

Conduite Automatique à Basse Vitesse. Architecture du Système de Voiturier Autonome

Mihai CHIRCA^{1,2,3}, Roland CHAPUIS¹, Christophe DEBAIN², Roland LENAIN², Guillaume MARTIN³

¹Institut Pascal - UMR 6602 CNRS / UBP / IFMA
Campus des Cézeaux, 24 Avenue des Landais, BP 80026, 63171 Aubière Cedex, France

²IRSTEA - Institut National de Recherche en Sciences et technologies pour l'Environnement et l'Agriculture
Campus des Cézeaux, 24 Avenue des Landais, 63171 Aubière Cedex, France

³Renault - Technocentre - Dep. ADAS, Véhicule Autonome et Technologies Electroniques
1 Avenue du Golf, 78084 Guyancourt, France

{mihai.chirca, guillaume.g.martin}@renault.com, roland.chapuis@univ-bpclermont.fr,
{christophe.cebain, roland.lenain}@irstea.fr

Résumé – Cet article présente le développement d'un système de voiturier autonome que Renault développe en partenariat avec des laboratoires de recherche dans le but d'en faire un produit à large échelle. Au cours du développement, un véhicule Renault Zoe a été équipé des capteurs, des actionneurs électromécaniques et des calculateurs de traitement de données. Le système de perception correspond à une fusion entre douze capteurs ultrasons et une caméra et fournit la localisation, la zone navigable autour du véhicule et une carte de l'environnement. Ces informations sont utilisées par la suite par le système de contrôle pour permettre au véhicule de se déplacer en automatique depuis la zone de dépose jusqu'à la place de parking ou depuis le parking jusqu'à la zone de reprise. Dans la suite, on va décrire l'architecture hardware et software sans entrer dans les détails des algorithmes.

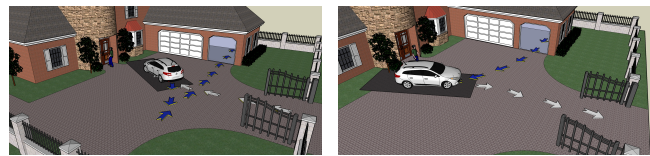
Abstract – This paper presents the development of an autonomous valet parking system that Renault is actively developing in partnership with research laboratories in order to make it a common product. During development, a common Zoe vehicle was equipped with sensors, electromechanical actuators and a system of data processing. The perception system corresponds to a fusion between an ultrasonic system and a camera. It provides the localization of the vehicle, the navigable area around the car and a map of already visited areas. This information is used afterwards as input for the car control system and to drive the from the drop-off zone to the parking and from the parking to the pick-up zone. In the following it will be described the hardware and the software system architecture, without detailing all algorithms.

1 Contexte général

Ce projet veut répondre au problème suivant: comment concevoir une voiture autonome en mesure d'effectuer une manœuvre de parking tout en tenant compte de la diversité des environnements? Le but est de réaliser un système à bas coût plus efficace que ceux actuellement commercialisés, tout en utilisant des capteurs de série ou de pré-série.

Pour que le véhicule sache où se garer, l'utilisateur effectue une seule fois la manœuvre en manuel. Cette phase on va l'appeler à la suite *étape d'apprentissage*. À la fin de ce processus, la voiture sera capable de se garer, en toute autonomie, dans un environnement connu.

Le projet est basé sur les recherches antérieures réalisées par l'équipe *PerSyst* de l'Institut Pascal, en partenariat avec IRSTEA et Renault. L'équipe de Renault fournit les connaissances pratiques sur les systèmes d'aide à la



(a) Conduite autonome vers la place de parking (b) Conduite autonome vers la zone de reprise du véhicule par le conducteur

Figure 1: Manœuvres de parking en automatique (flèches bleues - conduite automatique; flèches blanches - conduite manuelle)

conduite ainsi qu'un véhicule d'essai instrumenté et équipé d'actionneurs électromécaniques. La fusion de données se réalise avec l'outil de prototypage rapide d'application robotiques Effidence, qui intègrent facilement des données multicapteurs (une caméra grand angle, un system de capteurs à ultrasons, quatre capteurs odométriques, un

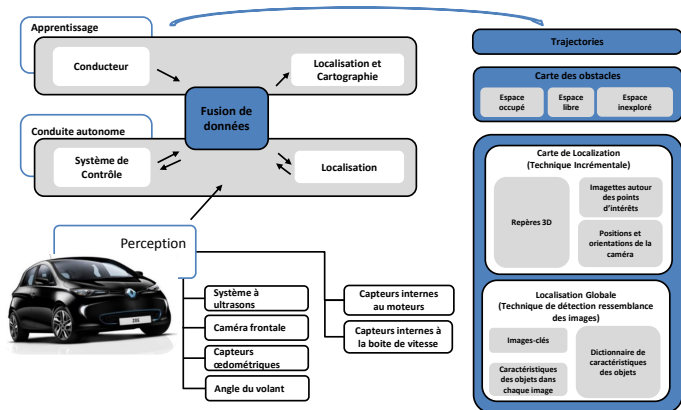


Figure 2: Architecture software du système

GPS et un GPS-RTK). La commande bas niveau et la connexion aux calculateurs du véhicule se fait par un système MicroAutoBox de DSpace. Les tests ont été effectués sur ce véhicule Zoe ainsi que sur le simulateur 4D-Virtualiz qui intègre une représentation graphique réaliste 3D des objets dans leur environnement.

Pour la transmission de données, plusieurs réseaux CAN sont utilisés (Figure 3). Étant donné que ceux-ci ne sont pas adaptés à la transmission de données volumineuses, comme des images, entre la caméra et le système de fusion a été utilisé une connexion Giga Ethernet. Dans la Figure 3 on retrouve l'architecture hardware du système.

2 Le système de perception

Dans les sections suivantes, on présente le système de perception. L'approche est d'exposer l'architecture et les solutions techniques nécessaires pour atteindre l'objectif prévu, sans détailler tous les algorithmes et les propriétés techniques du hardware utilisés.

2.1 La perception utilisant un système de capteurs à ultrasons

La véhicule doit évoluer en toute sécurité dans un environnement a priori connu avec peu d'obstacles dynamiques. Pour cela, il devra fonctionner indépendamment du résultat du système de localisation. Par conséquent, un système de 12 capteurs à ultrasons a été choisi pour permettre la détection en temps réel des obstacles autour du véhicule.

En terme de performance, le système détecte les obstacles situés entre 0.2 m et 3.5 m. Les ondes ultrasoniques sont émises par un seul émetteur et les échos sont reçus par plusieurs récepteurs. Cette réception multicapteurs permet la mise en œuvre d'un algorithme de triangulation pour estimer la position 2D des objets. Néanmoins, cette solution a des inconvénients en termes de temps de latence parce qu'elle oblige l'utilisation d'un seul émetteur à la fois

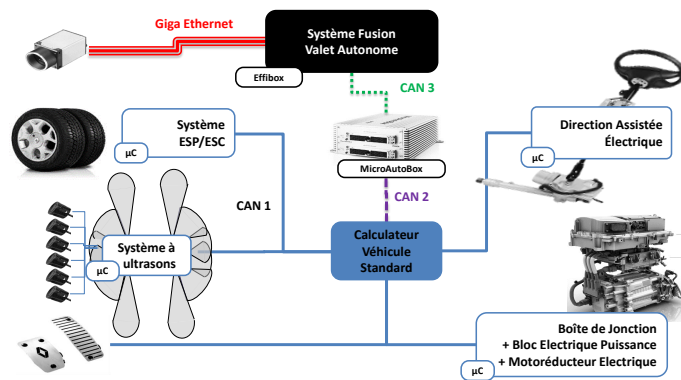


Figure 3: Architecture hardware du système

pour que la source des échos reste connue.

Les données envoyées par ce système sont sauvegardées utilisant une structure arborescente de quadtree [1]. Les quadtrees sont utilisés pour partitionner un espace à 2D par la subdivision récursive d'une région en quatre sous-régions égales. Cette structure de données permet de construire dynamiquement la carte d'obstacles, de l'enrichir facilement dans toutes les directions et aussi de modifier sa résolution. Un autre grand avantage est le fait que les zones inexplorées n'utilisent pas de mémoire de stockage.

Tout au long du développement, les tests en conditions réelles ont permis de mettre en évidence des différentes situations de fausses détections. Dans la plupart des cas, elles correspondent à des échos transmis par le sol. Pour éviter ce phénomène, un système de filtrage a été implémenté. Dans la Figure 4 on observe une carte d'obstacles réalisée dans les allées d'un parking souterrain.

La longueur maximale du trajet qu'on accepte que la voiture réalise en automatique est de 500 m. Dans ce test, la position de la voiture que le système a calculée a une erreur de moins de 5 degrés et 0,25 m si on utilise uniquement les odomètres qui ont une précision de 48 top par tour de roue. La qualité des résultats diminue, avec la baisse du coefficient de frottement en cas de conduite sur du gravier ou de ciment poli. Mais ce problème sera éliminé (ou fortement réduit) par la fusion avec les algorithmes de traitement vidéo.

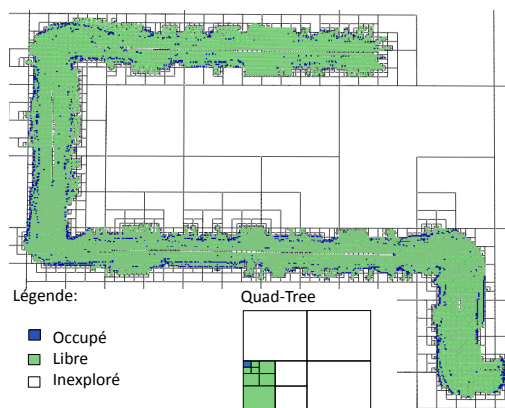


Figure 4: Carte d'obstacles

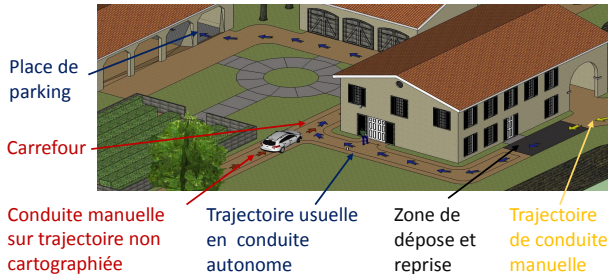


Figure 5: Cas de multiples trajectoires - possibilité d'activation du système depuis une position non apprise

2.2 La perception par vision

La fusion de données entre la caméra et les capteurs à ultrasons est très importante pour assurer la fiabilité du système dans les conditions réelles. Pour répondre au problème de localisation, une fusion des résultats de trois algorithmes de vision est réalisée. Le premier algorithme fournit une localisation incrémentale en utilisant une carte de repères (points 3D) et des images correspondant à ces repères observés depuis une position connue. Le deuxième algorithme a pour but de réaliser une localisation globale du véhicule en se basant sur l'appariement de l'image courante parmi des images clés. Le dernier algorithme réalise la liaison entre les deux algorithmes.

2.2.1 Cartographie et localisation incrémentale

Pour la localisation de la voiture, une technique SLAM visuel (localisation et cartographie simultanée) basée sur un filtre EKF (Extended Kalman Filtre) est utilisée.

Il a fallu choisir entre une technique d'ajustement de faisceaux et celle de SLAM EKF. Il est bien connu que le premier procédé est plus précis grâce à une minimisation globale, mais il nécessite beaucoup de puissance de calcul. Le SLAM EKF est plus adapté à la fusion des données bruitées et au fonctionnement temps réel et les résultats la précision est suffisante tenant compte de la longueur de la trajectoire maximale qu'on impose.

Dans le processus d'apprentissage, une carte est créée dans le but d'une localisation ultérieure. Dans chaque image, des points d'intérêt sont acquis en utilisant des opérateurs Shi et Tomasi. Pendant le déplacement du véhicule, le suivi de ces points, permet le calcul de leur position 3D dans le repère global. Quand un niveau de confiance est atteint, les points 3D sont enregistrés dans la carte. En absence d'une référence absolue, le repère global est fixé au début de la trajectoire.

Dans le processus de relocalisation, la position de la voiture est déterminée et certifiée une fois que les nouveaux points d'intérêt sont mis en correspondance à ceux déjà cartographiés. Pour le même point de départ, l'algorithme donne des très bonnes performances. La difficulté est de localiser la voiture dans le même repère global en cas de départ d'une position initiale incertaine. Mais à

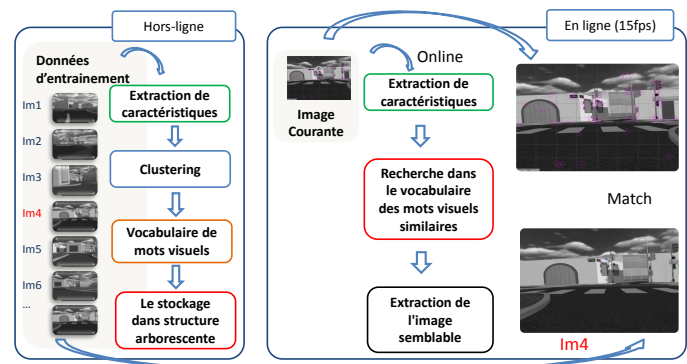


Figure 6: Localisation globale

cette question répond l'algorithme de localisation globale.

2.2.2 Localisation globale

Tel que présenté, les limites de l'algorithme précédent doivent être couvertes par cette méthode de localisation globale. Dans la Figure 5, on observe un cas d'utilisation où le conducteur engage le système de parking autonome depuis une position non apprise.

Si la manœuvre de stationnement commence depuis une position inconnue ou incertaine, le système ne devra pas permettre le démarrage en automatique. Cependant, le système peut suggérer à l'utilisateur de continuer en mode manuel jusqu'à ce que sa position soit détectée.

Au cours de la phase d'apprentissage, on sauvegarde régulièrement des images de l'environnement. La localisation globale est basée sur un algorithme du type FAB-MAP [2]. Ceci met en correspondance permanente l'image courante avec les images clés précédemment enregistrées au long de la trajectoire. Déterminer la paire de correspondance revient à trouver une position connue.

Cette méthode repose sur le concept de "BoW" (Bag of Words), qui est une approche développée par la communauté de traitement d'images et a ses sources dans le travail de J. Sivic et de A. Zisserman [2]. Pour chaque image clé une extraction de caractéristiques uniques est faite et toutes ces données sont réunies dans une structure nommée "dictionnaire" (voir Figure 6). Les limites de cette localisation globale sont atteintes pour les environnements symétriques ou similaires. Cette approche fournira plusieurs correspondances parce que, naturellement, les images sont similaires. Suite aux essais, on a considéré que la distance minimale entre deux images clés est de 0,5 m.

2.2.3 Estimation de la pose relative

Pour simplifier la transition entre la localisation globale et celle progressive (EKF SLAM) et pour restreindre l'espace de recherche, une information très importante est le cap du véhicule. Pour cela, les images fournies par la localisation globale vont être analysées par cet algorithme afin

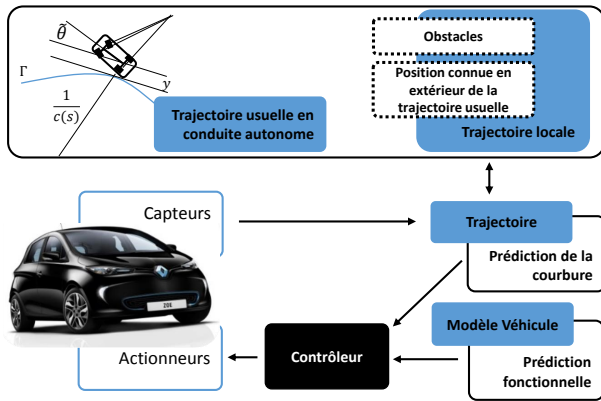


Figure 7: Contrôle et suivi de trajectoire

de détecter la pose relative. Pour faire cela, un algorithme de calcul de la matrice essentielle est utilisé [3]. Dans chaque image, les points d'intérêt sont déterminés et un appariement est réalisé en utilisant une technique RANSAC [4]. Cette technique permet de trouver et d'éliminer les mises en correspondances aberrantes entre les points d'intérêts qui vont servir dans le calcul de la pose relative.

3 Le contrôle du véhicule

En parallèle des travaux sur les algorithmes SLAM, un travail d'adaptation des lois de contrôle [5] a été réalisé dans le but de commander le véhicule au long d'une trajectoire.

La connaissance des caractéristiques du système de perception a guidé l'élaboration de l'algorithme de commande. Ce projet utilise la caractérisation des actionneurs (direction assistée électrique, système de freinage, moteur électrique, etc.) disponibles dans le laboratoire d'excellence IMOB3 et directement réalisée sur la voiture Zoe.

Les algorithmes de commande sont basés sur des techniques de contrôle non-linéaire, adaptatif et prédictif. La stratégie de perception nécessite le transfert du contrôle du robot depuis la référence globale à une référence locale. La difficulté du problème de contrôle est la faible manœuvrabilité du véhicule à basse vitesse. En effet, il a été démontré que la stabilisation théorique des véhicules non-holonomes autour d'une configuration fixe n'est pas possible [6]. Cependant, nous disposons de plusieurs moyens de contrôler ces systèmes comme l'approche de suivi de trajectoire [7], généralement utilisée pour se bons résultats. En outre, les approches adaptatives et prédictives sont capables de maintenir un degré élevé de précision et sont en mesure d'intégrer une anticipations des collisions, par l'évaluation de la zone traversable.

L'objectif est de créer une carte de l'environnement suffisamment précise pour assurer l'intégrité du positionnement du véhicule dans la carte et de générer une nouvelle trajectoire permettant l'évitement des obstacles [8].

Conclusions

Ce système de voiturier autonome est dans un état avancé de développement, mais pas terminé. La fusion de données entre la caméra avant et les capteurs à ultrasons donne de bons résultats et cela nous rassure que les approches mentionnées mèneront à l'accomplissement de nos objectifs.

L'utilisateur acceptera ce système uniquement si on arrive à prouver sa fiabilité à long terme. Pour cela il devrait fonctionner pendant différents conditions d'illumination, devrait ne pas être perturbé par les objets en mouvement et également il devrait être capable de fonctionner dans des espaces avec peu marge de manœuvre.

Le sujet n'est pas clos parce que la dernière étape sera de commercialiser le produit à grande échelle. Malheureusement cela n'est pas pour demain. Cependant la fusion de l'expérience des instituts de recherche avec celle des industriels, augmente les chances de répondre avec succès à toutes ces questions.

References

- [1] H. Samet, "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Surveys*, vol. 16, no. 2, pp. 187–260, 1984.
- [2] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.
- [3] J. Sivic and a. Zisserman, "A text retrieval approach to object matching in videos," in *Proc. CVPR*. IEEE, pp. 1470–1477.
- [4] F.-X. Chen, "Fast RANSAC with Preview Model Parameters Evaluation," *Journal of Software*, vol. 16, no. 8, p. 1431, 2005.
- [5] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "Mixed kinematic and dynamic sideslip angle observer for accurate control of fast off-road mobile robots," *Journal of Field Robotics*, vol. 27, no. 2, pp. 181–196, 2010.
- [6] R. W. Brockett, R. W. Brockett, Others, and Others, "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory*, vol. 27, pp. 181–191, 1983.
- [7] F. Pourboghrat, "Exponential stabilization of non-holonomic mobile robots," *Computers and Electrical Engineering*, vol. 28, no. 5, pp. 349–359, 2002.
- [8] C. Debain, P. Delmas, R. Lenain, and R. Chapuis, "Integrity of an autonomous agricultural vehicle according the definition of trajectory traversability," in *AgEng 2010, International ...*, 2010.