



HAL
open science

Localisation coopérative de véhicules autonomes communicants

Elwan Héry

► **To cite this version:**

Elwan Héry. Localisation coopérative de véhicules autonomes communicants. Robotique [cs.RO]. Université de Technologie de Compiègne, 2019. Français. NNT : 2019COMP2516 . tel-02883196

HAL Id: tel-02883196

<https://theses.hal.science/tel-02883196>

Submitted on 28 Jun 2020

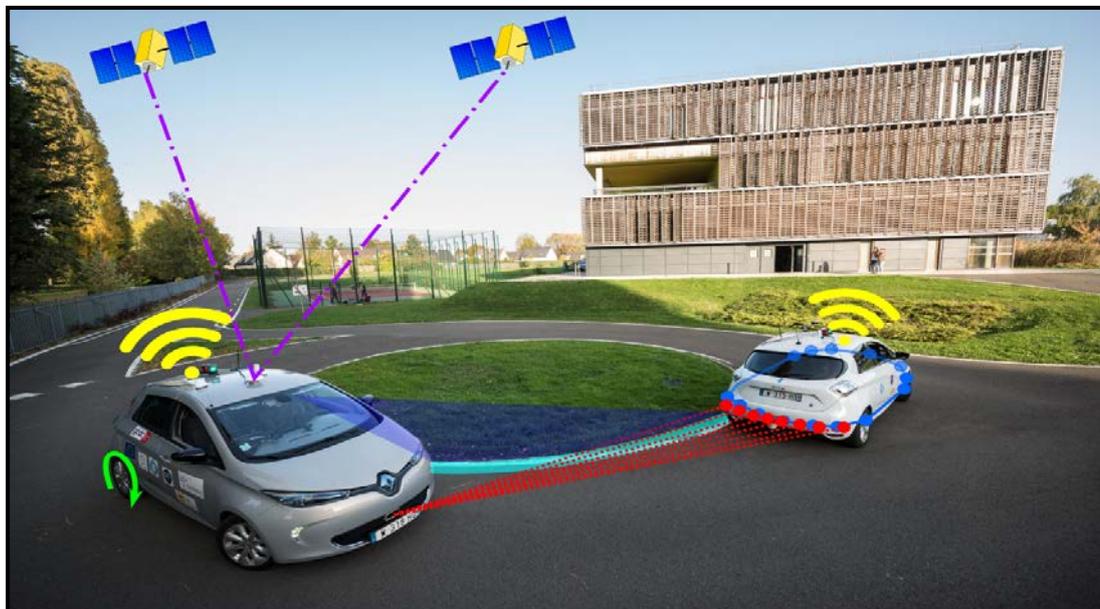
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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

Par Elwan HÉRY

Localisation coopérative de véhicules autonomes communicants

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 12 novembre 2019
Spécialité : Robotique : Unité de recherche Heudyasic (UMR-7253)

D2516

Localisation coopérative de véhicules autonomes communicants

Elwan Héry

Thèse de doctorat de l'université de technologie de Compiègne dans la spécialité robotique

Thèse soutenue publiquement le 12 novembre 2019 devant le jury composé de :

Prof. Dominique Meizel	Président du jury
Prof. Roland Chapuis	Rapporteur
Prof. Maan El Badaoui El Najjar	Rapporteur
Prof. Miguel Ángel Sotelo Vázquez	Examineur
Prof. Véronique Cherfaoui	Examinatrice
Dr. Philippe Xu	Co-directeur de thèse
Prof. Philippe Bonnifait	Co-directeur de thèse



Remerciement

Je souhaite tout d'abord remercier mes directeurs de thèse Philippe Xu et Philippe Bonnifait pour leur soutien et leurs nombreux conseils qui m'ont guidé tout au long de cette thèse.

Je tiens ensuite à remercier les rapporteurs de ma thèse Roland Chapuis et Maan El Badaoui El Najjar pour le temps qu'ils ont consacré à relire le manuscrit de thèse. Je voudrais aussi remercier tous les autres membres du jury Miguel Ángel Sotelo Vázquez, Dominique Meizel et Véronique Cherfaoui pour l'intérêt porté à ma thèse et les questions pertinentes qu'ils ont posées.

Je remercie aussi tout le laboratoire Heudiasyc pour son ambiance chaleureuse.

Mes remerciements vont en particulier aux doctorants, maîtres de conférences, professeurs pour les nombreuses discussions scientifiques qui m'ont beaucoup apporté, notamment les membres de mon bureau Romain Guyard, Hafida Mouhagir, Alia Chebly, Maxime Chaveroche ainsi que les membres du laboratoire commun entre Renault, l'UTC et le CNRS comme Anthony Welte, Joelle Al Hage, Stefano Masi, Corentin Sanchez, Antoine Lima et enfin les doctorants qui m'ont précédé comme Khaoula Lassoued et Franck Li.

Je voudrais aussi remercier les ingénieurs de recherche et tous ceux qui m'ont aidé lors de mes nombreuses expérimentations comme Stéphane Bonnet, Thierry Monglon et Gérard Dherbomez.

Je souhaite finalement remercier ma famille pour leur soutien tout au long de cette thèse.

Résumé

Afin de naviguer en autonomie un véhicule doit être capable de se localiser précisément par rapport aux bords de voie pour ne pas sortir de celle-ci et par rapport aux véhicules et piétons pour ne pas causer d'accident. Cette thèse traite de l'intérêt de la communication dans l'amélioration de la localisation des véhicules autonomes.

La navigation autonome sur route est souvent réalisée à partir de coordonnées cartésiennes. Afin de mieux représenter la pose d'un véhicule relativement à la voie dans laquelle il circule, nous étudions l'utilisation de coordonnées curvilignes le long de chemins enregistrés dans des cartes. Ces coordonnées généralisent l'abscisse curviligne en y ajoutant un écart latéral signé par rapport au centre de la voie et une orientation relative au centre de cette voie en prenant en compte le sens de circulation.

Une première approche de localisation coopérative est réalisée à partir de ces coordonnées. Une fusion de données à une dimension permet de montrer l'intérêt de la localisation coopérative dans le cas simplifié où l'écart latéral, l'orientation curviligne et la pose relative entre deux véhicules sont connus avec précision. Les problèmes de corrélation des erreurs dus à l'échange d'information sont pris en compte grâce à un filtre par intersection de covariance.

Nous présentons ensuite à une méthode de perception de type ICP (Iterative Closest Point) pour déterminer la pose relative entre les véhicules à partir de points LiDAR et d'un modèle polygonal 2D représentant la forme du véhicule.

La propagation des erreurs de poses absolues des véhicules à l'aide de poses relatives estimées avec leurs incertitudes se fait via des équations non linéaires qui peuvent avoir un fort impact sur la consistance.

Les poses des différents véhicules entourant l'égo-véhicule sont estimés dans une carte locale dynamique (CLD) permettant d'enrichir la carte statique haute définition décrivant le centre de la voie et les bords de celle-ci. La carte locale dynamique est composée de l'état de chaque véhicule communicant. Les états sont fusionnés en utilisant un algorithme asynchrone, à partir de données disponibles à des temps variables. L'algorithme est décentralisé, chaque véhicule calculant sa propre CLD et la partageant. Les erreurs de position des récepteurs GNSS étant biaisées, une détection de marquages est introduite pour obtenir la distance latérale par rapport au centre de la voie afin d'estimer ces biais. Des observations LiDAR avec la méthode ICP permettent de plus d'enrichir la fusion avec des contraintes entre les véhicules. Des résultats expérimentaux illustrent les performances de cette approche en terme de précision et de consistance.

Abstract:

To be able to navigate autonomously, a vehicle must be accurately localized relatively to all obstacles, such as roadside for lane keeping and vehicles and pedestrians to avoid causing accidents. This PhD thesis deals with the interest of cooperation to improve the localization of cooperative vehicles that exchange information.

Autonomous navigation on the road is often based on coordinates provided in a Cartesian frame. In order to better represent the pose of a vehicle with respect to the lane in which it travels, we study curvilinear coordinates with respect to a path stored in a map. These coordinates generalize the curvilinear abscissa by adding a signed lateral deviation from the center of the lane and an orientation relative to the center of the lane taking into account the direction of travel. These coordinates are studied with different track models and using different projections to make the map-matching.

A first cooperative localization approach is based on these coordinates. The lateral deviation and the orientation relative to the lane can be known precisely from a perception of the lane borders, but for autonomous driving with other vehicles, it is important to maintain a good longitudinal accuracy. A one-dimensional data fusion method makes it possible to show the interest of the cooperative localization in this simplified case where the lateral deviation, the curvilinear orientation and the relative positioning between two vehicles are accurately known. This case study shows that, in some cases, lateral accuracy can be propagated to other vehicles to improve their longitudinal accuracy. The correlation issues of the errors are taken into account with a covariance intersection filter.

An ICP (Iterative Closest Point) minimization algorithm is then used to determine the relative pose between the vehicles from LiDAR points and a 2D polygonal model representing the shape of the vehicle. Several correspondences of the LiDAR points with the model and different minimization approaches are compared.

The propagation of absolute vehicle pose using relative poses with their uncertainties is done through non-linear equations that can have a strong impact on consistency.

The different dynamic elements surrounding the ego-vehicle are estimated in a Local Dynamic Map (LDM) to enhance the static high definition map describing the center of the lane and its border. In our case, the agents are only communicating vehicles. The LDM is composed of the state of each vehicle. The states are merged using an asynchronous algorithm, fusing available data at variable times. The algorithm is decentralized, each vehicle computing its own LDM and sharing it. As the position errors of the GNSS receivers are biased, a marking detection is introduced to obtain the lateral deviation from the center of the lane in order to estimate these biases. LiDAR observations with the ICP method allow to enrich the fusion with the constraints between the vehicles. Experimental results of this fusion show that the vehicles are more accurately localized with respect to each other while maintaining consistent poses.

Table des matières

1. Introduction générale	11
1.1. Le véhicule intelligent communicant	11
1.2. Retour d'expérience sur le challenge GCDC	12
1.2.1. Présentation des scénarios du GCDC	13
1.2.1.1. Scénario d'insertion	13
1.2.1.2. Scénario d'intersection	13
1.2.1.3. Scénario de priorité au véhicule d'urgence	14
1.2.2. Présentation du véhicule expérimental utilisé	14
1.2.2.1. Architecture du système embarqué	15
1.2.2.2. Localisation et carte pour le contrôle latéral	16
1.2.2.3. Perception et communication pour le contrôle longitudinal	17
1.2.3. Quelques leçons apprises du challenge	20
1.3. Représentation du monde pour la navigation autonome	21
1.3.1. Cartes Haute Définition	21
1.3.2. Cartes Locales Dynamiques	23
1.4. Le problème de la localisation coopérative	24
1.5. Objectifs de la thèse	26
1.6. Plan du manuscrit	27
2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome	29
2.1. Introduction	29
2.2. Projections et coordonnées curvilignes sur différents modèles de route	31
2.2.1. Coordonnées curvilignes dans le cas général	31
2.2.2. Représentation avec des polygones	32
2.2.3. Représentation avec des splines	34
2.2.3.1. B-spline	36
2.2.3.2. Spline d'Hermite	36
2.2.4. Projection sur les polygones selon la méthode des <i>lanelets</i>	37
2.3. Résultats	39
2.3.1. Discontinuité et non-linéarité	39
2.3.1.1. Discontinuité et non linéarité	39
2.3.1.2. Discontinuité de <i>map-matching</i>	41
2.3.2. Conduite en convoi basée sur l'abscisse curviligne	41
2.3.3. Erreurs longitudinales et latérales par rapport à la voie	44
2.4. Conclusion sur les différents modèles de route pour la navigation	45
3. Localisation coopérative à une dimension en abscisse curviligne	47
3.1. Introduction	47

Table des matières

3.2.	Mécanismes de base en fusion de données séquentielles	47
3.2.1.	Prédiction	48
3.2.2.	Mise à jour de Kalman	48
3.2.3.	Mise à jour par intersection de covariance	48
3.3.	Localisation coopérative par fusion de données séquentielles	49
3.3.1.	Problématique	49
3.3.2.	Algorithme de fusion coopérative de données	51
3.3.3.	Corrélations des erreurs lors de la fusion coopérative	52
3.4.	Résultats	53
3.4.1.	Résultats de simulation	54
3.4.2.	Résultats expérimentaux	56
3.5.	Conclusion sur la localisation coopérative à une dimension en abscisse curviligne	59
 4. Pose inter-véhicules à partir de mesures LiDAR		61
4.1.	Introduction	61
4.2.	Problématique	62
4.3.	Regroupement de points LiDAR	65
4.4.	Localisation initiale par boîte englobante	65
4.5.	Localisation par tirage aléatoire avec champ de vraisemblance	68
4.6.	Localisation par optimisation itérative de type ICP	72
4.6.1.	Présentation de la méthode	72
4.6.2.	Correspondances entre les points LiDAR et le modèle polygonal	74
4.6.3.	Minimisation par calcul des racines d'un polynôme	77
4.6.4.	Minimisation par matrice pseudo-inverse	81
4.7.	Résultats de simulation	85
4.7.1.	Séquencement des étapes de l'algorithme	85
4.7.2.	Convergences des différentes méthodes itératives d'ICP	86
4.7.3.	PLICP avec élimination des points sans projection orthogonale	88
4.7.4.	Influence des paramètres et étude de sensibilité	89
4.7.4.1.	Influence du modèle	89
4.7.4.2.	Influence du scénario de conduite	90
4.7.4.3.	Influence de l'interdistance	92
4.7.4.4.	Influence du bruit LiDAR	92
4.7.4.5.	Influence de l'erreur sur la pose initiale	92
4.8.	Robustification de la localisation relative	92
4.8.1.	Élimination des mauvaises correspondances	92
4.8.2.	Élimination des mauvaises convergences	94
4.8.3.	Détection d'un mauvais conditionnement du problème	95
4.9.	Conclusion	95
 5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre		99
5.1.	Introduction	99
5.2.	Problématique	99
5.3.	Résultats expérimentaux	102
5.4.	Conclusion	104

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants	109
6.1. Introduction	109
6.2. Architecture de fusion de données décentralisée et asynchrone	110
6.3. Modèles et processus de la fusion CLD	112
6.3.1. Extrapolation de la CLD aux temps des observations	112
6.3.2. Mises à jour de la CLD	114
6.3.2.1. Mise à jour de Kalman	114
6.3.2.2. Mise à jour par intersection de covariance	114
6.3.3. Initialisation de l'état	116
6.3.4. Modèles d'observation	117
6.3.4.1. Mesures cinématiques	117
6.3.4.2. Poses GNSS	121
6.3.4.3. CLD reçue de la communication	122
6.3.4.4. Détection des bords de voie	123
6.3.4.5. Poses relatives mesurées avec des capteurs extéroceptifs	123
6.4. Résultats expérimentaux	124
6.4.1. Scénario	124
6.4.2. Résultats	125
6.4.3. Estimation du biais GNSS	125
6.4.4. Précision et consistance de la localisation coopérative	128
6.4.5. Performance de la méthode avec des contraintes de perception	129
6.5. Gestion spatio-temporelle des CLD	130
6.6. Conclusion	133
7. Conclusion et perspectives	135
A. Notations	141
B. Changements de repères et incertitudes associées	143
B.1. Introduction	143
B.2. Composition et opérateur \oplus	143
B.3. Inversion et opérateur \ominus	145
B.4. Combinaison des opérateurs	147
B.5. Opérateur \oplus appliqué à une position	148
B.6. Tableau résumé synthétique	149
B.7. Problèmes de non-linéarité lors de la propagation d'incertitude	149
C. Système de coordonnées géodésiques, repères ECEF et ENU	153
D. Intersection de covariance avec la forme de Joseph	157

1. Introduction générale

1.1. Le véhicule intelligent communicant

La localisation d'un véhicule dans son environnement est essentielle pour une navigation autonome. Il doit d'abord être capable de se localiser de façon absolue pour se diriger vers sa destination. Pour les véhicules intelligents, c'est-à-dire des véhicules évoluant sur des routes ouvertes à la circulation publique, la localisation utilise majoritairement un récepteur GNSS (Global Navigation Satellite System). Ceux-ci sont déjà présents dans quasiment tous les véhicules récents. Néanmoins, les récepteurs GNSS et les cartes de navigation actuels permettent seulement de planifier une trajectoire et de la suivre au niveau macroscopique de la route. Ce premier niveau de localisation doit être complété par une localisation relative basée sur une perception de l'environnement. Le véhicule autonome doit en effet être capable de détecter les bords de la route pour connaître l'espace navigable et éviter de sortir de sa voie. Il doit aussi pouvoir détecter l'ensemble des obstacles potentiels et agir en conséquence pour éviter des collisions. Cette perception permet alors de localiser relativement les obstacles dynamiques comme les autres véhicules, les piétons, les cyclistes ou les animaux, mais aussi des éléments statiques comme les bords de la voie. Une redondance de ces détectations avec de multiples capteurs (LiDAR, caméra, radar, etc.) est essentielle pour une navigation autonome sûre. La perception n'est toutefois pas la seule source d'information pour déterminer l'emplacement de ces éléments statiques environnants. Une carte haute définition (HD) permet, en effet, de faire le lien entre la localisation absolue du véhicule et son environnement, le centre et les bords de la voie apparaissant dans cette carte. Elle permet aussi d'avoir de multiples autres informations comme la position des signalisations, les limites de vitesse ou même des bâtiments pouvant être pris comme point de repères. Ainsi, une localisation absolue couplée au centre de la voie décrite dans la carte HD permet au véhicule de suivre une trajectoire et de rester dans sa voie. Cette même localisation permet de s'arrêter à un stop ou devant un passage piéton lorsque ceux-ci sont présents dans la carte. Enfin, une perception des repères présents dans cette même carte permet d'améliorer cette localisation absolue. Les utilisations simultanées de moyens de localisation absolue et de perception conjointement avec une carte HD sont ainsi étroitement liées, deux de ces informations permettent d'obtenir la troisième. On obtient ainsi une redondance d'information augmentant la fiabilité de la navigation.

Un véhicule intelligent autonome utilisant ces informations est ainsi plus avantageux qu'un conducteur humain. Il est constamment attentif et peut avoir un temps de réaction bien inférieur au conducteur. De plus, son champ de vue possède moins d'angles morts. Ces avantages sont toutefois dépendants de la portée limitée des capteurs. L'utilisation de la communication augmente alors les capacités de perception d'un véhicule intelligent. Cette nouvelle source d'information est cependant une source potentielle de données trompeuses et nécessite de faire confiance aux autres

1. Introduction générale

véhicules. En la combinant à d'autres sources d'information, il est toutefois possible de vérifier les informations reçues. La communication peut se faire au travers d'Internet grâce à une communication cellulaire, comme c'est le cas pour les systèmes de navigation GNSS actuels indiquant le trafic routier. Des applications comme Google Maps ou Waze indiquent par exemple les routes où le trafic est fortement ralenti ce qui permet au conducteur d'utiliser des routes annexes. Le 28 avril 2015, le parlement européen a rendu obligatoire l'intégration de l'eCall dans l'ensemble des modèles de véhicules construits à partir du 31 mars 2018 [Parlement européen, 2015]. Ce système d'appel d'urgence permet de contacter directement les secours en cas d'accident. En connectant ce système directement au bus CAN du véhicule, le choc de l'accident est détecté et la position absolue du véhicule est communiquée même si les passagers sont inconscients.

Des systèmes communicants sont dorénavant disponibles dans les véhicules. Dans le cas de véhicules intelligents, une communication entre véhicules (V2V) et entre un véhicule et une infrastructure (V2I) peut grandement améliorer l'interaction entre les véhicules, mais aussi avec l'environnement. Cette interaction n'est en effet plus limitée à l'utilisation des feux de signalisation et de l'avertisseur sonore. Lors d'une navigation en convoi [Axelsson, 2017] par exemple, il est important de connaître l'intention d'un véhicule avant même de le voir agir. Ainsi, si le véhicule de devant décide de freiner, le véhicule de derrière peut commencer à freiner en un temps très bref lié à la communication, sans attendre de le voir se rapprocher. La gestion d'intersection est aussi une problématique majeure pour la coopération de véhicules intelligents. L'utilisation de la communication permettrait aux véhicules de négocier entre eux l'ordre de passage sans utiliser les feux tricolores ou les priorités et ainsi d'avoir une gestion plus fluide du trafic. Cette communication permet aussi au véhicule de connaître la position d'un autre véhicule sans même le percevoir. Le standard européen CAM (Cooperative Awareness Message) [ETSI, 2014a] permet entre autres d'obtenir cette information. Le partage de la perception permet d'étendre le champ de vue d'un véhicule communicant. Le message CPM (Cooperative Perception Message) [ETSI, 2018] permet la communication de cette perception. Ces différents messages permettent d'améliorer la navigation bien au-delà des capacités d'un conducteur humain. De nombreux systèmes de communication comme les VANet (Vehicular Ad-Hoc Network) sont étudiés dans cette optique [Hoang *et al.*, 2018].

1.2. Retour d'expérience sur le challenge GCDC

Le laboratoire a participé à la deuxième édition du Grand Cooperative Driving Challenge (GCDC), ce qui nous a permis d'avoir une vision d'ensemble sur la navigation coopérative et ses objectifs. Cette compétition européenne, qui a eu lieu du 28 au 29 mai 2016 à Helmond, aux Pays-Bas, avait pour objectif de mettre en pratique et de démontrer l'intérêt de la navigation coopérative sur trois scénarios caractéristiques.

1. Introduction générale



FIGURE 1.1. – *Scénario d’insertion sur l’autoroute A270 à Helmond, Pays-Bas, fermée à la circulation pendant les épreuves du GCDC.*

1.2.1. Présentation des scénarios du GCDC

1.2.1.1. Scénario d’insertion

Le premier scénario (Fig. 1.1), le scénario d’insertion débutait par une conduite en convoi sur les deux voies d’une autoroute. Les véhicules de la voie de gauche se déplaçaient à une vitesse de 80 km/h tandis que ceux de la voie de droite allaient à 60 km/h. Des travaux étaient alors signalés aux véhicules par une unité de bord de route RSU (*roadside unit*), après quoi tous les véhicules de la file de gauche devaient s’insérer dans la file de droite. Cette communication I2V (*infrastructure-to-vehicle*) utilisait le standard européen DENM (*Distributed Environment Notification Message*) [ETSI, 2014b]. Pour réaliser la manœuvre de fusion des deux voies, chaque véhicule de la file de droite devait augmenter son interdistance avec le véhicule se situant devant lui (appelé MIO pour *Most Important Object*) pour que le véhicule à sa gauche (appelé le *forward MIO*) puisse s’insérer. Le message non standard iCLCM (*i-GAME Cooperative Lane Change Message*) était alors utilisé pour prévenir le véhicule de gauche qu’il pouvait s’insérer en toute sécurité.

1.2.1.2. Scénario d’intersection

Le second scénario (Fig. 1.2) traitait du problème de gestion d’une intersection. Trois véhicules approchaient d’une intersection en « T » à 30 km/h. Les deux véhicules de la route principale devaient alors négocier l’insertion du troisième véhicule qui était prioritaire.

1. Introduction générale



FIGURE 1.2. – *Scénario d’intersection. La Zoe autonome du laboratoire Heudiasyc est visible sur cette photo. La lumière verte allumée sur le toit indiquait aux juges-arbitres du GCDC que la manœuvre se faisant effectivement en mode autonome.*

1.2.1.3. Scénario de priorité au véhicule d’urgence

Finalement, le dernier scénario (Fig. 1.3) montrait l’intérêt des véhicules coopératifs pour la circulation de véhicules d’urgence. L’ensemble des véhicules roulant à 80 km/h sur les deux voies d’une autoroute devait libérer un passage au centre de ces voies pour laisser passer un camion de pompier. Ce dernier scénario était à but illustratif et n’a pas été réalisé en conduite autonome.

1.2.2. Présentation du véhicule expérimental utilisé

L’article [Xu *et al.*, 2018] décrit plus en détails les algorithmes qui ont été développés pour le GCDC et donne davantage de résultats expérimentaux que ceux décrits dans cette section.

Dix équipes internationales ont participé au GCDC, apportant ainsi une grande diversité de véhicules [Alonso *et al.*, 2018]. Ainsi deux camions et des voitures thermiques et hybrides étaient présents. Le véhicule que nous avons utilisé était une Renault Zoe. Il s’agissait du seul véhicule à motorisation électrique participant au challenge.

L’un des intérêts d’un véhicule électrique comme une Renault Zoe pour le contrôle autonome est l’accès au couple d’accélération et de décélération directement à partir du bus CAN (*Controller Area Network*) sans avoir à ajouter d’actionneur. Le bus CAN permet ainsi d’avoir accès aux actionneurs équipant d’origine le véhicule mais aussi aux capteurs proprioceptifs déjà intégrés au véhicule. Sur de nombreux véhi-

1. Introduction générale



FIGURE 1.3. – *Scénario de priorité au véhicule d’urgence (autoroute A270).*

cules récents déjà équipés d’ADAS (*Advanced driver-assistance systems*), ce bus de communication donne accès à divers capteurs et actionneurs. L’ABS (système antiblocage des roues, *Antiblockiersystem* en allemand) et l’ESP (*Electronic Stability Program*) ont par exemple besoin de la vitesse des roues ou des informations inertielles du véhicule (accélérations et vitesses de rotation). La direction assistée donne quant à elle accès à l’angle du volant et au contrôle de celui-ci. La robotisation du véhicule a été effectuée par Renault. Pour des raisons de sécurité, le mode autonome limite la vitesse à 50 km/h. Le scénario d’insertion étant prévu à 80 km/h et à 40 km/h, nous n’avons pu faire que les épreuves à 40 km/h en conduite autonome.

1.2.2.1. Architecture du système embarqué

Afin de communiquer avec ces systèmes et mettre en place une architecture logicielle modulaire (Fig. 1.4), le logiciel PACPUS développé au laboratoire a été utilisé. Ce logiciel joue le rôle d’une surcouche (*middleware*) du système d’exploitation, comme ROS (*Robot Operating System*) [Quigley *et al.*, 2009] ou RTMaps [Steux *et al.*, 2000]. Ce logiciel permet de communiquer avec les différents systèmes connectés à l’ordinateur et de définir une architecture en donnant la possibilité aux programmes de communiquer entre eux. La communication physique avec le bus CAN du véhicule se fait à travers le système dSpace MicroAutoBox qui est connecté à l’ordinateur embarqué. L’architecture logiciel se base sur différents modules. Le module de localisation utilise la position du récepteur GNSS NovAtel SPAN-CPT et une carte et communique des informations traitées aux modules de contrôle, de perception et de supervision. La perception permet la détection des autres véhicules grâce à un LiDAR Sick LD-MRS. Le superviseur permet la gestion du scénario en se

1. Introduction générale

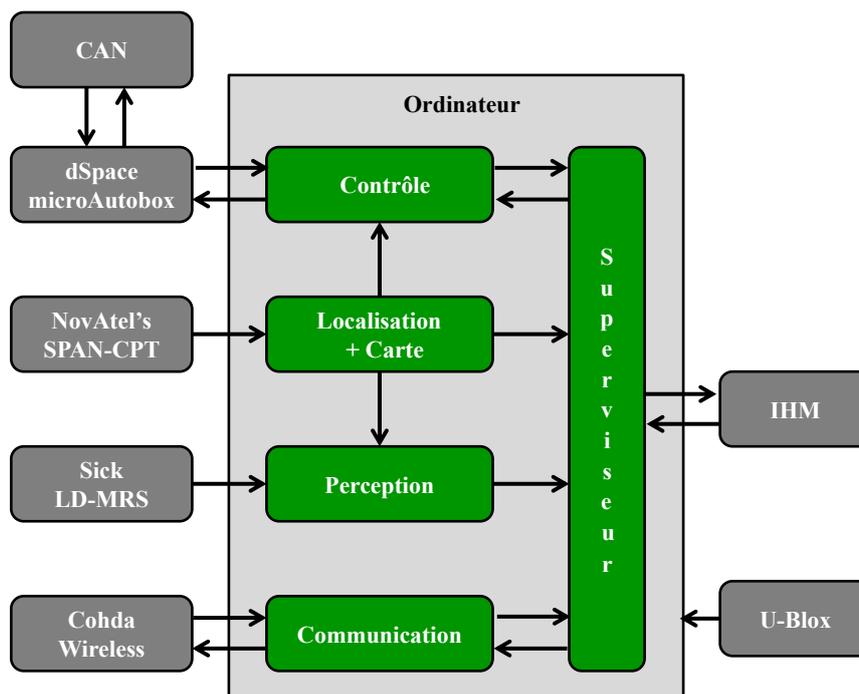


FIGURE 1.4. – Architecture globale de la plateforme. Les boîtes grises foncées représentent les systèmes physiques connectés à l'ordinateur embarqué. Les boîtes vertes sont les modules implémentés dans le logiciel PACPUS.

basant sur les données de communication obtenues au travers du module de communication et du modem Cohda, ainsi que des données de perception et de localisation. Il donne ensuite les instructions à suivre au module de contrôle qui gère l'asservissement du véhicule en communiquant aux actionneurs de celui-ci au travers du bus CAN. Un récepteur GNSS u-blox synchronise l'ordinateur embarqué au temps GPS. Une interface homme machine (IHM) envoie des informations visuelles à l'utilisateur.

1.2.2.2. Localisation et carte pour le contrôle latéral

Une carte haute définition contient en général le centre de la voie, ce qui peut servir directement de chemin de consigne pour le véhicule autonome. Une carte HD n'étant pas disponible au GCDC, les centres des deux voies du scénario d'insertion ont été préalablement enregistrés. Le maintien de trajectoire était alors réalisé en régulant la distance entre la position du véhicule et le centre de la voie. Le changement de voie consistait à changer simplement la trajectoire de consigne. L'utilisation de gains correspondants à des constantes de temps de quelques secondes pour le contrôleur latéral permettait alors des changements de voie suffisamment confortables.

Afin de déterminer la distance entre la position du véhicule et la trajectoire de consigne le plus précisément possible, une localisation précise est nécessaire. Nous avons utilisé pour cela une SPAN-CPT de NovAtel. Ce récepteur GNSS utilise les constellations américaines Navstar GPS et russes GLONASS ainsi que des satellites géostationnaires comme ceux du système EGNOS (*European Geo-Stationary Navigation System*). Une grande partie de l'incertitude sur la position d'un récep-

1. Introduction générale

teur GNSS provient de la déformation des signaux par les différentes couches de l’atmosphère, un biais sur la position peut alors apparaître. Des corrections RTK (*Real-Time Kinematic*) ont été utilisées pour une meilleure précision. Ces corrections RTK permettaient alors de corriger ces déformations en les observant à partir d’une antenne fixe dont la position était connue. La SPAN-CPT contient, en plus du récepteur GNSS, une centrale inertielle qui lui évite de dériver trop rapidement si les corrections RTK ne sont plus reçues. Le récepteur GNSS transmet la position sous forme de coordonnées géodésiques (latitude, longitude, altitude). En pratique, des coordonnées cartésiennes sont plus simples à utiliser. Nous avons utilisé pour cela des coordonnées ENU (*East, North, Up*) pour définir un repère de travail proche de la zone de navigation des véhicules. Ce système de coordonnées est décrit en détails dans l’annexe C.

Une fois la position obtenue, elle peut être mise en correspondance avec la carte afin de déterminer la distance la plus proche entre ce point et le centre de la voie. Cette distance signée (positive lorsque ce point est à gauche du centre de la voie et négative lorsqu’il est à droite) donne ainsi l’erreur latérale pour le contrôle. En déterminant la tangente au point le plus proche, on détermine l’erreur d’orientation en calculant la différence entre l’orientation du véhicule estimée par la SPAN-CPT et l’orientation de cette tangente. Cette information est utilisée pour estimer l’action dérivée du contrôleur.

1.2.2.3. Perception et communication pour le contrôle longitudinal

Dans le scénario d’insertion, le contrôle longitudinal consistait à réguler l’interdistance avec le véhicule de devant. Il commençait par une conduite en convoi. Lorsque le message signalant des travaux était reçu, les véhicules étaient alors associés deux à deux. La figure 1.5a montre ces associations ainsi que l’interdistance entre le véhicule bleu et le véhicule rouge. Cette interdistance doit ensuite être augmentée (Fig. 1.5b) afin de laisser la place au véhicule de gauche (le véhicule vert) de s’insérer (Fig. 1.5c). À partir du début de cette insertion, les interdistances pour la conduite en convoi sont calculées entre le véhicule rouge et le véhicule vert ainsi qu’entre le véhicule vert et le véhicule bleu (Fig. 1.5d). Une fois cette insertion terminée, c’est au tour des véhicules de derrière (les véhicules gris) d’effectuer cette insertion.

Le scénario d’intersection peut se ramener à un problème de conduite en convoi. En effet, en calculant la différence des distances restantes à parcourir par les deux véhicules jusqu’au centre de l’intersection et en soustrayant la longueur du véhicule le plus proche de cette intersection, l’interdistance virtuelle peut être déterminée. La figure 1.6 illustre cette conduite en convoi virtuel entre le véhicule prioritaire (en rouge) et le véhicule régulant l’interdistance virtuelle (en vert), en prenant en compte leurs trajectoires (en pointillés). Le véhicule bleu arrivant de la droite peut réaliser lui aussi une conduite en convoi virtuel avec le véhicule rouge.

Dans ces deux scénarios, les interdistances sont essentielles aux interactions entre les véhicules.

Pour déterminer ces interdistances, un LiDAR SICK LD-MRS utilisant quatre nappes balayant sur 110° a été exploité. Seule la couche parallèle au sol était utilisée lors de cette tâche de perception. Les scans LiDAR étaient découpés en groupes de points se trouvant proches les uns des autres. Des boîtes englobant ces groupes étaient alors déterminées, les boîtes trop grandes ou trop petites étaient filtrées. En

1. Introduction générale

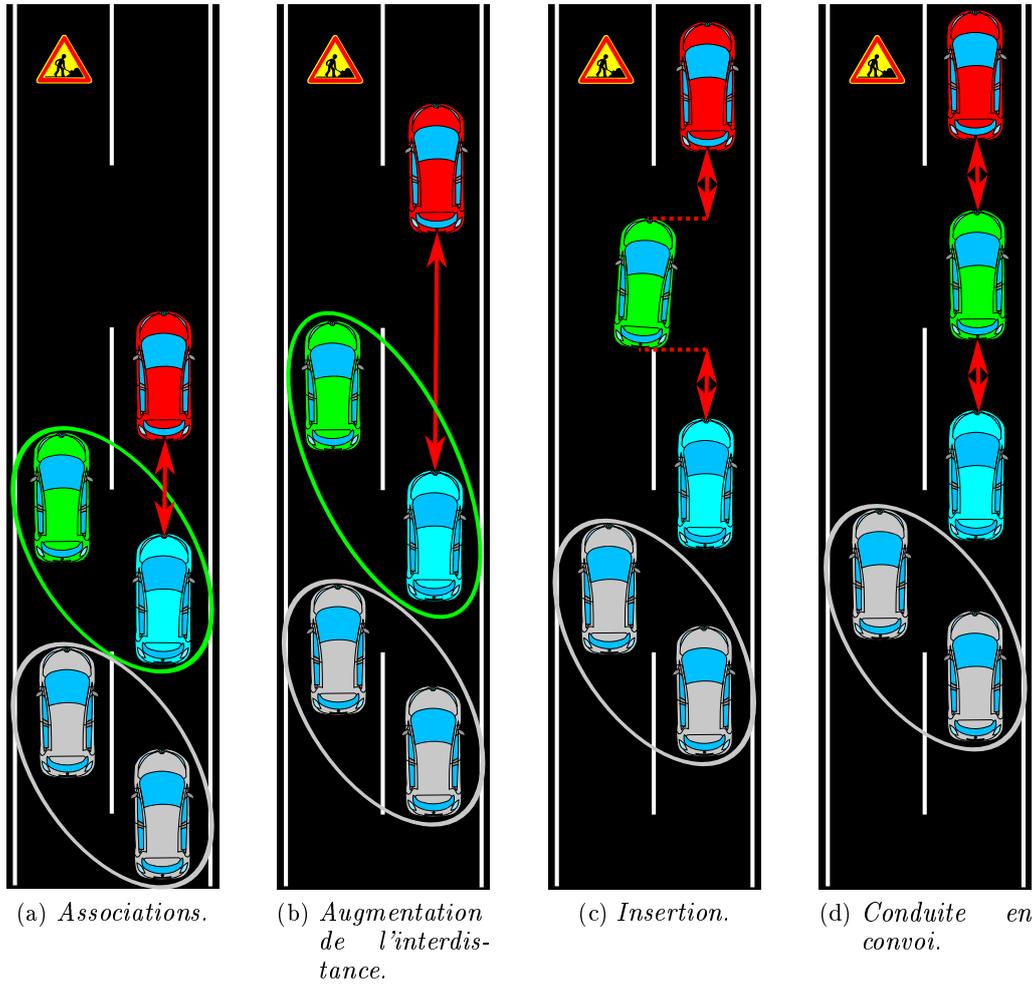


FIGURE 1.5. – *Scénario d'insertion.*

déterminant la distance de chaque coin au centre de la voie pour chaque boîte, il était possible de déterminer si c'était le véhicule de devant (en rouge sur la figure 1.7), le véhicule de devant se trouvant sur l'autre voie (en jaune sur la figure 1.7), un véhicule en train de changer de voie (en orange sur la figure 1.7) ou un véhicule sur le bord de la voie (en bleu sur la figure 1.7).

Dans le cas où un véhicule se trouve derrière un autre obstacle, comme un autre véhicule, un bâtiment ou de la végétation, une communication entre les véhicules permet de recevoir les positions des autres véhicules et ainsi d'étendre la perception. Dans le cas du GCDC, une communication 802.11p était utilisée à l'aide d'un modem Cohda. Pour permettre de partager sa position, son orientation, sa vitesse ainsi que les dimensions de son véhicule le standard européen CAM (*Cooperative Awareness Message*) a été implémenté. En calculant la distance euclidienne entre la position d'un véhicule et la position de l'égo-véhicule, une seconde estimation de l'interdistance était alors disponible.

1. Introduction générale

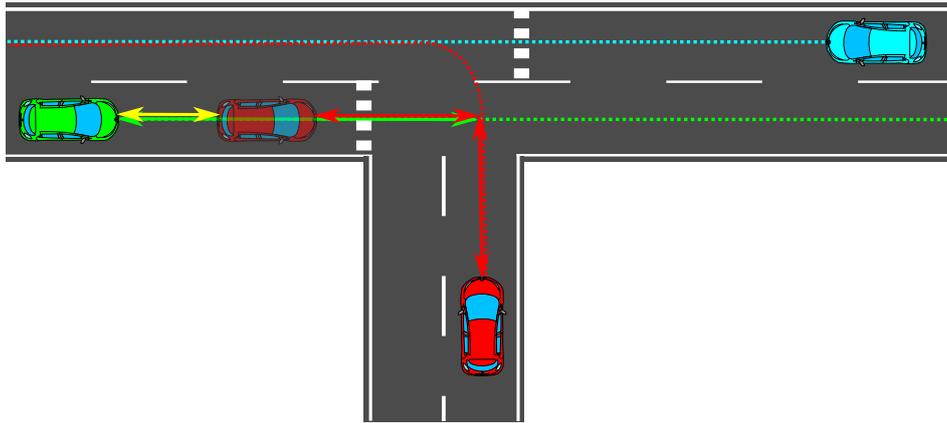


FIGURE 1.6. – Scénario d’intersection par la méthode du convoi virtuel. En ramenant la voiture rouge virtuellement sur la voie de la voiture verte, on voit qu’elle est plus proche de l’intersection. La voiture verte régule sa position par rapport à la voiture virtuelle ce qui garantit un franchissement d’intersection sans collision. La voiture bleue peut suivre une stratégie similaire.

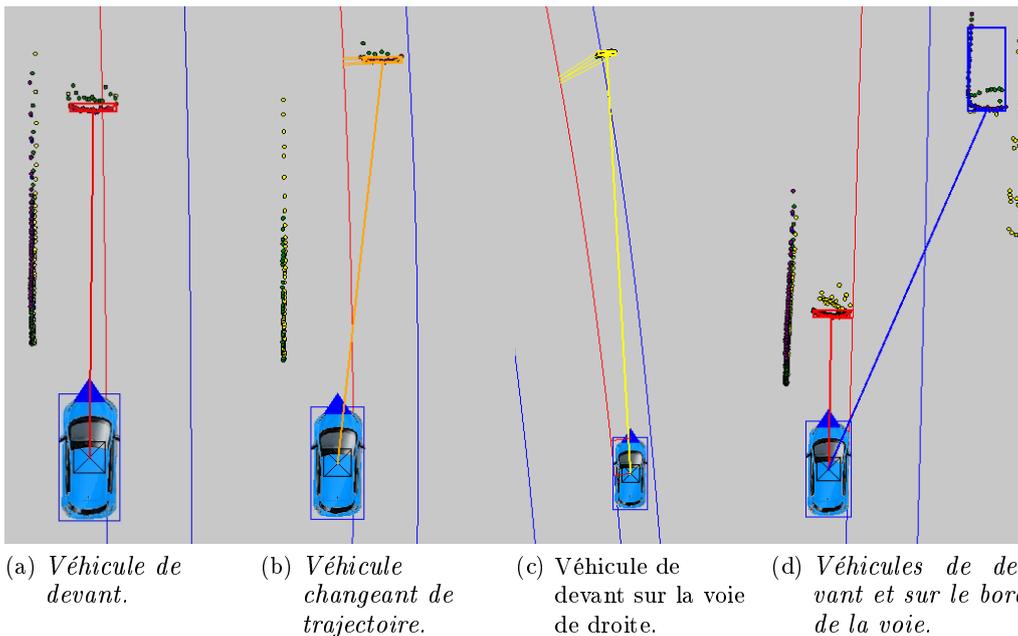


FIGURE 1.7. – L’égo-véhicule (en bleu) estime la boîte englobante du groupe de point LiDAR correspondant au véhicule de devant pour la nappe horizontale du LiDAR. Les points des différentes nappes du LiDAR sont représentés par des couleurs différentes. La voie sur laquelle le véhicule perçu roule est déterminée à partir de la distance entre les sommets de cette boîte englobante et de la trajectoire du égo-véhicule (en rouge) ou de la seconde voie (en bleu).

1. Introduction générale

TABLE 1.1. – Modes de correction avec précision [NovAtel, 2013].

Mode	Précision
RTK Fixed Ambiguities Solution (RTKFIXED)	$\sim 0,01\text{ m}$
RTK Floating Point Ambiguities Solution (RTKFLOAT)	$< 1,0\text{ m}$
Pseudorange Differential Solution (PSRDIFF)	$< 2,0\text{ m}$
Satellite-Based Augmentation System (SBAS)	$\sim 3\text{ m}$
Pseudorange Single Point Solution (PSRSP)	$\sim 5\text{ m}$

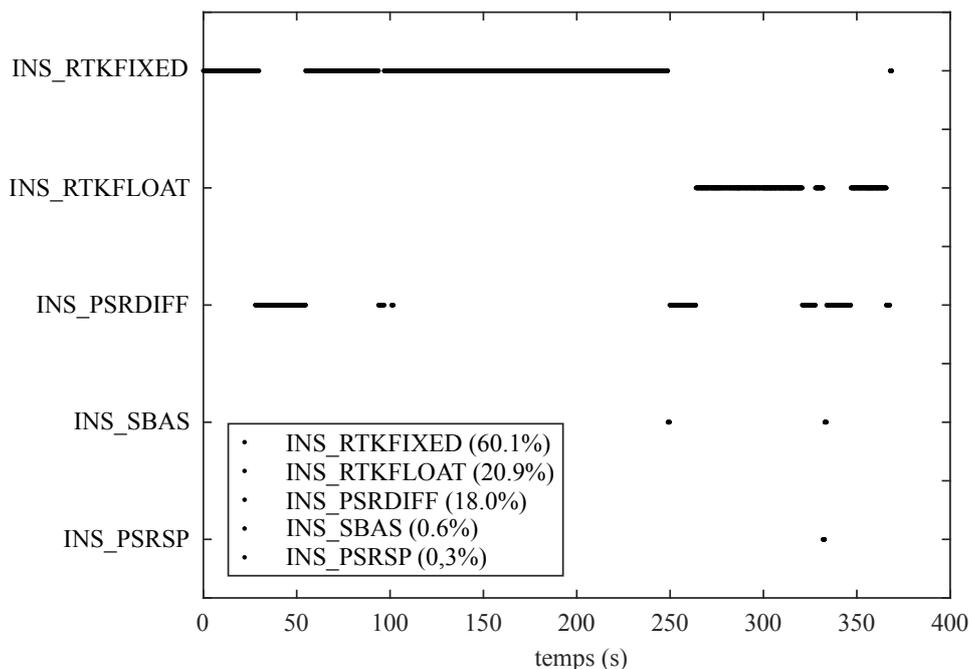


FIGURE 1.8. – Modes de correction durant un scénario d'insertion.

1.2.3. Quelques leçons apprises du challenge

Pour la localisation du véhicule, la SPAN-CPT était utilisée, ce système fusionnant la centrale inertielle et le récepteur GNSS avec correction RTK. Elle était alors capable de fournir des positions avec une précision de quelques centimètres. Le prix de ce dispositif est toutefois trop onéreux actuellement pour une commercialisation au grand public. Un récepteur GNSS bas coût, comme ceux du fabricant u-blox, a donc aussi été utilisé.

La table 1.1 présente les différents modes de correction de la position fournie par la SPAN-CPT avec la précision pouvant être attendue après correction. Les modes RTKFIXED, RTKFLOAT et PSRDIFF peuvent être obtenus lorsque des corrections différentielles sont reçues au contraire des modes SBAS et PSRSP. Le mode RTKFIXED est le mode le plus précis et compatible avec les exigences d'une navigation autonome. Grâce à la centrale inertielle et à la dérive lente des biais GNSS, il est possible d'utiliser les modes dégradés RTKFLOAT ou PSRDIFF sur de courtes périodes de temps lorsque le RTKFIXED a été perdu (de l'ordre de quelques dizaines de secondes selon notre expérience).

1. Introduction générale

La figure 1.8 montre les différents modes de correction disponible avec la SPAN-CPT pendant un essai du scénario d’insertion. On observe ainsi que même si le mode RTKFIXED est majoritairement présent, le calcul devient flottant pendant des durées assez longues. Lors du GDCD, nous avons utilisé la SPAN-CPT sans traitement supplémentaire et nous avons été satisfaits des performances. Cependant, pour un déploiement des véhicules autonomes, il ne faut pas espérer une navigation autonome uniquement basée sur un récepteur GNSS hybridé avec une centrale inertielle. Les corrections différentielles ne pourront en effet pas être disponibles sur toutes les routes. Une fusion avec d’autres capteurs est indispensable.

La détection du véhicule de devant était basée sur la perception par boîte englobante de points LiDAR. L’interdistance est alors calculée entre la position du LiDAR et le point le plus proche de celui-ci. Le principal problème observé lors du GDCD avec cette perception LiDAR était la diminution du nombre de points LiDAR sur le véhicule lorsque celui-ci s’éloignait. Il était ainsi très difficile de conserver la détection par boîte englobante lorsque l’interdistance était de l’ordre de 50 m. La communication était alors utilisée pour estimer cette interdistance lorsque le véhicule était trop loin et n’était plus perçu. Lorsque le véhicule était près, la perception LiDAR était préférée, la position provenant des autres véhicules n’étant pas toujours suffisamment précise. On s’est en effet aperçu que, même dans le cas du GDCD où aucun des participants n’était mal intentionné, les positions transmises dans les messages de communication pouvaient être très imprécises et trompeuses. Notre implémentation de la perception dans le GDCD a donc utilisé la complémentarité entre la perception LiDAR et la perception reçue de la communication.

1.3. Représentation du monde pour la navigation autonome

Comme nous avons pu le voir, un véhicule autonome doit être capable de connaître son environnement pour rester dans sa voie ou éviter des obstacles. Cela peut être réalisé grâce à une carte contenant des informations statiques et grâce à une perception et/ou une communication permettant de mettre à jour la connaissance qu’a le véhicule des informations dynamiques de l’environnement.

1.3.1. Cartes Haute Définition

Les cartes sont des bases de données géographiques permettant de représenter le monde pour une application donnée (Fig. 1.9, 1.10 et 1.11). Dans le cas de la navigation autonome, elles doivent permettre de planifier des trajectoires, à long terme, pour aller à une destination prédéfinie ou à court terme pour rester dans sa voie ou changer de voie. Ce sont des cartes vectorielles navigables. Elles sont ainsi nécessaires pour la navigation et le contrôle, mais sont aussi utiles pour la perception et la localisation. Elles font en effet le lien entre la perception et la localisation absolue en plaçant le plus exactement possible de nombreux points de repère. Ainsi, ces cartes contiennent des informations pour chaque voie, les centres et largeurs de ces voies ou les bords de celles-ci. Ce centre de voie peut être le chemin à suivre dans la

1. Introduction générale



FIGURE 1.9. – *Vue satellite du centre d'innovation de l'UTC. La piste SEVILLE est visible sur la partie haute. (Leaflet | Tiles © Esri - Source : Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community).*



FIGURE 1.10. – *Carte Open Street Map. La piste SEVILLE est en rose.*

1. Introduction générale

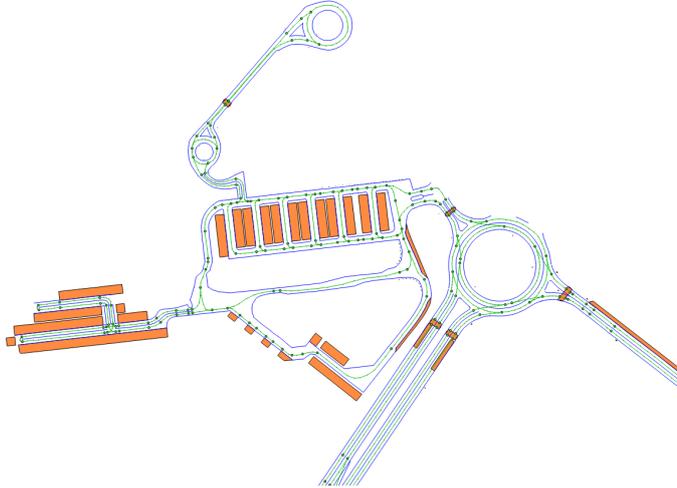


FIGURE 1.11. – *Carte Haute Définition.* Le centre de chaque voie est tracé en vert, les bords de voie sont en bleu et des zones particulières (parkings, passages piétons, arrêts de bus, etc.) sont représentées en orange.

majorité des situations. À l'exception des manœuvres d'urgence et des changements de voies, un véhicule se contentera en général de rester au centre de la voie. Dans le cas d'évitement d'obstacle, s'il a le temps d'anticiper, il peut simplement s'arrêter, passer sur la voie de gauche ou éventuellement dépasser en restant sur la voie courante si l'obstacle a une largeur suffisamment faible (vélo ou piéton par exemple). Elles contiennent aussi les bords des voies qui peuvent servir pour rester dans ces voies ainsi que comme amers pour se localiser. De nombreuses recherches sont réalisées pour détecter ces bords de voies comme les travaux de [Mario et Rife, 2010]. Des signalisations peuvent aussi être ajoutées dans la carte haute définition, comme les panneaux, feux tricolores ainsi que les portions de voies où ces informations s'appliquent. De nombreux autres amers peuvent être présents (bâtiments, poteaux électriques, etc.). La carte haute définition que nous utilisons dans la suite de cette étude a une précision centimétrique. La figure 1.11 la décrit plus en détails.

1.3.2. Cartes Locales Dynamiques

Les cartes n'étant pas mises à jour suffisamment régulièrement, seules des données statiques peuvent y être stockées. Grâce à la communication, il est possible d'enrichir ces cartes avec des surcouches dynamiques (Fig. 1.12) [Eiter *et al.*, 2019, Shimada *et al.*, 2015]. Une connexion internet permet déjà aux cartes des applications GNSS grand public de donner des informations sur le trafic routier.

Une connexion internet n'assure pas une latence fixe et ainsi ne permet pas de recevoir des informations à temps de façon fiable. Il est alors nécessaire d'utiliser de nouvelles méthodes de communication, en utilisant les standards dédiés aux véhi-

1. Introduction générale

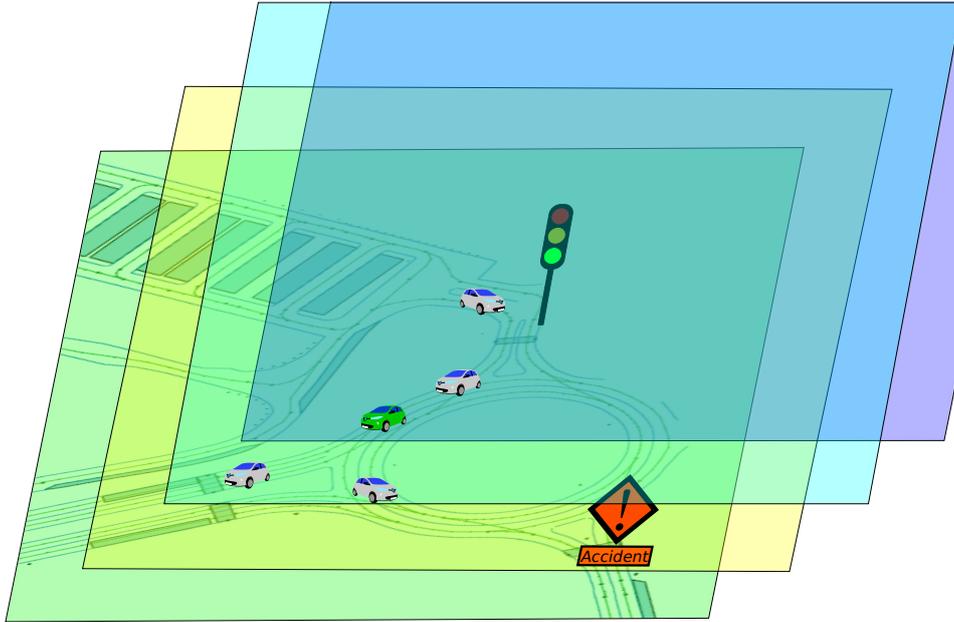


FIGURE 1.12. – La couche statique de la carte (en vert) contient la route ainsi que l’environnement statique du véhicule. Le cloud (en jaune) permet d’avoir des renseignements sur le trafic comme les accidents. Les infrastructures communicantes (en violet) permettent de compléter la carte statique avec des informations dynamiques, comme l’état des feux. Enfin, la carte locale dynamique (en bleu) donne l’état des obstacles dynamiques (les véhicules gris) au voisinage du égo-véhicule (en vert).

cules autonomes (comme la future 5G) ou des communications plus directes avec des modems Wi-fi utilisant le standard 802.11p comme ceux utilisés pour le GCDC. Les véhicules autonomes reçoivent alors de nouvelles informations pouvant être ajoutées à leur carte. Les infrastructures (V2I) aident alors le trafic notamment en communiquant l’état des feux. Une autre communication peut avoir lieu entre les véhicules (V2V) et ainsi permettre de connaître l’emplacement de ces véhicules communicants ainsi que leurs informations cinématiques permettant de prédire leurs poses jusqu’aux messages suivants. Cette communication permet aussi d’augmenter la portée de la perception en communiquant celle-ci [Zoghby *et al.*, 2014]. Il est alors possible de percevoir un obstacle hors des champs de vue des différents capteurs du véhicule.

1.4. Le problème de la localisation coopérative

Une première approche [Kurazume *et al.*, 1994] issue de la robotique mobile consiste à déplacer deux groupes de robots l’un après l’autre, le groupe statique servant d’amers au groupe en mouvement. Cette méthode consistant à utiliser les autres robots comme points de repère lorsque ceux-ci communiquent leurs positions ou leurs poses s’est généralisée au cas où ces autres robots se déplacent et, dans ce cas, ils servent de points de repère dont il faut tenir compte de l’incertitude [Martinelli *et al.*, 2005]. Certaines de ces recherches traitent ainsi de structures ri-

1. Introduction générale

guides entre robots [Zelazo *et al.*, 2014, Zelazo *et al.*, 2015] permettant à ceux-ci de se déplacer en essaim. Les liens entre les nœuds formés par les positions partagées par les robots sont déterminés en mesurant les interdistances entre ces robots ou les angles d’observations [Martinelli *et al.*, 2005]. Afin de mesurer ces interdistances ou ces angles d’observation [Montesano *et al.*, 2005], différents capteurs extéroceptifs peuvent être utilisés : des caméras [Challita *et al.*, 2009], [Khammari *et al.*, 2005], des radars ou encore des LiDARs.

Ces méthodes sont présentes dans différents domaines de la robotique mobile. Les AUV (Autonomous Underwater Vehicules) utilisent les ondes acoustiques pour déterminer les distances entre ces sous-marins autonomes avant de déterminer leurs positions par trilatération [Bahr *et al.*, 2009]. Les ondes acoustiques sont aussi utilisées par des robots terrestres [Lin *et al.*, 2005]. Dans le cas des véhicules communicants, les interdistances [Rohani *et al.*, 2015] peuvent être déterminées à partir d’autres signaux. Les modems utilisés pour une communication directe de type 802.11p peuvent ainsi retourner l’indicateur de la puissance du signal reçu RSSI (received signal strength indicator) [Cruz *et al.*, 2017, Hoang *et al.*, 2015, Severi *et al.*, 2018]. Une autre solution bas coût est d’utiliser des systèmes Ultra Large Bande UWB (Ultra WideBand) [Hoang *et al.*, 2017, Hoang *et al.*, 2018, Severi *et al.*, 2018]. Le temps d’arrivée de messages communiqués peut aussi permettre d’estimer une interdistance [Peker et Acarman, 2017].

Les véhicules intelligents autonomes peuvent améliorer leur localisation en utilisant la communication.

La localisation coopérative peut être déterminée de façon centralisée en définissant une centrale réalisant l’ensemble du traitement [Worrall et Nebot, 2007]. Dans le cas de véhicules autonomes, il est toutefois préférable de pouvoir rester indépendant d’une centrale afin d’être robuste à une coupure de communication. Une méthode décentralisée [Mu *et al.*, 2011, Roumeliotis et Bekey, 2000] est alors souvent préférée malgré le fait qu’elle soit moins optimale. Cette collaboration est réalisée grâce au partage d’informations. Ainsi, les poses relatives des robots observés [Howard *et al.*, 2003] ou les états des robots peuvent être échangés [Aufrère *et al.*, 2010, Karam *et al.*, 2006a, Karam *et al.*, 2006b, Karam, 2009, Tischler et Hummel, 2005]. Une autre solution est de partager directement les mesures capteurs [Madhavan *et al.*, 2002] comme les bords de voies ou les mesures GNSS [Rife et Xiao, 2010].

Différentes recherches traitent de la correction qui est effectuée sur les informations GNSS grâce à la communication entre véhicules. Les modules GNSS récupèrent en effet de nombreuses mesures entachées d’erreurs [Drawil et Basir, 2010]. Les erreurs liées aux éphémérides, à l’ionosphère et à la troposphère peuvent être assimilées à un biais sur la mesure. Afin de les soustraire, des stations GNSS fixes détectent ces biais à partir de leur position réelle connue. Elles envoient alors ces biais aux véhicules en mouvement. Une coopération entre véhicules permet de détecter ces biais et de les supprimer [Lassoued *et al.*, 2015, Lassoued *et al.*, 2014] sans nécessiter l’utilisation de stations fixes en échangeant les pseudo-distances GNSS [Lassoued *et al.*, 2015, Lassoued *et al.*, 2017]. Une autre solution est d’utiliser une méthode de GPS différentiel DGPS (Differential Global Positioning System) [Müller *et al.*, 2014] comme la méthode de station de base dynamique DGPS (DDGPS) [Rohani *et al.*, 2016].

Les corrélateurs bas niveau GNSS fournissent aussi parfois des informations erro-

1. Introduction générale

nées provenant des effets de multi-trajets qui apparaissent lorsque les signaux émis par les satellites sont réfléchis par des surfaces d'eau, des bâtiments, etc. Le récepteur GNSS reçoit alors une trame identique à celle qu'il aurait dû recevoir directement, mais avec un temps de trajet plus long. Là aussi, la coopération multi-véhicules permet de supprimer ces erreurs en détectant ces mesures aberrantes [Drawil et Basir, 2010, Zhang *et al.*, 2018].

Lors de la localisation coopérative, il est important de pouvoir correctement propager les incertitudes [Roumeliotis et Rekleitis, 2003, Roumeliotis et Rekleitis, 2004] pour estimer la confiance associée aux états recherchés [Roumeliotis et Bekey, 2002] afin de conserver une bonne intégrité [Liu *et al.*, 2018]. Il est aussi important de pouvoir détecter les défauts et rejeter les mesures aberrantes [Al Hage, 2016]. Les méthodes d'analyse par intervalle [Lassoued *et al.*, 2014, Kieffer *et al.*, 2000, Jaulin *et al.*, 2001] comme la propagation de contraintes permettent de conserver une bonne intégrité [Jo et Lee, 2008] en les combinant avec des méthodes robustes appelées « q-relaxées » [Jaulin *et al.*, 2001]. Les méthodes probabilistes permettent aussi de manipuler les incertitudes efficacement [Fox *et al.*, 2000] et les données aberrantes peuvent aussi être filtrées.

Une des principales sources d'erreur dans l'estimation de cette incertitude dans un système de localisation coopérative provient de la communication de l'état des véhicules. En effet, si un véhicule estime son état à partir de celui d'un second et réciproquement, alors certaines informations sont utilisées plusieurs fois. Certaines méthodes de fusion de données comme le filtre de Kalman suppose que les erreurs sur les informations à fusionner sont indépendantes. En cas d'utilisation multiple de la même information, une consanguinité de données apparaît ce qui entraîne une surconfiance dans le résultat obtenu. La communication des mesures capteurs permet d'éviter ce problème, mais augmente significativement la quantité d'information communiquée. Chaque véhicule peut alors mettre à jour un arbre de dépendance pour éviter ce problème [Howard *et al.*, 2003]. Dans le cas où les erreurs ne sont pas indépendantes, la corrélation entre ces erreurs est complexe à estimer. L'intersection de covariance [Julier et Uhlmann, 1997b, Li et Nashashibi, 2012, Reinhardt *et al.*, 2012, Carrillo-Arce *et al.*, 2013] permet de rester consistant dans ce type de situation. Elle consiste à trouver la plus petite ellipse d'incertitude contenant les ellipses d'incertitude à fusionner. Cette méthode de fusion de données donne cependant souvent des résultats pessimistes et moins précis que ceux donnés par un filtre de Kalman. Afin d'éviter d'être trop pessimiste et de rester précis, le filtre par intersection de covariance partitionnée peut être utilisé [Delobel, 2018, Julier et Uhlmann, 2001, Li et Nashashibi, 2012, Li, 2012, Li et Nashashibi, 2013, Wanasinghe *et al.*, 2014]. Il présente un bon compromis entre le filtre de Kalman et le filtre par intersection de covariance.

1.5. Objectifs de la thèse

Cette thèse couvre l'étude et la modélisation du problème de la localisation coopérative de véhicules intelligents pouvant communiquer et se percevoir mutuellement, car ils sont équipés de capteurs extéroceptifs leur permettant de réaliser des tâches de navigation autonome. On s'intéresse aussi à la prise en compte d'informations issues de cartes haute définition. Les informations de perception ne pouvant être modélisées

1. Introduction générale

comme des contraintes spatiales déterministes sans erreur, on considèrera le cas où ces données sont entachées d'incertitude.

L'objectif de la thèse est d'étudier des méthodes qui permettent à chaque véhicule d'obtenir une représentation dynamique des agents communicants en interaction avec lui. Ces méthodes sont au cœur des systèmes fournissant une couche d'abstraction en réalisant en temps réel la fusion de toutes les données disponibles. Cette représentation dynamique doit être consistante et intègre de façon à ne pas contenir d'informations trompeuses dues à des erreurs de mesure mal manipulées. Un objectif est donc d'estimer les grandeurs inconnues qui nous intéressent avec des domaines de confiance fiables.

À partir de cette représentation dynamique du monde, les méthodes de décision et de contrôle peuvent assurer des tâches de navigation complexes comme la navigation en convoi, la gestion des intersections, le changement de voie ou encore les dépassements. D'un point de vue de l'automatique, cette approche peut être considérée comme une généralisation du principe de séparation entre les problèmes estimation d'état et de synthèse de lois de contrôle.

Il est à noter qu'on suppose dans cette thèse que tous les agents en interaction sont communicants et non malveillants.

1.6. Plan du manuscrit

Ce manuscrit de thèse est rédigé de façon incrémentale. Les chapitres 2 à 5 décrivent les problèmes élémentaires qui seront présents dans la fusion de données de la carte locale dynamique présentée dans le chapitre 6.

Suite aux méthodes de navigation mise en œuvre lors du GCDC pour réaliser une conduite en convoi ou en convoi virtuel, une localisation relative à la carte navigable nous semble être très appropriée pour les tâches de navigation. Dans le chapitre 2, nous décrirons d'abord les coordonnées curvilignes constituées de l'abscisse curviligne, de la distance signée par rapport au centre de la voie (l'écart latéral) et de l'orientation relative à cette voie. Plusieurs coordonnées curvilignes seront décrites à partir de différentes représentations des voies de navigation. Ces différentes approches seront comparées en termes de continuité, de dérivabilité et de vitesse de calcul.

Le problème de la localisation coopérative sera ensuite posé de façon simplifiée sous la forme d'une fusion coopérative à une dimension en abscisse curviligne dans le chapitre 3. En supposant qu'une perception des bords de voie permet de connaître parfaitement (ou bien avec des erreurs négligeables) l'écart latéral et l'orientation relative, l'abscisse curviligne reste incertaine et difficile à estimer. Or, dans le cas d'une conduite en convoi, cette donnée est essentielle pour obtenir une bonne régulation et éviter les accidents. En prenant l'hypothèse supplémentaire qu'une perception permet d'obtenir la pose relative précise entre les deux véhicules conduisant en convoi, on montrera l'intérêt de la coopération dans ce cas simplifié, notamment lorsque les deux véhicules se déplacent dans des virages.

Le chapitre 4 décrira différentes méthodes de localisation relative basées LiDAR améliorant la perception par boîte englobante utilisée lors du GCDC. Les principales méthodes exposées dans ce chapitre seront basées sur une méthode de type ICP (Itérative Closest Point). Quatre correspondances entre les points LiDAR et

1. Introduction générale

un modèle polygonal 2D du véhicule perçu et deux minimisations seront comparées. La méthode proposée permettra d'estimer une matrice de covariance associée à la pose relative entre les deux véhicules. La consistance de cette covariance estimée sera étudiée expérimentalement sur des données simulées.

Les problèmes de propagation de la pose relative et de sa matrice de covariance provenant de non-linéarités seront décrits dans le chapitre 5. Deux formulations de cette propagation seront proposées. L'approche communicante consiste à utiliser la pose du véhicule perçu avec la pose relative pour estimer la pose du véhicule. La seconde approche consiste à estimer la pose du véhicule perçu à partir de la pose relative et de la pose du égo-véhicule. Les problèmes de consistance de ces deux approches seront finalement comparés expérimentalement sur des données réelles.

Finalement, le problème de la localisation coopérative sera posé de façon complète dans le chapitre 6. Nous étudierons une méthode décentralisée basée sur un échange de cartes locales dynamiques et fonctionnant de façon asynchrone. L'algorithme implémenté sous ROS sera testé avec des données réelles. Cette implémentation permettra de montrer l'intérêt de l'estimation du biais GNSS afin d'obtenir une localisation précise ainsi que l'intérêt de la communication pour améliorer la précision de la pose relative. La communication permettra aussi d'estimer le biais GNSS lorsque le véhicule ne peut l'estimer seul en observant son environnement et en comparant ses observations à une carte.

Le manuscrit se terminera par une conclusion générale avec des perspectives.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

2.1. Introduction

Pour la navigation d'un véhicule intelligent autonome, il n'est pas évident à priori que la connaissance de sa position dans un repère cartésien soit la représentation la mieux adaptée. Par exemple, pour le contrôle du véhicule, il peut être plus intéressant de connaître sa position relativement à la voie. Pour ce faire, l'utilisation de cartes haute définition [Bender *et al.*, 2014, Betaille et Toledo-Moreo, 2010] devient primordiale. Ces cartes peuvent également aider à améliorer la localisation et la perception coopératives.

L'utilisation récente des coordonnées curvilignes a montré plusieurs intérêts pour la navigation de véhicules intelligents [Chu *et al.*, 2012, Kim *et al.*, 2015]. La première coordonnée correspond dans ce cas à l'abscisse curviligne le long du centre de la voie. Dans ce travail, une route peut être composée d'une ou deux chaussées elles-mêmes composées d'une ou plusieurs voies chacune. La deuxième coordonnée est un écart latéral (une distance latérale signée) par rapport au centre de la voie : elle est positive quand le véhicule est à gauche du centre de la voie et négative quand il est à droite, l'orientation de la voie étant donnée par le sens de circulation. La dernière coordonnée, l'orientation curviligne, est également définie relativement à la voie à partir d'angles tangents.

Ces coordonnées peuvent alors être utilisées directement pour le contrôle latéral avec l'écart latéral signé pour l'action proportionnelle et l'orientation relative par rapport au centre de la voie pour l'action dérivée. L'abscisse curviligne est également très bien adaptée pour le contrôle longitudinal de plusieurs véhicules coopératifs. En effet, en utilisant une communication V2V avec un partage de positions, il est alors possible d'obtenir les interdistances entre véhicules en calculant simplement des différences d'abscisses curvilignes. Cela permet par exemple de faire de la conduite en convoi le long de routes courbées ou de franchir des intersections en utilisant une conduite en convoi virtuel. Cette méthode peut par exemple être utilisée pour le franchissement de rond-points comme dans le travail proposé par [Masi *et al.*, 2018]. Lors de virages ou dans les intersections, les interdistances euclidiennes ne sont en effet pas adaptées car elles sous-estiment les vrais écarts relatifs.

Dans ce chapitre, nous allons étudier et comparer différentes manières de modéliser les routes pour le calcul de coordonnées curvilignes adaptées à la navigation de véhicules autonomes. Nos principales conclusions ont été publiées dans l'article [Héry *et al.*, 2017a]. Des cartes haute définition contenant le centre de chaque voie, les marquages et diverses autres informations ont été utilisées. La figure 2.1 illustre le concept de polyligne centrale.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

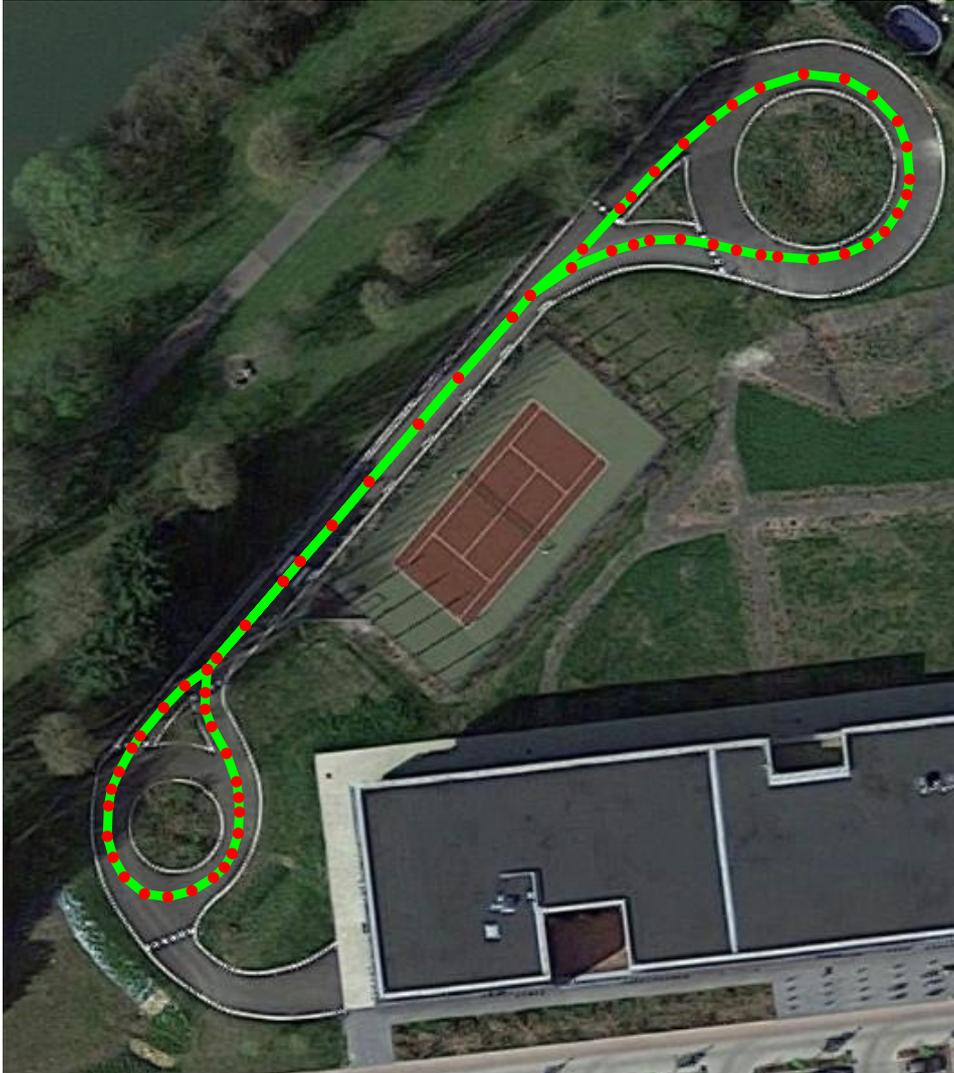


FIGURE 2.1. – Carte haute définition de la piste expérimentale de Séville de l'UTC (la vue satellite est extraite de Google Map). Le centre de la voie (en vert) est représenté par un ensemble de points de forme (en rouge).

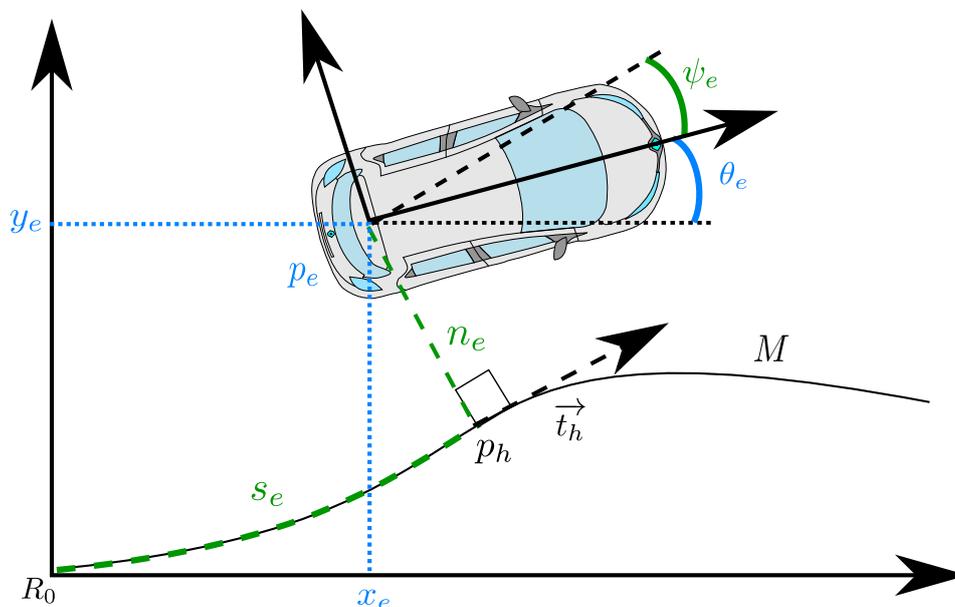


FIGURE 2.2. – Pose curviligne $g_e = [s_e \ n_e \ \psi_e]^T$ relative à la courbe M (abscisse curviligne s_e , écart latéral n_e et orientation relative ψ_e) et pose cartésienne $q_e = [x_e \ y_e \ \theta_e]^T$ du véhicule. Le vecteur tangent \vec{t}_h est pris au point p_h .

2.2. Projections et coordonnées curvilignes sur différents modèles de route

De la même façon que pour les coordonnées géodésiques utilisées par les systèmes GNSS, un modèle de référence est nécessaire pour les coordonnées curvilignes.

Dans cette thèse nous utiliserons les notations présentées dans l'annexe A.

2.2.1. Coordonnées curvilignes dans le cas général

Supposons que le véhicule autonome suive un chemin donné par le centre de la voie dans laquelle ce dernier circule. S'il ne suit pas exactement ce chemin, nous supposons qu'il en reste toujours proche. Ce centre de voie est représenté par une courbe M servant de référence pour les coordonnées curvilignes.

Nous définissons les coordonnées curvilignes comme

$$g_e = [s_e \ n_e \ \psi_e]^T, \quad (2.1)$$

où s_e est l'abscisse curviligne, n_e l'écart latéral (signé) par rapport à M et ψ_e l'angle relatif par rapport à la tangente à cette courbe M , Fig. 2.2.

Pour calculer une pose curviligne à partir d'une pose cartésienne, la première étape est appelée « *map-matching* » et elle consiste à trouver le point p_h appartenant à la courbe M tel que la distance $\|p_e - p_h\|$ soit minimale. On peut alors déterminer l'abscisse curviligne s_e comme étant la longueur de la courbe reliant un point de référence de la voie (la dernière intersection par exemple) jusqu'au point p_h . On peut ensuite calculer le vecteur tangent \vec{t}_h au point p_h . L'ordonnée curviligne n_e

TABLE 2.1. – Notations des poses curvilignes pour une courbe M .

Symbole	Description
s	Abscisse curviligne.
n	Ordonnée curviligne (écart latéral signé).
ψ	Orientation curviligne (orientation relative).
$g = \begin{pmatrix} s \\ n \\ \psi \end{pmatrix}$	Pose curviligne.
\vec{t}	Tangente.
\vec{n}	Normale.

correspond à la distance $\|p_e - p_h\|$ signée où ce signe est déterminé avec la tangente \vec{t}_h : lorsque p_e est à gauche de t_h , n_e est positif et est négatif, sinon. Il s'agit de la même convention que celle d'un repère de Frenet.

En pratique,

$$n_e = (p_e - p_h)^T \frac{\vec{n}_h}{\|\vec{n}_h\|}, \quad (2.2)$$

où $\vec{n}_h = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \vec{t}_h$ est le vecteur normal à la courbe M au point p_h tel que (\vec{t}_h, \vec{n}_h) forme une base orthonormée directe. Rappelons que d'après nos hypothèses, $\|n_e\|$ reste petit par rapport au rayon de courbure de la voie. Enfin, l'orientation relative ψ_e est la différence entre l'orientation absolue θ_e et l'orientation de la tangente \vec{t}_h .

Pour la transformation inverse, donnant la pose cartésienne à partir de la pose curviligne, le point p_h peut être déterminé à partir de l'abscisse curviligne s_e . On peut alors déterminer la tangente \vec{t}_h pour utiliser la distance latérale signée (orthogonale à cette tangente) pour déterminer le point p_e ainsi que l'orientation θ_e grâce à l'orientation relative ψ_e .

Les notations liées à la pose curviligne sont résumées dans la table 2.1.

2.2.2. Représentation avec des polygones

La forme géométrique des voies est dans la plupart des cartes HD, dont celles que nous utiliserons dans cette thèse, représentée d'une manière discrète en utilisant un ensemble de points $M = \{m_i = [x_i \ y_i]^T, i \in \{1, \dots, N\}\}$ codant un échantillonnage spatial du centre de la voie. Ces points seront parfois désignés comme des points de jonction ou bien des points de contrôle selon le modèle de route utilisé. Un premier choix pour modéliser la route est d'utiliser une polyligne, une suite de segments construits entre chaque couple de points successifs.

Pour calculer les coordonnées curvilignes, le *map-matching* doit être fait en calculant la plus petite distance entre le point p_e et chaque segment de la polyligne¹.

1. Dans le cas où deux courbes sont définies pour les deux sens de circulation d'une voie à double sens et sont donc très proches voire superposées, l'orientation relative doit être utilisée afin de résoudre cette ambiguïté. Cette orientation permet en effet de distinguer les sens de circulation de ces deux courbes.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

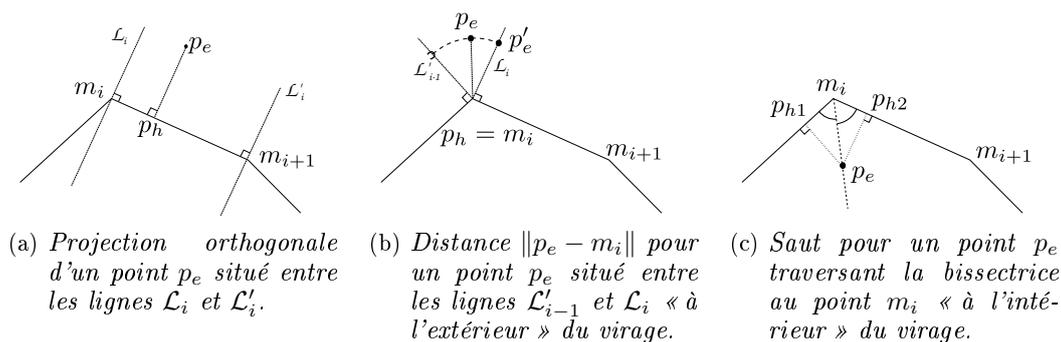


FIGURE 2.3. – Map-matching sur polyligne.

Comme illustré sur la figure 2.3, plusieurs cas se présentent :

- (a) Le point p_e est entre les droites orthogonales au point de début \mathcal{L}_i et au point de fin \mathcal{L}'_i du segment i (Fig. 2.3a) : la distance correspond à la distance orthogonale.
- (b) Le point p_e est entre la droite orthogonale de fin du segment précédent \mathcal{L}'_{i-1} et la droite orthogonale de début du segment \mathcal{L}_i à « l'extérieur » du virage (Fig. 2.3b) : la distance vaut $\|p_e - m_i\|$. Elle est la même pour tous les points de l'arc de cercle illustré sur la figure 2.3b.
- (c) Le point p_e est entre la droite orthogonale de fin du segment précédent \mathcal{L}'_{i-1} et la droite orthogonale de début du segment \mathcal{L}_i à « l'intérieur » du virage (Fig. 2.3c) : la distance correspond à la distance orthogonale comme pour le cas (a).

Par convention, le point p_h appartenant à la polyligne avec la plus courte distance est le point associé par *map-matching*. Le point p_h est donc ici soit la projection orthogonale dans le cas (a), soit le point m_i dans le cas (b). Dans le cas (c), le point p_h est projeté orthogonalement comme dans le cas (a). Dans le cas où le point p_e se trouve exactement sur la bissectrice à l'intérieur du virage comme illustré sur la figure 2.3c, deux projections apparaissent p_{h1} et p_{h2} avec des distances égales sur les deux segments successifs². Rappelons que le point p_h est utilisé pour calculer l'abscisse curviligne, qui est ici calculée comme la somme des longueurs de chaque segment du premier jusqu'à celui précédant le segment associé au *map-matching* et de la longueur $\|p_h - m_i\|$. L'ordonnée curviligne est l'écart latéral du point ayant la distance $\|p_e - p_h\|$ la plus courte lors du *map-matching*. Dans le cas (a) ou (c), cette ordonnée curviligne correspond au produit scalaire entre $p_e - p_h$ et la tangente unitaire $\frac{m_{i+1} - m_i}{\|m_{i+1} - m_i\|}$. Dans le cas (b) l'ordonnée curviligne correspond à la distance euclidienne $\|p_e - m_i\|$ multipliée par le signe du produit scalaire précédent. L'orientation curviligne est la différence entre l'orientation du véhicule et l'orientation du segment associé au *map-matching*.

Pour trouver une pose cartésienne à partir d'une pose curviligne définie le long d'une polyligne, il est possible de trouver le point p_h sur la polyligne à partir de l'abscisse curviligne et d'utiliser ensuite l'écart latéral pour déterminer la position

2. En pratique, la probabilité d'un point soit exactement sur la bissectrice est nulle.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

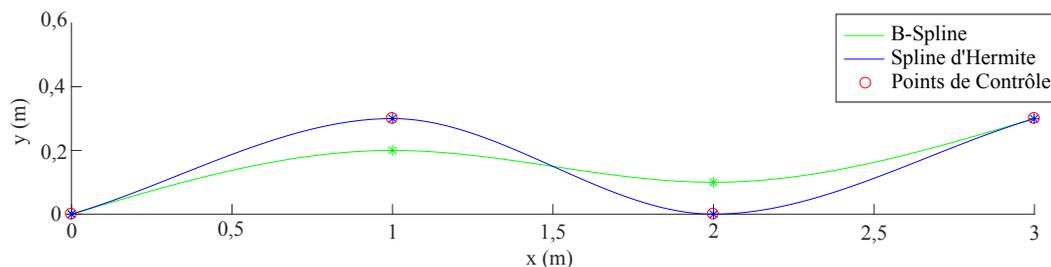


FIGURE 2.4. – *B-spline (en verte) et spline d’Hermite (en bleue) utilisant les mêmes points de contrôle (cercles rouges) donnés par la carte.*

cartésienne. Avec la tangente au point p_h , l’orientation dans le repère cartésien peut être déterminée. Cependant, on peut voir que dans le cas (b), la transformation n’est pas bijective. Un choix doit être fait. Le point choisi peut être le point p'_e ou n’importe quel autre point sur l’arc en pointillés de la figure 2.3b.

Les transformations entre une pose cartésienne et une pose curviligne définie long d’une polygline sont donc simples à réaliser. Cette méthode est donc facile à utiliser en temps réel.

Malheureusement, cette approche présente plusieurs inconvénients. D’abord, le retour vers des coordonnées cartésiennes peut ne pas être bijectif lorsque le véhicule s’approche d’un point m_i . Ensuite, la conversion en coordonnées curvilignes crée des déformations car la polygline n’est pas dérivable aux points de jonction des segments. Cette discontinuité des tangentes implique des problèmes de discontinuité proche des points de jonction. Dans le cas (b), l’abscisse curviligne reste constante pour tout point p_e entre les lignes \mathcal{L}'_{i-1} et \mathcal{L}_i , car la projection est identique. Dans le cas (c), deux projections apparaissent et la plus courte distance dépend de la position du point p_e par rapport à la bissectrice. Quand cette bissectrice est traversée lorsque le véhicule avance, un saut apparaît dans l’abscisse curviligne. A cause du changement de segment associé au *map-matching*, un saut apparaît aussi pour l’orientation curviligne du fait qu’elle est relative au segment. Dans ce cas, on observe que des morceaux de segment ne peuvent jamais être *matchés* à cause de ces sauts.

2.2.3. Représentation avec des splines

Une solution pour éviter les problèmes de discontinuité est d’utiliser une courbe plus régulière pour remplacer la polygline. Chaque segment peut être remplacé par un vecteur de deux polynômes représentant une courbe paramétrique et permettant d’obtenir une continuité au moins C^1 aux points de jonction. Nous avons utilisé des polynômes de degré 3 pour obtenir des splines cubiques.

Une spline peut être définie de la manière suivante :

$$\left\{ m_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}, i \in \{0, \dots, N\}, t \in [0; 1] \right\}, \quad (2.3)$$

où $x_i(t)$ et $y_i(t)$ sont deux polynômes paramétrés par t . Chaque point $m_i(t)$ correspond à un point de la courbe. La tangente à la courbe au point $m_i(t)$ est la dérivée $m'_i(t)$ et la courbure est la dérivée seconde $m''_i(t)$.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

Il est possible de trouver le point de la spline le plus proche du point $p_e = [x_e \ y_e]^T$ en cherchant les racines d'un polynôme. Tous les minimums et maximums locaux peuvent être trouvés lorsque $(p_e - m_i(t)) \perp m'_i(t)$, on peut donc résoudre l'équation :

$$(p_e - m_i(t)) \cdot m'_i(t) = 0. \quad (2.4)$$

Pour obtenir le résultat plus rapidement, on commence par chercher le plus proche vecteur de polynômes correspondant à un indice i_{min} , pour lequel on cherchera ensuite les racines du vecteur de polynôme correspondant. Pour trouver i_{min} , on calcule la distance à chaque point de jonction $m_i = m_i(0)$. Quand la distance minimale est trouvée pour un indice i , $i_{min} = i$ si la projection de p_e sur la tangente $m'_i = m'_i(0)$ au point de jonction est positive et $i_{min} = i - 1$ si cette projection est négative.

Le produit scalaire est réalisé (Eq. 2.4) entre $p_e - m_i(t)$, un vecteur de polynômes de degré 3 et $m'_i(t)$, un vecteur de polynômes de degré 2. Il correspond donc à un polynôme de degré 5. Il existe alors, au maximum, cinq racines. Les racines complexes et celles qui ne sont pas comprises dans $[0, 1]$ peuvent être éliminées. On peut alors déterminer t_{min} tel que $\|p_e - m_{i_{min}}(t_{min})\|$ est la plus courte distance. Cette distance est l'ordonnée curviligne n_e , un signe négatif est ajouté lorsque p_e est à droite de la tangente $m'_{i_{min}}(t_{min})$.

Pour calculer l'abscisse curviligne s_e , on doit intégrer $m'_{i_{min}}(t)$:

$$s_e = \int_0^{t_{min}} \|m'_{i_{min}}(t)\| dt + \sum_{j=0}^{i_{min}-1} \left(\int_0^1 \|m'_j(t)\| dt \right). \quad (2.5)$$

La norme d'un vecteur de deux polynômes de degré 2 est la racine carrée d'un polynôme de degré 4. Il n'existe pas de primitive de cette fonction calculable analytiquement. L'intégration doit donc être faite numériquement. Afin de calculer plus rapidement la solution, la longueur de chaque courbe déterminée par un vecteur de polynômes peut être calculée à l'avance. Seul le premier terme de l'équation 2.5 reste alors à calculer.

Réciproquement, pour déterminer une pose cartésienne à partir d'une pose curviligne, il est nécessaire d'inverser l'équation 2.5, ce qui revient à calculer t_{min} à partir d'une abscisse curviligne s_e . Afin de résoudre ce problème, des approches numériques doivent, là aussi, être utilisées. Une autre solution est d'approximer l'abscisse curviligne et son inverse par un polynôme ou une autre fonction [Wang *et al.*, 2002].

Nous avons étudié deux splines différentes. La B-spline (en verte sur la figure 2.4), qui est une approximation de la carte³, mais qui donne une trajectoire à suivre plus régulière, avec moins d'oscillations et la spline d'Hermite (en bleue sur la figure 2.4) qui est une interpolation du centre de la voie et donc une représentation plus fidèle de celui-ci puisqu'elle passe par les points donnés par la carte. Cette dernière implique un contrôle toutefois moins confortable si les points de contrôle sont mal positionnés. En effet, si les points enregistrés sont bruités et proches les uns des autres, la spline peut avoir de nombreux points d'inflexion. Elle ne sera pas lissée comme pourrait l'être la B-spline. On remarque, en effet, sur la figure 2.4 que la B-spline est moins contrainte que la spline d'Hermite.

3. dans le sens où les points de jonction cartographiés qui appartiennent réellement au centre de la voie ne sont plus présents dans la nouvelle représentation

2.2.3.1. B-spline

La B-spline est une approximation de la carte et ne passe pas par tous les points de la carte $\{m_i, i \in \{1, \dots, N\}\}$. Le vecteur de polynômes est déterminé par l'équation suivante [de Boor, 2001] :

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \frac{1}{6} (m_{i-1} \ m_i \ m_{i+1} \ m_{i+2}) \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}. \quad (2.6)$$

L'avantage d'utiliser une approximation est d'obtenir une courbe plus régulière et plus confortable pour le contrôle du véhicule. Cependant, cette approximation peut poser des problèmes. La spline peut, par exemple, couper les virages au niveau des points de jonction, comme la figure 2.4 le montre. On peut en effet voir que la courbe verte coupe les virages définis par les points de contrôle rouges.

2.2.3.2. Spline d'Hermite

Dans le cas où il est nécessaire d'utiliser une courbe passant par tous les points de la carte, une interpolation est plus appropriée. Une spline d'Hermite peut alors être utilisée. Elle est définie par l'équation suivante [Lalescu, 2009] :

$$\begin{pmatrix} x_i(t) \\ x_i(t) \end{pmatrix} = \begin{pmatrix} m_i & \vec{t}_i & m_{i+1} & \vec{t}_{i+1} \end{pmatrix} \begin{pmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} t^3 \\ t^2 \\ t \\ 1 \end{pmatrix}, \quad (2.7)$$

où $\{\vec{t}_i, i \in \{1, \dots, N-1\}\}$ sont les tangentes aux points de contrôle. Elles peuvent être calculées de la manière suivante :

$$\vec{t}_i = (m_{i+1} - m_{i-1}) / 2. \quad (2.8)$$

La première et la dernière tangentes peuvent être prises parallèles au segment associé :

$$\vec{t}_0 = (m_1 - m_0) / 2, \quad (2.9)$$

$$\vec{t}_N = (m_N - m_{N-1}) / 2. \quad (2.10)$$

En pratique, le premier et le dernier point d'une polyligne correspondent à l'extrémité d'une impasse à sens unique. Il est donc rare d'en avoir besoin.

La spline d'Hermite est d'un certain point de vue plus sûre, puisqu'elle passe par les points de contrôle, la carte ne coupera donc pas les virages. Cependant, ces points de passage doivent être correctement choisis afin d'éviter des oscillations trop importantes. La carte haute définition utilisée possède des points suffisamment espacés et peu bruités pour éviter des trop fortes oscillations.

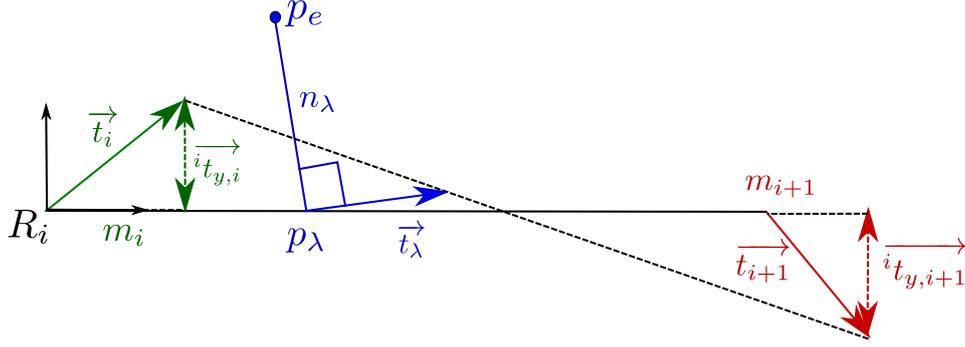


FIGURE 2.5. – *Map-matching* d'un point p_e sur un segment i avec une projection des lanelets.

2.2.4. Projection sur les polygones selon la méthode des lanelets

Une autre solution pour éviter les problèmes de discontinuité aux voisinages des points de jonction lorsqu'on utilise des polygones est d'utiliser la projection des lanelets. La notion de lanelet a été récemment proposée dans [Bender *et al.*, 2014, Poggenhans *et al.*, 2018] et a montré de bonnes propriétés pour la planification de trajectoires à partir de cartes haute définition.

Dans ce cas, le point associé au *map-matching* n'est plus le plus proche mais est déterminé à partir d'une combinaison convexe du point de début et du point de fin du segment associé au *map-matching* :

$$p_\lambda = \lambda m_{i+1} + (1 - \lambda) m_i, \quad (2.11)$$

où λ est un réel compris entre 0 et 1.

Les segments *matchés* sont les mêmes que ceux de la polygone.

De la même façon, on peut déterminer la tangente au sens des lanelets à partir d'une combinaison linéaire des tangentes \vec{t}_i et \vec{t}_{i+1} aux points de début et de fin de chaque segment de la polygone :

$$\vec{t}_\lambda = \lambda \vec{t}_{i+1} + (1 - \lambda) \vec{t}_i. \quad (2.12)$$

Ces deux équations étant des combinaisons linéaires, elles sont vraies pour n'importe quel repère orthonormé. Le coefficient λ est indépendant de ce repère.

Afin de calculer les repères des lanelets, le vecteur tangent \vec{t}_λ et le vecteur normal \vec{n}_λ doivent être orthogonaux,

$$\vec{n}_\lambda^T \vec{t}_\lambda = 0. \quad (2.13)$$

Ils ne sont toutefois pas unitaires. On définit les tangentes aux points extrêmes du segment de la même manière que pour les splines d'Hermite et en ajoutant les contraintes $\vec{t}_i = (1 \ i t_{i,y})^T$ et $\vec{t}_{i+1} = (1 \ i t_{i+1,y})^T$ dans le repère local du $i^{\text{ème}}$ segment R_i ⁴. On a alors :

$$\vec{t}_\lambda = \begin{pmatrix} i t_{x,\lambda} \\ i t_{y,\lambda} \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ i t_{i+1,y} \end{pmatrix} + (1 - \lambda) \begin{pmatrix} 1 \\ i t_{i,y} \end{pmatrix}. \quad (2.14)$$

4. Une singularité pourrait apparaître dans le cas où l'angle $m_{i+1} \widehat{m_i} m_{i-1}$ est nul. Cela revient à faire demi-tour sur le point m_i . En pratique des points auront été ajoutés pour éviter un demi-tour aussi brusque. Les angles $m_{i+1} \widehat{m_i} m_{i-1}$ sont donc toujours proches de 180° en pratique.

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

De plus

$$\begin{aligned}\vec{n}_\lambda &= \begin{pmatrix} {}^i n_{x,\lambda} \\ {}^i n_{y,\lambda} \end{pmatrix} = {}^i p_e - {}^i p_\lambda \\ &= {}^i p_e - (\lambda {}^i m_{i+1} + (1-\lambda) {}^i m_i) \\ &= \begin{pmatrix} {}^i x_e \\ {}^i y_e \end{pmatrix} - \left(\lambda \begin{pmatrix} l_i \\ 0 \end{pmatrix} + (1-\lambda) \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right),\end{aligned}\quad (2.15)$$

où l_i est la longueur du segment i . Comme

$$\vec{n}_\lambda^T \vec{t}_\lambda = 0, \quad (2.16)$$

on a donc

$$({}^i x_e - \lambda l_i) + (\lambda {}^i t_{i+1,y} + (1-\lambda) {}^i t_{i,y}) {}^i y_e = 0, \quad (2.17)$$

$${}^i x_e + {}^i y_e {}^i t_{i,y} + \lambda (-l_i + {}^i y_e ({}^i t_{i+1,y} - {}^i t_{i,y})) = 0. \quad (2.18)$$

À partir de cette contrainte, λ peut être calculé :

$$\lambda = \frac{{}^i x_e + {}^i y_e {}^i t_{i,y}}{l_i - {}^i y_e ({}^i t_{i+1,y} - {}^i t_{i,y})}. \quad (2.19)$$

Une fois le projeté p_λ calculé grâce à l'équation (2.11), on peut déterminer la tangente \vec{t}_λ à partir de l'équation (2.12). Enfin, on peut déterminer le vecteur normal non unitaire $\vec{n}_\lambda = p_e - p_\lambda$. La distance du point p_e au $i^{\text{ème}}$ segment est alors la norme $\|\vec{n}_\lambda\|$. Elle peut être utilisée pour déterminer le segment le plus proche selon la projection des *lanelets*.

On peut déterminer la position curviligne : $s_e = \sum_{k=1}^{i-1} l_k + \lambda l_i$ et $n_e = n_\lambda$, où l_k est la longueur du $k^{\text{ème}}$ segment et n_λ est la seconde composante du vecteur $\vec{n}_\lambda = (0 \ n_\lambda)^T$ dans le repère local (la première composante étant nulle).

Réciproquement, pour déterminer la position cartésienne à partir de la position curviligne, on peut déterminer λ avec l'équation :

$$\lambda = \frac{s_e - \sum_{k=1}^{i-1} l_k}{l_i}, \quad (2.20)$$

où i est déterminé par soustraction itérative de chaque longueur l_k à s_e jusqu'à obtenir un résultat négatif ou nul. À partir de λ et i , on peut alors déterminer p_λ et \vec{t}_λ . On a alors

$$p_e = p_\lambda + R \cdot \vec{t}_\lambda / \|\vec{t}_\lambda\| \cdot n_e, \quad (2.21)$$

avec $R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ la matrice de rotation de $\frac{\pi}{2}$ permettant d'obtenir la normale unitaire au vecteur $\vec{t}_\lambda / \|\vec{t}_\lambda\|$.

Il est à noter que cette transformation de type *lanelet* est bijective et est facilement calculable en temps réel.

2.3. Résultats

Dans cette section, nous comparons les différents modèles de route. Un cas simple est d'abord présenté afin d'illustrer les discontinuités et les non-linéarités. Des données réelles enregistrées sur la route expérimentale Séville sont utilisées pour illustrer un scénario de conduite en convoi. Nous comparons ensuite la localisation le long et latéralement à la voie pour les quatre différents modèles.

2.3.1. Discontinuité et non-linéarité

Considérons tout d'abord un cas simple où un véhicule se déplace à vitesse constante le long d'une ligne droite. Dans la figure 2.6 (a), les pointillés représentent la trajectoire du véhicule. Le but est de calculer les coordonnées curvilignes du véhicule par rapport à la route qui est représentée par quatre points de contrôle, illustrés par des cercles rouges dans la figure 2.6 (a). Le modèle de route de la polyligne et des *lanelets* est représenté par les segments noirs et rouges entre les points de contrôle. Pour les modèles de splines, la courbe passe par tous les points de contrôle pour la spline d'Hermite (en bleu) et coupe les angles sans passer par les points de contrôle pour la B-spline (en vert).

2.3.1.1. Discontinuité et non linéarité

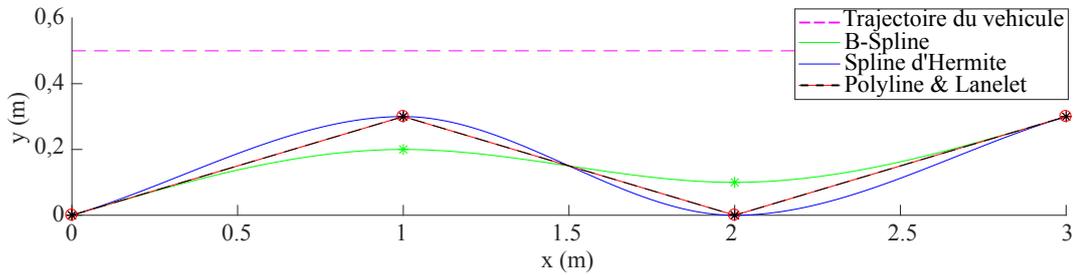
Dans la figure 2.6 (b), on peut voir l'évolution de l'abscisse curviligne pour les quatre modèles. Pour la méthode des polygones (courbe rouge), un point stationnaire apparaît au niveau du second point de contrôle, autour d'un mètre. L'abscisse curviligne reste constante alors que le véhicule se déplace le long de l'axe x . Au niveau du troisième point de contrôle, autour de deux mètres, une discontinuité de l'abscisse curviligne apparaît cette fois-ci. Pour la méthode des *lanelets* (courbe noire), il n'y a aucun problème de discontinuité de l'abscisse curviligne. Cependant, contrairement à la méthode des polygones, la courbe entre deux points de contrôle n'est pas linéaire. Cela signifie que malgré le fait que le véhicule se déplace à vitesse constante, la dérivée de l'abscisse curviligne n'est pas constante. De plus, la dérivée de l'abscisse curviligne au niveau des points de jonction n'est pas continue. Des accélérations infinies apparaissent donc au niveau des points de jonction. Le même phénomène apparaît pour les polygones.

Les problèmes de continuité de l'abscisse curviligne et de ses dérivées peuvent être évités en utilisant des modèles de splines. L'abscisse curviligne est continue et dérivable pour les splines d'Hermite (courbe bleue) et les B-splines (courbe verte). Ces deux modèles présentent cependant des non-linéarités encore plus importantes que pour les *lanelets*, ceci est particulièrement visible sur la courbe de la spline d'Hermite.

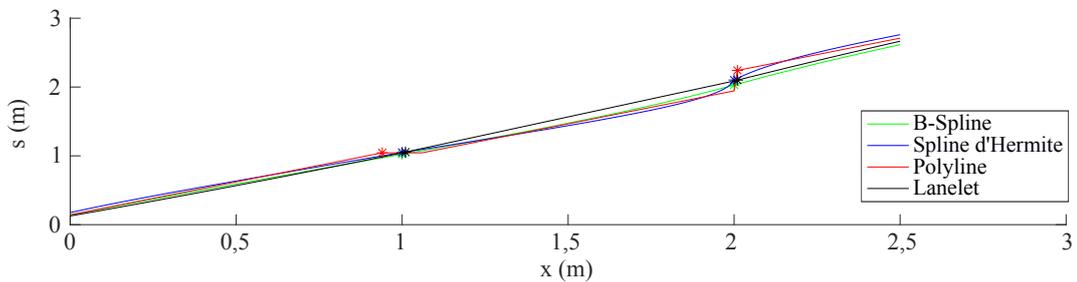
Dans la figure 2.6 (c), on peut voir que l'ordonnée curviligne est toujours continue pour les quatre méthodes. Cependant, les méthodes des splines donnent des ordonnées plus régulières au niveau des points de jonction. Des cassures apparaissent avec les méthodes des polygones et des *lanelets*. Les dérivées des ordonnées curvilignes de ces deux méthodes ne sont donc pas continues au niveau des points de jonction.

Dans la figure 2.6 (d), on peut voir que l'orientation curviligne calculée avec la polyligne présente des sauts lorsque le segment associé au *map-matching* change.

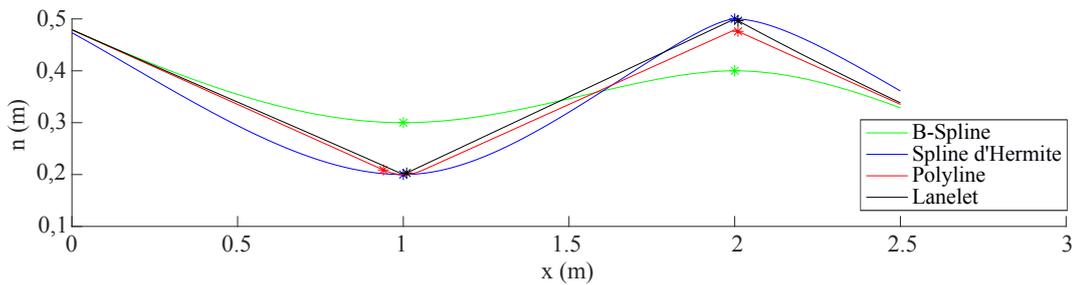
2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome



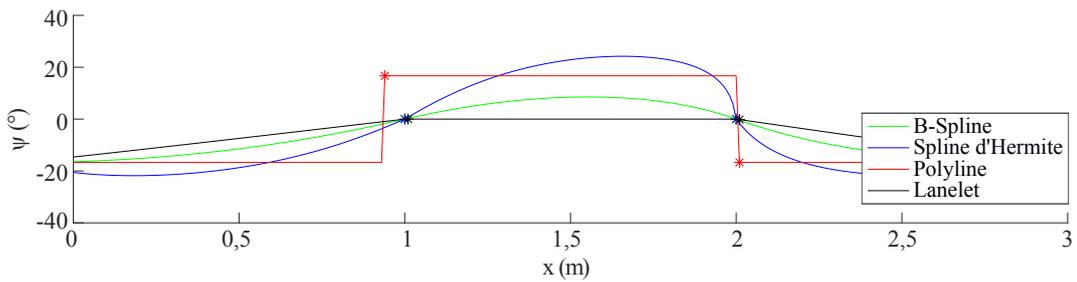
(a) Trajectoire du véhicule (en magenta pointillé), B-Spline (en vert), Hermite spline (en bleu) et polyligne utilisable aussi pour la méthode des lanelets (en noir et en rouge).



(b) Abscisse curviligne.



(c) Ordonnée curviligne.



(d) Orientation curviligne.

FIGURE 2.6. – Problèmes de discontinuités de la pose curviligne d'un véhicule utilisant une B-Spline (en vert), une Hermite spline (en bleu), une polyligne (en rouge) ou des lanelets (en noir).

TABLE 2.2. – Propriétés des différents modèles de route.

	Polyligne	Lanelets	Splines
s continuité	X	✓	✓
\dot{s} continuité	X	X	✓
s linéarité	✓	X	X
n continuité	✓	✓	✓
\dot{n} continuité	X	X	✓
n linéarité	✓	X	X
ψ continuité	X	✓	✓
$\dot{\psi}$ continuité	X	X	✓
ψ linéarité	✓	X	X
Temps réel	✓	✓	X

L'orientation curviligne provenant des *lanelets* est continue, mais des cassures apparaissent au niveau des points de jonction et, comme pour l'ordonnée curviligne, la dérivée n'est pas continue. Les orientations curvilignes pour les splines sont toujours très régulières avec une continuité au moins C^2 (au niveau des points de jonction).

Les différentes propriétés des modèles de route sont résumées dans la table 2.2.

2.3.1.2. Discontinuité de *map-matching*

Dans l'étude de cas précédente, nous avons montré que les coordonnées curvilignes étaient continues pour les méthodes des *lanelets* et des splines. Cependant, ces résultats sont vrais si et seulement si le *map-matching* est lui-même continu, le véhicule étant associé aux segments suivant le même ordre que celui de la carte. Dans le cas où le véhicule est loin de la polyligne de la route, le *map-matching* peut devenir discontinu, ce qui va entraîner des discontinuités de l'abscisse curviligne.

La figure 2.7 montre les zones de *map-matching* pour chaque segment pour les méthodes des *lanelets* et des splines. Comme les tangentes sont les mêmes pour ces trois modèles, les limites entre les différentes zones sont aussi identiques. On peut voir qu'un véhicule qui roulerait dans la partie haute, suffisamment éloigné de la carte, pourrait passer de la zone rouge à la zone bleue sans passer par la zone verte. Cela entraînerait alors des discontinuités importantes dans les coordonnées curvilignes.

Cependant, dans le contexte de véhicules autonomes, on peut raisonnablement supposer que le véhicule reste relativement proche de la polyligne car il doit naviguer sur la voie. Dans ce cas, les coordonnées curvilignes calculées à partir des *lanelets* ou des splines restent continues. En effet, les zones de discontinuité apparaissent loin de la polyligne ou lorsque la courbure est très importante.

2.3.2. Conduite en convoi basée sur l'abscisse curviligne

L'abscisse curviligne peut être utilisée pour la conduite coopérative de véhicules en convoi. L'utilisation de l'abscisse curviligne permet de prendre directement en compte la forme de la route. Dans le cas de scénarios complexes, comme ceux de conduite en convoi, de croisement ou de rond-point, l'utilisation d'un capteur de perception, comme une caméra ou un LiDAR donne une mesure de distance entre

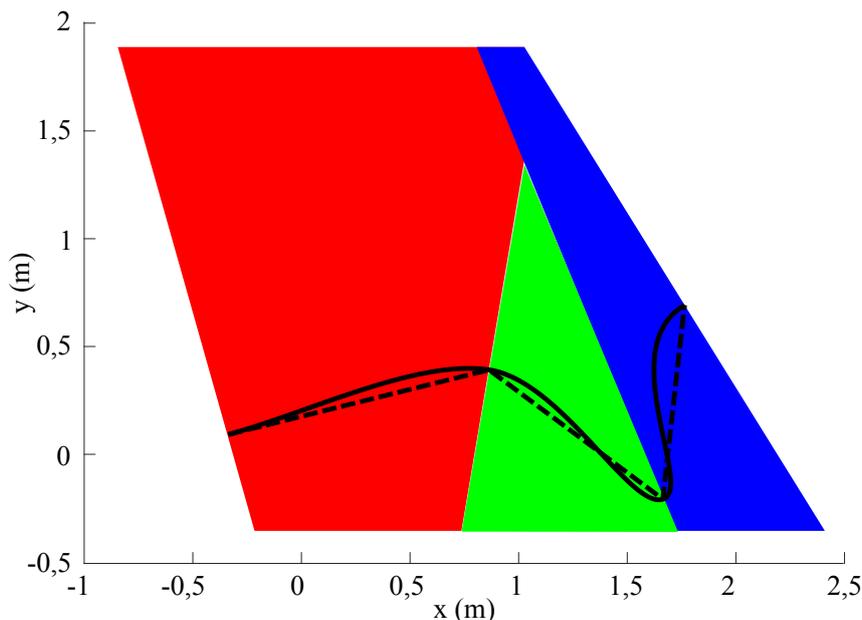


FIGURE 2.7. – *Discontinuité de map-matching.* Les zones rouges, vertes et bleues correspondent aux zones géographiques où le map-matching est fait, respectivement, sur le premier, le second et le troisième segment.

deux véhicules qui n'est pas directement utilisable. Par exemple, si deux véhicules sont à l'opposé l'un de l'autre à l'approche d'un rond-point, le capteur de perception va mesurer le diamètre du rond-point alors que la différence des deux abscisses curvilignes est la demie-circonférence. L'interdistance curviligne correspond à la différence entre l'abscisse curviligne du véhicule leader et celle du véhicule suiveur. Une conduite en convoi virtuel permet de négocier une intersection (comme nous l'avons vu lors du challenge GCDC). L'interdistance virtuelle est dans ce cas calculée en faisant la différence des distances curvilignes entre chaque véhicule et l'intersection. Le signe de l'interdistance virtuelle permet de savoir quel véhicule est le plus proche de l'intersection. De plus, le signe de la distance curviligne entre un véhicule et une intersection permet de savoir facilement si ce véhicule est à l'approche de l'intersection ou bien s'il en sort. De la même façon, le signe d'une interdistance dans un convoi classique permet de savoir lequel des deux véhicules est devant.

Afin de tester le scénario de conduite en convoi, nous avons enregistré la trajectoire d'une voiture Renault Zoe expérimentale se déplaçant sur la piste Séville. Le véhicule était équipé d'un récepteur GNSS hybridé avec une centrale inertielle (NovAtel SPAN-CPT) utilisant des corrections RTK afin d'obtenir une pose précise. La trajectoire enregistrée a ensuite été échantillonnée et interpolée afin de simuler un déplacement à vitesse constante. Cette trajectoire a été enfin décalée temporellement afin d'obtenir deux véhicules se déplaçant à la même vitesse constante et donc avec une interdistance constante. Une interdistance de 10 m a été choisie.

La figure 2.9 montre la différence des abscisses curvilignes entre les deux véhicules en convoi. En théorie, comme les deux véhicules suivent exactement la même trajectoire et se déplacent à une même vitesse constante, la distance curviligne entre eux devrait rester constante et égale à 10 m. Cela serait vrai si les deux véhicules

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

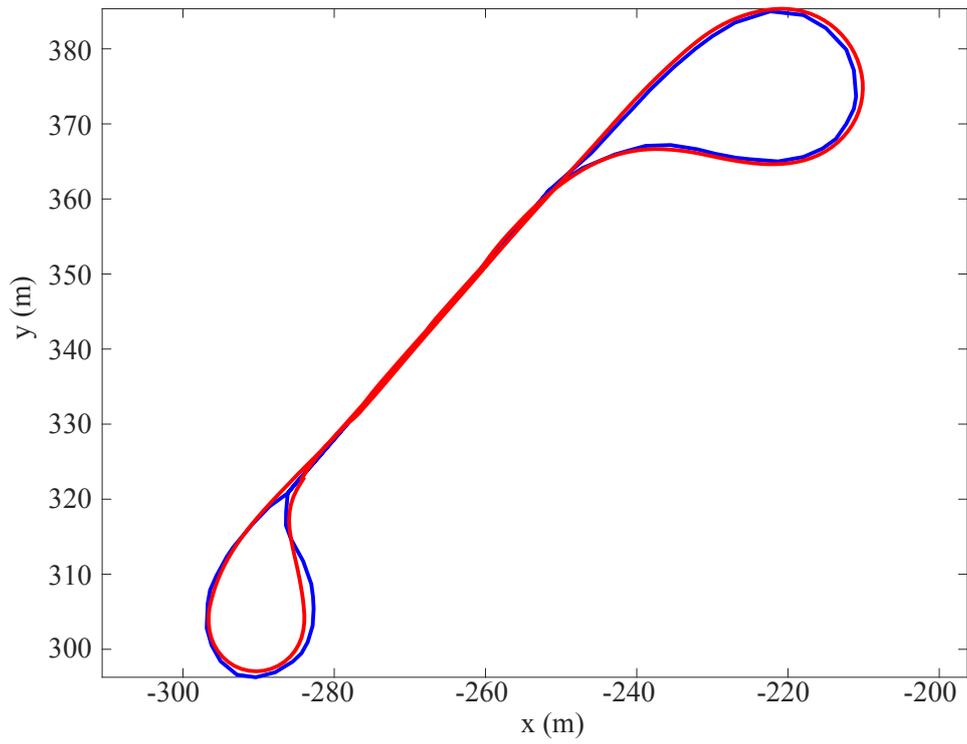


FIGURE 2.8. – Carte de Séville (en bleue) avec la trajectoire des véhicules (en rouge).

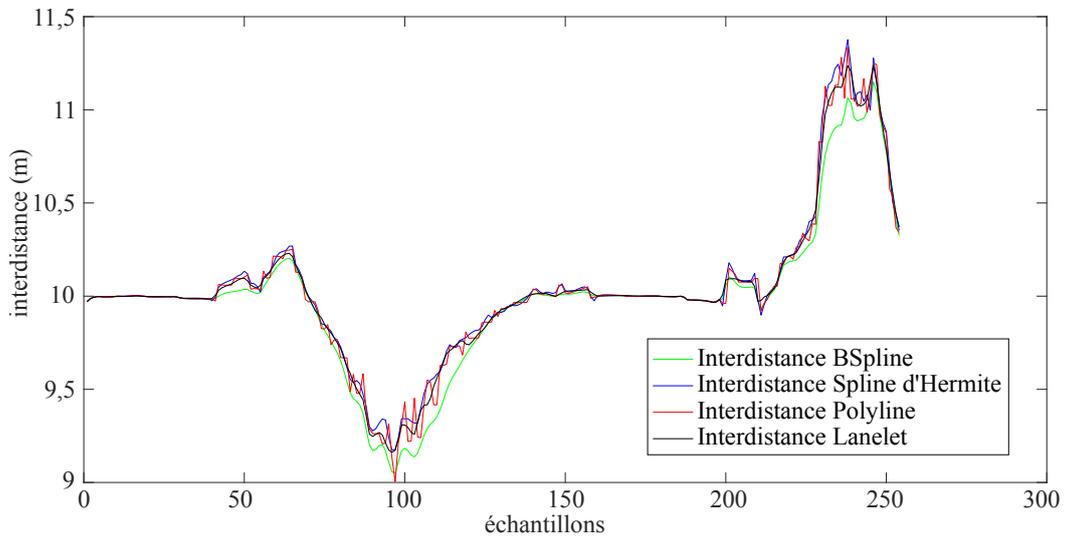


FIGURE 2.9. – Différence des abscisses curvilignes des deux véhicules avec une interdistance de 10 m.

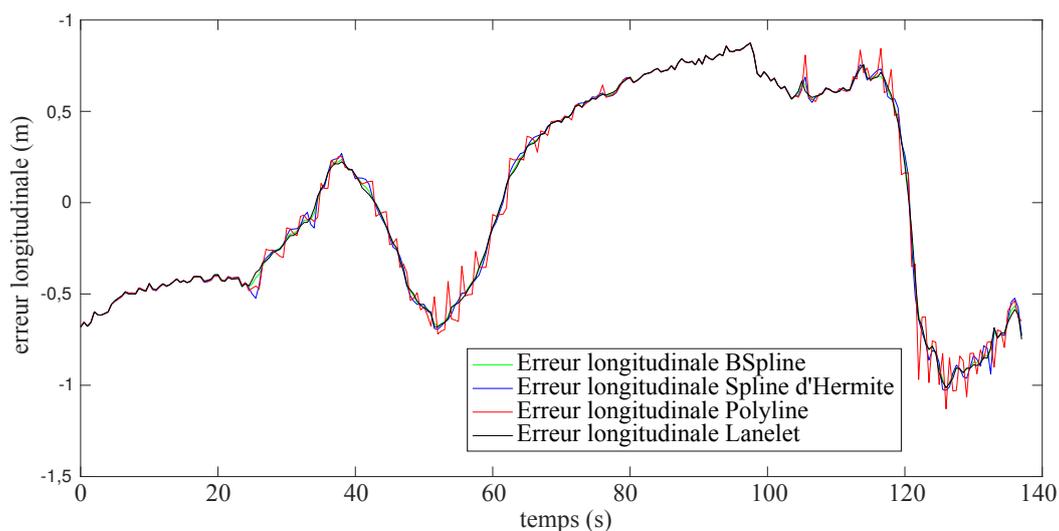


FIGURE 2.10. – Erreurs longitudinales.

s'étaient déplacés exactement le long de la carte de référence (le centre de la voie). Ainsi, la distance parcourue serait identique à la différence d'abscisse curviligne. En pratique, ce n'est pas le cas et la figure 2.8 montre que les véhicules se déplacent à l'intérieur du rond-point du bas et à l'extérieur de celui du haut. De ce fait, la distance curviligne devient plus longue dans un des ronds-points (pour les échantillons 40 à 140) et plus courte dans l'autre (pour les échantillons 200 à 250). Dans le cas d'un contrôle longitudinal, si le but était de conserver cette distance curviligne constante, le véhicule suiveur aurait accéléré dans l'un des ronds-points et décéléré dans le second.

Lorsque la polyligne (courbe rouge) est utilisée, on peut voir de nombreuses discontinuités qui pourraient engendrer un contrôle inconfortable. Ces discontinuités disparaissent pour les trois autres méthodes. Il est intéressant de remarquer que la distance provenant des *lanelets* (courbe noire) est constamment entre la distance provenant de la spline d'Hermite (courbe verte) et celle provenant de la B-spline (courbe bleue). En effet, la B-spline est une approximation de la polyligne et est donc plus courte que cette polyligne, alors que la spline d'Hermite est une interpolation et est donc plus longue que la polyligne. La B-spline étant la courbe la plus régulière, elle est la méthode donnant le contrôle le plus confortable.

2.3.3. Erreurs longitudinales et latérales par rapport à la voie

Pour la navigation autonome, la localisation doit souvent être évaluée à partir des erreurs longitudinales (le long de la voie) et latérales (orthogonales à la voie) ce qui permet une analyse plus fine que l'utilisation d'erreurs euclidiennes. Cependant, le calcul de ces erreurs dépend de la façon dont les coordonnées curvilignes sont calculées. Dans la précédente base de données, nous avons aussi enregistré les données d'un récepteur GNSS bas coût, mono-fréquence (un u-blox 8T).

La figure 2.10 montre les erreurs longitudinales et la figure 2.11 montre les erreurs latérales calculées pour les différents modèles de route. On peut voir les problèmes

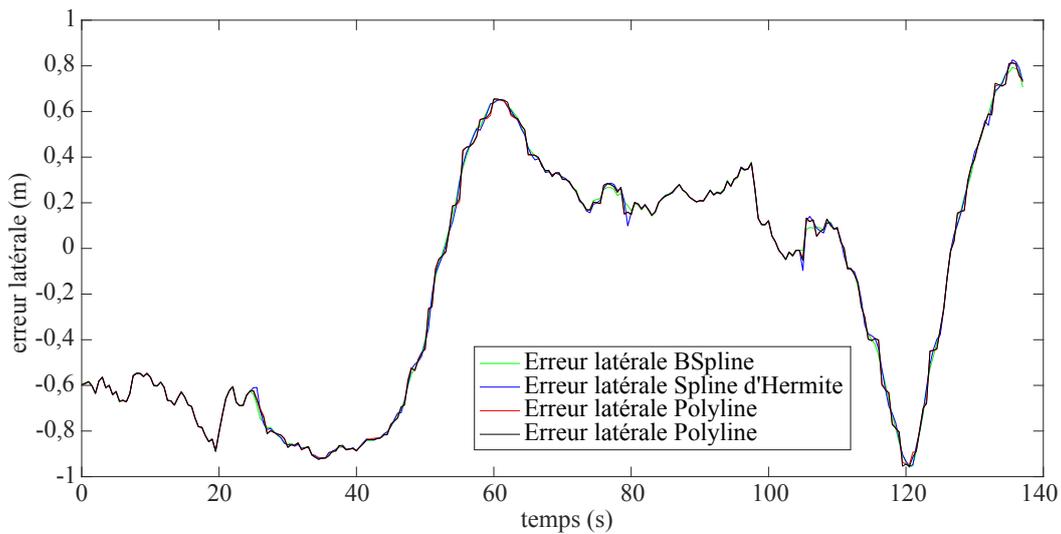


FIGURE 2.11. – *Erreurs latérales.*

TABLE 2.3. – *Erreurs quadratiques moyennes longitudinales et latérales.*

	Polyligne	Lanelet	Hermite	B-spline
Erreurs longitudinales (m)	0.5745	0.5704	0.5733	0.5692
Erreurs latérales (m)	0.5512	0.5499	0.5501	0.5512

de discontinuité des erreurs longitudinales pour le modèle de type polyligne. Aucune discontinuité liée aux erreurs latérales n'est présente. Ceci indique que le modèle polyligne est le moins adapté au contrôle longitudinal.

La table 2.3 montre les moyennes quadratiques des deux erreurs pour les différents modèles de route. On peut voir que ces erreurs moyennes sont très proches comme ce qui peut être vu sur la figure 2.10 et la figure 2.11. Comme le récepteur u-blox ne donne pas d'orientation suffisamment précise, les erreurs relatives à l'orientation curviligne n'ont pas été étudiées.

2.4. Conclusion sur les différents modèles de route pour la navigation

Différents modèles de route basés sur des cartes haute définition ont été présentés. Ces modèles permettent de calculer différentes poses curvilignes, composées de l'abscisse curviligne, l'écart latéral par rapport au centre de la voie et d'une orientation relative à la voie. La pose curviligne est une représentation très intéressante pour le suivi de chemin et pour la navigation coopérative de véhicules intelligents autonomes.

Les différentes représentations de poses curvilignes correspondant à différents modèles de route ont été testées et comparées en utilisant des données réelles et des données simulées. La polyligne étant constituée d'une suite de segments, c'est une méthode simple pour calculer des coordonnées curvilignes. Néanmoins, cette méthode présente plusieurs problèmes de discontinuité et de dérivabilité au niveau des

2. Coordonnées curvilignes sur des cartes haute définition pour la navigation autonome

points de jonction entre les segments. Cependant, c'est la seule méthode parmi les quatre étudiées qui ne présente que des transformations linéaires. Cela peut être intéressant dans le cas où un modèle d'évolution doit être dérivé, comme pour un modèle à vitesse constante. La spline d'Hermite et la B-spline permettent d'obtenir des coordonnées curvilignes régulières, continues et dérivables, mais sont coûteuses en temps de calcul et ne sont donc pas particulièrement adaptées à des applications temps réel dans des systèmes embarqués. Les *lanelets* présentent un bon compromis entre la polyligne et la spline. Les coordonnées curvilignes sont continues, mais ne sont pas dérivables. La non-linéarité lors de l'utilisation des *lanelets* n'est pas aussi importante que lors de l'utilisation d'une spline et les transformations en coordonnées curvilignes peuvent être faites en temps réel facilement. Il est donc recommandé d'utiliser ce modèle de route.

3. Localisation coopérative à une dimension en abscisse curviligne

3.1. Introduction

Un véhicule intelligent autonome ne possédant pas de récepteur GNSS de précision centimétrique est contraint de se baser sur une observation de son environnement pour pouvoir naviguer. Une perception des bords de voie ou des marquages au sol avec une caméra ou bien un LiDAR lui permet assez facilement de connaître son ordonnée curviligne, c'est-à-dire l'écart latéral signé par rapport au centre de la voie, et son orientation curviligne, c'est-à-dire l'orientation du véhicule relative à cette voie. Grâce à des mesures latérales et angulaires suffisamment précises, le véhicule peut maîtriser son comportement latéral en suivant par exemple une consigne de distance au bord ou bien en maintenant sa position dans la voie. De nombreux dispositifs d'aide à la conduite de maintien de trajectoire sont déjà commercialisés. Néanmoins et malgré l'utilisation de capteurs extéroceptifs embarqués, le positionnement longitudinal reste assez imprécis et de l'ordre de grandeur de la précision du récepteur GNSS bas-coût utilisé dans le véhicule. Disposer d'un positionnement longitudinal précis est essentiel pour beaucoup de tâches de navigation comme la conduite en convoi ou encore le franchissement d'intersections. Une perception basée sur des mesures LiDAR par exemple permet à un véhicule de mesurer précisément dans son repère la pose relative d'un autre véhicule perçu.

Ce chapitre s'intéresse donc à une première étude de systèmes de localisation coopératifs dans un cadre simplifié. On suppose en effet que l'écart latéral, l'orientation relative à la voie ainsi que la pose relative sont connues avec des erreurs négligeables par rapport à l'erreur longitudinale. On se concentre ainsi sur l'étude d'une localisation coopérative n'utilisant que l'abscisse curviligne le long d'une courbe de centre de voie contenue dans une carte haute définition.

Dans la section 3.2, les étapes de prédiction et de mise à jour de la fusion de données sont présentées. La problématique et l'algorithme de fusion de données à une dimension en abscisse curviligne sont ensuite introduits dans la section 3.3. Finalement, des résultats de simulation et des résultats obtenus à partir de données réelles sont présentés dans la section 3.4.

3.2. Mécanismes de base en fusion de données séquentielles

En fusion de données séquentielles, les mécanismes de base reposent sur des étapes de prédiction et de mise à jour alternées.

3.2.1. Prédiction

La première information utilisable pour la localisation est la vitesse du véhicule donnée par les capteurs proprioceptifs mesurant la vitesse des roues. Cette information est accessible par le bus CAN du véhicule.

Pour obtenir la vitesse du véhicule le long de la route en lien avec les coordonnées curvilignes, on projette la vitesse du véhicule sur le segment associé au *map-matching*. On utilise cette vitesse pour calculer une localisation à l'estime qui servira dans les étapes de prédiction d'un estimateur d'état de type filtre de Kalman :

$$\hat{s}_e [k|k-1] = \hat{s}_e [k-1|k-1] + \Delta t v_e [k] \cos(\hat{\psi}_e [k-1|k-1]), \quad (3.1)$$

$$\hat{\sigma}_{s_e}^2 [k|k-1] = \hat{\sigma}_{s_e}^2 [k-1|k-1] + \nu_v, \quad (3.2)$$

où $\hat{\sigma}_{s_e}$ est l'écart type de l'erreur d'estimation de l'abscisse curviligne \hat{s}_e , Δt la période d'échantillonnage entre deux itérations k et $k-1$, ν_v la variance du modèle de prédiction (elle inclut les erreurs de mesure de vitesse et les erreurs de projection), $\hat{\psi}_e$ l'orientation curviligne et v_e la vitesse longitudinale mesurée du véhicule. La notation $X [k_2|k_1]$ signifie l'élément X à l'itération k_2 calculé à partir des informations de l'itération k_1 .

3.2.2. Mise à jour de Kalman

On suppose disposer d'une seconde information comme une position GNSS. Cette information est prise en compte en utilisant une mise à jour de Kalman qui suppose que l'erreur GNSS est décorrélée de l'erreur de prédiction :

$$K = \hat{\sigma}_{s_e}^2 [k|k-1] / (\hat{\sigma}_{s_e}^2 [k|k-1] + \sigma_{gnss s_e}^2 [k]), \quad (3.3)$$

$$\hat{s}_e [k|k] = \hat{s}_e [k|k-1] + K (gnss s_e [k] - \hat{s}_e [k|k-1]), \quad (3.4)$$

$$\hat{\sigma}_{s_e}^2 [k|k] = \hat{\sigma}_{s_e}^2 [k|k-1] (1-K)^2 + \sigma_{gnss s_e}^2 [k] K^2, \quad (3.5)$$

où $gnss s_e$ est l'abscisse curviligne de l'observation GNSS, $\sigma_{gnss s_e}$ son écart type et K le gain de Kalman.

3.2.3. Mise à jour par intersection de covariance

Pour être optimal, le filtre de Kalman doit être utilisé avec des bruits blancs (décorrélés dans le temps) et, dans sa forme la plus simple, l'étape de mise à jour suppose que les bruits d'état et d'observation sont décorrélés entre eux. Dans le cas où les mêmes informations se retrouvent être fusionnées à nouveau, à cause de cycles et d'échanges dans les étapes de fusion, un phénomène de consanguinité de données apparaît : les erreurs se corrélaient au fur et à mesure des étapes de fusion et le résultat d'estimation devient trop confiant, c'est à dire que la variance estimée est inférieure à la vraie variance. Afin d'éviter ce problème, une intersection de covariance peut être utilisée à la place de la mise à jour du filtre de Kalman. Ce filtre présenté par [Julier et Uhlmann, 1997b] est plus pessimiste, mais reste consistant, la variance estimée étant supérieure ou égale à la vraie variance.

3. Localisation coopérative à une dimension en abscisse curviligne

Dans cette thèse, on dira qu'un estimateur est consistant¹ si les intervalles de confiance associés à un risque α ont un taux de couverture, i.e. une probabilité de contenir la vraie valeur, supérieur ou égal à $1 - \alpha$. Si l'estimateur dispose des bons modèles de bruit et si les paramètres du modèles sont exacts, alors le taux de couverture est égal à $1 - \alpha$. Dans le cas gaussien, cela est équivalent à avoir une estimation de la variance supérieure ou égale à la vraie variance. En pratique, la consistance dépend de la valeur de α . En d'autres termes, un estimateur peut être empiriquement consistant à 95% mais pas à 99%. Dans ces travaux, nous utiliserons toujours une confiance (i.e. $1 - \alpha$) à 95%. Nous chercherons à être consistants tout en évitant d'être trop pessimistes.

En une dimension, la mise à jour par intersection de covariance d'une prédiction $\hat{\sigma}_{s_e}^2 [k|k-1]$ par une observation notée $g_{nss} s_e [k]$ s'écrit

$$\hat{\sigma}_{s_e}^2 [k|k] = \left(\frac{\omega}{\hat{\sigma}_{s_e}^2 [k|k-1]} + \frac{1-\omega}{\sigma_{g_{nss} s_e}^2 [k]} \right)^{-1}, \quad (3.6)$$

$$\hat{s}_e [k|k] = \hat{\sigma}_{s_e}^2 [k|k] \left(\frac{\omega \hat{s}_e [k|k-1]}{\hat{\sigma}_{s_e}^2 [k|k-1]} + \frac{(1-\omega) g_{nss} s_e [k]}{\sigma_{g_{nss} s_e}^2 [k]} \right), \quad (3.7)$$

où $\omega \in [0, 1]$ est tel que la variance $\hat{\sigma}_{s_e}^2 [k|k]$ soit minimale.

Minimiser $\hat{\sigma}_{s_e}^2 [k|k]$ est équivalent à maximiser

$$\omega \left(\frac{1}{\hat{\sigma}_{s_e}^2 [k|k-1]} - \frac{1}{\sigma_{g_{nss} s_e}^2 [k]} \right) + \frac{1}{\sigma_{g_{nss} s_e}^2 [k]}. \quad (3.8)$$

Cette équation est celle d'une droite. Le maximum dépend donc du signe de la pente : $1/\hat{\sigma}_{s_e}^2 [k|k-1] - 1/\sigma_{g_{nss} s_e}^2 [k]$.

Si $\hat{\sigma}_{s_e}^2 [k|k-1] < \sigma_{g_{nss} s_e}^2 [k]$, le maximum est attend pour $\omega = 1$ et

$$\hat{s}_e [k|k] = \hat{s}_e [k|k-1], \quad (3.9)$$

$$\hat{\sigma}_{s_e}^2 [k|k] = \hat{\sigma}_{s_e}^2 [k|k-1], \quad (3.10)$$

sinon, le maximum est attend pour $\omega = 0$ et

$$\hat{s}_e [k|k] = g_{nss} s_e [k], \quad (3.11)$$

$$\hat{\sigma}_{s_e}^2 [k|k] = \sigma_{g_{nss} s_e}^2 [k]. \quad (3.12)$$

L'intersection de covariance en une dimension est alors équivalente à conserver l'estimateur avec la plus petite variance comme l'ont montré [Reinhardt *et al.*, 2012]. La fusion en abscisse curviligne devient dans ce cas trivialement simple.

3.3. Localisation coopérative par fusion de données séquentielles

3.3.1. Problématique

La localisation du véhicule est réalisée grâce à une prédiction utilisant la vitesse mesurée ainsi qu'une mise à jour de Kalman utilisant la position calculée par le

1. ou « intègre » par abus de vocabulaire car la notion d'intégrité est en général utilisée pour des risques très faibles, ce qui n'est pas le cas dans ces travaux.

3. Localisation coopérative à une dimension en abscisse curviligne

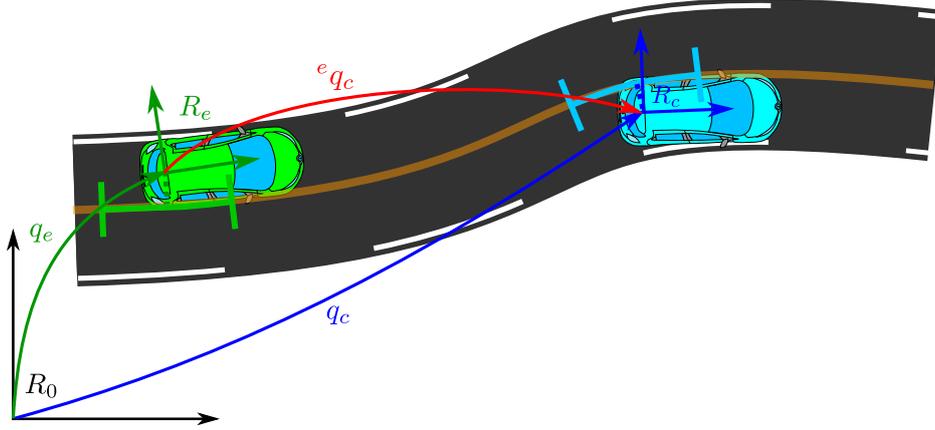


FIGURE 3.1. – Localisation coopérative entre la pose de l'égo-véhicule q_e (en vert) et celle du véhicule communicant q_c (en bleu) à partir de la pose relative ${}^e q_c$ (en rouge). La fusion à une dimension est réalisée le long du centre de la voie (en orange). Les incertitudes de chaque véhicule sont représentées par des intervalles de confiance.

récepteur GNSS. Il est possible de rajouter une mise à jour (de type Kalman ou par intersection de covariance) en utilisant une communication V2V pour recevoir la pose d'un autre véhicule. Pour être capable d'utiliser cette nouvelle information, le véhicule doit être capable de localiser ce second véhicule dans son repère local. À partir de cette pose relative et de la pose absolue du second véhicule, ce dernier peut être vu comme une antenne déportée et être utilisé pour améliorer l'estimation de la pose absolue du véhicule par fusion de données (Fig. 3.1).

La pose relative perçue ${}_{perçu}^e q_c = ({}_{perçu}^e x_c \ {}_{perçu}^e y_c \ {}_{perçu}^e \theta_c)^T$ du véhicule communicant c dans le repère local du égo-véhicule e est nécessaire afin d'utiliser la pose communiquée du premier pour calculer celle du second. En utilisant un des opérateurs de composition de [Smith *et al.*, 1990] présentés dans l'annexe B, on peut écrire :

$$com q_e = com q_c \ominus {}_{perçu}^e q_c. \quad (3.13)$$

Si on suppose que l'ordonnée curviligne et l'orientation curviligne de $com g_e$ et de $com g_c$ sont connues sans erreur :

$$\Sigma_{com g_e} = \begin{bmatrix} \hat{\sigma}_{s_e}^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{et} \quad \Sigma_{com g_c} = \begin{bmatrix} \sigma_{s_c}^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.14)$$

on peut utiliser la matrice de rotation

$${}^e R_c = \begin{bmatrix} \cos \Delta \alpha & -\sin \Delta \alpha & 0 \\ \sin \Delta \alpha & \cos \Delta \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.15)$$

où $\Delta \alpha = \psi_c - \hat{\psi}_e$ est l'angle entre les segments associés aux *map-matching* des deux véhicules, pour déterminer la matrice de covariance du égo-véhicule en fonction de

3. Localisation coopérative à une dimension en abscisse curviligne

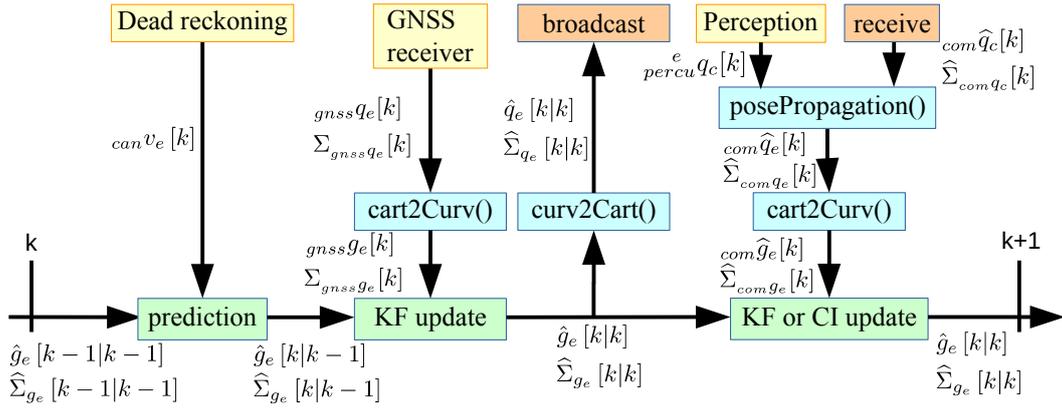


FIGURE 3.2. – Algorithme de fusion de données exécuté dans chaque véhicule.

celle du véhicule communicant :

$$\begin{aligned}
 {}_{com}\Sigma_{s_e} &= {}^e R_c \Sigma_{com s_c} {}^e R_c^T \\
 &= \sigma_{com s_c}^2 \begin{bmatrix} \cos^2 \Delta\alpha & \cos \Delta\alpha \cdot \sin \Delta\alpha & 0 \\ \cos \Delta\alpha \cdot \sin \Delta\alpha & \sin^2 \Delta\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix}.
 \end{aligned} \tag{3.16}$$

Par identification, on a

$$\hat{\sigma}_{com s_e}^2 = \sigma_{com s_c}^2 \cos^2(\Delta\alpha). \tag{3.17}$$

Cette nouvelle pose ${}_{com}q_e$ et cette nouvelle variance $\sigma_{com s_c}^2$ peuvent être utilisées dans une mise à jour de Kalman ou par intersection de covariance.

3.3.2. Algorithme de fusion coopérative de données

La figure 3.2 montre un algorithme réalisant une fusion de données complète sur un pas de temps. Il est exécuté par chaque véhicule.

Pour prendre en compte les corrélations entre les erreurs des observations, une seule mise à jour de l'état doit être réalisée. L'« observation » utilisée lors de cette mise à jour est alors un vecteur reprenant l'ensemble des observations de l'itération courante. La matrice de covariance de ce vecteur d'observation est alors composée des matrices de covariance des différentes observations sur sa diagonale définie par blocs et des matrices d'inter-corrélation des erreurs des observations autour de cette diagonale. En pratique, il est très difficile d'estimer ces matrices d'inter-corrélation. Si la matrice de covariance du vecteur d'observation est diagonale par blocs, il est possible de décomposer l'étape de mise à jour unique en plusieurs en appliquant une mise à jour pour chaque observation, car l'erreur est non corrélée aux autres. Cette propriété est beaucoup utilisée en fusion multi-capteur, chaque capteur étant utilisé à la volée.

À chaque itération k , l'abscisse curviligne \hat{s}_e et sa variance $\hat{\sigma}_{s_e}^2$ sont prédites (cf. module « prediction ») en utilisant la vitesse du véhicule fournie par le bus CAN ${}_{can}v_e$. La pose provenant des données du récepteur GNSS est ensuite convertie en coordonnées curvilignes (cf. module « cart2Curv() ») avant d'utiliser la nouvelle abscisse curviligne lors d'une mise à jour de Kalman (cf. module « KF update »). Une

3. Localisation coopérative à une dimension en abscisse curviligne

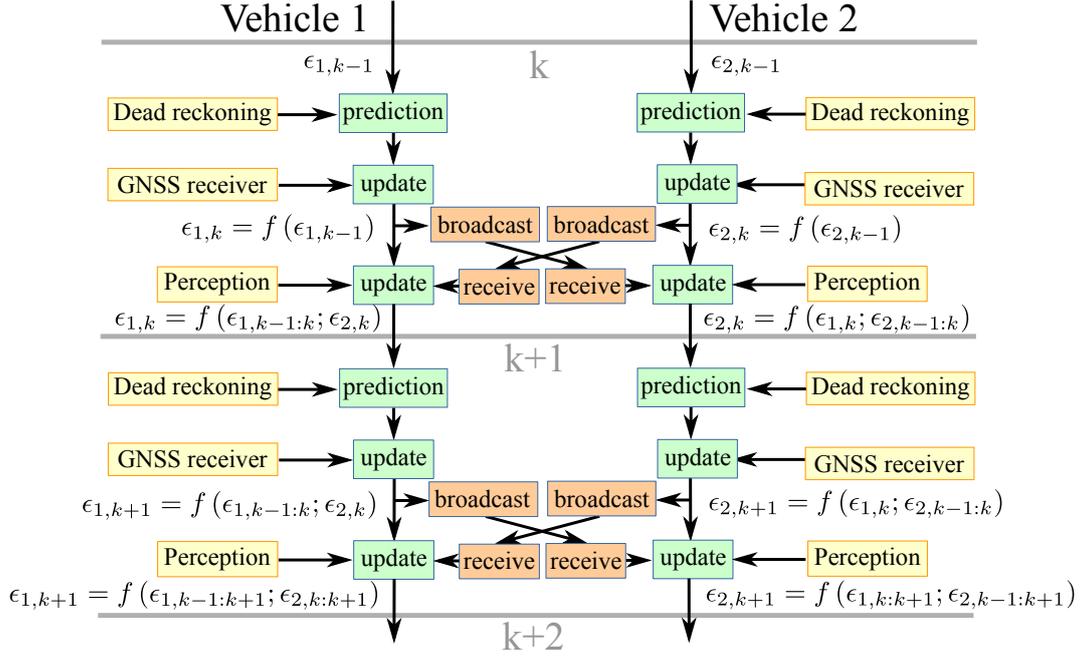


FIGURE 3.3. – Communication entre deux véhicules lors de la fusion coopérative.

fois cette étape réalisée, la pose curviligne complète \hat{g}_e et sa matrice de covariance $\hat{\Sigma}_{\hat{g}_e}$ sont envoyées aux autres véhicules après avoir été converties en coordonnées cartésiennes (cf. module « curv2Cart() »). Le véhicule peut ensuite utiliser la pose reçue du second véhicule ${}_{com}\hat{q}_c$ et la matrice de covariance associée $\hat{\Sigma}_{{}_{com}q_c}$ en la combinant à la pose relative déterminée par la perception ${}_{percu}^e q_c$ (la matrice de covariance est supposée être nulle dans cette étude) pour obtenir une nouvelle observation de sa pose ${}_{com}\hat{q}_e$ (cf. module « posePropagation() »). Cette pose est ensuite convertie en coordonnées curvilignes ${}_{com}\hat{g}_e$ (cf. module « cart2Curv() »). L'abscisse curviligne peut alors être utilisée comme observation dans une mise à jour de Kalman ou par intersection de covariance (cf. module « KF or CI update »).

3.3.3. Corrélations des erreurs lors de la fusion coopérative

La figure 3.3 présente la communication entre deux véhicules lors de la fusion coopérative en suivant l'algorithme précédemment présenté. Ce chronogramme présente l'évolution des erreurs des véhicules ϵ_1 (pour le véhicule 1) et ϵ_2 (pour le véhicule 2) ainsi que leurs corrélations temporelles aux erreurs des différentes itérations et des différents véhicules. On note ainsi $\epsilon_{i,k}$ l'erreur du véhicule i pour l'itération k . Ainsi $\epsilon_{1,k} = f(\epsilon_{1,k-1})$ signifie que l'erreur du véhicule 1 à l'itération k est fonction de l'erreur du véhicule 1 à l'itération $k-1$. On note $\epsilon_{i,h:l}$ l'ensemble des erreurs $\epsilon_{i,k}$ pour $k \in [h, l]$.

Lors de la mise à jour coopérative, la pose du véhicule communicant et la pose relative obtenue grâce à la perception sont combinées afin d'obtenir une pose observée du égo-véhicule comme montré dans l'équation (3.13). D'après cette équation, la pose observée utilisée pour la fusion est donc fonction de l'erreur du véhicule communicant. On a ainsi juste après la première mise à jour coopérative de la figure 3.3, $\epsilon_{1,k} =$

3. Localisation coopérative à une dimension en abscisse curviligne



FIGURE 3.4. – Voitures autonomes électriques Renault Zoe utilisées pour l'expérimentation.

$f(\epsilon_{1,k-1:k}; \epsilon_{2,k})$ et $\epsilon_{2,k} = f(\epsilon_{1,k}; \epsilon_{2,k-1:k})$. L'erreur de chaque véhicule se corrèle alors à l'erreur de l'autre véhicule.

On remarque aussi à partir de ce moment que l'erreur de chaque véhicule à l'itération k est corrélée avec l'erreur de ce même véhicule à l'itération $k-1$ ². Les erreurs d'estimation des deux véhicules n'étant pas blanches, la fusion en utilisant un filtre de Kalman n'est donc pas optimale.

Le second problème apparaît lors de la seconde fusion coopérative. L'état dépendant de $\epsilon_{1,k+1} = f(\epsilon_{1,k-1:k}; \epsilon_{2,k})$ est alors fusionné avec une observation dépendant de $\epsilon_{2,k+1} = f(\epsilon_{1,k}; \epsilon_{2,k-1:k})$ pour le véhicule 1. De même l'état dépendant de $\epsilon_{2,k+1} = f(\epsilon_{1,k}; \epsilon_{2,k-1:k})$ est fusionné avec une observation dépendant de $\epsilon_{1,k+1} = f(\epsilon_{1,k-1:k}; \epsilon_{2,k})$ pour le véhicule 2. Des informations avec des erreurs corrélées sont donc fusionnées. ϵ_1 et ϵ_2 sont, en effet, corrélées car elles dépendent en commun de $\epsilon_{1,k-1}$ et de $\epsilon_{2,k-1}$. Les fusionner avec une mise à jour de Kalman en les supposant décorrélées entraîne alors une consanguinité de données ce qui donne en pratique des résultats surconfiants.

3.4. Résultats

Les expérimentations ont été réalisées à partir de deux voitures autonomes Renault Zoe (cf Fig. 3.4) se suivant sur la piste expérimentale Séville de l'UTC (Fig. 2.1 page 30). Cette route est composée de deux ronds-points reliés par une ligne droite. Les véhicules étaient équipés d'un récepteur GNSS avec centrale inertielle (NovAtel SPAN-CPT) utilisant des corrections RTK afin d'obtenir une vérité terrain. Des récepteurs GNSS mono-fréquence bas coût (récepteurs u-blox 8T multi-constellation GLONASS et GPS) ont été utilisés pour les données de localisation. Les vitesses des

2. En effet, on peut montrer que $\epsilon_k = (I - KC)A\epsilon_{k-1} - (I - KC)\alpha_{k-1} + K\beta_k$, où A est la matrice de régime libre, C la matrice d'observation, α l'erreur du bruit de modèle et β l'erreur du bruit de mesure. I correspond à l'identité et K au gain de Kalman. Cette équation indique clairement la corrélation temporelle des erreurs d'estimation.

3. Localisation coopérative à une dimension en abscisse curviligne

TABLE 3.1. – Erreurs quadratiques moyennes de simulation $\|\epsilon_s\|$ et taux de couverture C des intervalles de confiance à 95%. Les écarts types σ_{x_e} et σ_{y_e} des deux véhicules sont de 1 m.

	Suiveur		Leader	
	C (%)	$\ \epsilon_s\ $ (m)	C (%)	$\ \epsilon_s\ $ (m)
KF sans communication	95	0,41	95	0,31
KF avec communication	90,29	0,20	89,71	0,20
CI avec communication	96,18	0,25	97,65	0,25

TABLE 3.2. – Erreurs quadratiques moyennes de simulation $\|\epsilon_s\|$ et taux de couverture C des intervalles de confiance à 95%. Les écarts types σ_{x_e} et σ_{y_e} sont de 1 m pour le véhicule suiveur et de 0,01 m pour le véhicule leader.

	Suiveur		Leader	
	C (%)	$\ \epsilon_s\ $ (m)	C (%)	$\ \epsilon_s\ $ (m)
KF sans communication	95	0,41	95	0,077
KF avec communication	94,12	0,071	93,82	0,071
CI avec communication	94,12	0,074	95	0,077

roues des véhicules ont aussi été enregistrées à partir du bus CAN. La pose relative entre les deux véhicules a été simulée à partir des données de vérité terrain afin d'utiliser une pose relative parfaite.

3.4.1. Résultats de simulation

Les données GNSS ont tout d'abord été simulées en ajoutant un bruit blanc avec des écarts types σ_{x_e} et σ_{y_e} de 1 mètre sur les positions des vérités terrain. Les résultats de simulation ont été présentés dans [Héry *et al.*, 2017b].

On peut voir sur la table 3.1 qu'utiliser la communication par mise à jour de Kalman n'est pas consistant. En effet, le taux de couverture C , c'est-à-dire le pourcentage d'estimés à l'intérieur des intervalles de confiance, atteint 90%. L'intersection de covariance reste consistante même si elle est pessimiste ($C > 95\%$). L'erreur quadratique moyenne $\|\epsilon_s\|$ des trois méthodes montre que la précision augmente lorsque la communication est utilisée. Dans un second scénario, les écarts types σ_{x_e} et σ_{y_e} du bruit blanc du leader a été réduit à 0,01 m (cf. Tab. 3.2). Comme le filtre par intersection de covariance est équivalent à un filtre par minimum de variance, on peut voir que la grande précision du leader est propagée au suiveur.

La figure 3.5 montre la distribution de l'erreur en valeur absolue du véhicule suiveur en fonction de l'angle relatif entre les segments *map-matchés* par les deux véhicules pour le premier scénario. On peut vérifier que lorsque les véhicules partagent des informations, l'erreur de localisation décroît en fonction de cet angle, comme on peut le voir théoriquement dans l'équation 3.17. Cette tendance est vraie pour les erreurs des deux véhicules.

3. Localisation coopérative à une dimension en abscisse curviligne

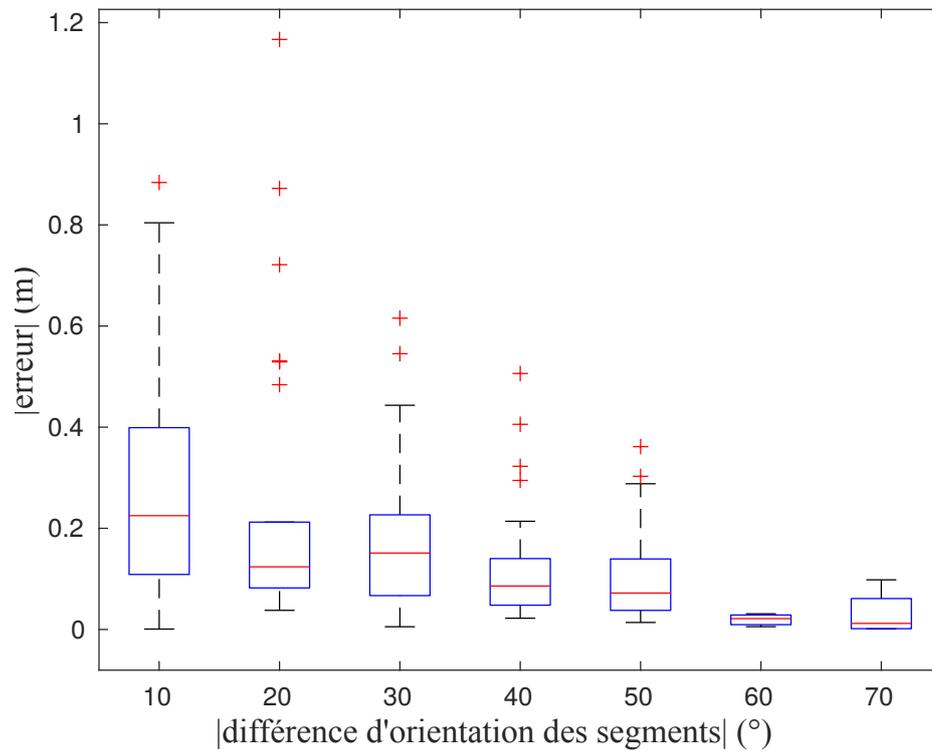


FIGURE 3.5. – Boîte à moustaches de la valeur absolue de l'erreur $|\epsilon_s|$ en fonction de l'angle entre les segments associés aux map-matching des deux véhicules (CI avec communication).

3. Localisation coopérative à une dimension en abscisse curviligne

TABLE 3.3. – Erreurs quadratiques moyennes expérimentales $\|\epsilon_s\|$ et taux de couverture C des intervalles de confiance à 95%.

	Véhicule suiveur		Véhicule leader	
	C (%)	$\ \epsilon_s\ $ (m)	C (%)	$\ \epsilon_s\ $ (m)
KF sans communication (Fig. 3.6 et 3.7 B)	94,83	0,55	94,1	0,96
KF avec communication du leader au suiveur (Fig. 3.6 C)	52,40	0,64 (+0,09)	94,1	0,96 (=)
CI avec communication du leader au suiveur (Fig. 3.6 D)	94,83	0,55 (=)	94,1	0,96 (=)
KF avec communication du suiveur au leader (Fig. 3.7 C)	94,83	0,55 (=)	47,24	0,50 (-0,46)
CI avec communication du suiveur au leader (Fig. 3.7 D)	94,83	0,55 (=)	93,36	0,51(-0,45)
KF avec une communication bidirectionnelle (Fig. 3.6 et 3.7 E)	38,38	0,44 (-0,11)	31	0,43 (-0,53)
CI avec une communication bidirectionnelle (Fig. 3.6 et 3.7 F)	95,2	0,46 (-0,09)	95,57	0,45 (-0,51)

3.4.2. Résultats expérimentaux

Des mesures u-blox ont ensuite été utilisées. Les résultats expérimentaux ont été présentés dans [Héry *et al.*, 2017c].

Les figures 3.6 et 3.7 montrent les erreurs (en bleue) et leurs intervalles de confiance à 95% (en pointillé rouge) du suiveur et du leader en utilisant différentes sources d'information fusionnées avec différentes méthodes (filtre de Kalman ou filtre par intersection de covariance).

- Le graphique A représente l'erreur et l'intervalle de confiance de l'abscisse curviligne provenant du récepteur GNSS utilisé seul sans filtrage.
- Le graphique B montre le résultat obtenu avec un filtre de Kalman en utilisant la vitesse du véhicule et les données GNSS.
- Le graphique C représente le cas où la communication d'un autre véhicule (du leader au suiveur pour le suiveur ou du suiveur au leader pour le leader) est fusionnée lors d'une mise à jour de Kalman.
- Le graphique D correspond au même scénario lorsque la mise à jour de Kalman est remplacée par une intersection de covariance.
- Le graphique E montre le résultat lorsque la communication est utilisée dans les deux directions (le suiveur reçoit la pose du leader et le leader celle du suiveur) avec une mise à jour de Kalman.
- Enfin, le graphique F est similaire au cas précédent en utilisant une mise à jour par intersection de covariance à la place de celle de Kalman.

La table 3.3 montre la racine carrée de l'erreur quadratique moyenne $\|\epsilon_s\|$ et le taux de couverture C des intervalles de confiance à 95% pour les différentes sources d'information fusionnées avec différentes méthodes.

Les résultats obtenus dans la table 3.3 montrent que lorsqu'une communication bidirectionnelle est utilisée avec un filtre de Kalman, une surconvergence de l'inter-

3. Localisation coopérative à une dimension en abscisse curviligne

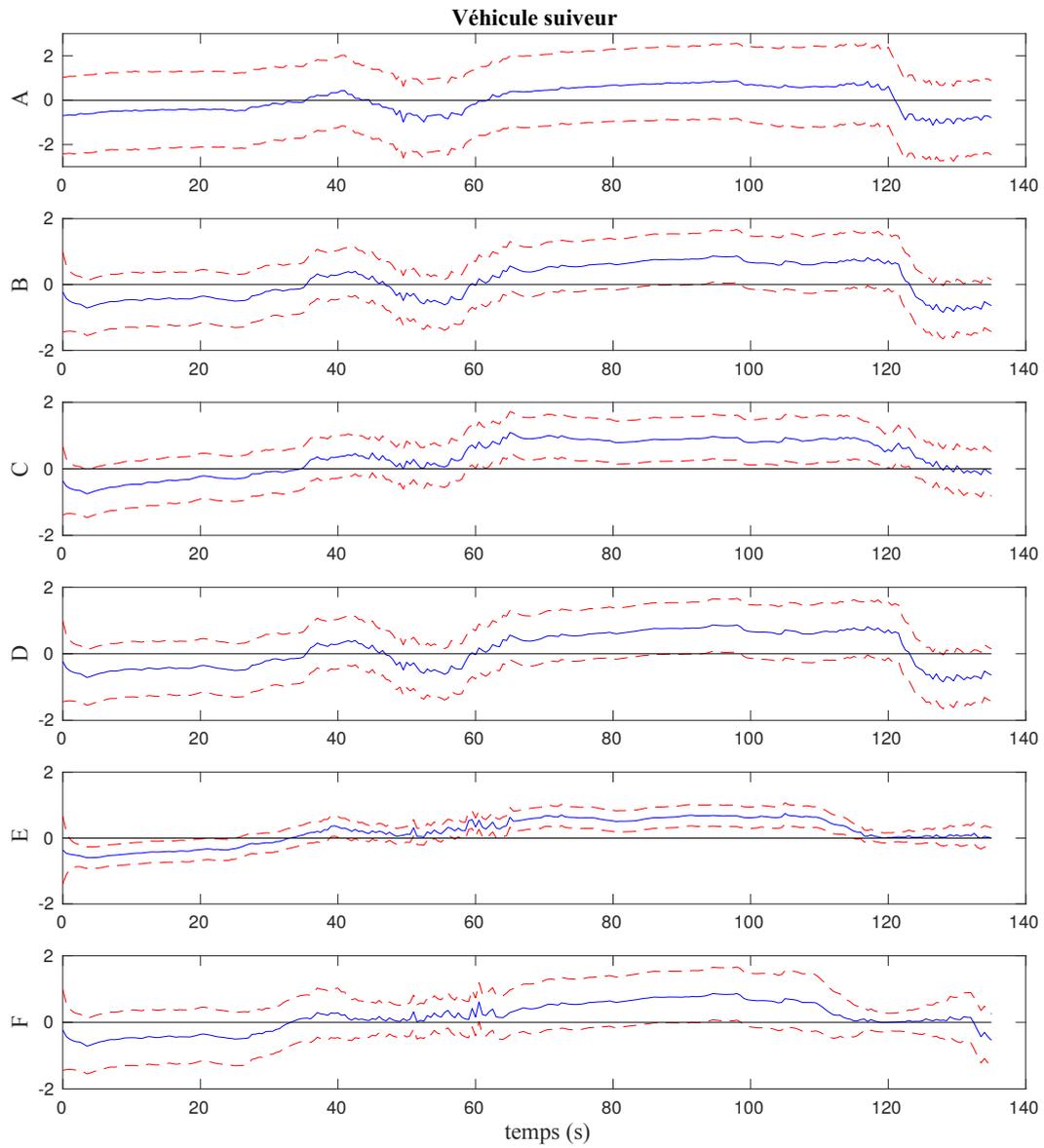


FIGURE 3.6. – *Erreurs longitudinales ϵ_s (m) avec leurs intervalles de confiance à 95% du véhicule suiveur.*

3. Localisation coopérative à une dimension en abscisse curviligne

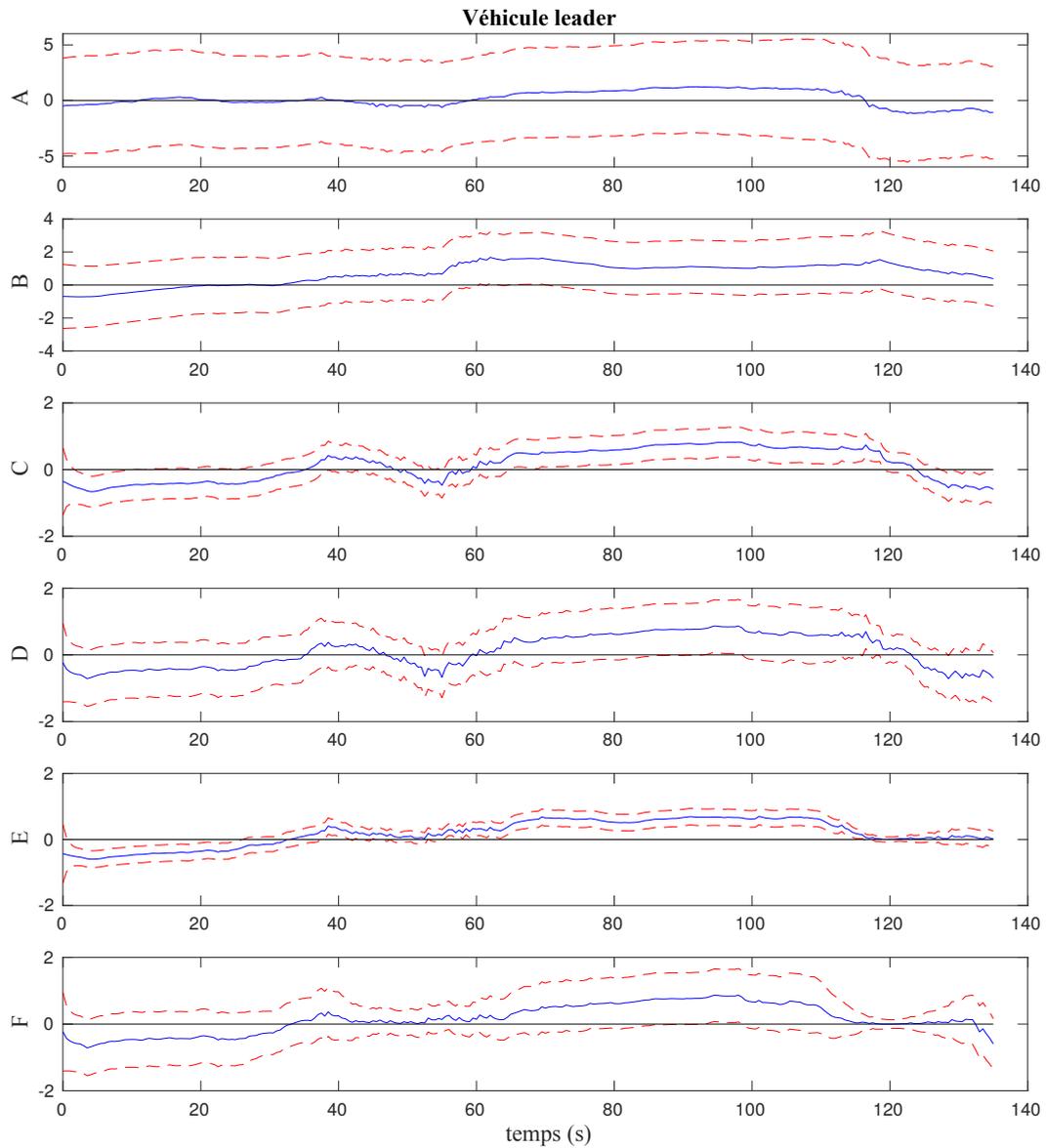


FIGURE 3.7. – Erreurs longitudinales ϵ_s (m) avec leurs intervalles de confiance à 95% du véhicule leader.

3. Localisation coopérative à une dimension en abscisse curviligne

valle de confiance apparaît. En d'autres termes, le filtre devient surconfiant car son taux de couverture C diminue. Au contraire, lorsqu'une communication bidirectionnelle est utilisée avec un filtre par intersection de covariance, le taux de couverture augmente légèrement. En effet, la mise à jour par intersection de covariance fournit des résultats pouvant être pessimistes.

Dans ces expérimentations, chaque fois que l'information reçue par communication est utilisée avec une mise à jour de Kalman, une surconfiance apparaît. En effet, les erreurs provenant des récepteurs GNSS des deux voitures ne sont pas décorréliées du fait des erreurs de mode commun dues aux retards ionosphériques et troposphériques et aux erreurs d'éphémérides des satellites. Une autre conséquence résultant de cette dépendance apparaît lorsqu'un véhicule communique un état moins précis : l'état du second véhicule devient alors également moins précis. Ce phénomène peut être vu dans la table 3.3. Le suiveur, qui était plus précis que le leader, voit son erreur augmenter lorsque la communication est utilisée avec une mise à jour de Kalman. Au contraire, lorsque le leader, qui était moins précis, utilise la communication avec une mise à jour de Kalman, il voit son erreur diminuer. Lorsqu'une communication est utilisée, la mise à jour par intersection de covariance doit donc toujours être utilisée pour assurer l'intégrité de l'estimation de l'incertitude.

Cette étude a montré aussi clairement que lorsqu'un véhicule possède un récepteur GNSS plus précis que celui de ces voisins, il peut propager sa précision aux autres véhicules du convoi ce qui peut être intéressant dans de nombreuses applications.

D'après l'Eq. 3.17 et comme nous l'avons vu précédemment, la précision augmente lorsque l'angle entre les segments *matchés* par les deux véhicules augmente. Ce phénomène peut être à nouveau observé sur la Fig. 3.6 et 3.7 autour de 40s, 60s et 115s, 130s lorsque les véhicules sont dans les ronds-points et lorsque la communication bidirectionnelle est utilisée. La distance latérale étant sans erreur ici, elle permet d'augmenter la précision de l'estimation d'abscisse curviligne de façon très significative dans ces situations.

3.5. Conclusion sur la localisation coopérative à une dimension en abscisse curviligne

Nous avons présenté des méthodes de localisation coopératives à une dimension en utilisant l'abscisse curviligne de chaque véhicule le long d'une carte de référence. Cette étude a traité le problème de l'estimation de la position le long de la voie, l'écart latéral à la voie et l'orientation étant supposés estimés avec des erreurs négligeables devant celle de l'erreur longitudinale. Une stratégie de fusion décentralisée a été présentée et mise en œuvre. Des résultats de simulation et expérimentaux ont été obtenus à partir de données réelles de deux voitures autonomes Renault Zoe. On a montré que l'utilisation de la communication pouvait engendrer des problèmes de surconfiance à cause d'une consanguinité de données ou simplement lorsque les erreurs GNSS n'étaient pas indépendantes. Afin d'éviter ces problèmes de consistance, le filtre par intersection de covariance est une bonne alternative au filtre de Kalman. Cette stratégie de fusion devient extrêmement simple à implémenter en une seule dimension. Nous avons montré que l'algorithme proposé est particulièrement efficace pour réduire l'erreur longitudinale lorsque les véhicules roulent dans des virages. Un autre résultat intéressant apparaît lorsqu'un véhicule utilise un récepteur GNSS plus

3. Localisation coopérative à une dimension en abscisse curviligne

précis que les autres : sa précision est alors propagée aux autres véhicules. Ainsi, si un convoi contient un véhicule capable de se localiser très précisément en absolu, alors chaque véhicule de ce convoi peut se localiser avec la même précision si les conditions suivantes sont remplies : il dispose d'une carte haute définition, il peut se localiser précisément latéralement et angulairement et il peut localiser précisément de manière relative les véhicules à proximité avec ses capteurs de perception embarqués. Dans ce chapitre, l'écart latéral à la voie et l'orientation relative à celle-ci étaient supposées parfaites tout comme la pose relative. Dans la suite de ces travaux, nous nous intéresserons au cas où ces informations ne sont plus parfaites et où leurs incertitudes doivent être estimées. Nous commencerons ainsi par estimer des poses relatives et leurs matrices de covariance associées à partir de scans LiDAR.

4. Estimation de la pose inter-véhicules et de la matrice de covariance associée à partir de mesures LiDAR

4.1. Introduction

Dans le chapitre 3, la localisation coopérative de véhicules communicants a été étudiée avec un modèle à une dimension basé sur l'abscisse curviligne des véhicules. La localisation relative issue de la perception était alors supposée parfaite et avait été simulée à partir de la vérité terrain. Ainsi une étape clé pour la mise en œuvre d'une méthode de localisation coopérative est l'estimation de la pose relative provenant de la perception.

Il existe de nombreux algorithmes de détection de véhicules utilisant la vision dans la littérature, notamment avec l'émergence de l'apprentissage profond et de détecteurs avec de très bonnes performances [Chateau *et al.*, 2017]. Néanmoins, ces méthodes retournent simplement une boîte englobante dans le repère image sans donner souvent une pose métrique du véhicule détecté et, lorsque c'est le cas, la précision est en général trop faible pour nos besoins de localisation coopérative. À l'inverse, une détection de véhicule utilisant des mesures 3D est bien plus adaptée pour l'estimation de la pose relative. La détection de véhicule par mise en correspondance de points LiDAR avec un modèle comme le *L-shape fitting* [Zhang *et al.*, 2017] donne une bonne estimation de la pose relative. Étant donné que la forme réelle du véhicule perçu n'est à priori pas connue, seules des formes géométriques simples comme des boîtes englobantes peuvent être utilisées pour ces correspondances. Dans le cas de véhicules communicants, des modèles plus précis donnant de meilleures représentations des géométries des véhicules peuvent être échangés entre véhicules voisins. L'ICP (*Iterative Closest Point*) est un algorithme souvent utilisé pour mettre en correspondance deux scans LiDAR successifs afin de réaliser de l'odométrie basée sur des mesures LiDAR [Censi, 2008]. Il peut aussi être utilisé pour mettre en correspondance un scan LiDAR avec un modèle [Shetty, 2017]. L'ICP donne une très bonne estimation de la pose relative, mais la matrice de covariance de l'erreur d'estimation n'est souvent pas calculée. Pour obtenir des algorithmes capables d'être intégrés à des mécanismes de contrôle d'intégrité, cette matrice de covariance doit être estimée de façon la plus juste possible [Censi, 2007, Prakhya *et al.*, 2015, Bengtsson et Baerveldt, 2003].

Dans ce chapitre, nous présentons plusieurs algorithmes de localisation relative utilisant des correspondances entre des points LiDAR et un modèle. Nous cherchons à étudier des algorithmes « *snapshots* » n'utilisant pas de pistage et capables de fournir une pose relative sans prendre en compte ce qui s'est passé à une étape précédente. De cette façon, les erreurs de perception seront indépendantes d'une étape à l'autre

4. Pose inter-véhicules à partir de mesures LiDAR

ce qui sera une hypothèse importante pour mettre en œuvre par la suite un système de fusion dynamique reposant sur du filtrage d'état. De plus, ces méthodes « non dynamiques » n'ont pas à converger pour donner un résultat précis et permettent donc de fournir un résultat dès qu'une mesure LiDAR est disponible. Un autre avantage de cette approche est qu'il est plus simple d'évaluer la consistance d'une méthode de perception de ce type.

Dans la section 4.2, nous introduisons le problème de la localisation relative. La section 4.3 présente ensuite une méthode de regroupement de points, permettant d'isoler les points LiDAR correspondant au véhicule perçu. Dans la section 4.4, nous décrivons une première méthode de localisation permettant d'obtenir rapidement une pose relative initiale. Un algorithme de localisation relative par échantillonnage aléatoire utilisant un champ de vraisemblance est présenté dans la section 4.5. Cette même distance est ensuite utilisée dans la section 4.6 pour résoudre la localisation relative en utilisant une minimisation itérative. Enfin, des résultats de simulation de la localisation relative sont étudiés dans la section 4.7.

4.2. Problématique

La localisation relative de véhicules les uns par rapport aux autres est une problématique importante pour la navigation autonome, notamment pour l'évitement d'obstacles, les dépassements, le franchissement d'intersections et pour les stratégies de conduite en convoi.

Dans ce chapitre, nous présentons différents algorithmes de localisation relative à partir de mesures LiDAR. Ces algorithmes sont utilisés pour estimer la pose relative d'un véhicule cible dans le repère mobile du égo-véhicule à partir de l'échange de poses absolues et d'un modèle géométrique représentant la forme du véhicule perçu. Le repère commun de ces poses absolues est un repère ENU (*East, North, Up*) dont les axes sont orientés vers l'est, le nord et le haut. On suppose que l'origine se situe suffisamment proche des poses étudiées notamment pour que l'hypothèse de monde 2D dans le plan Est-Nord soit valide.

En ce qui concerne le LiDAR, nous ramenons aussi à un problème à deux dimensions en supposant que la nappe LiDAR est parfaitement parallèle au plan de la route et se trouve à la hauteur du pare-chocs du véhicule cible, la hauteur correspondant au modèle géométrique 2D. Les véhicules étant conçus pour qu'en cas d'accident les pare-chocs se rentrent dedans, ceux-ci sont généralement à la même hauteur. Nous supposons aussi que le roulis pouvant être dus à de hautes vitesses dans les virages ou le tangage provenant de changement d'inclinaison de la route ou lors des phases d'accélération sont négligeables.

On note dans ce document ${}^e q_t = [{}^e x_t \ {}^e y_t \ {}^e \theta_t]^T$, la pose du véhicule cible (*target*) dans le repère du égo-véhicule (voir Fig 4.1).

Chaque capteur possède un repère de mesure. La figure 4.2 illustre ces différents repères capteurs ainsi que le repère mobile. Nous supposons dans ce document que ces capteurs sont correctement calibrés et que leurs mesures sont ramenées dans le repère mobile R_e par une transformation du repère du capteur au repère mobile. Cette opération demande une calibration précise des différentes transformations géométriques. Ainsi, on considèrera qu'un égo-véhicule, équipé d'un LiDAR, perçoit un

4. Pose inter-véhicules à partir de mesures LiDAR

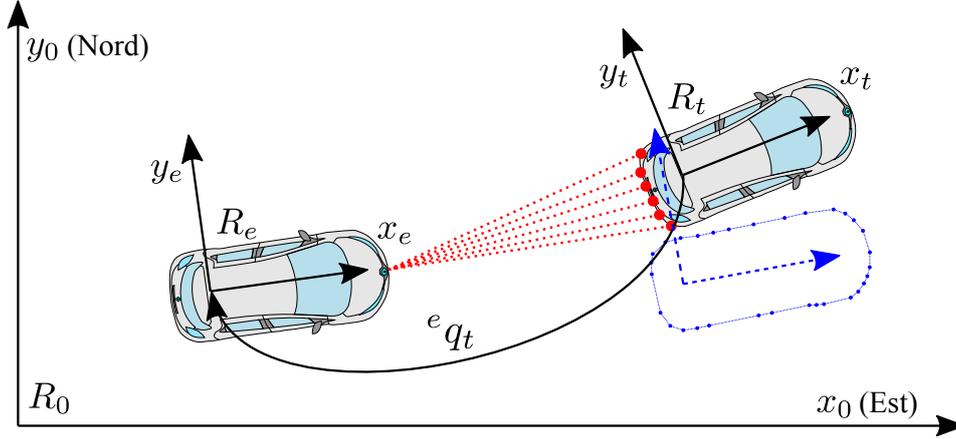


FIGURE 4.1. – *Le véhicule estime la transformation rigide ${}^e q_t$ du repère du véhicule perçu R_t dans son propre repère R_e . Le véhicule perçu communique une estimation de sa pose avec un modèle géométrique polygonal (représentée ici en bleu avec une erreur d'estimation importante).*

véhicule cible en obtenant un ensemble de points 2D

$$P = \{{}^e p_i = [{}^e x_i, {}^e y_i], i = 1, \dots, N_P\}, \quad (4.1)$$

exprimés directement dans le repère hôte R_e .

On utilise un récepteur GNSS en couplage lâche ce qui fournit ainsi directement la pose du égo-véhicule, l'orientation pouvant être calculée précisément par le récepteur à partir des mesures d'effets Doppler lorsque le véhicule est en déplacement (sous hypothèse de glissement négligeable).

Nous faisons l'hypothèse que le véhicule cible communique¹ la pose estimée de son repère mobile R_t ainsi qu'un modèle géométrique polygonal 2D,

$$M = \{{}^t m_j = [{}^t x_j, {}^t y_j], j = 1, \dots, N_M\}, \quad (4.2)$$

représentant le contour du véhicule dans le repère mobile en vue d'oiseau, comme illustré sur la figure 4.1. Dans ce chapitre les résultats ont été obtenus à partir d'un modèle simplifié où seul l'arrière du modèle (la partie visible par le LiDAR lors d'une conduite en convoi) était correctement discrétisé. La partie avant est carrée et correspond simplement à l'avant de la boîte englobante. Les points ${}^t m_j$ représentent les sommets du modèle et les arêtes sont définies par deux sommets consécutifs $({}^t m_j; {}^t m_{j+1})$, on notera $\vec{{}^t n}_j$ la normale unitaire à l'arête $({}^t m_j; {}^t m_{j+1})$ orientée vers l'extérieur du modèle.

Dans ce chapitre, nous supposons que toutes les données (pose du égo-véhicule, pose communiquée du véhicule cible et mesures LiDAR) sont synchronisées. Par la suite, ces algorithmes de localisation relative seront utilisés dans des méthodes de fusion de données où les poses du égo-véhicule et du véhicule cible pourront être mises à jour pour correspondre au temps d'acquisition des mesures LiDAR.

1. Dans ce chapitre, la pose communiquée sert à l'initialisation de l'algorithme. Dans le chapitre 6, la méthode de perception sera initialisée par une estimation locale de la pose du véhicule perçu.

4. Pose inter-véhicules à partir de mesures LiDAR

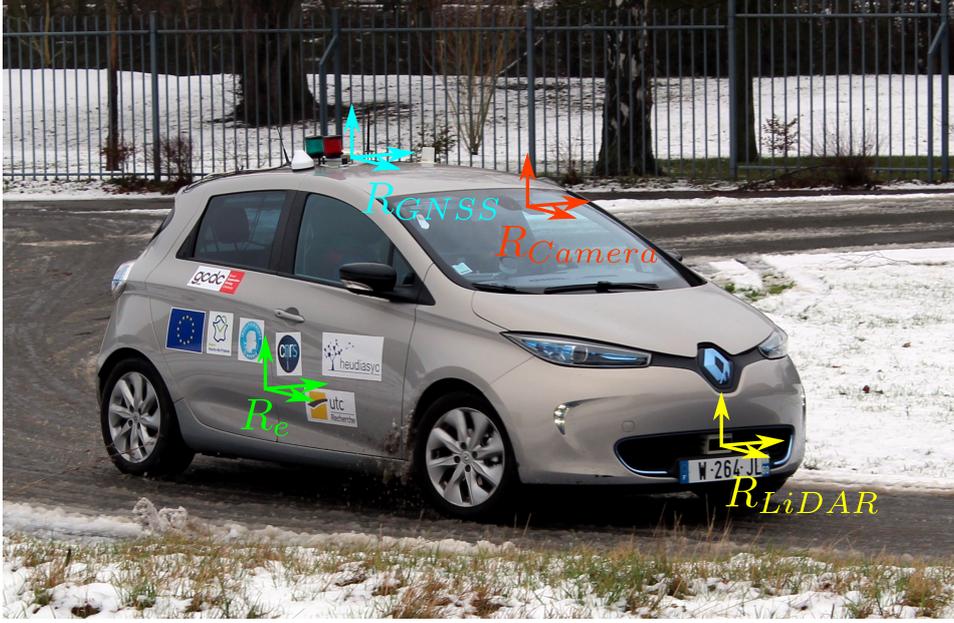


FIGURE 4.2. – Repère mobile R_e du égo-véhicule (centré au milieu de l'essieu arrière du véhicule) et repères des capteurs extéroceptifs de la Zoe instrumentée du laboratoire (LiDAR, Caméra et GNSS).

Nous cherchons ici à estimer la pose relative ${}^e q_t = [{}^e x_t \ {}^e y_t \ {}^e \theta_t]$ du véhicule cible dans le repère du égo-véhicule. ${}^e q_t$ représente la transformation rigide (en translation et en rotation) qui lie R_t à R_e (Fig. 4.1).

Ce problème revient alors à chercher la transformation qui associe le modèle avec l'ensemble des points LiDAR en minimisant une erreur E :

$${}^e \hat{q}_t = \arg \min_q E(q; P, M) = \arg \min_q \sum_{i=1}^{N_P} d(M, p_i; q)^2, \quad (4.3)$$

où $d(M, p_i; q)$ représente la distance entre le point p_i et le modèle M pour la transformation liée à une pose q .

Dans ce chapitre, nous comparerons différentes distances d (euclidiennes et orthogonales) ainsi que différentes méthodes de minimisation.

Les points LiDAR et le modèle n'étant pas définis dans le même repère, un repère commun de travail doit être choisi afin de pouvoir calculer ces distances. Il faut soit déplacer le modèle dans le repère de l'égo-véhicule soit les points LiDAR dans le repère du véhicule cible. Dans le second cas, la pose relative ${}^t q_e$ obtenue est celle de l'égo-véhicule dans le repère du véhicule cible. Il faut alors l'inverser pour obtenir celle recherchée ${}^e q_t$. Le temps nécessaire pour réaliser cette inversion est toutefois négligeable comparé à celui nécessaire pour réaliser la transformation de l'ensemble des points LiDAR ou du modèle. Le modèle contient non seulement des sommets, mais aussi des tangentes et des normales aux arrêtes ce qui rend la transformation plus complexe que celle des points LiDAR. Le modèle étant constant pour un véhicule donné, il n'est calculé qu'une seule fois. La rapidité de déplacer les points LiDAR plutôt que le modèle est cependant aussi discutable. Elle dépend en effet du nombre

4. Pose inter-véhicules à partir de mesures LiDAR

de points LiDAR et de la discrétisation du modèle. Si le modèle est très discrétisé, il sera composé de beaucoup de points et sera donc long à déplacer. De la même façon, si beaucoup de points LiDAR sont utilisés, leurs déplacements pourraient prendre du temps. Nous préférons faire le choix de déplacer les points LiDAR pour éviter d'avoir à réaliser le déplacement du modèle avec ces normales et ces tangentes. Un bon compromis entre une discrétisation précise permettant d'obtenir un modèle plus juste et une discrétisation plus grossière permettant d'accélérer la mise en correspondance des points LiDAR avec les arêtes du modèle sera fait pour le choix de ce modèle.

Les notations seront simplifiées en notant les éléments ${}^t p_i$, ${}^t m_j$ et ${}^t \vec{n}_j$ par p_i , m_j et \vec{n}_j , le repère du véhicule cible étant le repère utilisé par la perception.

4.3. Regroupement de points LiDAR

La première étape est de grouper les points LiDAR appartenant à chaque élément de la scène (Alg. 4.1) afin que le véhicule perçu soit représenté par un seul groupe de points. Comme ces points sont mesurés au cours d'un balayage, les points d'un même élément de la scène sont enregistrés de façon successive. Pour chaque point LiDAR, on peut alors calculer les distances qui le séparent des K points le précédant. Mesurer la distance uniquement avec le point précédent ($K = 1$) pourrait poser problème dans le cas où des points aberrants apparaissent ou lorsqu'un élément est partiellement caché (les points LiDAR d'un élément pourraient être séparés dans plusieurs groupes).

On compare ensuite la plus petite de ces distances à un seuil D . Si celle-ci est inférieure au seuil, on regroupe les deux points utilisés pour calculer cette distance dans un même groupe. En faisant cela pour chaque point, on obtient alors un ensemble de groupes de points G représentant chacun un élément de la scène. Le seuil peut être choisi constant, mais en raison de l'angle séparant deux faisceaux laser successifs, un objet éloigné aura moins de points et ceux-ci seront plus éloignés les uns des autres qu'un objet se trouvant proche du LiDAR. Ainsi, il est utile de faire varier ce seuil D en fonction de la distance des points LiDAR $\rho_i = \|p_i\|$ et de l'angle entre deux faisceaux α . Le seuil doit alors répondre à la condition suivante :

$$D > \rho_i \tan(\alpha) \approx \rho_i \alpha, \quad (4.4)$$

pour les deux points p_i dont la distance est comparée au seuil D .

Dans la suite de ce chapitre, $P = \{p_i = [x_i, y_i], i = 1, \dots, N_P\}$ fera référence aux points LiDAR du véhicule perçu et non plus à l'ensemble du scan.

4.4. Localisation initiale par boite englobante

Nous présentons tout d'abord une première méthode d'estimation de la pose relative servant d'initialisation aux méthodes développées plus tard dans le chapitre.

Lors du GCDC une boite englobante était calculée pour chaque groupe de points. Les boites trop grandes ou trop petites étaient ensuite filtrées avant que les boites englobantes restantes soient comparées aux différentes voies. On pouvait ainsi savoir si un obstacle était sur la route et sur quelle voie il était. Dans ce travail, nous ne cherchons plus l'ensemble des véhicules présents mais le véhicule qui communique sa pose. L'utilisation de celle-ci permet d'éviter une mauvaise association.

4. Pose inter-véhicules à partir de mesures LiDAR

Algorithme 4.1 *Algorithme de regroupement de points.*

```

1:  $G = \text{clustering}(\{p_i, i \in \{0 \dots N_P\}\})$ 
   # Entrée : ensemble du scan LiDAR
   # Sortie : tableau de groupe de points
2: for  $i \in [1 \dots N_P]$  do
3:    $g_i = 0$  # Identifiant de groupe de chaque point
4: end for
5:  $G = []$  # Initialisation du tableau de groupe de points
6:  $N_G = 0$  # Initialisation du nombre de groupes
7: for  $i \in [k \dots N_P]$  do # Boucle sur tous les points LiDAR
8:   for  $j \in [1 \dots K]$  do # Boucle sur les K points précédents
9:      $d[j-1] = \|p_i - p_{i-j}\|$  # Distance de  $p_i$  à  $p_{i-j}$ 
10:  end for
11:   $[d_{min}, j_{min} + 1] = \min_j (d)$  # Distance de  $p_i$  au point le plus proche
12:  if  $d_{min} < D$  then # On regroupe les points
13:    if  $g_{i-j_{min}} = 0$  then # Si  $p_{i-j_{min}}$  n'a pas de groupe
14:       $N_G = N_G + 1$  # Nouveau groupe
15:       $G \leftarrow []$  # Ajout d'un groupe vide
16:       $g_{i-j_{min}} = N_G$  # Ajout de  $p_{i-j_{min}}$  dans ce nouveau groupe
17:       $G[i - j_{min}] \leftarrow p_{i-j_{min}}$  # Ajout de  $p_{i-j_{min}}$  dans  $G[i - j_{min}]$ 
18:    end if
19:     $g_i = g_{i-j_{min}}$  # L'identifiant de  $p_i$  est le même que celui de  $p_{i-j_{min}}$ 
20:     $G[i - j_{min}] \leftarrow p_i$  # Ajout de  $p_i$  dans le groupe de  $p_{i-j_{min}}$ 
21:  end if
22: end for

```

Une correction de la position initiale permet de trouver la pose relative minimisant le problème 4.3 en évitant une mauvaise correspondance. Les points LiDAR de l'arrière du véhicule pourraient en effet être mis en correspondance avec un des côtés ou l'avant du véhicule si la position initiale est trop mauvaise. Cette première approche de localisation relative permet de corriger cette position initiale.

Considérons le cas de deux véhicules se suivant et s'échangeant leurs poses absolues entachées d'incertitudes.

Cette première approche permet d'estimer l'erreur de position relative à corriger. Cette erreur est la différence entre la position de la boîte englobante du modèle communiqué et celle de la boîte englobante des points LiDAR correspondant au véhicule. La pose initiale du modèle communiqué est calculée à partir de la pose communiquée par le véhicule cible et de la pose du égo-véhicule.

Afin de prendre en compte l'orientation du véhicule cible dans la boîte englobante, les points LiDAR sont passés du repère du égo-véhicule au repère du véhicule cible (Alg. 4.2 lignes 2-4) grâce à la transformation :

$${}^t p_i = \ominus^e q_t \oplus {}^e p_i. \quad (4.5)$$

Ainsi, en prenant simplement les coordonnées minimum et maximum des points LiDAR, une boîte englobante orientée dans le sens et la direction du véhicule cible

4. Pose inter-véhicules à partir de mesures LiDAR



FIGURE 4.3. – Localisation initiale calculée à partir des coordonnées minimum et maximum d'un ensemble de points LiDAR et de la pose communiquée par un autre véhicule. Le groupe de points LiDAR est représenté en bleu clair. La pose et le modèle communiqués par le véhicule cible, en rouge, sont utilisés pour déterminer la boîte englobante, en bleu foncé, donnée par l'algorithme permettant d'obtenir la pose initiale. Les vérités terrain du égo-véhicule, à gauche, et du véhicule cible, à droite, sont en noir.

est déterminée (Alg. 4.2 ligne 5) :

$$box = \begin{bmatrix} \min(\{x_i^t\}) & \max(\{x_i^t\}) \\ \min(\{y_i^t\}) & \max(\{y_i^t\}) \end{bmatrix}. \quad (4.6)$$

On obtient ainsi la boîte englobante, le plus petit rectangle contenant l'ensemble des points LiDAR pour une orientation donnée.

Une fois que la boîte englobante des points LiDAR est déterminée (en trait plein bleu sur la figure 4.3), elle est comparée à celle du modèle polygonal (en rouge pointillé sur la figure 4.3). Un point de repère sur la boîte englobante des points LiDAR est ensuite comparé au point de repère correspondant sur la boîte englobante du modèle polygonal. Comme les véhicules se déplacent en convoi, le milieu de l'arrière du véhicule cible est toujours dans le champs de vision du LiDAR et peut donc être utilisé comme point de repère. L'erreur de position à corriger correspond alors à la différence de position entre ces deux points de repère, erreur matérialisée par une flèche en violet sur la figure 4.3 (Alg. 4.2 lignes 6-7). La figure 4.3 montre la boîte englobante aux dimensions du véhicule corrigée par cet algorithme en pointillés et en bleu foncé.

Pour déterminer le point de repère, on peut déterminer le cône de perception du véhicule en prenant les droites passant par le LiDAR et les quatre sommets de la boîte englobante du modèle communiqué. La droite partant la plus à droite et celle partant la plus à gauche forment ce cône. Si un seul segment de la boîte englobante se trouve dans ce cône, le point de repère est le centre de celui-ci (véhicules 3 et 4 de la figure 4.4). Sinon, deux segments sont compris dans ce cône et leur intersection est le point de repère (véhicules 1 et 2 de la figure 4.4). Cette méthode suppose que les deux segments sont tous les deux au moins partiellement visibles et dans le même groupe de points ou que l'unique segment est complètement visible.

En pratique, si nous sommes dans le cas de conduite en convoi, nous utilisons le centre de l'arrière de la boîte englobante comme point de référence. Ainsi la différence

4. Pose inter-véhicules à partir de mesures LiDAR

Algorithme 4.2 *La localisation initiale corrige la position en comparant la position d'un point de référence de la boîte englobante des points LiDAR et de la boîte englobante du modèle du véhicule.*

```

1:  ${}^e\hat{q}_t = \mathbf{initLocalisation}(\{{}^e p_i, i \in [1..N_P]\}, width, {}^e q_t = [{}^e x_t {}^e y_t {}^e \theta_t]^T)$ 
   # Entrée : points LiDAR, largeur du véhicule perçu et pose relative initiale
   # Sortie : pose relative corrigée
2: for  $i \in [1..N_P]$  do
3:    ${}^t p_i = \ominus {}^e q_t \oplus {}^e p_i$  # Passage des points LiDAR dans le repère du véhicule perçu
4: end for
5:  $box = \begin{bmatrix} \min(\{{}^t x_i\}) & \max(\{{}^t x_i\}) \\ \min(\{{}^t y_i\}) & \max(\{{}^t y_i\}) \end{bmatrix}$  # Calcul de la boîte englobante
6:  $\Delta p = [0 \frac{width}{2}]^T - [box(1,1) \frac{box(2,1)+box(2,2)}{2}]^T$  # Erreur de position
7:  ${}^e\hat{q}_t = \begin{pmatrix} {}^e q_t \oplus (-\Delta p) \\ {}^e \theta_t \end{pmatrix}$  # Correction de la pose relative
8: end

```

de position entre ces points de référence est calculée par

$$\Delta p = \begin{bmatrix} 0 & \frac{width}{2} \end{bmatrix}^T - \begin{bmatrix} box(1,1) & \frac{box(2,1) + box(2,2)}{2} \end{bmatrix}^T, \quad (4.7)$$

où *width* correspond à la largeur du véhicule. On peut ainsi corriger la pose relative à partir de la pose relative ${}^e q_t$ calculée en utilisant la pose communiquée par le véhicule cible :

$${}^e\hat{q}_t = \begin{pmatrix} {}^e q_t \oplus (-\Delta p) \\ {}^e \theta_t \end{pmatrix}. \quad (4.8)$$

4.5. Localisation par tirage aléatoire avec champ de vraisemblance

Afin de calculer précisément la pose du véhicule cible dans le repère du égo-véhicule, une approche par tirage aléatoire peut être utilisée. Ce genre de méthode est connu pour être souvent lent en temps de calcul, mais permet facilement de calculer une matrice de covariance². Pour accélérer les opérations, une vraisemblance peut être précalculée et stockée dans une grille. Ce champ de vraisemblance discrétisé avec des cellules de 1 cm de côté est illustré sur la figure 4.5.

La vraisemblance d'une cellule de la grille est calculée en fonction de la distance au modèle du véhicule avec une distribution normale :

$$l_{ij} = \frac{1}{\sqrt{2\pi\sigma_l^2}} e^{-\frac{d_{ij}^2}{2\sigma_l^2}}, \quad (4.9)$$

2. Même s'il faut beaucoup d'échantillons pour que la matrice de covariance soit correctement estimée.

4. Pose inter-véhicules à partir de mesures LiDAR

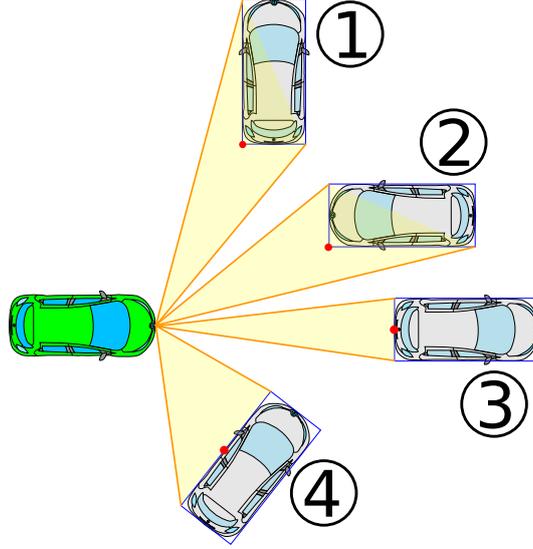


FIGURE 4.4. – *Les cônes de perception (en orange) du égo-véhicule (en vert) sont estimés à partir des boîtes englobantes des modèles communiqués (en bleu). Ils permettent d’en déduire le point de référence (en rouge) permettant de comparer la boîte englobante des points LiDAR à celle du modèle.*

où d_{ij} est la distance du centre de la cellule (i, j) au modèle polygonal et où σ_l est l'écart type du LiDAR. On suppose pour cet algorithme que l'écart type est identique dans toutes les directions.

Le problème de minimisation 4.3 devient, pour cette méthode, un problème de maximisation de l'ensemble des vraisemblances des points LiDAR.

L'algorithme commence par générer K poses aléatoires ${}^e q_{tk}$ selon une distribution normale (Alg. 4.3 ligne 3). La moyenne correspond à la pose initiale et les écarts types sont proportionnels à ceux de la pose initiale de telle sorte que suffisamment de poses soient proches de la pose réelle³.

Pour chaque tirage aléatoire, les points LiDAR sont déplacés dans la grille d'occupation pour évaluer la vraisemblance que la pose générée corresponde à la vraie pose relative du repère du véhicule cible (le repère du modèle) dans le repère du égo-véhicule (le repère des points LiDAR) (Alg. 4.3 ligne 5). Pour passer d'un point dans le repère du modèle à une cellule de la grille de vraisemblance, une translation entre la position de l'origine du repère dans le repère de la grille et l'origine de ce repère se trouvant en haut à gauche de cette grille est nécessaire. Les points obtenus sont divisés par la résolution de la grille pour être ensuite tronqués et correspondre à des indices $(w$ et $l)$ de cette grille de vraisemblance :

$$[w \ h]^T = \begin{bmatrix} \lfloor (p_{ikx} - \text{area}(1, 1)) / ds \rfloor \\ \lfloor (p_{iky} - \text{area}(2, 1)) / ds \rfloor \end{bmatrix}, \quad (4.10)$$

area correspond à la zone dans laquelle le champ de vraisemblance est défini. $\text{area}(1, 1)$ et $\text{area}(2, 1)$ sont respectivement l'abscisse et l'ordonnée du point en bas

3. En pratique, on utilisera deux fois les écarts types de la pose initiale

4. Pose inter-véhicules à partir de mesures LiDAR

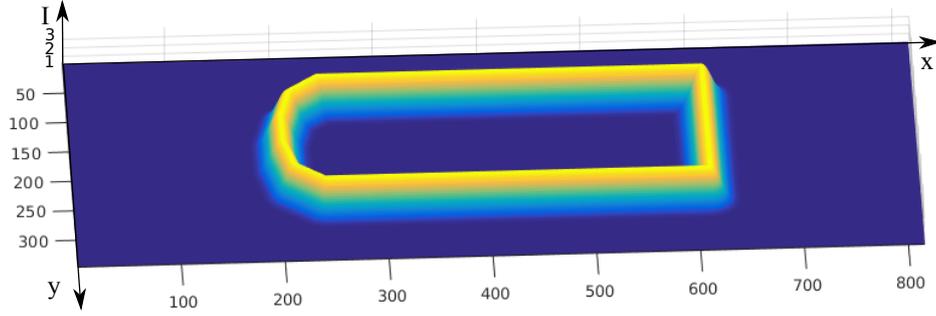


FIGURE 4.5. – *Champ de vraisemblance donnant la probabilité qu'un point LiDAR soit sur le véhicule perçu. Cette vraisemblance est déterminée à partir d'une gaussienne calculée en fonction de la distance au modèle polygonal. Cette figure montre la vraisemblance l en fonction de chaque cellule de 1 cm de côté.*

à gauche de la grille et $area(1,2)$ et $area(2,2)$ sont respectivement l'abscisse et l'ordonnée du point en haut à droite. ds représente le pas de discrétisation de la grille.

Une pondération est calculée dans le cas où les points LiDAR tombent dans la zone définie pour la grille de vraisemblance :

$$area(1,1) < p_{ikx} < area(1,2) \text{ ET } area(2,1) < p_{iky} < area(2,2). \quad (4.11)$$

Pour obtenir un score lié à chaque pose, nous multiplions un poids initial ($1/K$) par le produit des vraisemblances trouvées pour chaque cellule sur laquelle un point LiDAR tombe (Alg. 4.3 lignes 6-12) :

$$\omega_k = \frac{1}{K} \prod_{i=1}^{N_P} l_{ik}, \quad (4.12)$$

où l_{ik} correspond à la vraisemblance du point LiDAR i lié à la pose k déterminée à partir des indices w et h et de la grille de vraisemblance.

Si les points LiDAR sont en dehors de cette zone, la pondération ω_k est considérée nulle.

La pose estimée par cet algorithme est la moyenne de l'ensemble des poses pondérées par leur poids (Alg. 4.3 ligne 16) :

$${}^e \hat{q}_t = \sum_{k=1}^K \omega_k {}^e q_{tk}. \quad (4.13)$$

Pour calculer la matrice de covariance, la formule suivante est utilisée (Alg. 4.3 ligne 17) :

$$\Sigma_{e \hat{q}_t} = \sum_{k=1}^K \omega_k ({}^e q_{tk} - {}^e \hat{q}_t) ({}^e q_{tk} - {}^e \hat{q}_t)^T. \quad (4.14)$$

Cette matrice de covariance est la somme des carrés scalaires des différences entre les poses générées aléatoirement et la pose moyenne pondérée par leur poids.

4. Pose inter-véhicules à partir de mesures LiDAR

Algorithme 4.3 *Méthode projetant les points LiDAR liés aux poses générées aléatoirement sur un champ de vraisemblance. Pour chaque pose, un poids est alors déduit de ces vraisemblances afin de calculer une pose et une matrice de covariance à partir des poses pondérées.*

```

1:  $\{e\hat{q}_t, e\hat{\Sigma}_t\} = \mathbf{randSamplesLocalization}(\{p_i, i \in [1..N_P]\}, field, area, ds, e q_t)$ 
   # Entrée : points LiDAR, champ de vraisemblance, surface et pas de discrétisation de celui-ci et pose relative initiale
   # Sortie : pose relative estimée et matrice de covariance associée
2: for  $k \in [1..K]$  do # Génération de K poses relatives aléatoires
3:    $q_k = e q_t + \mathbf{rand}(\sigma_x, \sigma_y, \sigma_\theta)$ 
4:   for  $i \in [1..N_P]$  do
5:      $p_{ik} = \ominus e q_t \oplus p_i$  # Passage des points LiDAR dans le repère du véhicule perçu
6:     if  $area(1, 1) < p_{ikx} < area(1, 2)$ 
7:       and  $area(2, 1) < p_{iky} < area(2, 2)$  then # Si le point est dans le champ de vraisemblance
8:          $[w\ h]^T = \begin{bmatrix} \lfloor (p_{ikx} - area(1, 1)) / ds \rfloor \\ \lfloor (p_{iky} - area(2, 1)) / ds \rfloor \end{bmatrix}$  # Indices du point dans la grille de vraisemblance
9:          $l_{ik} = field(w, h)$  # Vraisemblance du point
10:         $\omega_k = \omega_k l_{ik}$  # Mise à jour du score
11:       else then
12:          $\omega_k = 0$  # Score nul si le point est à l'extérieur du champ de vraisemblance
13:       end if
14:     end for
15:   end for
16:    $e\hat{q}_t = \sum_{k=1}^K \omega_k q_k$  # Estimation de la pose relative par moyenne pondérée
17:    $e\hat{\Sigma}_t = \sum_{k=1}^K \omega_k (q_k - e\hat{q}_t)(q_k - e\hat{q}_t)^T$  # Estimation de la matrice de covariance associée
18: end

```

Nous avons implémenté cette méthode dans un programme Matlab. Pour une pose relative donnée, la figure 4.6 montre les poses aléatoires générées et la pose finale estimée par la méthode d'échantillonnage aléatoire avec l'ellipse et le cône d'incertitude en vert. Le modèle final obtenu par l'algorithme est représenté en vert.

Pour une étape d'estimation convenable, de nombreuses poses sont nécessaires (au minimum 1000). Le temps de calcul est de l'ordre de 176 ms en moyenne, ce qui est très long. La fréquence d'un LiDAR étant de l'ordre de 10 Hz, seulement un scan sur deux pourrait être utilisé.

Cet algorithme peut aussi avoir des problèmes numériques, notamment lors de l'estimation de la matrice de covariance. Lorsqu'une pose est un peu éloignée de la pose réelle, il suffit d'un point LiDAR loin de la moyenne de la gaussienne du champ de vraisemblance pour que son poids devienne très faible ($\omega_k \ll \frac{1}{K}$). Si une très faible partie des poses sont proches de la pose réelle, on doit alors calculer les sommes pondérées pour la pose et la matrice de covariance avec quelques pondérations très

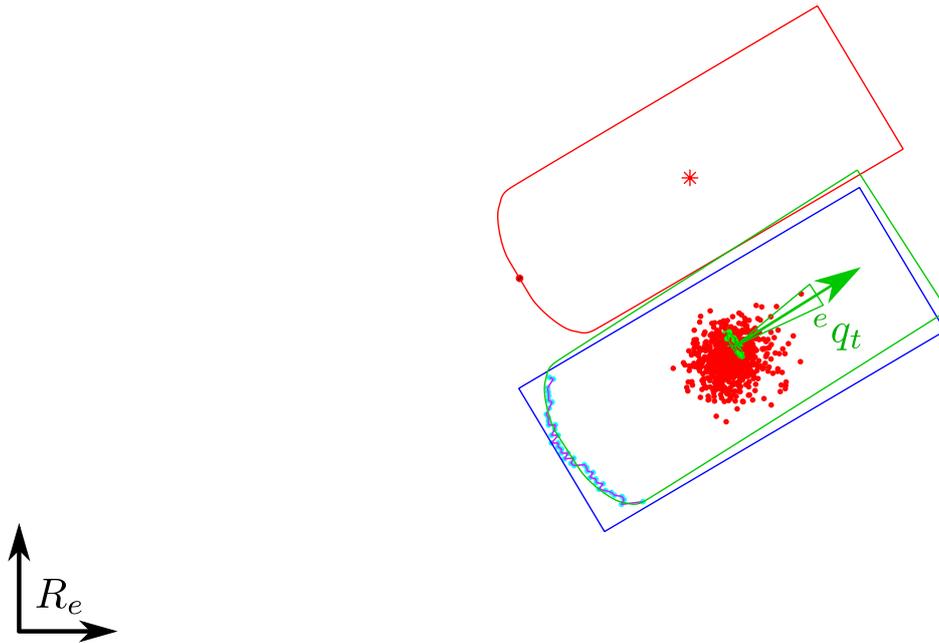


FIGURE 4.6. – La méthode par échantillonnage aléatoire génère des poses représentées par le nuage de points, en rouge pour celles qui ont un poids faible et en vert pour celles qui ont un poids suffisamment important. Elle en déduit une pose pondérée ${}^e q_t$ avec une matrice de covariance, représentée par une ellipse et un cône d'incertitude en vert.

élevées ($\omega_k \gg \frac{1}{K}$) et toutes les autres pondérations très faibles ($\omega_k \ll \frac{1}{K}$). Des problèmes numériques peuvent alors apparaître et les poses se trouvant loin de la pose réelle, si elles sont nombreuses, peuvent alors avoir une influence plus importante qu'elles ne devraient, faussant ainsi le calcul de la pose et de la matrice de covariance. Cet algorithme nécessite une pose initiale suffisamment proche de la pose réelle et des poses correctement réparties pour éviter ces problèmes.

Pour ces raisons, nous cherchons dans la suite de ce chapitre des méthodes alternatives plus rapides⁴ et plus robustes.

4.6. Localisation par optimisation itérative de type ICP

4.6.1. Présentation de la méthode

L'algorithme *Iterative Closest Point* (ICP) est souvent utilisé pour mettre en correspondance deux scans LiDAR afin de trouver la pose relative entre les deux [Censi, 2008]. On utilise ici cet algorithme pour déterminer la pose relative entre un scan LiDAR et le modèle polygonal du véhicule cible (Alg. 4.4).

L'ICP est un algorithme itératif qui, pour chaque itération k , va débiter en mettant en correspondance les points LiDAR p_i avec les éléments géométriques du modèle M

⁴. Les méthodes d'optimisation itérative présentées dans la section suivante sont environ dix fois plus rapides.

4. Pose inter-véhicules à partir de mesures LiDAR

Algorithme 4.4 *L'optimisation itérative met en correspondance les points LiDAR avec le modèle et cherche ensuite la pose permettant de minimiser la somme des carrés des distances entre chaque point et le modèle pour chaque itération. Les itérations s'arrêtent lorsque la variation de la somme devient faible.*

```

1:  $\{^e\hat{q}_t, \Sigma_{^e\hat{q}_t}\} = \mathbf{ICP}(\{p_i, i \in [1..N_P]\}, model, ^e q_t = [^e x_t \ ^e y_t \ ^e \theta_t]^T)$ 
   # Entrée : points LiDAR, modèle polygonal et pose relative initiale
   # Sortie : pose relative estimée et matrice de covariance associée
2:  ${}^t q_{e0} = \ominus ^e q_t$ 
3: for  $i \in [1..N_P]$  do   # Passage des points LiDAR dans le repère du véhicule
   perçu
4:    $p_{i0} = \ominus ^e q_t \oplus p_i$ 
5: end for
6:  $E_0 = 10000$ 
7:  $E_1 = 1000000$ 
8:  $k = 1$ 
   while  $E_k - E_{k-1} > \epsilon$  do   # Tant que la variation de l'erreur est plus grande
   qu'un seuil
9:    $k = k + 1$ 
10:   $matching = match(\{p_{i_{k-1}}, i \in [1..N_P]\}, model)$  # Mise en correspondance
   des points LiDAR avec le modèle
11:   $\{\Delta q_k, \Sigma_{^t\hat{q}_{ek}}, E_k\} = minimization(matching, ^e\hat{q}_{tk-1})$  # Minimisation des
   distances relatives aux correspondances
12:   ${}^t\hat{q}_{ek} = \Delta q_k \oplus {}^t\hat{q}_{ek-1}$  # Correction de la pose relative
13:  for  $i \in [1..N_P]$  do
14:     $p_{ik} = {}^t\hat{q}_{ek} \oplus p_i$  # Rapprochement des points LiDAR du modèle
15:  end for
16: end while
17:  $^e\hat{q}_t = \ominus {}^t\hat{q}_{ek}$  # Inversion de la pose relative
18:  $\Sigma_{^e\hat{q}_t} = J_{\ominus} \Sigma_{^t\hat{q}_{ek}} J_{\ominus}^T$  # Inversion de sa matrice de covariance
19: end

```

(Alg. 4.4 ligne 10) :

$$matching = match(\{p_{i_{k-1}}, i \in [1..N_P]\}, M), \quad (4.15)$$

où p_{ik} est la position du $i^{\text{ème}}$ point LiDAR à l'itération k . Il utilise un algorithme de minimisation pour trouver le déplacement à appliquer à la pose relative entre le repère des points LiDAR (le repère mobile du égo-véhicule) et le repère du modèle (le repère mobile du véhicule cible) qui minimise l'ensemble des distances de chaque point LiDAR au modèle (Alg. 4.4 ligne 11). À partir de l'erreur obtenue et de la mise en correspondance, une matrice de covariance peut être calculée :

$$\{\Delta q_k, \Sigma_{^t\hat{q}_{ek}}, E_k\} = minimization(matching, ^e\hat{q}_{tk-1}), \quad (4.16)$$

où Δq_k est le déplacement à appliquer à la pose relative, $\Sigma_{^t\hat{q}_{ek}}$ est la matrice covariance de ${}^t\hat{q}_{ek}$ et E_k est l'erreur de la fonction de coût à l'itération k . La pose relative est alors corrigée en utilisant ce déplacement (Alg. 4.4 ligne 12) :

$${}^t\hat{q}_{ek} = \Delta q_k \oplus {}^t\hat{q}_{ek-1}. \quad (4.17)$$

4. Pose inter-véhicules à partir de mesures LiDAR

On obtient alors une nouvelle pose et une matrice de covariance associée. La pose relative entre le repère du modèle du véhicule cible et le repère du LiDAR du égo-véhicule étant corrigée, les points LiDAR se rapprochent du modèle (Alg. 4.4 lignes 11-13) :

$$p_{ik} = {}^t\hat{q}_{ek} \oplus p_i. \quad (4.18)$$

La pose relative et sa matrice de covariance sont ensuite ramenées dans le repère du égo-véhicule (Alg. 4.4 ligne 17-18) :

$${}^e\hat{q}_{tk} = \ominus {}^t\hat{q}_{ek} \quad (4.19)$$

$$\Sigma_{e\hat{q}_t} = J_{\ominus} \Sigma_{t\hat{q}_e} J_{\ominus}^T \quad (4.20)$$

L'erreur de la fonction de coût E_k peut être divisée par le nombre de points LiDAR. Ainsi, l'erreur obtenue est indépendante de ce nombre et représente la distance moyenne au carré entre les points LiDAR et le modèle.

Cette nouvelle erreur est utilisée pour évaluer le résultat final. Si la variation entre cette erreur et l'erreur à l'itération précédente varie peu (inférieure par exemple à 1 cm^2 entre deux itérations), on suppose que l'optimisation a suffisamment convergé et on arrête l'algorithme. Sinon, on commence une nouvelle itération (Alg. 4.4 ligne 5).

Cet algorithme utilisant plusieurs itérations, on peut améliorer la correspondance entre les points LiDAR et le modèle. En effet, si des points LiDAR ne sont pas suffisamment proches du modèle, des mauvaises correspondances peuvent être faites. La minimisation permet alors de réduire les distances entre les points et le modèle, néanmoins les points LiDAR n'étant pas associés aux bons éléments du modèle (les arêtes ou les sommets), ne sont pas le plus proche possible du modèle. L'itération suivante permet de corriger la correspondance et de rapprocher les points LiDAR encore un peu plus du modèle.

Nous estimons ensuite l'incertitude en déterminant la matrice de covariance de l'erreur que l'on suppose proportionnelle à la fonction de coût E ainsi qu'à l'inverse de sa matrice Hessienne comme proposé par [Bengtsson et Baerveldt, 2003] :

$$\Sigma_{t\hat{q}_e} = 2 \frac{E(\Delta q)}{N_P - K} \left(\frac{\partial^2 E}{\partial \Delta q^2}(\Delta q) \right)^{-1}, \quad (4.21)$$

où k est la dimension de Δq , i.e., $K = 3$.

La figure 4.7 montre la pose corrigée par cet algorithme avec l'ellipse et le cône d'incertitude en vert.

4.6.2. Correspondances entre les points LiDAR et le modèle polygonal

On peut calculer la distance minimale d'un point LiDAR au modèle polygonal en prenant la plus courte des distances entre le point LiDAR et chaque segment du modèle. Dans le cas de cette méthode par optimisation itérative, on cherche à construire un problème de minimisation qui soit rapide à résoudre, de préférence avec une solution analytique. L'utilisation d'une distance point à segment introduit des conditions pour savoir si une distance euclidienne ou une distance orthogonale doit être utilisée. Le problème de minimisation avec cette distance risque ainsi d'être plus complexe à résoudre. Pour éviter d'avoir ces conditions, on choisit une distance

4. Pose inter-véhicules à partir de mesures LiDAR



FIGURE 4.7. – *La méthode par optimisation itérative permet de déterminer une nouvelle pose avec sa matrice de covariance, en vert.*

(euclidienne ou orthogonale) pour chaque point LiDAR, qui ne change pas pendant toute la phase de minimisation. Un point LiDAR peut alors être associé au modèle par différentes méthodes, quatre correspondances différentes sont comparées.

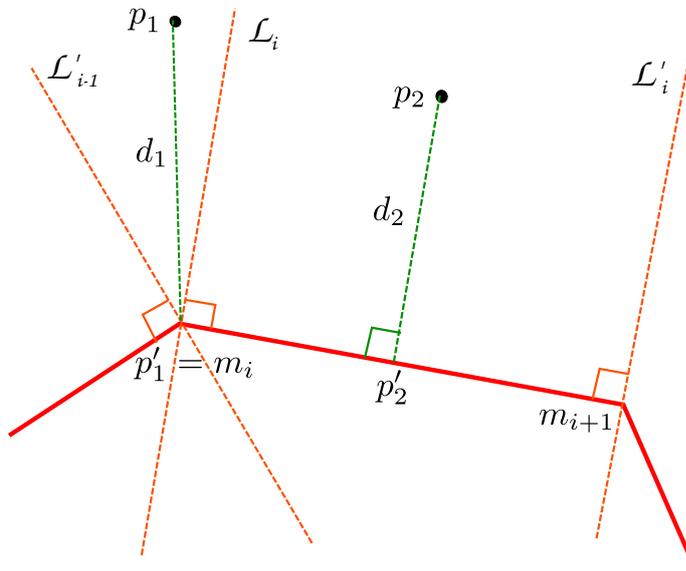


FIGURE 4.8. – *Distance minimale d'un point LiDAR à un segment d'un modèle pour deux points différents.*

La première méthode est la correspondance point à point (ICP). Pour chaque point LiDAR p_i , on cherche simplement le sommet m_j du modèle le plus proche en utilisant la distance euclidienne (Fig. 4.9a). Cette première méthode ne met pas en correspondance la partie du modèle réellement la plus proche du point LiDAR. Le point du modèle mis en correspondance peut même être très éloigné de la partie la plus proche si le modèle polygonal n'est pas suffisamment discrétisé.

Pour trouver le point le plus proche, on peut faire une correspondance point à projection (Fig. 4.9b) (ICPP). On cherche alors le point du modèle le plus proche du point LiDAR. On utilise la plus courte distance du point LiDAR aux segments du modèle, comme illustré dans la figure 4.8, et on prend le point du modèle associé à cette distance. On obtient p'_1 pour p_1 dans le cas d'une distance euclidienne et p'_2

4. Pose inter-véhicules à partir de mesures LiDAR

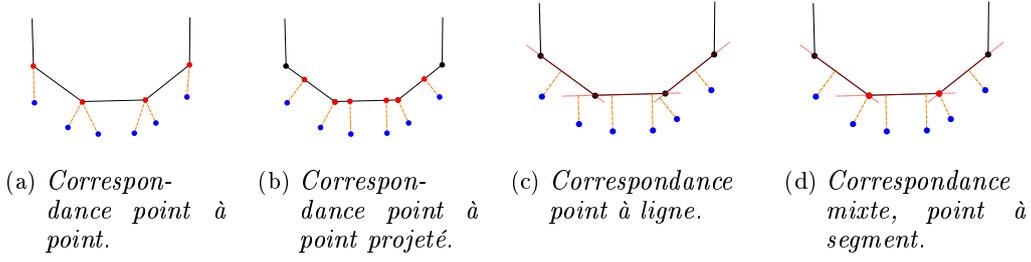


FIGURE 4.9. – *Différentes correspondances entre les points LiDAR et le modèle polygonal.*

le point de la projection orthogonale de p_2 sur le segment le plus proche, dans le cas d'une distance orthogonale à un segment.

Une correspondance point à ligne peut aussi être utilisée (PLICP). On ne cherche plus le point du modèle le plus proche du point LiDAR, mais le segment (Fig. 4.9c). On associe ensuite le point LiDAR avec la droite passant par ce segment. Dans le cas où la distance la plus courte est celle par rapport à un sommet, deux segments sont aussi proches du point LiDAR. Le choix du segment associé dans le cas d'une distance euclidienne est un choix arbitraire puisqu'une vraie correspondance point à ligne orthogonale sans tomber en dehors d'un segment n'est plus possible. On choisit le segment $(m_j; m_{j+1})$ lorsque le sommet m_j est le plus proche. Un autre choix pourrait être de tracer une bissectrice à chaque sommet m_j et d'associer le point p_i au segment se trouvant du même côté de cette bissectrice.

La dernière correspondance que nous avons étudiée est une version mixte (mixICP) : une correspondance point à ligne dans le cas d'une distance orthogonale et une point à point dans le cas d'une distance euclidienne (Fig. 4.9d). Les points LiDAR peuvent ainsi être mis en correspondance avec deux types de géométries, des points et des droites. La correspondance avec un point est souvent moins réaliste que celle avec une droite. En effet, tant que la correspondance ne change pas, pendant une minimisation par exemple, la distance au point est constante sur un cercle autour de ce point alors que pour une correspondance point à ligne la distance est constante sur une droite parallèle à celle mise en correspondance. Ainsi, dans le cas d'une correspondance avec une droite représentant un bord de véhicule, le point LiDAR pourra être déplacé le long de ce bord sans influence sur l'erreur. Au contraire, dans le cas d'une correspondance avec un point, le point LiDAR sera contraint autour de ce point. Ainsi, une correspondance avec une ligne permet de converger en moins d'itérations. Néanmoins, étant donné que cette droite n'est pas bornée, l'algorithme peut aussi diverger. En pratique, si la pose initiale est proche de la vraie pose et que le modèle a suffisamment de segments, la divergence n'est pas observée.

Dans le cas d'une correspondance entre un point LiDAR p_i et un point du modèle m_j , un sommet ou une projection du point LiDAR, la distance vaut

$$d(M, p_i; q) = \|\mathop{\ominus}^e q_t \oplus m_j - p_i\|^2 = \|m_j - \mathop{\ominus}^e q_t \oplus p_i\|^2. \quad (4.22)$$

Les minimisations liées à la correspondance entre le point LiDAR p_i et le point du modèle m_j en utilisant la transformation liée à $\mathop{\ominus}^e q_t$ (en déplaçant le modèle) ou celles liées à ${}^t q_e = \mathop{\ominus}^e q_t$ (en déplaçant les points LiDAR) sont équivalentes.

4. Pose inter-véhicules à partir de mesures LiDAR

Dans le cas d'une correspondance entre un point LiDAR p_i et une arête $(m_j; m_{j+1})$ avec une normale unitaire n_j , la distance vaut

$$d(M, p_i; q) = ((m_j - \ominus^e q_t \oplus p_i) \cdot n_j)^2. \quad (4.23)$$

On note m_i et n_i le sommet et la normale à l'arête du modèle associée au point LiDAR p_i après la correspondance.

4.6.3. Minimisation par calcul des racines d'un polynôme

Dans un premier temps, l'algorithme *Point to Line ICP* (PLICP) de [Censi, 2008] a été utilisé (Alg. 4.5).

Cet algorithme converge en beaucoup moins d'itérations qu'un ICP avec une correspondance point à point et une forme analytique existe pour résoudre la minimisation. Cet algorithme est basé sur des correspondances point à ligne, même s'il est possible d'utiliser des correspondances point à point. Le problème 4.3 utilise alors les distances 4.22 ou 4.23.

Pour appliquer la méthode, les non-linéarités introduites par les fonctions cosinus et sinus de la rotation doivent être éliminées.

En faisant le changement de variable

$$\Delta q = [\Delta x \ \Delta y \ \Delta \theta]^T, \quad (4.24)$$

à

$$q_{4D} = [\Delta x \ \Delta y \ \Delta c \ \Delta s]^T, \quad (4.25)$$

où Δc et Δs représentent $\cos(\Delta \theta)$ et $\sin(\Delta \theta)$, respectivement, sous la contrainte

$$\Delta c^2 + \Delta s^2 = 1, \quad (4.26)$$

on peut écrire la somme à minimiser sous la forme :

$$\begin{cases} \min_{q_{4D}} & \sum_{i=1}^N (M_i q_{4D} - m_i)^T C_i (M_i q_{4D} - m_i) \\ \text{subject to} & q_{4D}^T W q_{4D} = 1 \end{cases}, \quad (4.27)$$

avec

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.28)$$

$$M_i = \begin{bmatrix} 1 & 0 & p_{ix} & -p_{iy} \\ 0 & 1 & p_{iy} & p_{ix} \end{bmatrix}, \quad (4.29)$$

où

$$p_i = [p_{ix} \ p_{iy}]^T, \quad (4.30)$$

et

$$C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (4.31)$$

pour une correspondance point à point ou

$$C_i = n_i n_i^T, \quad (4.32)$$

4. Pose inter-véhicules à partir de mesures LiDAR

Algorithme 4.5 *Minimisation par calcul des racines d'un polynôme.*

```

1:  $\{\Delta\hat{q}, \Sigma_{\Delta\hat{q}}, E\} = \mathbf{minPoly}(\text{match} = \{p_i, m_i, n_i, i \in [1..N_P]\})$ 
   # Entrée : correspondance entre les points LiDAR et les points et normales du
   # modèle
   # Sortie : écart de pose, matrice de covariance et erreur
2:  $M = 0_4$ 
3:  $g = 0_{4,1}$ 
4:  $c = 0$ 
5: for  $i \in [1..N_P]$  do
   # Détermination du polynôme  $P(\lambda)$  à partir de la correspondance
6:   if pointToPointMatching
7:      $C_i = I_2$ 
8:   else if pointToLineMatching
9:      $C_i = n_i n_i^T$ 
10:   end if
11:    $M_i = [I_2 p_i R(\frac{\pi}{2}) p_i]$ 
12:    $M = M + M_i^T C_i M_i$ 
13:    $g = g - 2m_i^T C_i M_i$ 
14:    $c = c + m_i^T C_i m_i$ 
15: end for
16:  $A = 2M (1 : 2, 1 : 2)$ 
17:  $B = 2M (1 : 2, 3 : 4)$ 
18:  $D = 2M (3 : 4, 3 : 4)$ 
19:  $S = D - B^T A^{-1} B$ 
20:  $S^A = \det(S) S^{-1}$ 
21:  $P(\lambda) = 4g^T \begin{bmatrix} A^{-1} B B^T A^{-T} & -A^{-1} B \\ \text{(symétrique)} & I \end{bmatrix} g \cdot \lambda^2 + 4g^T \begin{bmatrix} A^{-1} B S^A B^T A^{-T} & -A^{-1} B S^A \\ \text{(symétrique)} & S^A \end{bmatrix} g \cdot \lambda + g^T \begin{bmatrix} A^{-1} B S^A B^T A^{-T} & -A^{-1} B S^A B^T A^{-T} \\ \text{(symétrique)} & S^A S^A \end{bmatrix} g - \det(S + 2\lambda I)^2$ 
22:  $\{\lambda_j\} = \text{roots}(P)$  # Calcul des racines du polynôme  $P(\lambda)$ 
23:  $W = \begin{bmatrix} 0_2 & 0_2 \\ 0_2 & I_2 \end{bmatrix}$ 
24: for  $j \in [1..4]$  do
25:   if isreal( $\lambda_j$ ) # Si la racine est réelle
26:      $\Delta q_j = -(2M + 2\lambda_j W)^{-1} g$  # Estimation de l'écart de pose à corriger
27:      $E_j = \Delta q_j^T M \Delta q_j + g^T \Delta q_j + \lambda_j (\Delta q_j^T W \Delta q_j - 1) + c$  # Calcul de l'erreur
28:   end if
29: end for
30:  $\{E, j_{\min}\} \min(E_j)$  # Détermination de l'erreur la plus petite
31:  $\Delta q_{4D} = \Delta q_{j_{\min}}$  # Écart de pose 4D liée à cette erreur
32:  $\Delta\hat{q} = [\Delta q_x \Delta q_y \arctan 2(\Delta q_s, \Delta q_c)]^T$  # Écart de pose 3D ( $x, y, \theta$ )
33:  $\Sigma = \frac{E(\Delta\hat{q}_{4D})}{N_P - 3} \left( \frac{\partial^2 E(\Delta\hat{q}_{4D})}{\partial \Delta\hat{q}^2} \right)^{-1}$  # Estimation de la matrice de covariance

```

4. Pose inter-véhicules à partir de mesures LiDAR

pour une correspondance point à ligne où n_i est la normale au segment i (Alg. 4.5 lignes 6-10).

On a alors :

$$\begin{aligned} \sum_{i=1}^N (M_i q_{4D} - m_i)^T C_i (M_i q_{4D} - m_i) &= \sum_{i=1}^N (q_{4D}^T M_i^T C_i M_i q_{4D} \\ &\quad - 2m_i^T C_i M_i q_{4D} + m_i^T C_i m_i) \\ &= q_{4D}^T M q_{4D} + g^T q_{4D} + c, \end{aligned} \quad (4.33)$$

avec

$$M = \sum_{i=1}^N M_i^T C_i M_i, \quad (4.34)$$

$$g^T = -2 \sum_{i=1}^N m_i^T C_i M_i, \quad (4.35)$$

et

$$c = \sum_{i=1}^N m_i^T C_i m_i \quad (4.36)$$

(Alg. 4.5 lignes 11-14).

La somme à minimiser devient un problème de minimisation quadratique sous contraintes :

$$\begin{cases} \min_{q_{4D}} & E(q_{4D}) = q_{4D}^T M q_{4D} + g^T q_{4D} + c \\ \text{subject to} & q_{4D}^T W q_{4D} = 1 \end{cases} \quad (4.37)$$

À partir de ce problème, un multiplicateur de Lagrange λ est utilisé pour obtenir la fonction de minimisation :

$$L(q_{4D}) = q_{4D}^T M q_{4D} + g^T q_{4D} + c + \lambda (q_{4D}^T W q_{4D} - 1). \quad (4.38)$$

Tous les minima et maxima locaux annulent la dérivée de L en fonction de q_{4D} :

$$\frac{\partial L}{\partial q_{4D}}(q_{4D}) = 2q_{4D}^T M + g^T + 2\lambda q_{4D}^T W = 0^T. \quad (4.39)$$

q_{4D} est alors isolé de cette équation :

$$q_{4D} = -(2M + 2\lambda W)^{-T} g. \quad (4.40)$$

En utilisant la contrainte, on obtient l'équation :

$$g^T (2M + 2\lambda W)^{-1} W (2M + 2\lambda W)^{-T} g = 1. \quad (4.41)$$

$(\cdot)^{-T}$ correspond à la transposée de l'inverse.

On transforme alors cette équation en polynôme d'ordre quatre en λ en décomposant $(2M + 2\lambda W)^{-1}$ en sous matrices (Alg. 4.5 lignes 16-18) :

$$\begin{aligned} (2M + 2\lambda W)^{-1} &= \begin{bmatrix} A & B \\ B^T & D + 2\lambda I \end{bmatrix}^{-1} \\ &= \begin{bmatrix} * & -A^{-1} B Q^{-1} \\ & Q^{-1} \end{bmatrix}^{-1}, \end{aligned} \quad (4.42)$$

4. Pose inter-véhicules à partir de mesures LiDAR

avec

$$Q = D - B^T A^{-1} B + 2\lambda I = S + 2\lambda I. \quad (4.43)$$

La contrainte peut alors être écrite sous la forme (Alg. 4.5 :19) :

$$g^T \begin{bmatrix} A^{-1} B Q^{-1} Q^{-T} B^T A^{-T} & -A^{-1} B Q^{-1} Q^{-T} \\ \text{(symétrique)} & Q^{-1} Q^{-T} \end{bmatrix} g = 1. \quad (4.44)$$

Q peut être écrit sous la forme :

$$Q = S + 2\lambda I = \frac{S^A + 2\lambda I}{p(\lambda)}, \quad (4.45)$$

où

$$S^A = \det(S) S^{-1}, \quad (4.46)$$

et

$$p(\lambda) = \det(S + 2\lambda I), \quad (4.47)$$

(Alg. 4.5 ligne 20).

On a donc :

$$\begin{aligned} Q^{-1} Q^{-T} &= \frac{(S^A + 2\lambda I)(S^A + 2\lambda I)^T}{p(\lambda)^2} \\ &= \frac{4\lambda^2 I + 4\lambda S^A + S^A S^A{}^T}{p(\lambda)^2}. \end{aligned} \quad (4.48)$$

En injectant $Q^{-1} Q^{-T}$ dans la contrainte, on obtient alors un polynôme d'ordre 4 en λ (Alg. 4.5 :21) :

$$\begin{aligned} P(\lambda) &= 4g^T \begin{bmatrix} A^{-1} B B^T A^{-T} & -A^{-1} B \\ \text{(symétrique)} & I \end{bmatrix} g \cdot \lambda^2 \\ &+ 4g^T \begin{bmatrix} A^{-1} B S^A B^T A^{-T} & -A^{-1} B S^A \\ \text{(symétrique)} & S^A \end{bmatrix} g \cdot \lambda \\ &+ g^T \begin{bmatrix} A^{-1} B S^A{}^T S^A B^T A^{-T} & -A^{-1} B S^A{}^T S^A \\ \text{(symétrique)} & S^A{}^T S^A \end{bmatrix} g \\ &- p(\lambda)^2. \end{aligned} \quad (4.49)$$

Les racines de ce polynôme correspondent aux valeurs de λ des différents extrema locaux. À partir de λ , la valeur de q_{4D} est calculée en utilisant l'équation 4.40. La fonction de coût associée à cette pose peut aussi être déterminée à partir de l'équation 4.38. Le minimum global est alors trouvé en calculant la fonction de coût pour chaque racine (Alg. 4.5 lignes 24-31). Pour obtenir la pose en dimension trois, on peut alors utiliser la formule (Alg. 4.5 ligne 32) :

$$\Delta q = [\Delta x \ \Delta y \ \Delta \theta]^T = \left[\Delta x \ \Delta y \ \text{atan} \left(\frac{\Delta s}{\Delta c} \right) \right]^T, \quad (4.50)$$

où $q_{4D} = [\Delta x \ \Delta y \ \Delta c \ \Delta s]^T$.

La matrice de covariance peut alors être calculée à partir de l'équation 4.21 (Alg. 4.5 ligne 33) :

$$\Sigma = 2 \frac{E(q_{4D})}{N - K} \left(\frac{\partial^2 E(q_{4D})}{\partial \Delta q^2} \right)^{-1}. \quad (4.51)$$

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.1. – Résultats de l’algorithme de minimisation par calcul des racines d’un polynôme en termes de précision avec la moyenne de la norme de l’erreur de position relative $\|\bar{\epsilon}_p\|$ et en termes de taux de couverture C . Ce taux de couverture est calculé pour un risque de 5%.

	ICP	ICPP	PLICP	mixICP
$\ \bar{\epsilon}_p\ $ (cm)	8,2	7,8	13,7	11,0
C (%)	85,5	58,8	69,8	70,0

La table 4.1 montre que cette minimisation donne des résultats de simulation qui ne sont pas consistants pour aucune des correspondances. L’erreur de position moyenne $\|\bar{\epsilon}_p\|$ reste toutefois suffisamment précise pour chacune des correspondances. Le protocole expérimental de cette simulation sera décrit en détails dans la section 4.7.

4.6.4. Minimisation par matrice pseudo-inverse

Cette méthode est basée sur l’algorithme de [Low, 2004] développé pour une correspondance point à plan. Afin d’obtenir une méthode en deux dimensions, nous l’avons simplifiée pour utiliser des correspondances point à ligne (Alg. 4.6). Nous l’avons ensuite généralisée pour des correspondances point à point ou point à point projeté, ainsi que pour des correspondances mixtes.

Algorithme 4.6 *Minimisation par matrice pseudo-inverse.*

```

1:  $\{\Delta\hat{q}, \Sigma_{t_{\hat{q}_e}}, E\} = \mathbf{minPinv}(match = \{p_i, m_i, n_i, i \in [1..N_P]\})$ 
   # Entrée : correspondance entre les points LiDAR et les points et normales du
   # modèle
   # Sortie : écart de pose, matrice de covariance et erreur
2:  $A = [\emptyset]$ 
3:  $b = [\emptyset]$ 
4: for  $i \in [1..N_P]$  do
   # Détermination de la matrice  $A$  et du vecteur  $b$  à partir de la correspondance
5:   if pointToPointMatching
6:      $A = \begin{bmatrix} A \\ [I_2 R(\frac{\pi}{2}) p_i] \end{bmatrix}$ 
7:      $b = \begin{bmatrix} b \\ m_i - p_i \end{bmatrix}$ 
8:   else if pointToLineMatching
9:      $A = \begin{bmatrix} A \\ n_i^T [I_2 R(\frac{\pi}{2}) p_i] \end{bmatrix}$ 
10:     $b = \begin{bmatrix} b \\ n_i^T (m_i - p_i) \end{bmatrix}$ 
11:   end if
12: end for
13:  $\Delta\hat{q} = \mathbf{pinv}(A) b$  # Estimation de l’écart de pose à corriger
14:  $E = A\Delta\hat{q} - b$  # Calcul de l’erreur
15:  $\Sigma_{t_{\hat{q}_e}} = \frac{E}{N_P - 3} (A^T A)^{-1}$  # Estimation de la matrice de covariance

```

4. Pose inter-véhicules à partir de mesures LiDAR

Le problème des non-linéarités des cosinus et sinus de la rotation est résolu dans cette minimisation en approchant au premier ordre la matrice de transformation homogène ΔT ⁵ :

$$\Delta T = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) & \Delta x \\ \sin(\Delta\theta) & \cos(\Delta\theta) & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -\Delta\theta & \Delta x \\ \Delta\theta & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.52)$$

On fait ici l'hypothèse des petits angles en supposant que l'angle varie peu d'une itération à la suivante.

Pour un point LiDAR, le produit scalaire $(\Delta T p_i - m_i) \cdot \vec{n}_i$ est écrit sous la forme d'une équation linéaire en Δx , Δy et $\Delta\theta$ grâce au produit de deux vecteurs en coordonnées homogènes :

$$\begin{aligned} (\Delta T p_i - m_i) \cdot n_i &= \left(\Delta T \begin{pmatrix} p_{ix} \\ p_{iy} \\ 1 \end{pmatrix} - \begin{pmatrix} m_{ix} \\ m_{iy} \\ 1 \end{pmatrix} \right) \cdot \begin{pmatrix} n_{ix} \\ n_{iy} \\ 0 \end{pmatrix} \\ &\approx n_{ix} \Delta x + n_{iy} \Delta y + (n_{iy} p_{ix} - n_{ix} p_{iy}) \Delta\theta \\ &\quad - (n_{ix} m_{ix} + n_{iy} m_{iy} - n_{ix} p_{ix} - n_{iy} p_{iy}). \end{aligned} \quad (4.53)$$

L'ensemble des $(\Delta T p_i - m_i) \cdot n_i$ pour les N_P points LiDAR est ensuite écrit sous la forme d'une fonction linéaire :

$$A \Delta q - b, \quad (4.54)$$

$$\text{où } A = \begin{bmatrix} n_{1x} & n_{1y} & n_{1y} p_{1x} - n_{1x} p_{1y} \\ \vdots & \vdots & \vdots \\ n_{N_P x} & n_{N_P y} & n_{N_P y} p_{N_P x} - n_{N_P x} p_{N_P y} \end{bmatrix}$$

$$\text{et } b = \begin{bmatrix} n_{1x} m_{1x} + n_{1y} m_{1y} - n_{1x} p_{1x} - n_{1y} p_{1y} \\ \vdots \\ n_{N_P x} m_{N_P x} + n_{N_P y} m_{N_P y} - n_{N_P x} p_{N_P x} - n_{N_P y} p_{N_P y} \end{bmatrix}$$

(Alg. 4.6 lignes 9-10).

Dans le cas d'une correspondance point à point :

$$\begin{aligned} \Delta T p_i - m_i &= \Delta T \begin{pmatrix} p_{ix} \\ p_{iy} \\ 1 \end{pmatrix} - \begin{pmatrix} m_{ix} \\ m_{iy} \\ 1 \end{pmatrix} \\ &\approx \begin{pmatrix} \Delta x - p_{iy} \Delta\theta + p_{ix} - m_{ix} \\ \Delta y + p_{ix} \Delta\theta + p_{iy} - m_{iy} \end{pmatrix}. \end{aligned} \quad (4.55)$$

On a pour cette correspondance :

$$A = \begin{bmatrix} 1 & 0 & -p_{1y} \\ 0 & 1 & p_{1x} \\ \vdots & \vdots & \vdots \\ 1 & 0 & -p_{N_P y} \\ 0 & 1 & p_{N_P x} \end{bmatrix} \quad \text{et} \quad b = \begin{bmatrix} m_{1x} - p_{1x} \\ m_{1y} - p_{1y} \\ \vdots \\ m_{N_P x} - p_{N_P x} \\ m_{N_P y} - p_{N_P y} \end{bmatrix} \quad (\text{Alg. 4.6 lignes 6-7})$$

5. En dimension 3 : $\Delta q = [\Delta x \ \Delta y \ \Delta z \ \Delta\alpha \ \Delta\beta \ \Delta\gamma]^T$ et $\Delta T \approx \begin{bmatrix} 1 & -\Delta\gamma & \Delta\beta & \Delta x \\ \Delta\gamma & 1 & -\Delta\alpha & \Delta y \\ -\Delta\beta & \Delta\alpha & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

4. Pose inter-véhicules à partir de mesures LiDAR

On peut alors écrire le problème de minimisation sous la forme :

$$\min_{\Delta q} \|A\Delta q - b\|^2. \quad (4.56)$$

On a alors un problème linéaire de moindres carrés qui peut être résolu par matrice pseudo-inverse en résolvant l'équation (4.54) égale à zéro (Alg. 4.6 ligne 13) :

$$\Delta \hat{q} = \text{pinv}(A)b. \quad (4.57)$$

On peut démontrer facilement que $(A^T A)^{-1} A^T$ est un pseudo-inverse simple de A . En effet, on peut écrire :

$$A\Delta \hat{q} = b, \quad (4.58)$$

$$A^T A\Delta \hat{q} = A^T b. \quad (4.59)$$

La matrice $A^T A$ étant symétrique réelle, elle est diagonalisable et inversible si aucune de ces valeurs propres n'est nulle :

$$\Delta \hat{q} = (A^T A)^{-1} A^T b. \quad (4.60)$$

Cette méthode présente toutefois des problèmes de stabilité numérique. Le pseudo-inverse de Moore-Penrose [Ben-Israel, 2002] est donc la fonction implémentée dans de nombreux langages. Ce pseudo-inverse est basé sur une décomposition en valeurs singulières (SVD) :

$$A = U\Sigma V^T, \quad (4.61)$$

où Σ est la matrice contenant les valeurs singulières sur sa diagonale et U et V sont des matrices unitaires telles que

$$U^T U = U U^T = I, \quad (4.62)$$

$$V^T V = V V^T = I. \quad (4.63)$$

Chaque valeur singulière σ_i est définie telle que

$$A v_i = \sigma_i u_i, \quad (4.64)$$

$$A^T u_i = \sigma_i v_i, \quad (4.65)$$

avec

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_3 \\ 0 & \cdots & 0 \end{bmatrix}, \quad (4.66)$$

$$U = [u_1 \cdots u_{N_P}], \quad (4.67)$$

$$V = [v_1 \cdots v_3]. \quad (4.68)$$

Le pseudo-inverse peut alors être calculé avec la formule suivante :

$$\text{pinv}(A) = V \text{pinv}(\Sigma) U^T, \quad (4.69)$$

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.2. – Résultats de l’algorithme de minimisation par matrice pseudo-inverse en termes de précision avec la moyenne de la norme de l’erreur de position relative $\|\bar{\epsilon}_p\|$ et en termes de taux de couverture C . Ce taux de couverture est calculé pour un risque de 5%.

	ICP	ICPP	PLICP	mixICP
$\ \bar{\epsilon}_p\ $ (cm)	8,2	7,8	11,5	10,8
C (%)	93,5	83,9	91,6	89,8

où

$$\text{pinv}(\Sigma) = \begin{bmatrix} \frac{1}{\sigma_1} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_3} & 0 \end{bmatrix}. \quad (4.70)$$

En pratique, nous utilisons les pseudo-inverses implémentés dans Matlab ou Python.

La matrice de covariance peut être alors calculée avec l’équation 4.21, en utilisant la matrice Hessienne (Alg. 4.6 ligne 15) :

$$\frac{\partial^2 E(\Delta q)}{\partial \Delta q^2} = \frac{\partial^2 \|A\Delta q - b\|^2}{\partial \Delta q^2} \quad (4.71)$$

$$\begin{aligned} &= \frac{\partial^2 (A\Delta q - b)^T (A\Delta q - b)}{\partial \Delta q^2} \\ &= \frac{\partial^2 (\Delta q^T A^T - b^T) (A\Delta q - b)}{\partial \Delta q^2} \\ &= \frac{\partial^2 (\Delta q^T A^T A\Delta q - b^T A\Delta q - \Delta q^T A^T b + b^T b)}{\partial \Delta q^2} \\ &= \frac{\partial (A^T A\Delta q - b^T A - A^T b)}{\partial \Delta q}, \end{aligned}$$

$$\frac{\partial^2 E(\Delta q)}{\partial q^2} = 2A^T A. \quad (4.72)$$

Avec des résultats numériques correspondant au même cas de figure que précédemment, on voit dans la table 4.2 que l’erreur moyenne de position ϵ_p donne des résultats similaires à la minimisation par calcul de racines d’un polynôme. La minimisation par matrice pseudo-inverse donne cependant des résultats qui sont plus consistants. L’ICP est consistant, mais étant donné que les distances utilisées sont les distances aux sommets du modèle polygonal, ce taux de couverture dépend fortement de la discrétisation du modèle. Ce taux de couverture va alors tendre vers celui de l’ICPP lorsque la discrétisation tendra vers l’infini. Il est donc plus intéressant d’utiliser cette minimisation par pseudo-inverse avec la correspondance PLICP qui donne le meilleur taux de couverture.

4.7. Résultats de simulation

Des résultats ont été obtenus par simulation, permettant ainsi de tester l'influence de nombreux paramètres sur ces algorithmes de localisation relative.

Comme seule la pose relative est importante, l'égo-véhicule a été placé à l'origine et le véhicule cible 10 m devant. Les poses utilisées pour les algorithmes ont été bruitées avec un bruit gaussien tel que les écarts types soient $\sigma_x = \sigma_y = 0.5$ m et $\sigma_\theta = 5^\circ$.

Le LiDAR a été simulé en prenant un modèle lissé possédant plus de segments que le modèle polygonal utilisé pour les algorithmes. Les problèmes de discrétisation du modèle peuvent ainsi être étudiés. Un bruit gaussien a été ajouté à la distance de ces points LiDAR simulés, avec $\sigma_\rho = 0.1$ m. L'angle n'a pas été bruité.

Les résultats sont comparés en termes de taux de couverture C en vérifiant l'inégalité :

$$\epsilon_{e_{q_t}}^T \Sigma_{e_{q_t}}^{-1} \epsilon_{e_{q_t}} < \chi_{3,1-\alpha}^2, \quad (4.73)$$

où $\chi_{3,1-\alpha}^2 = 7.81$ est le quantile d'une distribution du χ^2 avec trois degrés de liberté (pour un problème à trois dimensions x , y et θ) et pour un risque $\alpha = 5\%$ (pour une confiance $1 - \alpha = 95\%$).

Ces résultats sont aussi comparés en termes de précision à partir de la moyenne des normes des erreurs de position :

$$\|\bar{\epsilon}_p\| = \frac{1}{K} \sum_{k=1}^K \|{}^e p_t^* [k] - {}^e p_t [k]\|, \quad (4.74)$$

où K correspond au nombre d'échantillons, ${}^e p_t^* [k]$ est la pose relative ${}^e p_t$ de la vérité terrain pour l'échantillon k et ${}^e p_t [k]$ est l'estimée de cette pose pour le même échantillon.

4.7.1. Séquencement des étapes de l'algorithme

L'ordre classique est celui utilisé dans l'ICP. La correspondance est réalisée en premier et l'algorithme de minimisation est ensuite appliqué. À partir des résultats de la minimisation de l'erreur, la matrice de covariance est calculée et enfin l'erreur est évaluée afin de décider si l'algorithme a convergé.

Cet ordre donne une erreur et une matrice de covariance représentatives de la convergence de l'algorithme de minimisation.

Les correspondances étant fixes pendant toute la minimisation, un point mis en correspondance avec une ligne peut se déplacer au-delà des limites du segment et ainsi fausser l'erreur et la matrice de covariance réelle.

Avec une correspondance corrigée, l'erreur et la matrice de covariance ne sont pas directement celles de la minimisation. Une nouvelle étape de mise en correspondance est réalisée afin d'obtenir l'erreur et la matrice de covariance réellement représentatives de la nouvelle disposition des points LiDAR par rapport au modèle. Dans le cas où l'ICP a bien convergé et que la pose ne varie plus énormément, les deux erreurs et les deux matrices de covariances finales sont sensiblement similaires.

On voit que le PLICP de l'algorithme de minimisation par calcul de racine d'un polynôme donne des résultats différents avec la correspondance corrigée (Tab. 4.3). Cette méthode a en effet des problèmes pour converger convenablement et peut

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.3. – Moyenne de la norme de l'erreur de position en cm, erreur de minimisation moyenne en cm^2 et taux de couverture C pour le PLICP par calcul de racine d'un polynôme, dans le cas de l'ordre classique et de l'ordre avec correspondance corrigée.

	$\ \epsilon_p\ $	E	C
Ordre classique	22,4	69	64,6%
Ordre avec correspondance corrigée	13,7	80	69,8%

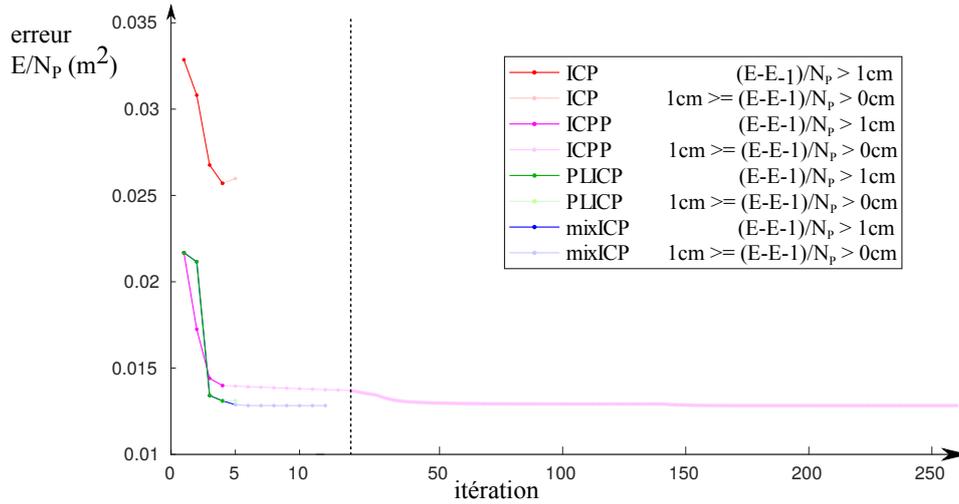


FIGURE 4.10. – Convergence des erreurs de la minimisation par matrice pseudo-inverse pour les quatre différentes correspondances. Les couleurs foncées correspondent à l'erreur avant la convergence, pour le critère de convergence arrêtant l'algorithme lorsque la variation de la fonction de coût E entre deux itérations passe en dessous d'un seuil, ici 1 cm^2 . Les couleurs claires montrent la suite de la convergence si le seuil de variation de l'erreur est descendu à 0 cm^2 . L'erreur illustrée ici représente la distance moyenne au carré entre un point LiDAR et le modèle.

parfois diverger. La correction de la correspondance avant le calcul de l'erreur et de la matrice de covariance permet alors d'améliorer les résultats.

4.7.2. Convergences des différentes méthodes itératives d'ICP

On voit dans la figure 4.10, un exemple, pour une époque donnée, les convergences des différentes correspondances. Lorsqu'un seuil de variation de la fonction de coût⁶ 1 cm^2 est utilisé, la convergence se fait pour un nombre d'itérations similaire pour les quatre correspondances. L'erreur de l'ICP est plus importante car elle ne prend pas les distances les plus courtes au modèle contrairement aux autres correspondances. L'ICPP semble parfois avoir convergé mais, après avoir varié très peu pendant plu-

6. La fonction de coût correspond à la somme des distances entre les points LiDAR et le modèle au carré. En la divisant par le nombre de points LiDAR, on obtient la distance moyenne au carré des points LiDAR au modèle.

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.4. – Nombre d'itérations nécessaires pour converger pour un seuil nul et un seuil de 1 cm^2 .

seuil	ICP	ICPP	PLICP	mixICP
0,0	6,9	243,3	5,6	8,1
1,0	4,0	4,3	3,3	3,6

TABLE 4.5. – Moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ en cm pour un seuil nul et un seuil de 1 cm^2 .

seuil	ICP	ICPP	PLICP	mixICP
0,0	8,2	10,4	11,5	10,9
1,0	8,2	7,8	11,5	10,8

sieurs itérations, il retrouve une erreur plus petite. Lorsqu'on laisse la fonction de coup converger sans utiliser le seuil de variation de l'erreur de 1 cm^2 (lorsque ce seuil est mis à 0 cm^2 , en pratique l'erreur doit augmenter ou sa variation doit être nulle à l'épsilon machine près pour que la convergence s'arrête), l'ICPP met alors beaucoup d'itérations pour converger. En effet, les correspondances point à point limitent le déplacement des points LiDAR pendant la phase de minimisation. L'ICPP recalcule les points correspondants contrairement à l'ICP qui utilise directement les points du modèle. Il recalcule alors à chaque itération des points de plus en plus proches de la solution finale, mais l'erreur va alors décroître de moins en moins vite. Lorsque des correspondances point à ligne sont utilisées, comme pour le PLICP et le mixICP, les points LiDAR ont plus de liberté de mouvement durant la minimisation et peuvent ainsi converger très rapidement.

Les résultats entre les deux critères d'arrêt varient très peu, un seuil de 1 cm^2 permet en effet une convergence suffisante pour les différentes correspondances. Le nombre moyen d'itérations nécessaires augmente, principalement pour l'ICPP qui passe de 4 à 243 en moyenne (Tab. 4.4). Malgré cette augmentation du nombre d'itérations, la moyenne de la norme de l'erreur de position (Tab. 4.5) et le taux de couverture (Tab. 4.6) ne varient quasiment pas, à l'exception de l'ICPP. En effet, ne convergeant pas parfaitement à chaque fois, on peut voir que lorsqu'il a la possibilité de le faire, il va alors avoir la possibilité de diverger en glissant le long du véhicule, comme le ferait le PLICP ou le mixICP, mais sans la capacité de correctement calculer sa matrice de covariance. Il va donc être beaucoup moins consistant.

La figure 4.11 montre des exemples de convergence pour les différentes correspondances. On voit que dans le cas du PLICP et du mixICP, lorsqu'une correspondance point à ligne est utilisée, les points LiDAR peuvent avoir tendance à « glisser » sur le modèle.

Le temps de calcul étant proportionnel au nombre d'itérations, lorsque ce nombre est faible, le temps de calcul est considérablement réduit. On peut voir dans la table 4.7 que l'ICP donne le temps de calcul le plus faible du fait qu'il s'agit de la correspondance la plus simple. ICPP met souvent plus d'itérations à converger, son temps de calcul moyen est donc plus important. Le mixICP est très proche du PLICP, qui donne un temps de calcul très acceptable, suffisamment faible pour que le calcul soit terminé avant le scan LiDAR suivant ($< 0,1\text{ s}$). Les deux minimisations sont

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.6. – Taux de couverture (%) pour un seuil nul et un seuil de 1 cm^2 .

seuil	ICP	ICPP	PLICP	mixICP
0,0	93,5	56,5	91,5	89,7
1,0	93,5	83,9	91,6	89,8

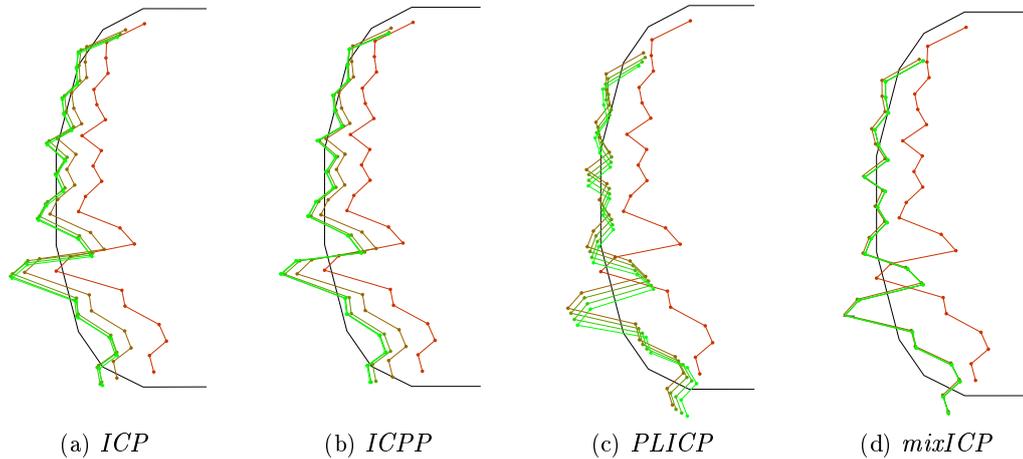


FIGURE 4.11. – Exemple de convergence des points LiDAR sur le modèle polygonal pour les différentes projections. Le modèle polygonal est représenté en noir et les points LiDAR varient du rouge, avant la première itération, au vert quand l'algorithme a fini de converger. Un seuil de $0,0001$ est utilisé.

difficilement comparables. En effet, la minimisation par pseudo-inverse semble plus rapide, mais la fonction de minimisation par pseudo-inverse est plus optimisée que la minimisation par calcul de racine de polynôme. Dans tous les cas, ces durées sont bien inférieures au temps de calcul de la méthode d'optimisation par échantillonnage aléatoire avec champ de vraisemblance ($0,176\text{ s}$).

4.7.3. PLICP avec élimination des points sans projection orthogonale

La correspondance point à ligne ne tombe pas toujours sur un segment. Dans le cas où la distance la plus courte est une distance euclidienne à un point du modèle, le segment mis en correspondance est choisi de manière arbitraire comme étant le segment suivant ce point. Il est aussi possible de supprimer les correspondances qui ne tombent pas sur un segment. L'algorithme a alors moins de points sur lesquels se baser, mais toutes les distances restantes sont justes. La table 4.8 montre que l'élimination de ces points améliore très peu la précision et le taux de couverture de l'algorithme. En effet très peu de points sont éliminés en pratique.

4. Pose inter-véhicules à partir de mesures LiDAR

TABLE 4.7. – Temps de calcul en secondes des différentes minimisations et correspondances pour une implémentation sous Matlab.

	ICP	ICPP	PLICP	mixICP
Minimisation polynomiale	0,0095	0,0258	0,0184	0,0181
Minimisation par pseudo-inverse	0,0059	0,0230	0,0153	0,0158

TABLE 4.8. – Moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ et taux de couverture du PLICP utilisant une minimisation basée sur des matrices pseudo-inverses avec et sans élimination des points LiDAR ne pouvant pas être projetés orthogonalement sur le modèle.

	$\ \bar{\epsilon}_p\ $ (cm)	C (%)
Sans élimination des points	11,5	91,6
Avec élimination des points	11,3	91,7

4.7.4. Influence des paramètres et étude de sensibilité

4.7.4.1. Influence du modèle

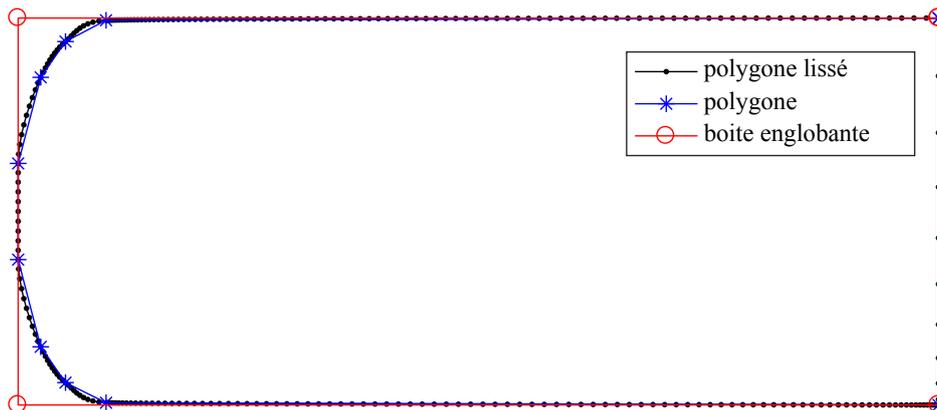


FIGURE 4.12. – Modèles géométriques polygonaux : boite englobante constituée de 4 segments, polygone constitué de 10 segments et polygone lissé constitué de 280 segments. Le polygone est le modèle utilisé par défaut alors que le modèle polygonal lissé est celui utilisé pour simuler les points LiDAR.

La figure 4.13a montre la moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ pour les différentes correspondances en fonction du modèle utilisé. Ces modèles sont basés sur le modèle à 10 segments utilisé pour obtenir les résultats précédents (en bleu sur la figure 4.12). La boite englobante de ce modèle permet d'obtenir le modèle à 4 segments (en rouge sur la figure 4.12). En retirant des sommets, on obtient ensuite des modèles à 6 et 8 segments. Pour obtenir des modèles avec plus de sommets, des interpolations cubiques sont appliquées au modèle à 10 segments (en noir sur la figure 4.12). L'erreur de l'ICP diminue de manière significative. Il utilise les sommets du polygone et a donc besoin d'un modèle suffisamment discrétisé contrairement aux

4. Pose inter-véhicules à partir de mesures LiDAR

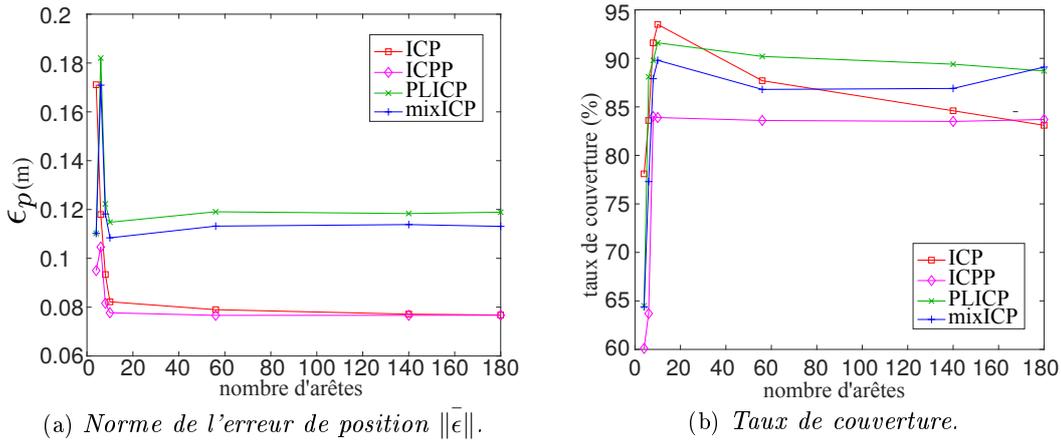


FIGURE 4.13. – Influence du nombre de segments du modèle sur la moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ et sur le taux de couverture pour les quatre correspondances.

autres correspondances.

La figure 4.13b donne le taux de couverture des quatre correspondances pour les différents modèles testés. Ce taux de couverture augmente lorsque la discrétisation augmente. Néanmoins, elle diminue à nouveau dans le cas de l'ICP pour les modèles lissés. L'erreur de la minimisation, utilisée dans la matrice de covariance, augmente en effet lorsque le modèle est peu discrétisé puisque les sommets sont plus éloignés des points LiDAR. Cela a pour effet d'augmenter l'ellipse d'incertitude donnant alors un taux de couverture plus important. On observe alors que dans le cas de l'ICP, ce taux de couverture comme l'erreur de position converge vers ceux de l'ICPP. Ces deux algorithmes agissent de manière identique et donnent donc des résultats similaires lorsque la discrétisation du modèle est importante.

Ces figures montrent qu'un modèle polygonal donne de meilleurs résultats qu'une boîte englobante. Toutefois, une discrétisation importante ralentit l'algorithme pendant la phase de correspondance sans en augmenter les performances.

4.7.4.2. Influence du scénario de conduite

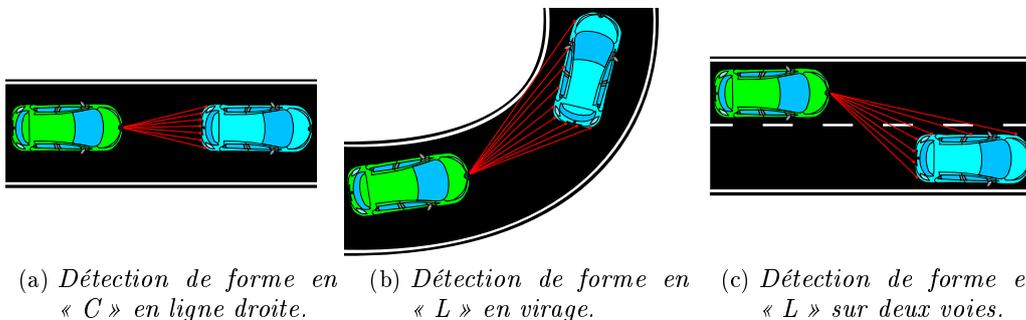


FIGURE 4.14. – Détection LiDAR en ligne droite et en virage.

4. Pose inter-véhicules à partir de mesures LiDAR

Trois scénarios ont été testés, des conduites sur ligne droite, en virage et sur deux voies.

Dans la ligne droite (Fig. 4.14a), seul l'arrière du véhicule est détecté (détection en « C »). L'arrière étant légèrement arrondi, une invariance par rotation et une invariance par translation apparaissent. Le problème est alors mal conditionné, la pose obtenue est peu précise et le taux de couverture est plus difficile à obtenir.

Dans la conduite en virage (Fig. 4.14b) ou sur deux voies (Fig. 4.14c), l'arrière et un des côtés du véhicule sont détectés par le LiDAR (détection en « L »). Le problème est bien conditionné et une pose précise avec un bon taux de couverture est souvent trouvée.

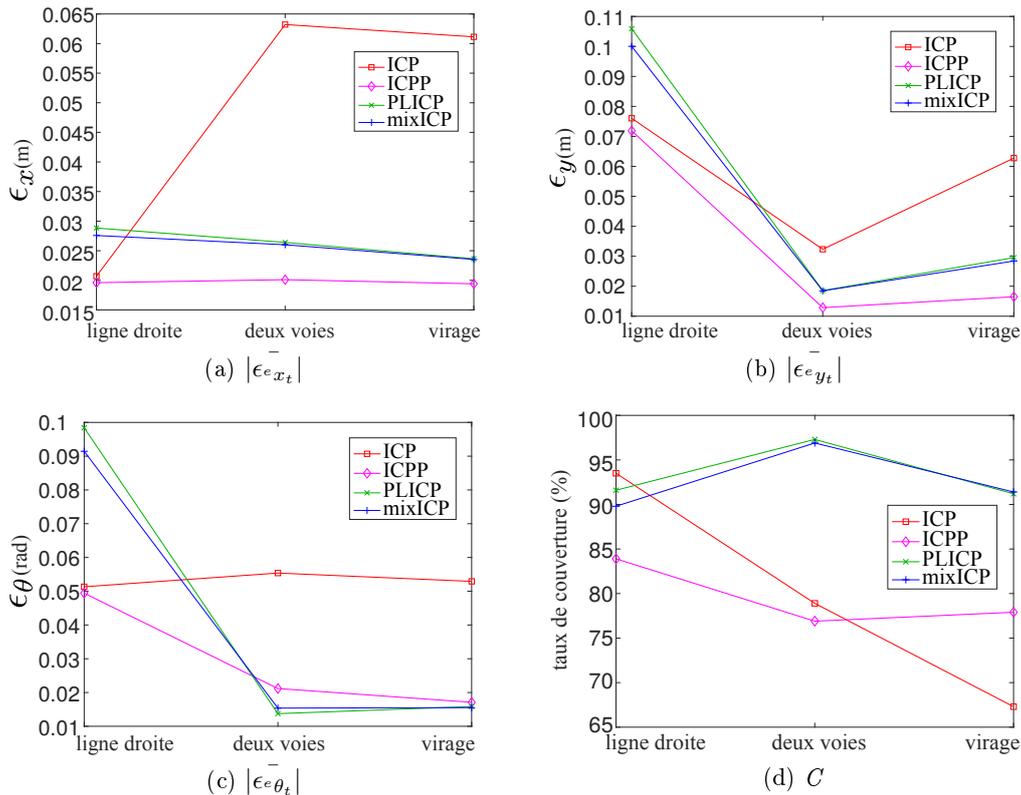


FIGURE 4.15. – Erreurs moyennes selon l'axe longitudinal e_{x_t} , et l'axe latéral e_{y_t} du égo-véhicule et selon l'orientation relative e_{θ_t} et taux de couverture C pour les quatre correspondances dans le cas d'une configuration de conduite sur route droite, sur deux voies et en virage.

La figure 4.15 confirme les hypothèses précédentes, la précision selon l'axe latéral e_{y_t} et selon l'orientation relative e_{θ_t} augmente lorsque deux faces du véhicule sont visibles, dans les scénarios de conduite en virage et sur deux voies.

Le taux de couverture est meilleur pour le scénario de conduite sur deux voies (Fig. 4.15).

4.7.4.3. Influence de l'interdistance

Lorsque l'interdistance entre le véhicule cible et l'égo-véhicule augmente, le nombre de points LiDAR diminue. Le problème est alors de moins en moins bien conditionné. L'erreur de position relative devient alors plus importante (Fig. 4.16a) et le taux de couverture augmente. L'utilisation de la matrice hessienne permet en effet de prendre en compte le conditionnement du problème.

4.7.4.4. Influence du bruit LiDAR

L'erreur de position et le taux de couverture augmentent lorsque le bruit LiDAR augmente (Fig. 4.17). La pose relative devient bien plus précise lorsque le LiDAR est plus précis.

4.7.4.5. Influence de l'erreur sur la pose initiale

Bien que l'algorithme de localisation initiale permette de corriger grandement l'erreur de la position initiale, la minimisation itérative est très sensible au bruit d'orientation. Ainsi même si les figures 4.18a et 4.18b décrivent l'évolution de l'erreur et de la consistance lorsque la variance de la position et de l'orientation augmentent, la variance de l'orientation a le plus gros impact. Lorsque celui-ci augmente, l'erreur devient importante (Fig. 4.18a) et le taux de couverture diminue. Des ambiguïtés peuvent apparaître s'il est trop important et l'algorithme peut se tromper de face à mettre en correspondance avec les points LiDAR.

Dans ces simulations, un seul véhicule était détecté par le LiDAR du égo-véhicule. Dans un cas à plus de deux véhicules, l'erreur de position peut engendrer des ambiguïtés sur la correspondance entre les groupes de points et les véhicules.

4.8. Robustification de la localisation relative

Dans le cas où le problème est mal conditionné ou mal initialisé, de mauvaises convergences peuvent être observées. Pour éviter ce problème, des tests peuvent détecter ces défauts et permettre de robustifier la méthode.

4.8.1. Élimination des mauvaises correspondances

Le premier problème est la mauvaise mise en correspondance de la pose communiquée avec le groupe de points LiDAR détectés. En effet, en reprenant la méthode proposée de localisation par boîte englobante, il n'est pas garanti que le groupe de points LiDAR le plus proche du modèle corresponde à celui du véhicule. Il est alors plus sûr de conserver tous les groupes de points LiDAR à proximité du modèle. On modifie donc la méthode de sorte à conserver tous les groupes (en rouge sur la figure 4.19) dont le point de référence de la boîte englobante (en orange sur la figure 4.19) se trouve dans une ellipse d'incertitude calculée à partir de la matrice de covariance de la pose relative. Cette ellipse (à 95% ou 99% par exemple, selon la fiabilité recherchée) est centrée sur le point de référence de la boîte englobante du modèle communiqué (en bleu sur la figure 4.19).

Il existe des cas où, par exemple lorsque le modèle communiqué est associé à un groupe de points LiDAR correspondant à un buisson dépassant sur la route

4. Pose inter-véhicules à partir de mesures LiDAR

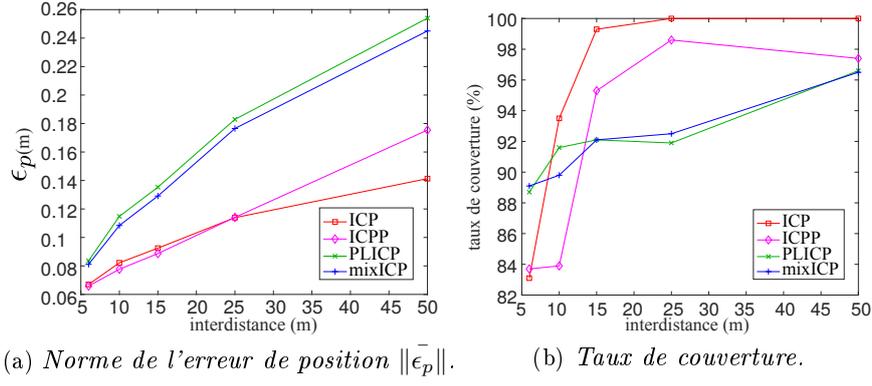


FIGURE 4.16. – Moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ et taux de couverture pour les quatre correspondances en fonction de l'interdistance.

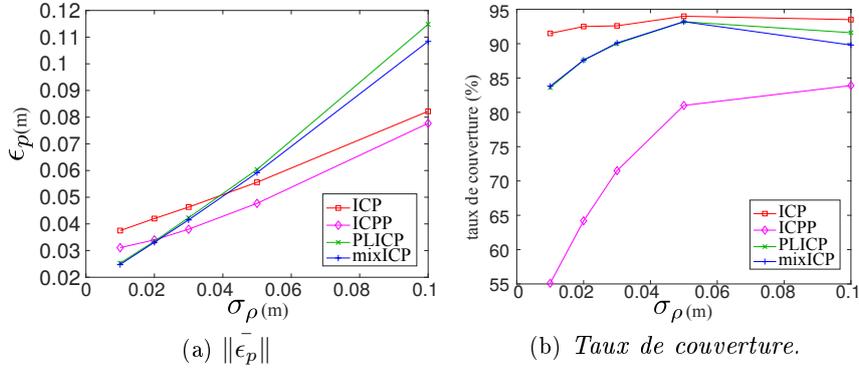


FIGURE 4.17. – Moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ et taux de couverture pour les quatre correspondances en fonction du bruit de LiDAR.

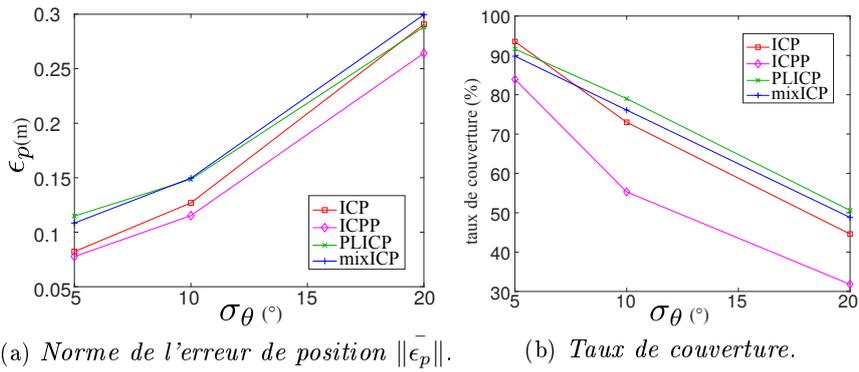


FIGURE 4.18. – Moyenne de la norme de l'erreur de position relative $\|\bar{\epsilon}_p\|$ et taux de couverture pour les quatre correspondances en fonction du bruit de pose pour $[\sigma_x \sigma_y \sigma_\theta] = [0.5 0.5 5]$, pour $[\sigma_x \sigma_y \sigma_\theta] = [2.5 2.5 10]$ et pour $[\sigma_x \sigma_y \sigma_\theta] = [5 5 20]$ ($[\sigma_x] = [\sigma_y] = m$ et $[\sigma_\theta] = ^\circ$).

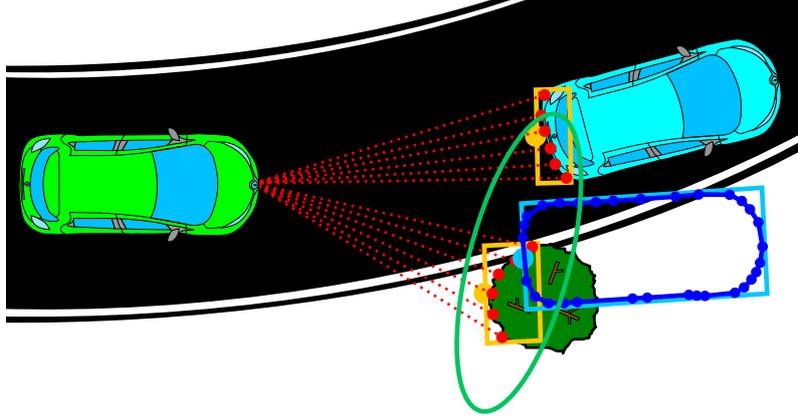


FIGURE 4.19. – *Problème de correspondance entre le groupe de points LiDAR et le modèle. Le groupe de points LiDAR (en rouge) du buisson étant plus proche du modèle (en bleu foncé) que celui du véhicule, il risque d’être mis en correspondance avec celui-ci. Les boîtes englobantes des groupes de points sont illustrées en orange avec leurs points de références et celle du modèle et son point de référence est en bleu clair. L’ellipse d’incertitude de la position est en vert.*

(Fig. 4.19), l’ICP fournit une erreur qui permet de détecter de mauvaises correspondances. On calcule ainsi la distance moyenne des points LiDAR au modèle après convergence :

$$d_{mean} = \sqrt{\frac{1}{N}E(\Delta q)}. \quad (4.75)$$

En comparant cette distance moyenne à un seuil, on peut éliminer les mauvaises correspondances. En pratique, on peut prendre un seuil de l’ordre de l’écart-type des mesures LiDAR, dans notre cas 10 cm. En conservant tous les groupes de points LiDAR cohérents avec la pose initiale, on applique l’ICP sur chaque groupe de points en partant du plus proche jusqu’au plus éloigné. Grâce au test de la distance moyenne, on peut trouver la bonne correspondance même si un autre groupe de points (dû à l’environnement) est plus proche que la bonne solution.

Dans le cas où plusieurs véhicules ont des géométries similaires, ce test peut toutefois ne pas être suffisant. Le cas présentant le plus de risque d’une mauvaise correspondance étant celui où un véhicule communicant est caché par un véhicule non communicant et que la pose communiquée soit très inexacte (Fig. 4.20). Il est donc important d’avoir une pose la plus exacte possible et d’éviter les cas où une zone importante de l’ellipse d’incertitude est hors de la zone auscultée par le LiDAR.

4.8.2. Élimination des mauvaises convergences

L’ICP fournissant avec la pose relative la matrice de covariance, un test de taux de couverture est réalisé en calculant une distance de Mahalanobis entre la pose relative initiale ${}^e q_{t,init}$ et la pose relative estimée par l’ICP ${}^e \hat{q}_t$:

$$D = \sqrt{({}^e q_{t,init} - {}^e \hat{q}_t)^T (\Sigma_{{}^e q_{t,init}} + \Sigma_{{}^e \hat{q}_t})^{-1} ({}^e q_{t,init} - {}^e \hat{q}_t)}. \quad (4.76)$$

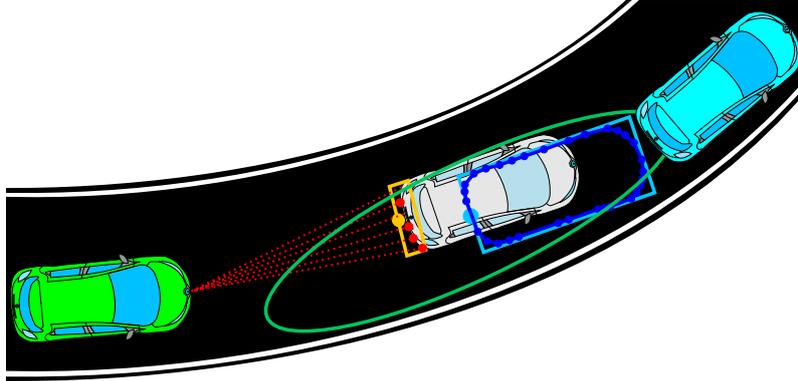


FIGURE 4.20. – *Problème de correspondance, lorsque le véhicule communicant (en bleu) est caché par un véhicule non communicant (en gris) ayant une géométrie identique. L'association se fait alors entre le modèle communiqué et les points LiDAR du mauvais véhicule.*

Cette distance est alors comparée à un seuil. En pratique lorsque les points LiDAR ne convergent pas correctement, par exemple, lorsque les points LiDAR de l'arrière du véhicule convergent sur un des côtés du véhicule, un simple seuil sur l'erreur angulaire suffit à supprimer ces mauvaises convergences. L'utilisation en continu de la localisation initiale par boîte englobante évite l'apparition de ces mauvaises convergences.

4.8.3. Détection d'un mauvais conditionnement du problème

Dans le cas où trop peu de points sont détectés ou lorsque seulement une face du véhicule est détectée, le problème est mal conditionné. Des invariances par translation ou rotation peuvent alors entraîner une mauvaise convergence de l'ICP. Pour éviter ce problème, des fonctions d'analyses numériques permettent de calculer le conditionnement de la matrice A ($\text{cond}(A)$). En pratique, le calcul du déterminant $A^T A$ permet de trouver ces mauvais conditionnements, lorsque cette matrice est proche de ne pas être inversible. Deux tests sont donc possibles (présentés ici avec des seuils arbitraires) pour déterminer un bon conditionnement du problème lorsque ceux-ci sont valides :

$$\text{cond}(A) < 10^5, \quad (4.77)$$

$$\det(A^T A) > 10^{-5}. \quad (4.78)$$

Ces tests sont très peu sensibles et permettent de détecter facilement les mauvais conditionnements même avec ces seuils grossiers trouvés expérimentalement. Ils sont suffisamment sensibles pour détecter la différence entre une forme du groupe de points en « C » et en « L ».

4.9. Conclusion

Ce chapitre a présenté différentes méthodes de localisation relative utilisant des points LiDAR initialisée avec et assistée d'une pose communiquée par le véhicule

4. Pose inter-véhicules à partir de mesures LiDAR

perçu. Un premier algorithme utilisant des boîtes englobantes permet de réduire considérablement l’erreur de position, l’erreur d’orientation reste toutefois inchangée. Afin d’améliorer la précision, des algorithmes de minimisation itérative ont été testés avec deux algorithmes de minimisation et quatre techniques de mise en correspondance différentes entre les points LiDAR et le modèle polygonal. L’algorithme de minimisation utilisant des matrices pseudo-inverses a donné de meilleurs résultats en termes de consistance, la précision des deux algorithmes de minimisation étant sensiblement identique. La correspondance point à ligne permet une meilleure estimation de la matrice de covariance, en modélisant les bords du véhicule par des droites et non par des points. Cette correspondance donne la possibilité aux points de glisser sur la surface du véhicule, ce qui permet à l’algorithme de converger rapidement. Différents modèles ont été comparés et le modèle polygonal donne un meilleur taux de couverture que la boîte englobante utilisée lorsque seules la longueur et la largeur du véhicule perçu sont connues. L’utilisation d’un modèle très discrétisé n’apporte aucune amélioration et ralentit la phase de correspondance. En faisant varier le bruit LiDAR ou le bruit de pose des véhicules, nous avons observé que plus l’incertitude est grande plus notre modèle pour estimer cette incertitude est mauvais. Finalement, des scénarios de conduite en ligne droite, sur deux voies et en virage ont aussi été comparés. Lorsque deux faces du véhicule sont visibles, comme dans le scénario de conduite sur deux voies et de conduite en virage, le problème est alors mieux conditionné et la précision augmente considérablement.

Dans cette étude, nous avons implémenté des algorithmes d’optimisation itérative basés sur des minimisations déjà existantes, mais adaptées pour de nouvelles correspondances. La contribution principale de ce travail est l’utilisation de ces algorithmes pour déterminer la pose relative entre deux véhicules s’échangeant des modèles polygonaux de leurs géométries en deux dimensions. Une autre contribution est l’ajout de critères permettant de robustifier la méthode.

Cette méthode de perception avec un capteur 2D à une nappe présente quelques limitations. Les poids lourds et véhicules tout-terrain peuvent avoir leur pare-chocs trop haut pour être détectés par la nappe horizontale. Lorsque le véhicule équipé du LiDAR passe sur un dos-d’âne ou prend un virage trop rapidement, les angles de tangage et de roulis ne permettent plus au véhicule de conserver la nappe initialement horizontale parallèle au plan de la route.

Une première approche serait d’utiliser des algorithmes en trois dimensions avec un LiDAR multinappes et un modèle 3D polyédrique, un modèle représentant l’enveloppe extérieure du véhicule cible par une discrétisation en triangles semblable à un format de modélisation 3D représentant un objet grâce à des facettes triangulaires. Cette représentation en trois dimensions permet de s’affranchir des limitations précédentes et il existe des modèles 3D simplifiés comme celui présenté par [Coenen *et al.*, 2017], où la géométrie n’est définie que par deux paramètres. Néanmoins, l’utilisation d’un LiDAR mononappe moins coûteux donne déjà de bons résultats et est plus rapide en temps de calcul et en temps de communication des modèles. Dans le cas où les hypothèses nécessaires pour utiliser un modèle 2D ne seraient pas valides, il est possible d’extraire un modèle 2D de celui en 3D en prenant la section de celui-ci selon le plan de la nappe LiDAR utilisée. Une autre méthode serait de projeter tous les points d’un LiDAR 3D sur le plan horizontal (en conservant seulement les coordonnées x et y) afin de récupérer l’enveloppe extérieure qui pourrait

4. Pose inter-véhicules à partir de mesures LiDAR

être utilisée avec un modèle 2D. L'enveloppe du véhicule est en effet plus grande à la hauteur du pare-chocs afin que cette zone du véhicule soit la première à rentrer dans un obstacle en cas d'accident.

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

5.1. Introduction

La pose relative obtenue grâce à la perception LiDAR fournit une contrainte entre les poses absolues du égo-véhicule (le véhicule équipé du LiDAR) et du véhicule cible perçu. L'une de ces deux poses peut alors être propagée pour estimer la seconde. Cette première approche de localisation coopérative permet d'observer les problèmes liés aux propagations d'incertitude représentée par les matrices de covariance des poses absolues et relatives. Nous nous intéressons ici au cas où deux véhicules communiquent leur pose respective et où l'un des deux véhicules est capable de percevoir le second. Une autre solution serait de communiquer en plus des poses absolues, les poses relatives avec des messages CPM (Cooperative Perception Message) [ETSI, 2018] par exemple. Dans cette seconde approche, le véhicule perçu peut donc estimer sa pose absolue à partir de la pose relative et de la pose absolue communiquées. Les calculs à réaliser restent cependant identiques, seul le véhicule réalisant la propagation change. Les résultats sont donc identiques, cette approche ne sera pas étudiée dans ce chapitre.

Nous commencerons ce chapitre en posant la problématique dans la section 5.2 et en décrivant deux approches que nous avons étudiées : l'« approche communicante » lorsqu'on cherche la pose du véhicule qui perçoit et l'« approche coopérative » lorsqu'on cherche la pose du véhicule perçu. Nous étudierons ensuite dans la section 5.3 des résultats expérimentaux obtenus à partir d'un scénario de conduite en convoi avec deux véhicules.

5.2. Problématique

Une fois que la pose relative ${}^e q_t$ (Fig. 5.1) du véhicule cible dans le repère du égo-véhicule et la matrice de covariance associée ${}^e \Sigma_t$ ont été estimées, la pose absolue de chacun des véhicules peut être déterminée à l'aide de la pose de l'autre.

Deux formulations peuvent être faites. L'« approche communicante » consiste à estimer la pose absolue du égo-véhicule en utilisant sa perception pour estimer la pose relative ${}^e q_t$ ainsi que la pose communiquée par le véhicule perçu q_t . La seconde formulation dite « approche coopérative » consiste à estimer la pose du véhicule perçu q_t à partir de la pose relative ${}^e q_t$ estimée par le véhicule percevant et à partir de sa pose absolue q_e (c'est le véhicule percevant qui localise).

Dans l'approche communicante, l'estimation de la nouvelle pose¹ et de sa matrice

1. dans le repère de travail ENU

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

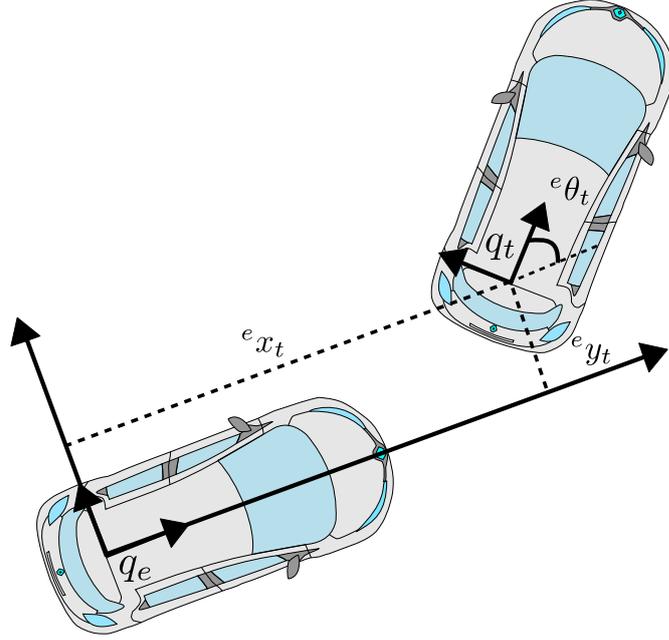


FIGURE 5.1. – La pose relative du véhicule cible mesurée dans le repère du égo-véhicule ${}^e q_t = [{}^e x_t \ {}^e y_t \ {}^e \theta_t]^T$ permet de déterminer la pose absolue de l'hôte q_e en utilisant la pose communiquée du véhicule cible q_t (qui est utilisée à la manière d'un « récepteur GNSS déporté ») ou inversement.

de covariance se fait grâce à la transformation suivante :

$$\begin{aligned} q_e &= q_t \ominus {}^e q_t & (5.1) \\ &= \begin{pmatrix} x_t + -{}^e x_t \cos(\theta_t - {}^e \theta_t) + {}^e y_t \sin(\theta_t - {}^e \theta_t) \\ y_t + -{}^e x_t \sin(\theta_t - {}^e \theta_t) - {}^e y_t \cos(\theta_t - {}^e \theta_t) \\ \theta_t - {}^e \theta_t \end{pmatrix}, \end{aligned}$$

$$\Sigma_{q_e} \approx J_{\oplus} \begin{pmatrix} \Sigma_{q_t} & \Sigma_{q_t, {}^e q_t} J_{\ominus}^T \\ J_{\ominus} \Sigma_{q_t, q_t} & J_{\ominus} \Sigma_{q_t} J_{\ominus}^T \end{pmatrix} J_{\oplus}^T. \quad (5.2)$$

L'approche coopérative utilise la transformation

$$\begin{aligned} q_t &= q_e \oplus {}^e q_t & (5.3) \\ &= \begin{pmatrix} x_e + {}^e x_t \cos(\theta_e) - {}^e y_t \sin(\theta_e) \\ y_e + {}^e x_t \sin(\theta_e) + {}^e y_t \cos(\theta_e) \\ \theta_e + {}^e \theta_t \end{pmatrix}, \end{aligned}$$

$$\Sigma_{q_t} \approx J_{\oplus} \begin{pmatrix} \Sigma_{q_e} & \Sigma_{q_e, {}^e q_t} \\ \Sigma_{q_t, q_e} & \Sigma_{q_t} \end{pmatrix} J_{\oplus}^T. \quad (5.4)$$

On estime ainsi les matrices de covariance grâce à des approximations linéaires au premier ordre. On observe cependant que les équations (5.1) et (5.3) dépendent de

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

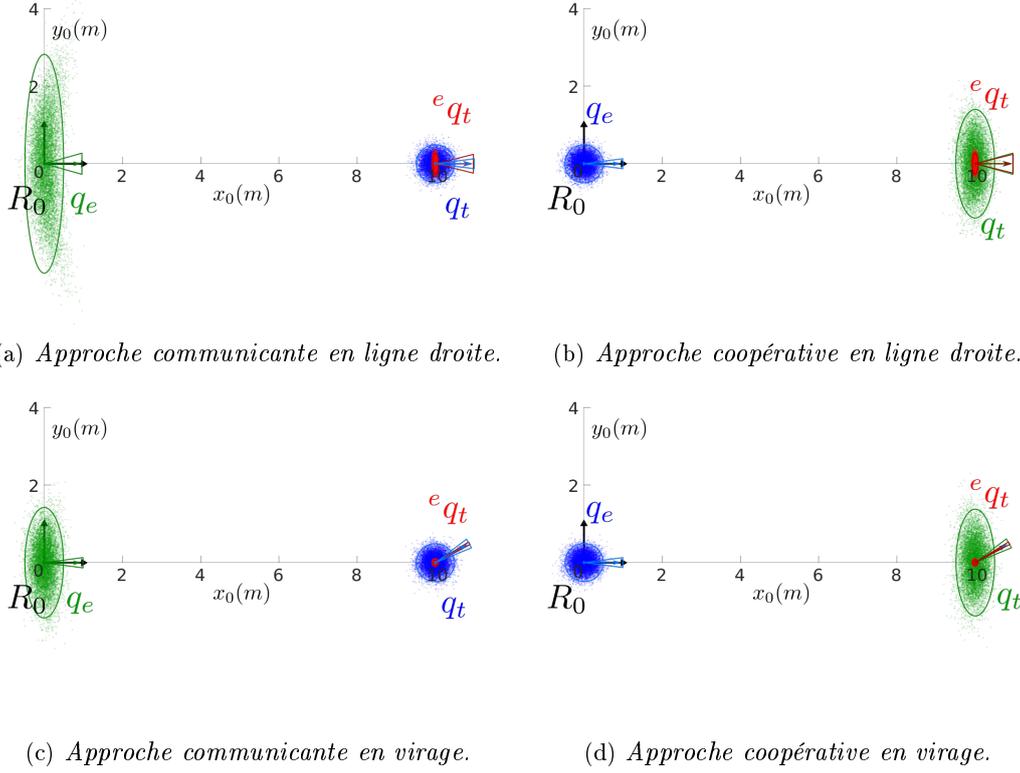


FIGURE 5.2. – *Pose estimée (en vert) à partir de la pose absolue (en bleu) et de la pose relative (en rouge). Les nuages de points représentent les distributions des poses approximées par des distributions normales illustrées par les ellipses et les cônes d'incertitude.*

fonctions trigonométriques pouvant être fortement non linéaires. Ces non-linéarités dépendent alors des angles sur lesquels ces fonctions s'appliquent. L'approche communicante (Eq. (5.1)) est à la fois sensible à l'erreur d'orientation du véhicule perçu ϵ_{θ_t} et à celle de l'orientation relative $\epsilon_{e\theta_t}$, au contraire de l'approche coopérative (Eq. (5.3)) qui n'est sensible qu'à l'erreur d'orientation du véhicule percevant ϵ_{θ_e} . Dans l'approche communicante, l'égo-véhicule calcule lui-même la pose relative à partir de la pose initiale du véhicule communicant. Cette approche est plus sûre, puisqu'elle permet à l'égo-véhicule de vérifier si sa perception et ce qui lui est communiqué sont cohérents. Il est donc capable de rejeter les poses communiquées aberrantes. La figure (5.2) illustre la propagation de pose et de son incertitude avec les approches communicante et coopérative en reprenant les erreurs de la pose relative obtenues dans le chapitre (4) lors d'une détection en « C » en ligne droite et d'une détection en « L » en virage. Afin d'étudier la propagation d'une pose absolue grâce à une pose relative, 10 000 poses absolues et 10 000 poses relatives sont générées aléatoirement selon des lois normales. Les écarts types $\sigma_x = 0,2$ m, $\sigma_y = 0,2$ m et $\sigma_\theta = 3^\circ$ sont utilisés pour la pose absolue à propager. Pour la pose relative, nous nous sommes basé sur les erreurs obtenues dans le chapitre (4) lors d'une détection en « C » et d'une détection en « L », ainsi en ligne droite $\sigma_{e_{x_t}} = 0,03$ m, $\sigma_{e_{y_t}} = 0,11$ m et $\sigma_{e_{\theta_t}} = 0,1$ rad ($\approx 5,73^\circ$) et en virage $\sigma_{e_{x_t}} = 0,024$ m, $\sigma_{e_{y_t}} = 0,03$ m et

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

$\sigma_{e\theta_t} = 0,015 \text{ rad} (\approx 0.86^\circ)$. La figure (5.2) illustre ainsi la pose absolue à propager en bleu, la pose relative en rouge et la pose estimée après propagation en vert. Les flèches sont les poses réelles utilisées comme moyennes des distributions. Ces distributions des positions sont représentées par les nuages de points. Des ellipses et des cônes d'incertitude à 95% représentent les incertitudes. On peut remarquer que les ellipses d'incertitude estimées à partir de l'approximation au premier ordre (en vert) sont plutôt bonnes à l'exception de celle obtenue avec l'approche communicante en ligne droite. La distribution obtenue supposée être une gaussienne est en réalité une distribution en banane. La matrice de covariance n'est donc pas consistante dans ce cas. Cette forte non-linéarité provient de l'erreur d'orientation relative devenue très importante en ligne droite. Dans le cas de l'approche coopérative, on peut remarquer que l'erreur de position estimée est sensiblement similaire entre le scénario du virage et celui de la ligne droite. En effet, cette position ne dépend pas de l'orientation relative. On remarque aussi que les cônes d'incertitudes des orientations estimées ne dépendent pas de l'approche utilisée, les orientations estimées dépendent en effet des orientations des poses absolues à propager et des poses relatives comme l'indiquent les équations (5.1) et (5.3).

On pourrait penser qu'il est plus intéressant de mettre le véhicule avec l'estimation de pose la plus exacte à l'arrière d'un convoi pour que cette exactitude se propage aux suivants en utilisant le plus possible l'approche coopérative. Néanmoins, dans le cas où il y a plus de deux véhicules, comme l'orientation est estimée à partir de l'orientation relative et d'une orientation absolue, cette nouvelle incertitude angulaire détériorera la position suivante, quelque soit l'approche choisie. On peut voir sur la figure 5.3 que l'incertitude d'orientation se propage très vite aux autres véhicules du convoi. Dans cette simulation, la pose du véhicule connue précisément a des écarts types d'1 cm sur x et y et de $0,1^\circ$ sur son orientation. La pose relative a des écarts types de 10 cm sur x et y et 5° sur l'orientation. On voit que comme l'incertitude d'orientation intervient directement sur l'incertitude de position dans le cas de l'approche communicante, les incertitudes des véhicules à l'avant ressemblent à celles de l'arrière à partir du second véhicule se situant devant.

Des scénarios plus complexes peuvent être imaginés dans le cas où les véhicules peuvent être vus par de multiples véhicules. Plusieurs solutions peuvent alors être envisagées pour propager l'incertitude d'un véhicule. Il sera alors intéressant de fusionner l'ensemble de ces possibilités.

5.3. Résultats expérimentaux

Les données utilisées pour ces résultats ont été enregistrées au cours d'un scénario de conduite en convoi, où deux Renault Zoe roulaient sur la piste Séville composée de deux ronds-points et d'une ligne droite. Neuf tours ont été réalisés sur cette piste, pour un total de 2,82 km en 10 minutes et 7 489 scans LiDAR. L'interdistance entre les poses des deux véhicules était comprise entre 6,2 m et 18,8 m. Elle était d'environ 8 m dans les ronds-points et d'environ 13 m en ligne droite. Chaque véhicule était équipé d'un récepteur GNSS hybridée avec une centrale inertielle (SPAN-CPT de NovAtel) pour la localisation et la vérité terrain. La fréquence de ce capteur étant grande comparée à la vitesse, l'erreur temporelle est supposée négligeable devant l'erreur spatiale. Le véhicule suiveur était aussi équipé d'un LiDAR SICK LD-MRS

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

TABLE 5.1. – Erreur moyenne $[\bar{\epsilon}_x \bar{\epsilon}_y \bar{\epsilon}_\theta]^T$ dans le repère de la vérité terrain, erreur moyenne de position $\|\epsilon_p\|$ et taux de couverture C à 95% des deux approches pour le scénario complet, pour les perceptions en « L » seulement et les perceptions en « C » seulement.

formes	approche communicante			approche coopérative		
	toutes	C	L	toutes	C	L
$\bar{\epsilon}_x$ (m)	0,27	0,35	0,13	0,11	0,11	0,09
$\bar{\epsilon}_y$ (m)	1,16	1,72	0,22	0,36	0,41	0,27
$\ \epsilon_p\ $ (m)	1,24	1,82	0,27	0,39	0,44	0,29
$\bar{\epsilon}_\theta$ (°)	6,92	8,55	4,20	6,92	8,55	4,20
C (%)	91,5	86,9	99,2	98,0	97,2	99,3

à quatre nappes, seule la nappe horizontale était utilisée pour déterminer la pose relative. Le modèle géométrique 2D a été construit manuellement à partir d'un plan de la Renault Zoe. La vérité terrain ayant une incertitude de l'ordre du décimètre, sa matrice de covariance a été prise en compte pour calculer le taux de couverture. Comme cette erreur n'est pas négligeable en comparaison de celles des mesures LiDAR, les petites erreurs doivent être interprétées avec précaution.

Dans les scénarios réels de conduite, on obtient des résultats similaires à ceux obtenus en simulation dans la section précédente. On voit ainsi dans la figure 5.4a que les erreurs angulaire et latérale deviennent très importantes lorsqu'une forme en « C » est détectée avec l'approche communicante. On peut voir dans la figure 5.4b, illustrant l'approche coopérative avec une détection en ligne droite, que même si l'erreur d'orientation est impactée par l'erreur d'orientation relative, l'erreur de position ne l'est pas.

La table 5.1 montre les erreurs et les taux de couverture des approches communicante et coopérative pour les scénarios présentant des détections en « C », en « L » ou les deux. Les erreurs moyennes sont décomposées dans le repère de la vérité terrain afin d'observer l'erreur longitudinale $\bar{\epsilon}_x$, l'erreur latérale $\bar{\epsilon}_y$ et l'erreur d'orientation $\bar{\epsilon}_\theta$. L'erreur moyenne de position $\|\epsilon_p\|$ est définie en prenant la distance euclidienne moyenne entre la position estimée et la position de la vérité terrain. Le taux de couverture C est donnée en calculant le pourcentage d'échantillons vérifiant l'inégalité :

$$\epsilon_{q_e}^T (\Sigma_{q_e} + \Sigma_{q_e^*})^{-1} \epsilon_{q_e} < \chi_{3,1-\alpha}^2, \quad (5.5)$$

où $\chi_{3,1-\alpha}^2 = 7,81$ pour une confiance de $1 - \alpha = 95\%$. La précision de la vérité terrain n'étant pas beaucoup plus importante que celle des mesures LiDAR, la matrice de covariance de cette vérité terrain $\Sigma_{q_e^*}$ est prise en compte dans l'estimation du taux de couverture.

Lorsqu'une détection en « C » est observée, l'erreur d'orientation double en passant à $8,55^\circ$ contre $4,20^\circ$ lorsqu'une détection en « L » est présente. Cette erreur amplifie la non-linéarité de l'approche communicante. Le taux de couverture chute alors de $99,2\%$ à $86,9\%$ et l'erreur de position passe de $0,27$ m à $1,82$ m. On observe que l'inexactitude est majoritairement sur la composante latérale puisqu'elle passe de $0,22$ m à $1,72$ m conformément aux distributions en banane. Toutefois, même en détection en « C », la disposition des points LiDAR permet de contraindre la pose

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

relative sur la composante longitudinale. Ainsi, l'erreur longitudinale passe de 0,13 m à seulement 0,35 m. Cette composante reste donc utilisable pour un scénario de conduite en convoi, où la détection LiDAR permet principalement d'éviter le véhicule et où l'interdistance doit être correctement estimée. Même si cette erreur longitudinale augmente, une interdistance prenant en compte cette incertitude permettrait de pouvoir l'utiliser avec sécurité. On voit que dans un scénario complet présentant des virages et des lignes droites que l'exactitude et le taux de couverture sont altérés par les non-linéarités obtenues dans les lignes droites. Ainsi, l'erreur de position est de 1,24 m et la pose reste inconsistante avec un taux de 91,5%.

Dans le cas de l'approche coopérative, l'exactitude et le taux de couverture diminuent beaucoup moins entre une détection en « C » et en « L ». L'erreur de position passe de 0,29 m à 0,44 m du fait de l'erreur latérale passant de 0,27 m à 0,41 m. L'erreur longitudinale reste quasiment identique en passant de 0,09 m à 0,11 m. Cette différence étant proche de l'erreur de la vérité terrain, il est difficile de commenter cette variation. L'erreur d'orientation est identique à celle de l'approche communicante, les calculs sont en effet les mêmes. Lorsque la perception passe d'une détection en « C » à une détection en « L », le taux de couverture augmente en passant de 97,2% à 99,3% rendant la pose plus pessimiste. Cela ne signifie pas pour autant que l'estimation de l'incertitude augmente, les écarts types moyens estimés passent en effet de $\bar{\sigma}_x \approx 0,33\text{ m}$, $\bar{\sigma}_y \approx 0,35\text{ m}$ et $\bar{\sigma}_\theta \approx 0,15\text{ rad}$ à $\bar{\sigma}_x \approx 0,17\text{ m}$, $\bar{\sigma}_y \approx 0,18\text{ m}$ et $\bar{\sigma}_\theta \approx 0,026\text{ rad}$. La meilleure pose présentant la plus grande précision avec la plus petite incertitude est donc obtenue avec l'approche coopérative en virage malgré le fait qu'elle soit très pessimiste.

On peut voir au travers de ces résultats que l'approche coopérative donne de bien meilleurs résultats en termes de précision et de consistance que l'approche communicante, comme le démontre théoriquement la section 5.

5.4. Conclusion

La propagation d'une pose avec sa matrice de covariance a été testée expérimentalement avec deux approches différentes. Une approche communicante où l'égo-véhicule perçoit un autre véhicule et une approche coopérative où l'égo-véhicule est perçu par un autre véhicule. L'exactitude et la consistance ont été évaluées dans différentes situations. En virage où l'on perçoit des formes en « L », les deux approches ont donné de bons résultats avec une précision qui est de l'ordre du décimètre. Cependant, sur les lignes droites où la perception conduit à des formes en « C », présentant des invariances par rotation, l'approche coopérative est alors plus consistante et plus exacte. Ainsi il est préférable de se faire localiser en utilisant l'approche coopérative plutôt que se localiser avec l'approche communicante. Dans ce cas, il faut toutefois faire confiance au véhicule coopératif puisqu'il n'est pas possible sans autre source d'information de vérifier ce qui a été communiqué.

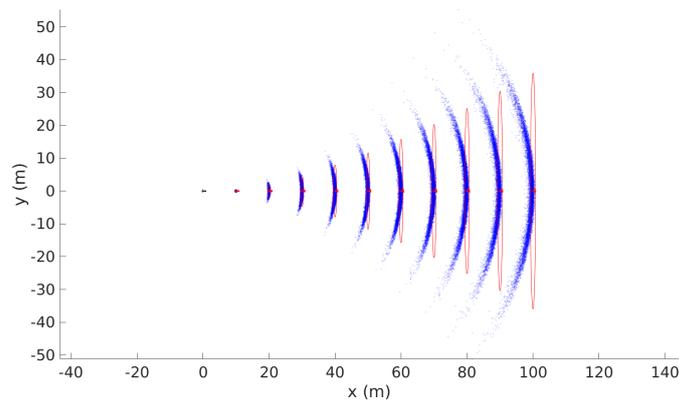
Dans ce chapitre nous avons utilisé des propagations d'incertitudes basées sur des approximations linéaires en utilisant des matrices Jacobiennes. Comme nous avons pu le voir, cette méthode donne de mauvais résultats lorsqu'elle est appliquée à des transformations fortement non linéaires. Pour remédier à ce problème la transformation non parfumée introduite par [Julier et Uhlmann, 1997a] permet une meilleure approximation, mais toujours avec des distributions gaussiennes. Le filtre particu-

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

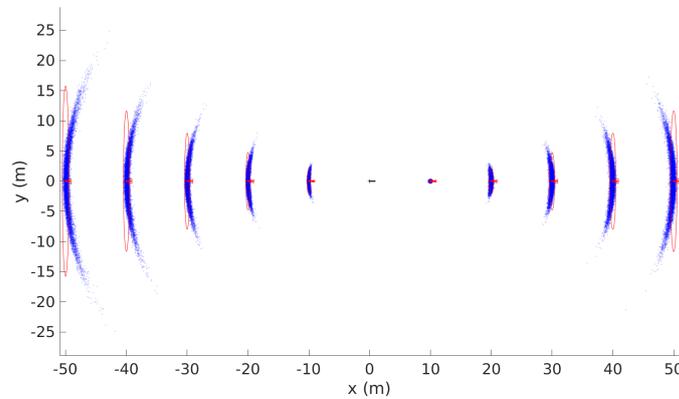
laire présenté par [Del Moral, 1997, Liu et Chen, 1998] et certaines méthodes ensemblistes comme SIVIA (Set Inversion Via Interval Analysis) ([Jaulin *et al.*, 2001]) ne font pas d'hypothèses sur la linéarité des fonctions utilisées et sur les distributions des erreurs. Ces méthodes peuvent toutefois être longues en temps de calcul. La mixture de gaussiennes introduite par [Sorenson et Alspach, 1970] permet d'obtenir une approximation d'une distribution quelconque à partir d'une somme de gaussiennes. [Wasik *et al.*, 2019] utilise cette approximation avec un filtre PHD (Probability Hypothesis Density Filter) ([Vo et Ma, 2006]) pour réaliser une localisation robuste d'une formation de plusieurs robots dans le cas où la communication est mauvaise.

Nous réalisons dans ce chapitre une propagation unidirectionnelle, nous estimons la pose d'un véhicule à partir de celle d'un second véhicule et de la pose relative. En pratique, nous souhaitons utiliser cette pose relative dans une fusion de données, elle permettra alors d'améliorer la pose du véhicule qui perçoit et celle du véhicule perçu simultanément en ajoutant une contrainte entre ces deux véhicules. Dans le cas où la transformation est fortement non linéaire, la pose obtenue n'est pas seulement inconsistante, elle n'est aussi pas suffisamment précise latéralement pour une navigation autonome. Pour remédier à ce problème, une contrainte latérale provenant d'un capteur de perception pourrait permettre d'améliorer cette précision latérale ainsi que conserver une estimation consistante.

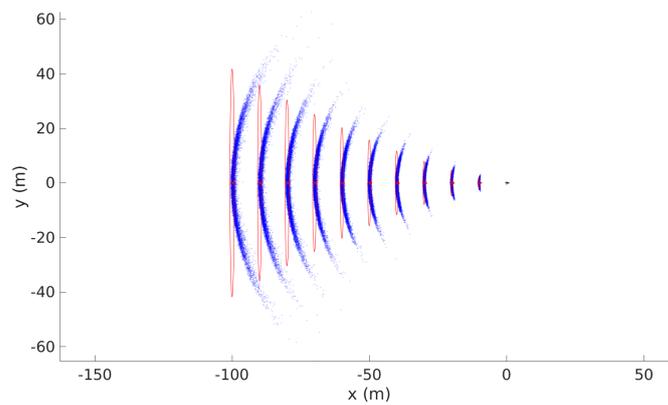
5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre



(a) Convoi avec 10 véhicules à l'arrière.



(b) Convoi avec 5 véhicules à l'arrière et 5 véhicules à l'avant.



(c) Convoi avec 10 véhicules à l'avant.

FIGURE 5.3. – Propagation de la pose et de la matrice de covariance d'un véhicule (en noir) avec une précision centimétrique ($\sigma_x = \sigma_y = 1 \text{ cm}$ et $\sigma_\theta = 0.1^\circ$) dans un convoi. Les nuages de points (en bleu) représentent les distributions des poses approximées par des distributions normales illustrées par les ellipses et les cônes d'incertitude (en rouge).

5. Propagation de la pose et de la matrice de covariance d'un véhicule à un autre

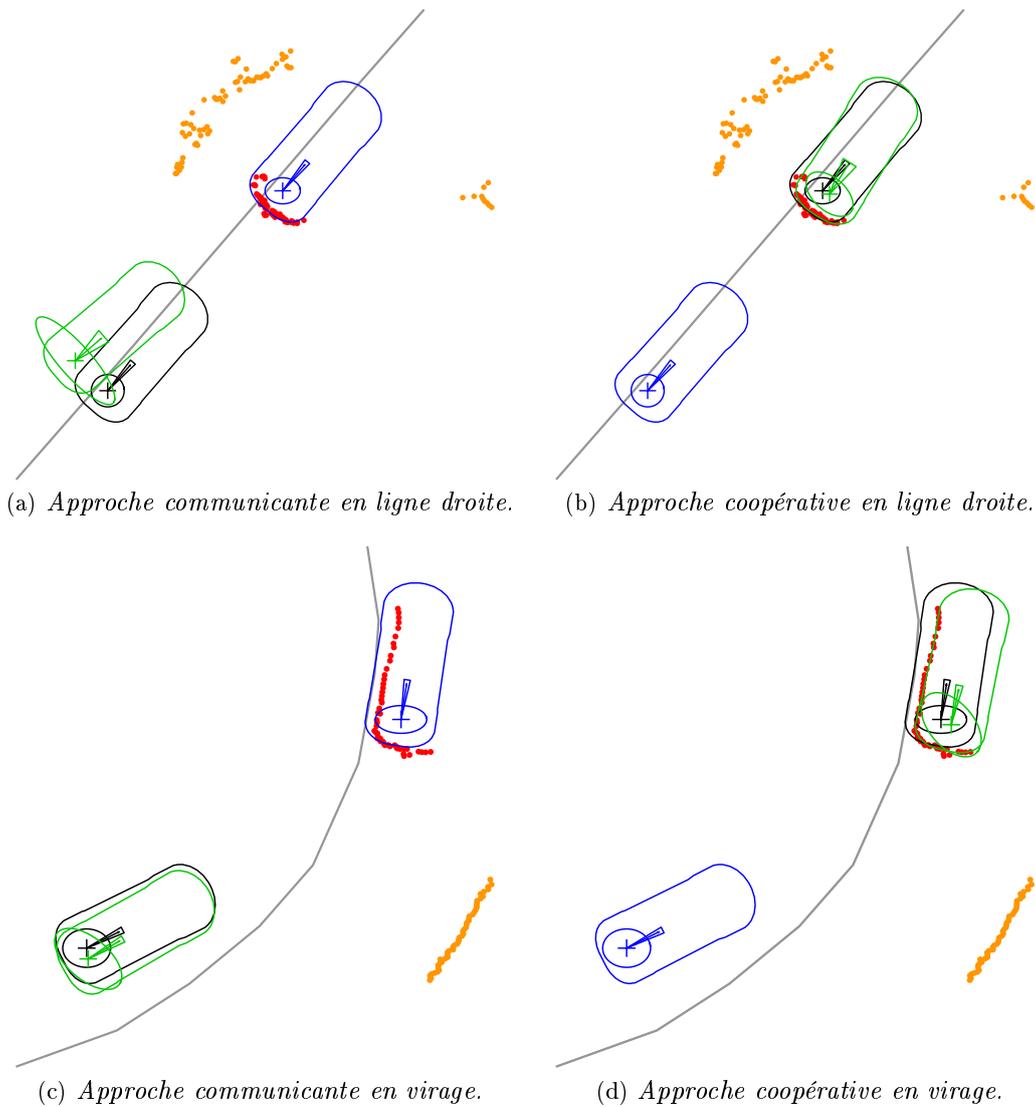


FIGURE 5.4. – Estimation de la pose du égo-véhicule (en vert) calculée à partir de la pose du véhicule communicant (en bleu) et des points LiDAR (en rouge). Cette pose est comparée à la vérité terrain (en noir). La poly-ligne grise représente le centre de la voie. Le modèle et l'ellipse et le cône d'incertitude sont dans la même couleur que leur pose.

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

6.1. Introduction

Afin de pouvoir naviguer de façon autonome, un véhicule doit être capable de se localiser et de localiser les véhicules à proximité avec lesquels il est en interaction. La localisation absolue est importante pour la conduite autonome, mais peut être parfois très imprécise. Elle est importante pour la planification du trajet et pour la navigation dans des environnements où les amers locaux sont peu nombreux ou difficiles à percevoir. D'un autre côté, les poses relatives ont besoin d'être précises et consistantes en particulier lorsque les véhicules sont proches les uns des autres. D'une façon simplifiée, on dit que le véhicule doit bien se localiser dans sa voie pour éviter d'en sortir par erreur et doit se localiser par rapport aux obstacles dynamiques pour contrôler ses interactions et éviter les collisions. Ainsi, l'utilisation d'une carte locale dynamique (CLD) en surcouche de la carte statique permet de représenter les différents agents en interaction.

Un algorithme centralisé permet de localiser l'ensemble des véhicules de façon optimale, il prend en effet en compte l'ensemble des observations de tous ces véhicules et permet de gérer les corrélations potentielles des erreurs de ces observations. [Biswas *et al.*, 2006] présente une méthode d'optimisation pour la localisation d'un réseau de capteurs. Cette solution centralisée requière néanmoins qu'un serveur ou qu'un véhicule leader fasse l'ensemble des calculs. L'ensemble des observations doivent toutefois être communiquées, ce qui est difficilement réalisable lorsque des LiDAR ou des caméras sont utilisés. Toutefois, certains calculs peuvent être distribués dans chaque véhicule afin de réduire la taille des informations communiquées, il est par exemple possible de communiquer des poses relatives estimées à partir d'un capteur de perception à la place des données brutes de celui-ci. Des problèmes peuvent, cependant, toujours apparaître dans le cas où la communication est rompue entre l'un des véhicules et le système en charge de l'estimation de la localisation globale. Afin d'éviter ces problèmes, des méthodes décentralisées permettent de prendre en compte les informations communiquées dans chaque véhicule tout en laissant à ces véhicules une autonomie par rapport au reste de la flotte [Mu *et al.*, 2011, Roumeliotis et Bekey, 2000]. Ainsi même dans des cas où la communication est limitée, la localisation de chaque agent peut être estimée [Zhu et Kia, 2019, Safaei et Mahyuddin, 2019]. Différentes méthodes de localisation coopérative décentralisées ont été ainsi présentées dans la littérature [Gao *et al.*, 2019].

Dans ce chapitre, nous commencerons par définir l'architecture de la fusion de données décentralisée et asynchrone. Nous décrirons ensuite les modèles et les processus d'évolution et de mise à jour de l'état de la CLD. Nous présenterons ensuite les résul-

tats expérimentaux obtenus grâce à cette fusion de données. Enfin, nous discuterons de la gestion d'une CLD et du choix des véhicules qui doivent y apparaître.

6.2. Architecture de fusion de données décentralisée et asynchrone

Dans notre cas, les agents présents dans une CLD sont uniquement des véhicules intelligents comme des voitures. La précision recherchée portant principalement sur les poses relatives, la CLD peut être exprimée dans le repère mobile de chaque véhicule. Pour ce faire, un changement de repère doit être appliqué à toute la CLD mais les modèles d'évolutions deviennent beaucoup plus complexes à établir et sensibles aux non-linéarités. Nous faisons donc le choix de définir la CLD dans le repère absolu pour éviter ces difficultés.

L'état estimé $\hat{X} = (\hat{X}_1, \dots, \hat{X}_N)^T$ de la CLD contient les états des N agents en coopération.

Chaque état

$$\hat{X}_i = (\hat{q}_i, \hat{b}_i, \hat{u}_i)^T,$$

contient la pose de l'agent $\hat{q}_i = (\hat{x}_i, \hat{y}_i, \hat{\theta}_i)^T$, le biais $\hat{b}_i = (\hat{b}_{xi}, \hat{b}_{yi})^T$ de son récepteur GNSS (car nous considérons des récepteurs GNSS bas-coût non différentiels) et le vecteur cinématique $\hat{u}_i = (\hat{v}_i, \hat{\omega}_i)^T$. La vitesse longitudinale et la vitesse de rotation sont utilisées pour extrapoler la pose à un instant donné. Ainsi chaque sous-état \hat{X}_i est un vecteur de dimension 7.

La figure 6.1 montre l'algorithme de fusion de données asynchrone que nous proposons d'étudier. Au centre, l'état de la CLD est représenté en vert. Lorsqu'une information (en jaune) est transmise par une source d'information (en orange), l'état est mis à jour par un module (en bleu). Cette mise à jour consiste tout d'abord à extrapoler l'état au temps de l'observation considéré en utilisant le modèle d'évolution avant d'appliquer le modèle d'observation. On suppose que les horloges des différents agents sont synchronisées avec des erreurs négligeables, ce qui est assez facile à réaliser avec des récepteurs GNSS. Les données entre les différents agents sont donc supposées toutes datées dans un système de temps commun.

Cet algorithme est complètement décentralisé : chaque véhicule exécute le même algorithme et utilise la communication pour envoyer et recevoir des CLDs. Les localisations absolues et relatives de tous les agents peuvent être globalement améliorées grâce à cet échange d'informations et grâce à la coopération et c'est l'objectif que nous cherchons à atteindre.

Puisque chaque observation est prise en charge par un processus informatique (*thread*) dédié, des mises à jour en parallèle peuvent être effectuées¹. La figure 6.2 illustre ce cas. On peut voir que lors de la troisième mise à jour (MaJ 3), l'état en sortie dépend de l'état au temps t_1 . Ainsi, l'état au temps t_2 généré par la seconde mise à jour (MaJ 2) dépendant de la deuxième observation (Obs 2) est ignoré. Un *mutex* est donc utilisé pour protéger les informations d'état lorsqu'elles sont déjà

1. ce mécanisme est typiquement celui qui est utilisé lorsqu'on fait une implémentation sous ROS

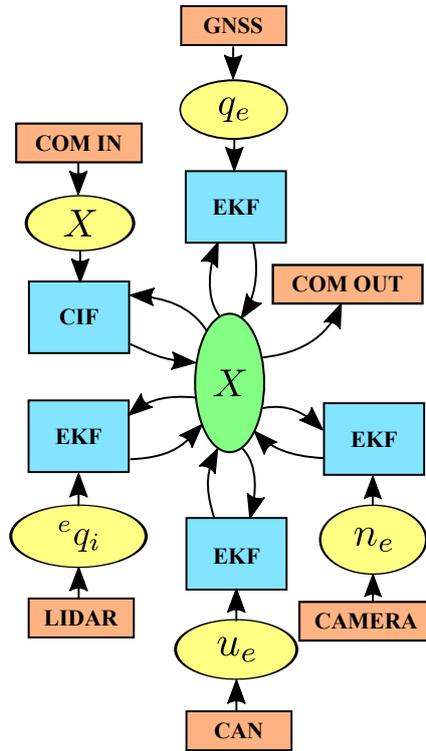


FIGURE 6.1. – Fusion de donnée asynchrone. La CLD X est partagée entre les différents modules de mise à jour et envoyée aux autres agents en coopération.

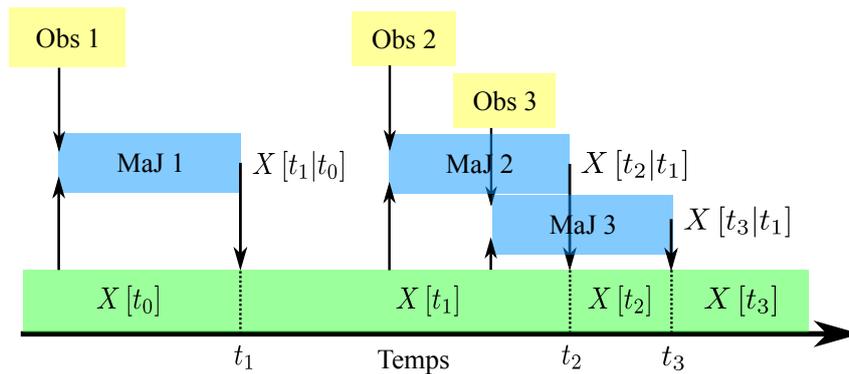


FIGURE 6.2. – Vue temporelle de la fusion de données asynchrone. L'état de la CLD X (en vert) est mis à jour (MaJ en bleu) grâce aux observations (Obs en jaune).

utilisées par un autre *thread*. Cette méthode permet aussi d'éviter qu'une observation ne soit ignorée.

6.3. Modèles et processus de la fusion CLD

6.3.1. Extrapolation de la CLD aux temps des observations

Lorsqu'une nouvelle observation est reçue, l'état doit être extrapolé au temps de cette observation avant de fusionner cette nouvelle information.

Pour chaque agent i le modèle d'évolution utilise ses données cinématiques :

$$X_i [t|t_{-1}] = X_i [t_{-1}|t_{-1}] + V_i [t|t_{-1}], \quad (6.1)$$

où

$$V_i [t|t_{-1}] = \begin{pmatrix} v_i [t_{-1}|t_{-1}] \Delta t \cos \left(\theta_i [t_{-1}|t_{-1}] + \omega_i [t_{-1}|t_{-1}] \frac{\Delta t}{2} \right) \\ v_i [t_{-1}|t_{-1}] \Delta t \sin \left(\theta_i [t_{-1}|t_{-1}] + \omega_i [t_{-1}|t_{-1}] \frac{\Delta t}{2} \right) \\ \omega_i [t_{-1}|t_{-1}] \Delta t \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (6.2)$$

et où $\Delta t = t - t_{-1}$ correspond à l'écart entre le temps de l'observation t et le temps t_{-1} associé à la dernière estimation de la CLD.

Ainsi, on obtient le modèle d'évolution de la CLD globale :

$$X [t|t_{-1}] = X [t_{-1}|t_{-1}] + V [t|t_{-1}], \quad (6.3)$$

avec

$$V [t|t_{-1}] = \begin{pmatrix} V_1 [t|t_{-1}] \\ \vdots \\ V_N [t|t_{-1}] \end{pmatrix}. \quad (6.4)$$

La matrice Jacobienne associée au véhicule i peut ensuite être déterminée comme suit :

$$J_i = \begin{pmatrix} 1 & 0 & -v_i [t_{-1}|t_{-1}] \Delta t s_i & 0 & 0 & \Delta t c_i & -v_i [t_{-1}|t_{-1}] \frac{\Delta t^2}{2} s_i \\ 0 & 1 & v_i [t_{-1}|t_{-1}] \Delta t c_i & 0 & 0 & \Delta t s_i & v_i [t_{-1}|t_{-1}] \frac{\Delta t^2}{2} c_i \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (6.5)$$

où $c_i = \cos \left(\theta_i [t_{-1}|t_{-1}] + \omega_i [t_{-1}|t_{-1}] \frac{\Delta t}{2} \right)$ et $s_i = \sin \left(\theta_i [t_{-1}|t_{-1}] + \omega_i [t_{-1}|t_{-1}] \frac{\Delta t}{2} \right)$.

On utilise une matrice de covariance pour prendre en compte les incertitudes de modèle. Le biais et les grandeurs cinématiques étant supposés constants tant qu'une

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

nouvelle observation n'est pas disponible, on ajoute une variance pour indiquer que ces valeurs deviennent incertaines :

$$\Sigma_{model,i} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \nu_{bias} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \nu_{bias} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \nu_v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \nu_\omega \end{pmatrix}. \quad (6.6)$$

La variance du biais GNSS ν_{bias} est supposée être constante et identique selon la direction Est et la direction Nord du repère ENU. Cette hypothèse simplificatrice est due aux caractéristiques des récepteurs GNSS utilisés dans ces travaux. Cette incertitude permet à la position de pouvoir converger vers une estimation plus juste tout en conservant ce biais constant lorsque celui-ci n'est plus observé. Sa variance est un paramètre de réglage et est choisie faible pour éviter une divergence de ce biais, si celui-ci n'est pas observé. Les variances des erreurs de vitesse longitudinale ν_v et de vitesse de rotation ν_ω peuvent être estimées expérimentalement à partir de la vérité terrain et de leurs valeurs provenant du bus CAN, lors d'une phase de réglage sur un scénario de conduite en convoi destiné à la calibration des capteurs. On suppose avec cette matrice qu'il n'existe pas d'erreur de modèle pour la pose. Cette pose n'est en effet pas supposée constante au contraire du biais et des grandeurs cinématiques. En pratique, le modèle d'évolution de la pose est inexact, mais il dépend des grandeurs cinématiques v et ω . L'erreur de modèle de ces grandeurs va ainsi se propager sur la pose et permettre de prendre en compte l'inexactitude de ce modèle d'évolution. On peut par exemple noter que le modèle d'évolution de la pose ne dépend pas de l'accélération du véhicule. En modélisant le bruit de cette façon, on suppose que c'est le modèle d'évolution de la vitesse qui doit en tenir compte et non le modèle d'évolution de la pose.

La matrice de covariance de l'erreur de prédiction liée au modèle d'évolution peut être calculée à partir de la matrice de covariance de l'état estimée précédemment $\Sigma_X [t_{-1}|t_{-1}]$. En considérant les modèles d'évolution $u [t|t_{-1}] = u [t_{-1}|t_{-1}] + \int_{t_{-1}}^t \epsilon_u(\tau) d\tau$ et $b [t|t_{-1}] = b [t_{-1}|t_{-1}] + \int_{t_{-1}}^t \epsilon_b(\tau) d\tau$ de façon à avoir des erreurs ϵ_u et ϵ_b indépendantes du temps d'échantillonnage Δt (celui-ci n'étant pas supposé constant), la matrice du bruit de modèle Σ_{model} est multipliée par $|\Delta t|$ (la valeur absolue est nécessaire si $t < t_{-1}$) :

$$\Sigma_X [t|t_{-1}] = J \Sigma_X [t_{-1}|t_{-1}] J^T + \Sigma_{model} |\Delta t|, \quad (6.7)$$

où

$$J = \begin{pmatrix} J_0 & 0_7 & \cdots & \cdots & 0_7 \\ 0_7 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & J_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0_7 \\ 0_7 & \cdots & \cdots & 0_7 & J_N \end{pmatrix}, \quad (6.8)$$

et

$$\Sigma_{model} = \begin{pmatrix} \Sigma_{model,0} & 0_7 & \cdots & \cdots & 0_7 \\ 0_7 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \Sigma_{model,i} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0_7 \\ 0_7 & \cdots & \cdots & 0_7 & \Sigma_{model,N} \end{pmatrix}. \quad (6.9)$$

6.3.2. Mises à jour de la CLD

6.3.2.1. Mise à jour de Kalman

Lorsqu'une nouvelle mesure est reçue, elle peut être utilisée dans un filtre de Kalman grâce au modèle d'observation lui correspondant.

La mesure $\mathbf{Z}[t]$ au temps d'acquisition t est fusionnée avec l'état $\mathbf{X}[t|t_{-1}]$ provenant de l'état au temps t_{-1} , ce dernier ayant été extrapolé au temps t grâce au modèle d'évolution $obs()$. L'innovation $Y[t]$ est obtenue en calculant la différence entre la mesure observée et la mesure prédite à partir de l'état :

$$Y[t] = Z[t] - \mathbf{obs}(X[t|t_{-1}]). \quad (6.10)$$

La matrice Jacobienne H peut ensuite être calculée (dans le cas où le modèle d'observation est linéaire, cette Jacobienne est égale à la matrice d'observation). Cette Jacobienne est donc une fonction de l'état :

$$H = \mathbf{obs}_H(X[t|t_{-1}]). \quad (6.11)$$

La matrice de covariance de l'innovation $\Sigma_Y[t]$ est ensuite calculée :

$$\Sigma_Y[t] = H\Sigma_X[t|t_{-1}]H^T + \Sigma_Z[t]. \quad (6.12)$$

À partir de cette matrice, de la covariance de l'état $\Sigma_X[t|t_{-1}]$ et de la Jacobienne H , le gain de Kalman est calculé :

$$K = \Sigma_X[t|t_{-1}]H^T\Sigma_Y^{-1}[t]. \quad (6.13)$$

En utilisant ce gain, l'état $X[t|t]$ et sa matrice de covariance $\Sigma_X[t|t]$ sont estimés :

$$X[t|t] = X[t|t_{-1}] + KY[t], \quad (6.14)$$

$$\Sigma_X[t|t] = (I - KH)\Sigma_X[t|t_{-1}](I - KH)^T + K\Sigma_Z[t]K^T. \quad (6.15)$$

Cette dernière expression est connue sous le nom de « forme de Joseph ».

6.3.2.2. Mise à jour par intersection de covariance

Lorsqu'une CLD provenant de la communication doit être fusionnée avec celle que le véhicule a en mémoire, les erreurs d'estimation des deux CLDs sont corrélées. Comme nous l'avons montré au chapitre 3 dans un cas simplifié à une dimension, ceci se produit dès que le véhicule coopératif a utilisé la CLD du égo-véhicule pour estimer la sienne. Une consanguinité de données apparaît donc et, si l'on cherche à fusionner ces CLDs avec un filtre de Kalman, qui suppose ces informations affectées

par des erreurs décorréelées, le résultat sera inconsistant et une matrice de covariance surconfiante est alors obtenue en général. Ne connaissant pas nécessairement l'algorithme utilisé par le véhicule échangeant sa CLD, il est difficile d'estimer cette corrélation. Même dans le cas où, les mêmes algorithmes fonctionnent sur les deux véhicules, ils ne seront certainement pas codés à l'identique. Un filtre par intersection de covariance (CIF) introduit par [Julier et Uhlmann, 1997b] permet alors d'assurer une fusion ne risquant pas de fournir une estimation surconfiante de la matrice de covariance. Ceci aussi a été montré au chapitre 3.

La mise à jour par intersection de covariance que nous utilisons est formulée comme celle de Kalman.

L'innovation et la Jacobienne sont tout d'abord calculées :

$$Y[t] = Z[t] - \mathbf{obs}(X[t|t_{-1}]), \quad (6.16)$$

$$H = \mathbf{obs}_H(X[t|t_{-1}]). \quad (6.17)$$

En suivant la méthode de l'intersection de covariance, on ajoute les pondérations $1/\omega$ et $1/1-\omega$ (avec $\omega \neq 1$ et $\omega \neq 0$ qui sont des cas particuliers faciles à gérer) aux matrices de covariance de l'état $\Sigma_X[t|t_{-1}]$ et d'observation $\Sigma_Z[t]$, ce qui permet de calculer un gain et mettre à jour la covariance avec une généralisation de la forme de Joseph avec des $1/\omega$ et $1/1-\omega^2$:

$$\Sigma_{Y,\omega}[t] = H \left(\frac{1}{\omega} \Sigma_X[t|t_{-1}] \right) H^T + \frac{1}{1-\omega} \Sigma_Z[t], \quad (6.18)$$

$$K_\omega = \frac{1}{\omega} \Sigma_X[t|t_{-1}] H^T \Sigma_{Y,\omega}^{-1}[t], \quad (6.19)$$

$$\begin{aligned} \Sigma_{X,\omega}[t|t] &= (I - K_\omega H) \left(\frac{1}{\omega} \Sigma_X[t|t_{-1}] \right) (I - K_\omega H)^T \\ &\quad + K_\omega \frac{1}{1-\omega} \Sigma_Z[t] K_\omega^T. \end{aligned} \quad (6.20)$$

Cette formulation est détaillée dans l'annexe D et on discute de son intérêt par rapport à d'autres formulations plus classiques.

Les pondérations sont choisies de façon à minimiser une norme de la matrice de covariance de l'état final $\Sigma_X[t|t]$:

$$\omega_{min} = \mathbf{arg} \min_{\omega \in [0,1]} (\mathbf{norm}(\Sigma_{X,\omega}[t|t])). \quad (6.21)$$

Cette norme est typiquement la trace ou le déterminant. En pratique, nous avons remarqué des résultats similaires lors de l'utilisation de ces deux normes. Une fois la pondération ω_{min} trouvée, l'état $X[t|t]$ et sa matrice de covariance $\Sigma_X[t|t]$ sont déterminés :

$$X[t|t] = X[t|t_{-1}] + K_{\omega_{min}} Y[t], \quad (6.22)$$

$$\Sigma_X[t|t] = \Sigma_{X,\omega_{min}}[t|t]. \quad (6.23)$$

2. Cette forme somme des matrices définies positives et est donc numériquement bien conditionnée.

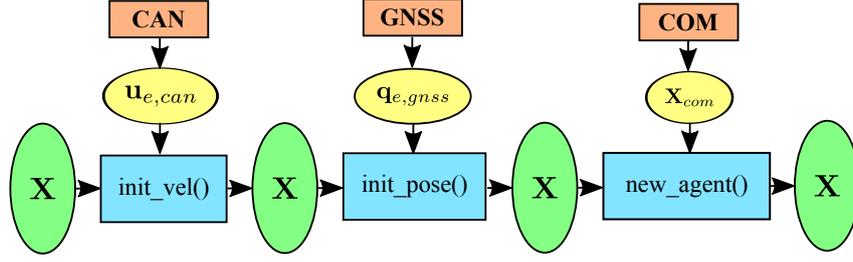


FIGURE 6.3. – Initialisation de la fusion coopérative.

Dans les deux cas particuliers, on montre que (cf annexe D) :

Si $\omega_{min} = 0$:

$$X[t|t] = X[t|t-1] + \Sigma_X[t|t] H^T \Sigma_Z^{-1}[t] (Z[t] - H X[t|t-1]), \quad (6.24)$$

$$\Sigma_X[t|t] = (H^T \Sigma_Z^{-1}[t] H)^{-1}. \quad (6.25)$$

Et si $\omega_{min} = 1$:

$$X[t|t] = X[t|t-1], \quad (6.26)$$

$$\Sigma_X[t|t] = \Sigma_X[t|t-1]. \quad (6.27)$$

Pour effectuer ces différentes étapes d'observation, il est nécessaire de déterminer le modèle d'observation de chaque information reçue. C'est ce que nous allons voir dans les sections suivantes.

6.3.3. Initialisation de l'état

La figure 6.3 décrit l'initialisation de l'algorithme de fusion coopérative asynchrone. L'état X est initialisé arbitrairement tel que $X = 0_{7 \times 1}$ et sa matrice de covariance Σ_X par $\Sigma_X = I_{7 \times 7} \nu_{init}$ où ν_{init} est la variance initiale. Même si les positions sont exprimées dans un repère local ENU, l'état initial pouvant être très éloigné de l'état réel, cette variance est choisie assez importante. Afin d'obtenir un état correctement initialisé, l'état est complété lorsqu'une nouvelle information est disponible (nouvelle vitesse provenant du bus CAN, nouvelle pose GNSS ou nouvel état d'un agent communicant sa CLD). Lorsque la première mesure de vitesse est reçue, la vitesse de l'état est initialisée en utilisant cette mesure et la matrice de covariance associée. De la même façon, lorsque la première pose GNSS est disponible, la pose et la matrice de covariance associée sont initialisées par cette observation. Lorsqu'un message provenant de la communication contient un nouvel agent, le modèle d'observation de la communication permet alors de le rajouter à la CLD du égo-véhicule. Nos résultats réels ont été obtenus à partir de deux véhicules uniquement, les problèmes d'ajout et de suppression d'agent de la CLD n'ont donc pas été étudiés en pratique. Ces problèmes seront cependant discutés dans la section 6.5.

6.3.4. Modèles d'observation

6.3.4.1. Mesures cinématiques

Capteurs proprioceptifs du véhicule Les encodeurs sont des capteurs permettant de mesurer les variations angulaires des roues en comptant des impulsions. Ces capteurs permettent de déterminer la vitesse longitudinale et la vitesse de rotation du véhicule, de façon redondante avec la mesure d'un gyromètre de lacet. Des disques troués ou équipés d'aimants sont fixés aux axes des roues et des capteurs optiques ou à effet hall (magnétiques) permettent de mesurer des fronts montants et descendants. En comptant ces fronts, on peut alors savoir de combien d'incrémentes chaque roue a tourné. On peut estimer le déplacement Δs de chaque roue i pour tout échantillon k à partir de l'équation suivante :

$$\Delta s_i[k] = (cpt_i[k] - cpt_i[k-1]) \frac{cir_i}{res}, \quad (6.28)$$

où $cpt_i[k]$ est le compteur d'impulsions de la roue i à l'itération k , res correspond à la résolution de l'encodeur et cir_i est la circonférence de la roue i .

Sur les Zoes, 48 impulsions peuvent être mesurées pour chaque tour de roue. Le périmètre de la roue étant d'environ 1,92 m, une erreur de comptage d'une impulsion implique une erreur pouvant aller jusqu'à 4 cm ($cir_i/res \approx 1,92/48 = 0,04$ m) sur la distance parcourue par une roue en considérant un roulement sans glissement. Malgré cette résolution réduite comparée aux encodeurs pouvant être présents sur de nombreux robots avec des résolutions pouvant être de 4 096 impulsions par tour de roue, une précision de 4 cm reste très correcte. Le périmètre des roues étant très dépendant du gonflement et de l'usure des pneus, cette précision peut être considérablement réduite si ce périmètre n'est pas correctement estimé.

Une façon simple d'estimer le déplacement longitudinal et angulaire du véhicule est d'utiliser un modèle de robot à deux roues (le modèle char) en prenant seulement les deux roues arrière :

$$\Delta s[k] = \frac{1}{2} (\Delta s_r[k] + \Delta s_l[k]), \quad (6.29)$$

$$\Delta \theta[k] = \frac{1}{w} (\Delta s_r[k] - \Delta s_l[k]), \quad (6.30)$$

où Δs_r représente le déplacement de la roue arrière droite, Δs_l celui de la roue arrière gauche et w la distance entre les deux roues arrières ($w = 1,510$ m). Comme chaque roue est équipée d'encodeurs, cette odométrie peut aussi être calculée par fusion à partir des déplacements des quatre roues. Pour ce faire, il faut calculer l'orientation des roues avant par rapport au châssis, par exemple à partir de l'angle au volant qui transmis sur le bus CAN [Welte *et al.*, 2019].

Dans le cas de notre modèle d'évolution, la vitesse longitudinale v et la vitesse de rotation ω sont estimées de la manière suivante :

$$v[k] = \frac{\Delta s[k]}{\Delta t}, \quad (6.31)$$

$$\omega[k] = \frac{\Delta \theta[k]}{\Delta t}. \quad (6.32)$$

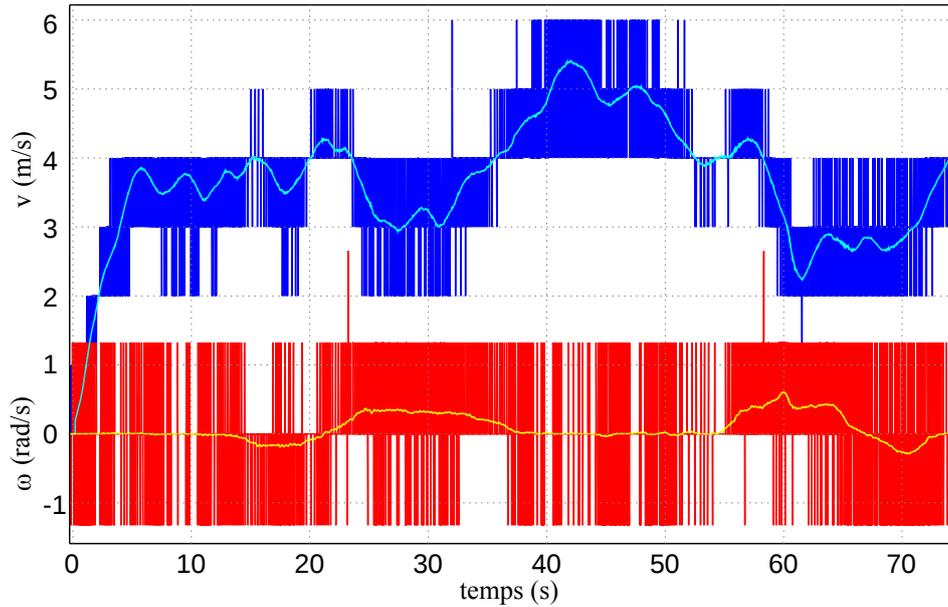


FIGURE 6.4. – Vitesses longitudinales (m/s) calculées par l'odométrie (en bleu foncé) et fournies par le bus CAN (en bleu clair) et vitesses de rotation (rad/s) calculées par l'odométrie (en rouge) et fournies par le bus CAN (en jaune).

La période Δt étant très faible ($\Delta t = 0,02$ s), les différences d'impulsions sont elles aussi très faibles, une variation d'une impulsion dans l'une de ces différences engendre alors une variation de 2m/s sur la vitesse estimée d'une roue ($cir_i/res/\Delta t \approx 1,92/48/0,02 = 2$ m/s). Ces mesures de vitesses sont assez imprécises. La figure 6.4 montre la vitesse longitudinale (en bleu) et la vitesse de rotation (en rouge) estimées à partir des encodeurs. On peut voir que ces estimations sont très discrétisées.

D'un autre côté, le calculateur ABS et l'ESP reliés au bus CAN fournissent une estimation de la vitesse longitudinale et de la vitesse de rotation, calculées à partir des encodeurs ainsi que d'un gyromètre pour la vitesse de rotation. La vitesse longitudinale est affichée en bleu clair et la vitesse de rotation en jaune. Dans nos expérimentations, nous préférons utiliser ces informations fournies par le bus CAN plutôt que les mesures brutes des encodeurs.

Calibration des vitesses provenant du bus CAN La calibration des capteurs permet de réduire les erreurs dues à de mauvais paramètres. Par exemple, un gyromètre non calibré peut renvoyer une mesure de vitesse angulaire assez biaisée.

Dans le cas d'une navigation à l'estime, l'orientation courante ne dépend que de l'orientation initiale et de la vitesse de rotation. Sur la figure 6.5, on observe que l'erreur d'orientation présente une pente qui est due à un biais sur la vitesse de rotation donnée par le gyromètre de l'ESP. On estime ce biais en calculant la moyenne de l'erreur signée entre la vitesse de rotation provenant du bus CAN et la vitesse de rotation estimée par la vérité terrain. Sur un essai, on a observé un biais de 0,004075 rad/s pour le véhicule suiveur et un biais de 0,003463 rad/s pour le Véhicule leader. En supprimant ce biais, on élimine alors la pente observée (Fig. 6.5).

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

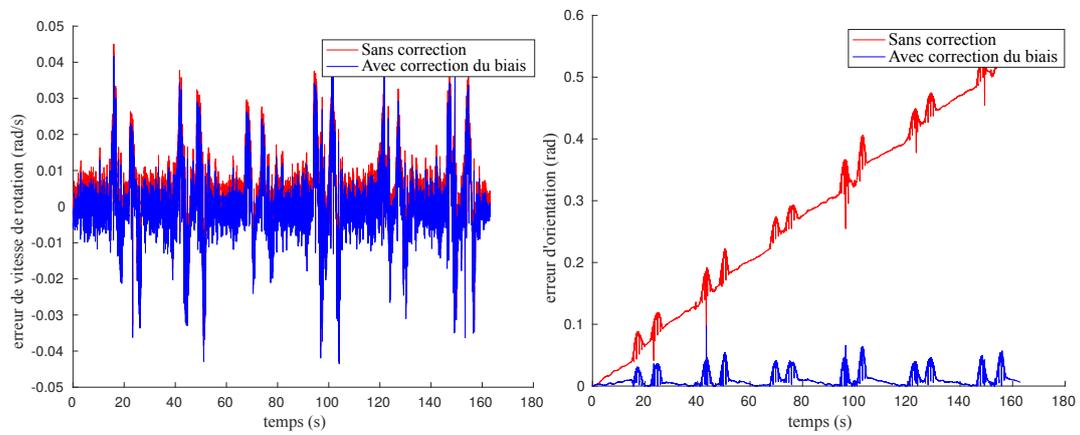


FIGURE 6.5. – Correction de la vitesse de rotation et influence de cette vitesse sur l'orientation du véhicule.

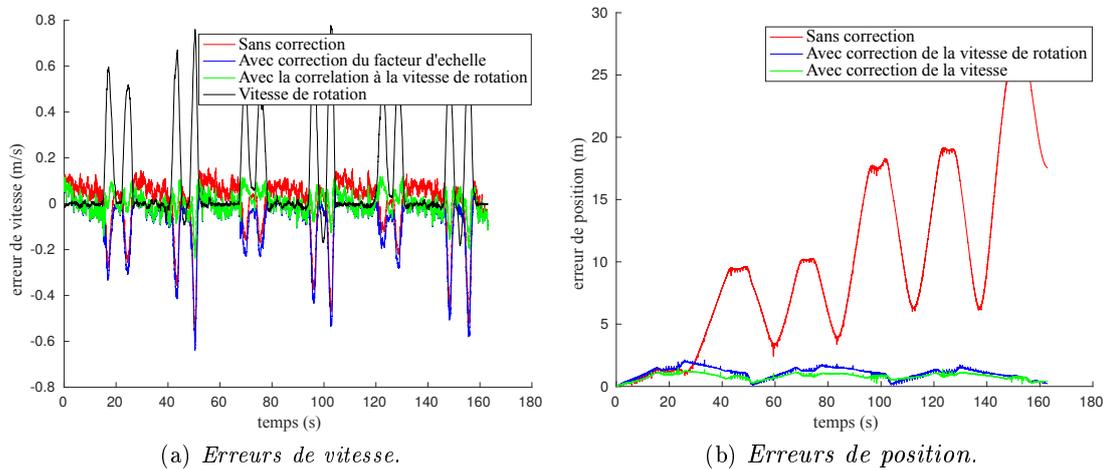


FIGURE 6.6. – Erreurs de vitesse du Véhicule leader avec et sans corrections et influence sur les erreurs de position.

TABLE 6.1. – Erreurs de position $\|\bar{\epsilon}_p\|$, de vitesse $|\bar{\epsilon}_v|$, d'orientation $|\bar{\epsilon}_\theta|$ et de vitesse de rotation $|\bar{\epsilon}_\omega|$ sans correction avec correction du biais du gyromètre (0.003325 rad/s) en ajoutant un facteur d'échelle (1.0157) puis en utilisant un bras de levier (0.5284 m). Ces corrections sont cumulées pour réduire au maximum les erreurs.

	Sans correction	Biais gyro	Facteur d'échelle	Bras de levier
$ \bar{\epsilon}_\omega $ (rad/s)	0.006768	0.006202	0.006202	0.006202
$ \bar{\epsilon}_\theta $ (rad)	0.2819	0.0117	0.0117	0.0117
$ \bar{\epsilon}_v $ (m/s)	0.088207	0.088207	0.083360	0.037656
$\ \bar{\epsilon}_p\ $ (m)	10.0807	1.0799	1.2150	0.8014

Une fois l'erreur d'orientation corrigée, on peut corriger ensuite l'erreur longitudinale. Si l'on considère uniquement les lignes droites, lorsque la vitesse de rotation est proche de 0, on peut observer que l'erreur de vitesse est constamment supérieure à 0 (en rouge sur la figure 6.6a). Les périmètres des roues pouvant changer selon l'état des pneus, il est important de pouvoir les estimer régulièrement pour ne pas avoir des vitesses erronées pour chaque roue. En pratique, ce phénomène peut être considéré comme un facteur d'échelle sur la vitesse. Après calibrage, nous avons trouvé un facteur de 1,01 pour les deux véhicules. Ainsi la moyenne de l'erreur longitudinale lorsque le véhicule se déplace en ligne droite est considérablement réduite (en bleu sur la figure 6.6b).

Une fois cette erreur corrigée, on peut voir que l'erreur restante semble corrélée à la vitesse de rotation du véhicule (en bleu sur la figure 6.6a). En effet, les pneus du véhicule se déforment dans les virages. L'estimation de la vitesse longitudinale par l'ESP étant basée uniquement sur un modèle cinématique et non dynamique, l'erreur résultante est conservée. En ajoutant la vitesse de rotation en valeur absolue multipliée par un bras de levier, on arrive à corriger la vitesse longitudinale (en vert sur la figure 6.6a).

La table 6.1 présente les erreurs de vitesse de rotation, d'orientation, de vitesse longitudinale et de position. Lorsque le biais du gyromètre est corrigé, l'erreur de vitesse de rotation est réduite et les erreurs d'orientation et de position sont aussi réduites. Le facteur d'échelle appliqué à la vitesse permet ensuite de réduire son erreur. Enfin l'ajout du bras de levier permet de rendre les erreurs de vitesse et de position plus faibles.

Modèle d'observation Le modèle d'observation des mesures cinématiques obtenues par l'intermédiaire du bus CAN consiste simplement à utiliser le vecteur cinématique

$$u_e [t|t-1] = [v_e [t|t-1] \ \omega_e [t|t-1]]^T, \quad (6.33)$$

du véhicule inclus dans son vecteur d'état :

$$\mathbf{obs}_{CAN} (X [t|t-1]) = u_e [t|t-1]. \quad (6.34)$$

Ce modèle est linéaire et on en déduit une matrice H_e très simple pour le véhicule

$$H_e = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (6.35)$$

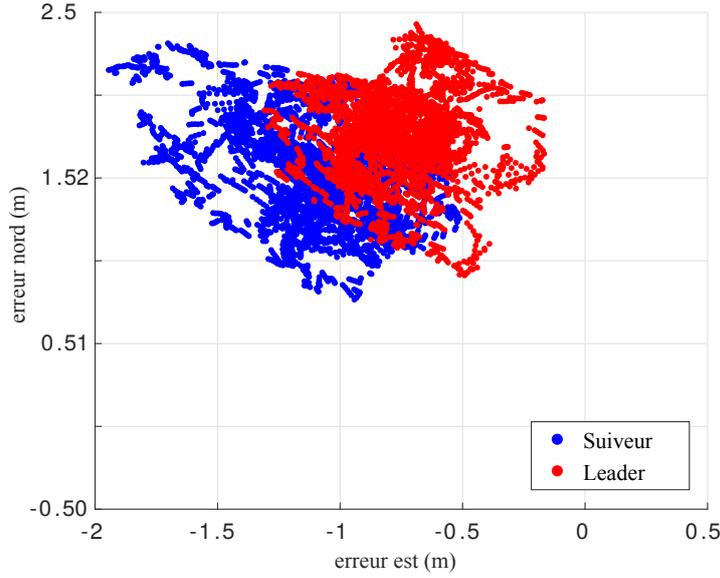


FIGURE 6.7. – Erreurs des positions GNSS de deux véhicules dans un scénario de conduite en convoi.

Ce qui donne comme matrice d'observation de la CLD :

$$H = \begin{pmatrix} 0_{2 \times 7(e-1)} & H_e & 0_{2 \times 7(N-e)} \end{pmatrix}. \quad (6.36)$$

6.3.4.2. Poses GNSS

Pour connaître la pose d'un véhicule précisément, il est possible d'utiliser un récepteur GNSS haut de gamme, comme une NovAtel SPAN-CPT, qui couple une centrale inertielle à un récepteur GNSS multi-constellation et multi-fréquence avec des corrections RTK. Il est cependant difficilement envisageable de commercialiser des véhicules autonomes équipés de récepteurs GNSS si onéreux. De plus, même si les corrections RTK se démocratisaient, il serait difficile de les recevoir en permanence (lorsque le véhicule s'éloigne des zones couvertes par les réseaux de télécommunication) et le mode RTK est un mode « fragile » pour les véhicules intelligents, car la perte du mode RTK est fréquente. Nous utilisons donc en pratique un récepteur GNSS Ublox 8 mono-fréquence utilisant les constellations Navstar GPS et GLO-NASS. Nous l'utilisons ici en couplage lâche, en utilisant directement la pose estimée par le récepteur Ublox, l'angle de cap étant supposé être l'angle de « track » du calcul GNSS obtenu avec les mesures de Doppler. Cette mesure n'est pas disponible à l'arrêt et suppose que les glissements des pneus du véhicule sont négligeables.

On observe expérimentalement la présence d'un biais sur la position calculée par un récepteur GNSS bas-coût comme un Ublox. La figure 6.7 montre les erreurs GNSS de deux véhicules en convoi. On observe que les biais semblent corrélés, mais ne sont pas identiques. L'estimation de la corrélation entre ces biais étant difficile, nous avons choisi de modéliser le biais de chaque agent comme étant indépendant. Chaque véhicule i a ainsi un biais GNSS $b_i[t|t-1]$ dans son état qui est estimé pendant la

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

fusion. Nous supposons ce biais lentement variable et tel que sa dérivée soit nulle à un bruit blanc près. C'est le modèle d'évolution qui a été présenté dans l'équation 6.2.

Le modèle d'observation des poses GNSS est donc le suivant :

$$\text{obs}_{GNSS}(X[t|t_{-1}]) = q_e[t|t_{-1}] + \begin{pmatrix} b_e[t|t_{-1}] \\ 0 \end{pmatrix}. \quad (6.37)$$

L'utilisation d'un biais GNSS dans ce modèle rajoute des degrés de liberté à l'état. Ainsi, la partie de l'erreur qui n'est pas prise en compte par la matrice de covariance de la pose GNSS est considérée comme faisant partie du biais.

Ce modèle d'observation étant linéaire, la matrice H permettant de transformer la matrice de covariance de l'état est déterminée facilement :

$$H_e = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.38)$$

Soit au niveau global de la CLD :

$$H = \begin{pmatrix} 0_{3 \times 7(e-1)} & H_e & 0_{3 \times 7(N-e)} \end{pmatrix}. \quad (6.39)$$

Un capteur permettant d'observer ce biais GNSS est alors nécessaire, en utilisant par exemple des observations de l'environnement qui sont comparées à une carte. Si ce biais n'est pas observé, la pose peut diverger.

6.3.4.3. CLD reçue de la communication

Plaçons-nous en régime permanent et supposons que chaque CLD contienne à tout moment les états de tous les véhicules en coopération. La CLD complète reçue via la communication est fusionnée avec la CLD locale du véhicule. Le modèle d'observation est donc l'identité :

$$\text{obs}_{COM}(X[t|t_{-1}]) = X[t|t_{-1}], \quad (6.40)$$

$$H = I_{7N}. \quad (6.41)$$

En pratique, on pourrait limiter la taille de la CLD. Les états des véhicules s'étant trop éloignés du égo-véhicule seraient supprimés de la CLD du égo-véhicule et la matrice de covariance serait réduite de la même façon que celle du vecteur d'observation. Inversement, les véhicules proches provenant de la CLD reçue qui ne seraient pas présents dans la CLD du égo-véhicule seraient ajoutés après la fusion. Les lignes et colonnes de la matrice de covariance associées à ces états seraient ajoutées à la matrice de covariance du égo-véhicule.

On suppose que chaque véhicule a un identifiant unique³. On peut adopter comme stratégie de trier les poses par ordre croissant des identifiants par exemple. Au moment de faire la fusion, les listes des identifiants de la CLD du égo-véhicule et de la CLD reçue doivent correspondre rigoureusement.

3. Par exemple son immatriculation ou bien l'adresse MAC de son modem 802.11p qui ne peut être modifiée

6.3.4.4. Détection des bords de voie

Nous supposons qu'une caméra détecte les bords de la voie à partir des marquages ou de l'espace navigable et en déduit le centre de cette voie. Le centre ou les bords de voie sont alors comparés à une carte avant d'être utilisés dans la fusion. Cette observation de l'environnement permet alors d'observer la composante latérale à la voie du biais GNSS, ce qui est très important pour la convergence des biais GNSS comme nous l'avons évoqué précédemment.

Une caméra intelligente (de type Mobileye équipant déjà des véhicules du marché) peut fournir directement ces observations. Elle ne fournit toutefois pas les matrices de covariance associées à ces observations, ce qui la rend assez difficile à utiliser et ce qui nécessite de faire des expérimentations de réglage de paramètres avec une vérité terrain. Elle est de plus destinée à des scénarios de conduite sur autoroutes avec peu de courbure, du fait du champ de vue étroit du capteur. Cette caméra utilise de plus des modèles de bord de voie limitant la détection de bord de voie trop courbé. Elle modélise les marquages sous la forme d'un polynôme dans un repère capteur à l'avant du véhicule. Ce modèle ne permettant pas de modéliser des routes trop courbées, cette caméra ne peut pas être utilisée dans un rond-point par exemple. Sur notre piste expérimentale Séville de l'UTC, une caméra Mobileye ne fournit ainsi des détections de marquage que dans la ligne droite, ce qui est trop peu pour nos essais. Une détection différente des bords de voie, fournissant des matrices de covariance associées à ces mesures peut toutefois être utilisée.

Compte tenu de ce problème, nous avons donc simulé l'observation de bords de voie en calculant la distance signée entre la position de la vérité terrain et le centre de la voie en utilisant la projection des *lanelets*. Un bruit blanc, centré et Gaussien est ajouté à cette distance.

Le modèle d'observation utilise lui aussi la correspondance avec projection des *lanelets* entre la pose du égo-véhicule $q_e [t|t_{-1}]$ et la carte haute définition M :

$$\left[p_{frenet}, \vec{t}_{frenet}, \vec{n}_{frenet} \right] = \text{map_match}(q_e [t|t_{-1}], M). \quad (6.42)$$

Cette correspondance donne le repère de Frenet le plus proche de la position $p_e [t|t_{-1}]$. Ce repère est composé de son origine p_{frenet} et des axes composés de la tangente unitaire \vec{t}_{frenet} et de la normale unitaire \vec{n}_{frenet} au centre de la voie.

Afin de déterminer la distance latérale signée $n_e [t|t_{-1}]$, on projette la position $p_e [t|t_{-1}]$ sur la normale \vec{n}_{frenet} :

$$\text{obs}_{CAM}(X [t|t_{-1}]) = (p_e [t|t_{-1}] - p_{frenet})^T \cdot \vec{n}_{frenet}. \quad (6.43)$$

La matrice H est alors déterminée à partir de cette fonction affine :

$$H_e = \begin{pmatrix} -n_{x,frenet} & -n_{y,frenet} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (6.44)$$

$$H = \begin{pmatrix} 0_{1 \times 7(e-1)} & H_e & 0_{1 \times 7(N-e)} \end{pmatrix}. \quad (6.45)$$

6.3.4.5. Poses relatives mesurées avec des capteurs extéroceptifs

On rappelle que dans ces travaux, les poses relatives sont mesurées à partir d'un capteur LiDAR. Le modèle d'observation de la pose relative du véhicule i dans le

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

repère du égo-véhicule e est donné à partir des poses $q_i [t|t_{-1}]$ et $q_e [t|t_{-1}]$:

$$\mathbf{obs}_{LiDAR}(X [t|t_{-1}]) = \ominus q_e [t|t_{-1}] \oplus q_i [t|t_{-1}]. \quad (6.46)$$

Cette fonction n'étant pas linéaire, des matrices Jacobiennes doivent être calculées en combinant $J_{\ominus} = \frac{\partial(\ominus q_e)}{\partial q_e}$ et $J_{\oplus} = \frac{\partial(^e q \oplus q_i)}{\partial(^e q, q_i)}$.

En posant $c = \cos(\theta_e [t|t_{-1}])$ et $s = \sin(\theta_e [t|t_{-1}])$, on a

$$H_e = \begin{pmatrix} -c & -s & -(x_i [t|t_{-1}] - x_e [t|t_{-1}]) s + (y_i [t|t_{-1}] - y_e [t|t_{-1}]) c & 0 & 0 & 0 & 0 \\ s & -c & -(x_i [t|t_{-1}] - x_e [t|t_{-1}]) c - (y_i [t|t_{-1}] - y_e [t|t_{-1}]) s & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (6.47)$$

$$H_i = \begin{pmatrix} c & s & 0 & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (6.48)$$

$$H = \begin{pmatrix} 0_{3 \times 7(e-1)} & H_e & 0_{3 \times 7(i-1-e)} & H_i & 0_{3 \times 7(N-i)} \end{pmatrix}. \quad (6.49)$$

On peut voir grâce à ce modèle d'observation l'intérêt d'avoir l'état du égo-véhicule ainsi que ceux des véhicules environnants dans l'état à fusionner. En effet, si l'on déterminait la pose du égo-véhicule à partir de la pose relative et de la pose communiquée, on améliorerait seulement la pose du égo-véhicule. Dans notre cas, nous améliorons la pose du égo-véhicule ainsi que celle du véhicule perçu tout en prenant en compte les corrélations entre ces deux poses introduites par ce modèle d'observation.

6.4. Résultats expérimentaux

6.4.1. Scénario

Le scénario de conduite en convoi utilisé pour obtenir les résultats est identique à celui utilisé dans la section 5. Deux Renault Zoes roulent en convoi sur la route d'essai de Séville, composée de deux ronds-points liés par une ligne droite. Les deux véhicules sont équipés d'Ublox 8 pour l'obtention de la pose GNSS. La vérité terrain est assurée par des NovAtel SPAN-CPT, des centrales inertielle couplées à des récepteurs GNSS avec corrections RTK. Les données cinématiques des véhicules sont fournies par les bus CAN. Le scan LiDAR correspond à la couche horizontale du LiDAR SICK LD-MRS. Les bords de voies sont simulés à partir de la carte et de la vérité terrain pour donner une mesure de la distance latérale au centre de la voie. Un bruit Gaussien centré avec un écart type de 20 cm est ajouté à cette distance. La communication est elle aussi simulée.

Cette fusion est implémentée en Python sous ROS (Robot Operating System). Les différents algorithmes de cette localisation sont exécutés sur les messages d'une base de données rejouée en vitesse réelle. La solution proposée est en effet suffisamment rapide pour une exécution temps réel sur un PC standard.

6.4.2. Résultats

Dans ces résultats, chaque véhicule possède sa propre CLD contenant la pose du Véhicule leader q_l (comme *leader vehicle*) et la pose du véhicule suiveur q_f (comme *follower vehicle*). À partir de ces poses absolues, la pose du Véhicule leader dans le repère du véhicule suiveur ${}^f q_l$ et la pose du véhicule suiveur dans le repère du Véhicule leader ${}^l q_f$ peuvent être déterminées.

$\|\bar{\epsilon}_p\|$ correspond ici à la moyenne des erreurs de position 2D sur les K échantillons par rapport à la vérité terrain :

$$\|\bar{\epsilon}_p\| = \frac{1}{K} \sum_{k=1}^K \|p_k^* - p_k\|, \quad (6.50)$$

$|\bar{\epsilon}_\theta|$ correspond à la moyenne des valeurs absolues des erreurs d'orientation pour les K échantillons :

$$|\bar{\epsilon}_\theta| = \frac{1}{K} \sum_{k=1}^K |\theta_k^* - \theta_k|. \quad (6.51)$$

Finalement, le taux de couverture est calculé pour une probabilité d'erreur de 5% avec un test du χ^2 . C correspond au pourcentage des K échantillons validant le test suivant :

$$(q_k^* - q_k)^T \left(\Sigma_{q_k^*} + \Sigma_{q_k} \right)^{-1} (q_k^* - q_k) < \chi_{3;0.05}^2, \quad (6.52)$$

où $\chi_{3;0.05}^2 = 7.815$ est le quantile d'une distribution du χ^2 avec 3 degrés de liberté (x, y, θ) et une confiance de 95%. La matrice de covariance de la pose de la vérité terrain $\Sigma_{q_k^*}$ n'est pas négligeable comparé aux mesures LiDAR et doit donc être prise en compte.

6.4.3. Estimation du biais GNSS

Lorsque les bords de voies sont détectés et mis en correspondance avec une carte haute définition, le biais GNSS peut être estimé. En pratique, seule la composante orthogonale à la route est estimée, cependant, si le véhicule se déplace sur une route présentant des variations d'orientation (comme c'est le cas), le biais total est estimé, en particulier après un virage. La figure 6.8 montre les erreurs GNSS et les biais estimés dans la CLD du véhicule suiveur pour ce véhicule (en bleu et en noir) et dans la CLD du Véhicule leader pour ce véhicule (en magenta et en rouge). On voit que le biais est correctement estimé et reste au centre du bruit. Même si des corrélations sont observées entre les erreurs GNSS des deux récepteurs Ublox, les biais GNSS ne sont pas identiques. Ces derniers sont différents d'un véhicule à l'autre et changent durant les 10 min du scénario.

À partir des biais estimés, les erreurs GNSS sont corrigées. La figure. 6.9 représente ces erreurs débiaisées avec des intervalles de confiance à 95%. L'incertitude horizontale donnée par le Ublox 8 est supposée être l'écart type⁴. Les intervalles de confiances selon l'axe Est ou l'axe Nord sont ainsi identiques. On voit sur la figure 6.9

4. Nous avons utilisé un message Ublox contenant des valeurs appelées « horizontal accuracy » et « vertical accuracy » mais dont la signification mathématique n'est pas décrite dans la documentation du récepteur Ublox.

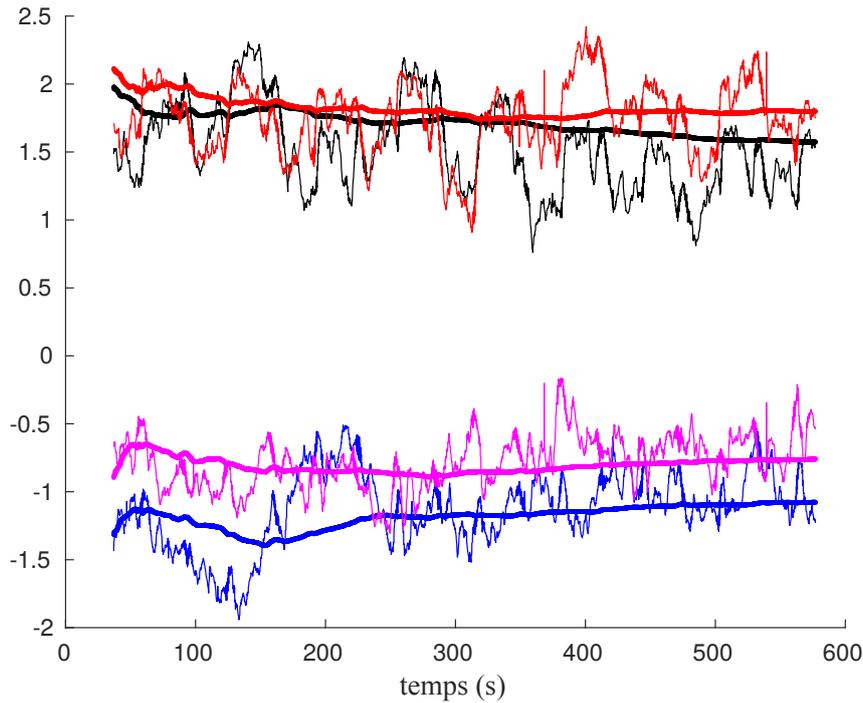


FIGURE 6.8. – *Erreurs GNSS biaisées avec estimation des biais (en gras) pour le véhicule suiveur selon l'Est (en bleu) et le Nord (en noir) ainsi que le Véhicule leader selon l'Est (en magenta) et le Nord (en rouge).*

que les intervalles sont pessimistes. Les 5 864 échantillons des données sont en effet tous compris dans ces intervalles de confiance à 95%.

La table 6.2 montre l'erreur de position et d'orientation ainsi que le taux de couverture lorsque la détection des marquages n'est pas disponible pour les deux véhicules. Les biais ne peuvent pas être estimés dans ce cas. Les erreurs de position $\|\bar{\epsilon}_p\|$ sont considérablement réduites lorsque les biais peuvent être estimés et les estimations deviennent consistantes, avec des taux de couverture C relativement pessimistes. Ces taux de couverture sont similaires à ceux des erreurs GNSS lorsque les biais sont compensés.

TABLE 6.2. – *Erreur de position $\|\bar{\epsilon}_p\|$ et d'orientation $|\bar{\epsilon}_\theta|$ et taux de couverture C de la localisation coopérative avec et sans estimation des biais et sans mesure LiDAR.*

	Sans estimation des biais				Avec estimation des biais			
	CLD de f		CLD de l		CLD de f		CLD de l	
	q_f	q_l	q_l	q_f	q_f	q_l	q_l	q_f
$\ \bar{\epsilon}_p\ $ (m)	2,11	1,99	1,99	2,11	0,22	0,24	0,23	0,22
$ \bar{\epsilon}_\theta $ (°)	1,67	1,36	1,37	1,67	2,20	3,37	3,66	2,10
C (%)	37,1	77,3	75,4	37,8	98,4	99,7	99,8	98,8

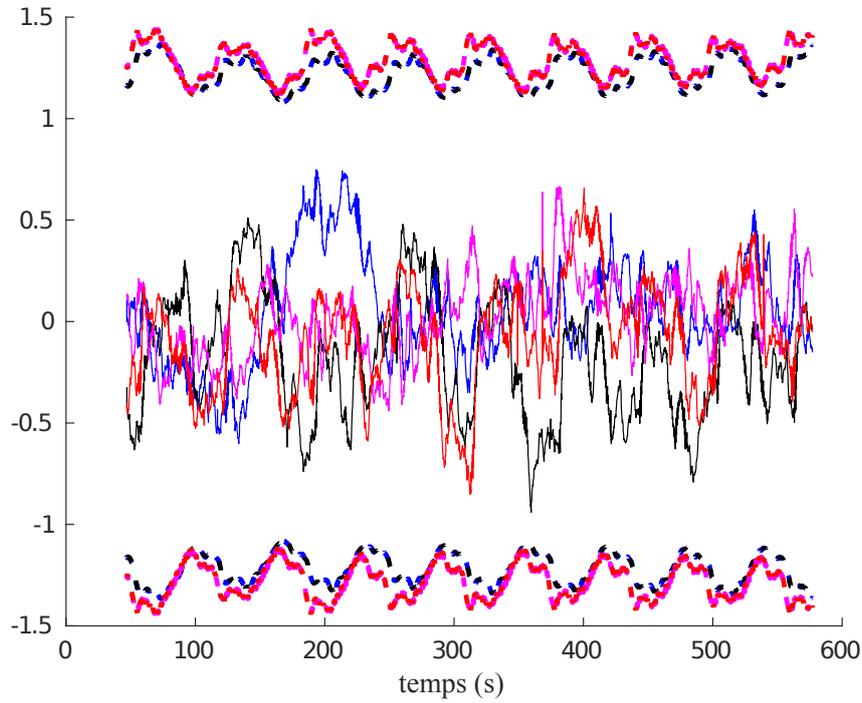


FIGURE 6.9. – Erreurs GNSS avec correction des biais (en trait plein). Les intervalles de confiance à 95% (en pointillés) ont été calculés à partir des incertitudes fournies par les récepteurs Ublox, des incertitudes des SPAN-CPT et des écarts types estimés du biais. L'Est est en bleu et le Nord en noir pour le véhicule suiveur et en magenta et en rouge pour le Véhicule leader.

TABLE 6.3. – Erreur de position $\|\bar{\epsilon}_p\|$ et d'orientation $|\bar{\epsilon}_\theta|$ et taux de couverture C de la localisation coopérative avec un CIF et avec ou sans mesure LiDAR.

	Sans LiDAR						Avec LiDAR					
	CLD de f			CLD de l			CLD de f			CLD de l		
	q_f	q_l	Jq_l	q_l	q_f	lq_f	q_f	q_l	Jq_l	q_l	q_f	lq_f
$\ \bar{\epsilon}_p\ $ (m)	0,21	0,23	0,54	0,23	0,21	0,66	0,20	0,26	0,15	0,25	0,20	0,42
$ \bar{\epsilon}_\theta $ (°)	2,04	2,65	3,45	3,47	1,74	4,02	1,23	2,38	1,83	2,47	1,15	2,17
C (%)	99,4	99,7	100	99,7	99,3	100	96,8	95,5	99,4	94,9	97,2	98,4

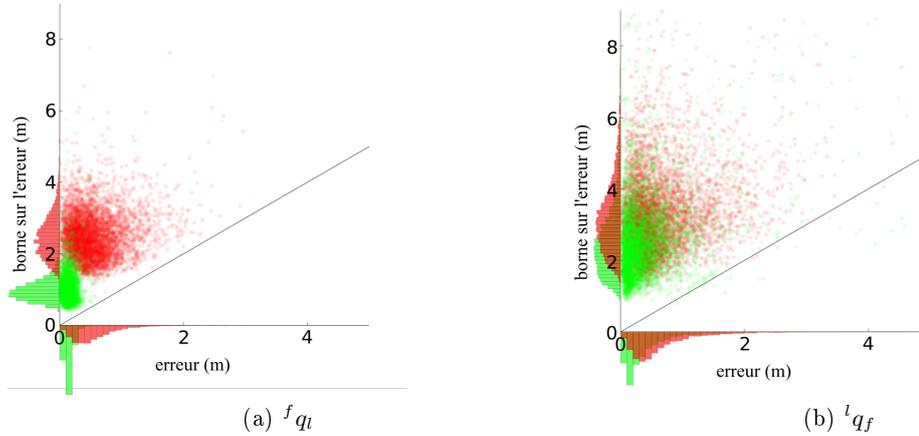


FIGURE 6.10. – Diagrammes de Stanford simplifiés des erreurs de position relative pour les poses relatives f_{q_l} et l_{q_f} en utilisant un CIF sans (en rouge) et avec (en vert) mesures LiDAR.

6.4.4. Précision et consistance de la localisation coopérative

Sans observation LiDAR, les poses des deux véhicules peuvent être considérées comme non corrélées. Même si l'état ne semble pas s'améliorer lorsque des mesures LiDAR sont utilisées (Fig. 6.3), la pose relative calculée à partir de l'état devient plus précise. Les erreurs de positions relatives passent ainsi de 0,54 m et de 0,66 m à 0,15 m et 0,43 m et les erreurs d'orientation relative de 3,45° et 4,02° à 1,83° et 2,17°. Ces résultats sont illustrés par les diagrammes de Stanford simplifiés de la figure 6.10 où les bornes de confiance en fonction des erreurs de positions relatives sont affichées. Les bornes de confiance sont calculées pour une confiance de 95% en utilisant une distribution de Rayleigh afin d'approximer la norme dans le calcul de l'erreur de position. L'utilisation des mesures LiDAR (en vert) améliore nettement la précision tout en restant consistant, plus de 95% des points restent en effet au-dessus de la diagonale noire. Le nuage vert est plus compact et reste proche de l'axe vertical et les bornes de confiance sont plus petites. On voit que la pose relative l_{q_f} calculée à partir de l'état du Véhicule leader est améliorée, même si ce véhicule n'est pas équipé d'un LiDAR et donc incapable de se localiser relativement par rapport à un autre véhicule, par lui-même. La communication améliore les états des CLD de tous les agents si chaque agent est lié aux autres par au moins une pose relative. De ce fait, une perception bidirectionnelle n'est pas essentielle pour cette méthode de fusion de données décentralisée. On remarque toutefois que le véhicule équipé du capteur de perception (le véhicule suiveur dans notre cas) peut calculer une pose relative f_{q_l} beaucoup plus précise (Fig. 6.10).

Dans ces différents scénarios, les erreurs restent consistantes. Une bonne estimation de l'incertitude est très importante pour la localisation coopérative, pour la fiabilité,

TABLE 6.4. – Erreur de position $\|\bar{\epsilon}_p\|$ et d'orientation $|\bar{\epsilon}_\theta|$ et taux de couverture C lorsque les bords de voie ne sont pas disponibles pour le véhicule suiveur ou le Véhicule leader.

	Bords de voie non disponibles pour le véhicule suiveur						Bords de voie non disponibles pour le véhicule tête					
	CLD de f			CLD de l			CLD de f			CLD de l		
	q_f	q_l	${}^f q_l$	q_l	q_f	${}^l q_f$	q_f	q_l	${}^f q_l$	q_l	q_f	${}^l q_f$
$\ \epsilon_p\ $ (m)	0,23	0,28	0,15	0,26	0,24	0,43	0,20	0,39	0,15	0,39	0,21	0,32
$ \epsilon_\theta $ (°)	1,35	2,47	1,82	2,55	1,26	2,54	1,54	2,58	1,79	2,26	1,47	1,66
C (%)	90,4	93,7	99,3	93,2	90,6	97,9	96,4	94,0	99,5	93,5	96,7	98,5

mais aussi pour obtenir des estimées précises des poses. En effet, les biais donnent des degrés de liberté au filtre. Si l'incertitude est mal estimée, les biais peuvent avoir des difficultés pour converger convenablement. La mise à jour avec les messages de la communication a aussi été testée avec un EKF à la place du CIF. Nous avons cependant remarqué que les problèmes de consanguinité de données augmentaient la rigidité des biais qui ne pouvaient alors pas correctement converger. Les résultats sont dans ce cas imprécis et inconsistants.

6.4.5. Performance de la méthode avec des contraintes de perception

Trois nouveaux scénarios sont présentés pour tester la robustesse de cette localisation coopérative lorsque les mesures de perception ne sont pas disponibles. Les deux premiers scénarios décrivent les cas où le véhicule suiveur ou le Véhicule leader est dans l'incapacité de détecter les bords de voie. Ces détection de bords de voie étant essentielles pour estimer le biais GNSS, il est intéressant d'étudier le cas où un des véhicules n'est pas capable de caractériser son biais GNSS par lui même. Le troisième scénario simule un champ de vue serré pour le LiDAR. Dans ce dernier scénario, la pose relative calculée à partir des points LiDAR n'est pas disponible dans les virages. Le LiDAR que nous avons utilisé a un angle d'ouverture de 110° , ce qui rendait la détection en virage difficile. Dans notre cas les véhicules roulaient proches l'un de l'autre, le véhicule leader restait à l'extérieur des virages et le véhicule suiveur restait à l'intérieur. Cette conduite permettait de garder le véhicule leader dans le champs de vue du LiDAR du véhicule suiveur. Il est donc intéressant d'étudier le cas où il ne reste alors que les détections les moins précises (les détections en « C ») disponible dans les lignes droites.

On observe dans la table 6.4 que même si un véhicule ne peut pas détecter les bords de voies et n'est donc pas capable d'estimer son biais GNSS par lui même, ce biais peut être estimé indirectement à partir de la pose du second véhicule et de leur pose relative. Le taux de couverture fourni lorsque les bords de voie ne sont pas disponibles pour le véhicule suiveur est moins consistant que celui fourni lorsque le Véhicule leader n'a pas accès aux bords de voie. En effet, la transformation du repère du Véhicule leader vers celui du véhicule suiveur présente plus de problèmes de non-linéarité que la transformation inverse. Les poses relatives restent cependant consistantes dans ces deux scénarios.

On voit dans la table 6.5 que lorsque le LiDAR est capable de détecter les autres

TABLE 6.5. – Erreur de position $\|\bar{\epsilon}_p\|$ et d'orientation $|\bar{\epsilon}_\theta|$ et taux de couverture C lorsque le champ de vue du LiDAR est serré.

	Champ de vue serré					
	CLD de f			CLD de l		
	q_f	q_l	${}^f q_l$	q_l	q_f	${}^l q_f$
$\ \epsilon_p\ $ (m)	0,21	0,21	0,36	0,21	0,21	0,55
$ \epsilon_\theta $ (°)	1,57	2,40	2,85	2,85	1,40	3,21
C (%)	99,6	99,7	99,9	99,6	99,7	99,5

véhicules seulement en ligne droite, les poses relatives sont moins précises. En effet, l'algorithme de perception donne de meilleurs résultats en virage, lorsque l'arrière et un des côtés du véhicule sont dans le champ de vue du LiDAR (détection en « L »). En ligne droite, seul l'arrière est visible (détection en « C ») et l'orientation relative est difficile à estimer. Les virages donnent alors des poses relatives plus précises.

6.5. Gestion spatio-temporelle des CLD

La carte locale dynamique contient les états de l'ensemble des véhicules communicants proches, chacun avec un identifiant unique. Il est donc nécessaire de définir comment ces véhicules sont déterminés et comment ajouter ou retirer des véhicules qui se rapprocheraient ou s'éloigneraient du égo-véhicule.

Une méthode simple est de calculer la distance euclidienne entre la position du égo-véhicule et les autres. La figure 6.11 illustre cette méthode. Tous les véhicules se trouvant dans un cercle autour du égo-véhicule sont alors intégrés dans la CLD de celui-ci. Le rayon de ce cercle peut dépendre de la vitesse du véhicule, voir de la vitesse relative entre les véhicules. Un véhicule éloigné se déplaçant rapidement peut rentrer en collision avec l'égo-véhicule plus rapidement qu'un véhicule proche se déplaçant lentement. Il est ainsi plus intéressant de raisonner en temps avant rencontre plutôt que simplement en distance. Ce temps peut être calculé à partir des informations des CLD :

$$\Delta T = \frac{\|p_i - p_e\|}{\|v_i \cos(\theta_i - \alpha) - v_e \cos(\theta_e - \alpha)\|}, \quad (6.53)$$

avec $\alpha = \arctan2(y_i - y_e, x_i - x_e)$. Ce temps est déterminé en divisant la distance entre les deux véhicules par la différence de vitesse projetée selon la direction du véhicule i par rapport au égo-véhicule e .

Cette méthode suppose cependant que le véhicule ne tournera pas. L'utilisation des cartes avec des coordonnées curvilignes peut alors permettre de prédire beaucoup plus précisément ces rencontres.

Si l'on suppose que les trajectoires des véhicules sont elles aussi communiquées, on peut déterminer les portions de voie de la carte commune à la trajectoire du égo-véhicule et à celle de chaque véhicule. Ces trajectoires peuvent être élargies à toutes les voies de la route pour prendre en compte les véhicules ne se situant pas sur la trajectoire du égo-véhicule, mais qui peuvent la croiser. La figure 6.12 illustre cette méthode. L'ensemble des véhicules (en bleu) rencontrant ou croisant la trajectoire du égo-véhicule (en vert) sont ajoutés à la CLD de celui-ci. Le temps avant rencontre

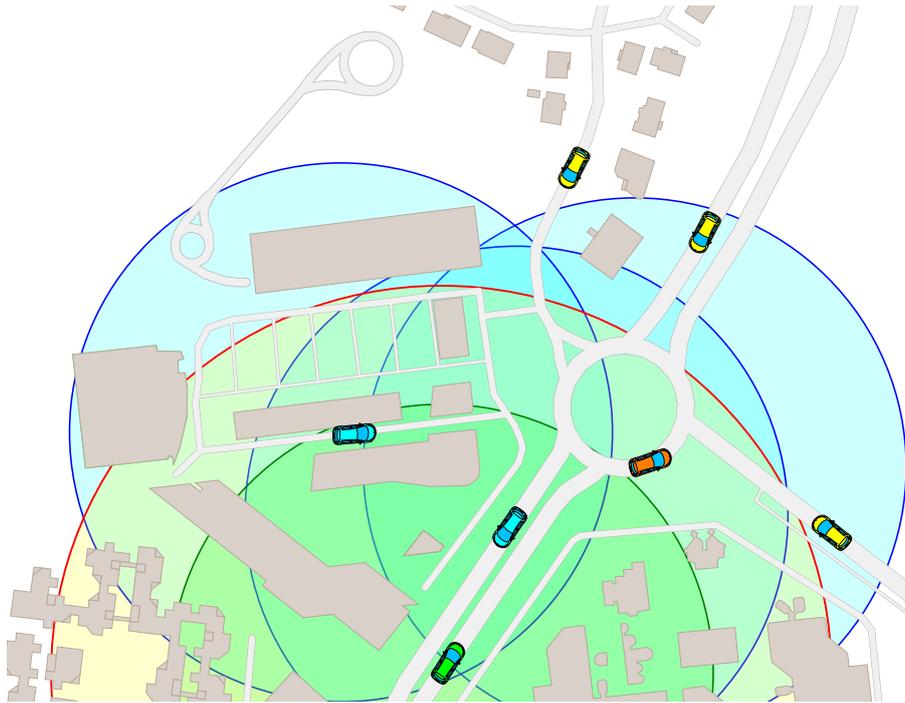


FIGURE 6.11. – *Carte Locale Dynamique avec distances euclidiennes. La CLD du égo-véhicule (en vert) contient l'ensemble des véhicules bleus contenus dans le cercle vert. La communication 802.11p permet toutefois de communiquer avec des véhicules plus éloignés (en orange). La frontière de la communication est représentée en rouge. L'égo-véhicule reçoit les CLD (les cercles bleus) des véhicules se situant à l'intérieur de cette frontière. Les véhicules trop éloignés pour pouvoir communiquer avec l'égo-véhicule (en jaune) peuvent être connus de celui-ci par l'intermédiaire des CLD d'autres véhicules.*



FIGURE 6.12. – *Carte Locale Dynamique basée sur des distances curvilignes en utilisant les trajectoires des véhicules (de la même couleur que les véhicules). L'égo-véhicule (en vert) peut communiquer avec l'ensemble des véhicules du cercle rouge. Il peut néanmoins ajouter dans sa CLD (véhicules bleus) des véhicules qui ne communiquent pas avec lui directement, mais qui sont présents dans les CLD de véhicules avec lesquels il peut communiquer. La trajectoire du véhicule orange ne croisant pas celle du égo-véhicule, il n'a pas besoin d'être dans la CLD.*

peut être calculé en divisant la distance curviligne ou la distance curviligne virtuelle par la vitesse relative. On suppose ici que le véhicule conserve sa vitesse jusqu'à la rencontre. Ces vitesses peuvent alors être régulées afin d'éviter une collision en réalisant une conduite en convoi réel ou bien virtuel. Le choix des véhicules devant être dans la CLD du égo-véhicule peut être fait en utilisant deux critères : le temps et la distance. En effet, le temps avant rencontre d'un véhicule se trouvant devant et allant à la même vitesse que l'égo-véhicule peut être infini voir négatif si les véhicules s'éloignent l'un de l'autre. Cependant si ce véhicule est très proche, il reste un danger potentiel. Il faut donc l'avoir dans la CLD du égo-véhicule. Deux critères sont donc définis en comparant le temps avant rencontre et la distance curviligne à deux seuils.

On peut voir ici que grâce à la communication, il est possible de connaître l'emplacement, la direction et la vitesse des véhicules qui s'approchent du égo-véhicule sans pour autant être visibles. L'utilisation des CLD par rapport à la communication de l'état de son véhicule uniquement, permet de détecter des véhicules plus loin que la portée de sa communication. Elle apporte de plus une redondance au niveau de la communication. Il est en effet possible de connaître l'état d'un véhicule par l'intermédiaire des CLD de tous les véhicules proches de lui. Il est alors plus facile d'obtenir les états des véhicules voisins lorsque des obstacles empêchent une communication directe. Un véhicule se trouvant par exemple dans un canyon urbain a ainsi plus de chance de recevoir des informations de l'extérieur. La méthode curviligne présente toutefois le désavantage d'ignorer les véhicules qui ne croise pas la trajectoire du égo-véhicule. Ces véhicules peuvent toutefois croiser la trajectoire d'un troisième véhicule qui ne peut pas communiquer directement avec eux. En ajoutant des véhicules qui ne sont pas directement utiles à l'égo-véhicule, on peut aussi aider les autres véhicules.

6.6. Conclusion

Ce chapitre a présenté une architecture et un système informatique embarqué communicant (un système cyberphysique) pour la localisation coopérative par échange de cartes locales dynamiques (CLD) entre véhicules voisins. Cette méthode est basée sur des mécanismes de fusion de type Kalman étendu et de type intersection de covariance pour gérer la corrélation entre les CLD des différents véhicules. L'état de chaque véhicule est inclus dans la CLD de chaque véhicule ce qui est essentiel à sa navigation. Dans cet algorithme, de multiples sources d'information sont fusionnées : les données cinématiques provenant du bus CAN, la pose GNSS, les poses relatives entre les véhicules calculées à partir des points LiDAR, les bords de voie pouvant être détectés par une caméra et les CLD des autres véhicules reçues par la communication. L'importance de l'estimation du biais GNSS a été montrée grâce à une détection de bords de voies. Le cas où les bords ne sont plus détectés par l'un des véhicules et où celui-ci doit estimer le biais grâce à la coopération a ensuite été étudié. Des résultats expérimentaux ont montré que l'utilisation d'un LiDAR pour lier les poses des différents véhicules améliore les poses relatives calculées à partir des états des CLD. Ainsi, la navigation en convoi est plus précise et sûre. Nous avons aussi montré dans ce chapitre que même si la détection de voie n'était pas disponible pour un des véhicules, la coopération donne la capacité à ce véhicule d'estimer son biais GNSS et ainsi d'améliorer sa localisation.

Malgré le fait de n'avoir utilisé que deux véhicules dans ce chapitre, nous avons

6. Localisation coopérative décentralisée et asynchrone pour les véhicules communicants

pu montrer que notre méthode permettait de faire intervenir une contrainte relative entre deux véhicules. Ainsi dans le cas d'un déploiement à grande échelle, chaque observation permettrait l'ajout de cette contrainte relative entre le véhicule qui perçoit et le véhicule perçu. Dans le cas où tous les véhicules d'une flotte percevraient ou seraient perçus, ces contraintes pourraient permettre de propager une pose précise au travers de la flotte. Malgré cela, l'imprécision de la perception pourrait réduire la portée de cette propagation comme nous avons pu l'observer dans le chapitre 5.

7. Conclusion et perspectives

Dans cette thèse, nous avons commencé par définir les coordonnées curvilignes permettant de localiser un véhicule par rapport aux voies de navigation. Dans les formats actuels, ces voies sont décrites dans une carte haute définition sous la forme de polygones, c'est-à-dire une suite de points reliés par des segments de droite. Nous avons donc défini les coordonnées curvilignes relatives à cette représentation. Nous avons comparé un calcul classique par projection orthogonale directement sur les polygones à des coordonnées curvilignes calculées à partir de splines et à base de projection de type *lanelet*. Nous avons ainsi observé que cette dernière méthode permet d'éviter les problèmes de discontinuité tout en restant suffisamment rapide pour être temps réel contrairement à celle utilisant des splines.

Dans cette approche, nous supposons que le modèle de la voie est parfaitement connu. Les cartes HD construites à grande échelle par les cartographes ont actuellement des précisions absolues de l'ordre du mètre mais sont très précises en relatif. Afin de corriger le biais qui persiste, une possibilité pourrait être de l'estimer en le rajoutant à la carte locale dynamique et en le partageant aux autres véhicules pour une estimation coopérative. Comme ce biais change très lentement dans le temps (à cause des mouvements des plaques tectoniques de la terre, par exemple), une solution serait que chaque véhicule autonome soit capable de l'estimer et de le partager sur un cloud pour qu'il puisse être estimé avec une méthode de fusion de données centralisée et robuste aux mauvaises estimations de certains véhicules. Cette méthode pourrait aussi permettre de corriger localement des erreurs de la carte. Une seconde source d'erreur (plus locale) provient de l'approximation faite lors de la discrétisation. Cette erreur pourrait être estimée à partir de la flèche (l'erreur maximale entre le modèle continu et le modèle discret utilisé). Ne connaissant pas ce modèle continu, il pourrait être approximé par la spline d'Hermite.

Les coordonnées curvilignes ont ensuite été utilisées dans une approche simplifiée du problème de localisation coopérative. L'écart latéral et l'orientation par rapport à la voie ainsi que la pose relative entre les deux véhicules présents dans le scénario étaient supposées parfaitement connues. Cette première approche nous a permis de montrer, dans ce cas idéal, l'intérêt de la localisation coopérative sur l'incertitude longitudinale et notamment l'influence de la courbure de la route. Afin de résoudre les problèmes de consanguinité de données, une intersection de covariance a été utilisée. Dans un espace à une dimension, la fusion par intersection de covariance consiste simplement à garder l'estimateur de plus faible variance. Cette méthode permet à la pose estimée de rester consistante mais elle peut toutefois devenir pessimiste. Même s'il est préférable, pour être plus fiable, de sous-estimer la précision que de la surestimer, une mauvaise estimation de la matrice de covariance de l'état risque de détériorer la fusion. Une intersection de covariance partitionnée pourrait permettre de réduire ce problème.

Nous avons ensuite étudié des méthodes de perception basées sur une mise en correspondance d'un scan LiDAR avec un modèle géométrique polygonal de véhi-

7. Conclusion et perspectives

cule. Cela nous a permis d'obtenir la pose relative entre deux véhicules avec son incertitude modélisée sous la forme d'une matrice de covariance. Plusieurs méthodes de mise en correspondance et de minimisation ont été comparées. Nous avons tout d'abord observé en simulation que la minimisation par pseudo-inverse donnait les meilleurs résultats en termes de consistance. En ce qui concerne la mise en correspondance, les résultats en simulation, ont montré une convergence plus rapide des points LiDAR sur le modèle polygonal lorsque la correspondance point à ligne était utilisée. De plus, la géométrie du véhicule perçu étant mieux représentée avec cette correspondance, les matrices de covariance estimées en utilisant ce modèle étaient plus consistantes que celles estimées à partir des autres correspondances. Des tests de robustification ont ensuite été ajoutés afin de rejeter des mesures fausses dues à de mauvaises détections LiDAR ou à de mauvaises convergences de l'algorithme. Enfin, nous avons étudié l'influence du scénario sur les performances de la localisation relative. Nous avons ainsi montré une précision plus importante obtenue à partir d'une détection en « L » où l'arrière et un des côtés du véhicule étaient dans le champ de vue du LiDAR par rapport à une détection en « C » où seul l'arrière était détecté. La correspondance point à ligne a permis d'observer les problèmes d'invariance par rotation liés à une détection en « C » et de conserver ainsi une consistance similaire à celle d'une détection en « L ».

Dans le cas réel, notre perception 2D peut réduire le nombre de détections disponibles, lorsque la hauteur des pare-chocs est trop haute (avec des camions ou des véhicules tout terrain par exemple) ou lorsque la pente de la route varie (lors de dos d'âne par exemple). La projection d'un scan LiDAR 3D sur le plan horizontal du LiDAR est une première solution. L'utilisation d'un modèle 3D polyédrique pour remplacer le modèle 2D polygonal pourrait aussi être utilisé. Les correspondances points à lignes seraient alors remplacées par des correspondances points à plans (les segments étant remplacés par des facettes triangulaires). Des détections 3D pourraient permettre de réduire les problèmes d'invariance rendant le problème mal conditionné lorsque des formes en « C » sont détectées. Les modèles 2D et 3D pourraient être communiqués ou être disponibles par requête sur un serveur de type « cloud ». Cependant, le modèle pourrait aussi être reconstruit, pour se rapprocher des messages CPM en utilisant seulement la pose et les dimensions de la boîte englobante. Une reconstruction des modèles des agents permettrait de plus de prendre en compte les agents (fixes ou dynamiques) non communicants.

Les problèmes de propagation d'incertitude lorsque la pose d'un véhicule est déterminée à partir de la pose d'un second véhicule et de la pose relative entre ces deux véhicules ont ensuite été étudiés. On a ainsi observé des distributions en banane à la place de gaussiennes provenant des non-linéarités des transformations. Pour minimiser ces effets, on a pu montrer qu'une orientation relative précise était nécessaire. Les erreurs d'orientations absolue et relative sont, en effet, à l'origine de ces non-linéarités. Afin de prendre en compte l'incertitude d'une pose absolue propagée à partir d'une pose relative et d'une pose absolue, il est essentiel de propager l'incertitude de la pose absolue en prenant en compte l'incertitude de la perception. On a observé dans nos résultats qu'il était préférable, d'un point de vue de la précision et de la consistance, d'être localisé par un véhicule qui nous perçoit plutôt que de se localiser à partir de notre perception d'un véhicule communicant sa pose. Une perception mutuelle pourrait ainsi grandement améliorer la localisation d'un

7. Conclusion et perspectives

convoi. Afin de réduire les problèmes de non-linéarité lors de la propagation, des transformations non parfumées pourraient aussi être utilisées. Nous avons observé la propagation de l'incertitude dans un convoi de dix véhicules en simulation. On pourrait toutefois étudier le cas plus complexe où les véhicules ne seraient plus « chaînés » et où plusieurs « chemins » seraient possibles pour réaliser la propagation d'une pose. Toutefois, lors de l'utilisation de la fusion de données suivante, la pose relative a été considérée comme une observation permettant de mettre à jour les poses du véhicule observateur et du véhicule perçu. La propagation se fait donc d'elle-même de proche en proche lorsqu'une perception « lie » deux poses entre elles. Lorsque plusieurs chemins de propagations seraient possibles, dans le cas où les véhicules ne seraient pas chaînés, cette méthode pourrait alors permettre d'éviter des problèmes de corrélation entre ces différentes propagations.

Le problème de la localisation coopérative a enfin été formalisé par un modèle décentralisé et asynchrone utilisant des cartes locales dynamiques. Plusieurs sources d'information ont été fusionnées : les données cinématiques du véhicule, la pose GNSS, la pose relative déterminée à partir du LiDAR et les détections de bords de voies simulées à partir de la vérité terrain. Cette fusion de données a permis de montrer l'importance de l'estimation du biais GNSS et comment il peut être déterminé à partir de la coopération. Les poses relatives précises obtenues grâce aux mesures LiDAR sont aussi très importantes pour le véhicule équipé du LiDAR, mais aussi pour celui qui est détecté par celui-ci.

L'état des véhicules a été représenté par une carte locale dynamique intégrant les poses et vitesses des véhicules, des données qui sont utiles pour la décision et le contrôle. Ainsi, la carte locale dynamique joue le rôle d'une couche d'abstraction fournissant les informations haut niveau pour la navigation. Cette approche asynchrone et décentralisée permet de prendre en compte la problématique d'une implémentation dans un vrai véhicule avec des capteurs réalisant des mesures asynchrones. L'implémentation qui a été réalisée dans ces travaux est toutefois non temps réel, mais les algorithmes sont suffisamment rapides pour l'être. En effet, l'utilisation de ROS et de Python ne permet pas d'obtenir une implémentation « temps réel dur » en pratique.

En convertissant les poses de chaque agent de la carte locale dynamique (CLD) en coordonnées curvilignes, on pourrait ainsi avoir accès aux erreurs latérales et d'orientation pour le contrôle latéral et aux interdistances curvilignes avec les autres véhicules ou aux distances curvilignes jusqu'aux prochaines intersections ou aux passages piétons pour le contrôle longitudinal. Grâce à ces informations en coordonnées curvilignes, la CLD d'un véhicule pourrait être utilisée pour sa navigation et servir de couche d'abstraction.

Dans le scénario de conduite en convoi étudié, seuls deux véhicules roulant avec une interdistance faible de dix à vingt mètres étaient présents. Des scénarios plus complexes pourraient être étudiés, notamment le cas de véhicules éloignés à la limite de la détection LiDAR, ou celui des véhicules cachés par des obstacles par moment ou cachés par des véhicules non communicants possédant la même forme géométrique. Les problèmes de correspondance entre les véhicules et les groupes de points LiDAR pourraient ainsi être observés. Dans le cas où de fausses informations sont communiquées, la robustesse de la fusion décentralisée pourraient aussi être étudiée. Des scénarios d'intersections, d'insertions ou de croisements de face avec deux véhicules

7. Conclusion et perspectives

ou plus permettraient d’observer les capacités de la localisation coopérative dans des scénarios plus variés.

La localisation coopérative pourrait aussi être réalisée lorsque les véhicules ne sont plus capables de se percevoir, mais perçoivent le même objet statique (comme un bâtiment, un bord de voie ou un panneau) ou dynamique (comme un véhicule) présent dans leurs cartes locales dynamiques. Ils seraient alors capables de s’aider de façon indirecte. Les éléments statiques pourraient provenir d’une carte imprécise dont l’incertitude est inconnue (comme une carte Open Street Map). Cette dernière ne servira alors que pour fournir les poses initiales des objets. Ces poses seraient corrigées en étant présentes dans la carte locale dynamique des véhicules.

Lors de la fusion de données, la détection de bord de voie a été simulée. Elle pourrait être remplacée par une vraie détection afin d’étudier les fausses détections et les zones où ces observations ne sont pas observables. Les poses GNSS calculées par le récepteur GNSS ont déjà des erreurs corrélées temporellement et ont été prises en compte dans la fusion de données comme si elles avaient des erreurs décorrélées. Une fusion en couplage serré en prenant en compte directement les pseudo-distances et les effets Doppler permettrait d’éviter ces corrélations temporelles. Cette méthode permettrait de plus de communiquer les biais liés à chaque satellite qui pourraient être fusionnés avec ceux estimés par les autres véhicules puisqu’ils seraient communs.

La fusion de données a été réalisée en prenant comme état une CLD dans un repère absolu. Cette méthode a permis de simplifier les prédictions des poses des véhicules. En pratique, les poses des véhicules devraient être communiquées sous la forme de coordonnées globales GNSS, les repères de travail ENU de deux véhicules différents pouvant en effet être différents. Notre méthode pourrait donc être adaptée au cas de la fusion de CLDs fournies dans les repères relatifs des véhicules communicants. D’autres approches pourraient être étudiées comme l’approche où chaque observation est communiquée afin d’éviter les problèmes de consanguinité de données.

Cette thèse a permis de mettre en évidence l’intérêt des cartes locales dynamiques pour la navigation de véhicules autonomes notamment pour rendre plus robuste la localisation du véhicule et la perception des véhicules qui l’entourent. La communication de CLD permet également aux véhicules d’étendre leur champs de visibilité qui est limité par les capacités des capteurs de perception embarqués. Si la communication et la perception ne sont pas disponibles simultanément, la vitesse de chaque véhicule contenue dans la CLD permet de prédire le déplacement des véhicules pendant une courte période de temps. Ainsi dans le cas d’un déploiement à grande échelle, chaque véhicule serait capable de connaître l’emplacement de tous les usagers et de naviguer avec eux en toute sécurité lors de conduite en convoi ou de franchissement d’intersection par exemple.

Contributions

La première contribution de ces travaux est la définition de coordonnées curvilignes basées sur des polygones avec la comparaison de différentes méthodes [Héry *et al.*, 2017a].

Une première formulation de localisation coopérative à une dimension en abscisse curviligne permet de montrer l’intérêt de la coopération. Ce problème simplifié illustre l’effet de la courbure de la route sur la réduction des incertitudes longitudinales des estimations des différents véhicules en coopération [Héry *et al.*, 2017b,

7. Conclusion et perspectives

Héry *et al.*, 2017c].

La troisième contribution est l'étude et l'implémentation d'une localisation relative par LiDAR s'appuyant sur la réception d'une pose estimée et du modèle géométrique du véhicule perçu. Différentes méthodes de correspondance entre les points LiDAR et ce modèle sont comparées [Héry *et al.*, 2018]. Les problèmes de non-linéarités lors de la propagation d'estimations de poses entre véhicules avec des localisations relatives incertaines sont ensuite étudiés [Héry *et al.*, 2019b].

Finalement, un algorithme de localisation coopérative basé sur le partage de cartes locales dynamiques est présenté. Cette méthode permet alors de fusionner l'ensemble des données disponibles, à savoir les mesures des capteurs et les messages provenant de la communication. Elle fournit en sortie une couche d'abstraction pouvant être utilisée pour la navigation autonome [Héry *et al.*, 2019a].

A. Notations

On note $X = (qbu)^T$ l'état d'un véhicule contenant la pose q , le biais GNSS b ainsi que le vecteur cinématique u . La pose $q = (xy\theta)^T$ représente la position $p = (xy)^T$ et l'orientation θ d'un véhicule. $b = (b_x b_y)^T$ est le biais GNSS exprimé sur l'axe des abscisses et celui des ordonnées. Finalement, $u = (v\omega)^T$ est le vecteur cinématique représentant la vitesse longitudinale et la vitesse de rotation du véhicule. Il est important d'exprimer chacune de ces grandeurs dans un repère donné. Dans le cas où plusieurs véhicules sont présents, il est aussi utile de pouvoir différencier les données des différents véhicules. Nous utilisons des indices et exposants pour résoudre ces problèmes.

La table A.1 présente les différents éléments de l'état d'un véhicule.

La table A.2 présente les notations des indices et exposants utilisés pour une donnée z quelconque. L'indice i en bas à droite représente l'agent, le véhicule, lié à la donnée z . L'exposant en haut à gauche, indique le repère dans lequel la donnée z est exprimée. Comme nous utilisons des repères orthonormés, chaque repère est complètement défini par sa pose, qui donne la position (2D ou 3D) de l'origine et l'orientation (2D ou 3D) du repère. Ainsi un indice i donné pourra faire référence à la pose ou au repère. Finalement, la source de l'information z est indiquée en bas à gauche, cette source peut être un capteur, ou un autre véhicule dans le cas de véhicules communicants. \hat{z} représente l'estimé de z , z^* sa vérité terrain et $\epsilon_z = z^* - \hat{z}$ l'erreur entre ces deux grandeurs. On exprime la matrice de covariance de z par Σ_z et la matrice d'intercorrélacion entre deux données z_1 et z_2 par Σ_{z_1, z_2} . Dans de nombreux cas il est important d'avoir la matrice de covariance d'une donnée z en plus de celle-ci. Afin d'alléger les notations nous notons $\mathbf{z} = \{z, \Sigma_z\}$ l'ensemble de ces deux informations.

En combinant les tables A.1 et A.2 on peut ainsi définir tout élément de l'état pour tout véhicule dans un repère donné et provenant d'une certaine source. Par exemple, la grandeur ${}_{LiDAR}^i q_j$ désigne la mesure de la pose du véhicule j dans le repère i calculée à partir d'un scan LiDAR. Dans le cas de véhicules communicants, la grandeur ${}_i X_j$ désigne l'état du véhicule j provenant du véhicule i . On utilise la notation

$${}_i X = \begin{pmatrix} {}_i X_1 \\ \vdots \\ {}_i X_N \end{pmatrix}, \quad (\text{A.1})$$

pour représenter les états de tous les véhicules estimés par un véhicule i .

La table A.3 décrit les indices et exposants couramment utilisés.¹

1. Quand une pose sera décrite dans le repère global, nous simplifierons les notations en omettant ce repère, ainsi ${}^0 q_e$ sera simplement écrite q_e .

A. Notations

TABLE A.1. – *Notations de l'état d'un véhicule.*

Symbole	Signification	Description
x		Abscisse.
y		Ordonnée.
θ		Orientation.
p	$\begin{pmatrix} x \\ y \end{pmatrix}$	Position.
q	$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$	Pose.
b	$\begin{pmatrix} b_x \\ b_y \end{pmatrix}$	Biais GNSS.
v		Vitesse longitudinale.
ω		Vitesse de rotation.
u	$\begin{pmatrix} v \\ \omega \end{pmatrix}$	Vecteur cinématique.
X	$\begin{pmatrix} q \\ b \\ u \end{pmatrix}$	État.

TABLE A.2. – *Notations des indices et exposants.*

Symbole	Signification	Description
${}^j_{src}z_i [t]$		Donnée z de l'agent i provenant de la source src exprimée dans le repère de j au temps t . src pourra être un capteur dans le cas où z est une observation ou un agent si les estimés de z de plusieurs agents sont comparés. j pourra être omis lorsque ce repère correspondra au repère monde 0.
\hat{z}		Estimé de z .
z^*		Vérité terrain de z .
$\epsilon_{\hat{z}}$	$z^* - \hat{z}$	Erreur de \hat{z} .
Σ_z		Matrice de covariance de z .
Σ_{z_1, z_2}		Matrice d'intercorrélacion entre z_1 et z_2 .
\mathbf{z}	$\{z, \Sigma_z\}$	Ensemble contenant la donnée z et sa matrice de covariance Σ_z .

TABLE A.3. – *Indices et exposants utilisés.*

Symbole	Description
e	Égo-véhicule.
c	Véhicule communicant ou coopératif.
t	Véhicule cible (target).
l	Véhicule leader.
f	Véhicule suiveur (follower).

B. Changements de repères et incertitudes associées

B.1. Introduction

Localiser un véhicule, c'est savoir trouver sa pose dans un repère donné. Comme nous utilisons uniquement des repères orthonormés directs, chaque pose permet de définir entièrement un repère dans un autre. En effet, la position de la pose définit l'origine du repère et l'orientation de ses axes.

Les opérateurs \oplus et \ominus introduits par Smith, Self et Cheeseman [Smith *et al.*, 1990] pour les changements de repères sont utilisés dans ces travaux. Ces notations permettent d'alléger les équations, en manipulant les changements de repères aussi simplement que la composition des vecteurs dans un repère cartésien, en respectant la relation de Chasles.

Dans cette annexe, on s'intéresse uniquement à des poses 2D caractérisées par des vecteurs de dimension trois, contenant une position cartésienne et un angle de cap. Ainsi, la pose ${}^k q_i$ décrivant la translation et la rotation à appliquer au repère R_k pour le transformer dans le repère R_i est :

$${}^k q_i = \begin{bmatrix} {}^k x_i \\ {}^k y_i \\ {}^k \theta_i \end{bmatrix}, \quad (\text{B.1})$$

${}^k q_i$ représente aussi le repère R_i dans le repère R_k , l'angle de cap de l'axe des abscisses de R_i étant ${}^k \theta_i$.

De plus, les transformations sont attachées d'incertitudes ; elles sont probabilistes avec des matrices de covariance. La propagation de ces incertitudes lors des transformations est expliquée en se basant sur des linéarisations à l'ordre un des différentes équations qui sont non linéaires à cause des rotations. Ces mécanismes de propagation d'erreur ne sont valables que pour des erreurs petites par rapport aux non-linéarités.

B.2. Composition et opérateur \oplus

L'opérateur de composition \oplus est défini de la manière suivante (cf Fig. B.1) :

$${}^k q_i = {}^k q_j \oplus {}^j q_i. \quad (\text{B.2})$$

On peut noter que les indices et exposants « j » s'enchaînent de façon simple.

Les notions de repères (orthonormés directes), de poses et de changements de repères contiennent les mêmes informations. En considérant les poses comme des fonctions de changement de repères on peut alors voir l'opérateur \oplus comme l'opérateur

B. Changements de repères et incertitudes associées

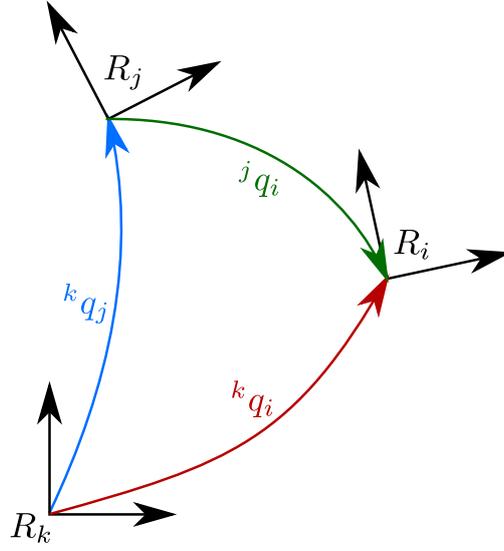


FIGURE B.1. – *Changement de repère. La flèche courbe ${}^k q_i$ représente la translation et la rotation à appliquer au repère R_k pour le transformer dans le repère R_i . La courbure de la flèche est du même signe que la rotation.*

o de composition de fonctions ainsi :

$${}^k q_i = {}^k q_j \oplus {}^j q_i, \quad (\text{B.3})$$

$$\begin{aligned} {}^k q_i () &= {}^k q_j ({}^j q_i ()) \\ &= {}^k q_j \circ {}^j q_i (). \end{aligned} \quad (\text{B.4})$$

L'opération géométrique résultante de l'opérateur \oplus représente une transformation de R_k en R_j (via la pose ${}^k q_j$) suivie d'une transformation de R_j en R_i (${}^j q_i$) dépendant de l'angle ${}^k \theta_j$. En d'autres termes, on combine la pose ${}^k q_j$ (en bleu sur la figure B.1) et la pose ${}^j q_i$ (en vert) pour obtenir la pose ${}^k q_i$ (en rouge). Une utilisation habituelle de cette composition est la transformation d'une pose exprimée dans le repère R_j (le terme de droite) en une autre pose exprimée dans R_k via le terme de gauche.

On a :

$$\begin{aligned} {}^k q_i &= {}^k q_j \oplus {}^j q_i \\ &= {}^k q_j + \begin{pmatrix} \cos({}^k \theta_j) & -\sin({}^k \theta_j) & 0 \\ \sin({}^k \theta_j) & \cos({}^k \theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^j q_i \\ &= \begin{pmatrix} {}^k x_j + {}^j x_i \cos({}^k \theta_j) - {}^j y_i \sin({}^k \theta_j) \\ {}^k y_j + {}^j x_i \sin({}^k \theta_j) + {}^j y_i \cos({}^k \theta_j) \\ {}^k \theta_j + {}^j \theta_i \end{pmatrix}. \end{aligned} \quad (\text{B.5})$$

On voit clairement que l'opération à appliquer sur le terme de droite dépend du terme de gauche.

B. Changements de repères et incertitudes associées

L'opérateur \oplus n'est pas commutatif mais est associatif comme la composition de fonction.

Afin de propager l'incertitude représentée par une matrice de covariance, une approximation au premier ordre peut être effectuée :

$$\Sigma_{kq_i} \approx J_{\oplus} \begin{pmatrix} \Sigma_{kq_j} & \Sigma_{kq_j,^j q_i} \\ \Sigma_{^j q_i, kq_j} & \Sigma_{^j q_i} \end{pmatrix} J_{\oplus}^T, \quad (\text{B.6})$$

Les sous-matrices d'inter-covariance $\Sigma_{kq_j,^j q_i}$ est la matrice symétrique de la sous-matrice $\Sigma_{^j q_i, kq_j}$ (ces deux sous-matrices sont souvent prises nulles).

La matrice Jacobienne J_{\oplus} est définie par :

$$\begin{aligned} J_{\oplus} &= \frac{\partial kq_i}{\partial (kq_j, ^j q_i)} = \frac{\partial (kq_j \oplus ^j q_i)}{\partial (kq_j, ^j q_i)} \\ &= \begin{pmatrix} 1 & 0 & -^j x_i \sin(^k \theta_j) & -^j y_i \cos(^k \theta_j) & \cos(^k \theta_j) & -\sin(^k \theta_j) & 0 \\ 0 & 1 & ^j x_i \cos(^k \theta_j) & -^j y_i \sin(^k \theta_j) & \sin(^k \theta_j) & \cos(^k \theta_j) & 0 \\ 0 & 0 & & 1 & 0 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (\text{B.7})$$

En calculant la pose $^k q_i$ avant la Jacobienne, on peut alors exprimer cette dernière directement à partir des nouvelles coordonnées sans recalculer certains sinus et cosinus. On obtient alors la formule originale de Smith, Self et Cheeseman [Smith *et al.*, 1990] :

$$J_{\oplus} = \begin{pmatrix} 1 & 0 & -(^k y_i - ^k y_j) & \cos(^k \theta_j) & -\sin(^k \theta_j) & 0 \\ 0 & 1 & (^k x_i - ^k x_j) & \sin(^k \theta_j) & \cos(^k \theta_j) & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{B.8})$$

B.3. Inversion et opérateur \ominus

On peut ensuite définir l'opérateur d'inversion de la pose $^j q_i$:

$$\begin{aligned} {}^i q_j &= \ominus^j q_i \\ &= \begin{pmatrix} \cos(-^j \theta_i) & -\sin(-^j \theta_i) & 0 \\ \sin(-^j \theta_i) & \cos(-^j \theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^{-j} q_i) \\ &= \begin{pmatrix} -^j x_i \cos(-^j \theta_i) + ^j y_i \sin(-^j \theta_i) \\ -^j x_i \sin(-^j \theta_i) - ^j y_i \cos(-^j \theta_i) \\ -^j \theta_i \end{pmatrix} \\ &= \begin{pmatrix} -^j x_i \cos(^j \theta_i) - ^j y_i \sin(^j \theta_i) \\ ^j x_i \sin(^j \theta_i) - ^j y_i \cos(^j \theta_i) \\ -^j \theta_i \end{pmatrix}. \end{aligned} \quad (\text{B.9})$$

En considérant les poses comme des fonctions de changement de repères, l'opérateur \ominus revient à inverser cette fonction :

$${}^i q_j = \ominus^j q_i, \quad (\text{B.10})$$

$${}^i q_j () = {}^j q_i^{-1} (). \quad (\text{B.11})$$

B. Changements de repères et incertitudes associées

De la même façon que pour l'opérateur \oplus , en appliquant une approximation au premier ordre, on peut propager la matrice de covariance :

$$\Sigma_{i q_j} \approx J_{\ominus} \Sigma_{j q_i} J_{\ominus}^T, \quad (\text{B.12})$$

avec

$$J_{\ominus} = \frac{\partial^i q_j}{\partial^j q_i} = \frac{\partial (\ominus^j q_i)}{\partial^j q_i} = \begin{pmatrix} -\cos(j\theta_i) & -\sin(j\theta_i) & {}^j x_i \sin(j\theta_i) - {}^j y_i \cos(j\theta_i) \\ \sin(j\theta_i) & -\cos(j\theta_i) & {}^j x_i \cos(j\theta_i) + {}^j y_i \sin(j\theta_i) \\ 0 & 0 & -1 \end{pmatrix}. \quad (\text{B.13})$$

Cette expression peut être simplifiée à partir des coordonnées de ${}^i q_j$:

$$J_{\ominus} = \begin{pmatrix} -\cos(j\theta_i) & -\sin(j\theta_i) & {}^i y_j \\ \sin(j\theta_i) & -\cos(j\theta_i) & -{}^i x_j \\ 0 & 0 & -1 \end{pmatrix}. \quad (\text{B.14})$$

On obtient par exemple :

$${}^j q_i = \ominus^i q_j = \ominus \ominus^j q_i. \quad (\text{B.15})$$

En propageant l'incertitude de la matrice de covariance $\Sigma_{j q_i}$ deux fois :

$$\Sigma_{j q_i} \approx J'_{\ominus} J_{\ominus} \Sigma_{j q_i} J_{\ominus}^T J'^T_{\ominus}, \quad (\text{B.16})$$

avec

$$\begin{aligned} J'_{\ominus} &= \begin{pmatrix} -\cos({}^i \theta_j) & -\sin({}^i \theta_j) & {}^j y_i \\ \sin({}^i \theta_j) & -\cos({}^i \theta_j) & -{}^j x_i \\ 0 & 0 & -1 \end{pmatrix} \\ &= \begin{pmatrix} -\cos(j\theta_i) & \sin(j\theta_i) & {}^j y_i \\ -\sin(j\theta_i) & -\cos(j\theta_i) & -{}^j x_i \\ 0 & 0 & -1 \end{pmatrix}. \end{aligned} \quad (\text{B.17})$$

En développant

$$J'_{\ominus} J_{\ominus} = \begin{pmatrix} c^2 + s^2 & cs - sc & -{}^i y_j c - {}^i x_j s - {}^j y_i \\ sc - cs & s^2 c^2 & -{}^i y_j s + {}^i x_j c + {}^j x_i \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{B.18})$$

avec $c = \cos(j\theta_i)$ et $s = \sin(j\theta_i)$.

À partir de l'équation (B.9) (pour l'équation ${}^j q_i = \ominus^i q_j$)

$$-{}^i y_j \cos(j\theta_i) - {}^i x_j \sin(j\theta_i) - {}^j y_i = -{}^i y_j \cos({}^i \theta_j) + {}^i x_j \sin({}^i \theta_j) - {}^j y_i = 0, \quad (\text{B.19})$$

$$-{}^i y_j \sin(j\theta_i) + {}^i x_j \cos(j\theta_i) + {}^j x_i = {}^i y_j \sin({}^i \theta_j) + {}^i x_j \cos({}^i \theta_j) + {}^j x_i = 0. \quad (\text{B.20})$$

On a donc

$$J'_{\ominus} J_{\ominus} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{B.21})$$

Comme

$$J_{\ominus}^T J_{\ominus}^T = (J'_{\ominus} J_{\ominus})^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{B.22})$$

On a bien

$$\Sigma_{j q_i} = J'_{\ominus} J_{\ominus} \Sigma_{j q_i} J_{\ominus}^T J'^T_{\ominus} = \Sigma_{j q_i}. \quad (\text{B.23})$$

B.4. Combinaison des opérateurs

On peut combiner les deux opérateurs pour, par exemple, déterminer ${}^k q_j$ en fonction de deux poses connues avec incertitude notées ${}^k q_i$ et de ${}^j q_i$ (en gardant les mêmes notations que précédemment).

On a :

$$\begin{aligned}
 {}^k q_j &= {}^k q_i \oplus^i q_j \\
 &= {}^k q_i + \begin{pmatrix} \cos({}^k \theta_i) & -\sin({}^k \theta_i) & 0 \\ \sin({}^k \theta_i) & \cos({}^k \theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^i q_j \\
 &= {}^k q_i \oplus (\ominus^j q_i) \\
 &= {}^k q_i + \begin{pmatrix} \cos({}^k \theta_i) & -\sin({}^k \theta_i) & 0 \\ \sin({}^k \theta_i) & \cos({}^k \theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(-{}^j \theta_i) & -\sin(-{}^j \theta_i) & 0 \\ \sin(-{}^j \theta_i) & \cos(-{}^j \theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^{-j} q_i) \\
 &= {}^k q_i + \begin{pmatrix} \cos({}^k \theta_i - {}^j \theta_i) & -\sin({}^k \theta_i - {}^j \theta_i) & 0 \\ \sin({}^k \theta_i - {}^j \theta_i) & \cos({}^k \theta_i - {}^j \theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^{-j} q_i) \\
 &= \begin{pmatrix} {}^k x_i + {}^{-j} x_i \cos({}^k \theta_i - {}^j \theta_i) + {}^j y_i \sin({}^k \theta_i - {}^j \theta_i) \\ {}^k y_i + {}^{-j} x_i \sin({}^k \theta_i - {}^j \theta_i) - {}^j y_i \cos({}^k \theta_i - {}^j \theta_i) \\ {}^k \theta_i - {}^j \theta_i \end{pmatrix}.
 \end{aligned} \tag{B.24}$$

Par convention on notera

$${}^k q_i \oplus (\ominus^j q_i) = {}^k q_i \ominus^j q_i, \tag{B.25}$$

ce qui est pratique, car en repartant de :

$${}^k q_i = {}^k q_j \oplus^j q_i, \tag{B.26}$$

on écrit

$${}^k q_i \ominus^j q_i = {}^k q_j \oplus^j q_i \ominus^j q_i, \tag{B.27}$$

soit le résultat recherché après simplifications :

$${}^k q_i \ominus^j q_i = {}^k q_j. \tag{B.28}$$

La nouvelle matrice de covariance est approchée par :

$$\Sigma_{k q_j} \approx J_{\oplus} \begin{pmatrix} \Sigma_{k q_i} & \Sigma_{k q_i, {}^i q_j} \\ \Sigma_{i q_j, {}^k q_i} & \Sigma_{i q_j} \end{pmatrix} J_{\oplus}^T \approx J_{\oplus} \begin{pmatrix} \Sigma_{k q_i} & \Sigma_{k q_i, {}^j q_i} J_{\ominus}^T \\ J_{\ominus} \Sigma_{j q_i, {}^k q_i} & J_{\ominus} \Sigma_{j q_i} J_{\ominus}^T \end{pmatrix} J_{\oplus}^T. \tag{B.29}$$

De même, on peut déterminer ${}^j q_i$ à partir de ${}^j q_k$ et de ${}^k q_i$ en repartant de la composition classique :

$$\begin{aligned}
 {}^j q_i &= {}^j q_k \oplus^k q_i \\
 &= (\ominus^k q_j) \oplus^k q_i.
 \end{aligned} \tag{B.30}$$

B. Changements de repères et incertitudes associées

En développant, on obtient :

$$\begin{aligned}
{}^j q_i &= \left(\begin{pmatrix} \cos(-{}^k \theta_j) & -\sin(-{}^k \theta_j) & 0 \\ \sin(-{}^k \theta_j) & \cos(-{}^k \theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^{-k} q_j) \right) \oplus {}^k q_i \quad (\text{B.31}) \\
&= \begin{pmatrix} \cos(-{}^k \theta_j) & -\sin(-{}^k \theta_j) & 0 \\ \sin(-{}^k \theta_j) & \cos(-{}^k \theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^{-k} q_j) \\
&+ \begin{pmatrix} \cos(-{}^k \theta_j) & -\sin(-{}^k \theta_j) & 0 \\ \sin(-{}^k \theta_j) & \cos(-{}^k \theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} {}^k q_i \\
&= \begin{pmatrix} \cos({}^k \theta_j) & \sin({}^k \theta_j) & 0 \\ -\sin({}^k \theta_j) & \cos({}^k \theta_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} ({}^k q_i - {}^k q_j) \\
&= \begin{pmatrix} ({}^k x_i - {}^k x_j) \cos({}^k \theta_j) + ({}^k y_i - {}^k y_j) \sin({}^k \theta_j) \\ -({}^k x_i - {}^k x_j) \sin({}^k \theta_j) + ({}^k y_i - {}^k y_j) \cos({}^k \theta_j) \\ {}^k \theta_i - {}^k \theta_j \end{pmatrix}.
\end{aligned}$$

Par convention on notera :

$$(\ominus {}^k q_j) \oplus {}^k q_i = \ominus {}^k q_j \oplus {}^k q_i, \quad (\text{B.32})$$

ce qui permet de faire des manipulations élégantes.

Par exemple, en repartant de la première équation (B.2)

$${}^k q_i = {}^k q_j \oplus {}^j q_i, \quad (\text{B.33})$$

on écrit

$$\ominus {}^k q_j \oplus {}^k q_i = \ominus {}^k q_j \oplus {}^k q_j \oplus {}^j q_i, \quad (\text{B.34})$$

soit le même résultat après simplifications :

$$\ominus {}^k q_j \oplus {}^k q_i = {}^j q_i. \quad (\text{B.35})$$

On obtient ensuite la matrice de covariance :

$$\Sigma_{j q_i} \approx J_{\oplus} \begin{pmatrix} \Sigma_{j q_k} & \Sigma_{j q_k, {}^k q_i} \\ \Sigma_{j q_k, {}^k q_i} & \Sigma_{k q_i} \end{pmatrix} J_{\oplus}^T \approx J_{\oplus} \begin{pmatrix} J_{\ominus \Sigma_{k q_j}} J_{\ominus}^T & \Sigma_{k q_j, {}^k q_i} J_{\ominus}^T \\ J_{\ominus \Sigma_{k q_j, {}^k q_i}} & \Sigma_{k q_i} \end{pmatrix} J_{\oplus}^T. \quad (\text{B.36})$$

Smith et al. [Smith *et al.*, 1990] montrent alors qu'il est plus rapide de calculer ces nouvelles matrices de covariance séquentiellement : estimer $\Sigma_{i q_j}$ et ${}^i q_j$ avant d'estimer $\Sigma_{k q_j}$; ou $\Sigma_{j q_k}$ et ${}^j q_k$ avant $\Sigma_{j q_i}$. En effet, exprimer les Jacobiennes à partir des données connues initialement complique leurs expressions.

B.5. Opérateur \oplus appliqué à une position

On peut appliquer l'opérateur \oplus pour le changement de repère d'une position à la place d'une pose [Li et Nashashibi, 2012]. Ainsi, la position ${}^k p_i$ de l'origine du repère

B. Changements de repères et incertitudes associées

i dans le repère k pourra s'exprimer à partir de la pose ${}^k q_j$ du repère j dans le repère k et de la position ${}^j p_i$ de l'origine du repère i dans le repère j :

$$\begin{aligned} {}^k p_i &= {}^k q_j \oplus {}^j p_i \\ &= {}^k p_j + \begin{pmatrix} \cos({}^k \theta_j) & -\sin({}^k \theta_j) \\ \sin({}^k \theta_j) & \cos({}^k \theta_j) \end{pmatrix} {}^j p_i \\ &= \begin{pmatrix} {}^k x_j + {}^j x_i \cos({}^k \theta_j) & -{}^j y_i \sin({}^k \theta_j) \\ {}^k y_j + {}^j x_i \sin({}^k \theta_j) & +{}^j y_i \cos({}^k \theta_j) \end{pmatrix}. \end{aligned} \quad (\text{B.37})$$

On note que seul le terme de droite peut être une position puisqu'une pose est nécessaire pour effectuer le changement de repère. On peut alors effectuer des combinaisons de pose à gauche avant le changement de repère de la position de droite. En considérant les poses comme des fonctions de changement de repères, cette nouvelle opération revient à appliquer la fonction définie par cette pose à la position :

$${}^k p_i = {}^k q_j \oplus {}^j p_i, \quad (\text{B.38})$$

$${}^k p_i = {}^k q_j ({}^j p_i). \quad (\text{B.39})$$

L'opérateur \ominus n'existe donc pas pour les positions. Un changement de repère n'est en effet pas possible avec une position. Il n'est pas possible de déterminer ${}^i p_j$ à partir de ${}^j p_i$ uniquement, l'orientation relative ${}^j \theta_i$ est nécessaire.

On peut alors exprimer la matrice de covariance :

$$\Sigma_{k p_i} \approx J_{\oplus} \begin{pmatrix} \Sigma_{k q_j} & \Sigma_{k q_j, {}^j p_i} \\ \Sigma_{j q_i, k p_j} & \Sigma_{j p_i} \end{pmatrix} J_{\oplus}^T, \quad (\text{B.40})$$

avec la Jacobienne J_{\oplus} :

$$J_{\oplus} = \frac{\partial {}^k p}{\partial ({}^k q_j, {}^j p_i)} = \frac{\partial ({}^k q_j \oplus {}^j p_i)}{\partial ({}^k q_j, {}^j p_i)} = \begin{pmatrix} 1 & 0 & -({}^k y_i - {}^k y_j) & \cos({}^k \theta_j) & -\sin({}^k \theta_j) \\ 0 & 1 & ({}^k x_i - {}^k x_j) & \sin({}^k \theta_j) & \cos({}^k \theta_j) \end{pmatrix}. \quad (\text{B.41})$$

B.6. Tableau résumé synthétique

La table (B.1) résume les opérations habituelles couramment utilisées en pratique.

B.7. Problèmes de non-linéarité lors de la propagation d'incertitude

Nous faisons ici la supposition que l'incertitude est gaussienne, elle peut être alors représentée par une ellipse. Pour propager cette incertitude lors d'opérations non linéaires, nous utilisons des approximations de Taylor à l'ordre 1. Ainsi les différentes opérations de changement de repère permettent de trouver une nouvelle pose ainsi que sa matrice de covariance.

Les figures B.3a, B.3b, B.3c et B.3d présente les deux opérations élémentaires ${}^k q_i = {}^k q_j \oplus {}^j q_i$ (Fig. B.3a) et ${}^i q_j = \ominus {}^j q_i$ (Fig. B.3b) ainsi que deux combinaisons de ces deux opérations ${}^k q_j = {}^k q_i \oplus (\ominus {}^j q_i)$ (Fig. B.3c) et ${}^j q_i = (\ominus {}^k q_j) \oplus {}^k q_i$ (Fig. B.3d).

B. Changements de repères et incertitudes associées

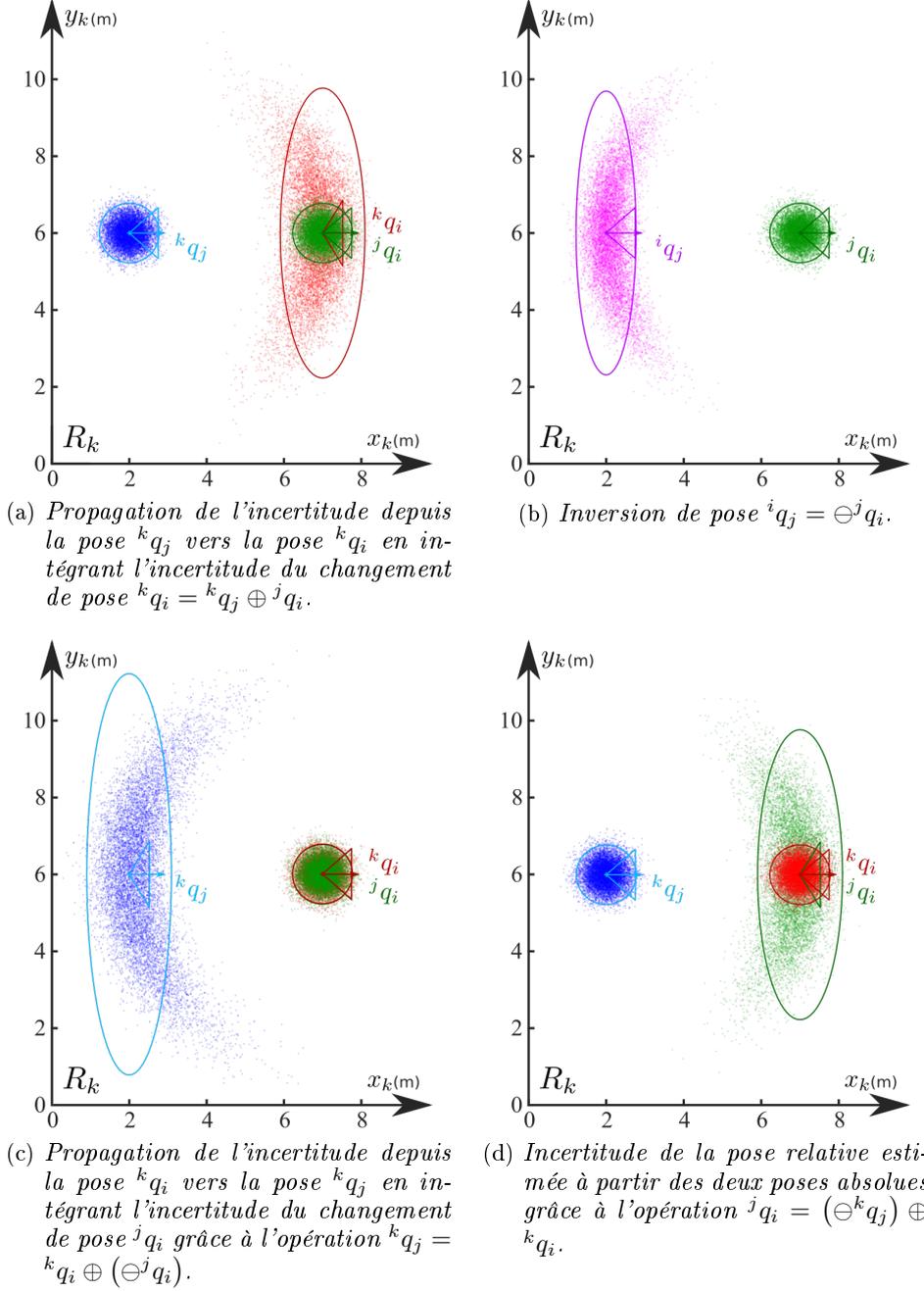


FIGURE B.3. – Les nuages de points représentent les distributions des poses approximées par des distributions normales illustrées par les ellipses et les cônes d'incertitude à 95%. Les éléments associés aux poses ${}^k q_i$, ${}^k q_j$, ${}^j q_i$ et ${}^i q_j$ sont respectivement représentés en rouge, en bleu, en vert et en violet. Les poses relatives ${}^j q_i$ et ${}^i q_j$ sont directement représentées dans les repères associés aux poses ${}^k q_j$ et ${}^k q_i$.

B. Changements de repères et incertitudes associées

Chaque figure présente les distributions des positions des entrées avec leurs poses moyennes ainsi que leurs ellipses et cônes d'incertitude. Ces distributions sont représentées par 10.000 positions avec des écarts-types de 0,316 m selon l'axe des abscisses et des ordonnées. L'orientation de la distribution à un écart-type de $16,9^\circ$. Les positions ${}^k p_i$ et ${}^k p_j$ sont séparées de 5 m (Fig. (B.2)). Chaque opération est appliquée aux moyennes d'entrées et à leurs incertitudes pour déterminer la moyenne du résultat de cette opération avec son incertitude. L'incertitude en position est illustrée par une ellipse d'incertitude, néanmoins en appliquant l'opération sur un échantillon aléatoire des distributions d'entrées, on obtient une distribution en banane du fait de la non-linéarité de l'opération. Ainsi l'ellipse d'incertitude du résultat ne représente pas correctement la distribution finale. On peut voir que la distribution en banane de l'opération ${}^k q_j = {}^k q_i \oplus (\ominus^j q_i)$ est plus prononcée que celles des autres opérations. Ce résultat vient du fait que la position finale ${}^k p_j$ dépend des deux orientations d'entrées ${}^k \theta_i$ et ${}^j \theta_i$ comme on peut le voir dans l'équation (B.24). Cette opération est donc plus sensible aux non-linéarités. Ces problèmes de distributions en banane sont observables et très significatifs lorsque les erreurs des orientations et les interdistances sont importantes.

Pour réduire ces problèmes de non-linéarité, la transformation non parfumée [Uhlmann, 1995] permet, à partir d'un certain nombre de points, de déterminer une gaussienne qui représente mieux la vraie distribution finale qui n'est plus une gaussienne.

C. Système de coordonnées géodésiques, repères ECEF et ENU

Pour des raisons de facilité de calcul, les repères cartésiens sont souvent utilisés. La Terre n'étant pas plane, il est néanmoins difficile d'utiliser un unique repère cartésien représentant de façon simple la position d'un véhicule, quelque soit le lieu où il se trouve sur Terre. Les récepteurs GNSS utilisent ainsi des coordonnées géodésiques décrivant une position à partir de deux angles et d'une hauteur. Le premier angle, la latitude lat , est relatif à l'équateur. Elle est positive dans l'hémisphère nord et négatif dans l'hémisphère sud. Le second angle, la longitude lon est relative au méridien de Greenwich. Cet angle augmente lors d'un déplacement d'ouest en est. Finalement, la hauteur h est donnée par rapport à un ellipsoïde de référence on appelle cette hauteur la hauteur ellipsoïdale. Cet ellipsoïde décrit la surface de la Terre de façon simplifiée en prenant en compte l'aplatissement aux pôles. Cette représentation est alors définie par une surface mathématique. En pratique, l'ellipsoïde WGS84 est le plus souvent utilisé. Il est la référence du système GPS. Dans certains cas comme dans la trame NMEA GGA, une forme plus complexe, le géoïde, peut-être utilisée pour décrire plus finement la forme terrestre. La figure C.1 représente ce géoïde avec l'ellipsoïde. Cette forme représente la surface équipotentielle de pesanteur au niveau moyen des océans et des mers. Elle est prolongée en dessous des continents. L'altitude décrit la hauteur par rapport à ce géoïde. La distance entre l'ellipsoïde et le géoïde s'appelle l'ondulation du géoïde. Définir l'ellipsoïde de référence est compliqué, l'axe de rotation de la Terre change constamment ainsi les pôles se déplacent de plusieurs mètres par an. Le centre de masse de la Terre se déplace lui aussi du fait des mouvements atmosphériques et océaniques. Les mesures sont généralement faites au niveau de l'écorce terrestre qui change en permanence du fait de la dérive des continents, des séismes, des marées... Ainsi des systèmes locaux plus exacts sont établis. L'ellipsoïde utilisé dans le système RGF93 [Duquenne, 2018] permet ainsi d'obtenir une référence dérivant avec la plaque tectonique de l'Eurasie, comme le système de référence terrestre européen ETRS89 sur lequel est basé le système français RGF93. Ainsi dans le cas où des mesures GNSS doivent être conservées pendant une longue période, comme c'est le cas pour des cartes définies à partir de ces mesures, il est

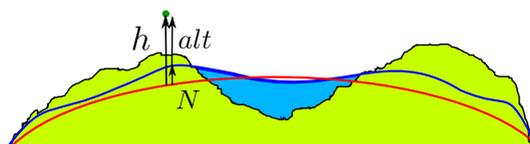


FIGURE C.1. – Ellipsoïde de référence (en rouge) et géoïde (en bleu) représentant le niveau moyen des océans et étant propagés sous les continents. L'altitude alt , donnée relativement au géoïde, peut-être calculée en soustrayant l'ondulation du géoïde N à la hauteur ellipsoïdale h .

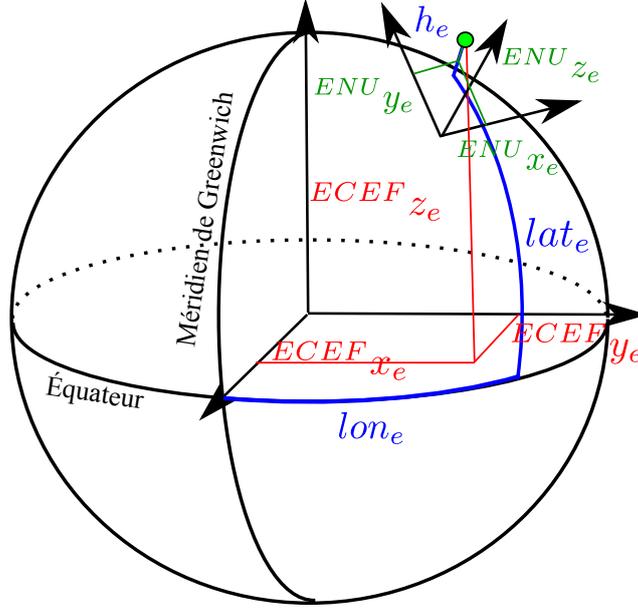


FIGURE C.2. – Position du égo-véhicule e dans un système de coordonnées géodésiques (lon_e, lat_e, h_e) ainsi que dans un repère ECEF $(ECEF x_e, ECEF y_e, ECEF z_e)$ et dans un repère ENU $(ENU x_e, ENU y_e, ENU z_e)$.

important de faire attention au système de référence utilisé. Il est conseillé d'utiliser un système local qui restera juste plus longtemps ainsi que de mettre à jour la carte tous les ans.

Pour simplifier les calculs, les coordonnées géodésiques sont converties en coordonnées métriques. L'une des méthodes les plus utilisées est la projection transverse universelle de Mercator (UTM). La Terre est découpée en fuseaux. Pour chaque fuseau un cylindre permet d'approximer la surface de la Terre. Cette méthode est néanmoins une projection. Ainsi des déformations apparaissent et l'algorithme n'est pas réversible, il n'est pas possible de retrouver exactement les mêmes coordonnées géographiques que celles qui ont été transformées en coordonnées UTM.

Une autre méthode est d'utiliser un repère orthonormé cartésien, le repère ECEF (Earth-Center Earth-Fixed), dont le centre est le même que celui du système géodésique, proche du centre de masse de la Terre. Les axes des abscisses et des ordonnées de ce repère sont définis sur le plan équatorial, l'axe des abscisses passe par le premier méridien et le dernier axe passe par le nord géographique. Pour effectuer cette conversion, il faut prendre en compte les paramètres géométriques de l'ellipsoïde [Duquenne, 2018]. Nous utilisons les paramètres de l'ellipsoïde IAG-GRS8 associé au référentiel géodésique RGF93 :

Son aplatissement $(1/f)$:

$$f = 1/298.257222101. \quad (C.1)$$

Les demi-axes $(a$ et $b)$:

$$\begin{aligned} a &= 6378137 \text{ m}, \\ b &= a(1 - f). \end{aligned} \quad (C.2)$$

C. Système de coordonnées géodésiques, repères ECEF et ENU

En posant :

$$e = \frac{\sqrt{a^2 - b^2}}{a}, \quad (\text{C.3})$$

et

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2(lat)}}. \quad (\text{C.4})$$

on peut alors définir les nouvelles coordonnées dans le repère ECEF :

$$\begin{aligned} ECEF x &= (N + h) \cos(lat) \cos(lon), \\ ECEF y &= (N + h) \cos(lat) \sin(lon), \\ ECEF z &= (N(1 - e^2) + h) \sin(lat). \end{aligned} \quad (\text{C.5})$$

Pour inverser cette transformation, on peut utiliser les équations suivantes [Zhu, 1994] :

En posant :

$$e_2 = \frac{\sqrt{a^2 - b^2}}{b}, \quad (\text{C.6})$$

$$\rho = \|ECEF p\|, \quad (\text{C.7})$$

$$F = 54b^2 ECEF z^2, \quad (\text{C.8})$$

$$G = \rho^2 + (1 - e^2) ECEF z^2 - e^2 (a^2 - b^2), \quad (\text{C.9})$$

$$C = \frac{e^4 F \rho^2}{G^3}, \quad (\text{C.10})$$

$$S = \sqrt[3]{1 + C + \sqrt{C^2 + 2C}}, \quad (\text{C.11})$$

$$P = \frac{F}{3(S + \frac{1}{S} + 1)^2 G^2}, \quad (\text{C.12})$$

$$Q = \sqrt{1 + 2e^4 P}, \quad (\text{C.13})$$

$$\rho_0 = \frac{-Pe^2\rho}{1+Q} + \sqrt{\frac{a^2}{2} \left(1 + \frac{1}{Q}\right) - \frac{P(1-e^2)ECEF z^2}{Q(1+Q)} - P\frac{\rho^2}{2}}, \quad (\text{C.14})$$

$$U = \sqrt{(\rho - e^2\rho_0)^2 + ECEF z^2}, \quad (\text{C.15})$$

$$V = \sqrt{(\rho - e^2\rho_0)^2 + (1 - e^2) ECEF z^2}, \quad (\text{C.16})$$

$$z_0 = \frac{b^2 z}{aV}. \quad (\text{C.17})$$

On peut alors calculer :

$$h = U \left(1 - \frac{b^2}{aV}\right), \quad (\text{C.18})$$

$$lat = \arctan \left(\frac{ECEF z + e_2^2 z_0}{\rho} \right), \quad (\text{C.19})$$

$$lon = \arctan 2 (ECEF y, ECEF x). \quad (\text{C.20})$$

C. Système de coordonnées géodésiques, repères ECEF et ENU

Ce changement de repère est simple à effectuer, il peut être éventuellement utilisé pour donner la position des satellites, mais n'est pas adapté pour la localisation de véhicules se déplaçant à la surface de la Terre. Le centre est trop loin de la zone où navigue le véhicule. De plus, si les axes pouvaient être orientés avec un plan tangent proche de la zone de navigation, des simplifications pourraient être faites en utilisant des positions à deux dimensions.

Le système de coordonnées ENU (East North Up) est un système de coordonnées cartésiennes tangent à la surface de la Terre ou parallèle à un plan tangent. Les trois axes sont alignés sur l'Est, le Nord et la verticale, i.e., selon l'axe perpendiculaire par rapport au plan Est-Nord. Les axes selon les directions Est et Nord définissent un plan tangent au modèle ellipsoïdal de la Terre.

Le passage du repère ECEF au repère ENU peut être fait en supposant la terre sphérique dont le centre est identique à celui de l'ellipsoïde. Le changement de repère est alors effectué de la façon suivante :

$$\begin{aligned} {}^{ENU}p &= \ominus {}^{ECEF}q_{ENU} \oplus {}^{ECEF}p \\ &= \begin{pmatrix} -\sin(lon_{ENU}) & \cos(lon_{ENU}) & 0 \\ -\sin(lat_{ENU})\cos(lon_{ENU}) & -\sin(lat_{ENU})\sin(lon_{ENU}) & \cos(lat_{ENU}) \\ \cos(lat_{ENU})\cos(lon_{ENU}) & \cos(lat_{ENU})\sin(lon_{ENU}) & \sin(lat_{ENU}) \end{pmatrix} ({}^{ECEF}p - {}^{ECEF}p_{ENU}). \end{aligned} \quad (C.21)$$

L'inverse de cette transformation est donnée par :

$$\begin{aligned} {}^{ECEF}p &= {}^{ECEF}q_{ENU} \oplus {}^{ENU}p \\ &= \begin{pmatrix} -\sin(lon_{ENU}) & -\sin(lat_{ENU})\cos(lon_{ENU}) & \cos(lat_{ENU})\cos(lon_{ENU}) \\ \cos(lon_{ENU}) & -\sin(lat_{ENU})\sin(lon_{ENU}) & \cos(lat_{ENU})\sin(lon_{ENU}) \\ 0 & \cos(lat_{ENU}) & \sin(lat_{ENU}) \end{pmatrix} {}^{ENU}p + {}^{ECEF}p_{ENU}. \end{aligned} \quad (C.22)$$

D. Intersection de covariance avec la forme de Joseph

Dans ce travail, nous avons présenté une utilisation de la forme de Joseph pour le filtre par intersection de covariance (CIF). Les différentes démonstrations présentées dans cette annexe sont basées sur des généralisations de démonstrations du Filtre de Kalman (KF). On peut, en effet, passer simplement d'une démonstration du KF à une démonstration du CIF en ajoutant les masses $\frac{1}{\omega}$ et $\frac{1}{1-\omega}$ devant les matrices de covariance des deux éléments à fusionner.

Afin de simplifier les notations, nous noterons dans cette annexe : $\hat{X} = X [t|t]$, $\hat{\Sigma}_X = \Sigma_X [t|t]$, $X = X [t|t-1]$, $\Sigma_X = \Sigma_X [t|t-1]$, $Z = Z [t]$ et $\Sigma_Z = \Sigma_Z [t]$.

Lorsque la fonction d'observation est l'identité, on a la première forme du CIF présenté par [Julier et Uhlmann, 1997b] :

$$\begin{cases} \hat{\Sigma}_X^{-1} &= \omega \Sigma_X^{-1} + (1-\omega) \Sigma_Z^{-1} \\ \hat{X} &= \hat{\Sigma}_X (\omega \Sigma_X^{-1} X + (1-\omega) \Sigma_Z^{-1} Z) \end{cases} \quad (\text{D.1})$$

Lorsque la matrice d'observation est différente de l'identité, la forme précédente peut s'écrire avec une matrice de gain K_ω :

$$\begin{cases} \hat{\Sigma}_X^{-1} &= \omega \Sigma_X^{-1} + (1-\omega) H^T \Sigma_Z^{-1} H \\ K_\omega &= (1-\omega) \hat{\Sigma}_X H^T \Sigma_Z^{-1} \\ \hat{X} &= X + K_\omega (Z - HX) \end{cases} \quad (\text{D.2})$$

Proposition 1 : La formulation (D.2) est équivalente à la formulation (D.3) suivante avec $\omega \neq 1$ et $\omega \neq 0$:

$$\begin{cases} K_\omega &= \frac{1}{\omega} \Sigma_X H^T \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1-\omega} \Sigma_Z \right)^{-1} \\ \hat{\Sigma}_X &= (I - K_\omega H) \frac{1}{\omega} \Sigma_X \\ \hat{X} &= X + K_\omega (Z - HX) \end{cases} \quad (\text{D.3})$$

Preuve : Rappelons tout d'abord l'identité matricielle de Woodbury :

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1}, \quad (\text{D.4})$$

où $A \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{n \times k}$, $C \in \mathbb{R}^{k \times k}$ et $V \in \mathbb{R}^{k \times n}$.

En prenant maintenant $A = \omega \Sigma_X^{-1}$, $U = H^T$, $V = H$ et $C = (1-\omega) \Sigma_Z^{-1}$, on a

$$\begin{aligned} \hat{\Sigma}_X &= (\omega \Sigma_X^{-1} + (1-\omega) H^T \Sigma_Z^{-1} H)^{-1} \\ &= \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1-\omega} \Sigma_Z \right)^{-1} H \frac{1}{\omega} \Sigma_X \\ &= \left(I - \frac{1}{\omega} \Sigma_X H^T \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1-\omega} \Sigma_Z \right)^{-1} H \right) \frac{1}{\omega} \Sigma_X \end{aligned} \quad (\text{D.5})$$

D. Intersection de covariance avec la forme de Joseph

à condition que $\omega \neq 1$ et $\omega \neq 0$.

Toujours avec ces conditions, en développant K_ω :

$$\begin{aligned}
 K_\omega &= (1 - \omega) \widehat{\Sigma}_X H^T \Sigma_Z^{-1} & (D.6) \\
 &= \left(I - \frac{1}{\omega} \Sigma_X H^T \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)^{-1} H \right) \frac{1}{\omega} \Sigma_X H^T (1 - \omega) \Sigma_Z^{-1} \\
 &= \frac{1}{\omega} \Sigma_X H^T \left(I - \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)^{-1} H \frac{1}{\omega} \Sigma_X H^T \right) (1 - \omega) \Sigma_Z^{-1}
 \end{aligned}$$

En remplaçant I par $\left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)^{-1} \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)$, on a

$$\begin{aligned}
 K_\omega &= \frac{1}{\omega} \Sigma_X H^T \left(\left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)^{-1} \frac{1}{1 - \omega} \Sigma_Z \right) (1 - \omega) \Sigma_Z^{-1} \\
 &= \frac{1}{\omega} \Sigma_X H^T \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1 - \omega} \Sigma_Z \right)^{-1} & (D.7)
 \end{aligned}$$

On en déduit donc

$$\widehat{\Sigma}_X = (I - K_\omega H) \frac{1}{\omega} \Sigma_X$$

CQFD.

Proposition 2 : La formulation (D.3) est équivalente à la forme de Joseph généralisée pour le CIF (D.8) ci-dessous avec $\omega \neq 1$ et $\omega \neq 0$:

$$\left\{ \begin{array}{l}
 \Sigma_{Y,\omega} = H \left(\frac{1}{\omega} \Sigma_X \right) H^T + \left(\frac{1}{1 - \omega} \Sigma_Z \right) \\
 K_\omega = \frac{1}{\omega} \Sigma_X H^T \Sigma_{Y,\omega}^{-1} \\
 \widehat{\Sigma}_X = (I - K_\omega H) \left(\frac{1}{\omega} \Sigma_X \right) (I - K_\omega H)^T + K_\omega \left(\frac{1}{1 - \omega} \Sigma_Z \right) K_\omega^T \\
 Y = Z - HX \\
 \widehat{X} = X + K_\omega Y
 \end{array} \right. \quad (D.8)$$

Preuve : En posant $Y = Z - HX$ et $\Sigma_{Y,\omega} = H \left(\frac{1}{\omega} \Sigma_X \right) H^T + \frac{1}{1 - \omega} \Sigma_Z$, on obtient directement les expressions de K_ω et \widehat{X} à partir de la formulation (D.3). En déve-

D. Intersection de covariance avec la forme de Joseph

loppant l'expression de $\widehat{\Sigma}_X$ de la formulation (D.8), on a

$$\begin{aligned}
\widehat{\Sigma}_X &= (I - K_\omega H) \left(\frac{1}{\omega} \Sigma_X \right) (I - K_\omega H)^T + K_\omega \frac{1}{1-\omega} \Sigma_Z K_\omega^T & (D.9) \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - (I - K_\omega H) \frac{1}{\omega} \Sigma_X H^T K_\omega^T + K_\omega \frac{1}{1-\omega} \Sigma_Z K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + K_\omega \left(H \frac{1}{\omega} \Sigma_X H^T + \frac{1}{1-\omega} \Sigma_Z \right) K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + K_\omega \Sigma_{Y,\omega} K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X - \frac{1}{\omega} \Sigma_X H^T K_\omega^T + \frac{1}{\omega} \Sigma_X H^T \cancel{\Sigma_{Y,\omega}^{-1} \Sigma_{Y,\omega}} K_\omega^T \\
&= (I - K_\omega H) \frac{1}{\omega} \Sigma_X
\end{aligned}$$

On retrouve l'expression de $\widehat{\Sigma}_X$ de la formulation (D.3), CQFD.

Discussion Les trois formulations (D.2), (D.3) et (D.8) sont équivalentes. Néanmoins, la forme de Joseph généralisée (D.8) est mieux conditionnée numériquement que les autres formulations car la mise à jour de la matrice de covariance s'obtient comme la somme de matrices définies positives et en évitant plusieurs inversions de matrices.

On remarque toutefois que lorsque ω vaut 0 ou 1, une division par 0 apparaît dans la formulation (D.8).

Les limites en 0 et 1 de $\widehat{\Sigma}_X$ sont données facilement pour la formulation (D.2) :

$$\widehat{\Sigma}_X^{-1} = \omega \Sigma_X^{-1} + (1 - \omega) H^T \Sigma_Z^{-1} H \quad (D.10)$$

$$\lim_{\omega \rightarrow 0} \widehat{\Sigma}_X = (H^T \Sigma_Z^{-1} H)^{-1} \quad (D.11)$$

$$\lim_{\omega \rightarrow 1} \widehat{\Sigma}_X = \Sigma_X \quad (D.12)$$

Les limites en 0 et 1 de \widehat{X} sont aussi obtenues avec la formulation (D.2) :

$$\widehat{X} = X + (1 - \omega) \widehat{\Sigma}_X H^T \Sigma_Z^{-1} (Z - HX) \quad (D.13)$$

$$\lim_{\omega \rightarrow 0} \widehat{X} = X + \widehat{\Sigma}_X H^T \Sigma_Z^{-1} (Z - HX) \quad (D.14)$$

$$\lim_{\omega \rightarrow 1} \widehat{X} = X \quad (D.15)$$

Ceci permet donc d'éviter les divisions par 0.

La forme de Joseph peut être alors utilisée que lorsque ω est différent de 0 et de 1 et, en pratique, suffisamment éloigné de ces valeurs. Il est donc recommandé d'utiliser la forme de l'équation (D.2) dans la fonction de coût qui doit être minimisée. En utilisant cette forme et en prenant comme norme le déterminant, la fonction de coût revient alors à :

$$\det(\widehat{\Sigma}_X) = \det\left((\omega \Sigma_X^{-1} + (1 - \omega) H^T \Sigma_Z^{-1} H)^{-1}\right) \quad (D.16)$$

$$= \frac{1}{\det(\omega \Sigma_X^{-1} + (1 - \omega) H^T \Sigma_Z^{-1} H)} \quad (D.17)$$

D. Intersection de covariance avec la forme de Joseph

En calculant avant la minimisation l'inverse de Σ_X et de Σ_Z , il est ainsi possible d'obtenir une fonction de coût sans risque de division par 0 et sans inversion de matrice à réaliser pendant la minimisation. On obtient ainsi le ω optimal. On calcule ensuite la matrice de covariance finale avec la forme de Joseph. Afin d'accélérer encore l'estimation de ω , il est aussi possible d'obtenir une approximation de ce paramètre de façon analytique ce qui évite la phase de minimisation lourde en calculs ([Niehsen, 2002, Franken et Hupper, 2005]). Ces méthodes sont appelées « intersection de covariance rapide » ou FastCIF en anglais.

Bibliographie

- [Al Hage, 2016] AL HAGE, J. (2016). *Fusion de Données Tolérante Aux Défaillances : Application à La Surveillance de l'intégrité d'un Système de Localisation*. Thèse de doctorat, Lille 1.
- [Alonso et al., 2018] ALONSO, I. P., GONZALO, R. I., ALONSO, J., GARCÍA-MORCILLO, Á., FERNÁNDEZ-LLORCA, D. et SOTELO, M. Á. (2018). The Experience of DRIVERTIVE-DRIVERless cooperaTIVE VEHicle-Team in the 2016 GCDC. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1322–1334.
- [Aufrère et al., 2010] AUFRÈRE, R., KARAM, N., CHAUSSE, F. et CHAPUIS, R. (2010). A state exchange approach in real conditions for multi-robot cooperative localization. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4346–4351, Taipei, Taiwan.
- [Axelsson, 2017] AXELSSON, J. (2017). Safety in Vehicle Platooning : A Systematic Literature Review. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1033–1045.
- [Bahr et al., 2009] BAHR, A., WALTER, M. R. et LEONARD, J. J. (2009). Consistent cooperative localization. *In IEEE International Conference on Robotics and Automation*, pages 3415–3422, Kobe, Japan.
- [Ben-Israel, 2002] BEN-ISRAEL, A. (2002). The Moore of the Moore-Penrose inverse. *Electronic Journal of Linear Algebra*, 9:150–157.
- [Bender et al., 2014] BENDER, P., ZIEGLER, J. et STILLER, C. (2014). Lanelets : Efficient map representation for autonomous driving. *In IEEE Intelligent Vehicles Symposium*, pages 420–425, Dearborn, MI, USA.
- [Bengtsson et Baerveldt, 2003] BENGTTSSON, O. et BAERVELDT, A.-J. (2003). Robot localization based on scan-matching—estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems*, 44(1):29–40.
- [Betaille et Toledo-Moreo, 2010] BETAILLE, D. et TOLEDO-MOREO, R. (2010). Creating Enhanced Maps for Lane-Level Vehicle Navigation. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):786–798.
- [Biswas et al., 2006] BISWAS, P., LIANG, T.-C., TOH, K.-C., YE, Y. et WANG, T.-C. (2006). Semidefinite Programming Approaches for Sensor Network Localization With Noisy Distance Measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371.
- [Carrillo-Arce et al., 2013] CARRILLO-ARCE, L. C., NERURKAR, E. D., GORDILLO, J. L. et ROUMELIOTIS, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417, Tokyo, Japan.
- [Censi, 2007] CENSI, A. (2007). An accurate closed-form estimate of ICP's covariance. *In IEEE International Conference on Robotics and Automation*, pages 3167–3172, Roma, Italy.
- [Censi, 2008] CENSI, A. (2008). An ICP variant using a point-to-line metric. *In IEEE International Conference on Robotics and Automation*, pages 19–25, Pasadena, CA, USA.
- [Challita et al., 2009] CHALLITA, G., MOUSSET, S., NASHASHIBI, F. et BENSRAHAIR, A. (2009). An application of V2V communications : Cooperation of vehicles for a better car tracking using GPS and vision systems. *In IEEE Vehicular Networking Conference*, pages 1–6, Tokyo, Japan.

Bibliographie

- [Chateau *et al.*, 2017] CHATEAU, T., CHABOT, F., TEULIÈRE, C., CHAOUCH, M. et RABARISOA, J. (2017). Deep MANTA : A Coarse-to-Fine Many-Task Network for Joint 2D and 3D Vehicle Analysis from Monocular Image. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 2040–2049, Honolulu, United States.
- [Chu *et al.*, 2012] CHU, K., LEE, M. et SUNWOO, M. (2012). Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1599–1616.
- [Coenen *et al.*, 2017] COENEN, M., ROTTENSTEINER, F. et HEIPKE, C. (2017). Detection and 3d modelling of vehicles from terrestrial stereo image pairs. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 42 (2017), Nr. 1W1*, 42(1W1):505–512.
- [Cruz *et al.*, 2017] CRUZ, S. B., ABRUDAN, T. E., XIAO, Z., TRIGONI, N. et BARROS, J. (2017). Neighbor-Aided Localization in Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, 18(10):2693–2702.
- [de Boor, 2001] de BOOR, C. (2001). *A Practical Guide to Splines*. Springer New York.
- [Del Moral, 1997] DEL MORAL, P. (1997). Nonlinear filtering : Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 325(6):653–658.
- [Delobel, 2018] DELOBEL, L. (2018). *Agrégation d'information pour la localisation d'un robot mobile sur une carte imparfaite*. Thèse de doctorat, Université Clermont Auvergne.
- [Drawil et Basir, 2010] DRAWIL, N. M. et BASIR, O. (2010). Intervehicle-Communication-Assisted Localization. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):678–691.
- [Duquenne, 2018] DUQUENNE, F. (2018). Les systèmes de référence terrestre et leurs réalisations Cas des territoires français. *XYZ*, (154):9.
- [Eiter *et al.*, 2019] EITER, T., FÜREDER, H., KASSLATTER, F., PARREIRA, J. X. et SCHNEIDER, P. (2019). Towards a Semantically Enriched Local Dynamic Map. *International Journal of Intelligent Transportation Systems Research*, 17(1):32–48.
- [ETSI, 2014a] ETSI (2014a). Intelligent transport systems (ITS) ; Vehicular communications ; Basic set of applications ; Part 2 : specification of cooperative awareness basic service. Rapport technique ETSI EN 302 637-2 v1.3.1.
- [ETSI, 2014b] ETSI (2014b). Intelligent transport systems (ITS) ; Vehicular communications ; Basic set of applications ; Part 3 : specification of decentralized environmental notification basic service. Rapport technique ETSI EN 302 637-3 v1.2.1.
- [ETSI, 2018] ETSI (2018). Intelligent Transport System (ITS) ; Vehicular Communications ; Basic Set of Applications ; Informative Report for the Collective Perception Service. Rapport technique ETSI TR 103 562 v0.0.8.
- [Fox *et al.*, 2000] FOX, D., BURGARD, W., KRUPPA, H. et THRUN, S. (2000). A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots*, 8(3):325–344.
- [Franken et Hupper, 2005] FRANKEN, D. et HUPPER, A. (2005). Improved fast covariance intersection for distributed data fusion. *In 2005 7th International Conference on Information Fusion*, volume 1, pages 7 pp.–.
- [Gao *et al.*, 2019] GAO, C., ZHAO, G. et FOURATI, H. (2019). *Cooperative Localization and Navigation : Theory, Research, and Practice*. CRC Press.
- [Héry *et al.*, 2017a] HÉRY, E., MASI, S., XU, P. et BONNIFAIT, P. (2017a). Map-based curvilinear coordinates for autonomous vehicles. *In IEEE Intelligent Transportation Systems Conference*, pages 16–19, Yokohama, Japan.
- [Héry *et al.*, 2017b] HÉRY, E., XU, P. et BONNIFAIT, P. (2017b). Along-track localization for cooperative autonomous vehicles. *In IEEE Intelligent Vehicles Symposium*, pages 11–14, Redondo Beach, CA, USA.

Bibliographie

- [Héry *et al.*, 2017c] HÉRY, E., XU, P. et BONNIFAIT, P. (2017c). One-dimensional cooperative localization for vehicles equipped with mono-frequency GNSS receivers. In *European Navigation Conference*, pages 9–12, Lausanne, Switzerland.
- [Héry *et al.*, 2018] HÉRY, E., XU, P. et BONNIFAIT, P. (2018). LiDAR based relative pose and covariance estimation for communicating vehicles exchanging a polygonal model of their shape. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Madrid, Spain.
- [Héry *et al.*, 2019a] HÉRY, E., XU, P. et BONNIFAIT, P. (2019a). Distributed asynchronous cooperative localization with inaccurate gnss positions. In *IEEE Intelligent Transportation Systems Conference*, Auckland, New Zealand.
- [Héry *et al.*, 2019b] HÉRY, E., XU, P. et BONNIFAIT, P. (2019b). Pose and covariance matrix propagation issues in cooperative localization with lidar perception. In *IEEE Intelligent Vehicles Symposium*, Paris, France.
- [Hoang *et al.*, 2018] HOANG, G. M., DENIS, B., HÄIRRI, J. et SLOCK, D. (2018). Cooperative Localization in VANETs : An Experimental Proof-of-Concept Combining GPS, IR-UWB Ranging and V2V Communications. In *Workshop on Positioning, Navigation and Communications*, pages 1–6, Bremen, Germany.
- [Hoang *et al.*, 2015] HOANG, G. M., DENIS, B., HÄIRRI, J. et SLOCK, D. T. M. (2015). Select thy neighbors : Low complexity link selection for high precision cooperative vehicular localization. In *IEEE Vehicular Networking Conference*, pages 36–43, Kyoto, Japan.
- [Hoang *et al.*, 2017] HOANG, G. M., DENIS, B., HÄIRRI, J. et SLOCK, D. T. M. (2017). Mitigating unbalanced GDoP effects in range-based vehicular Cooperative Localization. In *IEEE International Conference on Communications Workshops*, pages 659–664, Paris, France.
- [Howard *et al.*, 2003] HOWARD, A., MATARIC, M. J. et SUKHATME, G. S. (2003). Putting the 'I' in 'team' : An ego-centric approach to cooperative localization. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 868–874 vol.1, Taipei, Taiwan.
- [Jaulin *et al.*, 2001] JAULIN, L., KIEFFER, M., DIDRIT, O. et WALTER, E. (2001). *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer London Ltd.
- [Jo et Lee, 2008] JO, K. H. et LEE, J. (2008). Cooperative localization of multiple robots with Constraint Propagation technique. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3477–3482, Nice, France.
- [Julier et Uhlmann, 2001] JULIER, S. et UHLMANN, J. (2001). General Decentralized Data Fusion with Covariance Intersection (CI). In HALL, D. et LLINAS, J., éditeurs : *Multi-sensor Data Fusion*, volume 3. CRC Press.
- [Julier et Uhlmann, 1997b] JULIER, S. J. et UHLMANN, J. K. (1997b). A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference*, volume 4, pages 2369–2373 vol.4, Albuquerque, NM, USA.
- [Julier et Uhlmann, 1997a] JULIER, S. J. et UHLMANN, J. K. (Juiller 1997a). New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics.
- [Karam, 2009] KARAM, N. (2009). *Agrégation de données décentralisées pour la localisation multi-véhicules*. Thèse de doctorat, Clermont-Ferrand 2.
- [Karam *et al.*, 2006a] KARAM, N., CHAUSSE, F., AUFRERE, R. et CHAPUIS, R. (2006a). Cooperative Multi-Vehicle Localization. In *IEEE Intelligent Vehicles Symposium*, pages 564–570, Meguro-Ku, Japan.
- [Karam *et al.*, 2006b] KARAM, N., CHAUSSE, F., AUFRERE, R. et CHAPUIS, R. (2006b). Localization of a Group of Communicating Vehicles by State Exchange. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 519–524.

Bibliographie

- [Khammari *et al.*, 2005] KHAMMARI, A., NASHASHIBI, F., ABRAMSON, Y. et LAURGEAU, C. (2005). Vehicle detection combining gradient analysis and AdaBoost classification. *In IEEE Intelligent Transportation Systems*, pages 66–71, Vienna, Austria.
- [Kieffer *et al.*, 2000] KIEFFER, M., JAULIN, L., WALTER, É. et MEIZEL, D. (2000). Robust Autonomous Robot Localization Using Interval Analysis. *Reliable Computing*, 6(3):337–362.
- [Kim *et al.*, 2015] KIM, J., JO, K., LIM, W., LEE, M. et SUNWOO, M. (2015). Curvilinear-Coordinate-Based Object and Situation Assessment for Highly Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1559–1575.
- [Kurazume *et al.*, 1994] KURAZUME, R., NAGATA, S. et HIROSE, S. (1994). Cooperative positioning with multiple robots. *In IEEE International Conference on Robotics and Automation*, pages 1250–1257 vol.2, San Diego, CA, USA.
- [Lalescu, 2009] LALESCU, C. C. (2009). Two hierarchies of spline interpolations. Practical algorithms for multivariate higher order splines. *arXiv :0905.3564 [cs]*.
- [Lassoued *et al.*, 2017] LASSOUED, K., BONNIFAIT, P. et FANTONI, I. (2017). Cooperative Localization with Reliable Confidence Domains between Vehicles sharing GNSS Pseudorange Errors with no Base Station. *IEEE Intelligent Transportation Systems Magazine*, 9(1):22–34.
- [Lassoued *et al.*, 2015] LASSOUED, K., FANTONI, I. et BONNIFAIT, P. (2015). Mutual Localization and Positioning of Vehicles Sharing GNSS Pseudorange : Sequential Bayesian Approach and Experiments. *In IEEE International Conference on Intelligent Transportation Systems*, pages 1896–1901, Las Palmas, Spain.
- [Lassoued *et al.*, 2014] LASSOUED, K., STANOI, O., BONNIFAIT, P. et FANTONI, I. (2014). Mobile Robots Cooperation with Biased Exteroceptive Measurements. *In International Conference on Control, Automation, Robotics and Vision*, pages 1838–1840, Marina Bay Sands, Singapore.
- [Li, 2012] LI, H. (2012). *Cooperative Perception : Application in the Context of Outdoor Intelligent Vehicle Systems*. Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris.
- [Li et Nashashibi, 2012] LI, H. et NASHASHIBI, F. (2012). Cooperative multi-vehicle localization using split covariance intersection filter. *In IEEE Intelligent Vehicles Symposium*, pages 211–216, Alcal de Henares, Spain.
- [Li et Nashashibi, 2013] LI, H. et NASHASHIBI, F. (2013). Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44.
- [Lin *et al.*, 2005] LIN, Y., VERNAZA, P., HAM, J. et LEE, D. D. (2005). Cooperative relative robot localization with audible acoustic sensing. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3764–3769, Las Vegas, NV, USA.
- [Liu *et al.*, 2018] LIU, J., RIZOS, C. et CAI, B.-g. (2018). Integrity Monitoring of Vehicle Positioning using Cooperative Measurements under Connected Vehicles Environments. *In International Global Navigation Satellite Systems Symposium*, page 17, Sydney, Australia.
- [Liu et Chen, 1998] LIU, J. S. et CHEN, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044.
- [Low, 2004] LOW, K.-L. (2004). Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Technical Report TR04-004, Department of Computer Science University of North Carolina at Chapel Hill.
- [Madhavan *et al.*, 2002] MADHAVAN, R., FREGENE, K. et PARKER, L. E. (2002). Distributed heterogeneous outdoor multi-robot localization. *In IEEE International Conference on Robotics and Automation*, volume 1, pages 374–381 vol.1, Washington, DC, USA.

Bibliographie

- [Mario et Rife, 2010] MARIO, C. et RIFE, J. (2010). Integrity Monitoring of Vision-Based Automotive Lane Detection Methods. *International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 245–255.
- [Martinelli et al., 2005] MARTINELLI, A., PONT, F. et SIEGWART, R. (2005). Multi-Robot Localization Using Relative Observations. In *IEEE International Conference on Robotics and Automation*, pages 2797–2802, Barcelona, Spain.
- [Masi et al., 2018] MASI, S., XU, P. et BONNIFAIT, P. (2018). Adapting the Virtual Platooning Concept to Roundabout Crossing. In *IEEE Intelligent Vehicles Symposium*, pages 1366–1372.
- [Montesano et al., 2005] MONTESANO, L., GASPAR, J., SANTOS-VICTOR, J. et MONTANO, L. (2005). Cooperative localization by fusing vision-based bearing measurements and motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2333–2338, Edmonton, AB, Canada.
- [Mu et al., 2011] MU, H., BAILEY, T., THOMPSON, P. et DURRANT-WHYTE, H. (2011). Decentralised Solutions to the Cooperative Multi-Platform Navigation Problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1433–1449.
- [Müller et al., 2014] MÜLLER, F. d. P., DIAZ, E. M., KLOIBER, B. et STRANG, T. (2014). Bayesian cooperative relative vehicle positioning using pseudorange differences. In *IEEE/ION Position, Location and Navigation Symposium*, pages 434–444, Monterey, CA, USA.
- [Niehsen, 2002] NIEHSEN, W. (2002). Information fusion based on fast covariance intersection filtering. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, volume 2, pages 901–904 vol.2.
- [NovAtel, 2013] NOVATEL (2013). Positioning modes of operation application note. Rapport technique APN-051.
- [Parlement européen, 2015] PARLEMENT EUROPÉEN (2015). Déploiement du système eCall embarqué ***II.
- [Peker et Acarman, 2017] PEKER, A. U. et ACARMAN, T. (2017). VANET-Assisted Cooperative Vehicle Mutual Positioning : Feasibility Study. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E100.A(2):448–456.
- [Poggenhans et al., 2018] POGGENHANS, F., PAULS, J.-H., JANOSOVITS, J., ORF, S., NAUMANN, M., KUHN, F. et MAYR, M. (2018). Lanelet2 : A high-definition map framework for the future of automated driving. In *IEEE Intelligent Transportation Systems Conference*, Maui, HI, USA.
- [Prakhya et al., 2015] PRAKHYA, S. M., BINGBING, L., RUI, Y. et LIN, W. (2015). A closed-form estimate of 3D ICP covariance. In *IAPR International Conference on Machine Vision Applications*, pages 526–529, Tokyo, Japan.
- [Quigley et al., 2009] QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R. et NG, A. Y. (2009). ROS : an open-source robot operating system. In *ICRA workshop on open source software*, volume 3. Kobe, Japan.
- [Reinhardt et al., 2012] REINHARDT, M., NOACK, B. et HANEBECK, U. D. (2012). Closed-form optimization of covariance intersection for low-dimensional matrices. In *International Conference on Information Fusion*, pages 1891–1896, Singapore.
- [Rife et Xiao, 2010] RIFE, J. et XIAO, X. (2010). Estimation of Spatially Correlated Errors in Vehicular Collaborative Navigation with Shared GNSS and Road-Boundary Measurements. *International Technical Meeting of the Satellite Division of the Institute of Navigation*, page 12.
- [Rohani et al., 2016] ROHANI, M., GINGRAS, D. et GRUYER, D. (2016). A Novel Approach for Improved Vehicular Positioning Using Cooperative Map Matching and Dynamic Base Station DGPS Concept. *IEEE Transactions on Intelligent Transportation Systems*, 17(1): 230–239.

Bibliographie

- [Rohani *et al.*, 2015] ROHANI, M., GINGRAS, D., VIGNERON, V. et GRUYER, D. (2015). A New Decentralized Bayesian Approach for Cooperative Vehicle Localization Based on Fusion of GPS and VANET Based Inter-Vehicle Distance Measurement. *IEEE Intelligent Transportation Systems Magazine*, 7(2):85–95.
- [Roumeliotis et Bekey, 2000] ROUMELIOTIS, S. I. et BEKEY, G. A. (2000). Collective localization : A distributed Kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2958–2965 vol.3, San Francisco, CA, USA.
- [Roumeliotis et Bekey, 2002] ROUMELIOTIS, S. I. et BEKEY, G. A. (2002). Distributed multi-robot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795.
- [Roumeliotis et Rekleitis, 2003] ROUMELIOTIS, S. I. et REKLEITIS, I. M. (2003). Analysis of multirobot localization uncertainty propagation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1763–1770 vol.2, Las Vegas, NV, USA.
- [Roumeliotis et Rekleitis, 2004] ROUMELIOTIS, S. I. et REKLEITIS, I. M. (2004). Propagation of Uncertainty in Cooperative Multirobot Localization : Analysis and Experimental Results. *Autonomous Robots*, 17(1):41–54.
- [Safaei et Mahyuddin, 2019] SAFAEI, A. et MAHYUDDIN, M. N. (2019). Adaptive Cooperative Localization Using Relative Position Estimation for Networked Systems With Minimum Number of Communication Links. *IEEE Access*, 7:32368–32382.
- [Severi *et al.*, 2018] SEVERI, S., WYMEERSCH, h., HÄRRI, J., ULMSCHNEIDER, M., DENIS, B. et BARTELS, M. (2018). Beyond GNSS : Highly accurate localization for cooperative-intelligent transport systems. In *IEEE Wireless Communications and Networking Conference*, pages 1–6, Barcelona, Spain.
- [Shetty, 2017] SHETTY, A. (2017). *GPS-LiDAR Sensor Fusion Aided by 3D City Models for UAVs - Semantic Scholar*. Master of Science in Aerospace Engineering, University of Illinois, Urbana-Champaign.
- [Shimada *et al.*, 2015] SHIMADA, H., YAMAGUCHI, A., TAKADA, H. et SATO, K. (2015). Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems. *Journal of Transportation Technologies*, 05(02):102–112.
- [Smith *et al.*, 1990] SMITH, R., SELF, M. et CHEESEMAN, P. (1990). Estimating Uncertain Spatial Relationships in Robotics. In COX, I. J. et WILFONG, G. T., éditeurs : *Autonomous Robot Vehicles*, pages 167–193. Springer New York, New York, NY.
- [Sorenson et Alspach, 1970] SORENSON, H. W. et ALSPACH, D. L. (1970). Gaussian sum approximations for nonlinear filtering. In *1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control*, pages 193–193.
- [Steux *et al.*, 2000] STEUX, B., COULOMBEAU, P. et LAURGEAU, C. (2000). RTmaps : a framework for prototyping automotive multi-sensor applications. In *IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA.
- [Tischler et Hummel, 2005] TISCHLER, K. et HUMMEL, B. (2005). Enhanced environmental perception by inter-vehicle data exchange. In *IEEE Intelligent Vehicles Symposium*, pages 313–318, Las Vegas, NV, USA.
- [Uhlmann, 1995] UHLMANN, J. K. (1995). *Dynamic Map Building and Localization : New Theoretical Foundations*. Thèse de doctorat, University of Oxford.
- [Vo et Ma, 2006] VO, B.-N. et MA, W.-K. (2006). The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104.
- [Wanasinghe *et al.*, 2014] WANASINGHE, T. R., MANN, G. K. I. et GOSINE, R. G. (2014). Decentralized Cooperative Localization for Heterogeneous Multi-robot System Using Split Covariance Intersection Filter. In *Canadian Conference on Computer and Robot Vision*, pages 167–174, Montréal, QC, Canada.

Bibliographie

- [Wang *et al.*, 2002] WANG, H., KEARNEY, J. et ATKINSON, K. (2002). Arc-length parameterized spline curves for real-time simulation. *In International Conference on Curves and Surfaces*.
- [Wasik *et al.*, 2019] WASIK, A., LIMA, P. U. et MARTINOLI, A. (2019). A robust localization system for multi-robot formations based on an extension of a Gaussian mixture probability hypothesis density filter. *Autonomous Robots*.
- [Welte *et al.*, 2019] WELTE, A., XU, P. et BONNIFAIT, P. (2019). Four-Wheeled Dead-Reckoning Model Calibration using RTS Smoothing. *In International Conference on Robotics and Automation*, pages 312–318, Montreal, QC, Canada.
- [Worrall et Nebot, 2007] WORRALL, S. et NEBOT, E. (2007). Using Non-Parametric Filters and Sparse Observations to Localise a Fleet of Mining Vehicles. *In IEEE International Conference on Robotics and Automation*, pages 509–516, Roma, Italy.
- [Xu *et al.*, 2018] XU, P., DHERBOMEZ, G., HÉRY, E., ABIDLI, A. et BONNIFAIT, P. (spring 2018). System architecture of a driverless electric car in the grand cooperative driving challenge. *IEEE Intelligent Transportation System Magazine*, 10:47–59.
- [Zelazo *et al.*, 2015] ZELAZO, D., FRANCHI, A., BÜLTHOFF, H. H. et ROBUFFO GIORDANO, P. (2015). Decentralized rigidity maintenance control with range measurements for multi-robot systems. *International Journal of Robotics Research*, 34(1):105–128.
- [Zelazo *et al.*, 2014] ZELAZO, D., FRANCHI, A. et GIORDANO, P. R. (2014). Rigidity theory in SE(2) for unscaled relative position estimation using only bearing measurements. *In European Control Conference*, pages 2703–2708, Strasbourg, France.
- [Zhang *et al.*, 2018] ZHANG, G., WEN, W. et HSU, L. (2018). A novel GNSS based V2V cooperative localization to exclude multipath effect using consistency checks. *In IEEE/ION Position, Location and Navigation Symposium*, pages 1465–1472, Monterey, CA, USA.
- [Zhang *et al.*, 2017] ZHANG, X., XU, W., DONG, C. et DOLAN, J. M. (2017). Efficient L-shape fitting for vehicle detection using laser scanners. *In IEEE Intelligent Vehicles Symposium*, pages 54–59, Redondo Beach, CA, USA.
- [Zhu, 1994] ZHU, J. (1994). Conversion of Earth-centered Earth-fixed coordinates to geodetic coordinates. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):957–961.
- [Zhu et Kia, 2019] ZHU, J. et KIA, S. S. (2019). Cooperative Localization Under Limited Connectivity. *IEEE Transactions on Robotics*, 35(6):1523–1530.
- [Zoghby *et al.*, 2014] ZOGHBY, N. E., CHERFAOUI, V. et DENOËUX, T. (2014). Evidential distributed dynamic map for cooperative perception in VANets. *In IEEE Intelligent Vehicles Symposium*, pages 1421–1426, Ypsilanti, MI, USA.