in the x direction, is due to the translation of the vehicle along the y axis. As a consequence, the y component of the covariance error decreases as long as the vehicles keep moving on the same direction. As soon as the vehicles move in the x direction, the x element of the covariance matrix decreases and the y element increases. As depicted in the Figure 8-10, the (x, y) component of covariance matrix P captures the correlation of the vehicle motion in x and y axis. It also gives an idea on the changing on vehicles' direction, from x to y direction, and vice-versa.

Table 8-2 : GPS receivers' covariance matrices

Sensors	R
GPS 1	$R^1_{GPS} = \text{diag} [ 2.5^2, 3.5^2, (3\pi/180)^2, (2/3.6)^2 ]$
GPS 2	$R^2_{GPS} = \text{diag} [ 3.5^2, 2.5^2, (3\pi/180)^2, (2/3.6)^2 ]$
GPS 3	$R^3_{GPS} = \text{diag} [ 2.5^2, 2.5^2, (3\pi/180)^2, (2/3.6)^2 ]$
Lidars	$R^i_l = \text{diag} [ 0.1^2, (2\pi/180)^2 ]$

## 8.14 Experimental scenario

The experiential scenario consists of equipping a vehicle with a GPS receiver and driving it all along the Versailles Satory trackway. The experiment is being conducted by LIVIC laboratory at IFSTTAR in Versailles, France. For GPS post-processing, the GPS data are then collected and saved in mat files on the hard disk. In this work, the same experimental data are exploited. To test the performance of our suggested algorithms for cooperative localization, the GPS data are then duplicated by creating two other trajectories in order to match with the number of the vehicles in the cluster, as in the case of the simulated scenario, which involves a set of three vehicles as depicted in Figure 8-11. The GPS data are transformed to Cartesian coordinate. The x and y coordinate are then extracted and used as the observation vector in the local and the global fusion nodes. The position precision of the GPS receiver is five meters in each direction, Table 8-3.

The same initial conditions are associated to the different filters that are presented previously in this paper. The inter-vehicular distance is assumed to be assessed by the VANET means with the same accuracy as the GPS receiver, which is due to the use of the GPS data during the assessment of the inter-vehicular distance (due to the lack of the real position of the

vehicles, unlike in the simulated scenario. The covariance matrices of the different sensors are given in Table 8-3). The second assumption concerns the communication delay that is assumed to equal to zero. The VANET allows each vehicle to broadcast its own GPS measurements, local and global fusion nodes' state vectors estimate and their associated covariance and information matrices to its neighbors in the cluster.

To compare the performance of the different algorithms (LFN and GFN), the same idea of creating an optimal central fusion with an EIF to merge the different GPS and VANET observations, is adopted herein. Thus, its fusion result is considered as a reference to evaluate the LFNs and the GFN on distributed architecture performance; as presented previously in the simulated scenario (simulation section).

Figure 8-11 shows the blue vehicle trajectory on the Satory trackway according to the local and the global fusion nodes and to the GPS data. As illustrated in the figure, the gap between the local trajectory estimate of the LFNs (red, blue and green) is very small in straight lines and it

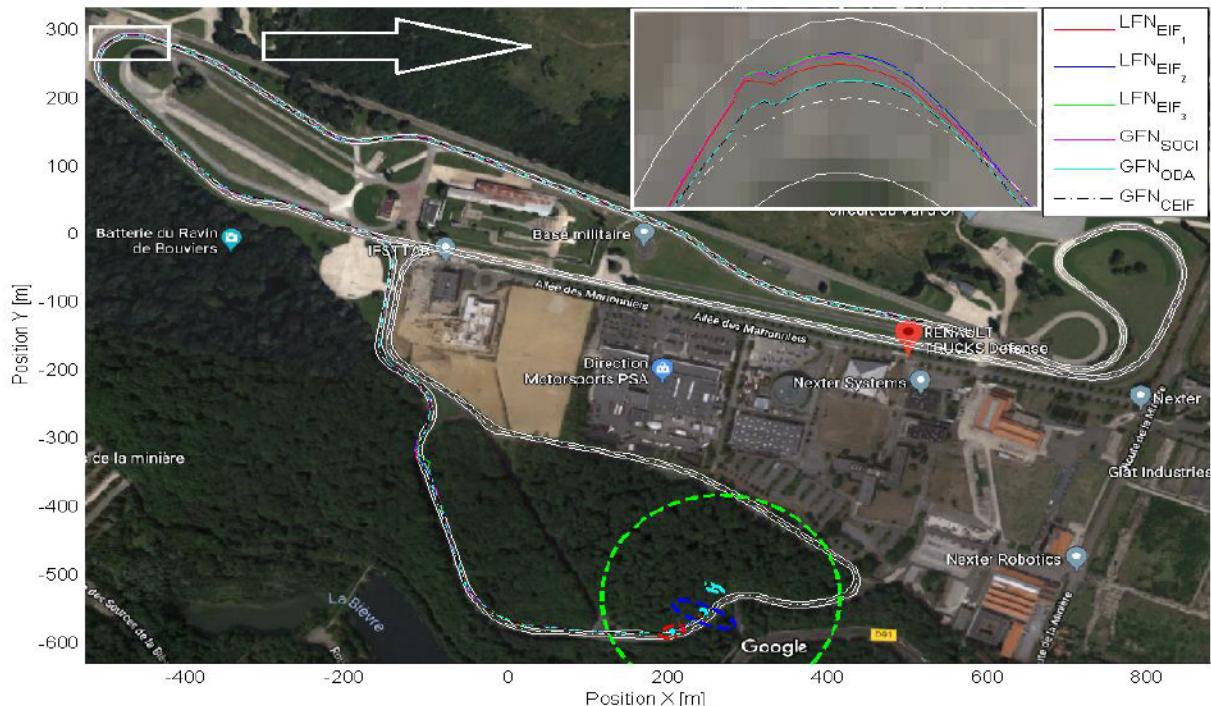


Figure 8-11: Blue vehicle trajectory estimation according to the local and to the global fusion nodes:  $\text{LFN}_{\text{EIF}_1}$  related to the first vehicle (red),  $\text{LFN}_{\text{EIF}_2}$  related to the second vehicle (blue),  $\text{LFN}_{\text{EIF}_3}$  related to the third vehicle (green),  $\text{GFN}_{\text{SOCI}}$  (magenta),  $\text{GFN}_{\text{ODA}}$  (cyan) and  $\text{GFN}_{\text{CEIF}}$  (black)

increases in sharp turns. Their respective convergence is related to the process noise, which is in this work the same for all LFNs, and to the sensors covariance matrices that vary in time (quality of the GPS signal). The same figure shows that the SOCI trajectory estimate (magenta) always evolves between the LFNs estimates while minimizing the trace of the resultant covariance matrix of the extended system for tracking the cluster of the three vehicles. In the meantime, the information matrix fusion algorithm (cyan), on distributed architectures (ODA), shows a better result compared to the LFNs and to the SOCI. Its trajectory estimate is as well as the global CEIF in the straight line and there is a very small gap in sharp turns. However, the performance of the method (ODA) decreases when the GPS receivers lose their respective signals.

Figure 8-12 shows the root mean square error evolution of the different LFNs and GFNs regarding the central EIF. The first peak in the graphs is due to the dynamic of the local filters  $\text{LFN}_{\text{EIF}_i}$ . The LFNs take some time to tune their respective covariance matrices to some steady

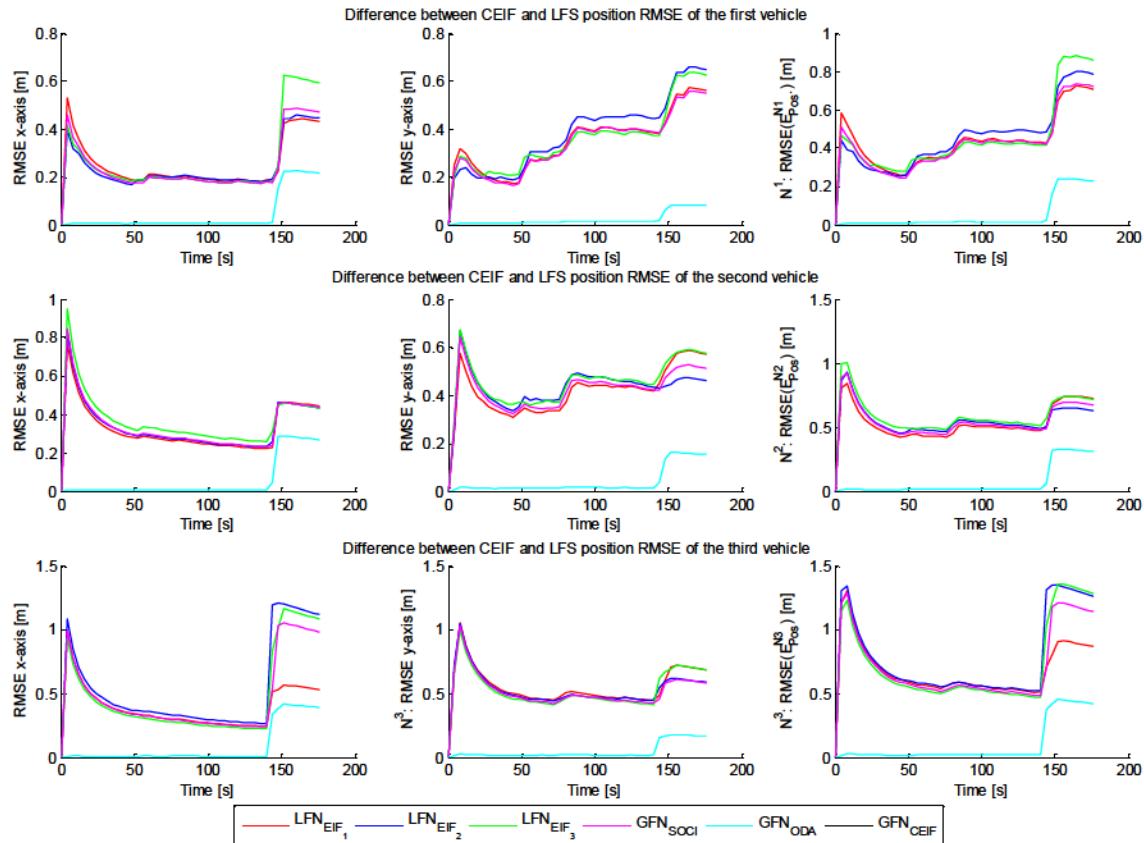


Figure 8-12: Root mean square error of the local fusion nodes and according to the central optimal EIF

values according to the parameters of the process noise and sensors covariance matrices, and to the sensors measurements. Then the RMSEs decrease gradually as their respective estimate converges to the CEIF's. The same figure shows that as long as the sensors measurements are regularly received, the RMSEs evolution in x direction is smaller and steadier than in the y direction. However, when the GPS signal is lost, the RMSEs in x direction brutally increase. This means that the LFNs are very sensitive in that direction. The same figure shows that the RMSEs of the SOCI filter follow faithfully the evolution of the LFNs's. The interesting part of the figure is the RMSE evolution of the information matrix fusion ODA. As long as the sensors measurements are available, its RMSEs do not exceed (0.01, 0.02) meter in x and y directions, and 0.03 meters on the position. However, when the GPS receivers lose their respective signals, the RMSEs rise to (0.4, 0.2) meter in x and y directions, and to the average of 0.3 meters on the position. These results confirm that the performance of the ODA depends on the quality of the LFNs estimates, which are in turn dependant on the sensors measurement quality.

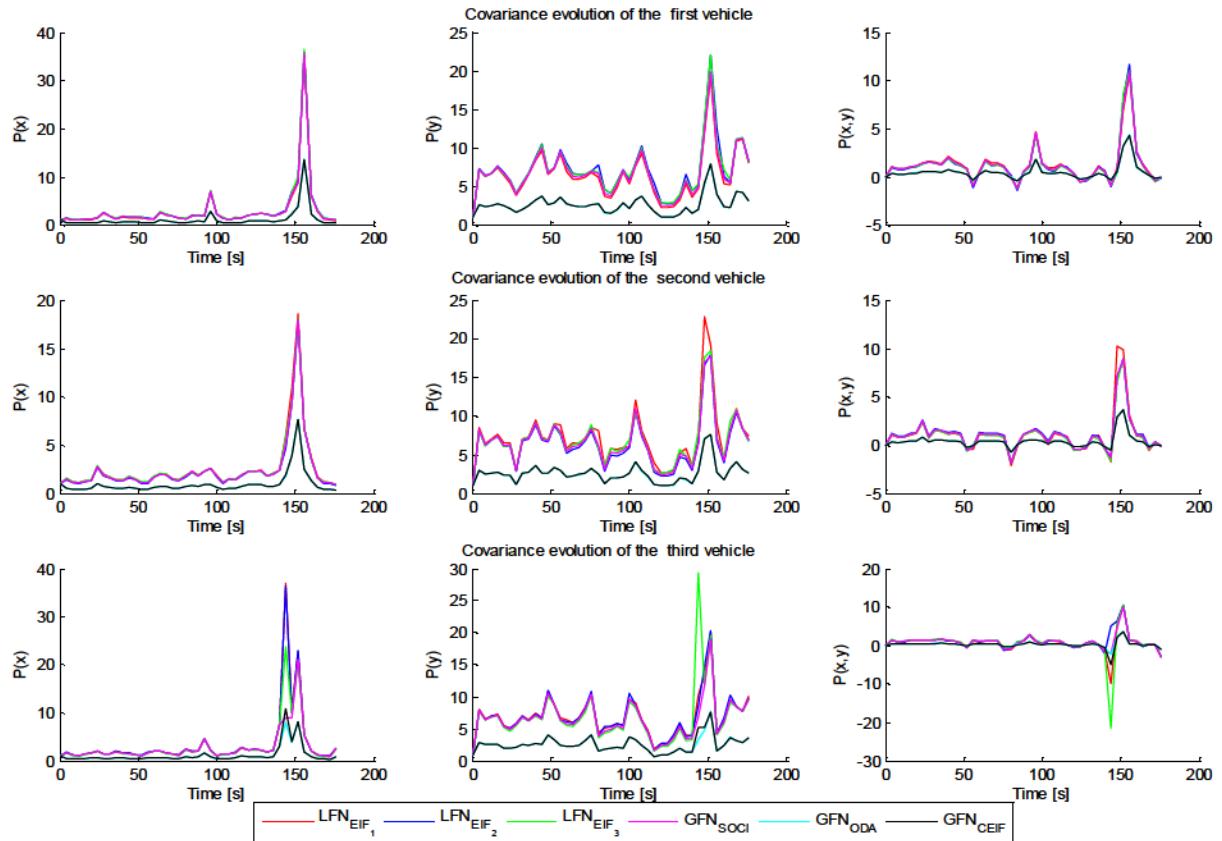


Figure 8-13: LFNs and GFNs covariance matrix elements evolution

Figure 8-13 shows the evolution of the local and global fusion nodes' covariance matrix elements, which are related to the vehicles position in x and y directions. The graphs show that the data fusion results of the GPS receiver data are more accurate in the x direction rather than in y direction, an average ratio of 2.5, and this is true for all fusion nodes. The different peaks on the graphs are the result of the GPS signal loss that is related to some regions of the trackway that are heavily forested as shown in Figure 8-11.

The SOCI fusion node results (magenta) confirm once again the behavior of the CI algorithm by determining the best weights  $\omega_i$  that are associated to the LFNs. Their determination, as presented previously, is based on minimizing the trace of the final covariance matrix. The consistency of the SOCI algorithm is still guaranteed. The comparison of the covariance matrices elements of SOCI and central fusion node (CEIF) shows that the SOCI's are all time greater than the CEIF's. This confirms the consistency of the SOCI even in the case of an experimental scenario. However, some limitations are noticed when the covariance matrices of the LFNs become important, then the vector estimate degrades over the remaining time of the fusion procedure.

The same figure shows that as long as the GPS data are available, the information matrix fusion algorithm (cyan), on distributed architectures (ODA), performs as well as the suboptimal CEIF in terms of the covariance estimation and consistency. Nevertheless, the accumulation of the process noise, when losing the GPS signal for a long time, yields the ODA to lose some of its consistency. This can be observed on the graphs on the third row that deal with the covariance estimation of the third vehicle. When the latter lost its GPS signal (between 140s and 152s), the ODA loses as well some of it consistency.

Table 8-3 : Real GPS receivers' and VANET covariance matrices

Sensors	Covariance matrices
Model	$Q = \text{diag} [ 0.01^2, 0.01^2, (\pi/180)^2, 0.15^2, 0.2^2, 0.2^2 ]$
GPS	$R^1_{\text{GPS}} = R^2_{\text{GPS}} = R^3_{\text{GPS}} = \text{diag} [ 5^2, 5^2 ]$
VANET	$R_{\text{VANET}} = 5^2$

## 8.15 Conclusion

This paper presents a real time cooperative localization method for collaborative and autonomous vehicles using on-board proprioceptive and exteroceptive sensors and the inter-vehicular communication means. In case of the simulated scenario, each vehicle uses its GPS receiver data to estimate its own ego vector state, and a rangefinder to measure the relative distance and the heading angle between itself and its neighbors. It also exploits the V2V communication means to broadcast the relevant information to the cluster of vehicles. It describes the main issues of working with distributed data fusion architectures, such as unknown cross-covariance, which causes over-covariance, due to the fusion of correlated information in a local fusion node. The paper proposes a new method for cooperative localization by adapting the simultaneous localization and mapping solution. The computational complexity of the suggested algorithm is quadratic to the number of the tracking vehicles compared to the usual solution that is based on KF with cubic complexity. A good implementation of the method can improve considerably the performance of the algorithm rather than directly compute the product of spars matrices with the different masks. The suggested strategy is flexible and easily adaptable for cooperative concurrent mapping, localization and tracking on distributed architectures with or without feedback. The proposed solution solves the bilateral transform issue between the moment and information representations, and uses information and state variables during the predicting and updating steps. By resorting to track-to-track fusion method. It also solves the over-covariance issue that is a major challenge in the case of cooperative localization. The simulation results show the efficiency and the consistency of the suggested data fusion algorithms. The strategy is based on a high-level fusion method using two data fusion nodes: an LFN to fuse the proprioceptive and exteroceptive non-correlated sensors' data according to an EIF or an EKF, and a GFN to merge the broadcast information between vehicles according to high level data fusion methods, T2TF among other algorithms. The results of the experimental data confirm once again the efficiency of the proposed methods and highlighted the limitation of the information matrix fusion algorithm in case of the GPS signals loss that induces the ODA to lose some of its consistency. However, some configurations of the cluster lead the LFNs to some instabilities. The future work will investigate the impact of the cluster configuration on the instability of the LFNs and on the extendibility of the proposed methods for heterogeneous data fusion for collaborative localization and mapping.

## 8.16 Appendix A

To prove the inverse of the sum of two matrices lemma (Chapitre 8, section 8.8), let us consider A and B two invertible matrices. Thus we can write:

$$\begin{aligned}
 (A+B)^{-1} &= A^{-1} + X \\
 \Rightarrow (A+B)^{-1}(A+B) &= (A^{-1} + X)(A+B) \\
 \Rightarrow I &= (A^{-1} + X)(A+B) \\
 \Rightarrow I &= A^{-1}A + A^{-1}B + X(A+B) \\
 \Rightarrow 0 &= A^{-1}B + X(A+B) \\
 \Rightarrow X(A+B) &= -A^{-1}B \\
 \Rightarrow X &= -A^{-1}B(A+B)^{-1} \\
 \Rightarrow X &= -A^{-1}B(A^{-1} + X) \\
 \Rightarrow X &= -A^{-1}BA^{-1} - A^{-1}BX \\
 \Rightarrow (I + A^{-1}B)X &= -A^{-1}BA^{-1} \\
 \Rightarrow X &= -(I + A^{-1}B)^{-1}A^{-1}BA^{-1}
 \end{aligned} \tag{A.1}$$

Let us replace X by its expression and develop the new equality:

$$\begin{aligned}
 (A+B)^{-1} &= A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1} \\
 &= A^{-1} - (A^{-1}B(B^{-1}A + I))^{-1}A^{-1}BA^{-1} \\
 &= A^{-1} - (B^{-1}A + I)^{-1}(A^{-1}B)^{-1}A^{-1}BA^{-1} \\
 &= A^{-1} - (B^{-1}A + I)^{-1}B^{-1}AA^{-1}BA^{-1} \\
 &= A^{-1} - (B^{-1}A + I)^{-1}A^{-1} \\
 &= A^{-1} - ((B^{-1} + A^{-1})A)^{-1}A^{-1} \\
 (A+B)^{-1} &= A^{-1} - A^{-1}(B^{-1} + A^{-1})^{-1}A^{-1}
 \end{aligned} \tag{A.2}$$

For invertible quadratic matrices A, B, U and V with appropriate dimensions, we can write:

$$\begin{aligned}
 (A+UBV)^{-1} &= A^{-1} - A^{-1}((UBV)^{-1} + A^{-1})^{-1}A^{-1} \\
 &= A^{-1} - A^{-1}(V^{-1}(UB)^{-1} + A^{-1})^{-1}A^{-1} \\
 &= A^{-1} - A^{-1}(V^{-1}B^{-1}U^{-1} + A^{-1})^{-1}A^{-1} \\
 &= A^{-1} - A^{-1}(V^{-1}(B^{-1} + VA^{-1}U)U^{-1})^{-1}A^{-1} \\
 (A+UBV)^{-1} &= A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1}
 \end{aligned} \tag{A.3}$$

# **Chapitre 9**

## **Online trajectory planning with a modified potential field method on distributed architectures for autonomous vehicles**

### **9.1 Avant-propos**

#### **9.1.1 Auteurs et affiliation**

**Farid Bounini** : étudiant au doctorat, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Membre du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Denis Gingras** : Professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et informatique. Responsable du Laboratoire sur l’Intelligence Véhiculaire (LIV).

**Hervé Pollart** : Responsable du développement commercial de la voiture intelligente, OPAL-RT Technologies inc. Lyon, France.

**Dominique Gruyer** : Chargé de recherche, IFSTTAR, CoSys – LIVIC, Versailles, France.

#### **9.1.2 Date de soumission**

27 avril 2017.

#### **9.1.3 État de l’acceptation**

Version finale publiée.

### **9.1.4 Conférence**

Intelligent Transportation Systems World Congress 2017.

### **9.1.5 Référence**

BOUNINI F., GINGRAS D., POLLART H., GRUYER D, "Online Trajectory Planning With a Modified Potential Field Method on Distributed Architectures for Autonomous Vehicles." World Congress on Intelligent Transportation Systems (ITS WC 2017), Montreal, Canada, October 29<sup>th</sup> – November 2<sup>nd</sup>, 2017.

### **9.1.6 Titre français**

Planification de la trajectoire en temps réel, à l'aide de la méthode des champs de potentiels artificiels modifiée, sur les architectures distribuées pour les véhicules intelligents.

### **9.1.7 Contribution au document**

La contribution principale de ce chapitre réside dans la méthode d'exploitation des données des capteurs embarqués à des fins de navigation d'un véhicule intelligent. Le degré d'autonomie de ce dernier, dans un environnement dynamique, est alors déterminé essentiellement par sa capacité à déterminer sa trajectoire en temps réel de sorte à prévenir et à éviter les éventuelles collisions. La première partie de la méthode proposée dans ce chapitre présente une solution au problème principal de la méthode des champs de potentiels artificiels : les minimums locaux. Ces derniers bloquent le processus de recherche d'une trajectoire viable, empêchant ainsi l'algorithme d'aboutir et donc aux véhicules d'atteindre leur destination finale. La deuxième partie reformule et adapte la méthode pour des scénarios véhiculaires sur des architectures hiérarchiques ; décentralisées et distribuées. La parallélisation de la conception des modèles des champs de potentiels attractifs et répulsifs, sur différents coeurs de calcul, a permis de réduire le temps d'exécution de la méthode à celui de la méthode standard A-Étoile « A\* », permettant ainsi à la méthode proposée de s'exécuter en temps réel.

### **9.1.8 Résumé français**

Ce chapitre décrit une nouvelle approche de la planification de la trajectoire en ligne. Elle est basée sur une méthode des champs de potentiels artificiels modifiée, appliquée à des

architectures distribuées pour les véhicules intelligents et autonomes. L'approche permet de contourner le problème de la méthode des champs de potentiels artificiels (MCPA), dû principalement aux minimums locaux qui amènent la MCPA à des états bloquants. L'avantage de cette méthode, comparativement à celles qui recourent aux méthodes d'optimisation globales, est le temps de calcul qui avoisine celui de la méthode A\*. La stratégie consiste à rechercher un chemin plausible dans le champ des potentiels artificiels à l'aide de l'algorithme de descente du gradient des potentiels (ADGP). La démarche se résume à ajouter un potentiel répulsif, à chaque fois qu'une configuration bloquante, un minimum local, se présente. Après que l'ADGP a atteint le minimum global, un nouveau champ de potentiel artificiel est créé. Ce dernier est façonné de sorte à avoir un seul minimum correspondant à la destination finale du véhicule (minimum global). Enfin, l'ADGP effectue une deuxième itération pour déterminer une trajectoire viable. En outre, d'autres modifications sont proposées pour adapter la méthode MCPA à une architecture distribuée pour le traitement parallèle. Les résultats préliminaires montrent que la méthode MCPA pourrait être une bonne solution pour les véhicules intelligents et autonomes dans un environnement dynamique.

## **9.2 Abstract:**

This paper presents an online trajectory planning strategy with a modified potential field method on distributed architectures for autonomous vehicles. The approach overcomes the well-known artificial potential field method (APFM) issue, which is due to local minima that induce the APFM to stick in. The asset of the new proposed method reverse to those that resort to the global optimization methods is the low computing time, which borders up the A-Star method. The strategy consists of looking for a realistic path in the potential field — according to the potential gradient descent algorithm — and affix a repulsive potential, to the current state, in the case of blocking configuration, a local minimum. When the PGDA reaches the global minimum, a new artificial potential field will be fashioned with only one minimum which matches the final destination of the vehicle, the global minimum. Finally, to determine the achievable trajectory, a second iteration is performed by the potential gradient descent algorithm. In addition, other modifications are put forward to adapt the MAPF method on distributed architecture for parallel computing. The preliminary simulation results show that the MAPF method could be a good solution for self-driving vehicles in a non-static environment.

## **9.3 Introduction**

Relying on proprioceptive and exteroceptive sensors measurement and artificial intelligence, intelligent and autonomous vehicles are able to learn, understand and interact with their surroundings. They use their sensors measurements to locate themselves and navigate in their environment safely [69], [119]-[121]. Today, several driver assistance systems (ADAS) are ubiquitous and operational on intelligent vehicles [121]-[122]. Their main role is to assist the driver to better understand his environment and minimize the risks and the impact of accidents. Overall, ADAS are divided into two categories: 1- Active ADAS: They include all functions and maneuvers that are performed directly by the vehicle to avoid or minimize the impact of any potential accident, and 2- Passive ADAS: They are involved after the occurrence of the accident. It warns the concerned authorities (hospital, fire departments, police stations, etc.) by sending the coordinates of the accident site, the material and human damage, etc. Another aspect of the artificial intelligence that features an autonomous vehicle is to move from its current position to any desired destination without involving human driver [119]-[120]. In other words,

the vehicle moves independently. Most decisions are fully taken autonomously. There are mainly three aspects that characterize most an autonomous vehicle:

- Perception: The navigation, in an instructed environment, requires a perfect understanding of the vehicle's surroundings. The vehicle must imperatively identify, classify and determine the positions of each mobile and static obstacle at each time stamp ;
- Decision: The sensors measurements are handled by a set of algorithms (filtering, position predicting, path planning, and decision-making.). Indeed, an autonomous vehicle should define a sequence of actions leading him from its current position to its final destination. In real environment, the vehicle is subjected to many uncertainties [119], [122]. Sensors' measurement will, as accurate as they are in the short-term, certainly cause deviation in the long-term. Subsequently, the cumulative errors result in localization inaccuracy, occasioning the vehicle to get gradually far away from its pre-planned path. The countermeasure to such issues is to fuse the exteroceptive and proprioceptive sensors' measurements, and include the vehicle dynamic's constraints and the digital map of its environment [121]-[122] ;
- Action: Once the decision is made, and a satisfactory trajectory is defined, the control signals are generated and transmitted to the actuators to act on vehicle dynamics. This allows the vehicle to avoid static and mobile obstacles, and change its course according to the new environment configurations in real time.

The paper describes a new strategy for intelligent vehicle real-time navigation in static and highly dynamic environment on distributed architecture. The approach exploits a modified artificial potential field method (MAPFM) to overcome the local minima issues that are due to the vehicles environment configuration, and determine a realistic trajectory for the vehicle path planning on distributed architecture. It elaborates on [142]. The remainder of the paper includes eight parts: The first part presents some related work on trajectory planning methods that stand on APF method. The second part depicts and highlights the advantages and the drawbacks of two known planning trajectory algorithms; A-Star ( $A^*$ ) and APF methods. The third part highlights the most constraining drawback of the APF method, which is local minima issues. The fourth part depicts, compares and discusses the simulation results of APF and  $A^*$  methods. The fifth and the sixth part highlights the adapted MAPF for vehicle path on distributed

architecture. The seventh part describes the simulated scenario and the results of the MAPF on the distributed architecture. The last part completes the paper with a conclusion and future works.

## 9.4 Trajectory planning methods

One of the most important aspects of intelligent vehicles and robots is the perception of their surroundings. As presumed in the last work [142], the proprioceptive and exteroceptive sensors' data fusion is assumed to be already performed according to specific algorithms. Then, the results are used for path planning. The procedure shapes the vehicle surroundings and locates it in. Then, it calls the trajectory planning algorithms to map out a practical path. The two following sections are a summary of the paper [142]. The path planning methods are numerous, most of the strategies are highlighted in [143]. This section describes two of trajectory planning methods: "A\*" and "APF" methods, as suggested in [142].

### 9.4.1 A-Star (A\*)

A\* is one of the most effective trajectory planning methods. It is used in the robots navigation in congested environment and in numerous digital games such as mazes [144]. The fundamental idea is to find the shortest path to the global state. The strategy resorts to Euclidean distance to evaluate the distance between the different states. The basic algorithm does not take into account the obstacles between the current state and the final state. The procedures are: 1- To look for other paths when the robot is in a blocking state; 2- To avoid any path leading to blocking nodes. To do so, two path lists are created. The first is the OPEN list, it includes the set of paths to explore the parent nodes and find the optimal path. The second is the CLOSED list leading to dead nodes. Sometimes, the path searching is time-consuming. To overcome such constraints, optimization criteria are associated with the planning issue.

### 9.4.2 Artificial potential field method

As depicted in [142], the basic concept of the vehicle motion, in the APF, may be interpreted as a moving particle in a field of two electrically charged particles with different signs [143]. Analogically, the positive particle is the vehicle. The negatively charged particle corresponds to global state, and the same charges, as the vehicles', are assigned to the obstacles. The field

potential gradient may be interpreted as forces that guide the positive particle to reach the position of the negative one. The obstacles create repulsive forces to keep the vehicle away from obstacles [145]-[146]. At each configuration  $Q_i$ , the positive APF forms an attractive force. It denotes the favorable direction of the vehicle motion to reach the global position  $Q_g$ . The combination of the attractive and repulsive forces drives the vehicle to its destination safely.

#### 9.4.2.1 Attractive potential field

As listed in [142], many forms of the positive and negative potential exist. The most commonly used throughout the literature of positive APF are parabolic and conical functions. The procedure takes advantage of the parabolic form when the vehicle is far away from its destination, and of the conical form, once the vehicle gets closer to its destination according to a specific distance  $R_G$ . The mathematical expression (9.1) formulates the shape of the positive APF.

$$U^a(Q) = \frac{1}{2} k^a \begin{cases} \|d(Q)\|^2 & \text{if } d(Q) < R_G \\ \|d(Q)\| & \text{if } d(Q) > R_G \end{cases} \quad (9.1)$$

where  $k^a > 0$  is a positive constant,  $d(Q) = Q_g - Q$  is the distance between the global state and the current state,  $Q_g = [x_g \ y_g]^T$  and  $Q = [x \ y]^T$  in 2D.

The function (9.1) is a positive monotone function, and its global minimum equals to zero. The APF shape provides the vehicle with the best direction and the optimal path. The potential gradient is defined as a proportional vector of the difference between the global  $Q_g$  and the current states. The further the vehicle is from its endpoint, the greater is the gradient, and the bigger the attractive forces. The resulting force of this field is the negative gradient of the attractive APF (9.2). The force decreases as well as the vehicle is getting closer to the global state, and once there, the force becomes null. This process is achieved repeatedly for each destination.

$$f^a(Q) = -\nabla U^a(Q) = -k^a \begin{cases} d(Q) & \text{if } d(Q) < R_G \\ \frac{d(Q)}{\|d(Q)\|} & \text{if } d(Q) > R_G \end{cases} \quad (9.2)$$

#### 9.4.2.2 Repulsive potential field

Several forms are quoted in literature. The hyperbolic function is introduced for the first time by Khatib [147]. It associates to every obstacle a repulsive charge and affects each spatial cell according to (9.3).

$$U_i^r(Q) = \begin{cases} \frac{1}{2} k_i^r \left( \frac{1}{n_i(Q)} - \frac{1}{n_i^o} \right)^2 & \text{if } n_i(Q) \leq n_i^o \\ 0 & \text{otherwise} \end{cases} \quad (9.3)$$

where  $k_i^r > 0$  is a positive constant,  $n_i^o$  is the distance of influence of the  $i^{\text{th}}$  obstacle,  $n_i(Q)$  is the smallest distance between the current vehicle state and the  $i^{\text{th}}$  obstacle.

$$n_i(Q) = \min_{Q' \in Q_i^o} \|Q - Q'\| \quad (9.4)$$

The repulsive artificial potential is defined as being zero outside the range of the objects' influence, positive inside, and infinite above the obstacles. Due to the proximity of the global position to the obstacles, the combination of the attractive and repulsive APF creates a local minimum in the neighborhood of the final state. The solution is to multiply the repulsive potential by the distance separating the two: the current and the final states [148].

$$U_i^r(Q) = \begin{cases} \frac{k_i^r}{2} \left( \frac{1}{n_i(Q)} - \frac{1}{n_i^o} \right)^2 d(Q) & \text{if } n_i(Q) \leq n_i^o \\ 0 & \text{otherwise} \end{cases} \quad (9.5)$$

The total potential field and forces are defined as the sums of two attractive and repulsive fields and forces, respectively.

$$\begin{cases} U^t(Q) = U^a(Q) + U^r(Q) = U^a(Q) + \sum_{i=1}^n U_i^r(Q) \\ f^t(Q) = f^a(Q) + f^r(Q) = f^a(Q) + \sum_{i=1}^n f_i^r(Q) \end{cases} \quad (9.6)$$

## 9.5 Trajectory planning technique with APF approach

An optimization problem of path planning can be formulated according to the different vehicle dynamic constraints. The APF method parameters ( $k^a$ ,  $k_i^r$ ,  $n_i^o$ ,  $R_G$ ) could be adapted in

accordance with the trajectory planning constraints. The exterior forces that are applied to the vehicle equal to the total forces of the APF. Then, Newton's second law of motion can be applied to the vehicle as given in (9.7).

$$\ddot{Q} = \frac{1}{m} f^t(Q) \quad (9.7)$$

## 9.6 Artificial potential field method and local minima

As mentioned previously, the major drawback of APF method is local minima [149] that are related to the vehicle's workspace configuration and especially to weight coefficients, associated to each obstacle during the APP design [147]. To overcome this issue, the paper elaborates on the solution presented previously in [142]. The idea consists of designing the total APF as the classic method according to equation (9.6). The strategy uses the potential gradient descent algorithm (PGDA) to discover the vehicle path. The PGDA is executed as long as the ongoing state is different from the global minimum. If PGDA cannot go forward, blocked in a local minimum, then the algorithm adds extra repulsive potential to the current state until it gets free, according to the set of equations (9.5). Once the vehicle destination is reached, the PGDA searches, once again, the final and practicable path in the new potential field.

The primary MAPF method skeleton for online trajectory planning is given in [142], as follows:

- 1: Design the attractive PF “ $U^a$ ” according to global state.
- 2: Design the repulsive APF “ $U_i^r$ ” according to each obstacle and its weight “ $k_i^r$ ” and  $n_i^o$ .
- 3: Assign the initial state  $Q_i$  to the path vector.
- 4: **while** ( $d(Q) \neq 0$ ) **do**
- 5:     Call the potential gradient descent algorithm to determine the next state.
- 6:     **if** (PGDA is blocked AND  $d(Q) \neq 0$ ) //local minima
- 7:         Add a new repulsive APF “ $U_i^r$ ” to “ $U^t$ ”.
- 8:     **else**
- 9:         Add the current state to the path vector.
- 10:     **end if**
- 11: **end while**
- 12: Call the potential gradient descent algorithm (PGDA) to determine the new path

## 9.7 Comparison of MAPF and A\* methods

To compare the computation efficiency of the MAPF and A\* methods, a vehicle workspace of ten meters wide and eighty meters long is fashioned on Matlab. The obstacles in the vehicle environment of the vehicle create two different local minima, between the initial vehicle position and its destination, as illustrated in Figure 9-1 and Figure 9-2. The vehicle workspace is partitioned into a set of squares of  $0.25 \times 0.25$ m ( $h=0.25$ m). During the first execution of the PGDA, the algorithm is attracted by the closest minimum. Consequently, the MAPF strategy gradually removes the local minima until the PGDA reaches the global state as shown in Figure 9-1 and Figure 9-2, with blue circles. The second operation of PGDA allows MAPF to recover the satisfactory path, which will be a reference path for vehicle planning trajectory. The simulated scenario is created with Matlab R2012a on windows 7 and Intel computer (Intel® Xeon™ CPU E3-1245 v5, CPU @3.50 GHz and RAM 16Go). The mean execution time of one hundred runs of the two methods (MAPF and A\*) are 65.638 and 64.264ms, respectively. The comparison of the these results to those of [142] shows that the A\* execution time depends on the size of the vehicle's workspace and the MAPF depends more on the number of local minima to eliminate and their size.

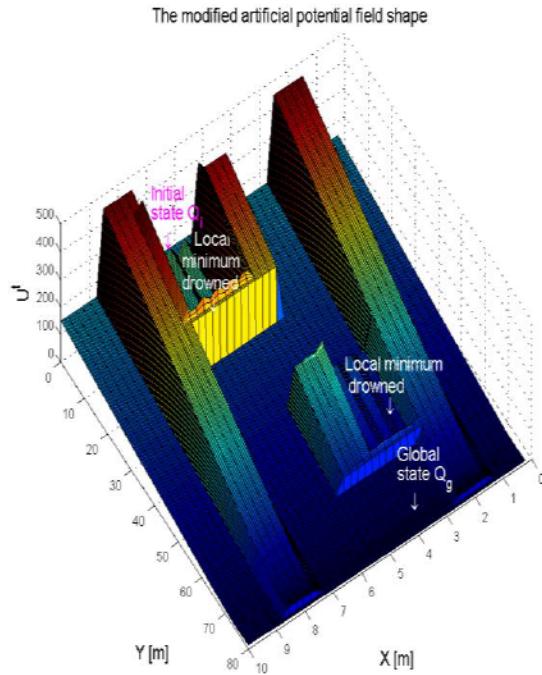


Figure 9-1: Modified potential field shape and local minima removing

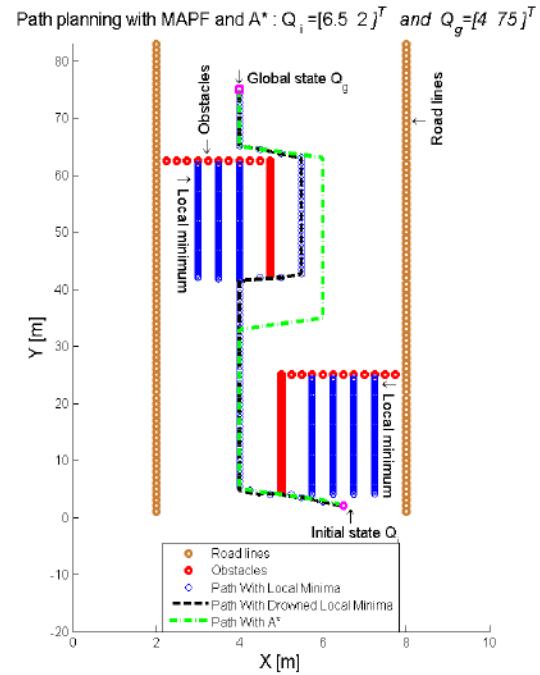


Figure 9-2: Path planning with MAPF and A\* method

## 9.8 Adapted MAPF to vehicle trajectory planning

To fit the vehicle dynamic, extra attractive potential field is added to the static and dynamic obstacles. This is according to the relative distance and speed between the hosted vehicle and the obstacles (other vehicles, roadside barriers and markers, etc.). The effect of the relative speed takes places on the obstacle coefficient, in the longitudinal and lateral direction, as follows:

$$\begin{cases} d_s^{Long} = d_0^{Long} & \text{if } v_i^{r|Long} > 0 \\ d_s^{Long} = d_0^{Long} + v_i^{r|Long} T_d & \text{if } v_i^{r|Long} \leq 0 \\ N^{Long} = d_s^{Long} / h \end{cases} \quad (9.8) \quad \begin{cases} d_{s,i}^{Lat} = d_0^{Lat} + |v_i^{r|Lat}| T_d \\ N_i^{Lat} = d_{s,i}^{Lat} / h \end{cases} \quad (9.9)$$

where:  $d_0^{Long}$  and  $d_0^{Lat}$  are the minimum, longitudinal and lateral, inter-vehicular distance;  $v_i^r(v_i^{r|Long}, v_i^{r|Lat})=v_i^{target i} - v^{hv}$  is the relative speed of the  $i^{\text{th}}$  target according to the hosted vehicle;  $T_d$  is the driver's reaction time —  $T_d=0.5\text{s}$  for a human driver, and for autonomous system  $T_d$  is set to the maximum time stamp of its data processing that can be expressed in milliseconds —; and  $h$  is the unitary length/size of the sampled space, thus, different values could be assigned to the sampling space in the longitudinal and lateral directions, respectively.

To direct the vehicle from its initial to its final position, the shape of the repulsive APF of the static and dynamic obstacles, which are on the road, is designed with the same curvature as the road's ones in order to smooth the APF path. To reduce the computation time of the attractive APF, a modified version of the parabolic APF according to its formulation (9.1) is given as follows:

$$U^a(Q) = \frac{1}{2} k^a \begin{cases} (x_Q - x_{Q_g})^2 & \text{if } d(Q) < R_G \\ |x_Q - x_{Q_g}| & \text{if } d(Q) > R_G \end{cases} \quad (9.10)$$

This new formulation allows to design a valid attractive APF for all configuration of the initial and global position of the vehicle. The initial position is set, herein, to the current vehicle position and its final position is set to the maximum laser scanner's range in the x direction and to the center of the  $i^{\text{th}}$  road lane in the y direction. Consequently, the effect of the attractive APF (AAPF) takes place just in the x direction and forces the vehicle to always move forward. The shape of the repulsive APF (RAPF) allows the vehicle to always stay on the road. The curvature

of RAPF, which is associated to the obstacle, smooths the proposed MAPF method path, especially in sharp turns. Moreover, other criteria could be associated to the MAPF method so that the PGDA method favors a specific center of road lanes in accordance with the vehicle's current position that will be presented by extra attractive APF ( $U_{ex}^a$ ) and will be placed on all along the desired road lane center. This criterion acts as a guide to the PGDA algorithm from the current position of the vehicle to its global position.

## 9.9 Path planning on distributed architectures

To adapt the MAPF method to our project, we suggest parallelizing the attractive and repulsive potential field design on different cores. For instance, a core will be assigned to compute the attractive part of the APF, additional cores could be assigned to construct repulsive APF according to the road lines. The coordinates of the road lines will be delivered, for instance, by an accurate on-board map according to vehicle local or global coordinates, which could be delivered by global and local proprioceptive sensors. Another strategy of determining the road lanes coordinates, regarding the vehicle's ones, is to fusion the vehicle proprio and exteroceptive sensors (Lidars, cameras, Inertial Navigation Systems INS, and Global Position Systems GPS) and apply appropriate classification methods to the various objects around the vehicle in order to assort them according to their size and speed.

However, such methods are extremely time consuming. Thus, in this study, we resort to the first method that uses an accurate on-board numerical map of the racetrack. To this end, the vehicle is equipped with a GPS/INS unit to locate itself in the global and local plans. The GPS/INS unit data are fused according to an Extended Kalman filter (EKF). The obstacles are detected by the embedded laser-scanner and processed according to the vehicle position estimate in some auxiliary cores. Finally, another core (in series) is used to fuse the two APFs and determine the vehicle itinerary with the potential gradient descent algorithm.

At this stage, the main challenge of the proposed method is the CPUs (cores) synchronization and the real-time execution. To this end, we previously developed a real-time simulator for collaborative and autonomous vehicles based on Pro-SiVIC and RT-LAB platform. ESI-CIVITEC provides us a real-time simulator of infrastructure environment, vehicles' dynamic, and embedded sensors (Inertial Navigation System, odometers, Light Detection And Ranging,

camera, etc.), named Pro-SiVIC. This simulator allows the user to conceive various scenarios for the real-time simulation of the advanced driver assistance systems (ADAS), intelligent and autonomous and even more for connected and collaborative vehicles. The user can activate several operation modes instantaneously, which can be modified during the simulation: to switch a sensor on or off, to record data in a file, and use Data Distributed Store “DDS” mode to send sensor data for external applications such as Orchestra.

Orchestra extends the RT-LAB connectivity capabilities to co-simulation, written in distinctive programming languages or generated by various simulation tools. In this study, Orchestra is mainly used to synchronize the different tools executed during the simulation scenarios (Pro-SiVIC, RT-LAB, C/C++, Matlab/Simulink), to broadcast the vehicles’ state estimates, to route signals with specific subsystems (cores) and Orchestra framework, and for inter-vehicular communication purposes, etc.

The RT-LAB allows the user to simulate the embedded real-time data processing, such as data fusion of embedded proprio and exteroceptive sensors with the on-board map for accurate ego and objects localization. OPAL-RT Technologies’ products are used for parallel computing and to ensure the real-time execution. The two platforms — RT-LAB and Pro-SiVIC — are complimentary [112]. Their association creates a powerful simulator for real-time simulation of ADAS, intelligent/autonomous and collaborative vehicles. For more information, the reader is referred to [112].

The vehicle path planning skeleton, on distributed architectures, is derived from the previous algorithm for online path planning.

- 1: On the first core: design the attractive APF “ $U^a$ ” according to : the global state; the selected road lane center; the laser scanner maximum range and the selected method.
- 2: On the second core: design the repulsive APF “ $U_i^r$ ”, in parallel with the first core, on the distributed architecture, according to: each obstacle and its relative position and speed n.<sup>o</sup> ( $N_o^{Long}$ ,  $N_o^{Lat}$ ), and weight coefficients “ $k_i^r$ ”.
- 3: On the third core:
- 4:     Retrieve the first two cores results and assign the initial state  $Q_i$  to the path vector.
- 5:     **while** (  $d(Q) \neq 0$  ) **do**: //while the vehicle current position is different from the  $Q_g$ .
- 6:         Call the potential gradient descent algorithm to determine the next state.

```

7:      if (PGDA is blocked AND d(Q) ≠ 0) //local minima
8:          Add a new repulsive APF “Uir” to “Ut”.
9:          if (PGDA is blocked N time in the same state)
10:             print: PGDA cannot overtake the obstacle.
11:             break
12:         end if
13:     else
14:         Add the current state to the path vector.
15:     end if
16:   end while
17: Call the potential gradient descent algorithm to determine the new path.

```

## 9.10 Simulation scenario

The main scenario consists in generating a virtual environment — set here to the standard ProSiVIC racing circuit named “HorseRing” — including bitumen road, traffic barriers, landscape, etc. Three cars are involved in this scenario. They are all equipped with proprioceptive sensors: a global position system (GPS) and inertial measurement unit (IMU), attached to the car chassis that measure the position, velocity, acceleration and orientation of the vehicle, and with an exteroceptive sensor, a laser scanner, to scan its environment, detect the static and moving obstacles and measure the relative distance between the hosted vehicle and the obstacles. The acquisition time stamp of the sensors ( $T_s^{GPS}$ ,  $T_s^{IMU}$ ,  $T_s^{laser}$ ), and the parameters’ value of the proposed path planning method are given in Table 9-1.

The time stamp of the cores that are used for vehicle localization and path planning, the time stamp of each block (subsystem), is set to 10 ms. The sensors’ data fusion is processed on the first core, then the results are transferred to the next two cores to compute the repulsive and attractive artificial potential fields,  $U^r$  and  $U^a$ , respectively. Finally, the two APFs are sent to the fourth core to fuse the two APF, find and plot the final vehicle path, as presented in the Figure 9-3. The computation time, for vehicle trajectory planning process, equal to the sum of the computational time of all the blocks in the cascade. For the cores that handles the two APFs design in parallel, the computation time is chosen as the maximum of both. It gives  $T_t = T_{N1} + \max ([T_{N2}, T_{N3}]) + T_{N4}$ . As RT-LAB insures the real time execution according to the

specified time stamp, as stated above, the time stamps are set herein to 10 ms, that makes  $T_t = 3 * T_{Ni} = 30$  ms.

The simulation results, Figure 9-4, show that the selected path, the green trajectory, by the potential gradient descent algorithm, incites the vehicle to always move forward, as expected, until it meets the center of the right road lane and gets repelled by the repulsive artificial field of the right road border. Once the PGDA stuck in the same state, behind the obstacle represented, in the same figure, by the red vehicle, the proposed method affixes some additional repulsive APF to the current state and looks for the next state by furthering the vehicle motion in the x and y directions. The same operations are performed repeatedly until the PGDA reaches the global or the final state. The effect of the inter-vehicular speed, as expressed previously by the two sets equations (9.9) and (9.10), are visible at each time the vehicle gets blocked and needs to move forward and change lane. As it can be seen in the right part of the Figure 9-4, the parameter  $h$ , which is the unitary length (size) of the sampled space, determines how smooth will be the vehicle trajectory. However, its value induces a new trade-off between the smoothness of the trajectory and computing time of the MAPF method. Thus, the smaller is the value of  $h$ , the smoother is the trajectory and the greater is the computing time, and vice versa. A solution to smooth the vehicle trajectory could be an EKF using a basic model of a 2D moving object, or a bicycle kinematic model.

Table 9-1: Simulation parameters.

Parameter	Value	Parameter	Value
$d_0^{Long}$	5 [m]	$T_s^{GPS}$	100[ms]
$d_0^{Lat}$	2.5 [m]	$T_s^{IMU}$	10[ms]
$h$	0.25 [m]	$T_s^{laser}$	20[ms]
$k^a$	1	$k^r$	1

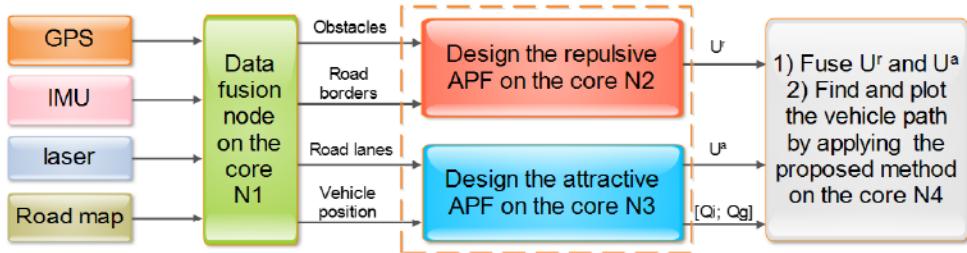


Figure 9-3: Distributed MAPF on Pro-SiVIC and RT-LAB platform

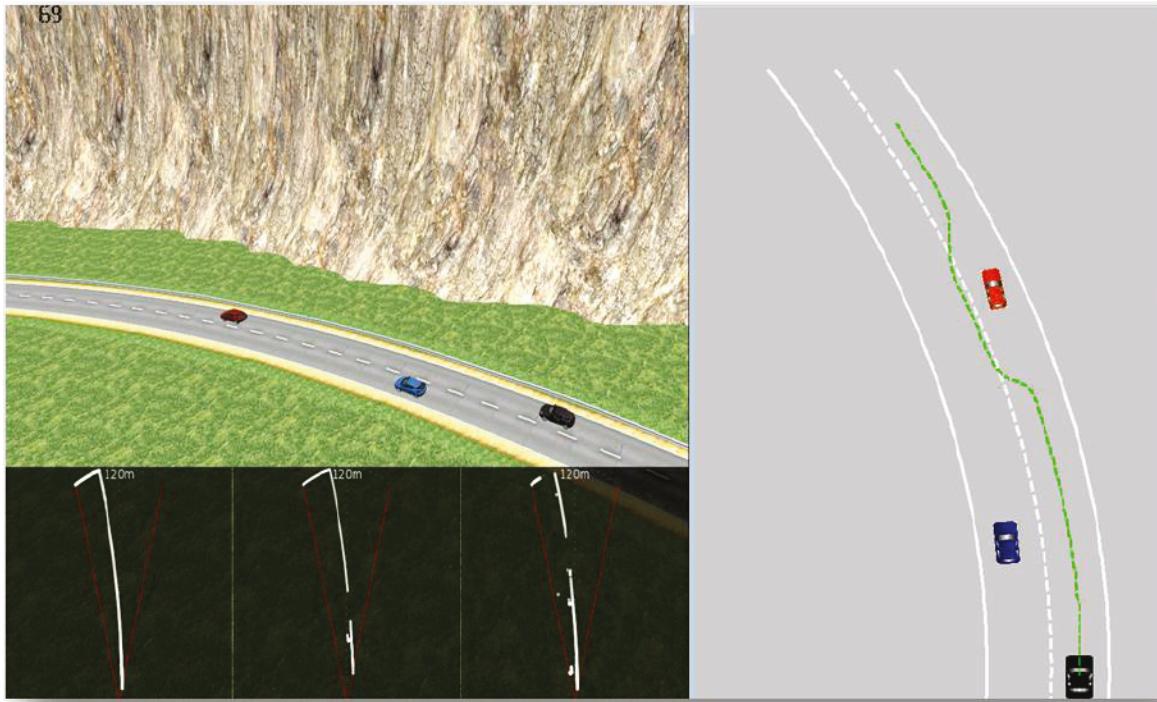


Figure 9-4: Trajectory planning for intelligent vehicles.

## 9.11 Conclusion

This paper deals with online trajectory planning with a modified potential field method on distributed architecture for intelligent vehicles, based on [142]. The approach adds dynamically repulsive artificial potential field to the standard APF, each time the vehicle gets trapped in a local minimum. The comparison between the A\* and MAPF performances shows the effectiveness of the MAPF method and its alignment with real-time path planning. However, as mentioned in [142], the MAPF method computation time depends on the number and the size of the local minima. A solution to this issue is to assign an adequate distance of influence  $n_i^o$  of each local minimum, in x and y directions, according to the size and the relative speed of the object. In addition, other criteria are proposed to reshape the repulsive artificial potential field according to the road curvature, and to specify some favorite direction. To reduce, in meanwhile, the computing time of the MAPF method. The different steps are computed separately on different cores on a distributed architecture. The future work will focus on the integration of the bicycle dynamic model in the path planning, and the design of an advanced controller having as an input vector the MAPF's trajectory as a reference path.

# **Chapitre 10**

## **Conclusion générale**

### **10.1 Bilan**

La croissance importante du nombre de véhicules est la cause principale de la saturation des infrastructures routières. Le trafic routier important, les accidents et la pollution sont tous des conséquences directes de cette croissance. Pour y remédier, les constructeurs automobiles se sont orientés, entre autres, vers le développement de systèmes d'aide à la conduite dont le rôle est à la fois d'assister le conducteur, et d'éviter les situations dangereuses. Le succès, la diversification et l'évolution rapide de ces systèmes font d'eux des solutions incontournables aux problèmes de la circulation routière. Cependant, la quantité gigantesque de données nécessaires à la validation des ADAS, leur temps de développement important et leur coût exorbitant sont des défis posés que les constructeurs cherchent constamment à surmonter. Dans ce contexte, la simulation temps-réel pourrait servir de tremplin à la fois dans la recherche académique, à la fois dans l'industrie automobile, afin de développer et de concevoir de nouveaux produits innovants tout en adressant les défis posés par la complexité de ces derniers. C'est dans ce cadre que s'inscrit notre projet de recherche, dans laquelle la démarche est caractérisée par les deux volets suivants.

Le premier comprend la recherche bibliographique et présente un certain nombre de travaux qui nous semblent les plus propices au projet. Ainsi, la partie de l'état de l'art est orientée principalement sur trois axes : les systèmes d'aide à la conduite équipant les véhicules intelligents mis sur le marché nord-américain ; les travaux les plus récents et les plus poussés menés sur les véhicules intelligents ; les plateformes de simulation de la dynamique des véhicules, des capteurs embarqués et des communications. À la lumière de l'étude des besoins de calcul de ces systèmes intégrés, deux plateformes parmi les plus adaptées ont été sélectionnées (RT-LAB et Pro-SiVIC) pour instaurer une preuve de concept de la nouvelle plateforme. Par la suite, d'autres outils de simulation ont été intégrés à la nouvelle plateforme.

Nous citons notamment : PreScan, CarMaker et EXata. Trois différents scénarios ont été développés par les membres de l'équipe d'I-Car chez OPAL-RT Technologies, ceci à des fins de démonstration pour la 9<sup>ème</sup> conférence internationale sur la simulation temps réel qui est organisée par OPAL-RT (RT17) [150].

Le deuxième volet aborde trois principaux problèmes liés au traitement des données et à la planification de la trajectoire sur des architectures distribuées. Les détails et les résultats de simulation des méthodes développées et implémentées, sur la nouvelle plateforme, y sont alors présentés. Parmi ces méthodes, nous retrouvons notamment :

- des méthodes de fusion de données des capteurs, à base du KF et de ses dérivées ;
- une nouvelle méthode de la détection des lignes de marquage au sol sur une architecture hiérarchique / décentralisée ;
- des méthodes de localisation collaborative basées sur le filtre d'information, sur la méthode de fusion de matrice d'information et sur la méthode de l'intersection des covariances sur des architectures distribuées ;
- une nouvelle méthode de planification de la trajectoire en temps réel à base de la méthode des champs de potentiels artificiels sur une architecture hiérarchique.

Les solutions proposées aux problèmes, présentées dans ce travail, nous ont permis d'apporter cinq contributions principales.

La première contribution correspond au développement d'une plateforme de conception, de simulation et de validation des ADAS avec les plateformes d'OPAL-RT et le simulateur Pro-SiVIC. À cet effet, une première version de preuve de concept est réalisée. L'avantage de cette nouvelle plateforme réside dans :

- Son aspect évolutif (mise à échelle) qui lui permet de s'adapter aux besoins de simulation des ADAS et des véhicules intelligents et connectés évoluant dans des scénarios multivéhiculaires. Cette plateforme permet de s'interfacer avec plusieurs simulateurs : Pro-SiVIC, PreScan, CarMaker et EXata ;
- Sa puissance de calcul parallèle, également évolutive et qui ne dépend essentiellement que du nombre des machines d'OPAL-RT utilisées.

La deuxième contribution correspond à l'implémentation et à la simulation d'une nouvelle méthode de détection des lignes de marquage au sol, adaptée aux calculs parallèles à des fins d'accélération du traitement d'image pour des applications en temps réel. Les atouts de la méthode résident dans sa capacité à réduire la taille des zones d'intérêt associées aux lignes sur la route, à distribuer et à répartir les besoins de traitement de ces zones sur des cœurs différents, et à suivre les paramètres des lignes même en absence de mesures pendant un certain laps de temps.

La troisième contribution correspond à l'implémentation et à la simulation des algorithmes de fusion de données des capteurs proprioceptifs dans le cadre de localisation, et extéroceptifs dans le cadre de la numérisation de l'environnement du véhicule pour la localisation collaborative.

La quatrième contribution correspond à l'implémentation et à la simulation d'une nouvelle approche de calcul des deux matrices d'information et de covariances, ceci pour l'estimation des deux premiers moments du vecteur d'état du filtre d'information. La méthode transforme la complexité du calcul du problème initial de  $O(n^3)$  vers  $O(q^3)$ , où  $n$  et  $q$  sont respectivement les tailles des vecteurs d'état et d'observation. Cette méthode permet ainsi d'obtenir les deux estimations, dans les deux représentations d'état et d'information, qui sont exploitées par des algorithmes de fusion de données de haut niveau sur des architectures de fusion hiérarchiques et distribuées. D'autre part, le problème de corrélation des estimées des nœuds est résolu à l'aide de la méthode de l'intersection des covariances et de la méthode de la matrice d'information.

La cinquième et dernière contribution correspond à l'implémentation et à la simulation d'une nouvelle stratégie éliminant le problème des minimums locaux de la méthode des champs de potentiels artificiels rencontrés par l'algorithme, puis à son adaptation aux scénarios véhiculaires.

## 10.2 Perspectives

Bien que la présente plateforme soit très flexible et adaptable aux scénarios de co-simulations en matière de puissance de calcul, il serait intéressant d'exporter complètement les simulateurs de la dynamique de véhicule, de capteurs embarqués et de communication véhiculaire sur des plateformes d'OPAL-RT. Ainsi, la future plateforme permettra de simuler et de valider, en mode

accéléré, les systèmes de la conduite automatisée d'un niveau d'autonomie supérieur à trois pour lesquels les prérequis sont de rouler sur plusieurs dizaines de milliards de kilomètres.

L'intégration de processeurs graphique GPU « *en* : Graphics Processing Unit » aux plateformes d'OPAL-RT serait un atout, voire une nécessité, à la fois pour le traitement d'image et pour l'entraînement et l'implémentation de l'intelligence artificielle embarquée à base des réseaux de neurones.

Comme toute nouvelle plateforme de simulation des systèmes complexes, il est nécessaire de maintenir la plateforme à jour en la dotant de nouvelles technologies pour répondre aux demandes et aux exigences de la conduite automatisée. Pour ce faire, il est recommandé d'envisager les réalisations suivantes :

- Développer et tester les scénarios multivéhiculaires communicants à travers EXata. Cette dernière a pu être intégrée à la plateforme sans être testée, à cause notamment de contraintes liées aux problèmes de licence ;
- Discuter avec les divers fournisseurs de plateformes de simulation de la dynamique des véhicules et des capteurs embarqués sur la possibilité d'intégrer leurs simulateurs sur les plateformes d'OPAL-RT.

Cependant, il reste également une partie importante portant sur la validation de la nouvelle plateforme et dont les procédures de validation restent à définir. Ainsi, après la validation de la plateforme, cette dernière permettra à son tour de valider les systèmes d'aide à la conduite et des véhicules intelligents et connectés en mode simulation. De plus, il est très important de mener une étude comparative plus poussée sur les différentes plateformes semblables à cette nouvelle plateforme et celles récemment développées, en particulier VEOS de dSpace, VeriStand de NI, SimWB de Concurrent-RT, Pro-SiVIC/RTMaps, etc. En revanche, d'autres acteurs de la simulation et du calcul parallèle se sont déjà orientés vers le calcul sur le *cloud* (informatique en nuage). En effet, on retrouve notamment Cognata [151] utilisant la technologie GPU de NVIDIA et les solutions cloud de « Microsoft Azure », ceci pour l'implémentation de l'intelligence artificielle embarquée et la génération de milliards de kilomètres de scénarios de test, en temps accéléré, à des fins de test et de validation des systèmes simulés [152]. Ainsi, il

serait intéressant d'investiguer la possibilité et les capacités d'adaptation de la plateforme, proposée dans ce projet de recherche, à la simulation accélérée sur le cloud.

D'autres limitations ont été constatées lors de la simulation des méthodes proposées, auxquelles il faut remédier. Parmi ces limitations, nous retrouvons notamment :

- Des difficultés d'automatisation du processus de sélection des bonnes lignes de marquage au sol à suivre par le véhicule parmi toutes les lignes que la transformée de Hough identifie dans l'étape d'initialisation. Si les valeurs initiales des différents paramètres de la méthode proposée ne correspondent pas aux bonnes lignes, la méthode sera alors induite en erreur. Par la suite, la méthode détectera et suivra d'autres lignes que celles des bordures de la route ;
- La submersion des minimums locaux pourrait obstruer le passage du véhicule dans certaines conditions : chemins étroits, virages serrés, demi-tours, etc. l'empêchant ainsi d'atteindre sa destination ;
- L'idéalisat ion du canal de transmission risque de réduire la performance des méthodes de fusion proposées sur des architectures distribuées. Par conséquent, il faut les adapter à la fois au retard de la communication inter-véhiculaire, et à celui de la co-simulation de la nouvelle plateforme Pro-SiVIC/RT-LAB.

Il serait également intéressant de développer et d'implémenter de nouvelles stratégies de contrôle collaboratif en se basant sur les résultats des méthodes de fusion de données développées et implémentées dans le cadre de ce projet.

# Références bibliographique

- [1] Gary Silberg and Richard Wallace "Self Driving Cars Next Revolution", KPMG and Car group report: [https://faculty.washington.edu/jbs/itans/self\\_driving\\_cars%5B1%5D.pdf](https://faculty.washington.edu/jbs/itans/self_driving_cars%5B1%5D.pdf)
- [2] US department of transportation "Federal Automated Vehicles Policy: Accelerating the Next Revolution In Roadway Safety" September 2016.
- [3] Web site: <https://www.csc.noaa.gov/digitalcoast>
- [4] Web site: <http://earthexplorer.usgs.gov/>
- [5] Alberto Broggi, Alexander Zelinsky, Michel Parent, Charles E. Thorpe, "Handbook of robotics. Chapter 51: Intelligent vehicles." Springer 2008.
- [6] Maaruf Ali, "Standards of Communications in the Intelligent Transport systems (ITS)" from: Nicu Bizon Lucian Dascalescu and Naser Mahdavi Tabatabaei "Autonomous vehicles intelligent transport systems and smart technologies." Nova Science Publishers, Inc. New York, 2014.
- [7] Richard A. Poisel "Target Acquisition in Communication Electronic Warfare Systems." Artech House, 2004.PP: 01-06. <http://books.google.ca/books?id=zooyUi9nA-cC4>
- [8] Pendleton, S.D.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.H.; Rus, D.; Ang, M.H. "Perception, Planning, Control, and Coordination for Autonomous Vehicles." Machines, 2017.
- [9] Denis Gingras, "An overview of positioning and data fusion techniques applied to land vehicle navigation systems." Book Chapter, appeared in Automotive Informatics and Communicative Systems: Principals in Vehicular Networks and Data Exchange, H. Guo Ed., IGI Edition, April 2009.
- [10] Remusan, P., Pollart H. "Intelligent Cars.", OPAL-RT's 9<sup>th</sup> International Conference on Real-Time Simulation, Montreal, Canada, September 5<sup>th</sup>-8<sup>th</sup>, 2017. [https://www.opal-rt.com/wp-content/themes/enfold-opal/pdf/L00161\\_0603.pdf](https://www.opal-rt.com/wp-content/themes/enfold-opal/pdf/L00161_0603.pdf)
- [11] Frédérique Vallée, Laurent Raffaelli, Guy Fayolle : « Fiabilité et sécurité fonctionnelle des ADAS » 20<sup>e</sup> Congrès de maîtrise des risques et de sûreté de fonctionnement - Saint-Malo 11-13 octobre 2016. <https://hal.inria.fr/hal-01398428/document>

- [12] Christian Piguet : « *La loi de Moore : quand s'arrêtera-t-elle ?* » Flash Informatique N° 10 : Spécial calcul à haute performance. 21 décembre 2010.
- [13] Barbara Wendling “*SAE J3016 - Taxonomy and Definitions for Terms Related To Driving Automation Systems for On-Road Motor Vehicles.*” Society of Automotive Engineers (SAE) Webinar, November 8th, 2017.
- [14] Audi Service “*ESP Electronic Stability Programme Design and function.*” Volkswagen AG, Wolfsburg. [http://www.volkspage.net/technik/ssp/ssp/SSP\\_204.pdf](http://www.volkspage.net/technik/ssp/ssp/SSP_204.pdf)
- [15] Shoutao Li, Di Zhao, Luyu Zhang, Yantao Tian “*Lateral Stability Control System Based on Cooperative Torque Distribution for a Four In-Wheel Motor Drive Electric Vehicle.*” Proceedings of the 36<sup>th</sup> Chinese Control Conference, Dalian, China, July 26-28, 2017.
- [16] Ishida, S., Gayko, J.E., "Development, evaluation and introduction of a lane keeping assistance system." Proc. IEEE Intelligent Vehicles Symposium, Parma, Italy, June, 14<sup>th</sup>-17<sup>th</sup>, 2004.
- [17] Kareem Habib “*Automatic vehicle control systems, The Automatic Emergency Braking (AEB) or Autopilot systems may not function as designed, increasing the risk of a crash.*” US department of transportation, national highway traffic safety administration, 2016.
- [18] Guoqing Xu, Li Liu, Yongsheng Ou, and Zhangjun Song "Dynamic Modeling of Driver Control Strategy of Lane-Change Behavior and Trajectory Planning for Collision Prediction.", IEEE transactions on intelligent transportation systems, vol. 13, no. 3, september 2012.
- [19] Alex Davies « *NURO's Self-Driving R-1 Doesn't Drive You—It Drives Your Stuff*» website: <https://www.wired.com/story/nura-ai-startup-self-driving/>
- [20] Web site : <http://navya.tech/navya-presentation/>
- [21] Web Site: <http://www.20minutes.fr/high-tech/1698779-20150930-navya-arme-vehicule-autonome-francais-veut-mesurer-google-car>
- [22] Web Site: <https://navya.tech/en/navya-at-its-world-congress-2015-a-world-premiere-for-the-navya-arma/>
- [23] Brochure technique « *MAVYA Mobility Solution Offer*» [https://navya.tech/wp-content/uploads/2017/09/Navya\\_Brochure\\_Print\\_FR\\_Website.pdf](https://navya.tech/wp-content/uploads/2017/09/Navya_Brochure_Print_FR_Website.pdf)
- [24] Web Site: <http://www.volvotrucks.com/en-en/about-us/automation.html>
- [25] Web Site: <http://www.volvologroup.com/en-en/news/2017/may/news-2561936.html>

- [26] Web Site: <https://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving/drive-me>
- [27] Elisabeth Uhlemann “*ITS Frequency Bands Are Being Debated.*” Connected Vehicles IEEE Vehicular Technology Magazine. December 2016.
- [28] Greenblatt, N.A. “*Self-driving Cars and the Law.*” IEEE Spectrum, Feb.16th, 2016, p.42.
- [29] Brian Markwalter “*The Path to Driverless Cars*” IEEE Consumer Electronics Magazine, Volume: 6, Issue: 2, April 2017.
- [30] Web Site: <https://www.theverge.com/2017/9/27/16373574/ford-lyft-self-driving-car-partnership-gm>
- [31] Web Site: [https://www.thestar.com/news/city\\_hall/2017/11/13/lyft-ubers-ride-hailing-rival-is-expanding-to-the-gta.html](https://www.thestar.com/news/city_hall/2017/11/13/lyft-ubers-ride-hailing-rival-is-expanding-to-the-gta.html)
- [32] Barry Brown “*The Social Life of Autonomous Cars.*” IEEE Computer Society, Computer, Volume: 50, Issue: 2, Feb. 2017.
- [33] Erico Guizzo “*How Google's Self-Driving Car Works*” IEEE Spectrum, automation, October 18<sup>th</sup>, 2011. <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>
- [34] Web Site: <https://waymo.com>
- [35] Andrzej Adamczyk, Grzegorz Taberski, Michal Tanas, and Witold Holubowicz, “*The TALOS Project. EU wide robotic border guard system.*”, 2011 16th International Conference on Methods and Models in Automation and Robotics (MMAR)22-25 Aug. 2011, pp: 336 – 340
- [36] Web Site: <https://www.valeo.com/fr/voiture-autonome-sur-route/>
- [37] D. Gruyer, S. Choi, C. Boussard, and B. d’Andréa-Novel, “*From Virtual To Reality, How to Prototype, Test and Evaluate ADAS: Application to Automatic Car Parking.*” 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, Michigan, USA, June 8-11, 2014.
- [38] Kevin S. Swanson, Alexander A. Brown, Sean N. Brennan, Cynthia M. LaJambe, “*Extending Driving Simulator Capabilities Toward Hardware-in-the-Loop Testbeds and Remote Vehicle Interfaces.*”, 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, June 23-26, 2013.

- [39] ESI-Group “*Pro-SiVIC™ 2016.0 User's Guide*” virtual systems & controls, ESI-Group, France, 2016.
- [40] Web site <https://www.esi-group.com/fr/entreprise/investisseurs/actualite/esi-group-fait-lacquisition-de-CIVITEC>
- [41] Technical Support “*PreScan 7.51 Manual*” TASS International, 2016.
- [42] Web site: <https://ipg-automotive.com/products-services/simulation-software/carmaker/>
- [43] Web site : <https://carsim.com/company/index.php>
- [44] Web site : <https://carsim.com/products/carsim/packages.php>
- [45] Web site : <http://www.oktal.fr/en/>
- [46] Web site : <http://www.oktal.fr/en/automotive/range-of-simulators/software>
- [47] SCANeR Studio user manual “*SCANeR Studio version 1.7 analysis mode*” OKTAL, Simulation In Motion, Oct. 2nd, 2017.
- [48] EXata 5.4 User's Guide , Scalable Network Technologies, Inc. August 2015
- [49] Web site : <http://www.sumo.dlr.de/userdoc/Downloads.html>
- [50] Pablo Urmenatea “*Simulation and Improvement of the Handover process in IEEE 802.11p based VANETs (Vehicle Ad-hoc NETworks)*” College of Electronics and Information Engineering, Tongji University, Shanghai, November 2010.
- [51] Web site : <http://web.scalable-networks.com/qualnet-network-simulator-software>
- [52] NS-3 official web site : <https://www.nsnam.org/docs/tutorial/html/introduction.html>
- [53] NS-3 « *Tutorial Release ns-3.28* » NS-3 project, March 21<sup>st</sup>, 2018.
- [54] Web site: [https://www.dspace.com/fr/fra/home/company/company\\_profile.cfm](https://www.dspace.com/fr/fra/home/company/company_profile.cfm)
- [55] dSPACE GmbH “*Automotive Simulation Models*” dSPACE ASM Product information, 2017. Web site:  
[https://www.dspace.com/fr/fra/home/products/sw/automotive\\_simulation\\_mode\\_ls.cfm](https://www.dspace.com/fr/fra/home/products/sw/automotive_simulation_mode_ls.cfm)
- [56] dSpace brochure “*Universal Real-Time Platform : SCALEXIO now also for rapid control prototyping.*”  
[https://www.dspace.com/shared/data/pdf/2017/08\\_nowalsoforrapidcontrolprototyping\\_en.pdf](https://www.dspace.com/shared/data/pdf/2017/08_nowalsoforrapidcontrolprototyping_en.pdf)
- [57] dSpace brochure : “*Virtual Validation with dSPACE Benefits the whole ECU development process*” Embedded Success, 2017.

[https://www.dspace.com/shared/data/pdf/2017/dSPACE\\_Virtual-Validation\\_Business-Field-Brochure\\_01-2017\\_English.pdf](https://www.dspace.com/shared/data/pdf/2017/dSPACE_Virtual-Validation_Business-Field-Brochure_01-2017_English.pdf)

- [58] Web site “*Virtual validation*”  
[https://www.dspace.com/en/inc/home/products/systems/virtual\\_validation/viva\\_usecases/usecase\\_cluster-simulation.cfm](https://www.dspace.com/en/inc/home/products/systems/virtual_validation/viva_usecases/usecase_cluster-simulation.cfm)
- [59] Web site (bus) <http://www.ni.com/white-paper/11540/en/>
- [60] Web site (APIs) <http://www.ni.com/white-paper/31488/en/>
- [61] Web site : <https://www.concurrent-rt.com/products/simulation-workbench/>
- [62] Shou-pon Lin, Yitian Gu and Nicholas F. Maxemchuk, "A Multiple Stack Architecture for Intelligent Vehicles." 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, Michigan, US, June 8-11, 2014.
- [63] Bouvet, D. “Guaranteed 3-D mobile robot localization using an odometer, an automatic theodolite and indistinguishable landmarks.” Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001.
- [64] Titterton, D. and J. Weston. 2004. "Strap down Inertial Navigation Technology.", s.l. AIAA: 2<sup>nd</sup> edition. Washington, DC : 2004
- [65] Parkinson, B. W., et al. “Global Positioning System: Theory and Applications.” Washington, DC: American Institute of Aeronautics and Astronautics: s.n., 1996.
- [66] I. Kaplan, Elliott D. II. Hegarty, C. (Christopher J.). “Understanding GPS principles and applications.” Second Edition, 2005, ISBN 1-58053-894-0.
- [67] Francesc Rocadenbosch "Course: Introduction to LIDAR (laser, radar) Remote Sensing Systems.", Remote Sensing Laboratory Universitat Politècnica de Catalunya, 2007.
- [68] National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. “Lidar 101: An Introduction to Lidar Technology, Data, and Applications.” Revised. Charleston, SC: NOAA Coastal Services Center. 2012.
- [69] A. Broggi, S. Cattani, P. P. Porta, and P. Zani, “A Laserscanner-Vision Fusion System Implemented on the TerraMax Autonomous Vehicle.” Procs. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Beijing, China, 2006, pp. 111–116.
- [70] Panagiotis Lytrivis, George Thomaidis, Angelos AmditisNada Milisavljević "Sensor and Data Fusion, chapter 7: Sensor Data Fusion in Automotive Applications." Published by In-Teh, Croatia, February 2009, InTech, DOI: 10.5772/6574.

- [71] David L. Hall and James Llinas, “*Handbook of Multisensor Data Fusion Theory and Practice. Chapter 1: Multisensor Data Fusion.*”, CRC Press, Second Edition, Taylor& Francis Group, New York, 2009.
- [72] Rudolf E. Kalman, “*A New Approach to Linear Filtering and Prediction Problems*” Transactions of the ASME, Journal of Basic Engineering, vol. 82,.1960
- [73] Kazuo Tanaka, Hua O. Wang: “*Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*” Copyright 2001 John Wiley & Sons, Inc. ISBNs: 0-471-32324-1 Hardback; 0-471-22459-6 Electronic.
- [74] Ramsey Faragher; "Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation." IEEE Signal Processing Magazine; September 2012.
- [75] Anderson, B.D.O., Moore, J.B.: “*Optimal Filtering*” Prentice Hall, New York, 1979.
- [76] Phil Kim “*Kalman Filter for Beginners with Matlab Examples.*” Korea: A-JIN, 2010.
- [77] Gerasimos G. Rigatos; "Derivative-free Kalman Filtering for sensorless control of MIMO nonlinear dynamical systems." IEEE International Conference on Mechatronics and Automation; Chengdu, China, August 5<sup>th</sup>-8<sup>th</sup>, 2012.
- [78] Mathieu St-Pierre and Denis Gingras; "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system." IEEE Intelligent Vehicles Symposium; Parma, Italy. June, 2004.
- [79] S. J. Julier and J. K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems." Proc. of AeroSense: The 11<sup>th</sup> Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls., 1997.
- [80] Eric A. Wan and Rudolph van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation." Oregon Graduate Institute of Science & Technology 20000 NW Walker Rd, Beaverton, Oregon 97006.
- [81] Eric A. Wan and Rudolph van der Merwe, "The Unscented Kalman Filter." Department of Electrical and Computer Engineering, Oregon Graduate Institute of Science and Technology, Beaverton, Oregon, U.S.A.
- [82] R. van der Merwe, J.F.G. de Freitas, D. Doucet, and E.A. Wan, “*The unscented particle filter.*” Technical Report CUED/F-INFENG=TR 380, Cambridge University Engineering Department, August 2000.

- [83] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking.", IEEE transactions on signal processing, vol. 50, no. 2, February 2002.
- [84] A. Doucet, N. de Freitas, and N. Gordon, "Sequential Monte Carlo Methods in Practice: An Introduction to sequential Monte Carlo methods," Springer, 2000.
- [85] A. Pouliakis, Martin E. Liggins David L. Hall, James Llinas, "Handbook of Multi-sensor Data Fusion, Theory and Practice. Chapter 17: Distributed Fusion Architectures, Algorithms, and Performance within a Network-Centric Architecture" Second Edition, CRC Press Taylor & Francis Group, Boca Raton, London, New York, 2009. PP: 411-435.
- [86] Sae Schatz, Denise Nicholson, and Rhianon Dolletski, "System approach to Simulation for Training, Instruction, Technology and Process Engineering", From Katalin Popovici, and Pieter J. Mosterman, "Real-Time Simulation Technologies: Principles, Methodologies, and Applications.", Published by CRC Press, August 17<sup>th</sup>, 2012.
- [87] Lawrence A. Klein, "Sensor and Data Fusion: A Tool for Information Assessment and Decision Making. Chapter 3: Data Fusion Algorithms and Architectures.", A Publication of SPIE—The International Society for Optical Engineering Bellingham, Washington USA, 2004, PP: 51-100.
- [88] Hong Cheng; "Autonomous Intelligent Vehicles, Theory, Algorithms, and Implementation"; Springer Advances in Computer Vision and Pattern Recognition; 2011.
- [89] Zhefu Shi, Cory Beard, and Ken Mitchell; "Analytical models for understanding space, backoff, and flow correlation in CSMA wireless networks." Springer Science Business Media, LLC, published online: 3 July 2012.
- [90] Zhefu Shi, Cory Beard, and Ken Mitchell; "Competition, Cooperation, and Optimization in Multi-Hop CSMA Networks with Correlated Traffic." International Journal of Next-Generation Computing, Vol. 3, No. 3, November 2012.
- [91] Jonathan Baber, Julian Kolodko, Tony Noël, Michael Parent; "Cooperative Autonomous Driving, Intelligent Vehicles Sharing City Roads Cooperative Autonomous Driving." IEEE Robotics & Automation Magazine, March 2005.

- [92] John Chiasson, Leon Tolbert, Keith McKenzie and Zhong; "*Real-Time Computer Control of a Multilevel Converter using the Mathematical Theory of Resultants.*" Proceedings of the Electrimacs Conference, Montreal, Canada, August 18<sup>th</sup>- 21<sup>st</sup>, 2002.
- [93] Sergio Cavalcanti Marco Papini, "*Preliminary Model Matching of the EMBRAER 170 Jet*" Journal of Aircraft, Vol. 41, No. 4 (2004), pp. 703-710.
- [94] RT-LAB User Guide Version 10.7 (RT-LAB -UG-107-00)
- [95] OP5600 HILBOX, USER GUIDE, Real-Time Simulator
- [96] D. Gruyer, S. Glaser, B. Monnier, "*Simulation of vehicle automatic speed control by transponder equipped infrastructure*", in proc. IEEE ITST2009, Lille (France), October 2009.
- [97] D. Gruyer, S. Glaser, B. Monnier, "*SiVIC, a virtual platform for ADAS and PADAS prototyping, test and evaluation*" Proc. FISITA, Budapest, Hungary, May 4<sup>th</sup>, June 2010.
- [98] Dominique Gruyer, Nicolas Hiblot, Philippe De Souza, Hervé Sauer, Bertrand Monnier "*A new generic virtual platform for cameras modeling.*" Proc. of International Conference VISION 2010, Montigny le Bretonneux (France), October 6<sup>th</sup>-7<sup>th</sup>, 2010.
- [99] B. Steux, « *RTMaps, un environnement logiciel dédié à la conception d'applications embarquées temps-réel. Utilisation pour la détection automatique de véhicules par fusion radar/vision* » PhD. thesis from Ecole des Mines de Paris, December 2001.
- [100] C.S.Ravichandran; S. Subha Rani and; T. Manikandan. "*Designing of PID Controller for Discrete Time Linear System Using Balanced Approach Reduced Order Model.*" American Journal of Applied Sciences, 2007.
- [101] Y.P. Chang, et al, "*Digital Modeling and PID Controller Design for MIMO Analog Systems with Multiple Delays in States, Inputs and Outputs.*" Circuits System Signal Process 2009.
- [102] Warsame H. Ali, Yongpeng Zhang, Penrose Cofie "*Digital controller design for analog MIMO systems with multiple I/O delays.*" IEEE Conference Publications, Taiyuan, China, May 23<sup>rd</sup>-25<sup>th</sup>, 2012.

- [103] Jennifer M. Madsen, Leang-San Shieh, and Shu-Mei Guo."*State-Pace Digital PID controller design for multivariable analog systems with multiple time delay.*" Asian Journal of Control, Vol. 8, No. 2, June 2006.
- [104] Han-Qin Zhou ; Leang-san Shieh ; Qing-Guo Wang ; Liu, C.R." *State-space digital PID controller design for linear stochastic multivariable systems with input delay*" International Conference on Control and Automation. ICCA, Budapest, Hungary, November 14<sup>th</sup>, 2005.
- [105] Hassania, Baibanou & Sicard, P & Abdelfattah, Ba-Razzouk "*Solutions to Typical Motor Load Emulation Control Problems*" Electrimacs, August 18<sup>th</sup>, 20<sup>th</sup>, 2002.
- [106] Assidiq, A.A.M.; Khalifa, O.O.; Islam, R.; Khan, S. "*Real-time lane detection for autonomous vehicles.*" International Conference on Computer and Communication Engineering. ICCCE. Kuala Lumpur, Malaysia, May13–15, 2008.
- [107] Kristijan Maček, Brian Williams, Sascha Kolski, "*A lane Detection vision module for driver assistance*" IEEE/APS International Conference on Mechatronics & Robotics, Aachen, Germany, September 13-15, 2004
- [108] Xiaodong Miao, Shunming Li, Huan Shen, "*On-Board Lane Detection System For Intelligent Vehicle Based On Monocular Vision.*" International journal on smart sensing and intelligent systems, vol. 5, no. 4, December 2012
- [109] Teoh, E.K.; Dinggang Shen. "*Lane detection using B-snake*" International Conference on Information Intelligence and Systems. Proceedings. 1999.
- [110] Nunez, P.; Drews Jr, P.; Rocha, R. and Dias, J. "*Data fusion calibration for a 3D-laser range finder and a camera using inertial data*" Proc. 4<sup>th</sup> European Conf. on Mobile Robots (ECMR), Sep. 2009.
- [111] Aliakbarpour, H.; Nez, P. ; Prado, J.; Khoshhal, K. and Dias, J. "*An efficient algorithm for extrinsic calibration between a 3D-laser range finder and a stereo camera for surveillance.*" 14<sup>th</sup> International Conference on Advanced Robotics (ICAR), Munich, Germany, June 22<sup>nd</sup>-26<sup>th</sup>, 2009.

- [112] Bounini, F.; Gingras, D.; Lapointe, V.; Gruyer, D. “*Real-time simulator of collaborative autonomous vehicles.*” International Conference on Advances in Computing, Communications and Informatics ICACCI, Greater Noida, India, Sept. 24<sup>th</sup>-27<sup>th</sup>, 2014.
- [113] Bounini, F.; Gingras, D.; Lapointe, V.; Gruyer, D. “*Poster: Real-time simulator of collaborative autonomous vehicles.*” International Conference on Advances in Computing, Communications and Informatics ICACCI, Greater Noida, India, Sept 24<sup>th</sup>-27<sup>th</sup>, 2014.
- [114] Cappelletti, M.; Macii, D. “*A RANSAC-based fast road lanes detection algorithm for high-speed wheeled vehicles.*” Instrumentation and Measurement Technology Conference I2MTC, Binjiang, China, May 10<sup>th</sup>-12<sup>th</sup>, 2011.
- [115] Ghassan H. Karin A. and Rafeef G. “*Project Report for the Computer Vision Course Automatic Line Detection.*” Image Analysis Group Department of Signals and Systems Chalmers University of Technology, September 1999.
- [116] John, B. M. and Donald, N. “*Application of the Hough Transform to Lane detection and Following on High Speed Roads*” signal & system Group, Department of Computer Science, National University of Ireland.
- [117] web site <http://www.CIVITEC.com/>
- [118] web site <http://www.livic.ifsttar.fr/>
- [119] Miguel A. Padilla Castaneda, Jesus Savage, Adalberto Hernandez and Fernando Arambula Cosío. “*Local Autonomous Robot Navigation Using Potential Fields, Motion Planning.*” Chapter from the book “*Motion Planning*” and edited by Xing-Jian Jing, ISBN 978-953-7619-01-5, 598 pages, Publisher: In Tech, Chapters published June 01, 2008 under CC BY-NC-SA 3.0 license. <http://cdn.intechopen.com/pdfs-wm/5350.pdf>
- [120] Xing-Jian Jing. “*Mobile Robots Motion Planning New Challenges.*” University of Sheffield United Kingdom, 2008.
- [121] Ming Huang , Rui Zhang , Yulin Ma , QingXin Yan . “*Research on Autonomous Driving Control Method of Intelligent Vehicle Based on Vision Navigation.*” International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, 2010.

- [122] Feng Luo, Chu Liu, Zechang Sun “*Intelligent Vehicle Simulation and Debugging Environment Based on Physics Engine.*” International Asia Conference on Informatics in Control, Automation and Robotics, Bangkok, Thailand, 2009.
- [123] João Borges Sousa; Fernando Lobo Pereira: “*Cooperative Autonomous Underwater Vehicle localization*”, IEEE OCEANS, Sydney, Australia, May 24<sup>th</sup> -27<sup>th</sup>, 2010.
- [124] Rohani M, Gingras D, Gruyer D, Vigneron V, “*A New Decentralized Bayesian Approach for Cooperative Vehicle Localization based on fusion of GPS and VANET based Intervehicle Distance Measurement*” Intelligent Transportation Systems Magazine, Summer 2015.
- [125] Fabio Morbidi, and Gian Luca Mariottini: “*Active Target Tracking and Cooperative Localization for Teams of Aerial Vehicles*”, IEEE transactions on control systems technology, vol. 21, no. 5, September 2013.
- [126] D. Gruyer, B. Andréa-Novel, S. Demmel, G. S. Larue and A. Rakotonirainy, “*Simulating Cooperative Systems Applications: a New Complete Architecture*” International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 4, Nov. 12<sup>th</sup> , 2013.
- [127] Bounini, F; Gingras, D; Pollart, H; Lapointe, V;” Autonomous Vehicle and Real Time Road Lanes Detection and Tracking.” IEEE Vehicular Power and Propulsion Conference (VPPC), Montréal, Canada. Oct 19<sup>th</sup>–22<sup>nd</sup>, 2015,
- [128] Nadir Karam. "Agrégation de données décentralisées pour la localisation multi-véhicules.*Automatic*". Université Blaise Pascal - Clermont-Ferrand II, 2009.
- [129] C. Chong, S. Mori, "Convex combination and covariance intersection algorithms in distributed fusion." ISIF Fusion, Montreal Quebec, Canada, August 2001.
- [130] Simon Julier and Jeffrey K. Uhlmann, "General Decentralized Data Fusion with Covariance Intersection: *Handbook of Multi-sensor Data Fusion, Theory and Practice.*" second Edition, CRC Press Taylor & Francis Group, Boca Raton London New York, 2009.
- [131] Ramsey Faragher, “*Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation*.”, IEEE Signal Processing Magazine, September 2012.

- [132] Ruth Elizabeth Petrie, "*Localization in the ensemble Kalman Filter.*" A dissertation submitted in partial fulfilment of the requirement for the degree of MSc. Atmosphere, Ocean and Climate, at the Departement of Meteorology, University of Reading, Aug, 2008
- [133] Viktor Eriksson: "*Evaluation of Decentralized Information Matrix Fusion for Advanced Driver-Assistance Systems in Heavy-Duty Vehicles.*" Master's Thesis in Optimization and Systems Theory. KTH Royal Institute of Technology School of Engineering Sciences. Stockholm, Sweden, 2016.
- [134] Fabio Morbidi, and Gian Luca Mariottini: "*Active Target Tracking and Cooperative Localization for Teams of Aerial Vehicles*", IEEE transactions on control systems technology, vol. 21, no. 5, September 2013.
- [135] K. C. Chang, T Zhi, R. K. Saha: "*Performance Evaluation of Track Fusion with Information Matrix Filter.*" IEEE Transactions on Aerospace and Electronic Systems VOL. 38, NO. 2, April 2002.
- [136] Qiao Xiangdong, Chang Kuochu, "*Information Matrix Fusion with Feedback Versus Number of Sensors.*" Proc. the 29th Chinese Control Conference, Beijing, Chana, pp. 1836-1843, 2010.
- [137] Rauch A., Klanner F., Rasshofer R. "*Car2X-based perception in a high-level fusion architecture for cooperative perception systems.*" IEEE Intelligent Vehicles Symposium (IV), Alcala de Henares, Spain, July 5th, 2012.
- [138] Rudolph van der Merwe : "*Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models.*" A dissertation submitted to the faculty of the OGI School of Science & Engineering at Oregon Health & Science University, April 2004.
- [139] Sebastian Thrun, Dieter Fox, Wolfram Burgard, "*Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).*" The MIT Press ©2005, ISBN:0262201623
- [140] Wolfgang Nichsen. "*Information Fusion based on Fast Covariance Intersection Filtering.*" In Proceedings of the 5th International Conference on Information Fusion (Fusion 2002). Annapolis, MD, July 2002.
- [141] X. Tian, Y. Bar-Shalom, T. Yuan, E. Blasch, K. Pham, G. Chen. "*A Generalized Information Matrix Fusion Based Heterogeneous Track-to-Track Fusion Algorithm.*"

Proc. SPIE Conf. on Signal Processing, Sensor Fusion, and Target Recognition, April 1<sup>st</sup>, 2011.

- [142] Bounini F, Gingras D, Pollart H, Gruyer D. “*Modified Artificial Potential Field Method for Online Path Planning Applications.*” IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 2017.
- [143] Steaven M. LaValle. “*Planning algorithms.*” Cambridge University Press, 2006.
- [144] Xiang Liu. “*A comparative study of A-star algorithms for search and rescue in perfect maze.*” International Conference on Electric Information and Control Engineering (ICEICE), Wuhan, China, 2011.
- [145] Yizhen Huang, Qingming Yi, Min Shi. “*The improved Dijkstra’s shortest path algorithm.*” 7<sup>th</sup> International Conference on Natural Computation (ICNC), Shanghai, China, 2011.
- [146] Takayuki Goto, Takeshi Kosaka, and Hiroshi Noborio. “*On the heuristics of A\* or A algorithm in ITS and robot path-planning.*” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, USA, 2003.
- [147] O. Khatib. “*Real-time obstacle avoidance for manipulators and mobile robots.*” International Journal of Robotic Research, Vol. 5, No 1. 1986.
- [148] S. S. Ge and Y. J. Cui. “*New Potential Functions for Mobile Robot Path Planning.*” IEEE Transactions on Robotics and Automation, vol. 16, no. 5, October 2000
- [149] Hao-Tien Chiang, Nick Malone, Kendra Lesser, Meeko Oishi, Lydia Tapia. “*Path-Guided Artificial Potential Fields with Stochastic Reachable Sets for Motion Planning in Highly Dynamic Environments.*” International Conference on Robotics and Automation (ICRA), Washington State Convention Center Seattle, USA, 2015.
- [150] OPAL-RT’s 9th International Conference on Real-Time Simulation, Montréal, Canada, Sept. 5<sup>th</sup>-8<sup>th</sup> 2017. <https://www.opal-rt.com/event/rt17/>
- [151] Web site: <http://www.cognata.com/>
- [152] Web site: <https://www.businesswire.com/news/home/20180109005360/en/Cognata-Builds-Cloud-Based-Autonomous-Vehicle-Simulation-Platform>