



Lot 3 : Service navettes zone parking

L3.2 : Rapport pour le système complet de courte distance

| | |
|------------------|------------------------|
| Programme | FUI23 |
| Référence | TORNADO L3.2 |
| Version | V.0 |
| Date | 06 / 05 / 2020 |
| Porteur | Institut Pascal |
| Auteur(s) | A. Gerardeaux |
| Contributeurs(s) | A. Guerrero R. Chapuis |



EASY MILE

exoskills



Inria



AVAIRX
PIONEERING NEXT FRONTIERS

RAMBOUILLET
TERRITOIRES



GROUPE RENAULT

Financé par

bpifrance

Région
Île-de-France



La Région
Auvergne-Rhône-Alpes



Yvelines
Le Département

Pôles de labellisation



mov'eo
Imagine mobility



Table des matières

| | |
|---|----|
| 1 Avant propos..... | 1 |
| 1.1 Contexte..... | 1 |
| 1.2 Description générale du système..... | 1 |
| 2 Guidage visuel..... | 3 |
| 2.1 Introduction..... | 3 |
| 2.2 L'algorithme mis en œuvre..... | 3 |
| 2.3 Conclusion..... | 6 |
| 3 Module de traversabilité (CMCDOT et détection des bordures de route)..... | 6 |
| 3.1 Introduction..... | 6 |
| 3.2 Estimation de la géométrie du sol..... | 7 |
| 3.3 Carte de risque et d'occupation..... | 8 |
| 3.4 Détection des bordures de route..... | 8 |
| 3.5 Résultats..... | 9 |
| 3.6 Conclusion..... | 12 |
| 4 Les interactions avec l'environnement haut niveau..... | 12 |
| 4.1 L'Unité Embarquée en Véhicule (UEV)..... | 12 |
| 4.2 Interaction avec la gestion de flotte..... | 15 |
| 4.2.1 Introduction..... | 15 |
| 4.2.2 Spécification des scenarii et de la syntaxe des échanges..... | 16 |
| 4.2.3 Développement interne à la navette..... | 19 |
| 4.3 Interaction avec l'infrastructure intelligente..... | 22 |
| 4.3.1 Introduction..... | 22 |
| 4.3.2 Spécifications..... | 23 |
| 5 Résultats pour l'ensemble du système..... | 25 |
| 5.1 Introduction..... | 25 |
| 5.2 Résultats du géoréférencement de la carte..... | 25 |
| 5.3 Résultats pour le guidage de la navette et Lidars plans..... | 26 |
| 5.4 Résultats associés aux télémètres..... | 28 |

| | |
|---|----|
| 5.5 Résultats pour la gestion des missions..... | 32 |
| 5.6 Conclusion sur les résultats obtenus avec le système complet..... | 33 |
| REFERENCES..... | 34 |

Table figures

| | |
|--|----|
| Figure 1 Vue d'ensemble de l'architecture complète du système..... | 2 |
| Figure 2 PAVIN : Plateforme Auvergnate pour les Véhicules Intelligents..... | 3 |
| Figure 3 illustration du principe de fonctionnement de la navigation autonome utilisant un SLAM visuel..... | 5 |
| Figure 4 Champ aléatoire conditionnel spatio-temporel..... | 7 |
| Figure 5 Champ aléatoire conditionnel spatio-temporel utilisant différentes valeurs pour les seuils supérieur et inférieur..... | 8 |
| Figure 6 Nuage de points VLP16: filtrage des points bruyants et aberrants lorsqu'ils sont présents sur l'ez10..... | 9 |
| Figure 7 Influence de bêta et gamma (estimations d'élévation) sur la segmentation du sol: (gauche) bêta =0,5 et gamma =0,1; (droite) bêta=0,75 et gamma=0,2..... | 10 |
| Figure 8 Comparaison des résultats de la détection des bordures de route (en rouge) avec la vérité terrain (en bleu)..... | 10 |
| Figure 9 EZ10 conduisant en mode suivi de trajectoire autonome sur PAVIN avec carte routière..... | 11 |
| Figure 10 EZ10 suivant une trajectoire de référence avec évitement d'obstacles et fusion des bordures de route et des obstacles sur la carte de navigation..... | 11 |
| Figure 11 L'Unité embarquée en véhicule (UEV)..... | 12 |
| Figure 12 L'environnement de l'UEV..... | 13 |
| Figure 13 TORNADO système : Architecture physique..... | 14 |
| Figure 14 Trajectoire de référence et liste des arrêts..... | 15 |
| Figure 15 Mission GoTo : Machine d'état..... | 16 |
| Figure 16 Tableau des composantes requises pour le message d'état du véhicule..... | 17 |
| Figure 17 Tableau des composantes requises pour le message d'envoi des missions..... | 18 |
| Figure 18 Tableau des composantes requises pour le message d'annulation des missions | 18 |
| Figure 19 Cas numéro 1 : La trajectoire de référence est connue en WGS84..... | 19 |
| Figure 20 Cas numéro 2 : La trajectoire de référence N'est pas connue en WGS84..... | 20 |
| Figure 21 Interface graphique pour le monitoring de missions indépendamment de la gestion de flotte..... | 20 |

| | |
|--|----|
| Figure 22 Graphe d'organisation des différents packages ROS développés autour du SLAM visuel pour la gestion des missions..... | 21 |
| Figure 23 ZOOM sur la zone drive : position et orientation de la caméra du système d'analyse infrastructure de l'IFSTTAR..... | 22 |
| Figure 24 Définition du repère commun..... | 24 |
| Figure 25 Résultat du géoréférencement de la carte SLAM pour la détection d'obstacles par le dispositif IFSTTAR..... | 26 |
| Figure 26 Trajectoire par fermeture de boucle après modification du code de guidage..... | 27 |
| Figure 27 Visualisation des obstacles détectés par les Lidars plans..... | 27 |
| Figure 28 Interface de gestion des vitesses..... | 28 |
| Figure 29 Installation des Velodyne sur le toit du véhicule..... | 28 |
| Figure 30 Configuration réseau bi-Velodyne..... | 29 |
| Figure 31 Réglages des paramètres extrinsèques des Velodyne..... | 30 |
| Figure 32 Architecture de traitement et filtrage des données Velodyne..... | 31 |
| Figure 33 Illustration des problèmes de détection du sol..... | 31 |
| Figure 34 Illustration du problème de distance d'arrêt..... | 32 |
| Figure 35 Module d'antennes de l'UEV sur le toit de la navette..... | 32 |
| Figure 36 Interface graphique interactive présente dans la navette..... | 33 |

1 Avant propos

1.1 Contexte



Le projet TORNADO est un projet pluri partenaires, subventionné par le Fonds Unique Interministériel (FUI), et qui a pour objet les interactions du véhicule autonome et de l'infrastructure pour les services de mobilité en zone peu dense.

Porté par Renault et labellisé par les pôles de compétitivité Mov'eo, Viaméca et Aerospace Valley, ce projet se concentre sur l'expérimentation et la spécification des technologies nécessaires aux véhicules et à l'infrastructure pour la mise en place de nouveaux services de mobilité. Cette faisabilité est démontrée à échelle et environnement réels.

L'objectif général du lot 3 dans ce projet est d'apporter une expertise pour la modélisation de l'environnement extérieur, des véhicules et des capteurs afin de faire évoluer une navette de façon autonome le long d'un centre commercial. Il est également envisagé de travailler sur l'élaboration des scénarios pour les tests des différents algorithmes et programmes développés par les différents partenaires.

Le lot 3 propose ainsi un service automatisé qui permet des déplacements de voyageurs dans un parking commercial. La navette voyage sur un site dédié, le centre commercial ou le parking mais ouvert aux usagers fréquentant habituellement un centre commercial. Par exemple, les piétons, probablement avec leurs chariots, peuvent emprunter le parcours de la navette.

A terme, les services doivent être opérés sans conducteur, et des études additionnelles sont nécessaires et prises en compte dans le projet.

1.2 Description générale du système

Le système complet comprend plusieurs composantes, dans cette partie nous allons introduire le rôle de chacune des composantes. L'objectif du lot 3 est la mise en place d'un service de navettes autonomes permettant à des usagers la desserte d'arrêts dans une zone commerciale de la ville de Rambouillet (Bel Air La Foret), les cas d'usages et descriptifs des démonstrations sont détaillés dans le livrable L2, et plus précisément pour le service courte distance dans le livrable «R2.2 - Rapport détaillé sur les cas d'usage à courte distance (RT)».

Dans un premier temps le système de navigation et de cartographie est basé sur un algorithme de SLAM visuel (Simultaneous Localization And Mapping) développé par l'Institut Pascal. Cet algorithme utilise les images fournies par le système multi-caméra du véhicules pour créer une trajectoire et retrouver en temps réel la position du véhicule dans cette carte.

Le système complet inclu un système de sécurité basé sur la notion de «virtual bumper» qui doit permettre l'arrêt immédiat du véhicule en mode autonome lors de la détection d'objets classés à haut risque, cette partie est indépendante du reste du système et fait l'objet d'un livrable «R3.3 – Rapport «virtual bumper» système».

Ensuite les composantes du système complet incluent un module d'évaluation temps réel de la traversabilité du couloir de navigation appelé CMCDOT (conditional Monte Carlo occupancy tracker) ce système est développé par l'INRIA Grenoble et est intégré dans une carte de calcul auxilliaire NVIDIA (Jetson TX2), placé sur le toit de la navette. Cette carte

utilise les données de deux LiDAR 3D installés sur le toit à l'avant et à l'arrière de la navette pour fournir à l'ordinateur de bord une grille d'occupation des obstacles statiques et dynamiques afin de planifier dans les algorithmes de navigation des stratégies d'évitement. Ce module sera complété par un algorithme hors ligne de détection des bordures de route et fera l'objet d'une partie entière dans ce livrable (Partie, 3.4).

Un autre module de communication fourni par NEAVIA (Groupe Lacroix) : l'Unité Embarquée en Véhicule (UEV) va permettre l'interaction avec une gestion de flotte et l'exploitation de caméras sur mats et boîtiers de perception. Les interactions qui transiteront via ce module seront explicitées plus en détail dans une partie dédiée de ce livrable (Partie, 4.2 et 4.3). Pour résumer, la navette pourra communiquer via ce matériel son état d'avancement avec la gestion de flotte et pourra recevoir des ordres de mission. De plus, une interface homme machine développée par EXOSKILLS bénéficiera des informations retournées par la navette pour nourrir une application utilisable par le public via un téléphone portable. Les utilisateurs pourront ainsi être informés de la venue de la navette et des arrêts qu'elle desservira. La navette pourra aussi bénéficier, via ce matériel, d'informations de localisation et d'orientation d'obstacles et/ou véhicules envoyées par une infrastructure intelligente (infrastructure développée par l'IFSTTAR et composée d'une camera présente dans une zone à risque et d'un algorithme de détection d'obstacle par réseau de neurones profonds).

Sur la figure 1 (Fig. 1) est présenté un schéma illustrant l'architecture du système complet :

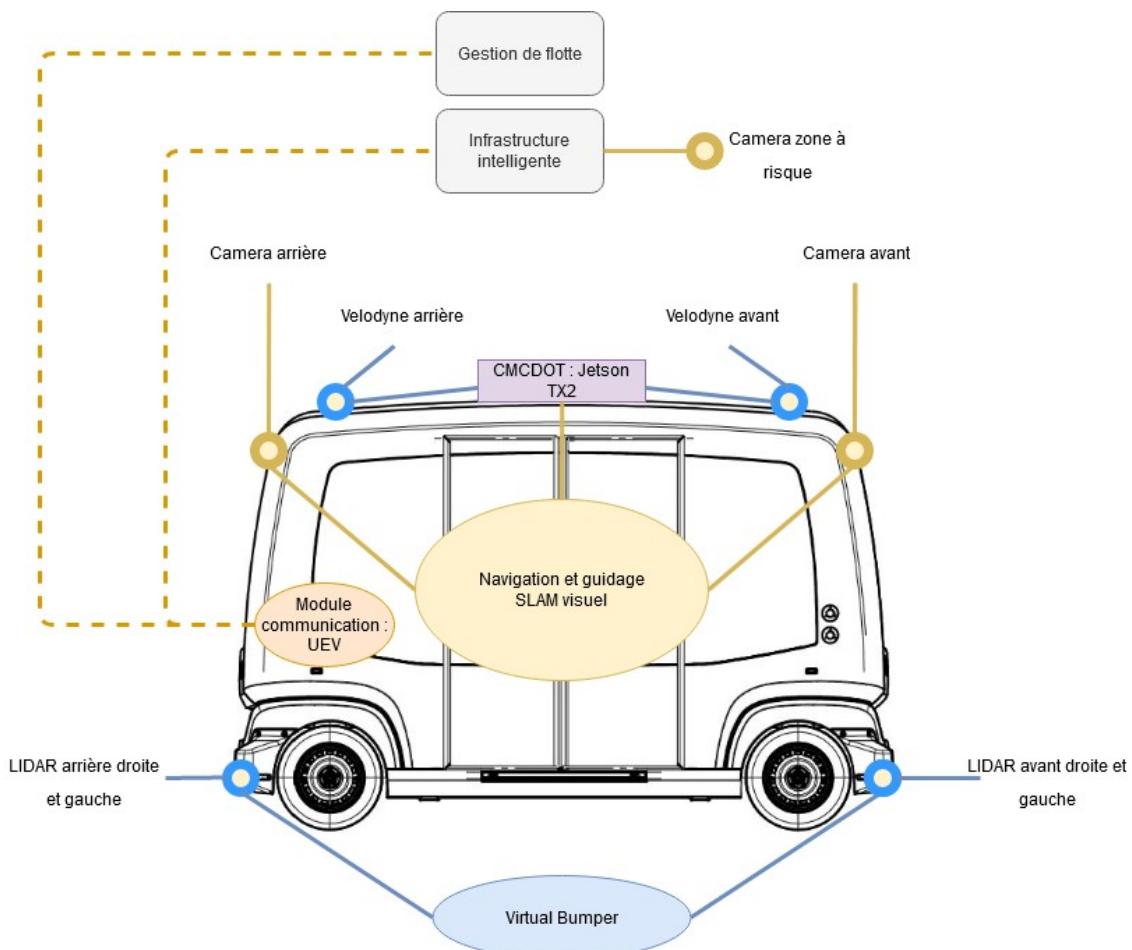


FIGURE 1 VUE D'ENSEMBLE DE L'ARCHITECTURE COMPLÈTE DU SYSTÈME

La plupart des tests et expérimentations sont réalisés sur la plateforme expérimentale PAVIN (Plateformes Auvergne pour Véhicules INtelligents) (Fig. 2) de l'Institut Pascal. Cette

plateforme permet la circulation de navettes dans une mini-ville équipée de feux tricolores, de ronds-points, de ruelles étroites et de zones de dépassement. Ce site fut le berceau du développement des premières navettes autonomes de l'Institut Pascal en partenariat avec la société Ligier, dans le cadre du projet VIPA-Fleet. Aujourd'hui doublée d'une plateforme virtuelle, permettant de rejouer des scénarios, elle permet des investigations fines sur le développement des algorithmes de guidage et d'évitement d'obstacles.



FIGURE 2 PAVIN : PLATEFORME AUVERGNATE POUR LES VÉHICULES INTELLIGENTS

2 Guidage visuel

2.1 Introduction

Cette partie va permettre de présenter l'algorithme utilisé pour la navigation de la navette de l'Institut Pascal (EZ10) et en particulier pour sa localisation. Cette navette a été conçue et produite par les Automobiles Ligier et a déjà été mise en service pendant plusieurs mois sur un site industriel avec plus de 1000 Km parcourus sans conducteur, le guidage étant réalisé grâce à un système de vision. L'élément clé utilisé pour la localisation du véhicule est un algorithme de SLAM (Simultaneous Localization And Mapping) visuel développé à l'Institut Pascal . Le développement des algorithmes nécessaires au bon fonctionnement du véhicule a débuté il y a plus de quinze ans. Au cours de cette période, de nombreuses expérimentations ont été réalisées. L'expérience acquise a permis de faire évoluer l'algorithme de localisation par vision avec des orientations dictées par le besoin de résoudre les problèmes qui se posent sur le terrain.

2.2 L'algorithme mis en œuvre

Le principe consiste à conduire manuellement dans un premier temps le véhicule sur la trajectoire qui devra être suivie lors de la phase de navigation autonome. Il s'agit de la phase d'apprentissage. Les images fournies par le système multi-caméra sont enregistrées et un algorithme de SLAM visuel basé sur un ajustement de faisceaux permet de reconstruire la trajectoire du véhicule ainsi qu'une carte d'indices visuels (communément appelés amers et qui sont des points d'intérêt de l'image caractérisés des détecteurs de Harris) pour lesquels les coordonnées 3D sont calculées. Lors de la phase de navigation autonome, les amers sont retrouvés dans les images et un calcul de la pose du système multi-caméra permet de connaître en temps réel la pose du véhicule dans cet environnement. En comparant la pose obtenue avec la trajectoire à suivre (enregistrée manuellement au préalable), nous calculons

L3.2 |Rapport pour le système complet courte distance

un écart latéral et un écart angulaire qui sont utilisés conjointement avec la courbure de la trajectoire afin de commander le véhicule et lui faire suivre la trajectoire d'apprentissage. L'odométrie (qui regroupe les capteurs de déplacement) est utilisée afin d'accélérer la mise en correspondance des amers visuels et en cas de perte des images pour rester sur la trajectoire pendant l'arrêt du véhicule. Sur la figure 3, un schéma illustrant le principe de fonctionnement (Fig. 3) est présenté.

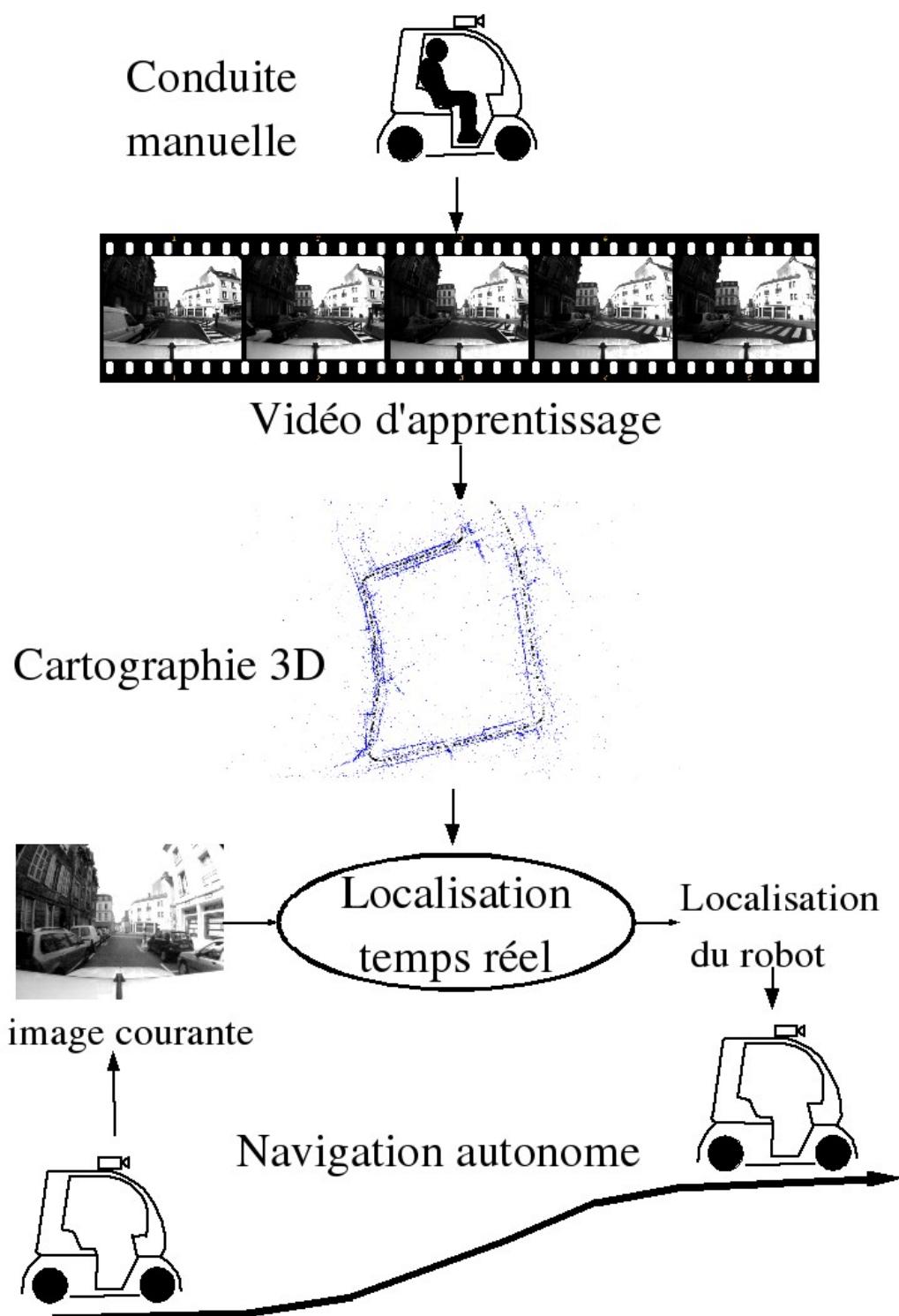


FIGURE 3 ILLUSTRATION DU PRINCIPE DE FONCTIONNEMENT DE LA NAVIGATION AUTONOME UTILISANT UN SLAM VISUEL

Dans le détail, le système utilisé est constitué de deux caméras à champs non recouvrants (une tournée vers l'avant et l'autre vers l'arrière). Cela est dicté par la volonté d'éviter au maximum les problèmes de sur-exposition ou sous-exposition des images lorsque le véhicule est face au soleil ou lorsqu'il entre ou sort d'un bâtiment. Cela permet aussi d'avoir un véhicule symétrique puisqu'il est prévu de l'utiliser en marche avant et en marche arrière. Cette contrainte pratique a des répercussions sur l'algorithme utilisé pour l'étalonnage et il est nécessaire d'en dire quelques mots. On commence par un étalonnage intrinsèque des caméras sur une mire (c'est-à-dire l'obtention des paramètres propres aux caméras). Cette étape est réalisée une fois pour toutes (sauf si l'on doit changer une caméra). L'étalonnage extrinsèque (l'extraction de la pose de la caméra arrière par rapport à la caméra avant) ne peut pas être obtenu avec les techniques classiques dédiées à la stéréo vision à champs recouvrants. Dans un premier temps, nous avons tenté de créer un champ recouvrant à l'aide d'un miroir plan. Même si la méthode donne de bons résultats, elle a été abandonnée car trop difficile à utiliser en pratique. Finalement, c'est un auto-étalonnage qui est utilisé : une trajectoire en boucle dans laquelle le véhicule fait un demi tour est enregistrée. La reconstruction 3D est faite en cherchant les appariements entre caméra avant et arrière et un ajustement de faisceaux final durant lequel les paramètres inter-caméras sont relâchés permet d'obtenir de très bons résultats. Cette étape de calibration extrinsèque doit être réalisée dès que les caméras ont été déplacées l'une par rapport à l'autre.

2.3 Conclusion

En conclusion le véhicule a pu rouler pendant près de 1500 km et transporter plus de 300 passagers à ce jour. L'algorithme de SLAM visuel qui a été développé montre une bonne robustesse puisqu'on observe très peu de défaut de localisation. Cependant, même si la technologie a atteint une certaine maturité, tous les problèmes ne sont pas résolus pour laisser du public aux mains d'un véhicule entièrement automatisé, y compris sur un site fermé tel que celui du site de la ville de Rambouillet (Bel Air La Foret). Pour plus de détails sur l'état de l'art, les expérimentations, les plans de validation, les difficultés rencontrées, les voies d'amélioration et verrous scientifiques voici un article complet retracant tous ces points : .

3 Module de traversabilité (CMCDOT et détection des bordures de route)

3.1 Introduction

La détection d'obstacles pour une navigation autonome dans un environnement semi-urbain peut être assurée à l'aide d'une grande variété de capteurs. Dans cette partie, nous discutons du problème de détection d'obstacles à l'aide d'un système de cartographie d'occupation développé par le centre de recherche Inria Grenoble. Ce système qui est intégré au véhicule sur une plateforme de calcul auxiliaire NVIDIA (Jetson TX2) et utilise des capteurs LiDAR 3D (Velodyne) positionnés sur le toit de la navette. Dans les sections suivantes, nous discutons de ces systèmes de perception développés par INRIA Grenoble. Notre discussion se concentrera sur les principes généraux de fonctionnement et comment ils peuvent être intégrés dans un système de contrôle de navigation autonome plus large. Afin de déterminer la présence d'obstacles et de risques de collision, INRIA a développé une méthode bayésienne pour générer des grilles de risques et d'occupation. Le processus de calcul des cartes de risque d'occupation et de collision à l'aide d'un capteur LiDAR 3D se compose de deux étapes : estimation du sol et génération de carte de risque d'occupation et de collision dont les capacités et les limites sont discutées dans les sections suivantes.

3.2 Estimation de la géométrie du sol

Comme décrit dans [6], le sol est modélisé comme un champ aléatoire conditionnel spatio-temporel (STCRF) et représenté par un réseau 2D régulier sur le plan XY. Chaque nœud $N_i = (nxi, nyi)$ du réseau est associé à une variable aléatoire continue cachée Gi . Les mesures LiDAR consistent en un ensemble d'emplacements 3D (xj, yj, zj) . Chaque mesure LiDAR est associée à son nœud le plus proche. Un champ aléatoire conditionnel est utilisé pour modéliser les interactions entre les nœuds et représentant le sol. La variable Gi est considérée comme dépendante de son voisinage spatial (Spatial Cliques), de son voisinage temporel (Temporal Clique) et des observations et variables cachées (Data Cliques). Ces potentielles «Cliques» suivent une distribution de Gibbs, avec les paramètres α, β et y qui représentent respectivement la confiance sur les estimations d'élévation des cellules voisines, les estimations d'élévation du «time step» précédent et les mesures, comme décrit dans . La représentation graphique du STCRF est illustrée sur la figure suivante (Fig. 4) :

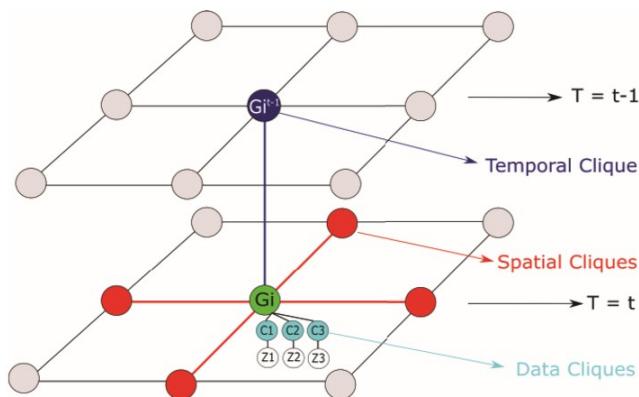


FIGURE 4 CHAMP ALÉATOIRE CONDITIONNEL SPATIO-TEMPOREL

L'estimation d'élévation de chaque variable de cellule Gi est calculée à l'aide de l'inférence postérieure maximale. Puisque le calcul de la solution exacte est insoluble, le processus d'inférence est simplifié en utilisant un modèle gaussien pour l'état fondamental. Pour approximer la distribution complète sur Gi et Ci un algorithme itératif d'attente-maximisation (EM) est utilisé. La première étape du processus EM calcule la probabilité de chaque mesure Zj et est un point fondamental. La probabilité Ci est calculée en utilisant l'état approximatif Gi du nœud le plus proche i . La probabilité est calculée en utilisant la hauteur relative dZj entre la surface locale du sol et la mesure, comme illustré sur la figure suivante (Fig. 5) :

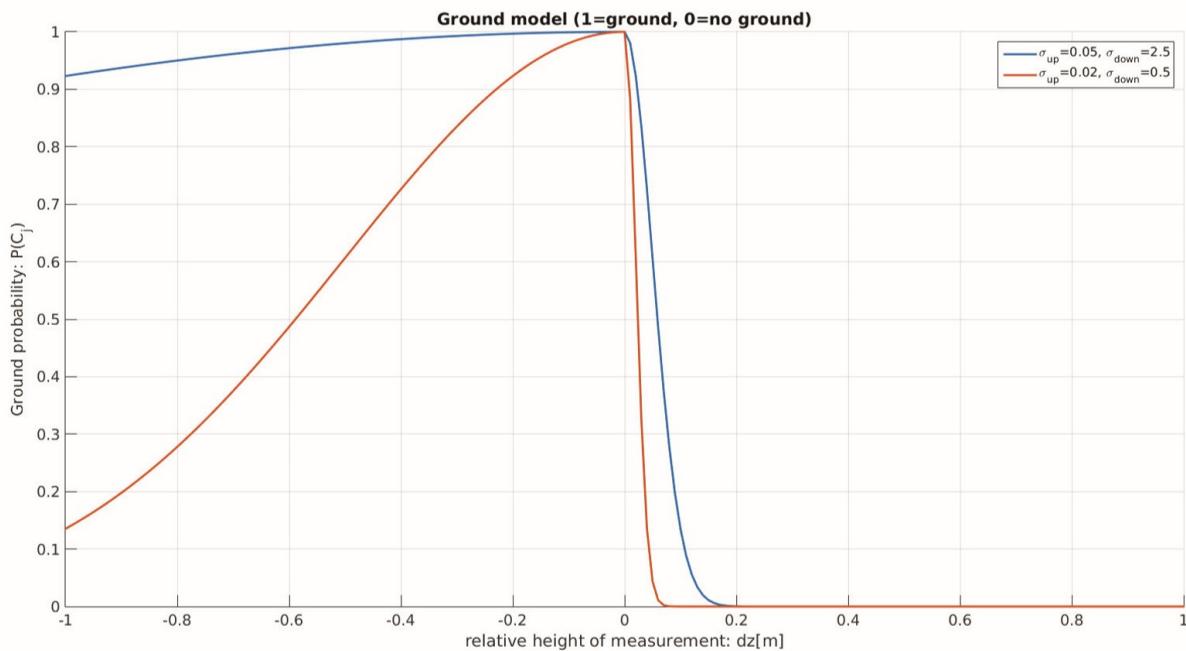


FIGURE 5 CHAMP ALÉATOIRE CONDITIONNEL SPATIO-TEMPOREL UTILISANT DIFFÉRENTES VALEURS POUR LES SEUILS SUPÉRIEUR ET INFÉRIEUR

3.3 Carte de risque et d'occupation

Les grilles d'occupation et de risque sont générées à l'aide du suivi d'occupation conditionnelle de Monte Carlo (CMCDOT) présenté dans . La partie statique est modélisée via une grille d'occupation tandis que la partie dynamique est modélisée à l'aide de particules en mouvement. Le CMCDOT représente des zones statiques, dynamiques, libres et inconnues comme des états formels dans leur processus de filtrage. Le CMCDOT est un filtre d'occupation bayésien modifié (BOF) qui comprend des états cachés qui représentent le contenu des cellules. Ensuite, au lieu de filtrer directement l'occupation d'une cellule, le CMCDOT détermine la distribution d'occupation d'une cellule en utilisant ses états cachés. L'algorithme CMCDOT consiste en deux étapes : la prédiction et l'évaluation. Une fois que les distributions des états ont été estimées, la distribution de particule est ré-échantillonnée pour travailler sur les nouvelles parties dynamique de la carte.

3.4 Détection des bordures de route

Afin de pallier les problèmes de détection des bordures de route qui sont trop petites pour être efficacement détectée par le LiDAR du toit lors du fonctionnement du CMCDOT, nous avons développé à l'Institut Pascal un algorithme de segmentation du sol à l'aide des données LiDAR 3D. Les bordures de route sont extraites préalablement grâce à un passage du véhicule sur la trajectoire cible. L'algorithme utilise des caractéristiques géométriques (différence d'élévation, distance horizontale) ainsi que la réflectance LiDAR pour projeter les bordures dans une carte des bordures de route utilisant la localisation fournie par le système VSLAM lors de la phase d'apprentissage. Ici la segmentation du sol n'est pas la même que celle utilisée par CMCDOT. Dans la partie suivante nous présenterons en plus des résultats obtenus avec le CMCDOT les résultats conjoints obtenus par la fusion des obstacles et des bordures.

3.5 Résultats

En raison de la géométrie du véhicule EZ10, le télémètre multi-nappe utilisé (VLP16 : télémètre 16 nappes de chez Velodyne) a été monté sur le dessus du véhicule avec une inclinaison sur l'axe des x (angle de tangage $\theta \approx 14.5^\circ$) ce qui maximise le champ de vision à courte et moyenne portée de ce capteur LiDAR 3D. On suppose que la matrice de transformation du LiDAR au repère véhicule est connue (il a cependant été nécessaire d'élaborer une technique de calibration permettant cela). L'utilisation de la pose VLP16 entraîne des impacts laser sur le toit de l'EZ10. Ainsi, les nuages de points LiDAR peuvent être filtrés pour rejeter les points aberrants et les points se trouvant à l'intérieur du cadre de délimitation de l'EZ10 avant d'être transférés vers le module de génération de la carte de risque, d'occupation et de collision (Fig. 6). Les points LiDAR situés à l'intérieur de la boîte de délimitation du véhicule pourraient conduire le CMCDOT à détecter des obstacles à l'intérieur de la boîte de délimitation du véhicule, ce qui pourrait amener les algorithmes de planification et de guidage à arrêter le véhicule ou, s'il est déjà arrêté, à éviter que la boîte de délimitation du véhicule ne commence à bouger.

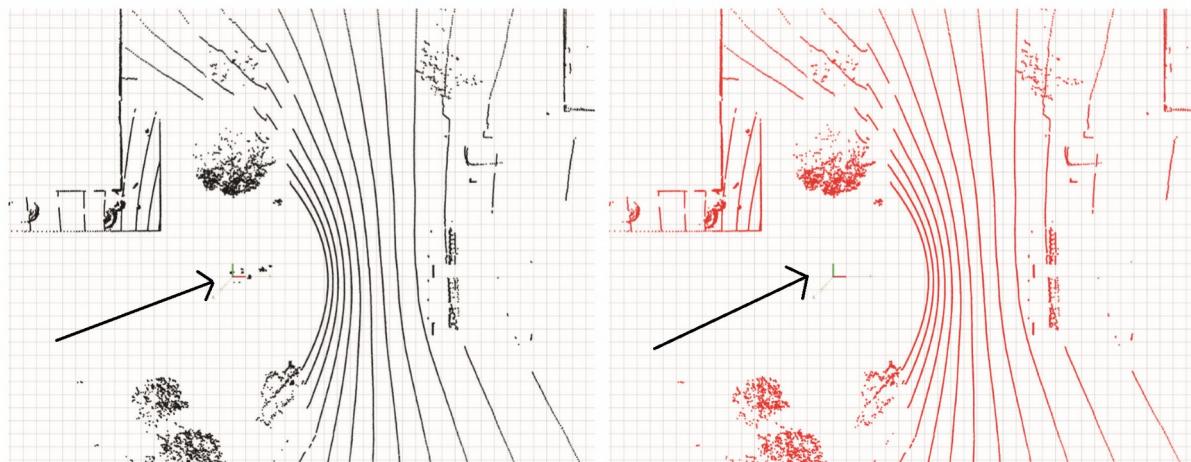


FIGURE 6 NUAGE DE POINTS VLP16: FILTRAGE DES POINTS BRUYANTS ET ABERRANTS LORSQU'ILS SONT PRÉSENTS SUR L'EZ10

La figure suivante (Fig. 7) montre l'influence des paramètres β et γ (les estimations d'élévation du «time step» précédent et les mesures) sur la segmentation au sol pour un jeu de données issue de Rambouillet. Il est clair que l'augmentation des paramètres β et γ permet une meilleure détection d'obstacles au prix d'avoir des obstacles dynamique plus longs du fait du filtrage temporel. Il faut noter que l'augmentation de ces paramètres peut conduire le CMCDOT à considérer, selon la géométrie du terrain, certaines régions du sol comme des obstacles.

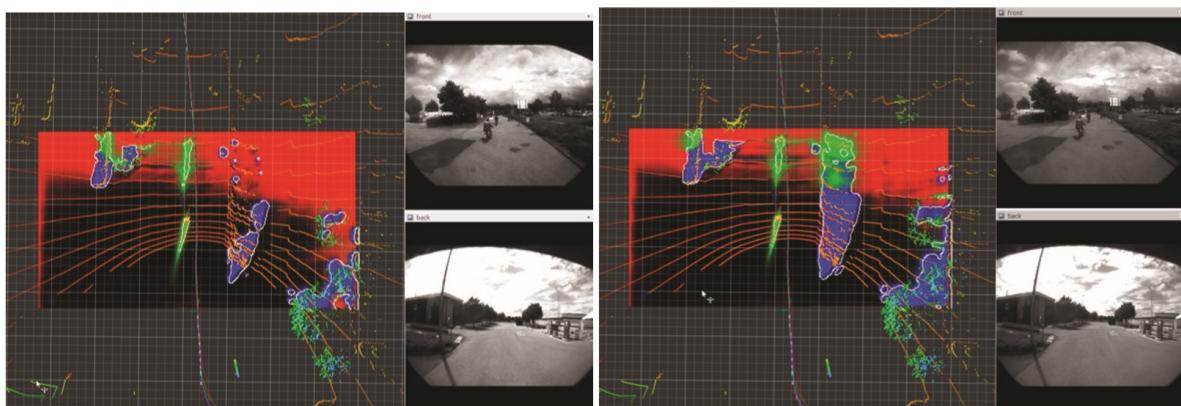


FIGURE 7 INFLUENCE DE BÊTA ET GAMMA (ESTIMATIONS D'ÉLÉVATION) SUR LA SEGMENTATION DU SOL: (GAUCHE) BÊTA =0,5 ET GAMMA =0,1; (DROITE) BÊTA=0,75 ET GAMMA=0,2

Comme expliqué dans la partie précédente (Partie, 3.4) la localisation est fournie par l'algorithme VSLAM fonctionnant à 10 Hz. La carte routière est générée lors de l'étape d'apprentissage de l'algorithme visuel. Voici la carte des bordures générée par l'algorithme présenté en (Partie, 3.4) (Fig, 8).

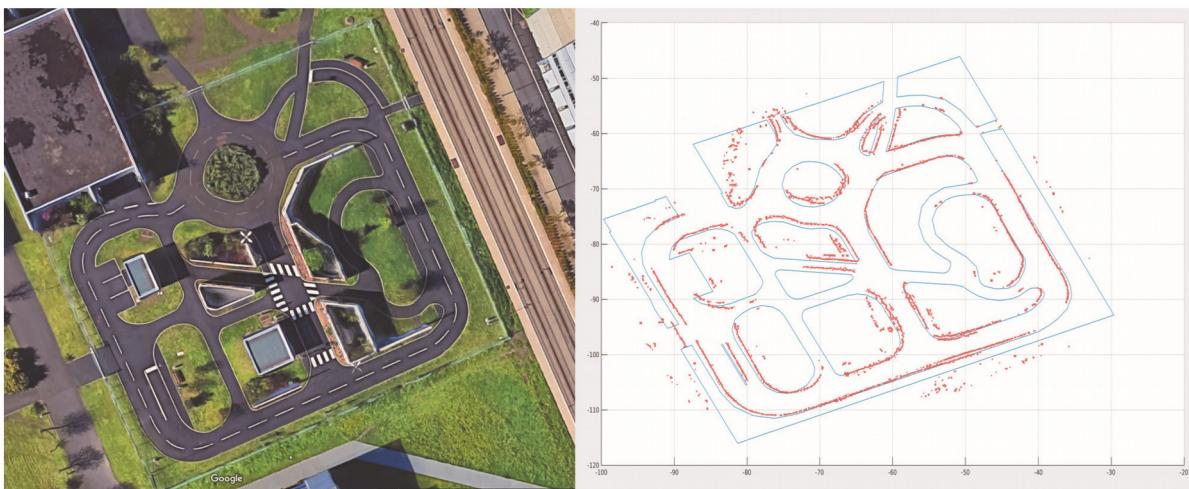


FIGURE 8 COMPARAISON DES RÉSULTATS DE LA DÉTECTION DES BORDURES DE ROUTE (EN ROUGE) AVEC LA VÉRITÉ TERRAIN (EN BLEU)

Une fois que la carte routière est générée, elle peut être utilisée pour définir les limites de la route dans les applications de guidage. La figure (Fig, 9) montre les bordures détectées (rouge), le chemin à suivre, le positionnement courant du véhicule ainsi que les données courantes des télémètres. Il est important de mentionner que les petites erreurs telles que la duplication des bordures de route sont dues au fait que les angles de roulis et de tangage du véhicule n'ont pas été pris en compte pour projeter les bordures de route sur la carte du monde.



FIGURE 9 EZ10 CONDUISANT EN MODE SUIVI DE TRAJECTOIRE AUTONOME SUR PAVIN AVEC CARTE ROUTIÈRE

La figure 10 (Fig.10) montre la fusion du CMCDOT et de la détection des bordures de route qui nourrissent ensemble la carte de PAVIN pour effectuer en guidage autonome l'évitement d'un obstacle statique déposé sur la chaussée. À gauche une vue de l'extérieur du véhicule et à droite une vue de la carte prise sur l'écran de bord avec en rouge les bordures de route et en blanc les obstacles détectés par le CMCDOT.



FIGURE 10 EZ10 SUIVANT UNE TRAJECTOIRE DE RÉFÉRENCE AVEC ÉVITEMENT D'OBSTACLES ET FUSION DES BORDURES DE ROUTE ET DES OBSTACLES SUR LA CARTE DE NAVIGATION

3.6 Conclusion

Cette partie présente une brève discussion des capacités de l'algorithme CMCDOT à l'aide d'un capteur LiDAR3D (Velodyne VLP16). Le CMCDOT est un algorithme efficace pour détecter les obstacles et les classer en obstacles statiques et dynamiques au coût de calcul élevé. Les performances du CMCDOT dans la détection de petits obstacles tels que les bordures de route et autres dépendent fortement de la densité des informations fournies par le système de perception (LiDAR 3D, caméras, etc.). L'utilisation du CMCDOT avec des capteurs LiDAR 3D clairsemés (et de faible densité) tels que le Velodyne VLP16 fournit une bonne cartographie des obstacles et des risques, avec l'inconvénient que les bordures de route et autres petits obstacles sont considérés comme faisant partie du sol dont la géométrie n'est pas modélisée par CMCDOT. Les véhicules autonomes dans les environnements urbains et semi-urbains sont autorisés à circuler sur des routes dont les limites sont souvent marquées de bordures de route et d'autres petits marqueurs de limites de route. Afin de fournir la carte complète des obstacles et des limites de route au véhicule autonome, la détection et la cartographie des bordures de route à l'aide des données LiDAR 3D a permis de pallier ce problème.

4 Les interactions avec l'environnement haut niveau

4.1 L'Unité Embarquée en Véhicule (UEV)

L'Unité Embarquée en Véhicule (UEV) (Fig. 11) est un dispositif qui permet la communication entre véhicules et infrastructure routière. Le matériel utilisé pour le projet est développé et fourni par Neavia Technologies, groupe Lacroix. À travers un lien radio G5 (5.8GHz/5.9GHz), elle est capable d'émettre et de recevoir des messages CAM/DENM. Son IHM embarquée permet de générer des alertes concernant un événement routier (accident par exemple) et d'avertir le conducteur sur les évènements alentours.



FIGURE 11 L'UNITÉ EMBARQUÉE EN VÉHICULE (UEV)

Elle est compacte et s'installe facilement à l'intérieur d'un véhicule. Munie d'un point d'accès wifi et d'un modem 3G, il est possible d'accéder à son IHM d'utilisation et de configuration à l'aide d'une tablette, ainsi que d'accéder à internet. Enfin, elle est pleinement compatible avec les dernières normes ETSI:

- GeoNet: ETSI EN 302 636-4-1
- BTP: ETSI EN 302 636-5-1
- DENM: ETSI EN 302 637-3
- CAM: ETSI EN 302 637-2
- SPAT/MAP: SAE J2735 –ISO 19091 Région D

Dans le cadre du lot 3, l'UEV est intégrée à la navette pour satisfaire deux applications explicitées dans les deux parties suivantes (4.2 et 4.3). Plus généralement, de la même manière qu'une antenne relais GSM, l'UEV est l'interface entre le réseau de supervision de l'exploitant routier et les véhicules/équipements routiers mobiles. Du fait de cette position, elle recueille un grand nombre d'informations provenant de la route, comme des informations sur le trafic et la découverte d'événements inopinés (accidents, congestion...). En conséquence, de nombreuses interfaces sont présentes tant au niveau logiciel qu'au niveau matériel. L'environnement de l'UEV peut être schématisé comme sur la figure 12 (Fig. 12) :

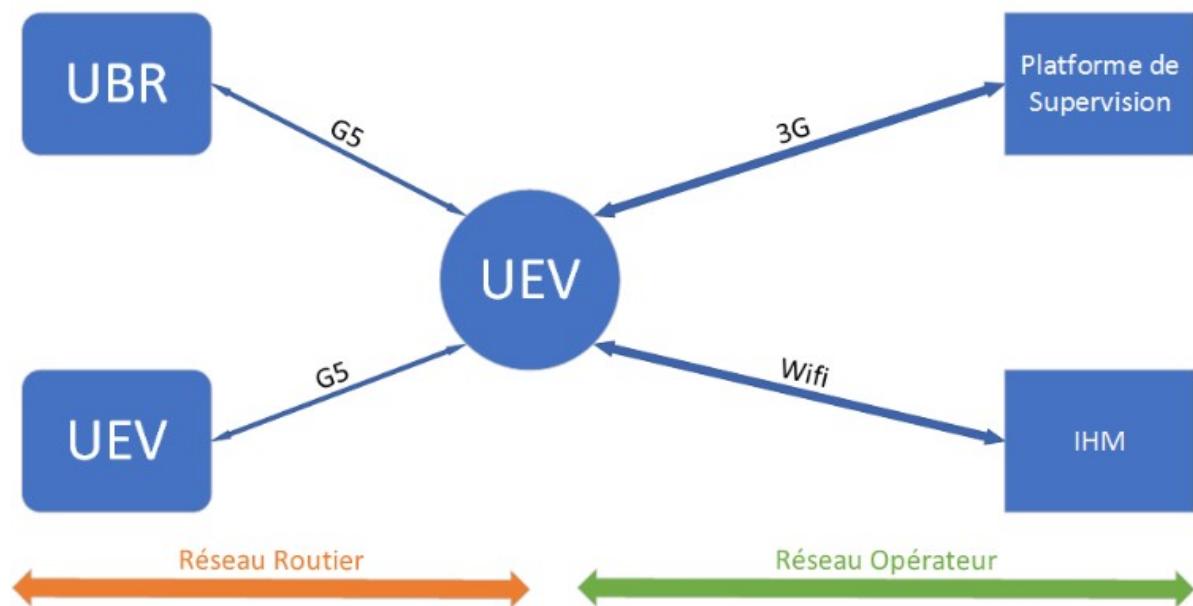


FIGURE 12 L'ENVIRONNEMENT DE L'UEV

L'URB (Unité Bord de Route) permet d'accompagner le déploiement de systèmes de transport intelligents et coopératifs (C-ITS). Dans le cadre du lot 3 l'URB va permettre la communication sécurisée entre l'infrastructure intelligente et la navette via l'UEV. L'URB possède de multiples interfaces disponibles pour l'échange de données. Elles sont :

- Une interface Wifi : cette interface est la plus importante. Elle permet à l'utilisateur de se connecter à l'IHM embarquée dans l'UEV, lancer des alertes et être notifié des événements à proximité.

- Une Interface 3G : elle sert pour l'accès à internet afin de télécharger les tuiles de la cartographie et la connexion à la plateforme du gestionnaire.
- Une interface G5 : elle permet la diffusion et la réception de messages d'alerte (DENM).

Pour comprendre plus précisément la place de l'UEV ou OBU (On Board Unit) dans le service de navette courte distance et pour apporter un complément à la figure 1 de ce rapport voici sur la figure 13 un schéma de l'architecture physique du système TORNADO (Fig. 13).

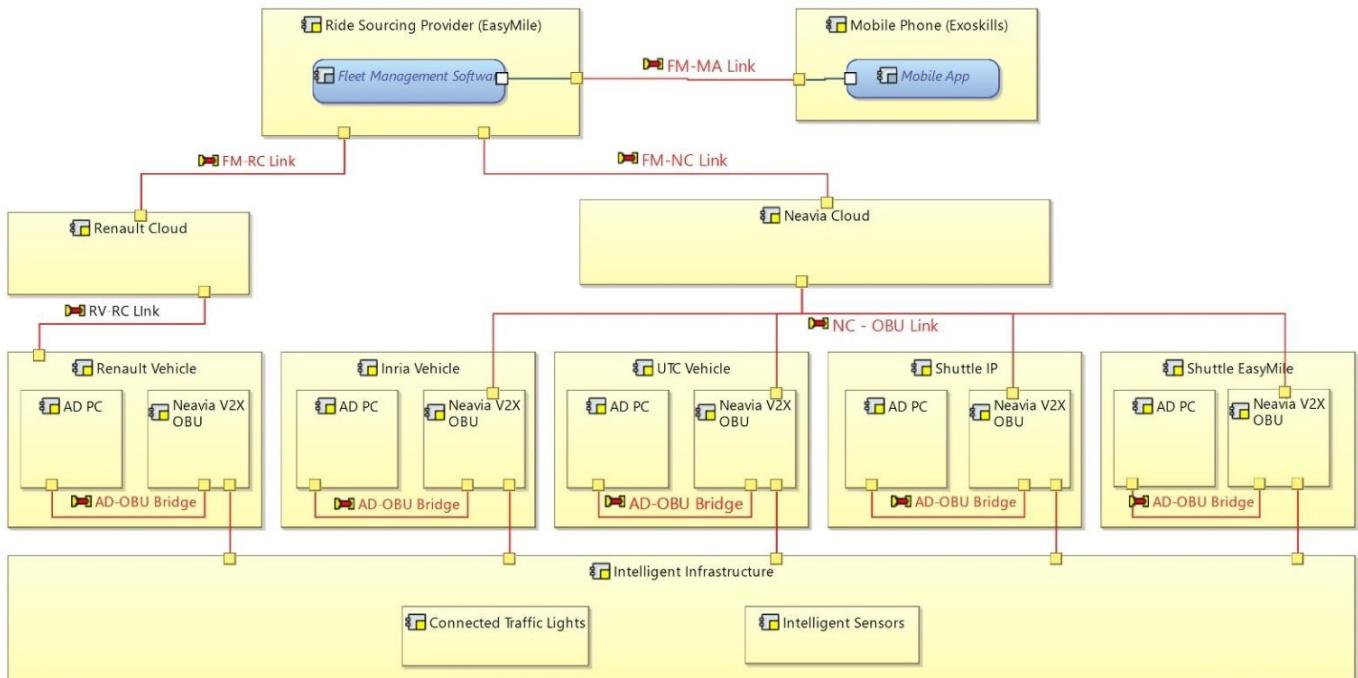


FIGURE 13 TORNADO SYSTÈME : ARCHITECTURE PHYSIQUE

Comme illustré sur cette figure, l'UEV prend position dans les véhicules et est reliée à l'ordinateur de bord par un driver V2X (AD-OBU Bridge). Elle communique avec l'infrastructure intelligente via l'UBR, et au Cloud de Neavia via sa connexion internet (NC-OBU Link). Ensuite une liaison (FM-NC Link) vers la gestion de flotte permet la réception par celle-ci des informations d'état de la navette, et l'envoi des ordres de missions de la gestion de flotte vers la navette. Enfin une liaison (FM-MA Link) permet à l'application mobile de proposer son service aux utilisateurs.

Les spécifications sur l'analyse des besoins, la conception globale et détaillée du système et des sous systèmes, et la conception détaillé de la plateforme d'intégration sont détaillées dans le livrable 1 (L1- Plateforme d'intégration système) orchestré par l'Université de Pau et des Pays de l'Adour (UPPA). Les spécifications sur l'interface cloud et définition de données, le système IHM pour tous les acteurs impliqués et le rapport final du système Cloud sont quant à eux présentés dans le livrable 5 (L5- Monitoring and Cloud Support) orchestré par l'entreprise partenaire Exoskills. Enfin, il faut noter le rapport sur le développement du système de communication I2V / V2I, le rapport sur les algorithmes d'infrastructure intelligents, le rapport sur la compréhension de la scène étendue du véhicule et le rapport final sur l'infrastructure intelligente sur la base de tests expérimentaux dans le livrable 6 (L6- Infrastructures intelligente) orchestré par l'Institut Français des Sciences et

Technologie des Transports, de l'Aménagement et des Réseaux (IFSTTAR : Université Gustave Eiffel) ainsi que l'entreprise partenaire Neavia Technologies, groupe Lacroix.

4.2 Interaction avec la gestion de flotte

4.2.1 Introduction

La gestion de flotte implique que plusieurs interactions sont nécessaires entre la navette et son environnement haut-niveau. L'environnement est composé de deux acteurs, la gestion de flotte EasyMile et les utilisateurs via l'IHM Exoskills (locale à la navette ou déportée). Cet environnement haut-niveau propose des conventions générales pour définir comment évolue la navette. Il définit ce que les utilisateurs peuvent faire avec la navette et propose une interface pour envoyer des ordres et recevoir l'état de la navette. La navette n'ayant pas la possibilité de faire demi-tour en début et fin de voie, elle reste donc orientée de la même manière et change simplement de sens (avant ou arrière) selon les besoins. La navette circule sur une trajectoire qui lui est réservée (Fig. 14), et peut faire un écart latéral ponctuel par rapport à cette trajectoire nominale pour éviter un obstacle. La gestion des portes sera effectuée par un agent présent dans la navette et assurant la sécurité à l'embarquement et pendant les déplacements. Afin de rester uniforme et compatible avec l'API NEAVIA, les informations échangées avec l'API sont également au format json et les données sont encodées en Base64. La lecture et l'écriture des données sont réalisées en utilisant un protocole réseau WebSocket.



FIGURE 14 TRAJECTOIRE DE RÉFÉRENCE ET LISTE DES ARRÊTS

4.2.2 Spécification des scenarii et de la syntaxe des échanges

La navette reçoit donc les ordres de missions depuis la gestion de flotte et les applique si l'état de la navette le permet, sinon ils sont ignorés. La liste définitive des points d'arrêts de la navette est composée des quatre arrêts présentés en figure14 et est partagée entre les acteurs pour pallier le fait que la navette utilise son propre repère local issue du SLAM visuel. La navette ne gère qu'une mission à la fois et deux types de fonctionnement ont été retenus :

- «Métro» : dans ce premier scénario la navette fonctionne comme un tramway et retourne uniquement son état d'avancement pour nourrir la gestion de flotte et l'application mobile sans prendre en compte les missions. Le message est composé du mode de fonctionnement du véhicule (autonome ou manuel), de la vitesse du véhicule et d'un horodatage.
- «GoTo» : le deuxième scénario de démonstration consiste à effectuer une mission nominale : le véhicule reçoit un ordre de mission associé à un identifiant de mission. Ce message reçu par la navette est constitué du nom de l'arrêt à desservir et d'un identifiant de la station. La mission est acceptée lorsque la navette retourne un état correspondant à l'attente d'une mission, une fois la mission acceptée, la navette génère une trajectoire temporaire à suivre jusqu'à la destination et retourne un état correspondant au fait qu'elle est en train de réaliser la mission. Une fois la mission finie cet état est mis à jour en conséquence. À tout moment la mission peut être annulée par la gestion de flotte

L'API gestion de flotte EasyMile utilise la méthode «GoTo» qui est générique, et la navette fera des trajets de type «Métro» mais gère aussi la méthode «GoTo».

Voici la machine d'état associée au mode de fonctionnement «GoTo» (Fig, 15) :

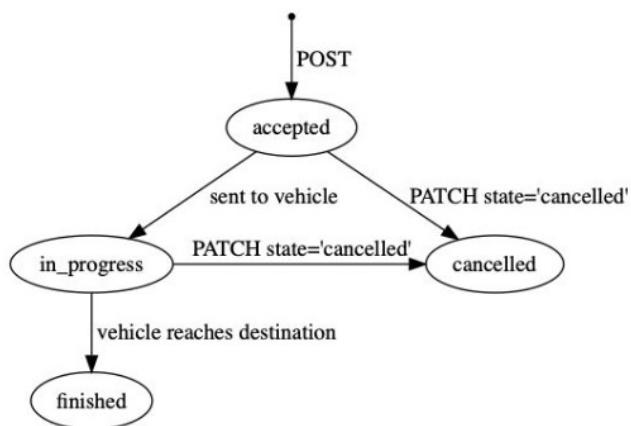


FIGURE 15 MISSION GOTO : MACHINE D'ÉTAT

Les informations supplémentaires nécessaires à l'exécution des missions sont envoyées par le véhicule à l'aide d'un message d'état du véhicule. Ce message est envoyé à une fréquence de 2 Hz. Voici un tableau spécifiant les composantes requises pour l'envoi de l'état du véhicule à la gestion de flotte (Fig, 16), suivi d'un exemple de syntaxe de ce message.

| Field name | Values |
|-----------------------|---|
| type | "vehicle_state" (string) |
| vehicle_mode | "manual" or "autonomous" (string) |
| mission_manager_state | "idle", "processing", "disabled" |
| vehicle_speed | Speed in m/s (double value) |
| mission_status | mission status (null until the execution of the first mission. After executing a mission this value keep the last mission_status of the finished mission with value completed or interrupted) |
| | |

| mission_status | Values |
|----------------|--|
| mission_id | mission id [uuid] (string) |
| state | "pending", "processing", "completed", "interrupted" (string) |
| route | GeoJSON with a polyline describing the trajectory [optional] |
| eta | Expected Time of Arrival to the station [optional] |

FIGURE 16 TABLEAU DES COMPOSANTES REQUISES POUR LE MESSAGE D'ÉTAT DU VÉHICULE

Exemple de syntaxe du message d'état :

```
{
  "type": "vehicle_state",
  "attributes": {
    "vehicle_speed": 5.42,
    "vehicle_mode": "autonomous",
    "mission_manager_state": "on_mission",
    "mission_status": {
      "mission_id": "...mission_id...",
      "eta": ...expected time of arrival in seconds...
      "route": ...GeoJSON Object...
      "state": "processing"
    }
  }
}
```

Ensuite l'envoi de mission de la gestion de flotte par le véhicule est spécifié par les composantes suivantes (Fig. 17) :

| Field name | Values |
|--------------|----------------------------|
| type | "mission" (string) |
| mission_id | mission id [uuid] (string) |
| station_id | Station ID (integer) |
| station_name | Station name (string) |

FIGURE 17 TABLEAU DES COMPOSANTES REQUISES POUR LE MESSAGE D'ENVOI DES MISSIONS

Exemple de syntaxe du message d'envoi des missions :

```
{
  "type": "mission",
  "mission_id": "mission_id",
  "attributes": {
    "to": {
      "station_name": "...name of the station...",
      "station_id": ...id of the station...
    },
  },
}
```

Enfin, le message d'annulation d'une mission est composé comme ceci (Fig. 18) :

| Field name | Values |
|------------|----------------------------|
| mission_id | mission id [uuid] (string) |
| type | "cancel_mission" (string) |

FIGURE 18 TABLEAU DES COMPOSANTES REQUISES POUR LE MESSAGE D'ANNULATION DES MISSIONS

Exemple de syntaxe du message d'annulation de mission :

```
{
  "type": "cancel_mission",
  "mission_id": "mission_id",
}
```

4.2.3 Développement interne à la navette

Afin de gérer l'accomplissement des missions, des développements internes à la navette ont été réalisés. Ces développements permettent à la navette, après réception de la mission, de générer un chemin temporaire jusqu'à la destination. Deux cas ont été implémentés, le premier cas repose sur le fait que la trajectoire de référence est connue dans un repère WGS84 et le deuxième cas repose sur le fait que la trajectoire, cette fois, n'est pas connue en WGS84. Les deux cas sont donc dépendants de la présence ou non d'un GPS pendant la phase d'apprentissage.

Cas numéro 1 (Fig. 19) : la trajectoire de référence est connue en WGS84

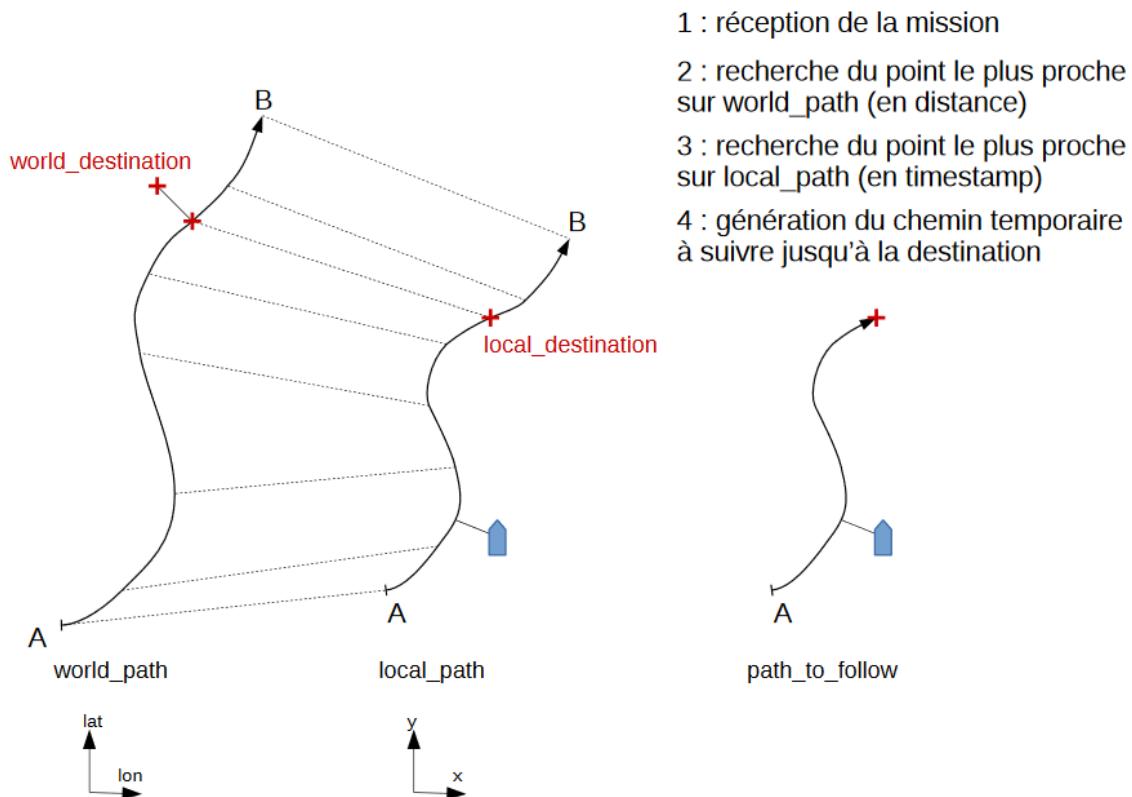


FIGURE 19 CAS NUMÉRO 1 : LA TRAJECTOIRE DE RÉFÉRENCE EST CONNUE EN WGS84

Cas numéro 2 (Fig. 20) : la trajectoire de référence n'est pas connue en WGS84

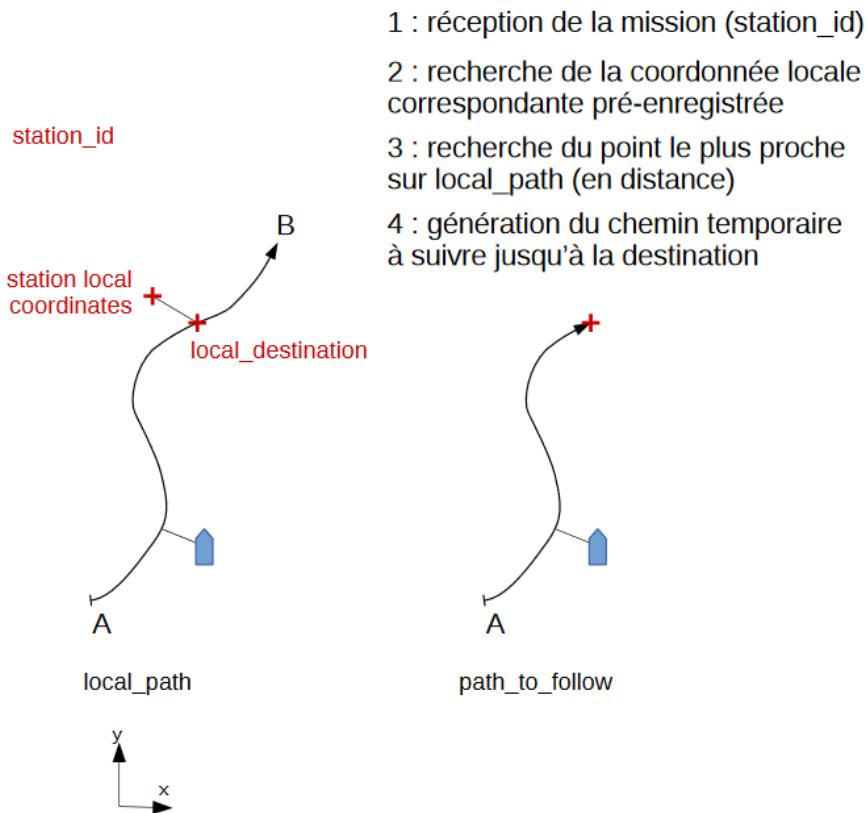


FIGURE 20 CAS NUMÉRO 2 : LA TRAJECTOIRE DE RÉFÉRENCE N'EST PAS CONNU EN WGS84

Pour faciliter les développements et pour être indépendant de la gestion de flotte haut niveau pendant les phases de tests, une interface graphique a été développée à l'Institut Pascal pour envoyer des ordres de missions sans l'UEV. Cette interface permet donc de simuler l'envoi et l'annulation de missions et se présente sous cette forme (Fig. 21) :

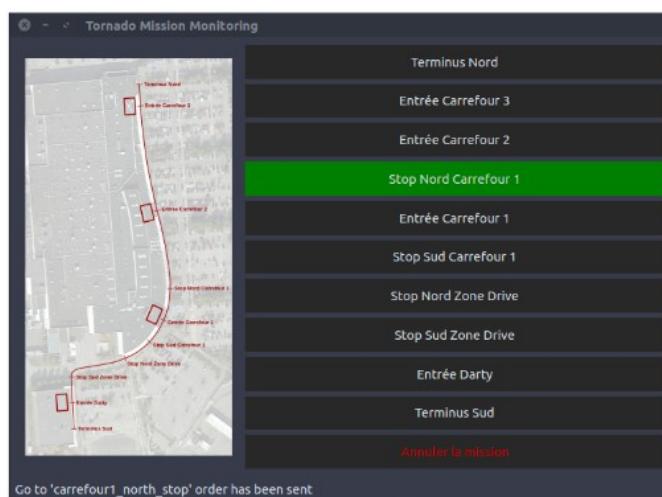


FIGURE 21 INTERFACE GRAPHIQUE POUR LE MONITORING DE MISSIONS INDÉPENDAMMENT DE LA GESTION DE FLOTTE

Afin de mieux comprendre l'organisation des développements liés à la gestion des missions un graphe complet composé des différents packages ROS intégrés à la navette et développés autour du SLAM visuel est présenté en figure 22 (Fig, 22) :

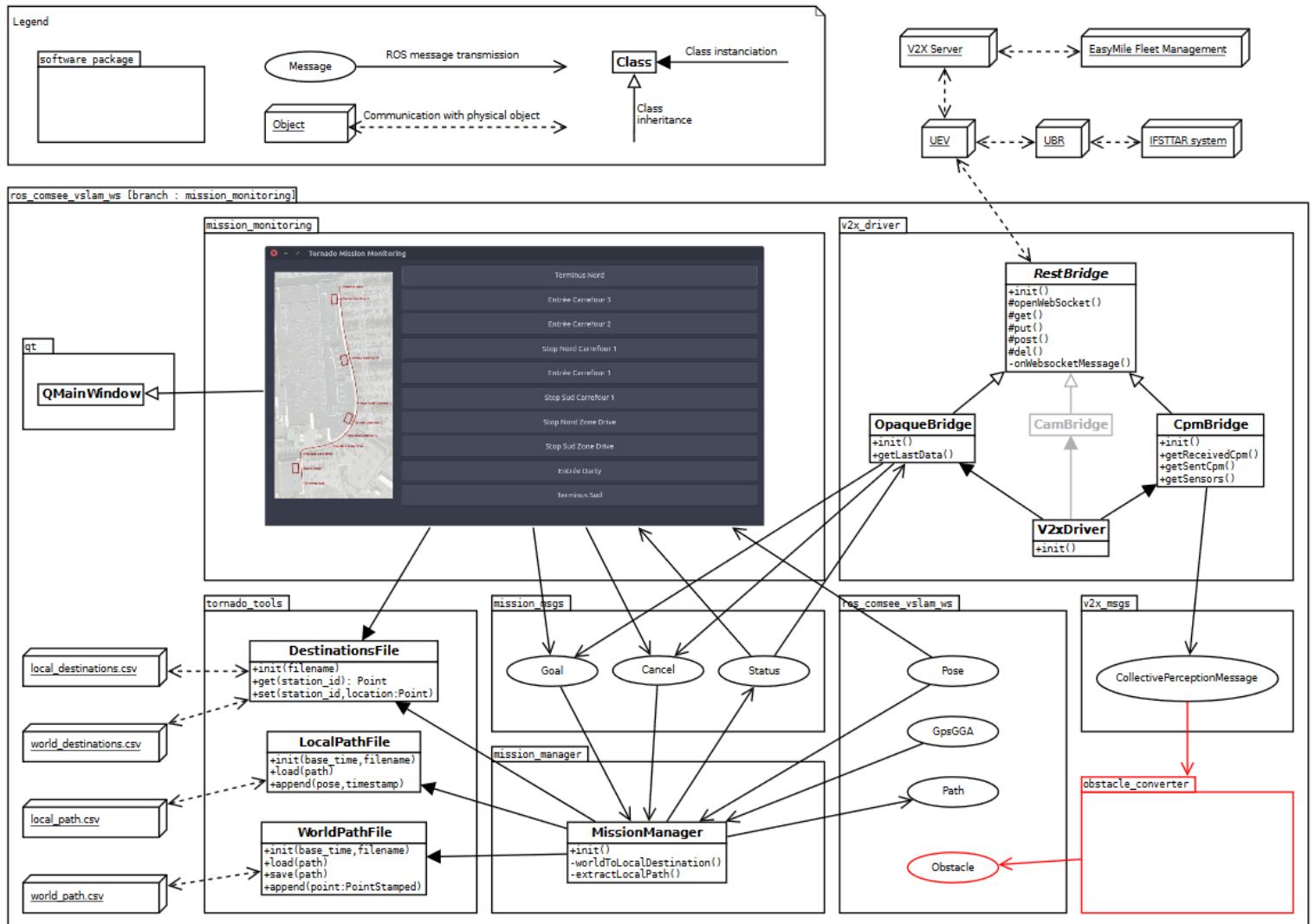


FIGURE 22 GRAPHE D'ORGANISATION DES DIFFÉRENTS PACAGES ROS DÉVELOPPÉS AUTOUR DU SLAM VISUEL POUR LA GESTION DES MISSIONS

Comme expliqué précédemment, la gestion des missions est réalisable de deux manières, premièrement via le «v2x_driver» qui reçoit les ordres de mission depuis la gestion de flotte haut niveau par l'intermédiaire de l'UEV. Deuxièmement, la gestion des missions est aussi proposée localement à la navette par l'intermédiaire de l'interface graphique présentée en figure 21 «mission_monitoring».

Ensuite comme le montrent les figures 19 et 20 la planification de trajectoire peut être effectuée de deux manières, avec ou sans référencement GPS «tornado_tools». La génération des trajectoires à suivre pour accomplir une mission est orchestrée par le «mission_manager» qui permet de faire le lien entre, d'un côté les messages reçus par la gestion de flotte «Goal et Cancel», la pose du véhicule ou la position GPS issue du VSLAM et de l'autre, le chemin à suivre «Path» qui est transmis au SLAM visuel et le status «Status» de la mission qui est retourné à la gestion de flotte.

4.3 Interaction avec l'infrastructure intelligente

4.3.1 Introduction

Comme illustré en figure 1, 13 et 22, une deuxième interaction avec l'environnement haut niveau est mise en place dans le service de navette courte distance. Cette interaction est une brique supplémentaire permettant via une communication à distance de faciliter la circulation des véhicules autonomes et de sécuriser leur navigation. Cette brique est proposée et pilotée par IFSTTAR et NEAVIA et correspond au lot 6 «Infrastructure intelligente». Dans le cadre du service de navette courte distance, cette infrastructure a pour objectif de retourner au véhicule des informations (pose, vitesse, caractérisation) sur les obstacles présents (piéton, voiture, deux-roues, camion). Composée d'une caméra, d'un PC et d'un module de communication, cette infrastructure utilise pour inférer un algorithme de détection d'objets par deep-learning. Ces informations sont reçues par le véhicule par le biais du matériel fourni par NEAVIA présenté dans la partie 4.1 et envoyées par l'infrastructure à l'UEV via une Unité Bord de Route (UBR). Dans le cadre du lot 3, l'infrastructure intelligente est déployée sur le toit du centre commercial et la caméra est orientée de manière à observer la zone drive de Carrefour qui est sur le passage de la navette. Cette zone est une zone à risque, car contrairement aux autres zones de passage de la navette elle est ouverte à la circulation du public et nécessite donc une sécurité supplémentaire. La figure 23 (Fig. 23) est un zoom de la figure 14 sur la zone drive correspondant, avec la position et l'orientation de la caméra.

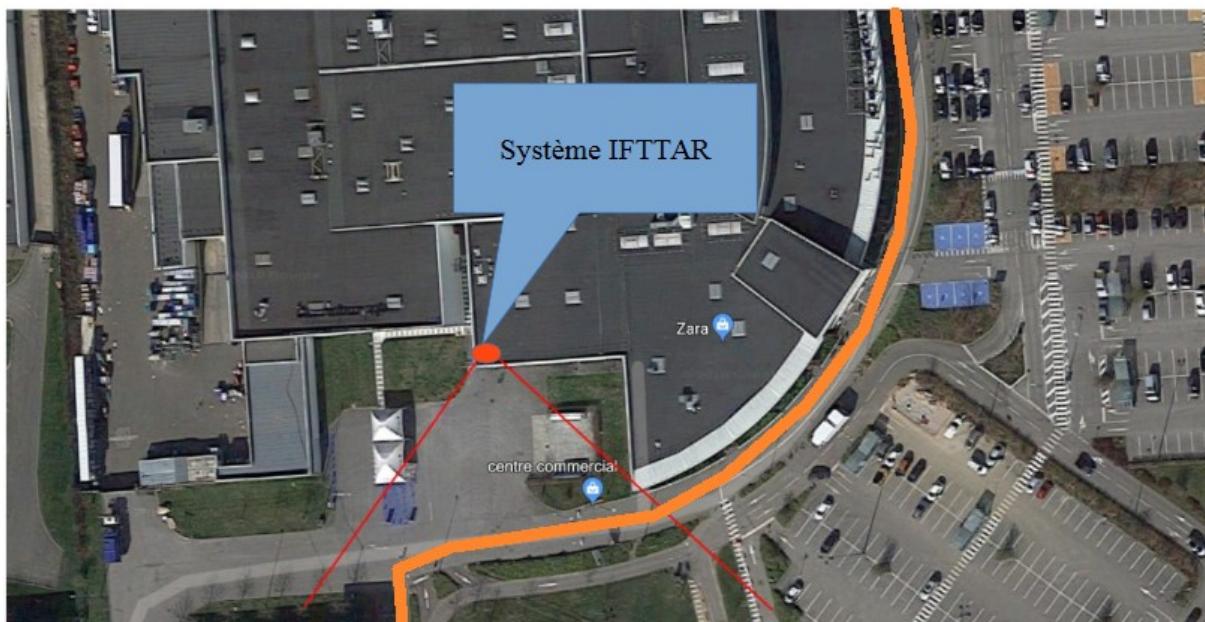


FIGURE 23 ZOOM SUR LA ZONE DRIVE : POSITION ET ORIENTATION DE LA CAMÉRA DU SYSTÈME D'ANALYSE INFRASTRUCTURE DE L'IFSTTAR

4.3.2 Spécifications

Comme illustré en figure 22, les messages transmis par l'infrastructure intelligente sont connus sous le nom de Message de Perception Collective (CPM). Étant donné le caractère réactif requis, les messages sont directement envoyés de l'UBR à l'UEV sans passer par le Cloud de NEAVIA comme illustré en figure 13 et 22.

L'objet perçu par l'algorithme et envoyé au véhicule est composé d'un horodatage, du centre de l'objet dans le repère du capteur, d'un vecteur vitesse, d'une dimension maximale de l'objet exprimé à partir du centre (rayon de la sphère englobante de l'objet) ainsi que de la classification de celui-ci. Plusieurs classes peuvent être détectées par l'algorithme :

- inconnue
- piéton
- cycliste
- vélo-moteur
- moto
- voiture
- bus
- camion léger
- camion lourd
- caravane
- véhicule spécial
- tram

Les positions des obstacles détectés dans cette zone sont exprimées dans un repère commun plan, orthonormé et métrique dont les axes X et Y sont définis sur les limites blanc/gris des marquages au sol suivants (Fig. 24) :

L3.2 |Rapport pour le système complet courte distance



FIGURE 24 DÉFINITION DU REPÈRE COMMUN

5 Résultats pour l'ensemble du système

5.1 Introduction

Dans cette partie, nous présentons les résultats obtenus pour l'ensemble du système. Une vidéo présentant les résultats est disponible sur le site internet du projet en suivant ce lien (<https://www.tornado-mobility.com/index.php/rediffusion-des-presentations-de-la-journee-du-5-novembre-2020/#InstitutPascal>). Dans un premier temps, nous aborderons un à un les développements et tests nécessaires au fonctionnement du système : géoréférencement, guidage, configurations matérielles et logicielles des capteurs utiles au CMCDOT et aux bordures de routes, configurations logicielles et matérielles utiles au fonctionnement de la partie gestion des missions et communication avec l'infrastructure intelligente. Ensuite, nous parlerons des résultats et tests obtenus avec le système complet en situation réelle. À noter que les résultats obtenus avec le système d'arrêt d'urgence développé par Easymile (virtual bumper) sont disponibles dans le livrable «R3.3 – Rapport «virtual bumper» système».

5.2 Résultats du géoréférencement de la carte

L'utilisation du GPS pour la navette a permis dans un premier temps d'obtenir un fond de carte aligné avec la trajectoire du véhicule. L'idée est de produire une transformation de repère UTM vers le repère issu du SLAM visuel afin d'afficher une photo aérienne derrière la trajectoire. Cela a été réalisé de deux manières différentes, soit à l'aide d'une optimisation par les moindres carrés entre les positions fournies par le GPS et celles fournies par le SLAM afin d'obtenir la transformation de repère, soit avec un géoréférencement de la trajectoire SLAM directement dans le repère UTM. Avec des données GPS de bonne qualité le positionnement UTM du véhicule est correct.

Le géoréférencement de la carte SLAM a aussi permis dans un second temps de récupérer les obstacles fournis par l'infrastructure intelligente dans un repère commun. Quand un obstacle est détecté par le dispositif de l'IFSTTAR, on reçoit ses coordonnées dans le repère du dispositif. Comme notre repère "map" est centré au même endroit et orienté de la même manière, les obstacles détectés sont directement dans notre repère. La figure (Fig, 25) montre la navette arrêtée à l'approche d'un obstacle reçu via le dispositif de l'IFSTTAR :

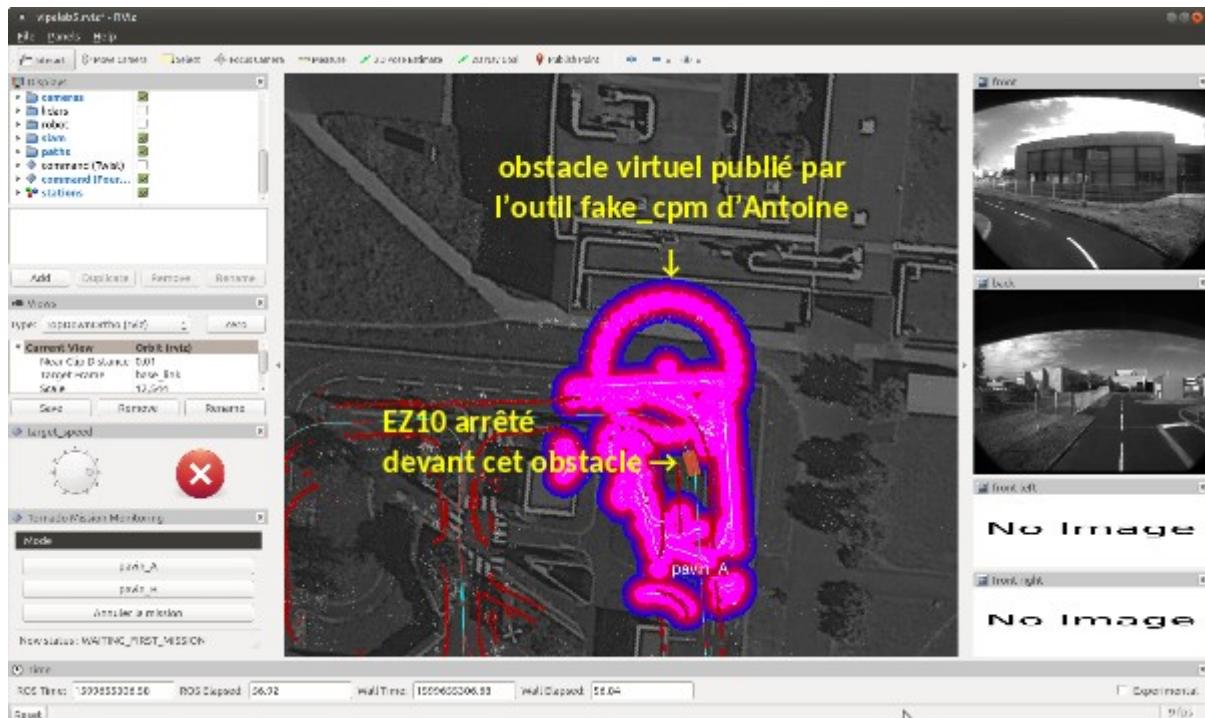


FIGURE 25 RÉSULTAT DU GÉORÉFÉRENCEMENT DE LA CARTE SLAM POUR LA DÉTECTION D'OBSTACLES PAR LE DISPOSITIF IFSTTAR

Malheureusement, cette partie n'a pas pu être testée en situation réelle à Rambouillet à cause des conditions sanitaires qui ont perturbé les phases d'expérimentation d'où l'utilisation de l'outil "fake_cpm" cité sur la figure 25.

5.3 Résultats pour le guidage de la navette et Lidars plans

Afin d'obtenir un guidage efficace, aussi bien d'un point de vue interface utilisateur (consigne de vitesse en marche avant et marche arrière, consigne de vitesse nulle pour arrêter le véhicule) que d'un point de vue précision dans le suivi de trajectoire (guidage avec deux trains directeurs, changement de repère entre la caméra et le repère du véhicule, limiteur de vitesse lorsqu'un obstacle est détecté sur le couloir de navigation) de nombreux tests et développements ont été réalisés afin d'obtenir des résultats à la hauteur des espérances du projet. L'objectif était d'atteindre une souplesse et une fluidité dans la navigation pouvant convenir pour un public large et inexpérimenté en la matière.

Au lancement des tests de guidage, de nombreux points n'étaient pas fonctionnels, le comportement du véhicule notamment en marche arrière en commande deux trains directeurs était inopérant avec le middleware ROS. Les tests de guidage ont été réalisés à deux vitesses : une appelée "lente" (1.7 m/s) et une appelée rapide (3.4 m/s) qui est une vitesse d'exploitation qui est considérée par le plus grand nombre comme agréable et rassurante. Après plusieurs modifications dans l'algorithme de guidage, notamment sur la sélection des images clés utiles au SLAM visuel, nous avons obtenu des trajectoires exploitables. Afin de tester la qualité du suivi, nous avons choisi d'utiliser un DGPS comme "vérité-terrain", ce test nous a permis de repérer des erreurs statiques et "d'overshoot" qui ont pu être réglées par modification des offsets de direction et la recalibration des changements de repères du SLAM visuel et rectification de l'estimation des informations de position et de vitesse du véhicule. L'image (Fig. 26) montre la fermeture de boucle après

exploitation de la partie haute de la plateforme d'expérimentation PAVIN des véhicules de l'Institut Pascal.

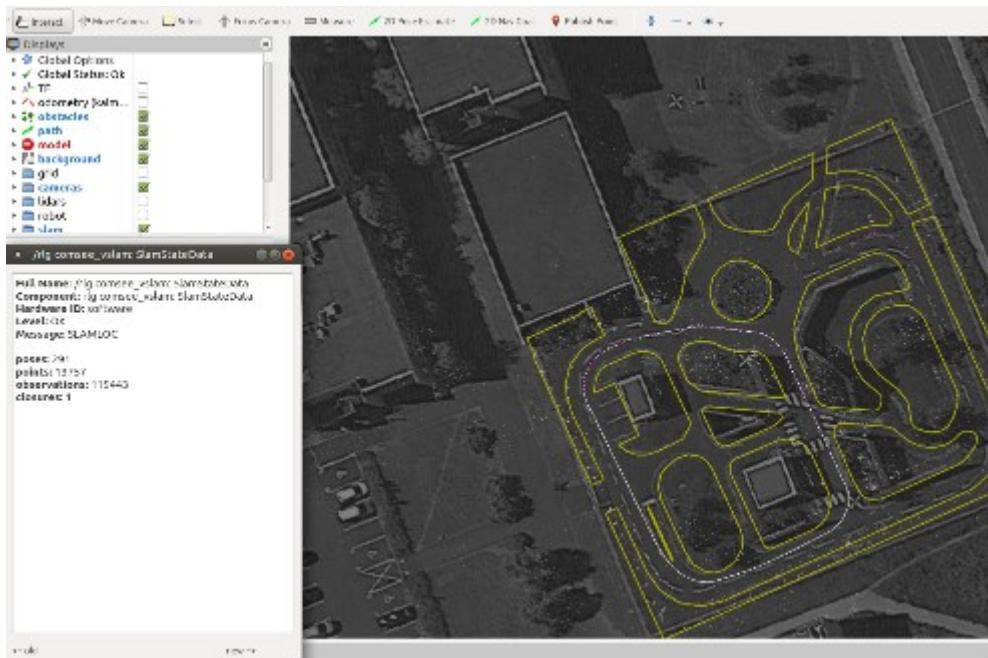


FIGURE 26 TRAJECTOIRE PAR FERMETURE DE BOUCLE APRÈS MODIFICATION DU CODE DE GUIDAGE

D'autres développements ont permis de rendre robuste la détection des obstacles par les Lidars plans qui est soumise à une forte sensibilité aux conditions de ruptures de pentes. Pour générer les obstacles à partir des données Lidar, un nœud de conversion est inséré dans la chaîne de traitement des données. Ce nœud permet également de filtrer les scans Lidar en portée et en ouverture. L'image (Fig. 27) illustre les résultats obtenus par ce filtrage :

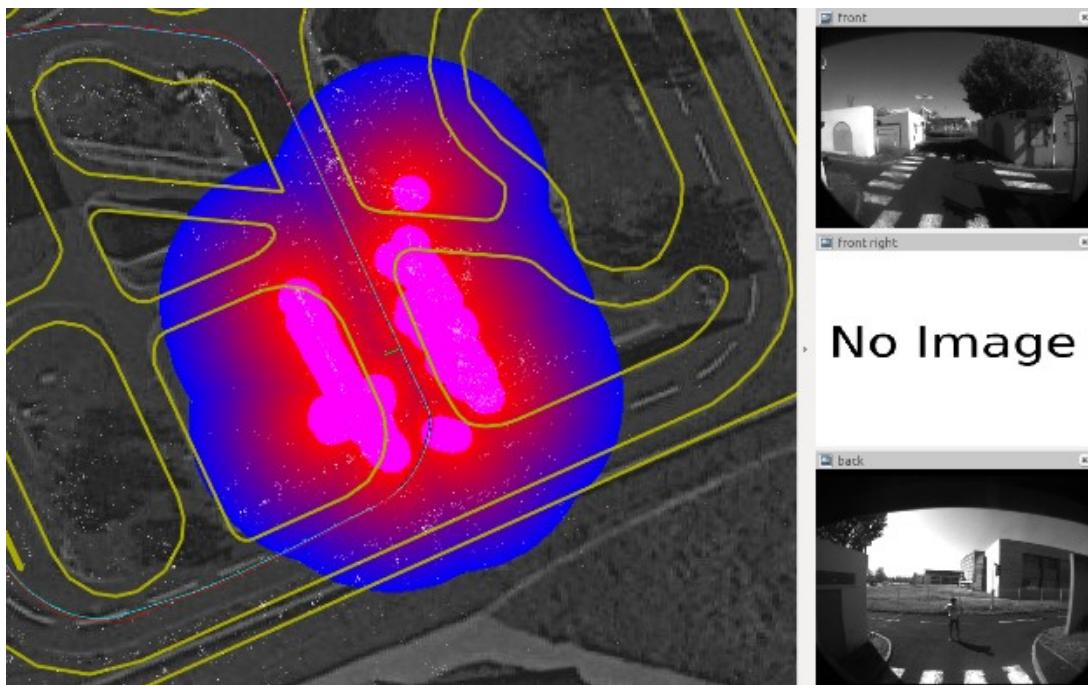


FIGURE 27 VISUALISATION DES OBSTACLES DÉTECTÉS PAR LES LIDARS PLANS

Avec une limitation de la portée de détection à 6 mètres les problèmes de sensibilité de dévers ont été résolus et le véhicule s'arrête correctement.

Afin de limiter le nombre d'éléments à démarrer, un plugin visuel a été développé pour gérer les consignes de vitesses (Fig, 28) :

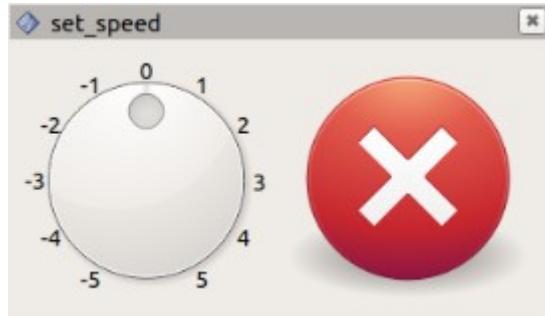


FIGURE 28 INTERFACE DE GESTION DES VITESSES

Cette nouvelle interface fonctionne correctement, elle est plus esthétique et simplifie le démarrage de l'application. Elle envoie une séquence de messages pour que la consigne suive une rampe sur une seconde.

5.4 Résultats associés aux télémètres

Dans cette partie, nous allons présenter les résultats obtenus à l'aide des capteurs Lidars Velodyne avant et arrière installés sur le toit du véhicule et nourrissant l'algorithme de l'INRIA (CMCDOT). Des supports capteurs, sur le toit ont été installés afin d'accueillir les Velodyne. Après connexion à l'alimentation et au réseau, les capteurs sont opérationnels. La figure (Fig, 29) montre l'installation des capteurs :

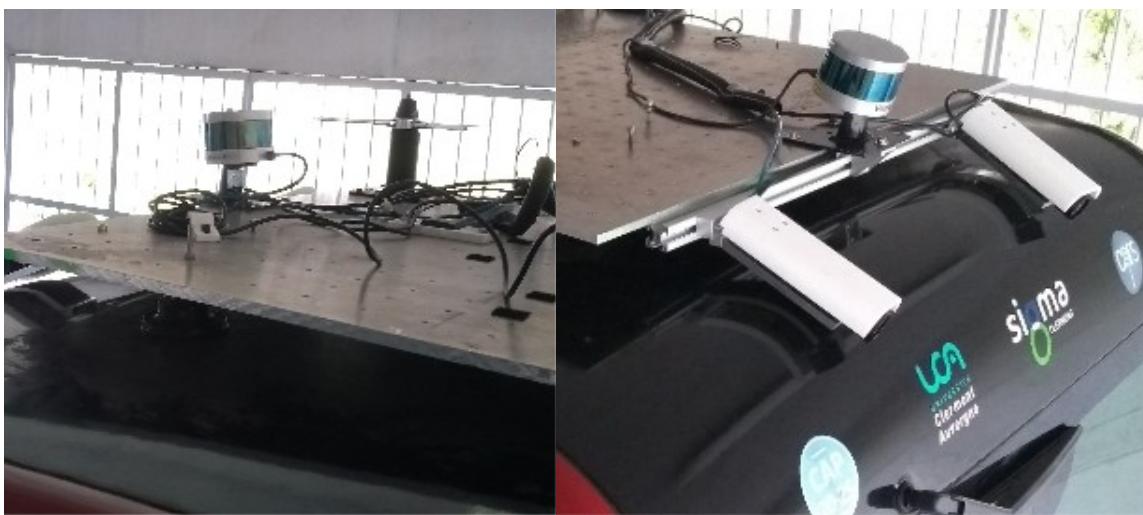


FIGURE 29 INSTALLATION DES VELODYNE SUR LE TOIT DU VÉHICULE

Dans un second temps, une vérification et une configuration du débit réseau a permis d'assurer la bonne transition des données. La configuration réseau est présentée sur le schéma ci-dessous (Fig, 30) :

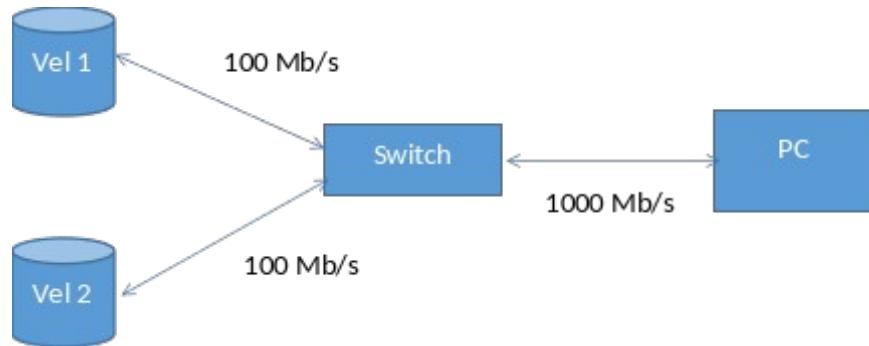


FIGURE 30 CONFIGURATION RÉSEAU BI-VELODYNE

Afin d'éviter de polluer la bande passante, la configuration des Velodyne est faite de manière à envoyer les données sur deux ports différents. Avec cette configuration, nous obtenons un système opérationnel, nous pouvons ainsi observer les données provenant des deux capteurs sur le logiciel de visualisation RVIZ.

Ensuite, nous avons réglé les paramètres extrinsèques des Lidars. Idéalement, cela devrait être réalisé par un outil d'alignement automatique. Comme cet outil n'était pas disponible sur étagère, nous avons reproduit la juxtaposition des nuages de points afin d'estimer les changements de repères. Pour ce faire, nous avons développé un plugin RVIZ afin d'observer en direct les corrections apportées et d'aligner visuellement les nuages de points à des repères dans le monde réel. Ici, nous avons utilisé des plots, les murs, et le sol afin de bien positionner les données par rapport au véhicule. La figure (Fig.31) illustre ce réglage et le développement réalisé :

L3.2 |Rapport pour le système complet courte distance

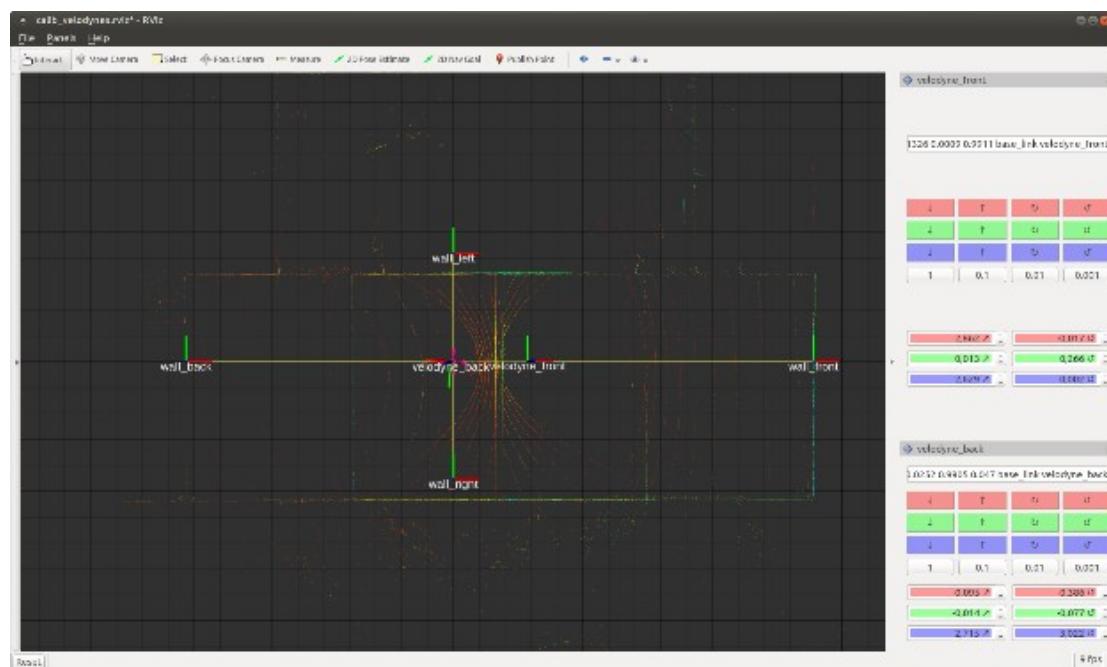


FIGURE 31 RÉGLAGE DES PARAMÈTRES EXTRINSÈQUES DES VELODYNE

Ensuite, nous avons eu besoin de filtrer les données Velodyne pour éviter la détection d'obstacles sur le toit du véhicule. Ce filtre a permis d'ignorer les obstacles CMCDOT dans un rectangle autour du véhicule. Ce filtre prend place comme illustré sur la figure (Fig. 32) :

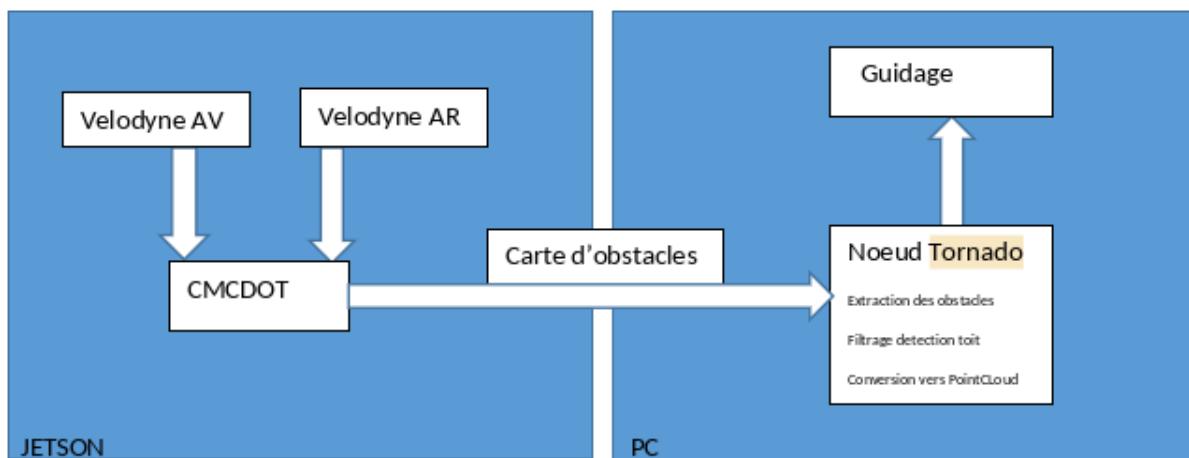


FIGURE 32 ARCHITECTURE DE TRAITEMENT ET FILTRAGE DES DONNÉES VELODYNE

Après une bonne configuration des distances d'arrêt, le comportement du véhicule à l'approche d'obstacles détectés par les Velodyne est satisfaisant. Couplé avec la détection des Lidars plans, le véhicule couvre une grande variété de situations. Des problèmes de détection du sol illustrés (Fig. 33) ont aussi pu être résolus grâce à une meilleure calibration des capteurs et à l'ajustement des paramètres offerts par le CMCDOT. La gestion avant arrière des détections se fait correctement grâce au repère proposé par le CMCDOT et à la possibilité par celui-ci de prendre en compte plusieurs capteurs.

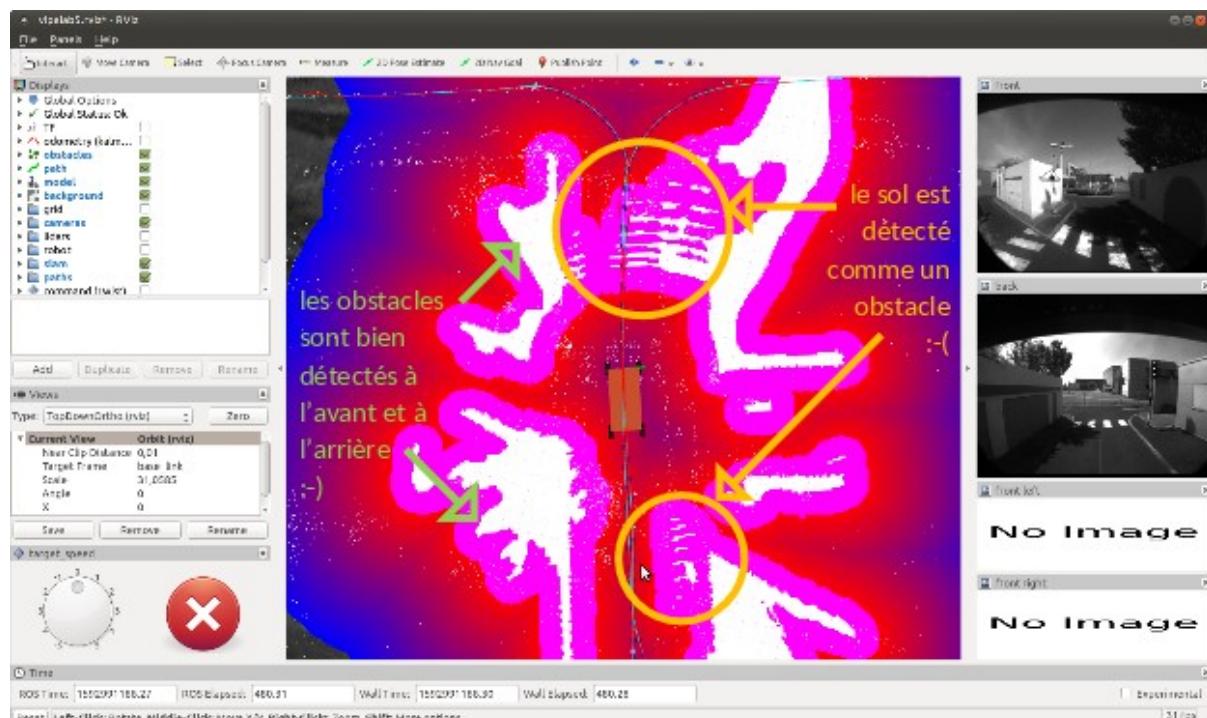


FIGURE 33 ILLUSTRATION DES PROBLÈMES DE DÉTECTION DU SOL

Pour préciser le problème de distance d'arrêt sur obstacle, voici (Fig. 34), des vues RVIZ montrant les distances d'arrêts différentes pour des tests à une vitesse de 2 m/s. À noter que pour des vitesses très lentes, la distance d'arrêt était identique. Le CMCDOT apporte donc la possibilité de détecter les obstacles plus longue distance avec un ralentissement progressif.

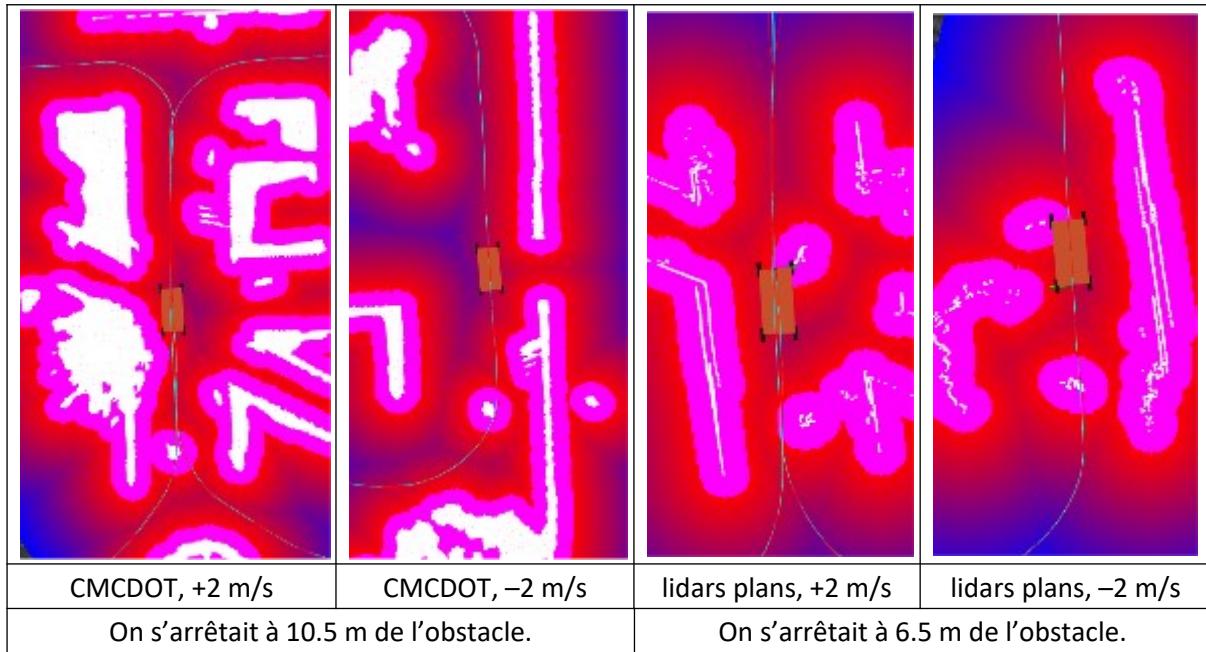


FIGURE 34 ILLUSTRATION DU PROBLÈME DE DISTANCE D'ARRÊT

5.5 Résultats pour la gestion des missions

Dans cette partie, nous parlons des résultats obtenus par l'intégration et l'utilisation de l'Unité Embarquée en Véhicule (UEV) pour la gestion de flotte. Ce matériel de communication et les développements associés pour obtenir le fonctionnement souhaité pour le projet sont explicités en détail dans la partie 4 du livrable, cela comprend les deux interactions hauts niveau, la gestion de flotte et l'interaction avec l'infrastructure intelligente. Les résultats obtenus avec l'infrastructure intelligente sont en partie présentés dans la partie 5.3 de ce livrable. Ici, nous traiterons plus précisément les résultats obtenus pour la gestion de flotte.

Dans un premier temps, le matériel de communication (UEV) a été intégré sur le toit de la navette dans un boîtier étanche et l'antenne permettant la communication est placée au centre du toit comme le montre la figure (Fig. 35) :



FIGURE 35 MODULE D'ANTENNES DE L'UEV SUR LE TOIT DE LA NAVETTE

La sélection des missions est effectuée à l'aide du mission manager présenté dans la partie 4.2. Quand une mission est reçue par le véhicule, le chemin est généré dans l'algorithme de

navigation et le véhicule démarre automatiquement soit en marche avant soit en marche arrière en fonction de la mission en question. À la fin de la mission, lorsque le véhicule a atteint le point d'arrivée, il s'arrête tout seul de manière souple, l'utilisateur peut ainsi sortir du véhicule.

L'interface graphique, présente sur l'écran d'affichage à l'intérieur de la navette, permet d'afficher la mission courante et sélectionner une autre mission sans obligatoirement passer par l'UEV. Cette interface est présentée dans la figure (Fig. 36), les boutons sont dans un panneau (en bas à droite dans l'exemple ci-dessous) et la position des arrêts est affichée dans la vue RVIZ (au centre) :



FIGURE 36 INTERFACE GRAPHIQUE INTERACTIVE PRÉSENTE DANS LA NAVETTE

5.6 Conclusion sur les résultats obtenus avec le système complet

Pour conclure, le système complet permet la navigation de la navette suivant une logique de desserte d'arrêts reçus soit par le biais de la gestion de flotte haut niveau soit par l'écran d'affichage de la navette. Une fois la mission reçue, la navigation est faite en sécurité grâce à trois couches de sécurités, le CMCDOT qui permet une visibilité des obstacles lointaine couplé aux Lidars plan qui eux utilisent soit le système de sécurité natif d'arrêt d'urgence soit le virtual bumper (livrable 3.3) qui permet aussi des arrêts d'urgence plus complexes qui prennent en compte la vitesse du véhicule et le braquage des roues. Additionnellement, le véhicule obtient des informations supplémentaires de position et classification des obstacles détectés par l'infrastructure intelligente dans une zone considérée à risque.

Des tests utilisant l'entièreté du système sur la place Vasarely du campus des Cézeaux (très régulièrement) à Clermont-Ferrand, et Rambouillet ont été réalisés (montée de la navette à deux reprise sur Rambouillet). Tests allants jusqu'à 5 m/s. Bien que la présentation de cette démonstration n'ait pas pu être faite comme prévu à un public large, les retours des personnes ayant utilisé ce système sont très positifs. Comme le montrent les vidéos faites pour le projet, la navette se comporte très bien et permet une navigation sûre et agréable.

REFERENCES

- [1] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–151, 1988.
- [2] B. Thuilot, J. Bom, F. Marmoiton, and P. Martinet. Accurate automatic guidance of an urban vehicle relying on a kinematic gps sensor. In Symposium on Intelligent Autonomous Vehicles IAV04, 2004.
- [3] P. Lébraly, C. Deymier, O. Ait-Aider, E. Royer, and M. Dhome. Flexible extrinsic calibration of non overlapping cameras using a planar mirror : Application to vision-based robotics. In IROS, October 2010.
- [4] P. Lébraly, E. Royer, O. Ait-Aider, C. Deymier, and M. Dhome. Fast calibration of embedded non overlapping cameras. In ICRA, 2011.
- [5] Eric Royer, Francois Marmoiton, Serge Alizon, Datta Ramadasan, Morgan Slade, Ange Nizard, Michel Dhome, Benoit Thuilot and Florent Bonjean. Retour d'expérience après plus de 1000 Km en navette sans conducteur guidée par vision. Journées transports intelligents, RFIA, 2016.
- [6] L. Rummelhard, A. Negre, A. Paigwar, and C. Laugier. Ground Estimation and Point Cloud Segmentation using SpatioTemporal Conditional RAndom Field. IEEE Intelligent Vehicles Symposium (IV), pp. 1105–1110, June 2017.
- [7] L. Rummelhard, A. Negre, and C. Laugier, Conditional monte carlo dense occupancy tracker. IEEE Intl. Conf. on Intelligent Transportation Systems, Sept. 2015.