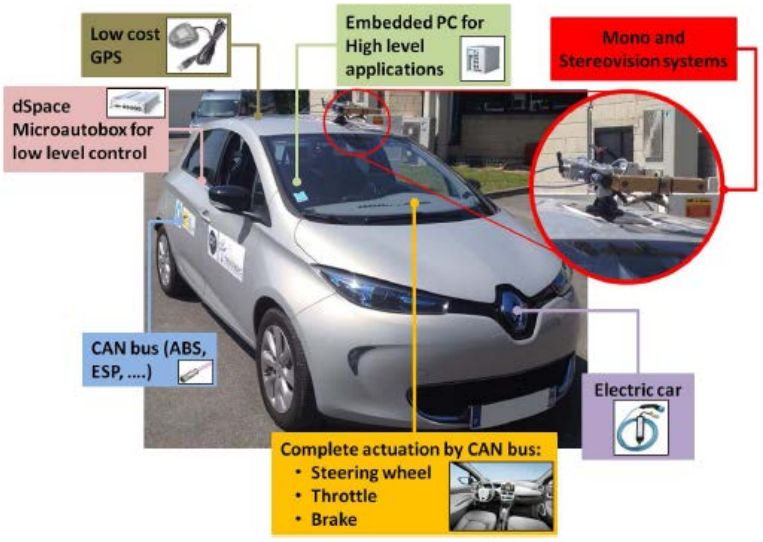
Sensor-based navigation applied to intelligent electric vehicles

Danilo Alves de Lima

Par **Danilo ALVES DE LIMA**

Sensor-based navigation applied to intelligent electric vehicles

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



Soutenue le 17 juin 2015
Spécialité : Information Technologies and Systems

D2191

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

THESIS

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Area of specialization: Information Technologies and Systems

by

Danilo ALVES DE LIMA

Sensor-based navigation applied to intelligent electric vehicles

Heudiasyc laboratory, UMR UTC/CNRS 6599

Defended on June 17th, 2015.
Thesis Committee:

Reviewers:	Patrick Rives	INRIA Sophia Antipolis
	Philippe Martinet	Ecole Centrale de Nantes-CNRS
Examiners:	Mustapha Mouaddib	Univ. de Picardie Jules Verne
	Philippe Bonnifait	Univ. de Technologie de Compiègne
	Said Mammar	Univ. d'Evry Val-d'Essonne
Supervisor:	Alessandro Corrêa Victorino	Univ. de Technologie de Compiègne

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

THÈSE

Pour obtenir le grade de

Docteur

Spécialité : Technologie de l'Information et de Systèmes (TIS)

par

Danilo ALVES DE LIMA

Navigation référencée capteurs appliquée aux véhicules électriques intelligents

Laboratoire Heudiasyc, Unité Mixte de Recherche UTC/CNRS 6599

Soutenue le 17 juin, 2015 devant le jury constitué de :
Thesis Committee :

Rapporteurs :	Patrick Rives	INRIA Sophia Antipolis
	Philippe Martinet	Ecole Centrale de Nantes-CNRS
Examinateurs :	Mustapha Mouaddib	Univ. de Picardie Jules Verne
	Philippe Bonnifait	Univ. de Technologie de Compiègne
	Said Mammar	Univ. d'Evry Val-d'Essonne
Directeur de Thèse :	Alessandro Corrêa Victorino	Univ. de Technologie de Compiègne

Acknowledgments

This work was carried out and funded in the framework of the Equipex ROBOTEX (Reference ANR-10-EQPX-44-01). It was equally supported by the French Picardie project VERVE, French Government, through the program “Investments for the future" managed by the National Agency for Research, and the European Fund of Regional Development FEDER.

Abstract

Autonomous navigation of car-like robots is a large domain with several techniques and applications working in cooperation. It ranges from low-level control to global navigation, passing by environment perception, robot localization, and many others in a sensor-based approach. Although there are very advanced works, they still presenting problems and limitations related to the environment where the car is inserted and the sensors used. This work addresses the navigation problem of car-like robots based on low cost sensors in urban environments. For this purpose, an intelligent electric vehicle was equipped with vision cameras and other sensors to be applied in three big areas of robot navigation: the Environment Perception, Local Navigation Control, and Global Navigation Management. In the environment perception, a 2D and 3D image processing approach was proposed to segment the road area and detect the obstacles. This segmentation approach also provides some image features to local navigation control. Based on the previous detected information, a hybrid control approach for vision based navigation with obstacle avoidance was applied to road lane following. It is composed by the validation of a Visual Servoing methodology (deliberative controller) in a new Image-based Dynamic Window Approach (reactive controller). To assure the car's global navigation, we proposed the association of the data from digital maps in order to manage the local navigation at critical points, like road intersections. Experiments in a challenging scenario with both simulated and real experimental car show the viability of the proposed methodology.

Keywords: Car-like robot navigation; Road segmentation; Visual Servoing; Dynamic Window Approach; Global navigation management.

Résumé

La navigation autonome des voitures robotisées est un domaine largement étudié avec plusieurs techniques et applications dans une démarche coopérative. Elle intègre du contrôle de bas niveau jusqu'à la navigation globale, en passant par la perception de l'environnement, localisation du robot, et autres aspects dans une approche référencée capteurs. Bien qu'il existe des travaux très avancés, ils présentent encore des problèmes et limitations liés aux capteurs utilisés et à l'environnement où la voiture est insérée. Ce travail aborde le problème de navigation des voitures robotisées en utilisant des capteurs à faible coût dans des milieux urbains. Dans cette thèse, nous avons traité le problème concernant le développement d'un système global de navigation autonome référencée capteur appliqué à un véhicule électrique intelligent, équipé avec des caméras et d'autres capteurs. La problématique traitée se décline en trois grands domaines de la navigation robotique : la perception de l'environnement, le contrôle de la navigation locale et la gestion de la navigation globale. Dans la perception de l'environnement, une approche de traitement d'image 2D et 3D a été proposé pour la segmentation de la route et des obstacles. Cette méthode est appliquée pour extraire aussi des caractéristiques visuelles, associées au milieu de la route, pour le contrôle de la navigation locale du véhicule. Avec les données perçues, une nouvelle méthode hybride de navigation référencée capteur et d'évitement d'obstacle a été appliquée pour le suivi de la route. Cette méthode est caractérisée par la validation d'une stratégie d'asservissement visuel (contrôleur délibératif) dans une nouvelle formulation de la méthode "fenêtre dynamique référencée image" (Dynamic Window Approach - DWA, en anglais) (contrôleur réactif). Pour assurer la navigation globale de la voiture, nous proposons l'association des données de cartes numériques afin de gérer la navigation locale dans les points critiques du chemin, comme les intersections de routes. Des essais dans les scénarios difficiles, avec une voiture expérimentale, et aussi en environnement simulé, montrent

la viabilité de la méthode proposée.

Mots-clés : Navigation d'une voiture robotisée ; Segmentation de la route ; Asservissement visuel ; Approche de la fenêtre dynamique ; Navigation globale.

List of Figures

List of Tables

List of Algorithms

List of Symbols and Acronyms

Car-like Robot Model

l	car axis length.
$\{\mathcal{O}\}, \{\mathcal{R}\}$	inertial and robot body frames.
u_r	robot velocity control input set, composed by (v_1, v_2) or (v, ω).
v_1, v_2, v, ω	car velocities.
x_r, y_r, θ, ϕ	car-like robot pose.

Workspace Parception

$\{\mathcal{C}\}, \{\mathcal{I}\}$	camera and image reference frames.
f_x, f_y, c_x, c_y	camera intrinsic parameters.
$\mathcal{I}_{\triangle}$	disparity map image.
r_{obst}, θ_{obst}	obstacle distance and angle measurements.
t_x, t_y, t_z, ρ	camera extrinsic parameters related to the robot frame $\{\mathcal{R}\}$.
u, v	image column and row variables.
u_{max}, v_{max}	size of the image frame $\{\mathcal{I}\}$ in pixels.
$2X_{\mathcal{I}}, 2Y_{\mathcal{I}}$	size of the image frame $\{\mathcal{I}\}$ in the normalized perspective.

Local Navigation Control

B_r^+	Moore-Penrose matrix pseudoinverse of B_r.
L_s	interaction matrix.
$\lambda, \alpha, \beta, \gamma$	controller gains.
s, s^*	Image features set (X, Y, Θ) and the final desired configuration.
${}^C T_R$	transformation matrix from the robot frame $\{\mathcal{R}\}$ to the camera frame $\{\mathcal{C}\}$.
u_c	camera velocity set, composed by $(v_{c,x}, v_{c,y}, v_{c,z}, \omega_{c,x}, \omega_{c,y}, \omega_{c,z})$.
$v_{c,x}, v_{c,y}, v_{c,z}, \omega_{c,x}, \omega_{c,y}, \omega_{c,z}$	camera velocities in the frame $\{\mathcal{C}\}$.
V_{DW}	dynamic window search space around the current robot velocity (v_a, ω_a), composed only by safe admissible velocities.
X, Y, Θ	line projected features in the image frame $\{\mathcal{I}\}$.

Global Navigation Management

q_{init}, q_{goal}	Initial and goal position for the global navigation.

Acronyms

ACC	Adaptive Cruise Control.
ADAS	Advanced Driver Assistance Systems.
ANN	artificial neural networks.
BM	Block Matching.
CCD	charge-coupled device.
CP	critical point.
DARPA	Defense Advanced Research Projects Agency.
DWA	Dynamic Window Approach.
EKF	Extended Kalman Filter.
FOV	field of view.
HSV	Hue, Saturation, and Value.
IBVS	Image-Based Visual Servoing.
ICC	instantaneous center of curvature.
IDWA	Image-Based Dynamic Window Approach.
IEV	intelligent electric vehicle.
IMU	inertial measurement unit.
KITTI	KITTI vision benchmark suite.
LC-WT	Local Condition Watershed Transform.
MSE	mean square error.
OSM	OpenStreetMap.
PBVS	Position-Based Visual Servoing.
RDDF	route definition data format.
RGB	Red, Green, and Blue.
RNDF	road network definition file.
ROI	region of interest.
SAD	Sum of Absolute Differences.
SGBM	Semi-Global Block Matching.
UTC	Université de Technologie de Compiègne.
V2I	vehicle-to-infrastructure.

VS Visual Servoing.

Contents

Chapter 1

Introduction

1.1 Motivation

The world is dynamic and in constant evolution. The current technologies belong to this evolution, which improve the daily activities of human beings, reducing accidents and enhancing their quality of life. Robotics is one between many technological resources used for this end. Industrial robotics, e.g., perform repetitive tasks in dangerous scenarios with considerable productivity gains.

Similar to industrial robotics, autonomous vehicles have been increasingly in evidence in the last few decades, once several gains for the security, power consumption, efficiency, etc. are involved (Broggi et al., 1999). Although there were important contributions before, it is after the DARPA Grand Challenges, held by the American's Defense Advanced Research Projects Agency (DARPA) between 2004 and 2007 (Thrun et al., 2006; Buehler et al., 2008), that the potentiality of these vehicles have been put on test. Nowadays, there are vehicles able to drive in different situations, for long distances and respecting the traffic laws (Luettel et al., 2012; Wei et al., 2013; Ziegler et al., 2014) (see Figure 1.1).

However, these vehicles use high cost sensors, some of them impractical for final commercial cars. In addition, they must deal with some problems caused by the environment where the car is inserted, like the localization problems common related to GPS signal losses, as described by many DARPA participants (Thrun et al., 2006; Buehler et al., 2008). Even with differential corrections, the GPS information fails when the urban environment has structures like tall buildings, overpasses and tunnels (von Hundelshausen et al., 2008). Hence, this leaves us several possibilities for new navigation

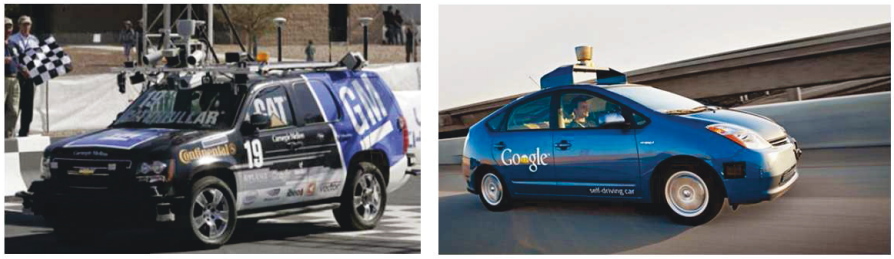


Figure 1.1 – Examples of car-like robots, where on the left is the Boss (Urmson et al., 2008), winner of the DARPA Urban Challenge and on the right is the Google Car (Guizzo, 2011).

approaches based on low cost sensors, which do not depend on accurate localization information and are better suited for the environment where the vehicle is inserted.

Sensor-based control is a useful strategy based on exteroceptive sensors data (like sonar, radar, LIDAR, and vision systems) to guide the robot during navigation tasks. This can be extended for car-like robots, once their workspace, mainly in the urban environments, are rich of perceptible information. Commercially, some manufacturers have already been used exteroceptive sensors in their Advanced Driver Assistance Systems (ADAS) for parking assistance, lane keeping, collision alert, Adaptive Cruise Control (ACC), etc. (Soualmi et al., 2014). In full autonomous applications, an industrial truck has been successfully applied in mining environments (Jamasmie, 2009).

The navigation tasks can be divided in local and global approaches. Local navigation strategies are those related to static or moving objects in the environment surrounding the robot. In urban environments they can be exemplified by road following, vehicle platooning, lane keeping, etc. For these applications, vision systems are a reliable low cost alternative which concentrate a large number of environment data in a single image (Bonin-Font et al., 2008). The viability of implementing many computer vision algorithms in hardware also increases the processing speed and reduces the power consumption, common problem in intelligent electric vehicles (IEVs). Visual Servoing (VS) is one of the many ways to deal with this guidance problem using visual features in a sensor-based navigation (Chaumette and Hutchinson, 2006). However, the approaches based on VS do not directly change the velocities of the vehicle to perform the obstacle avoidance, essential in urban environments navigation.

Apart from that, global navigation strategies are those responsible to guide the robot to a final destination. This requires a previous knowledge of the environment and a localization system in a global frame, which lead us to the localization problems mentioned before. Therefore, it is important to reduce the localization dependency during the robot navigation to minimize those problems.

In this context, we propose a new sensor-based navigation approach for IEVs, combining vision cameras for local navigation and digital map for global navigation management. The local navigation is based on a hybrid controller named (VS+IDWA), which uses a VS approach as deliberative control and a new reactive control derived from the Dynamic Window Approach (DWA) for obstacle avoidance (Fox et al., 1997). For this purpose, a complete environment perception using only cameras is proposed for road and obstacles detection. The global navigation management uses the semantic and geometric information from a digital map to assure the car navigation even in road intersections. The entire system performance is validated experimentally in simulation and in a real electric car. The objectives behind this method are detailed next.

1.2 Objectives

The present work addresses the sensor-based navigation problem for intelligent electric vehicles in urban environments. Focusing on approaches for low cost sensors, the following steps will be considered:

— The environment perception using vision systems, with the required information to navigate the vehicle, like the road area and obstacles;

— The local navigation control for road following and obstacle avoidance, based only on visual information of the environment;

— The global navigation management to ensure the vehicle motion in local tasks;

— The adjustment of an existing fully actuated IEV, an Renault Zoé, for sensor-based navigation, which includes the installation of sensors for perception of the environment, the definition of its motion model, the development of a global localization, and the implementation of a low-level velocity control;

— The development of a simulation environment for fast validation of the proposed navigation approach.

We expect to provide, with these previous resources, a complete navigation solution for our electric car, which can be improved with more sensors and local navigation strategies for future users.

The overall aim of this thesis is to provide original contributions to the existing methods and techniques in the development of a complete navigation solution for an electric vehicle that moves autonomously in urban areas, performing a preselected route defined by the user.

1.3 Contributions

This work presents a new sensor-based navigation strategy for autonomous robotic automobiles. Although the robot navigation area involves several important domains, here we considered three big areas during the development of the work: the Environment Perception, the Local Navigation Control, and the Global Navigation Management.

In the environment perception, the road and obstacles detection problem was considered for stereo vision cameras in urban scenarios. This brought us many troubles in the stereo image processing (using disparity maps) caused by shadows, light reflections, and low texture variations. Thus, we developed a methodology which improves the stereo information for road segmentation and disparity map refinement (Vitor et al., 2013; Lima et al., 2013). Building on these works, the current perception solution was implemented to allow the fast detection of road, obstacles, and the features required for the local navigation control. A quantitative evaluation compared the performance of these approaches for road segmentation on challenging scenarios.

For the local navigation control, we proposed a new reactive controller named Image-Based Dynamic Window Approach (IDWA) (Lima and Victorino, 2014c), integrating the VS control methodology for road lane following in the DWA for obstacle avoidance (Fox et al., 1997). However, following the desired road lane and avoiding obstacles are opposite tasks for the Image-Based Dynamic Window Approach (IDWA), which means that one of these tasks will not be realized with the best performance. Taking advantage of the VS (as deliberative) and the IDWA (as reactive) controllers, we presented a hybrid control solution (Lima and Victorino, 2014b), where the IDWA works as a validation method for the VS control outputs. In this approach, if the VS velocities are

not safe to be applied as control inputs in the car, the IDWA will perform the required reactive correction. This hybrid controller (VS+IDWA) takes into account the obstacles and the vehicle dynamic/kinematic constraints to perform the road lane following. Furthermore, the proposed reactive controller IDWA has been successfully applied for human driving behavior correction by another PhD. student (Kang et al., 2014a,b).

Considering only low cost sensors, the global navigation management focused on approaches which were less dependent on localization system, in this the GPS systems. Thus, we proposed a global navigation strategy (Pereira et al., 2014), combining digital maps with the previous local navigation control. In this strategy, the robot tasks were divided in road following (corridors) and road intersection maneuvers (critical points), where the car localization is required only when arriving at a critical point and without high precision. However, this approach presented limitations dealing with some intersections and was validated only in simulation environments. By means of the previous VS+IDWA controller, here we generalized this global navigation in a complete sensor-based approach.

In addition, all these presented contributions were experimented and validated in a real IEV, a full electric ZOE Renault from the Heudiasyc UMR CNRS 7253 Laboratory at the Université de Technologie de Compiègne. This zero emission vehicle was transformed by Renault with full actuation (throttle, brake and steering). In complement to our thesis work, we developed low level controllers for linear and steering velocities, that were useful to other developments on the Heudiasyc Lab (Vitor, 2014).

1.4 Organization

The Chapter 1 presented the motivations for the present work, the objectives in mind, and the proposed contributions. In the Chapter 2, a short overview about the autonomous robotic automobiles development will guide the reader about some important contributions in this domain. It also presents the related work in the environment perception, sensor-based navigation, and global navigation for car-like robots; with a special attention to those based on vision systems and low cost sensors. Then, in the Chapter 3 the IEV, used in the experimental results of this work, is shown in details, with the complete system design. A simulation environment, created to mimic the real

vehicle scenario, is also depicted in this chapter. The proposed solution implementation starts in effect in the Chapter 4, where the environment perception using vision cameras is detailed. Experiments in real urban scenarios show the robustness of this approach. Next, the Chapter 5 introduces the hybrid controller VS+IDWA, which integrates the deliberative control VS with the reactive control IDWA. The results and performance of these controllers were validated in both simulated and real car-like robot. Thus, this hybrid controller was combined with a global navigation management at Chapter 6, to afford the complete sensor-based autonomous navigation solution. The conclusions and perspectives for future works are discussed in the Chapter 7. The Appendix A brings further details about the distance to collision implementation for the reactive controller.

Introduction

1.1 Motivation

Le monde est dynamique et en constante évolution. Les technologies actuelles appartiennent à cette évolution, ce qui améliore les activités quotidiennes des êtres humains, avec la réduction des accidents et l'amélioration de leur qualité de vie. La robotique est un parmi d'autres nombreuses ressources technologiques utilisées à cette fin. La robotique industrielle, par exemple, effectue des tâches répétitives, dans des scénarios dangereux, et avec plusieurs gains de productivité considérables.

Comme pour la robotique industrielle, au cours des dernières décennies, les voitures robotisées ont été de plus en plus en évidence, une fois que plusieurs gains pour la sécurité, la consommation d'énergie, l'efficacité, etc. sont impliqués (Broggi et al., 1999). Bien qu'il y ait d'importantes contributions avant, c'est après les DARPA Grand Challenges, réalisés pour l'agence de la recherche du ministère de la Défense des États-Unis (DARPA) entre 2004 et 2007 (Thrun et al., 2006; Buehler et al., 2008), que la potentialité de ces véhicules a été testée. Actuellement, il y a des véhicules capables de se conduire en différentes situations, pendant des longues distances et en respectant les lois de circulation (Luettel et al., 2012; Wei et al., 2013; Ziegler et al., 2014) (voir la Figure 1.2).

Cependant, ces véhicules utilisent des capteurs de coût élevé, beaucoup d'entre eux impraticables pour les voitures commerciales. De plus, ils doivent faire face à quelques problèmes liés à l'interaction avec l'environnement où la voiture est insérée, comme les problèmes de localisation liés à la perte de signal GPS, décrits par plusieurs participants des chalenges DARPA (Thrun et al., 2006; Buehler et al., 2008). Même avec des corrections différentielles, les données GPS échouent lorsque l'environnement urbain a des structures comme des gros bâtiments, des viaducs et tunnels (von Hundelshausen et al.,

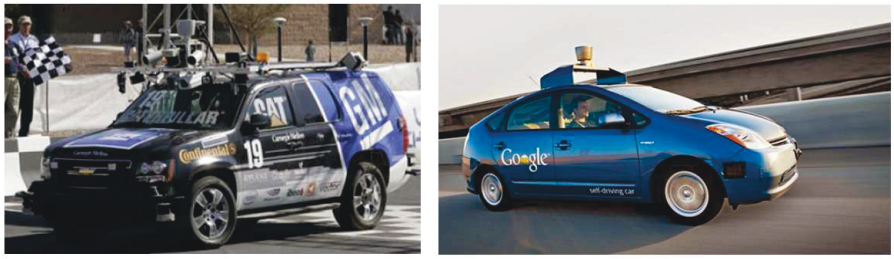


FIGURE 1.2 – Des exemples de voiture robotisées, où sur la gauche est le Boss (Urmson et al., 2008), vainqueur de la DARPA Urban Challenge, et sur la droite est la voiture de Google (Guizzo, 2011).

2008). Il est important donc de rechercher des nouvelles approches pour la navigation basées sûr des capteurs à faible coût, mieux adaptés à l'environnement dans lequel le véhicule est inséré.

La commande référencée capteurs est une stratégie utile pour guider le robot lors de tâches de navigation, en utilisant les données des capteurs extéroceptifs (comme le sonar, radar, lidar et les systèmes de vision). Ceci peut être étendu pour les voitures robotisées, une fois que leur espace de travail, principalement dans les milieux urbains, sont riches d'informations perceptible. Commercialement, certains fabricants ont déjà utilisé des capteurs extéroceptifs dans leurs systèmes d'aide à la conduite (ADAS) pour l'aide au stationnement, maintien sur la voie, alerte de collision, régulateur de vitesse adaptatif (ACC), etc. (Soualmi et al., 2014). Parmis les applications autonomes complètes, il y a celle où un véhicule de service robotisé a réalisé des tâches de navigation avec succès dans des environnements miniers (Jamasmie, 2009).

Les tâches de navigation peuvent être divisées dans les approches locales et globales. Les stratégies de navigation locale sont celles liées aux objets statiques ou dynamiques dans l'environnement autour du robot. En milieu urbain, elles peuvent être illustrées par les approches de suivie de route, platooning de véhicules, maintien sur la voie, etc. Pour ces applications, les systèmes de vision sont une alternative fiable à faible coût qui concentre un grand nombre de données sur l'environnement en une seule image (Bonin-Font et al., 2008). La viabilité de la mise en œuvre de nombreux algorithmes de vision par ordinateur au niveau du matériel augmente également la vitesse de traitement et réduit la consommation d'énergie, problème courant dans les véhicules électriques intelligents (IEVs). L'asservissement visuel (VS) est une des nombreuses façons de traiter

ce problème de guidage en utilisant des caractéristiques visuelles dans une navigation référencée capteurs (Chaumette and Hutchinson, 2006). Cependant, l'asservissement visuel est une méthode de commande délibérative qui n'effectue pas directement les changements de vitesse nécessaires pour l'évitement d'obstacles, essentiels dans les environnements urbains.

D'autre côté, les stratégies de navigation globale concernent le problème de guider le robot vers une destination finale. Cela demande une connaissance préalable de l'environnement et un système de localisation du robot dans un environnement urbain global, ce qui nous conduit aux problèmes de localisation mentionnés auparavant. Pour cela, il est important de réduire la dépendance sur la localisation pendant la navigation du robot et minimiser ces problèmes.

Dans ce contexte, nous avons proposé une nouvelle approche de navigation référencée capteurs pour les IEVs, en utilisant des caméras de vision pour la navigation locale et la carte numérique pour la gestion globale de la navigation. La navigation locale est basée sur un contrôleur hybride nommée (VS+IDWA), qui combine un contrôleur délibératif par le VS et un nouveau contrôleur réactif dérivé de l'approche de la fenêtre dynamique (DWA) pour l'évitement d'obstacles (Fox et al., 1997). Pour cela, une méthodologie complète de perception de l'environnement, en utilisant seulement des caméras, a été proposée pour la détection de la route et les obstacles. La gestion globale de la navigation utilise les informations sémantiques et géométriques d'une carte numérique pour assurer la navigation de la voiture, même dans les intersections. Les performances du système tout entier est validé expérimentalement embarqué dans une vraie voiture électrique, mais aussi dans un environnement de simulation. Les objectifs derrière cette méthode sont détaillés comme suit.

1.2 Objectifs

Le présent travail aborde le problème de navigation référencée capteurs pour les véhicules électriques intelligents dans des milieux urbains, en se concentrant sur les capteurs à faible coût pour la perception de l'environnement et localisation du robot, comme les systèmes de vision et GPS. Les étapes suivantes seront considérées :

— La perception de l'environnement en milieu urbain à l'aide de systèmes de vision, avec toutes les informations nécessaires pour faire naviguer le véhicule, comme la détection de la route et des obstacles ;
— Le contrôle de navigation locale pour le suivi de route et l'évitement d'obstacles, basée uniquement sur l'information visuelle de l'environnement ;
— La gestion globale de la navigation pour assurer la bonne voie à suivre pendant la navigation locale, les limitations de vitesse, etc ;
— L'adaptation d'un IEV, une Zoé Renault, d'actionnement complet, déjà existant, à la navigation autonome. Ce qui comprend l'installation des capteurs extéroceptifs pour la perception de l'environnement, la définition de son modèle de mouvement, et la mise en œuvre d'un système de localisation globale et un contrôle de vitesse de bas niveau ;
— Le développement d'un environnement de simulation pour la validation rapide de l'approche de navigation proposée.

L'objectif global de la thèse concerne à proposer des contributions originales aux méthodes et aux techniques existantes dans le développement d'une solution complète de navigation pour une voiture électrique qui se déplace de manière autonome en milieux urbains, en effectuant un itinéraire préalablement choisi par l'utilisateur.

1.3 Contributions

La navigation de robots comprend normalement plusieurs domaines importants, comme la localisation, la perception de l'environnement, le contrôle de mouvement et planification de la trajectoire, etc. Ce travail présente une nouvelle stratégie de navigation référencée capteurs pour les voitures autonomes. Par conséquent, nous avons étudié trois grands domaines au cours du développement de ce travail : la perception de l'environnement, le contrôle de navigation locale, et la gestion globale de la navigation (y compris localisation du robot et la planification de la trajectoire).

Dans la perception de l'environnement nous avons examiné les problèmes sur la détection de la route et des obstacles à l'aide des caméras de vision stéréo en milieu urbain. Cela nous a amené plusieurs problèmes dans le traitement de l'image stéréo (comme la carte de disparité) causés par des ombres, des reflets de lumière, et des faibles variations

de texture. Ainsi, nous avons développé une méthodologie qui améliore l'information stéréo pour la détection de la route et le raffinement de la carte de disparité (Vitor et al., 2013; Lima et al., 2013). Poursuivant les concepts présentés dans ces travaux, la solution de perception actuelle a été mise en œuvre pour permettre la détection de la route, des obstacles, et fournir des primitives visuelles utilisées dans la boucle de commande de la navigation locale. Tous les résultats ont été comparés quantitativement.

Pour le contrôle de navigation locale, nous avons proposé un nouveau contrôleur réactif nommé "approche de la fenêtre dynamique référencée image" (IDWA) (Lima and Victorino, 2014c), avec l'intégration d'une méthodologie de VS pour le suivi de la voie routière dans la DWA pour l'évitement d'obstacles (Fox et al., 1997). Cependant, suivre la voie routière désirée et éviter les obstacles sont des tâches opposées pour l'IDWA, ce qui cause une baisse de performance. Profitant des deux contrôleurs délibératif (VS) et réactif (IDWA), nous avons présenté une solution de contrôle hybride (Lima and Victorino, 2014b), où l'IDWA fait la validation des vitesses calculées par le VS. Dans cette approche, si les vitesses fournies par le VS ne sont pas applicables comme entrées de commande (dû aux limitations dynamiques et spaciales de la configuration du robot), l'IDWA effectuera la correction réactif. Ce contrôleur hybride (VS+IDWA) prend en compte les obstacles et les contraintes cinématiques/dynamiques du véhicule pour effectuer le suivi de la route en toute sécurité. En outre, le contrôleur IDWA réactive proposée a été appliquée avec succès dans la correction du comportement de conduite humaine par un autre doctorat (Kang et al., 2014a,b).

La méthodologie proposé ne considère que des capteurs à faible coût, de ce fait la gestion globale de la navigation doit être moins dépendante du système de localisation, dans ce cas basé sur le GPS. Ainsi, nous avons proposé une stratégie globale de navigation (Pereira et al., 2014), combinant des cartes numériques avec le contrôle de navigation locale décrite dans le paragraphe précédent. Dans cette stratégie, les tâches de navigation du robot peuvent être divisées en suivi de la route (couloirs) et les manœuvres dans les intersections (points critiques). La localisation de la voiture est nécessaire uniquement en arrivant à un point critique, sans précision exacte. Cette méthodologie a été validé dans des environnements simulées, en présentant certaines limitations aux croisements des certaines intersections de routes. Afin d'améliorer et enlever certaines de

ces limitations, nous avons développé une approche référencée capteurs pour effectuer des manœuvres dans les intersections de routes de manières plus robustes.

En plus, toutes ces contributions ont été testés et validés dans un vrai IEV, une ZOE Renault complètement électrique propriété du laboratoire Heudiasyc UMR CNRS 7253 à l'Université de Technologie de Compiègne. Ce véhicule zéro émission a été transformé par Renault avec un actionnement complet (accélération, freinage et direction). Parallèlement aux travaux développés dans la thèse, nous avons développé des régulateurs bas niveau pour la commande de la vitesse linéaire et de braquage du véhicule, qui a été profitable à d'autres développements au laboratoire Heudiasyc (Vitor, 2014).

1.4 Organization

Le Chapitre 1 a présenté les motivations pour ce travail, les objectifs envisagés, et les contributions proposées. Dans le Chapitre 2, un bref aperçu sur le développement des voitures autonomes guidera le lecteur sur des contributions importantes dans ce domaine. Il présente également les travaux liés à la perception de l'environnement, à la navigation référencée capteurs, et à la navigation globale pour les voitures robotisées ; en donnant priorité aux approches fondées sur les systèmes de vision. Puis, dans le Chapitre 3 l'IEV, utilisé lors des expérimentations de ce travail, est montré en détails, avec sa conception et les dépendances du système. Un environnement de simulation et ses capacités sont également illustrés dans ce chapitre. La mise en œuvre de la solution proposée commence en effet dans le Chapitre 4, où la perception de l'environnement basé sur des capteurs de vision est détaillée. Des expérimentations dans les scénarios réels urbains montrent la robustesse de cette approche. En suite, le Chapitre 5 présente le contrôleur hybride VS+IDWA, qui intègre les contrôleurs délibératif VS et réactif IDWA. Les résultats et les performances de tous les contrôleurs ont été validés dans le véhicule expérimental et en simulation. Ce contrôleur hybride a était combinée avec la gestion globale de la navigation dans le Chapitre 6, pour fournir la solution référencée capteurs complète. Les conclusions et les perspectives pour les travaux futurs sont indiqués dans le chapitre 8. L'Annexe A présente plus en détails le calcule de la distance de collision sur le contrôleur réactif.

Chapter 2

Related Work

In the last feel decades, several works have been developed for autonomous robotics automobiles, also known as driverless cars. The first step was given in the Tsukuba Mechanical Engineering Lab, Japan, at late 1970s, with a driverless car conceived with heavy and slow computers. It was able to track white street marks with computer vision at speeds up to 30 km/h. However, the first car-like robots in Europe were created 10 years after in the Bundeswehr University Munich (UniBW), in Germany, under supervision of professor Ernst Dickmanns (Dickmanns and Zapp, 1987; Dickmanns et al., 1990, 1994). Some American authors compare Dickmanns to the Wright brothers [1], due to his importance in the development of autonomous cars. The vehicles created were part of the project PROMETHEUS [2] (from 1987 to 1995) and covered thousands of kilometers in traffic at velocities up to 175 km/h, where about 95% was completely autonomous driving. They were based on vision cameras with saccadic movements.

Along with the project PROMETHEUS, during the 1990s more autonomous vehicles had emerged, following the advancement of sensors and portable computers. The first U.S. contribution, e.g., came in this period with the project called “No hands across America" from the Carnegie Mellon University (CMU). They developed a car named Navlab 5 (Pomerleau and Jochem, 1996), performing autonomous navigation from Washington DC to San Diego with 98% automated steering and manual longitudinal control. Another important contribution came from the Italian project ARGO proposed by (Broggi et al., 1999), an offshoot from the project PROMETHEUS with similar results.

1. The brothers Orville and Wilbur Wrigth are considered by the North-Americans the “Fathers of Modern Aviation".

2. PROMETHEUS stands for ***PRO****gra****M****me for a* ***E****uropean* ***T****raffic of* ***H****ighest* ***E****fficiency and* ***U****nprecedented* ***S****afety*.

Nevertheless, a large number of contributions have appeared in the 21th century, coinciding with several public challenges. The most significant were carried out by the American's DARPA, with the first Grand Challenge proposed in 2004 for off-road autonomous vehicles. However, the effective one was the second challenge proposed for 2005, which had a winner (Thrun et al., 2006). These competitions bring several new sensors, like the 360° laser scanners (Halterman and Bruch, 2010) (shown in the Figure 2.1c). The third competition, the Urban Challenge (Buehler et al., 2008), was held in November 2007 in a fake urban environment. Differently from the previous ones, the Urban Challenge had demanded some interaction with others vehicles and urban features from the participants. However, important interactions between pedestrian, bicyclists, or traffic lights, still not required. Almost all produced vehicles used very expensive sensors, like high-end laser scanners coupled with radars and high-precision GPS/INS, impractical for final commercial purposes. For the cooperative scenarios was proposed the Grand Cooperative Driving Challenge. The first one was performed at the Netherlands highways in 2011 (Ploeg et al., 2012), focusing on the ability to carry out longitudinal control (platooning). The next will be hold in 2016, adding the lateral control (steering) problem.

Nowadays, many teams worldwide have continued the development of autonomous cars. In U.S., it is impressive the results presented by the Google car (Guizzo, 2011), based on the expertise extension gained in the DARPA Urban Challenge. Its main components are a high-end laser scanner on the car's roof and a prerecorded map (constructed during a manual drive). In Europe, the group supervised by Alberto Broggi has performed an impressive long term autonomous navigation from Parma, Italy, to Shanghai, China, by applying cooperative driving (Broggi et al., 2012). Recently, they presented a full autonomous result in the public traffic area around the streets of Parma. Another group, based in Germany, has presented some contributions in autonomous driving, with the vehicle Bertha Benz (Ziegler et al., 2014), and validation of computer vision algorithms, with the KITTI Vision Benchmark (Geiger et al., 2012). Comparing with the Google's car, these European projects focus on the sensor setup regarding robustness, availability and redundancy for final commercial terms. It is also important to mention that there are other groups working with driverless cars around the world.

In Brazil, e.g., the projects CADU [3] (Lima and Pereira, 2013) and CaRINA [4] (Fernandes et al., 2014), deal with several problems related to unstructured urban environments, different from those normally seen in Europe and U.S.

In the present work, we focused our efforts on: environment perception, sensor-based control for autonomous navigation, and global navigation management; all based on vision systems and low cost sensors. The following sections will detail the related work on these areas and how they were embedded in the some vehicles.

2.1 Environment perception

Environment perception is a large domain in mobile robotics which involves different sensors, detection techniques, sensors fusion, and final data representation. Several authors say that the contribution of the perception layer for the robot navigation is around 70% of the entire system, since it provides all the environment information for the robot motion. Here, we present the most commonly used sensors in intelligent vehicles, some detection applications for vision cameras, and the possibilities for data fusion and final representation, as detailed in the next subsections.

Perception sensors

Considering the sensors applied on intelligent vehicles, the most common are the sonars, lidars, and vision systems, illustrated in the Figure 2.1. They are divided in active and passive ones, depending on the origin of the signal detected. Active sensors emit their own signal and passive sensors only detect the signals from the environment.

Sonar is an acronym for *sound navigation and ranging*, using the sound propagation for distance estimation. They are the cheapest active sensors present on the intelligent vehicles, used basically on parking applications (Park et al., 2008). However, they are limited to one direction and short distances. An example of sonar sensor for commercial vehicles is depicted in the Figure 2.1a.

3. Portuguese acronym for *Autonomous Car Developed at Federal University of Minas Gerais*.
4. Portuguese acronym for *Intelligent Robotic Car for Autonomous Navigation*.

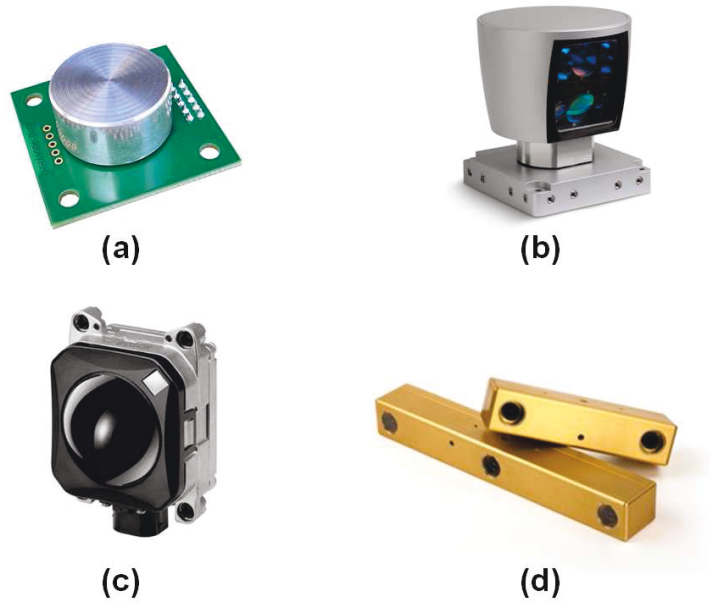


Figure 2.1 – Examples of perception sensors for intelligent vehicles, with a sonar (a), radar (b), multilayer laser scanner (c), and vision systems (d).

Lidar stands for *light and radar*, combining in the same definition the laser scanners and the radar systems (presented in the Figures 2.1b-c). They are related to light and radio emissions respectively. These are the sensors with most increasing applications, once they can deal with big distances with higher precision and light variations. These sensors were largely used in the DARPA Gran Challenges (Thrun et al., 2006; Buehler et al., 2008) and several projects after that (Guizzo, 2011; Ziegler et al., 2014). Comparing laser scanners with radars, the first ones have exhibit much better lateral resolution and accuracy for static and moving objects, but limited to planar measurements; the second ones are robust to different weather situations and more reliable in large-scale production, but it is restrict to mobile objects detection and return only the object center instead of the collision point in its surface. In a direct relation, the laser price increases considerably if increasing the number of available planes, as in the case of the Velodyne (Figure 2.1c) for point cloud applications (Halterman and Bruch, 2010). At present, radars are commercially applied in ACC applications, once they deal better with moving obstacles in higher speeds (Ploeg et al., 2012).

Vision systems are normally passive [5], depending on the visible light on the environment or infra-red in darkness situations. They produce a large amount of data in just one image frame [6], which, if combined with other cameras, can produces dense 3D information of the environment by stereo matching. Due to their large amount of data and a reasonable price, these are the most largely used sensors in intelligent vehicles (Bonin-Font et al., 2008). Despite that, they must deal with several problems like: light variations, low-texture surface, shadow areas, light reflection, range resolution, etc. Thus, the images must be worked by pre- and post-processing algorithms in the image frame (2D information) or in the stereo matching (3D information).

Although these previous sensors are able to detect different kind of obstacles as well as the road surface, they have a restricted working region. This means that, one element detected with one sensor will not be necessarily perceived by the other one or vice-versa. Real vehicles must combine and add more sensors to improve their perception capabilities. In the literature, e.g., it is hard to find autonomous vehicles using only one of these sensors. Normally, they combine cameras information and lidar data (Ji and Prokhorov, 2008; Flórez et al., 2014; Ziegler et al., 2014), using the best of these sensors in a fusion strategy.

With the sensors limitation in mind, this work will be concentrated in vision systems for road and obstacles detection only to validate our navigation approach, based on features detected in the image frame. Nevertheless, a recent work from *AdasWork* (AdasWorks, 2015), a research team from the Kishonti Informatics, showed a feasible self-driving car approach using only cameras applications and low-cost systems. It is also important to know that there are many other techniques to accomplish the road and obstacles detection (Nedevschi et al., 2007; Moras et al., 2012) with different sensors. Their integration in the present work as well as the addition of more sensors will be left for future works.

5. There are considerable works for active cameras based on Time-of-Flight (ToF), but their usage still limited due to problems related the low resolution and noise if compared to the passive cameras (Wei et al., 2011).

6. The amount of data available depends on the camera field of view (FOV) and resolution.

Vision based road and obstacle detection

Considering the mono and stereo color vision systems, there are several approaches where these cameras can be applied for urban environment perception. Usually, the mono vision is applied in image segmentation (to detect roads, vehicles and pedestrians) and primitives' extraction (like traffic signs and land marks). In the domain of image segmentation, the regions are defined accordingly to some similarity or discontinuity. For road detection the most common segmentations are based on color (Alvarez and Lopez, 2011; Rahman et al., 2012), texture (Shinzato and Wolf, 2011), and intensity (Bilodeau et al., 1990; Yu et al., 1992). In the case of stereo vision applications, the 3D information of the environment is typically used to estimate free spaces and obstacles position in the world with specific techniques, like the U/V-Disparity Maps (Labayrade et al., 2002; Broggi et al., 2005; Hu and Uchimura, 2005; Gao et al., 2011) for example.

However, to classify a road profile and potential obstacles from only 2D image segmentation data is a hard task. Mainly due to the high complexity of the urban environments and presence of many different elements like cars, pedestrians, trees, etc. In addition, the monocular vision cannot estimate the object position on the 3D world without a geometric approximation between the camera and the environment, resulting errors in long-distance. On the other hand, the stereo vision must deal with different noise sources, as shadows, weakly-textured surfaces, light variations, etc., which hinders the stereo matching and, consequently, the construction of disparity maps.

As a solution for these problems, some works enhance the 2D image data with 3D information from stereo cameras, improving the classification and detection of road segments (Soquet et al., 2007; Dornaika et al., 2011) and other environments objects (Sengupta et al., 2012). In this context, we proposed in (Vitor et al., 2013) an road surface detection with 94% of accuracy in urban environments. Based on the Watershed transformation for 2D image segmentation (Audigier and de Alencar Lotufo, 2010), V-Disparity maps to extract 3D information (Labayrade et al., 2002), and an artificial neural networks (ANN) for final classification (Haykin, 1998), several image features were tested in order to better classify the road surface. Furthermore, the usage of ANN helped to reduce the effects of the environment variations on the classification results.

Other works improve directly the disparity map pre- or post-processing, reducing its underestimation. The preprocessing aims to provide dense disparity maps, but resulting in high computational cost, as the Global Matching techniques (Kolmogorov and Zabih, 2001). There are also the intermediate solutions based on Semi-Global Matching techniques (Hirschmuller, 2005), which presents satisfactory results with fast embedded solutions (Banz et al., 2010), very important in ADAS. The post-processing, otherwise, refines the underestimation of the disparity map by grouping neighbors' similarities in clusters or superpixels. These clusters are formed by segmenting local features (e.g., corners, edges, colors, etc.) in the 2D reference image. Then geometrical forms are fitted in the clustered data, like planes (Gupta and Cho, 2010; Xu et al., 2012). In the post-processing domain, we presented in (Lima et al., 2013) a disparity map refinement, combining the same watershed based 2D segmentation of (Vitor et al., 2013) with a plane estimation in the formed clusters. The result enhanced the disparity map even in problematic areas.

Although our previous contributions had dealt with some problems associated with the vision systems, they had high computational costs and would require a long programming effort to become more time acceptable for our navigation approach. Instead of that, in this work the main ideas presented in (Vitor et al., 2013) were adapted to respect the entire system time while performing the road segmentation and obstacle detection.

Sensor fusion and final data representation

Once detected, the road and obstacles must be stored for the navigation purposes. They also need some processing to reduce false detection from the previous steps. Moreover, if more than one sensor is used, their detected data need to be merged by some fusion strategy. Occupancy Grids is an important solution proposed by (Elfes, 1989), applied since for data fusion, robot navigation, localization and obstacle avoidance (Thrun et al., 2005). It is one of the most largely used techniques on mobile robotics which divides the workspace in cells. These cells present fixed or variable size with some value associated to each one. Originally proposed for 2D representations of the environment, without any assumption of geometrical forms, it has been expanded to other

dimensions like 2.5D and 3D (Souza and Goncalves, 2012), also applied in dynamic conditions (Mitsou and Tzafestas, 2007; Moras et al., 2014), etc.

The occupancy grids can be divided in probabilistic, using Bayes filters (Thrun et al., 2005), or evidential, based on the Dempster-Shafer Theory (Dempster, 1967; Shafer, 1976). They differs on the fusion result, information metrics, and final decision (Moras et al., 2014). In the representation, whereas probabilistic occupancy grids deal only with occupied and free cell states, the evidential ones can deal with several different states in a semantic representation of environment (Vitor, 2014). However, more states means more computational cost associated. Considering only road (free) and obstacles (occupied) states, the probabilistic occupancy grid was enough for this initial perception application.

2.2 Sensor-based control for autonomous navigation

Sensor-based control is an important robot navigation strategy using external data. In this control approach, the error regulation is done on the sensor frame. For instance, the sensors used can be those from environment perception (like sonars, lidars, vision systems) (Victorino et al., 2003a), localization (as GPS) (Yang et al., 2013), contact (with the force sensors) (Ohishi et al., 1992), etc. The most common sensors for car-like robots are associated with perception of the environment and localization, once it must move safely from an initial configuration to a final goal state, and without any interaction contact with the environment objects.

As detailed in the previous section, vision systems play an important role in autonomous vehicle applications, due to the richness and versatility of the information that they supply (Bonin-Font et al., 2008). These applications include monovision (Dickmanns and Zapp, 1987; Kosecka et al., 1998; Thrun et al., 2006; Courbon et al., 2009) and stereovision (Broggi et al., 1999; Chen et al., 2009; Broggi et al., 2012; Ziegler et al., 2014) cameras. Focusing on the navigation control area, Visual Servoing (VS) is a robust way to deal with the vehicle guidance in urban environments, once it avoids car localization problems, like those from (Thrun et al., 2006) and (Buehler et al., 2008). VS can be divided in two main approaches: the Position-Based Visual Servoing (PBVS) and the Image-Based Visual Servoing (IBVS) (Chaumette and Hutchinson, 2006). In

the PBVS the control objective is expressed in the robot's Cartesian Space and requires 3D information of the scene, which can be estimated by monovision and stereovision cameras. In the IBVS, otherwise, the control objective is expressed in the image frame directly. Whereas PBVS better deals with large errors in the features set, IBVS is more robust against errors in the camera calibration (Chaumette, 1998).

Beyond these techniques, several control laws can be defined to allow a vehicle to converge and follow perceived features and primitives in the image, like points, lines, and ellipses (Espiau et al., 1992). These are called *model-based* approaches, which require some geometrical knowledge of the environment. Normally, these controllers combine a features tracking with the robot forward movement by means of task functions. These approaches can also be adapted for pose stabilization orientated by landmarks (Jurie et al., 1994; Lee et al., 1998; Usher et al., 2003) and path reach and following (Cherubini et al., 2011). Conversely, the *appearance-based* approaches use a topological graph of the environment represented by key images, which define the positions where the robot must pass (Courbon et al., 2009; Cherubini and Chaumette, 2013). It is important to mention that there are many others VS approaches for non-holonomic robots, as: the hybrids approaches that combine the both IBVS and PBVS information in a 2 1/2D visual servoing (Malis et al., 1999), and the methods where no previous knowledge about the scene are required (Rives, 2000; Silveira et al., 2006).

Although these visual navigation methodologies can guide the vehicle, they do not directly change their velocities to perform the obstacle avoidance. This is essential in urban environments navigation, where the road boundaries and other obstacles restrict the movement of the car. Considering the obstacle avoidance problem, some approaches define control laws combining the VS task with some reactive obstacle avoidance methodology (e.g., potential fields and tentacles) (Cadenat et al., 1999; Folio and Cadenat, 2006; Cherubini and Chaumette, 2013). These tasks are often merged in the control level by some switching strategy which changes the task weight in the presence of obstacles, consequently changes the control law. Moving obstacles can also be considered in this kind of approach, as presented in (Cherubini et al., 2014).

Instead of just switching or changing the gain between the deliberative and reactive strategies, we focused on a hybrid strategy that could work directly in the robot velocity space. The obstacle avoidance methodology must validate the VS control input

or choose an alternative which will result in less VS error. In this context, in (Lima and Pereira, 2013) was presented a method for car-like robot navigation based on the validation of a Velocity Vector Field (Pereira et al., 2009) based on a Dynamic Window Approach (DWA) (Fox et al., 1997). This hybrid controller follows the vector field when it is valid, and it avoids obstacles prioritizing the final orientation of the vector field. Comparing with other reactive techniques, like the tentacles (von Hundelshausen et al., 2008; Cherubini and Chaumette, 2013), whereas the tentacles approach uses predefined paths (or tentacles) to choose the best one (regarding some conditions) and then calculates the velocities to follow this path, the DWA calculates the reachable velocities around the current ones, to choose among then the best one related to some conditions (which can be similar to those from the tentacles).

In the present work, we will address the local navigation problem of an autonomous car using the VS methodology for road lane following with obstacle avoidance. The VS control will act as a deliberative controller and its velocities will be validated in a reactive controller based on the DWA. The VS uses an image-based approach (IBVS) with a reduced features set, similar to the one presented in (Cherubini et al., 2008), to calculate a robot control input to track the road lane center. The DWA defines a dynamic window considering the obstacles, the current vehicle state, and some dynamic/kinematic constraints to validate the current VS control input. When not valid, an alternative for the control input must be selected to perform the reactive obstacle avoidance with less VS error as possible. To do so, the VS equations were integrated in the DWA, compounding a new Image-Based Dynamic Window Approach (IDWA) [7] (Lima and Victorino, 2014c). This combination results in a hybrid controller (VS+IDWA) independent of the vehicle localization, diverging from the methodology presented in (Lima and Pereira, 2013), which required a global path planning and a localization system. This work also diverges from the previous ones (Cadenat et al., 1999; Folio and Cadenat, 2006; Cherubini and Chaumette, 2013) based on VS, once the obstacle avoidance proposed with the DWA incorporates in an intrinsic way path following and velocity control behavior in its calculation, without changing the control law. Just like (Cheru-

7. Once the IDWA uses image-based features and the 3D information of the obstacles, this controller can also be classified as a hybrid controller in the visual servoing context.

bini et al., 2014), the usage of the DWA also enable us to deal with moving obstacles, as described in (Seder and Petrovic, 2007), which will be left for future works.

2.3 Global navigation applied to car-like robots

The complete navigation of self-driving cars is the ability to go from an initial position (q_{init}) to a final destination (q_{goal}) in the world, considering that these points are not connected in the robot workspace. To achieve that, motion planning is fundamental to define a feasible path connecting these points. In this context, several approaches were proposed in the past years for nonholonomic car-like robots path planning (Svestka et al., 1995; Frazzoli et al., 2000; Lamiraux and Laumond, 2001), defined in the car's configuration space (Choset et al., 2005). They require the previous knowledge of the map structure with precision, the robot dimensions, and some movement constraints. However, the path planned with these strategies normally presents a high computational cost associated with the size of the urban environment. In addition, the robots have required accurate localization system while performing the path following tasks, impracticable for commercial cars.

To deal with these problems, it is common to consider the vehicle as a punctual holonomic robot during the path planning and simplify the environment in geometric forms and segments. These elements edges define the points where the robot must pass (waypoints). Then, managing the waypoints, the robot locally performs the navigation considering the environment information. Waypoints can be seen as a simplification of the entire robot path, recovered by connecting each consecutive waypoint. This approach was applied by several participants in the DARPA Grand Challenge of 2005 for off-road autonomous vehicles, based on the route definition data format (RDDF). The RDDF provided a list of longitudes and latitudes (waypoints), corridor widths that define the course boundary, and a list of associated speed limits. With the RDDF, the teams were able to accomplish the challenge globally, by connecting the waypoints and managing their information with GPS data, and locally, by navigating in the desert roads respecting the detected boundaries and speed limits (Thrun et al., 2006; Braid et al., 2006). Although the waypoints were able to guide the vehicles, they were previously defined by the organizers of the challenge with high precision.

In urban environments, one way to define waypoints is using road network maps, also known as digital maps. Digital maps provide rich information about streets, tracks, railways, waterways, points of interest, also detailing the number of lanes and speed limits. There are many approaches using digital maps to improve the global navigation problem, like enhancing the vehicle localization (Mattern et al., 2010; Irie and Tomono, 2012; Schindler, 2013) and the difficulties of autonomous vehicles navigation as a whole (Urmson et al., 2008). Based on digital maps, the DARPA Urban Challenge (Buehler et al., 2008) organizers provided the road network definition file (RNDF) to the participants. The RNDF contained the checkpoints where the vehicles must pass and the geometric information on lanes, lane markings, stop signs, parking lots, and special checkpoints. However, the RNDF was inaccurate and could add further errors if the vehicle blindly follows the road using that information. Thus, the participants locally planned paths between the checkpoints, defining waypoints to guide the robot navigation and correct the robot lateral position by means of a high precision localization system and the RNDF information.

The OpenStreetMap (OSM) (Haklay and Weber, 2008) is a user-generated street map interface, with the same benefits of several commercial digital maps, with the advantage of being completely open access. Nowadays, several applications have emerged using OSM data (Hentschel and Wagner, 2010; Tao et al., 2013; Kurdej et al., 2015), due to its friendly edition interface which enables to incorporate very specific elements to the robot task, like buildings, lane marks, and pedestrian crossing in the map. Recently, Ziegler et al. (2014) performed the autonomous navigation in urban environments using a prerecorded reference trajectory (waypoints) for the vehicle and OSM data. They use a DGPS-aided inertial navigation system during the recording process, adding more environment information to the digital map from the OSM (like lanes number, speed limits, stop signs, etc.) in a post-processing step. Thus, during the real car navigation, map based localization was performed with a less costly system.

Although all these previous methods were able to guide globally the robot, locally they were highly dependent on the localization system. Victorino et al. (2003b) proposed a new navigation approach for indoor environments with a topological representation of the robot path. In this case, each corridor was connected by nodes representing the relations of accessibility in the environment (doors and crossing corridors). Once at

the corridor, the robot can navigate only to the next node, i.e. by means of a sensor-based control strategy with no localization requirements. At the node, the robot must localize itself with relation to the node and take the next direction described in the topology.

Generalizing the topological representation from (Victorino et al., 2003b) to urban environments, corridors are roads, without any-other way to go, and nodes are points where some action must be taken (like turn to a next road in the intersection or stop in a traffic light). Nodes were also called by critical points, where, in comparison to waypoints, they are sparsely defined and do not represent the path's form if connected. Thus, the robot localization is not required for the road following navigation (due to the absence of waypoints), and some sensor-based approach (like those presented in the Section 2.2) is enough to guide it to the next critical point. At the critical point, the vehicle localization can be performed by low cost systems, which can be less dependent on GPS sensors (Comport et al., 2011; Hassan et al., 2013). This navigation principle mimics the one performed by the personal vehicular GPS when guiding the human driver to the desired destination by giving instructions only on some specifics points. A topological segmentation was also applied in (Regele, 2008), helping to define a high-level abstract world model which supports the decision-making process of an autonomous car.

In this work, the goal is to enable techniques like (Victorino et al., 2003a; Lima and Victorino, 2014b) to accomplish their navigation task when there is more than one possibility to go. For that, it is fundamental to provide information associated to the critical points, as their connections and approximated geometrical form. Thus, digital maps (like the OSM) are important tools for this purpose.

Chapter 3

System Design

To implement and validate the proposed navigation solution, this work was divided in: Environment Perception, Local Navigation Control, and Global Navigation Management. However, to implement a final workable solution, the robotic system must be considered when proposing any additional functionality. This is required by the final application, which must respect the robot system specifications and the available resources. Thus, the environment perception, e.g., must consider the available sensors (cameras, lidar, sonars, etc.) and the process time (in accordance to the robot speed); the same is applicable to the local navigation control and the global navigation management. In addition, we consider that it is not possible developing robust autonomous navigation methodologies without experimentation means. In this sense, a great effort was made, in parallel to the thesis scientific and methodological developments, to prepare and putting in operation an experimental car, in order to validate in real situations all the proposed methodology. In this chapter, the robotic vehicle and its available resources are described, as follows.

3.1 The experimental car-like robot

The experimental car is an electric Renault Zoé, hereby appointed APACHE and presented in the Figure 3.1. It was modified for computer actuation of the throttle, brake, and steering, receiving commands over the CAN bus. Although these are important modifications, the vehicle preserves the original human driver conduction capabilities, allowing e.g. a cooperative working mode between the human driver and the computational system. The electric vehicle also aggregates some benefits for the linear velocity

control, once the nonlinear dynamics caused by the transmission system are not presented in this electric car (Dias et al., 2014). Besides the low-level control (steering, throttle, and brake), several proprioceptive information can be retrieved from the CAN bus. Some of them are illustrated in the Figure 3.1, like the ABS and odometry from the wheels, the linear and steering velocities, steering angle, etc.

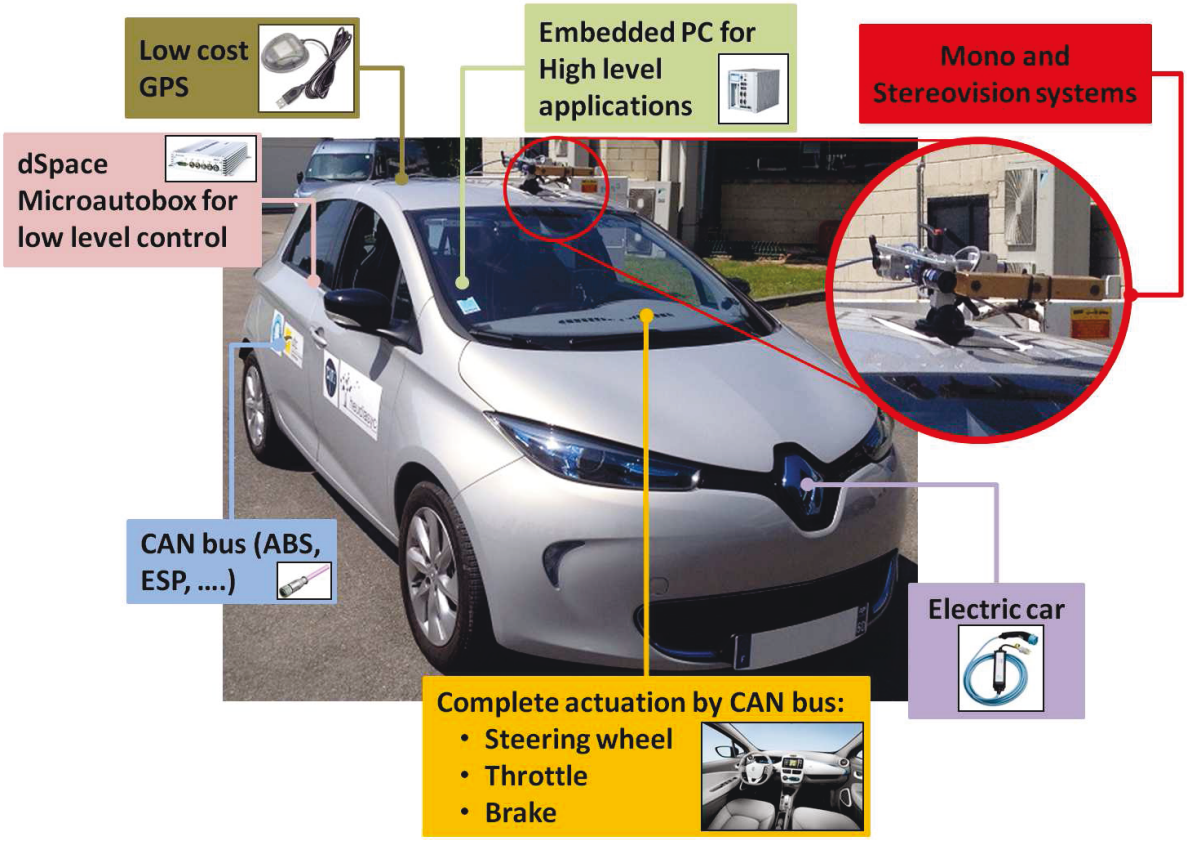


Figure 3.1 – Fully actuated electric car APACHE with some available resources.

This vehicle was equipped with two PointGrey cameras (Flea3 and BumblebeeXB3) for perception of the environment and a GPS for global localization (GPS Haicom HI-204E), chosen as a low cost alternative for exteroceptive sensors. The monocular camera (Flea3) has a focal length of $1.8mm$ and large field of view (FOV) of $\simeq 140°$ to detect the road lane center features. This large field of view (FOV) required a special setup, which will be detailed in the Section 3.2. The stereovision camera (BumblebeeXB3) has a focal length of $6.0mm$ and FOV of $\simeq 43°$, which allow the vehicle to detect road and obstacles up to $80m$ away from it. Both cameras were installed on a rigid structure on the car's roof only for validation purposes. The GPS works at $1Hz$ and produce position errors in a radius of $\approx 5m$.

To allow the implementation and execution of the navigation strategy, some embedded computers for data processing and vehicle control were installed in the APACHE (Figure 3.1). With these available resources, the solution was structured as in the Figure 3.2. The present work is concentrated in the block PC of this figure, which represents a computer with an Intel Core I7–3610QE CPU ($2.30GHz$). The inner blocks are:

— *EKF*: responsible for the car localization, described in the Section 3.4;
— *Velocity Control*: for the low-level velocity control of the car, providing the setpoints for the throttle, brake, and steering actuators, presented in the Section 3.5;
— *Environment Perception*: where the road segmentation and obstacle detection are performed, detailed in Chapter 4;
— *Local Navigation Control*: where the sensor-based local navigation task is executed, as will be shown in the Chapter 5; and
— *Global Navigation Management*: to control the global navigation of the car and define the right parameters to the local navigation control, presented in the Chapter 6.

This vehicle is an important test bed for our navigation approach, but, due to the current modifications, it is still not allowed to run in the city roads. Thus, at the Innovation Center of the Université de Technologie de Compiègne (UTC) was constructed a circuit called SEVILLE, Figure 3.3, for autonomous navigation experiments. It presents some challenging scenarios, like tight road intersections, roundabouts, sharp turns, and up and downhills (with $\approx 3.5^{\circ}$ of slope), which will be used during the validation of our navigation approach. The track is illustrated in the Figure 3.3.

3.2 Monocular camera setup

The large FOV capability of the monocular camera used (Section 3.1) required a special attention during the setup. The main interest in this camera is for road surface estimation and features extraction for the local navigation control. Due to that, the sky information can be completely discarded. In addition, images grabbed from this camera in a sunny day brings a lot of saturation and blooming problems, commons to the charge-coupled device (CCD) camera sensor (Theuwissen, 1995). A saturation and

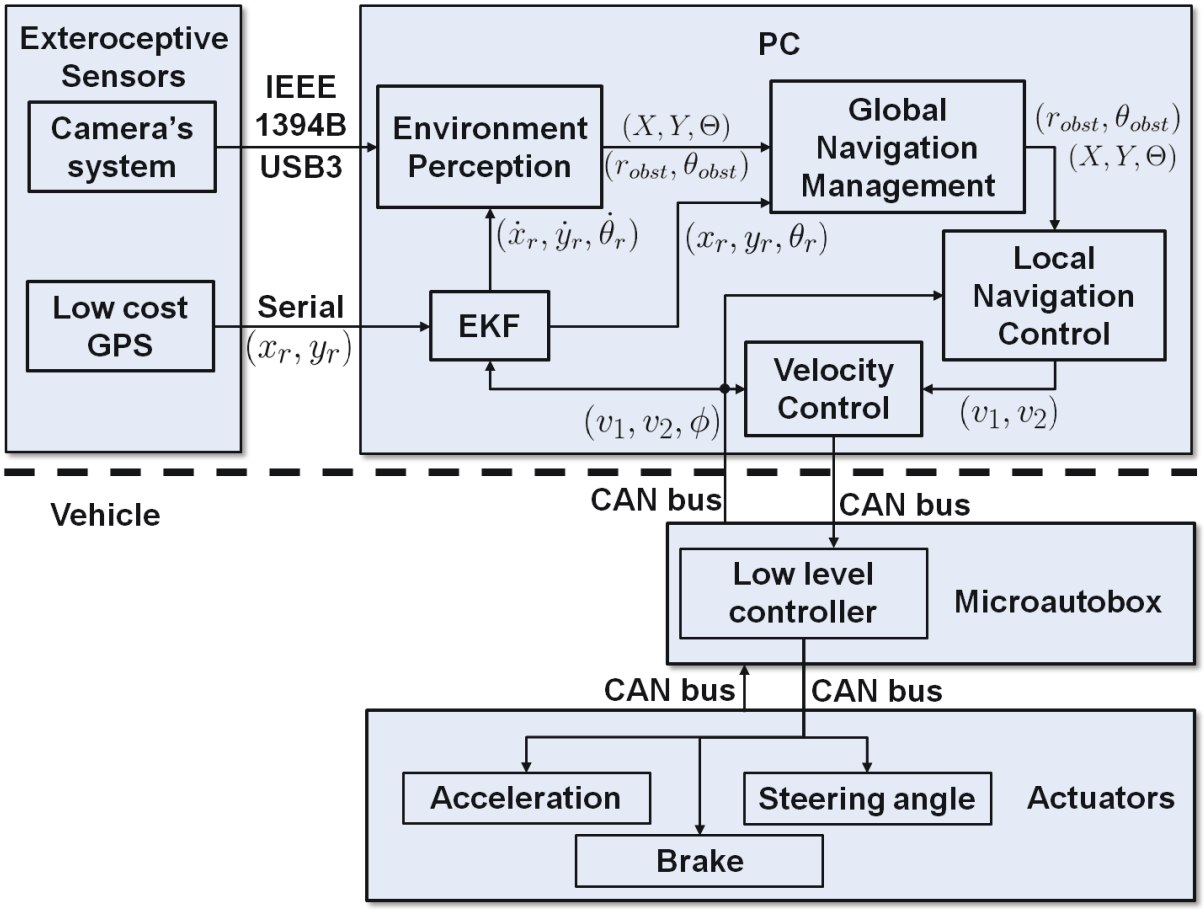


Figure 3.2 – Schema of the APACHE's system modules and the final software architecture in the embedded PC.

blooming effect can be seen in the Figure 3.4a. In this situation, a half portion of the image can be discarded by a region of interest (ROI) definition. Although a ROI could be defined in software, the blooming effects associated to the CCD sensor conception still appear and can compromise the other valid pixels. To avoid these problems, a protection similar to a hat's flap was installed above the camera, as illustrated in the Figure 3.4b.

However, this resulting image can compromise the camera's auto-shutter adjustment, once there are a lot of useless pixels, as the dark region (black pixels) formed by the flap and the car hood projection. This effect can be visualized on the images from the Figure 3.5a, where the car was oriented against or directed to the sun which resulted in different light conditions and pixels saturation in the final image. As a solution, a ROI was defined, as illustrated in the Figure 3.6, to eliminate the flap region and

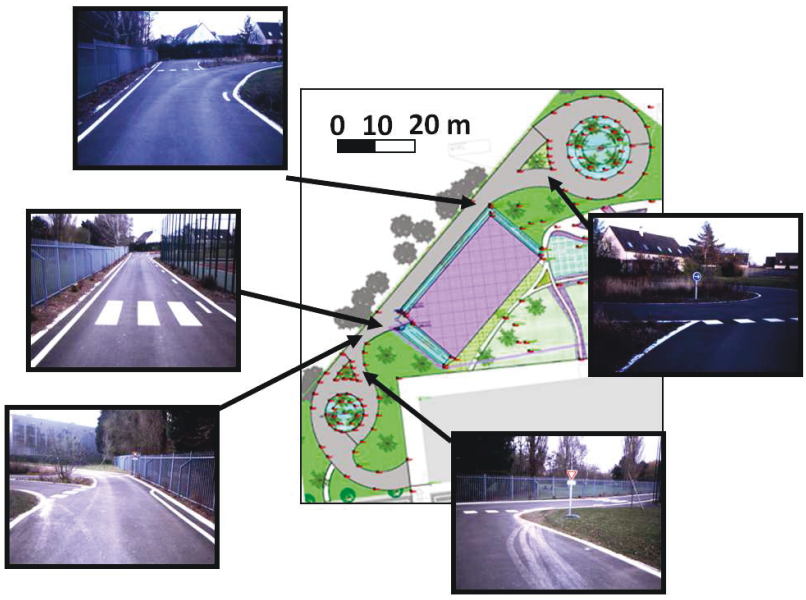


Figure 3.3 – SEVILLE test track at the UTC Innovation Center for autonomous navigation experiments.

the car hood from the final image. The pixels from the valid area were so applied to a proportional-derivative (PD) controller (Figure 3.6), based on the saturated pixels rate

$$w_{PV} = \frac{\sum (ROI_{pixels} > 250)}{\sum ROI_{pixels}} \tag{3.1}$$

in the final image to define a new shutter value (sh). With this PD controller, the variations caused by different light conditions were minimized, as shown in the Figure 3.5b.

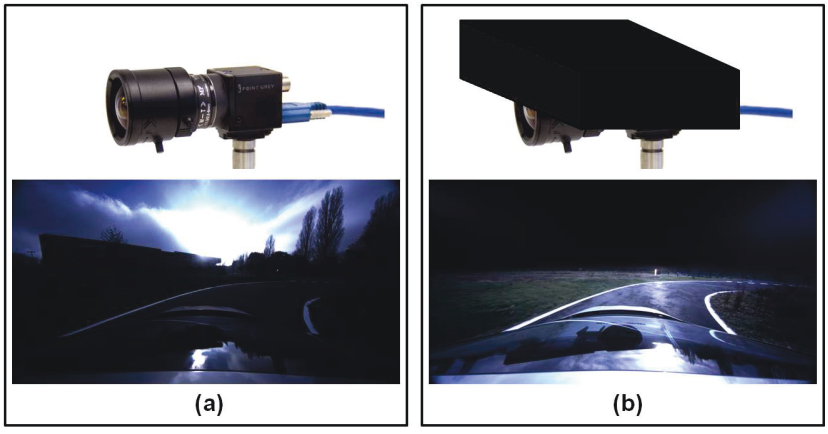


Figure 3.4 – Image from the monocular camera in a sunny day (a) and with the flap protection (b).

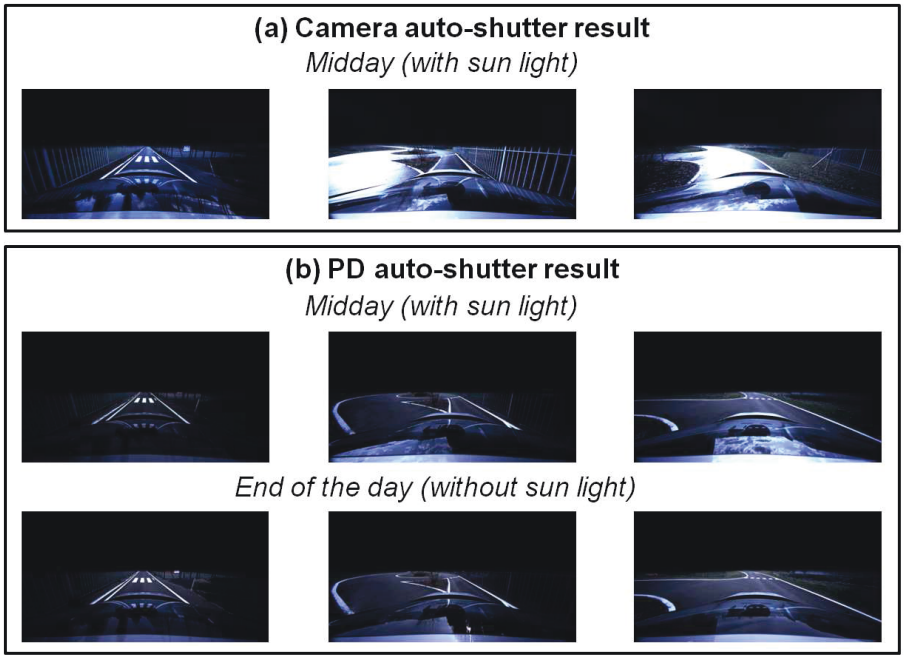


Figure 3.5 – Auto-shutter resulted by the camera software (a) and by the PD controller (b).

Note that the resulting images present minor light changes even in different times of day, important condition for more robust image processing algorithms.

3.3 The robot model

Recalling the Figure 3.2, the robot control inputs are the throttle, brake, and steering actuation setpoint. These inputs are derived from the velocity input set defined as $u_r = [v_1 \ v_2]^T$, where v_1 is the linear velocity of the front wheels and v_2 is the steering velocity. The robot movement are related to its body frame $\{\mathcal{R}\}$ in relation to an inertial frame $\{\mathcal{O}\}$, being recovered by $\dot{q} = [\dot{x_r} \ \dot{y_r} \ \dot{\theta}]^T$. These elements were approximated by the kinematic model of a front-wheel drive car, with the Ackerman's approximation for the steering angle ϕ, as follows (Luca et al., 1998):

$$\begin{bmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi/l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \, . \tag{3.2}$$

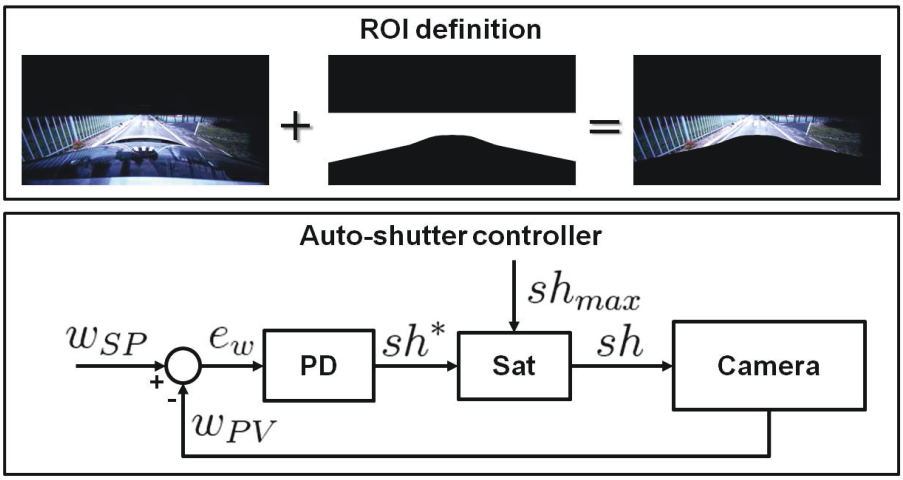


Figure 3.6 – Image ROI definition and the PD auto-shutter control diagram.

The complete diagram with the model variables are illustrated in the Figure 3.7. In this figure, the origin of $\{\mathcal{R}\}$ is located at the midpoint of the two rear wheels, performing circular trajectories related to the instantaneous center of curvature (ICC). The robot orientation θ and steering angles ϕ are positive counter-clockwise, with $\theta \in]-\pi, \pi]$ and $\phi \in [-\phi_{max}, \phi_{max}]$. By this model, the car linear velocity can be calculated from the front wheel velocity as:

$$v = v_1 \cos(\phi), \tag{3.3}$$

and the angular velocity

$$\dot{\theta} = v_1 \cos(\phi)/r_1 = \omega \tag{3.4}$$

is directed related to the steering angle (see the Figure 3.7). These considerations allow us to choose the robot control input as $u_r = [v\ \omega]^T$.

Note that this current model is an approximation used only for fast validation of our navigation method, once it is valid only for low speed applications. This low speed consideration was a security constraint due to the extension of the SEVILLE test track and its sharp turns. For high speed applications, a dynamic model must be considered (Tagne et al., 2015). However, using a dynamic model requires more processing capabilities, caused by the reduced integration time for the system state estimation. This consideration will be left for future works.

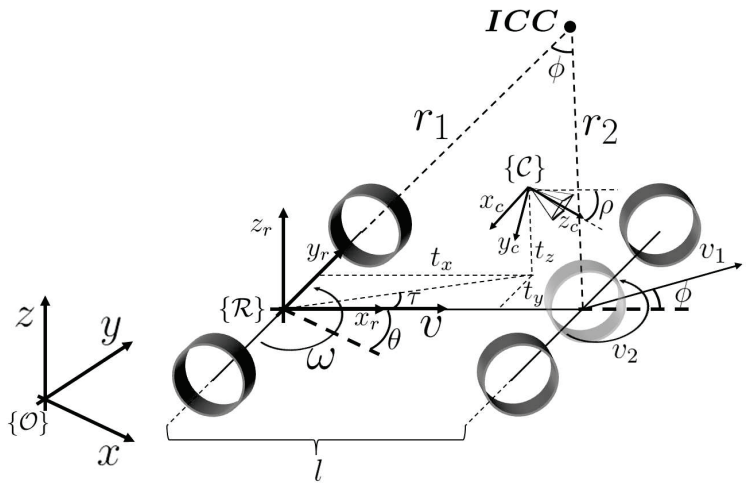


Figure 3.7 – Kinematic model diagram of a front-wheel drive car-like robot centered in the body frame $\{\mathcal{R}\}$. The pinhole camera frame is represented in $\{\mathcal{C}\}$.

3.4 Vehicle localization

The vehicle localization is one of the first components of the system proposed, which is represented by the block EKF in the Figure 3.2. It is the acronym for Extended Kalman Filter (EKF) (Einicke, 2012), used to provide the vehicle state:

$$x = \begin{bmatrix} x_r \\ y_r \\ \theta_r \\ \phi_r \\ v_1 \end{bmatrix}, \tag{3.5}$$

related to the inertial frame $\{\mathcal{O}\}$ in a dead-reckoning process. Its importance to this work is related to the Global Navigation Management, to assure the global task accomplishment.

The low-cost GPS of Figure 3.1 produces several errors while estimating the vehicle position (in a radius of $\approx 5m$), with data mismatches and low rate ($1Hz$). In order to filter these errors and increase the inter-frequency measurements with sensor's fusion, the data from the CAN bus of the vehicle was combined with the GPS position measurements for a better estimation of the car's pose. The chosen data from the CAN bus are the steering angle and the linear velocity from the front wheels $y_{can} = [\phi_r \; v_1]^T$, acquired

at $50Hz$, and the data from the GPS is the position $y_{gps} = [x_r \ y_r]^T$. The EKF model with the sensor's fusion can be defined as follows:

$$\begin{cases} x_{k+1} = f(x_k) + \alpha_k \\ y_{gps,k} = g_{gps}(x(t_0 + j \cdot T_e)) + \beta_{gps,j} \\ y_{can,k} = g_{can}(x(t_0 + j \cdot T_e)) + \beta_{can,j} \end{cases}, \tag{3.6}$$

where α_k, $\beta_{gps,j}$, $\beta_{can,j}$ are the process and observation noises. They are both assumed to be zero mean multivariate Gaussian noises with covariance Q and R.

3.4.1 Prediction

Based on the equations (3.2) and (3.6) for the car and EKF models respectively, the state prediction can be written as:

$$x(k+1) = \begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \theta_r(k+1) \\ \phi_r(k+1) \\ v_1(k+1) \end{bmatrix} = \begin{bmatrix} x_r(k) + v_1(k)\cos(\theta_r(k))\cos(\phi_r(k))\Delta t + \alpha_{x,k} \\ y_r(k) + v_1(k)\sin(\theta_r(k))\cos(\phi_r(k))\Delta t + \alpha_{y,k} \\ \theta_r(k) + v_1(k)(\sin(\phi_r(k))/l)\Delta t + \alpha_{\theta,k} \\ \phi_r(k) + \alpha_{\phi,k} \\ v_1(k) + \alpha_{v_1,k} \end{bmatrix}. \tag{3.7}$$

Once this is a non-linear model, its Jacobian is calculated by:

$$A = \left[\frac{\partial f}{\partial x}\right] = \begin{bmatrix} 1 & 0 & -v_1\sin(\theta_r)\cos(\phi_r)\Delta t & -v_1\cos(\theta_r)\sin(\phi_r)\Delta t & \cos(\theta_r)\cos(\phi_r)\Delta t \\ 0 & 1 & v_1\cos(\theta_r)\cos(\phi_r)\Delta t & -v_1\sin(\theta_r)\sin(\phi_r)\Delta t & \sin(\theta_r)\cos(\phi_r)\Delta t \\ 0 & 0 & 1 & v_1(\cos(\phi_r)/l)\Delta t & (\sin(\phi_r)/l)\Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.8}$$

and applied in the evaluation of the covariance matrix for $\hat{x}_{k|k}$:

$$P_{k+1|k} = A_k \cdot P_{k|k} \cdot A_k' + Q, \tag{3.9}$$

where Q is the matrix with the model noise.

3.4.2 Estimation

The measurements y_{gps} and y_{can} allow linear observation models (given by the functions g_{gps} and g_{can}), where the observation matrix are constants:

$$C_{gps} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } C_{can} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.10}$$

Thus, for each measurement the gain of Kalman is given by:

$$K = P_{k|k-1} \cdot C' \cdot (C \cdot Pk|k-1 \cdot C' + R)^{-1}, \tag{3.11}$$

and the data are estimated as:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K(y_k - C \cdot \hat{x}_{k|k-1}), \tag{3.12}$$

where R is the covariance matrix of the measured data.

This process assures the prediction of the sensor measurements and the estimation of the vehicle pose at 10 Hz, as illustrated in the Figure 3.8 for the SEVILLE test track (Figure 3.3). In this Figure, it is possible to see the low frequency rate of the GPS data, as well as the common odometry cumulative errors in the distance traveled estimation. Note that the EKF corrects the odometry errors and reduce the variations in the GPS data. However, it does not reduce the GPS error and cannot deal well with large time gaps between the data. The initialization of the state also must be close to the real one, for a fast convergence. In this case, the robot orientation could also be recovered from an inertial measurement unit (IMU).

Once there are cameras available in the vehicle, others localization approaches could also be implemented, like the one presented in (Comport et al., 2011), reducing the GPS problems in the final localization. Just like Visual Odometry (Geiger et al., 2011b) is an alternative for problems related to the limited access to the CAN bus of the vehicle.

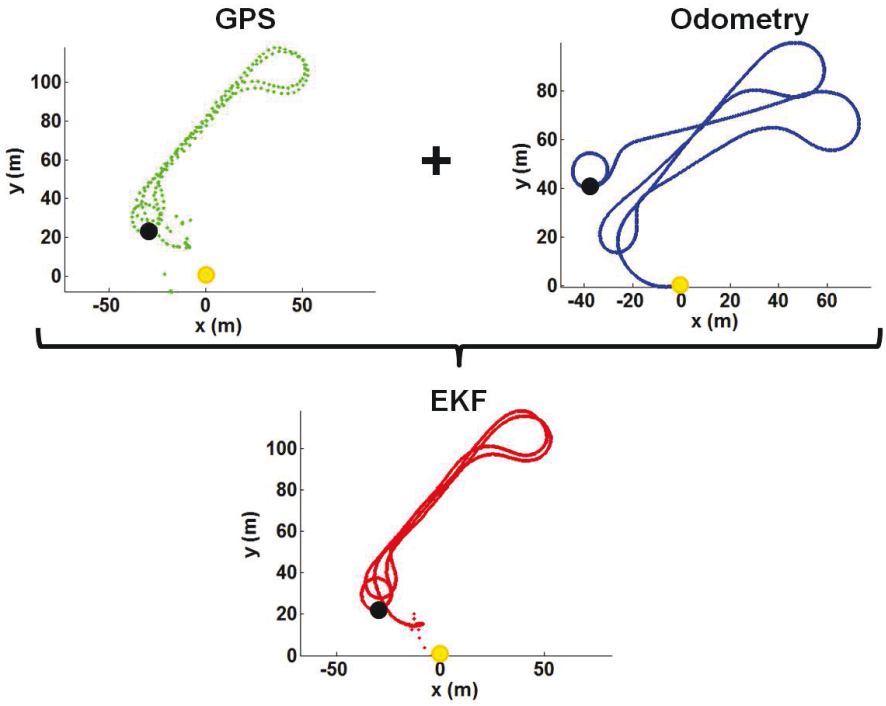


Figure 3.8 – APACHE localization results for a closed-loop movement in the SEVILLE test track using the EKF, based on the data fusion of the GPS and odomentry data. The localization is relative to the initial position (yellow circle) and finishes at the black circle.

3.5 Velocity control

To provide the low-level control actuation setpoint for the throttle, brake, and steering, a velocity control was implemented to regulate the robot model control inputs v_1 and v_2, provided by the local navigation control block (see Figure 3.2). For the linear velocity v_1, a cascade configuration of two PID controllers perform the velocity (outer loop) and acceleration (inner loop) regulation. The velocity was controlled by a proportional-derivative (PD) configuration and the acceleration by a proportional (P) one. The angular position of the steering wheel (ϕ), otherwise, was controlled by the low-level controller block (Figure 3.2) by applying the highest possible speeds ($\simeq 560°/s$). Due to a restrict access to this block, the steering velocity v_2 was regulated by limiting the angular variation with a proportional controller. The complete diagram illustrating these both controllers is in the Figure 3.9. These useful control setup guarantees a better human driver behavior to our final system, restricting high accelerations in the longitudinal control and applying smooth transitions in the lateral control. The

control response of both controllers can be visualized in the Figure 3.10 for different setpoints. The linear velocity control was validated in the SEVILLE test track (Figure 3.3), considering some perturbations caused by sharp turns and up and downhills.

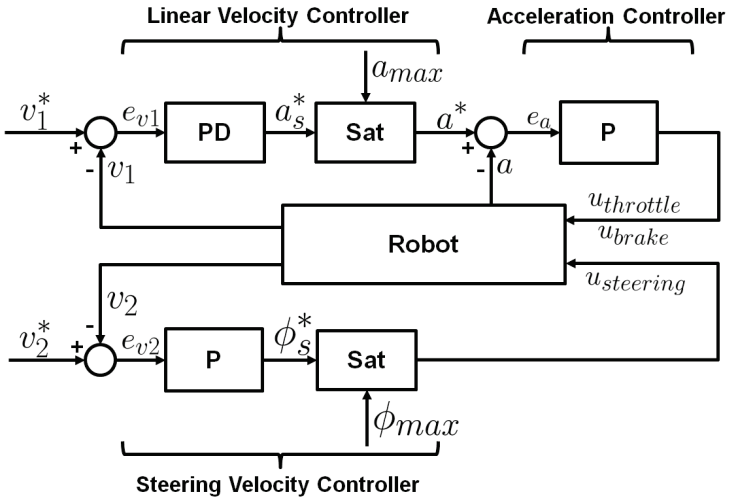


Figure 3.9 – Control diagram for the linear (v_1) and steering (v_2) velocities of a front-wheel drive car.

However, the setpoint values provided to our vehicle are not achieved instantaneously, and the time necessary for stabilization must be considered when performing the local navigation tasks. By applying a small $\Delta\phi$ step with maximum speed actuation in the steering angle, it produces the response in the Figure 3.11. The result has a dead time t_d and time constant τ approximately equal to 0.2s. In total, the system takes around 0.5s to reach the desired setpoint, which was observed for different setpoint values. Thus, the high level system for local navigation must consider that any desired setpoint for the steering angle will take around 0.5s to stabilize.

3.6 The simulation environment

To increase the experimental possibilities of our work and reproduce danger situations that may be unsafe for human drivers in the real world, a simulation environment was created preserving the robots kinematics and some dynamic constraints. It makes use of the Matlab interface to reproduce a car-like robot moving in an urban environment, based on the kinematic model of the equation (3.2). This simulated environment

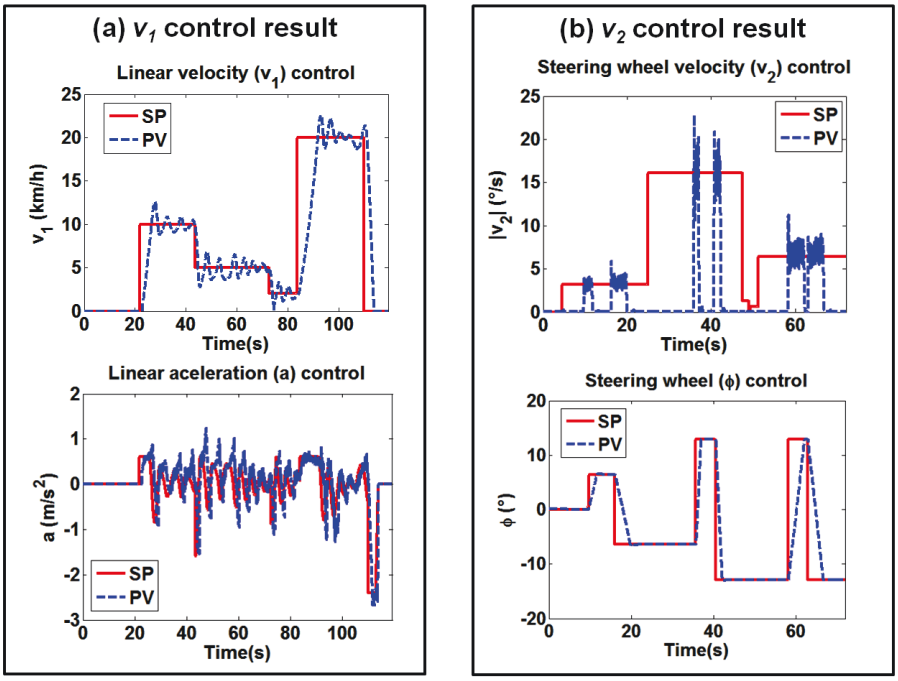


Figure 3.10 – Velocity control response for the experimental vehicle APACHE, where the linear velocity (v_1) is in (a) and the steering velocity (v_2) is in (b). SP is the current control setpoint and PV is the process value.

respects the real scale between the vehicle and the road lanes. A portion of it is illustrated in the Figure 3.12a.

The environment perception was also considered in the conception of the simulation environment. The idea is to reproduce the same sensor-based approach proposed for the vehicle APACHE. Thus, a simulated laser scanner sensor with a white noise was incorporated to detect the obstacles and the road boundaries (Figure 3.12b). It also includes a monocular camera with the same intrinsic and extrinsic parameters of the one installed on the APACHE's roof (Figure 3.12c). In addition, a localization system provides a Gaussian error with the same range of our real GPS.

3.7 Conclusion

Before describing our navigation approach, it is important to understand the system where it will be applied. This chapter briefly presented this system, the experimental car APACHE and some embedded resources. It was based on a commercial car completely

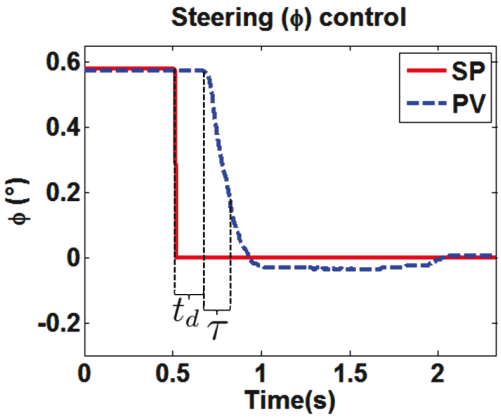


Figure 3.11 – Step response for the steering angle actuator to estimate the dead time t_d and time constant τ. SP is the current control setpoint and PV is the process value.

functional and with only low cost sensors installed. The sensors were chosen to allow an autonomous applications better suited for commercial ends.

The robot movement was reproduced by a kinematic model for low linear velocities experiments, which is enough for validation purposes. The system also includes a localization approach and a low-level velocity control. These are important considerations to allow the global navigation management of the car movement and to accomplish local navigation tasks. Finally, a simulation environment was created to increase the experimental possibilities of our method. It uses the same kinematic constraints and some actuator's dynamics from the APACHE. This simulation platform also includes exteroceptive sensors for perception of the environment and localization of the robot, using the same parameters of the real ones.

The next chapter will start to go into details of the proposed navigation solution. There is presented the environment perception layer for the vehicle APACHE.

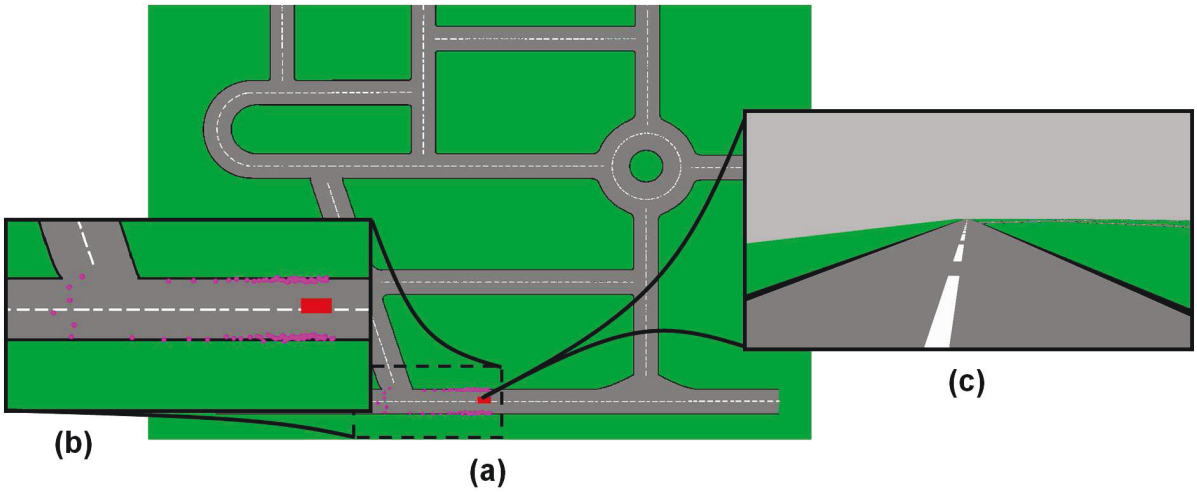


Figure 3.12 – Simulation platform (with a simulated urban environment, vehicle and sensors) to validate the proposed navigation approach in the car-like robot in red (a). The simulated laser scanner measurements are illustrated in pink (b), as well as the image view from a simulated monocular camera with the same parameters of the real one (c).

Chapter 4

Environment Perception

The most important decisions for several autonomous vehicles depend on the information recovered from the environment. Some perception capabilities applied to autonomous cars were presented in the DARPA Grand Challenges, competitions promoted by the American's DARPA between 2004 and 2007. Today, with the diffusion of autonomous and semi-autonomous vehicles, a great number of new applications for environment perception have emerged. However, for real applications, the number of sensors used and the final cost must be considered for viability. Based on the principle of viability, vision systems has some advantages, because they can provide large amount of data with a low cost, depending on their FOV and resolution.

This chapter presents the environment perception approach for the experimental vehicle APACHE (Figure 3.1) using vision systems. It is based on our contributions in 2D/3D image processing (Vitor et al., 2013; Lima et al., 2013) to minimize the common urban environments problems caused by shadows, road texture, light variations, etc. First a short review on stereo vision systems is proposed in Section 4.1. Then, our contributions on 2D/3D image processing using 2D segmentation, are presented in Section 4.2, with applications on road detection and disparity map refinement. Thus, the current solutions for road/obstacle detection and image feature extraction are described in the Sections 4.3 and 4.4 respectively. As an extension of the vehicle FOV, the road and obstacles detected were projected in a local occupancy grid, explained in the Section 4.5. The experimental validation of our approaches in a real urban scenario is shown in the Section 4.6. Finally, conclusions are proposed at Section 4.7.

4.1 Stereo vision systems

The most part of stereo vision systems are based on two (binocular) or more displaced cameras with synchronous image data. They are normally passive systems, where the 3D information of the environment must be extracted combining their image data in a stereo matching problem. The active ones, otherwise, apply a light in the environment (e.g. laser or structured light) to simplify the matching problem. However, the concepts around the 3D information projection are similar in both systems and can be found in (Faugeras, 1993).

Focusing on passive systems, to acquire the 3D information of the scene, synchronous images must be taken at first. These images are so processed to remove some distortions, mainly caused by lens imperfections. In order to reduce the processing time during the stereo matching, a rectification step is performed, based on the epipolar geometry (Faugeras, 1993). This geometry guarantees that the similar information, in both images, is related to the same row (v). The resulting images are then processed with a stereo matching algorithm to provide the disparity map ($\mathcal{I}_{\triangle}$), which the intensity values corresponding to the distances (in pixels) between the corresponding points from both images. The $\mathcal{I}_{\triangle}$ values are related to the distance between the camera and the point in the world, given by:

$$z_c = \frac{fB}{d}, \tag{4.1}$$

where f is the focal length of the camera, B is the camera baseline, and d is the disparity value.

There are several stereo matching algorithms available to create the $\mathcal{I}_{\triangle}$ (Geiger et al., 2011a), classified between local and global ones. The local algorithms typically combine image statistics in a small window to find the best corresponding point in the row (epipolar line). One example of local approach is the Sum of Absolute Differences (SAD) (Hiroshi et al., 1995), which produces a fast result with a lot of mismatches. In the other extremity are the global matching techniques (Kolmogorov and Zabih, 2001), considering the entire image during the matching process. This group of techniques produces good results with a high computational cost. As intermediate solutions, there are the semi-global matching techniques (Hirschmuller, 2005) to expands the line re-

search in different local directions. They have satisfactory results for fast embedded solutions (Banz et al., 2010), better suited for ADAS applications.

Due to its fast calculation, this work used the OpenCV Block Matching (BM) function, an implementation of the SAD algorithm for the stereo matching. However, the semi-global technique Semi-Global Block Matching (SGBM) (Hirschmuller, 2005) was also used to evaluate the disparity refinement contribution, which is explained below in Subsection 4.2.2. The resulting $\mathcal{I}_{\triangle}$ are shown in the Figure 4.1.

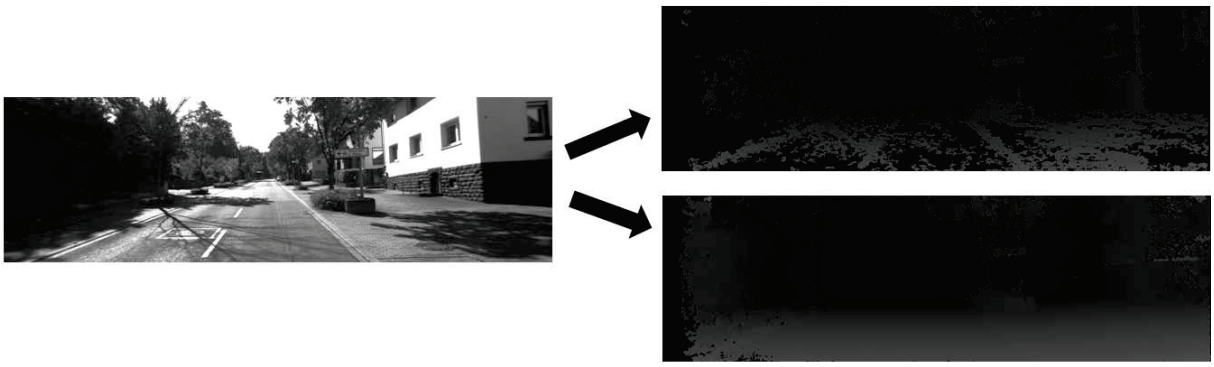


Figure 4.1 – Example of disparity maps calculated by the stereo matching algorithms BM (right top) and SGBM (right bottom), for the reference image from the KITTI flow benchmark (Geiger et al., 2012) on the left.

Several problems can be observed on the $\mathcal{I}_{\triangle}$ images of Figure 4.1, mainly caused by specific elements of the urban environments, such as shadows, light reflections, and low texture variations. These elements hamper the right pixel matching in the stereo pair. Some of these matching problems are detected and eliminated as black points (0 value) in the $\mathcal{I}_{\triangle}$. The remaining ones must be considered in further implementations, like the 2D/3D image processing approaches presented in the following sections.

4.2 2D/3D image processing based on 2D segmentation

There are many problems related to the stereo vision systems in urban environments (see Section 4.1). Therefore, the road and obstacles detection becomes a hard task using only sparse disparity information (Labayrade et al., 2002), like the one from Figure 4.1. However, once the $\mathcal{I}_{\triangle}$ is constructed from a 2D reference image, a lot of information

from this image can be added to improve the final detection problem (Soquet et al., 2007). This information can also be used to enhance the $\mathcal{I}_{\triangle}$ (Xu et al., 2012).

Image segmentation is a powerful tool for image analysis, once it combines near pixels with similar information which could belong to the same object in the world. The formed segments, also known as clusters or super pixels, can be used as a simplification of the original image. How close the segments are to the boundaries of the real objects, better they represent the environment. Giovani B. Vitor proposed in (Vitor et al., 2013) an image segmentation applying the Watershed transform based on local elements, named as Local Condition Watershed Transform (LC-WT) (Audigier and de Alencar Lotufo, 2010). It was performed after some preprocessing steps and the result is illustrated in the Figure 4.2. With this segmentation, we combined 2D and 3D image data to perform the road detection (Vitor et al., 2013) and the disparity map refinement (Lima et al., 2013). They are presented in the following subsections.

Figure 4.2 – Examples of image segmentation based on the Watershed transform (Vitor et al., 2013), using the KITTI flow benchmark (Geiger et al., 2012) (on the left) and the same stereo vision camera from APACHE (on the right).

4.2.1 Road detection

The principle beyond combining the 2D and 3D image data of the scene during the road detection is to complement the dubious information from each side. Considering the segmentation using the Watershed transform (see Figure 4.2), the idea proposed in (Vitor et al., 2013) was to classify each segment in road surface or non-road surface (obstacles, sky, etc.) based on a features descriptor. To do so, the entire area of the

segment was applied in a 2D and 3D features extraction, enhancing the sparse disparity information (of Figure 4.1).

The 3D features were calculated from the planar world approximation on the V-Disparity map ($\mathcal{I}_{V\triangle}$), presented by (Labayrade et al., 2002). In this approach, the $\mathcal{I}_{V\triangle}$ is segmented in small horizontal and vertical planes representing the drivable (free) and non-drivable (obstacle) areas. The selected features were the percentage of free, obstacles, and non-classified (without disparity) data in the area defined by the cluster. The 2D features used some statistical calculations from (Shinzato and Wolf, 2011), providing the mean, entropy, and variance of the Red, Green, and Blue (RGB) color space and the Hue, Saturation, and Value (HSV) representation of every image cluster. Then, an ANN was trained and applied to classify the clusters in road surfaces, non-road surfaces, and unknown areas.

The final solution was tested and validated in urban environment, with different light conditions and structures. Based on a personal dataset, a ground truth was created by classifying some Watershed segments in road surface or not, for different images conditions. The final solution reached an accuracy of 93.30%. Some results are replicated in the Figure 4.3, showing the robustness in presence of many different obstacles such as vehicles, pedestrians, trees, and road barriers. The same is perceived in shadow areas and sun light reflection. It was also evaluated in the KITTI vision benchmark suite (KITTI) (Geiger et al., 2012), presenting a recognition performance of 80.95% in non-sequential images. A complete analysis between this technique related to the new ones presented in the following sections is performed in the Section 4.6.

Although this approach was able to classify different urban scenarios, the processing time related to the image segmentation, features extraction, and ANN classification can compromise real time applications. This mainly caused by the number of clusters detected, which are variable, and the algorithm implementation, not optimized. In addition, the V-Disparity map also has some limitations for obstacle detection, which could be improved in the 3D features extraction. Finally, the ANN classifier requires a training step for every new scenario and a hard human work creating the dataset and ground truth for each image cluster.

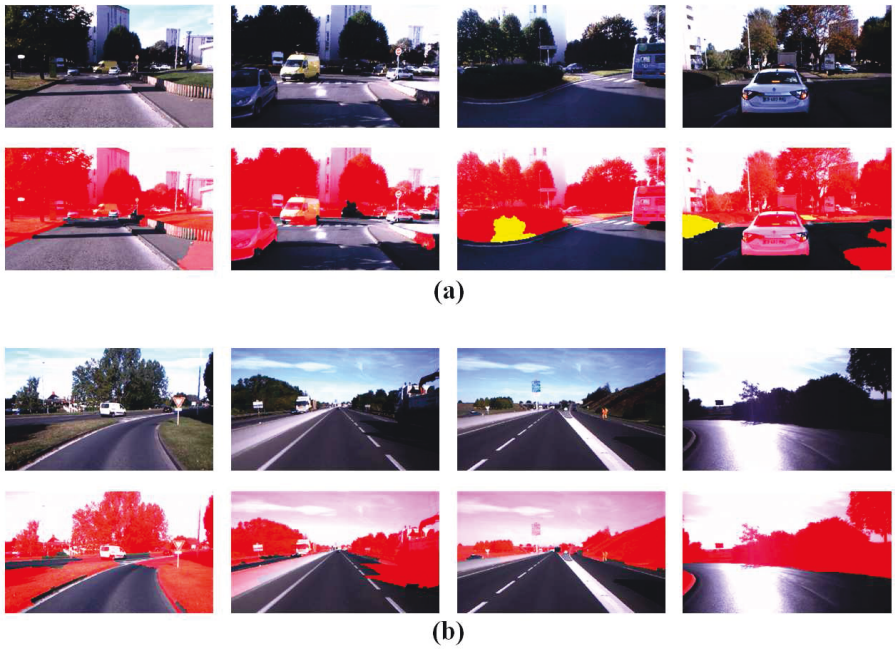


Figure 4.3 – Road segmentation result from the 2D/3D vision based approach (Vitor et al., 2013) for urban road (a) and highway (b) scenarios. The original image is on top and the segmented result is on bottom, where the original color represents the road surface, in red color the non-road surfaces, and yellow the unknown areas.

4.2.2 Disparity map refinement

As a solution to enhance the 3D information from stereo matching algorithms, the idea presented in (Lima et al., 2013) combined the image segmentation using the Watershed transform (see Figure 4.2) with the planar world approximation similar to (Labayrade et al., 2002). The $\mathcal{I}_{\triangle}$ refinement aims to reduce the inter-pixels variation and the missing disparity information (as shown in the Figure 4.1), by approximating the formed clusters into small planes. For each cluster, all disparity values were applied into a RANSAC based plane estimation. Thus, all points were projected in the plane and filtered by considering:

— The point distance to the plane;

— The percentage of valid points on the cluster;

— The maximum number of RANSAC interactions; and

— The difference between the original and the projected point.

The results were validated with the KITTI (Geiger et al., 2012), for the stereo algorithms BM and SGBM illustrated in the Figure 4.1. For the BM, where the average density of the $\mathcal{I}_{\triangle}$ was less than 50%, the final result was enhanced to 77.24%, without

introducing significant error in the final disparity value. However, in the case of the SGBM refinement, where more pixels are available to estimate the plane, the final $\mathcal{I}_{\triangle}$ had enhanced the density from 87.31% to 97.80% reducing the final disparity value error too. This new information also improved the visualization of small size elements, such as the difference between the road and the side-walks, once the error related to the pixel variation was reduced. Some qualitative outcomes from this approach are reproduced in the Figures 4.4 and 4.5

Figure 4.4 – Final result for the disparity map refinement of the BM method in different urban scenarios.

Similarly to the road detection approach (at Subsection 4.2.1), the $\mathcal{I}_{\triangle}$ refinement also presented a variable computational time related to the image segmentation and RANSAC planes approximation. It also added a new step before perform the road and obstacle detection. In practical situations, this could compromise final control applications which require a regular cycle time.

Due to that, the environment perception for the vehicle APACHE will focus on the road and obstacles detection based on 2D/3D image processing, speeding up the segmentation and classification process. This includes the estimation of the road surface directly on the 2D image and then combining it with the 3D information. The road surface estimation will also be applied to detect the features for the local navigation control. The next sections will describe these approaches.

Figure 4.5 – Final result for the disparity map refinement of the SGBM method in different urban scenarios.

4.3 Road and obstacles detection

The road and obstacle detection was based on the combination of 2D/3D image processing, similarly to the concept presented in (Vitor et al., 2013), and shortly described in the Section 4.2. However, the 2D image processing was adapted from the one detailed in (Miranda Neto et al., 2013), to reduce the processing time, and the 3D image processing was improved with the U/V Disparity maps for better estimate drivable and non-drivable areas. In addition, the final classification was simplified. The Figure 4.6 illustrates how the proposed solution was divided. It is presented in the following subsections.

4.3.1 2D image processing

The 2D image processing aims to estimate a possible road surface in the image frame, which will be validated with the 3D information in the classification step (Subsection 4.3.3) and used to detect the road lane center (Subsection 4.4.2). It requires a preprocessing to intensify the visual information and a segmentation to distinguish it from the rest, as presented as follows.

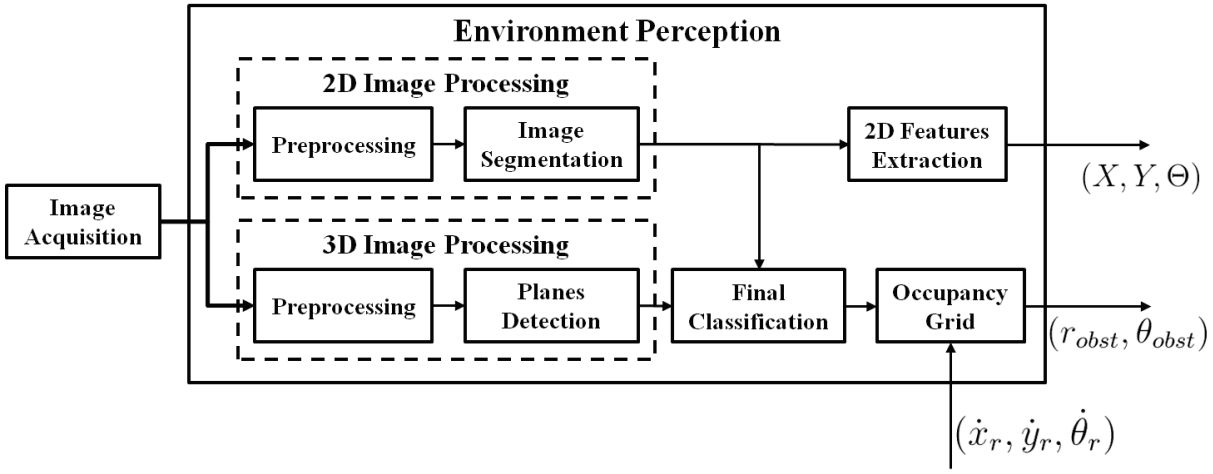


Figure 4.6 – Block diagram for the environment perception solution.

Preprocessing

There are several robust image processing algorithms to enhance the desired information to be detected (Bertozzi et al., 2000). In this work was applied a Gaussian filter (similar to a low-pass frequency filter) and an image resizing to smooth and reduce the processing time (Gonzalez and Woods, 2001). Thus, the final analyzed image has 120 rows proportionally. It is also important to consider only the regions with possible road information, by defining a ROI in the image frame and removing the car's hood and a portion of the sky. The resulting image is in the Figure 4.7a.

Once color images have substantial information about road surface, grass, and road marks, the RGB color model was considered in the final evaluation. However, differently from (Miranda Neto et al., 2013) where a single RGB channel was chosen for the segmentation step, we have based our segmentation in three preprocessed images. The first one is an enhancement of the blue channel $\mathcal{I}_B$ information, main component color in the asphalt, by removing the other two channels ($\mathcal{I}_R$ and $\mathcal{I}_G$) from $\mathcal{I}_B$ (see Figure 4.7d):

$$\mathcal{I}_{B^+} = (\mathcal{I}_B - \mathcal{I}_G) + (\mathcal{I}_B - \mathcal{I}_R). \tag{4.2}$$

The second image adds the lane marks information to the final segmentation. Considering white lane marks, they can be emphasized by equalizing the original image (Figure 4.7a) and selecting the saturated color region $\mathcal{I}_S$ (all pixels > 250), as shown in the Figure 4.7c. The last one uses the hue channel $\mathcal{I}_H$, from the HSV representation,

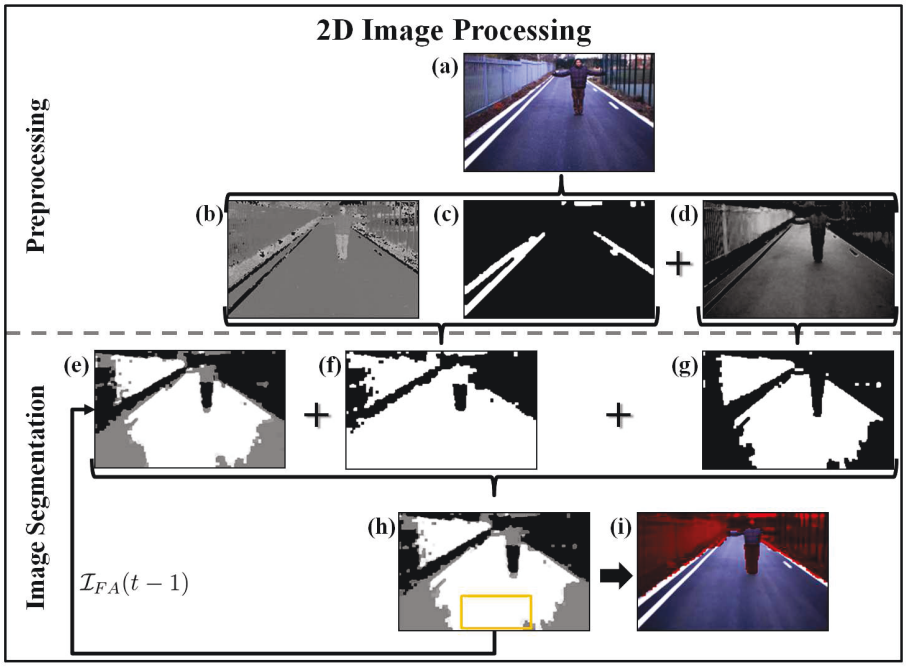


Figure 4.7 – The 2D image processing steps to estimate the road surface area in (i). The preprocessing is performed in the original image after resizing and selecting the ROI (a). Thus, the three preprocessed images related to the hue channel $\mathcal{I}_H$ (b), the saturated color region $\mathcal{I}_S$ (c), and the blue channel $\mathcal{I}_{B^+}$ are generated. The segmentation of these images result in $\mathcal{I}_{HSw}$ (f) and $\mathcal{I}_{Bw}$ (g). Finally a weighted sum between the previous estimation $\mathcal{I}_{FA}(t-1)$ (e), $\mathcal{I}_{HSw}$, and $\mathcal{I}_{Bw}$ gives the current free area estimation $\mathcal{I}_{FA}$ (h). The yellow rectangle in the image in (h) represents the region with higher percentage of navigable area, used to find (i).

to improve the robustness of the present approach to light variations, once only color information are presented in this channel (Figure 4.7b).

Image segmentation

The original proposition of (Miranda Neto et al., 2013) was based on the combination of a color segmented image from the RGB, using the Otsu method (Otsu, 1979), and an image formed from an edge and line finders. Although these considerations are good for large interclass variations, e.g. between road surfaces and herbs, dealing with urban environments many mismatches can appears due to light variations and road marks. To avoid these problems, the previous formed images $\mathcal{I}_{HS} = \mathcal{I}_H + \mathcal{I}_S$ and $\mathcal{I}_{B^+}$ were segmented. Applying a linear threshold, it results in the images $\mathcal{I}_{HSw}$ and $\mathcal{I}_{Bw}$, as the ones in the Figure 4.7f-g.

In order to estimate a multimodal 2D drivability free-area, $\mathcal{I}_{FA}$, which represents the road surface, a weighted average of the segmented images intensities was performed. It is described in equation (4.3).

$$\mathcal{I}_{FA}(t) = 0.2\mathcal{I}_{FA}(t-1) + 0.4\mathcal{I}_{HS}(t) + 0.4\mathcal{I}_{B^+}(t), \tag{4.3}$$

where $\mathcal{I}_{FA}(t-1)$ is the previous multimodal image calculated. By using $\mathcal{I}_{FA}(t-1)$, small variations between the previous $(t-1)$ and the current (t) reference frame can be used to improve the final result. However, to avoid the problems associated to large variations in these frames, the weighting parameters (0.2, 0.4, 0.4) were set to maintain only the repeating data. The resulting image $\mathcal{I}_{FA}$ is illustrated in the Figure 4.7h. The final segmentation is performed in the $\mathcal{I}_{FA}$, where the navigable area is composed by all segments belonging to the yellow rectangle defined in the bottom of the image (as seen in the Figure 4.7h), resulting the Figure 4.7i.

4.3.2 3D image processing

At the same time that the 2D processing performs the road segmentation, the 3D processing classifies the reference image in free space and obstacles (see Figure 4.6). This is acquired by means of the U/V disparity maps (Labayrade et al., 2002) and the approximation of the environment to small planes.

Preprocessing

The processing starts with the rectification of the stereo pair acquired from the cameras (Figure 4.8a), using the epipolar geometry (Faugeras, 1993). The rectified images are applied to the correlation algorithm SAD to built the disparity map ($\mathcal{I}_{\triangle}$), illustrated in the Figure 4.8b. The closest regions to the camera are in light grey and far way in dark grey. These grey values represent the disparity for each pixel and they are related to the distance between the camera and the 3D point in the world. In this figure, the influence several noise (shadows, light reflection, and low texture variance) produce a sparse $\mathcal{I}_{\triangle}$ with no disparity values (black points).

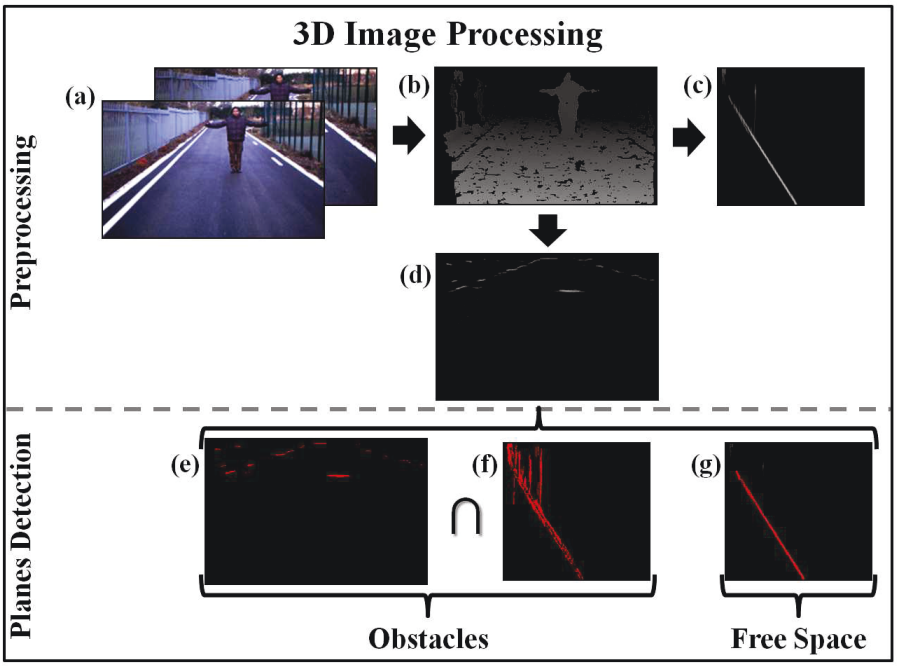


Figure 4.8 – The 3D image processing steps for the rectified stereo pair in (a). The preprocessing is performed to create the disparity map $\mathcal{I}_{\triangle}$ (b) and its row/column representations $\mathcal{I}_{V\triangle}$ (c) and $\mathcal{I}_{U\triangle}$ (d). Approximating the environment by small planes, the U/V disparity maps are classified in free space (g) and obstacles, by the intersection of (e) and (f).

The disparity map ($\mathcal{I}_{\triangle}$) is converted in a V and U disparity maps ($\mathcal{I}_{V\triangle}$ and $\mathcal{I}_{U\triangle}$) (Soquet et al., 2007), respectively in the Figures 4.8c and 4.8d. Basically, the $\mathcal{I}_{V\triangle}$ is computed accumulating the similar disparities in the same image row (v). In this image, the rows are the same of the $\mathcal{I}_{\triangle}$, the columns represent the disparity values (grey scale), and the intensity value store the number of pixels with the same disparity in the row analyzed. Generalizing, it is a composition of row histograms of the $\mathcal{I}_{\triangle}$. The $\mathcal{I}_{U\triangle}$ is calculated similarly to $\mathcal{I}_{V\triangle}$, changing row to columns in the previous definition, being a composition of column (u) histograms of the $\mathcal{I}_{\triangle}$.

Planes detection

The U/V disparity technique was chosen to allow an easy classification of the image data in free space and obstacles. As described in (Lima et al., 2013), the information on the world environment can be approximated by small horizontal and vertical planes in the disparity map ($\mathcal{I}_{\triangle}$) in a non-flat world representation. This means that free spaces have linear variations on their disparity value in a short horizon, whereas the obstacles

have almost the same disparity values. These both effects generate high intensity values in the $\mathcal{I}_{V\triangle}$ (for free spaces) and $\mathcal{I}_{U\triangle}$ (for obstacles). In this context, the noisy pixels are removed by a threshold and only the most significant ones are considered in the final classification process (Soquet et al., 2007).

The free space are classified as small lines with a slope bigger than $90°$ and locally refined for each $\mathcal{I}_{V\triangle}$ row. The final free space segmentation is represented in the Figure 4.8g. The obstacles candidates in the $\mathcal{I}_{V\triangle}$ are all the other pixels with no classification (Figure 4.8f). To confirm which candidates are real obstacles, the $\mathcal{I}_{U\triangle}$ was similarly segmented in the Figure 4.8e. This $\mathcal{I}_{U\triangle}$ segmentation intersected with the previous $\mathcal{I}_{V\triangle}$ candidates gives us the obstacles mapping, used in the final classification explained in the Subsecion 4.3.3 below.

4.3.3 Final classification

Once processed the 2D and 3D layers of Figure 4.6, the reference image may be finally classified in road, obstacles and unknown areas. Differently from (Vitor et al., 2013), where a Neural Network was trained to perform this classification, here the final result was acquired by intersecting both 2D and 3D information to speed up the process. The Figure 4.9 describes the proceedings to acquire the final segmented image.

The 3D Image Processing (Subsection 4.3.2) has detected the obstacles (Figure 4.8e-f) and free space (Figure 4.8g) in the $\mathcal{I}_{V\triangle}$ and $\mathcal{I}_{U\triangle}$. By converting back these images in the $\mathcal{I}_{\triangle}$ (Figure 4.8b), the resulting masks of obstacles and free space are obtained in the Figure 4.9a-b. The 2D Image Processing (Subsection 4.3.1) has performed the road segmentation (illustrated in the image mask of Figure 4.9c). Thus, the final road surface is defined as the intersection between the free space and the road segmentation (illustrated by the original color in the Figure 4.9d). The final obstacles are all the obstacles previously classified from the 3D processing which are not in the final road surface area (represented by the red color in the Figure 4.9d). The unknown area is represented by every pixel without classification or disparity information (shown by the yellow color in the Figure 4.9d).

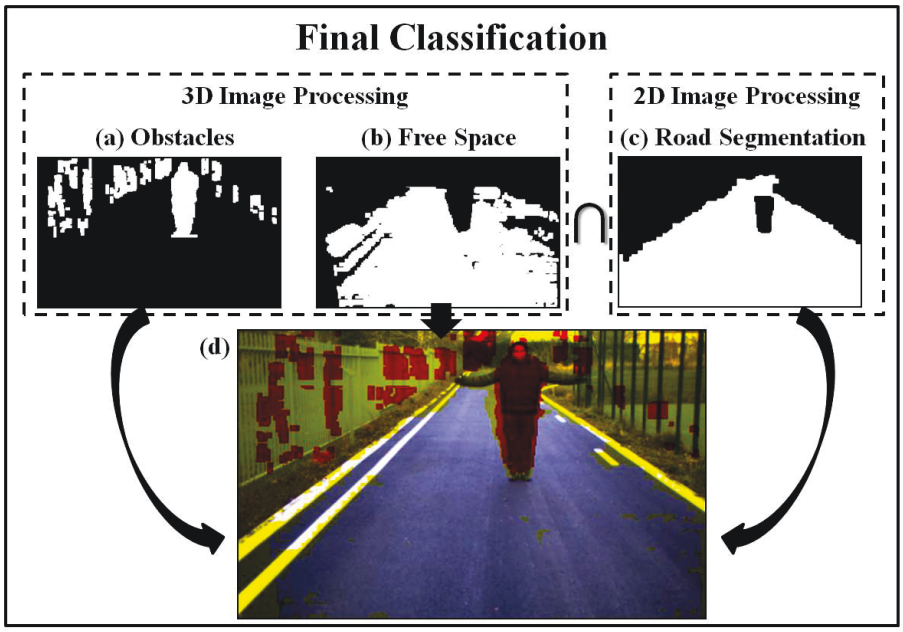


Figure 4.9 – Final classification of the road surface and obstacles using the 2D/3D image processing approach in the reference image (d). The detected obstacles (a) are mapped in red. The road surface is represented in the original color, formed by the intersection of the free space from the 3D processing (b) and the road segmentation from the 2D processing (c). The unknown area (without classification or disparity information) is in yellow.

4.4 Image feature detection for visual navigation

4.4.1 Features description

The sensor-based control of our vehicle was conceived for monovision cameras, where image features are used to calculate control velocities in a visual servoing approach. To achieve that, the robot described in the Subsection 3.3 also includes a fixed pinhole camera, represented by the reference frame $\{\mathcal{C}\}$ at the Figure 3.7. The optical center position in the robot frame is given by $(x_c, y_c, z_c) = (t_x, t_y, t_z)$, with z_c parallel to x_r and a constant tilt offset $0 < \rho < \frac{\pi}{2}$ related to the x_r axis. The image frame $\{\mathcal{I}\}$ for the camera at this configuration is illustrated in the Figure 4.10, where (u_{max}, v_{max}) and $(2X_{\mathcal{I}}, 2Y_{\mathcal{I}})$ are the image size in pixels and in normalized perspective respectively.

This image defines a path once differentiable in $\mathbb{R}^2$ on the road lane center P between the boundaries δ_1 and δ_2. For path continuity, the vehicle must always see the lane. In our visual servoing approach for path reach and following, the image features are the line (X, Y, and Θ) formed by the tangent Γ of the path P at the point

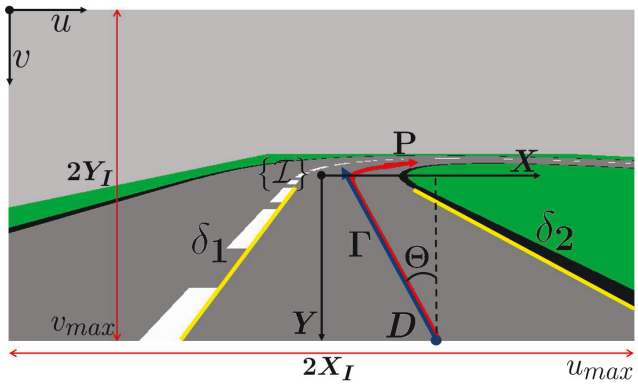


Figure 4.10 – Image frame $\{\mathcal{I}\}$ representation for the camera frame $\{\mathcal{C}\}$. The road lane center P (in red) is related to the boundaries δ_1 and δ_2 (in yellow). Its tangent Γ (in blue), at the point D and angle offset Θ from Γ to the axis $-Y$, defines the image features X, Y, and Θ.

$D = (X, Y)$. The angular offset $\Theta \in]-\pi, \pi]$ is from Γ to the axis $-Y$ (positive counter-clockwise).

Due to the camera's tilt offset ρ and the planar surface constraint, an image point $(u,\ v)$ can be easily projected on the road plane with relation to the robot frame, by using the homogeneous transformation from $\{\mathcal{C}\}$ to $\{\mathcal{R}\}$. For a pinhole camera model, with the intrinsic parameters focal length $(f_x,\ f_y)$ and image center $(c_x,\ c_y)$ in pixels, the extrinsic parameters ρ and t_z, and the normalized perspective

$$\begin{cases} X = \frac{u}{u_{max}-1} - c_x = \frac{x_c f_x}{z_c} \\ Y = \frac{v}{v_{max}-1} - c_y = \frac{y_c f_y}{z_c} \end{cases}, \tag{4.4}$$

the mapping between $\{\mathcal{I}\}$ and $\{\mathcal{C}\}$ is calculated by:

$$\begin{cases} x_c = \frac{X t_z}{\sin\rho + Y\cos\rho} \\ y_c = \frac{Y t_z}{\sin\rho + Y\cos\rho} \\ z_c = \frac{t_z}{\sin\rho + Y\cos\rho} \end{cases}. \tag{4.5}$$

There are no singularities in these equations caused by a zero on the denominator, once the planar projection is limited to $Y > -\tan\rho$. The intrinsic parameters $(f_x,\ f_y,\ c_x,\ c_y)$ were acquired from the camera calibration.

4.4.2 2D features extraction

As described in the Figure 4.6, the image once segmented is applied to a 2D features extraction component. These features are selected to be the reference to the local navigation approach based on visual servoing, as presented previously in the Subsection 4.4.1.

The principle beyond the present approach is the same of several lane detection algorithms, once the road is first segmented and the lane (or boundaries) are fitted to lines or curves (Beyeler et al., 2014). Following the Figure 4.11, the previous road segmentation (Figure 4.11a) are processed with an edge finder (Gonzalez and Woods, 2001) and combined with the saturated color region $\mathcal{I}_S$ (see Subsection 4.3.1) to form the Figure 4.11b. The boundaries candidates are detected in the Figure 4.11c and the reference line is fitted by the RANSAC approach (Fischler and Bolles, 1981) resulting the Figure4.11c.

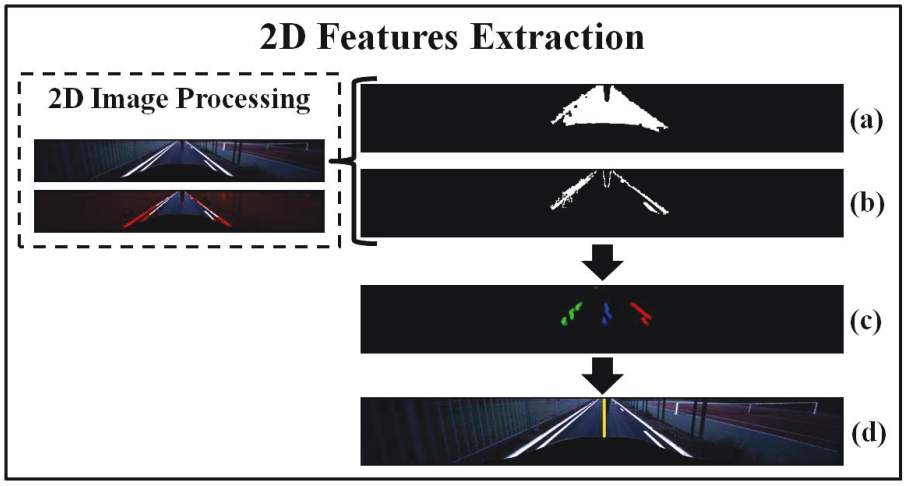


Figure 4.11 – The 2D features extraction process to detect the reference road center line (d) for visual servoing applications. For the road segmented image in (a), the image edges are combined with the white segments available to form (b). It is used to estimate the lane boundaries candidates in (c) and to fit a line by the RANSAC approach in (d).

4.5 Occupancy grid

The final classification presented in the Section 4.3 was simplified to speed up the road and obstacle detection process. However, this increases the occurrence of false positives that could compromise the robot navigation. Furthermore, the sensors used

have a limited FOV to the front of the vehicle, which need to be enhanced to allow the obstacle avoidance. For that, the local occupancy grid (Elfes, 1989) was chosen to store the detected obstacles and road surface and also perform the data fusion for the vision cameras.

Based on (Nguyen et al., 2012), the obstacles and road surface from the stereo vision camera (Figure 4.9d) were projected in the occupancy grid, as well as the road segmentation from the monocular camera (Figure 4.11a). In the case of the monocular camera, the projection was performed using the equations (4.4) and (4.5) in a short horizon ($12m$), due to possible calibration errors and the planar workspace approximation. The use of monocular camera data aims to extend the APACHE FOV during the road intersection maneuvers. The projection, in both camera's systems, respect a Gaussian distribution model with the standard deviation σ related to the 3D point estimation error. Thus, the occupancy probability of one cell c_i at one 3D measurement $P_c = (x_c, y_c, z_c)$ in the time t is given by:

$$P_t^{occ}(c_i, P_c) = k \cdot \oint_{A_{cell} \bigcap 3D} \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2} \left(\frac{P_c - P_{cell}}{\sigma} \right)^2, \tag{4.6}$$

where P_{cell} is the cell position, k represents the percentage of contribution of a single 3D point in the occupancy level of a cell, and $A_{cell \bigcap 3D}$ is the integral of the distribution over the area that overlaps with the cell c_i. In addition, if c_i has the influence of more than one 3D point, the final occupancy probability is:

$$P_t^{occ}(c_i) = \sum_{all P_{c_i}} P_t^{occ}(c_i, P_c). \tag{4.7}$$

These equations were used to represent the detected obstacles in the occupancy grid. For the detected road from the stereo and monocular cameras, the occupancy probability is calculated with the same formulation of Eq. (4.6), but inverted to assign zero probabilities to the occupancy grid:

$$P_t^{free}(c_i) = 1 - \sum_{all P_{c_i}} P_t^{occ}(c_i, P_c). \tag{4.8}$$

The final projection of both camera's systems are illustrated in the Figure 4.12, for the robot oriented to the right (red rectangle).

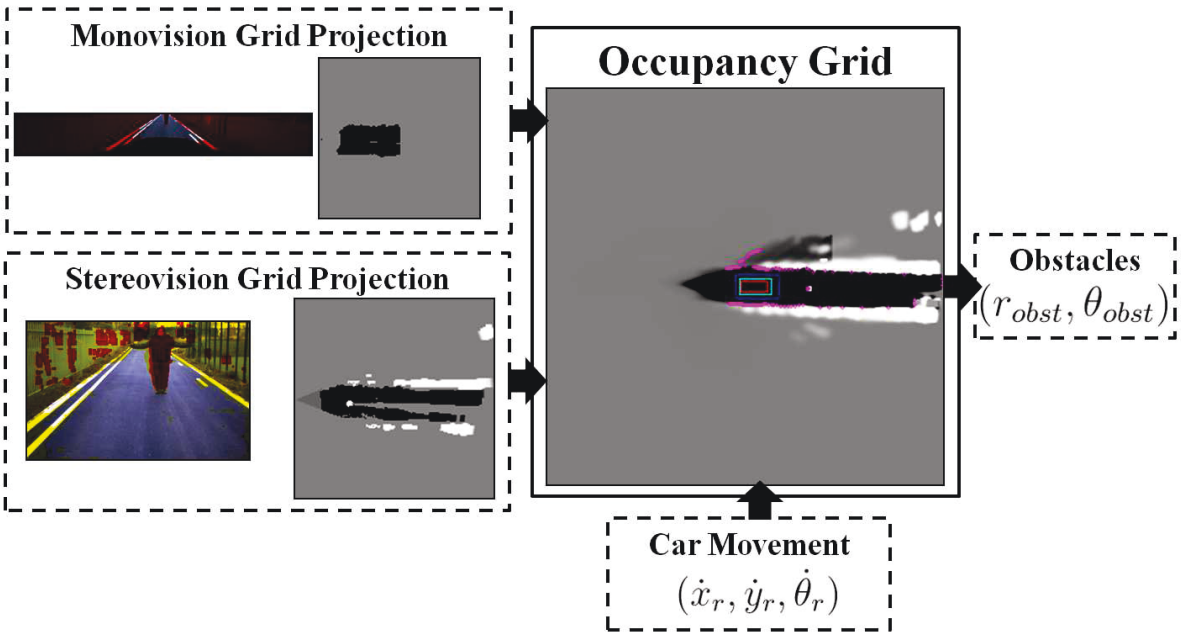


Figure 4.12 – Local occupancy grid projection based on the stereo and monocular vision data for road and obstacles, for the vehicle (red rectangle) oriented to the right. The light and dark blue rectangles represent an expansion of the vehicle size for security reasons. The car relative movement is also used to locally update the grid. The pink dots illustrate the effective obstacles detected for the later autonomous navigation steps.

Once only the local information are held in the grid, it was centralized on the robot frame $\mathcal{R}$ and updated with the car relative movement. To minimize detection mistakes and cumulative localization errors (caused by kinematic approximations), a forgetting term was applied to the grid in order to clear regions with no sensor reading after some time interval.

Since defined on the grid, the effective obstacles and navigable area limits related to the robot are detected and later applied in the navigation steps. They are represented by the pink dots in the Figure 4.12, which means the radius distance (r_{obst}) to the end of the free area at the orientation angle (θ_{obst}), in the robot body frame $\{\mathcal{R}\}$. This figure also shows an expansion of the robot dimensions (light and dark blue rectangles) used for security reasons during the navigation steps.

4.6 Experimental results

Some results have already been shown in the previous sections illustrating the methods. Here we will focus on the validation of the present approach with quantitative and qualitative outcomes. The quantitative analysis will evaluate the performance of this approach in an urban dataset, the same used in (Vitor et al., 2013). The qualitative results present some frames processed by our embedded approach on the experimental vehicle APACHE at the SEVILLE test track.

4.6.1 Quantitative results

The dataset is formed by a sequential video with 3704 frames, processed at the final resolution of (400x300). We are applying this sequential video instead of the widely used KITTI (Geiger et al., 2012), once the algorithm proposed for image segmentation uses the past information contained in the image $\mathcal{I}_{FA}(t-1)$ (see Eq. 4.3), and KITTI provides non-sequential frames. Our set contains several samples of unmarked, marked, and multiple marked road lanes. It also presents several noise sources, like shadows, light reflections, and low texture variations. Differently from (Vitor et al., 2013), where some formed segments were manually classified as road surface or not, here the ground truth was manually drawn for 210 random frames (non-sequential), respecting the road limits. This set was used to evaluate the road detection capabilities of the 2D road estimation (Subsection 4.3.1) and its final integration in the 2D/3D approach (Subsection 4.3.3), comparing with the previous results from (Vitor et al., 2013), shortly described in the Section 4.2. The experiments were all executed in the same embedded PC of the vehicle APACHE (see Section 3.1 for details).

They were evaluated using the following pixel-based metrics:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{4.9}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{4.10}$$

$$F\text{-measure} = (1 + \beta^2)\frac{\text{Precision Recall}}{\beta^2\text{Precision} + \text{Recall}}, \tag{4.11}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{4.12}$$

where TP and TN are the number of true positives/negatives, FP and FN are the number of false positives/negatives, and $\beta = 1$ defines the harmonic mean (F1-measure) between precision and recall. The results of this evaluation are listed in the Table 4.1. They are illustrated in the Figure 4.13 with the outcomes for fell frames.

Table 4.1 – Evaluation of the road detection algorithms.

Method	Precision	Recall	$F-$measure	Accuracy
Original 2D/3D	0.834	0.972	0.894	0.898
2D estimation	0.856	0.928	0.882	0.874
Final 2D/3D	0.897	0.686	0.774	0.806

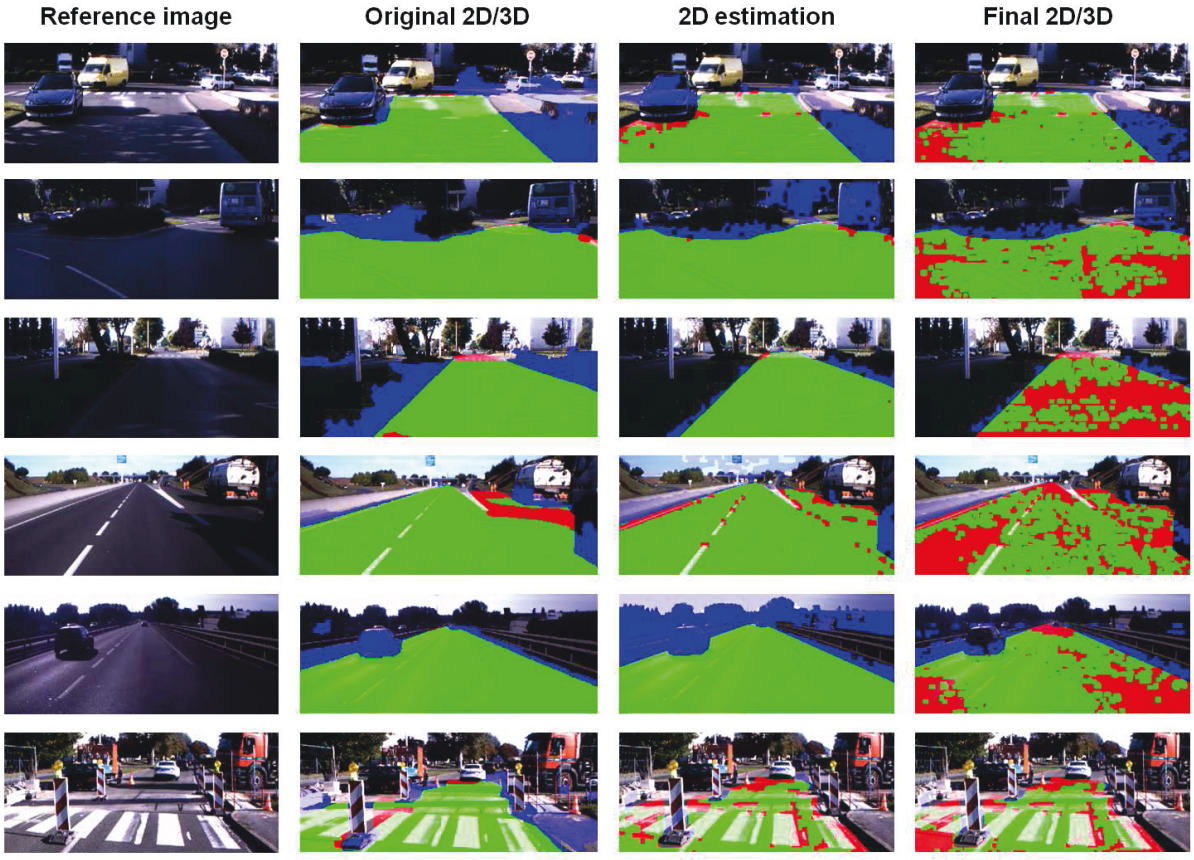


Figure 4.13 – Image results from the evaluation of the road detection algorithms. The colors represents: the true positives (TP) on green, the true negatives (TN) in the original color, the false positives (FP) on blue, and the false negatives (FN) on red.

In the Table 4.1 it is possible to see that the better results were acquired for the original 2D/3D method (Vitor et al., 2013), mainly for the recall and accuracy rates.

However, due to 3D features extraction based only on the V-Disparity map, many obstacles mismatches occurs increasing the number of false positives and, consequently, reducing the precision of the approach. In car navigation situations, this could be dangerous, once obstacles are classified as road surface in some cases, as shown as the blue regions of the Figure 4.13. Moreover, if the cluster has both road and obstacle data and is classified as a road surface, this also increases the false positives. As mentioned before, the variable number of clusters also generates a variable execution time (around 2$\sim$5 Hz) with some locking points, where the frame is visibly frozen.

The false positive problem is also observed using only our 2D road estimation approach during the evaluation. In this case, the detected road surface is constantly mismatched with obstacles which colors are near to the road surface (see Figure 4.13). Thus, validating the 2D estimation with the 3D data in our final approach, several mismatches are removed, like obstacles and the uncertain regions with no disparity value. Note that, this reduces the sensitiveness to road detection (recall rate in Table 4.1), due to many road surfaces with low texture information which result in no disparity values. However, this considerably increases the precision of the method against false positives. In addition, the execution time for this final application was successfully limited to 10 Hz with low CPU usage, leaving it available for others system processes. The most costly task here is the disparity map calculation, that can be implemented in GPU cards (Banz et al., 2010).

Comparing the original 2D/3D road detection from (Vitor et al., 2013) with the final one presented here, the main differences concern to the cluster's approximation and the ANN classification in the first one, and the color based segmentation and the U/V-Disparity maps in the second one. The cluster's features combined with the ANN classification are more robust to uncertain regions with no disparity information, once 2D features are also used during the classification process. On the other hand, the color based segmentation with the U/V-Disparity maps is more sensible to detect obstacles and different color surfaces (like grass and mud). Furthermore, the false negatives can be reduced performing the sensor fusion on the occupancy grid with the 2D estimation data.

4.6.2 Qualitative results

Taking into account the precision of our final 2D/3D road/obstacle detection and the sensibility (recall) of the 2D road estimation, we finally integrate the complete perception system in the experimental vehicle APACHE (Figure 3.1). For a constant control loop time, the perception rate was limited to 10 Hz. The monocular camera is installed on the robot sagittal plane with a rigid structure at (t_x, t_y, t_z) = (1.54, 0.0, 1.62)m and tilt offset $\rho \simeq 9.5°$ performing the 2D road estimation and the lane center features detection. The stereo camera is at (t_x, t_y, t_z) = (1.60, 0.29, 1.60)m with the same tilt offset of the monocular one. The occupancy grid range is up to 30 m, has a cell size of 0.2 m, and uses the CAN velocity information to update its stored data. The final system images during a tour around the SEVILLE test track are composed in the Figure 4.14.

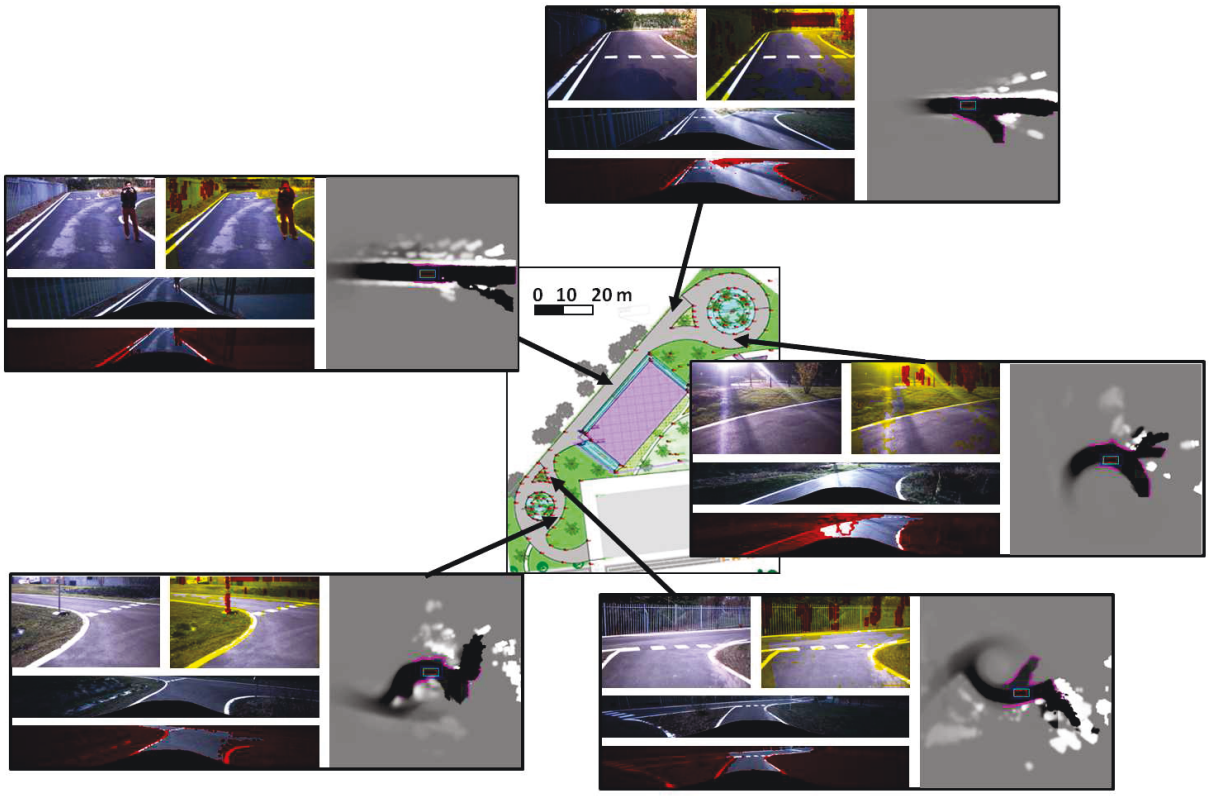


Figure 4.14 – Environment perception results in the SEVILLE test track for different lighting conditions.

These images illustrate the robustness of the entire solution to several noise sources, like the sun light variations and the environment shadows. They also present the importance of the occupancy grid filtering out the detection mismatches, resulting in constant

information in from of the vehicle. Moreover the sensor's fusion in the occupancy grid increase the FOV of the vehicle, mainly in sharp turns, where the roundabout form is easily recovered from these images. This will be an important issue for the global navigation management described in the Chapter 6.

4.7 Conclusions

The environment perception provides the most important information for robots navigation. For autonomous robotic automobiles, this means to provide a robust road and obstacle detection capabilities to allow the vehicle movement with further reactive actions. This chapter proposed some 2D/3D image based approaches to improve the road and obstacle detection problem. The 2D vision solution performed the image segmentation, used to estimate the road surface, and the 3D information (stereo vision data) calculated the drivable and non-drivable areas. The image was segmented using a color based approach and the stereo vision data were processed using the U/V-Disparity technique. The result were combined and classified in road surface and obstacles, with a precision of 89.7% and an accuracy of 80.6% in urban environments. It also presented a fast execution time, allowing regular scheduled applications.

As presented in the Section 4.6, the solution was robust to different environment conditions which normally lead to false detection. This includes several uncertain regions formed by shadows, light variations, and low texture surfaces, where the missing information was merged in the occupancy grid. Comparing the 2D estimation with the 2D/3D classification, the first one was more sensible to detect the road surface (with a higher recall rate), whereas the second was robust against false positives, increasing the precision. Both were useful to provide a dense occupancy grid in front of the vehicle.

Some improvement could be realized in the disparity map calculation, reducing the disparity problems caused by the uncertain regions and increasing the recall rate of the system. Better classifications methods, integrating the benefits of all techniques presented here can also be studied. This final system was validated during several autonomous navigation tasks. They are illustrated in the next chapters.

Chapter 5

Local Navigation Control

Navigation of mobile robots is a large topic with different technologies and applications. In general, robot navigation requirements are divided in three scales (Dixon and Henlich, 1997):

— *Global navigation*: where the robot has the ability to determine its absolute or map-referenced position, and move to a goal destination;

— *Local navigation*: where the robot localization is related to static or moving objects in the environment, and interact with them properly;

— *Personal navigation*: where the position of the body parts of the robot must be taken into account to make up oneself, acting between them or with other objects.

In terms of car-like robots, the global navigation defines an optimal path from a start and goal points in the city map (not directly connected) to guide the car during its movement. The local navigation allows the vehicle interaction with the environment to respect the traffic laws and drive in the right lane, avoiding pedestrians and other cars. However, personal navigation is not applied, once it is not desired any vehicle contact interaction with other elements of the environment.

This chapter will address the autonomous vehicle local navigation in urban environments, more specifically in the road lane following problem. For this purpose, it must take into account the road and lane limits, traffic laws and possible obstacles while performing the navigation task. In the current proposition the vehicle localization is not required, and only the near perceived environment information will be considered during the movement. Hence, the environment perception approach, previously pre-

sented in Chapter 4, will be an important support for the navigation, providing all these required information.

Based on the intelligent vehicle APACHE and its resources (Section 3.1), we propose a new vision-based navigation approach combining Visual Servoing (VS) and obstacle avoidance in a hybrid control. The focus is on the optimality of deliberation for road lane following and the real-time reaction to environment changes. The deliberative control will be performed by an Image-based Visual Servoing (IBVS) approach, adapted from (Cherubini et al., 2008) to follow the road lane, whereas the reactive control is assured by a new IDWA. The IDWA will also be used as a validation for the visual servoing velocities. Note that, we have chosen to work in the image space (IBVS) and in the 3D space (DWA) simultaneously. By using an IBVS approach instead of a PBVS avoids the knowledge of the path geometric model. Moreover, this also reduce errors associated with the calibration of the camera, as described in (Chaumette, 1998). It is important to mention that the VS methodology chosen is just one between many others which could be adapted to the present solution. This hybrid controller will be presented as follows.

5.1 Control design

The hybrid controller, combining VS and IDWA, aims to associate the best elements of both deliberative and reactive controls. The idea is using the benefits of the IBVS for path following, to keep the vehicle in the road lane center, and the IDWA to give priority to fast obstacle avoidance maneuvers and smooth convergence to the lane center. This section describes both controllers, compounding the hybrid controller called by VS+IDWA.

5.1.1 Statement of the problem

The problem of road lane following could be performed by several different visual approaches, as described in the Section 2.2. Here the road path was approximated to a line path on the floor and projected in the image frame $\{\mathcal{I}\}$ to define the features set $s = [X \ Y \ \Theta]^T$, presented in the Subsection 4.4.1. The objective is to guide these

features to the desired configuration $X = X^*$, $Y = Y^*$, and $\Theta = \Theta^*$, while avoiding obstacles. This means that the car must respect the road boundaries and speed, as well as cars, pedestrians, etc., on the way. Once the camera is installed on the car sagittal plane ($t_y = 0$), the desired configuration is the line vertically oriented ($90°$) on the center of the image.

5.1.2 Deliberative control: Visual Servoing (VS)

This subsection recall the formulation used by (Cherubini et al., 2008) for the Image-Based Visual Servoing (IBVS) approach of follow a line path projected on the image frame. Considering the image frame depicted in the Figure 4.10a, the features set $s = [X\ Y\ \Theta]^T$ defined by the tangent Γ, and the camera installed on the sagittal plane of the car, the goal is to compute a control input to drive these features to the final configuration $X^* = \Theta^* = 0$ and $Y^* = Y_I$, which means the vehicle in the center of the lane. This is achieved considering a constant linear velocity $v = v_d > 0$, in accordance with the speed limits of the road, and applying a nonlinear feedback control law in the angular velocity of the car ω, as defined below.

Diffeomorphism between image and cartesian velocities: the interaction matrix

The formulation is based on a row/column controller related to the error in X/Y, as seen in the Figure 5.1. Each controller must relate the image features velocities $\dot{s} = [\dot{X}\ \dot{Y}\ \dot{\Theta}]^T$ to the robot velocities $u_r = [v\ \omega]^T$. Remembering that these velocities are associated to the car control inputs by the equations (3.3) and (3.4). Initially, the image features velocities must be written in terms of the camera frame velocities $u_c = [v_{c,x}\ v_{c,y}\ v_{c,z}\ \omega_{c,x}\ \omega_{c,y}\ \omega_{c,z}]^T$, by

$$\dot{s} = L_s u_c. \tag{5.1}$$

This expression uses the interaction matrix L_s for the normalized perspective camera model of equation (4.4), derived from (Espiau et al., 1992) as:

$$L_s(X,Y,\Theta) = \begin{bmatrix} -\frac{1}{z_c} & 0 & \frac{X}{z_c} & XY & -(1+X^2) & Y \\ 0 & -\frac{1}{z_c} & \frac{Y}{z_c} & 1+Y^2 & -XY & -X \\ \frac{C\rho C^2\Theta}{t_z} & \frac{C\rho C\Theta S\Theta}{t_z} & -\frac{\zeta C\rho C\Theta}{t_z} & -\zeta C\Theta & -\zeta S\Theta & -1 \end{bmatrix}, \tag{5.2}$$

with $C\Theta = \cos\Theta$, $S\Theta = \sin\Theta$, $C\rho = \cos\rho$, and $\zeta = (Y\sin\Theta + X\cos\Theta)$. Note that each line of L_s is related to its respective image feature (L_X, L_Y and L_Θ). This matrix also requires the distance estimation between the camera and the projected image point D in the world frame, given by the expression (4.5).

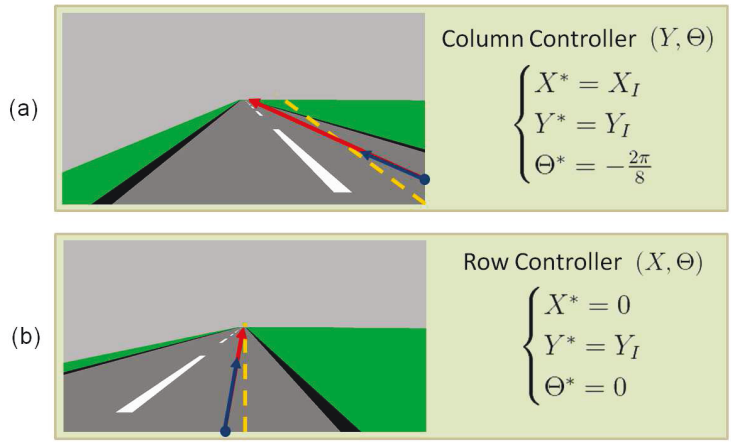


Figure 5.1 – Feature configuration to apply the column (a) and row (b) controllers and their corresponding setpoints in dashed yellow.

Thus, u_r may be expressed in the camera frame $\{\mathcal{C}\}$ by (5.3) using the homogeneous transformation (5.4):

$$u_c = {}^{C}T_R u_r, \tag{5.3}$$

$${}^{C}T_R = \begin{bmatrix} 0 & -t_x\cos\tau \\ -\sin\rho & t_y\cos\rho \\ \cos\rho & -t_y\sin\rho \\ 0 & 0 \\ 0 & -\cos\tau\cos\rho \\ 0 & -\cos\tau\sin\rho \end{bmatrix} \tag{5.4}$$

The columns of the transformation ${}^{C}T_{R}$, named as T_v and T_ω, are related to the robot velocities. Once the camera is on the sagittal plane of the car, $\tau = 0$.

Robot control calculation

The row controller must drive (X,Θ) to the desired state (X^*,Θ^*), regulating the error $e = [X - X^* \ \Theta - \Theta^*]^T$ to zero under the constraint $Y = const = Y^*$. This led us to the system state equations, obtained from combining the equations (5.1), (5.3), and (5.4):

$$[\dot{X} \ \dot{\Theta}]^T = A_r v + B_r \omega, \tag{5.5}$$

with:

$$A_r = \begin{bmatrix} L_X \\ L_\Theta \end{bmatrix} T_v \text{ and } B_r = \begin{bmatrix} L_X \\ L_\Theta \end{bmatrix} T_\omega. \tag{5.6}$$

When $B_r \neq 0$, the control law is:

$$\omega = -B_r^+(\lambda e + A_r v), \tag{5.7}$$

where B_r^+ is the Moore-Penrose matrix pseudoinverse of B_r, and $\lambda = [\lambda_X \ \lambda_\Theta]^T$ are positive gains.

Similarly to the row controller, the column controller must drive (Y,Θ) to the desired state (Y^*,Θ^*), regulating the error $e = [Y - Y^* \ \Theta - \Theta^*]^T$ to zero under the constraint $X = const = X^*$. It is analogously defined changing the row controller definitions from X to Y. The controllers are selected regarding the features' coordinates in the image plane (see Figure 5.1). If $Y \neq Y_I$, the column controller is applied for $X^* = X_I$, otherwise is the row controller the selected one.

This controller proposed by (Cherubini et al., 2008) presents some convergence problems related to the IBVS, mainly for large initial offsets between the robot and the line on the floor (here represented by the road lane center). However, during the line following, the robot presents good results, better than a PBVS approach (Cherubini et al., 2011). For more details about the implementation and stability analysis see (Cherubini et al., 2008, 2011). Some studies could be performed in order to change the control law and improve the controller convergence. Instead of that, here we pro-

pose the VS validation before applying its velocities in the vehicle, avoiding large movements when far away from the final desired setpoint. This will be performed by the reactive controller presented next.

5.1.3 Reactive control: Image-based Dynamic Window Approach (IDWA)

The Dynamic Window Approach is a reactive obstacle avoidance technique proposed originally by (Fox et al., 1997). It selects, between all available velocities in a Dynamic Window search space (V_{DW}), the best control input by optimizing the objective function (5.8):

$$DWA(v,\omega) = \alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) + \gamma \cdot velocity(v). \tag{5.8}$$

This takes into account the weighted sum (adjusted by the gains α, β, and γ) of three functions:

— *heading*: based on the final orientation of the robot regarding the goal position in the world;

— *dist*: which prioritize movements over the areas free of obstacles (with the highest distance to collision); and

— *velocity*: with focus on the desired linear velocity setpoint.

Due to the nature of this optimization function, the DWA was adapted to several major goals (Brock and Khatib, 1999; Ogren and Leonard, 2005; Saranrittichai et al., 2013) and different robot types, like car-like robots (Rebai et al., 2007; Rebai and Azouaoui, 2009), as well as to dynamic environments (Seder and Petrovic, 2007). However, the robot and goal relative position to the world were known in these works, which are susceptible to GPS localization problems.

To avoid these problems, we proposed a new approach for the DWA, by considering 2D image features to guide the robot and 3D obstacles information to avoid them. This reactive controller was named as Image-Based Dynamic Window Approach (IDWA). Note that this technique combines 2D and 3D data in the visual servoing context, thus the IDWA can also be classified as a hybrid visual servoing. The originality

of our method IDWA w.r.t DWA concerns to the function *heading* of the objective function (5.8), as explained bellow.

The IDWA functions

The $heading(v, \omega)$ function is responsible to guide the vehicle to a desired goal configuration. In the original DWA formulations (Fox et al., 1997), $heading$ returns high weights for those control inputs which lead the vehicle orientation nearest to goal position, based on the robot localization. However, in the present VS application (Subsection 5.1.2) the robot localization is not required, and the goal is to guide the error of the image features to zero. Extending to the DWA, the $heading$ function must estimate the error

$$e_{t+\triangle t} = \begin{bmatrix} X_{t+\triangle t} - X^* \\ Y_{t+\triangle t} - Y^* \\ \Theta_{t+\triangle t} - \Theta^* \end{bmatrix}$$

in the next frame $\mathcal{I}_{t+\triangle t}$, considering ($X^*$,$Y^*$,$\Theta^*$) as setpoint. This is illustrated in the Figure 5.2. Thus, high weights are given to the sets of control inputs $(v_i, \omega_j) \in V_{DW}$ which reduce the final error $e_{t+\triangle t}$.

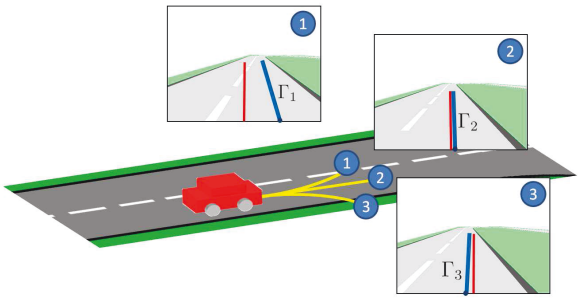


Figure 5.2 – Estimation of the image features set Γ_i (blue line) in the frame $\mathcal{I}_{t+\triangle t}$ applying the control inputs (v_1, ω_1), (v_2, ω_2) and (v_3, ω_3). The reference position is also represented in red, which means the vehicle in the center of the road lane.

Recalling the equations (5.1) and (5.3), it is possible to estimate the features velocities $\dot{s}$ in the current image frame $\mathcal{I}_t$ for all robot control input $u_r \in V_{DW}$. Thus, integrating each set of velocities over the time, we estimate the features configuration in the frame $\mathcal{I}_{t+\triangle t}$. As the row/column controllers are applied independently, the func-

tion $heading(v, \omega)$ was divided in: $XY_{error}(v, \omega)$, responsible for the row/column error (X or Y); and $\Theta_{error}(v, \omega)$ with the Θ error. Their final values were calculated by:

$$XY_{error}(v, \omega) = \begin{cases} 1 - \frac{|e_X(t+\triangle t)|}{e_{Xmax}}, & \text{if row controller,} \\ \\ 1 - \frac{|e_Y(t+\triangle t)|}{e_{Ymax}}, & \text{otherwise.} \end{cases} \tag{5.9}$$

$$\Theta_{error}(v, \omega) = 1 - \frac{|e_\Theta(t + \triangle t)|}{\pi}. \tag{5.10}$$

The errors in the image frame $\mathcal{I}_{t+\triangle t}$ are e_X, e_Y, and e_Θ, and the maximum measurable errors in X and Y are e_{Xmax} and e_{Ymax}. Then, we evaluate $heading(v, \omega)$ as the sum of these previous functions:

$$heading(v, \omega) = \alpha_1 XY_{error}(v, \omega) + \alpha_2 \Theta_{error}(v, \omega). \tag{5.11}$$

where α_1 and α_2 are the weighting parameters.

Next, the *dist* function is calculated by:

$$dist(v, \omega) = \frac{d_{coll}}{d_{max}},$$

where d_{coll} is the distance to collision given by (Arras et al., 2002) for polygonal robots moving in circular trajectories. This approach is better described in the Appendix A. The maximal perceived distance d_{max} is a sensor limitation, here equivalent to the limits of the occupancy grid (Section 4.5). To allow speed variations while moving along narrow roads or performing the obstacle avoidance, the limits of the robot were linearly expanded regarding its speed during the *dist* evaluation (see the light and dark blue rectangles at the Figure 4.12). This is the same consideration from (Fox et al., 1997; von Hundelshausen et al., 2008), creating the robot side clearance related to the speed, which is robust to errors in the circular trajectory approximation.

The last function $velocity(v)$ is defined based on the desired robot linear velocity v_d (from the VS approach in the Subsection 5.1.2). It is constant and respects the road

speed limit. Thus, the objective is to return high values to velocities near to v_d to prioritize these outputs in the IDWA. It was defined as:

$$velocity(v) = \begin{cases} \dfrac{v}{(v_d - v_{min})} & \text{if } v \leq v_d, \\ \\ \dfrac{(v_{max} - v)}{(v_{max} - v_d)} & \text{if } v > v_d, \end{cases} \tag{5.12}$$

where v_{min} and v_{max} are the robot minimal and maximal reachable velocities.

The Dynamic Window search space V_{DW}

Initially, for the current robot velocity (v_a, ω_a), the Dynamic Window V_d is defined for all reachable velocities in a time interval $\triangle t$ as:

$$\begin{aligned} V_d = \{(v,\omega)|\ v \in [v_a - \dot{v}\triangle t, v_a + \dot{v}\triangle t]\,, \\ \omega \in [\omega_a - \dot{\omega}\triangle t, \omega_a + \dot{\omega}\triangle t]\}\,, \end{aligned} \tag{5.13}$$

with the robot input set $u_r = [v\ \omega]^T$ (see Section 3.3) and the robot accelerations $(\dot{v}, \dot{\omega})$. Thus, each reachable velocity in V_d must be classified as admissible or not due to the distance to collision d_{coll} (see Appendix A), calculated by the function $dist(v,\omega)$, and the robot maximum decelerations ($\dot{v}_b$, $\dot{\omega}_b$). By the equation (5.14), based on the Torricelli's equation, if d_{coll} is bigger than the distance required to stop the vehicle safely, then the velocity is admissible.

$$\begin{aligned} V_a = \{(v,\omega)|\ v \leq \sqrt{2 \cdot dist(v,\omega) \cdot d_{max} \cdot \dot{v}_b}, \\ \omega \leq \sqrt{2 \cdot dist(v,\omega) \cdot d_{max} \cdot \dot{\omega}_b}\}\,. \end{aligned} \tag{5.14}$$

Finally, the Dynamic Window search space is computed considering the current speed of the vehicle, the maximum accelerations and decelerations, the physical limits, and the obstacles in the environment by the equation (5.15).

$$V_{DW} = V_d \cap V_a \cap V_s\,, \tag{5.15}$$

where V_s is the set of points that satisfy the maximum velocity constraints v_{max} and ω_{max}. By discretizing the search space V_{DW}, a velocity is selected following the criteria presented by the objective function (5.8). The discretization considers the smallest $\Delta\phi$ and Δv which significantly contribute in the vehicle navigation. In addition, the stabilization time required for the vehicle actuators were observed when defining the final $\triangle t$ (see Section 3.5). An example of V_{DW} is shown in the Figure 5.3, where w_a was converted in ϕ_a, by means of the equation (3.4), for visualization purposes.

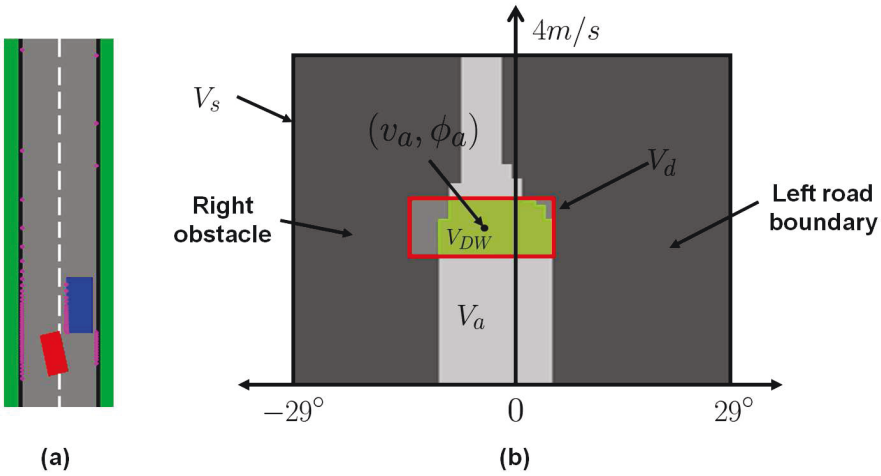


Figure 5.3 – Example of a Dynamic Window V_{DW} (b) calculated for a certain vehicle state in (a). In (a), the vehicle is illustrated by the red rectangle, where the obstacle (blue) and road boundaries are detected by a laser (pink dots) projected in an occupancy grid. In (b), v_a and ϕ_a are the current linear velocity and steering angle of the vehicle, resulting the V_{DW} in green.

5.1.4 Hybrid control: VS+IDWA

The VS methodology presented in the Subsection 5.1.2 does not guarantee safeness to the car movement, once its main objective is only the road lane following. To achieve that, it neglects the obstacles information and the vehicle dimensions, allowing movements out of the road surface or directly throw an obstacle. The IDWA, otherwise, has opposite tasks defined by the functions *heading* and *dist*, which can compromise the optimal road following. Due to this, we combine both controllers in the hybrid solution VS+IDWA proposed in this work. In this controller, the VS output is validated before being applied in the vehicle.

The equation (5.15) defines the IDWA search space V_{DW}, with all reachable velocities available in a time interval $\triangle t$. Basically, if the VS velocities (v_{VS}, ω_{VS}) are not in the current V_{DW}, a new control input is calculated by the objective function defined in (5.8). In order to increase the robot reaction against obstacles, a maximal distance to collision (d_{VS}), for the VS velocities, was added to this evaluation. Thus, the VS velocities will be considered valid if:

$$(v_{VS}, \omega_{VS}) \in V_{DW} \text{ and } dist(v_{VS}, \omega_{VS}) > \frac{d_{VS}}{d_{max}}. \tag{5.16}$$

5.2 Convergence Analysis

To analyze the convergence of the current approach, it is necessary to define the expected objectives of the local navigation approach. Once we are not using global motion planning to assure global convergence or local minima-free movements, three local objectives were set:

1. Stabilize in the road lane center and follow it when there are no obstacles;
2. Change to the next free lateral lane if available;
3. Stop when there are no options available.

These are the same main tasks that a human driver must deal when navigating in a road to get to the next intersection. For all tasks it is considered that the environment perception is able to define the right free-lane setpoint to follow.

The first main task (1) is guaranteed by the deliberative control (VS), once its stabilization is proved by (Cherubini et al., 2008). The next task (2) is performed by the hybrid controller VS+IDWA, when there are obstacles in the current lane and the robot must avoid all obstacles and limits of the road to converge to the next desired setpoint. Similarly to (Brock and Khatib, 1999), where a global planning was used with the DWA to guide the robot with no local minimas, the IDWA uses the VS equations to move toward to the next image setpoint. Note that, for the IDWA, following the road lane and avoiding an obstacle in the same lane are opposite tasks. Thus, if we want a reactive obstacle avoidance, we must reduce the gain for road lane following ($heading$ function) and consequently affect the robot stabilization in the lane center. Here is the

main contribution of associating the deliberative controller VS with the IDWA, because when the robot is closer to the next desired setpoint, the VS starts to actuate and the condition (1) guarantees the robot stabilization in the road lane center. Finally, the task (3) is performed when there are no options available, once the IDWA moves the vehicle just before collide and all control inputs are not allowed. In the next section, these situations will be illustrated with some experiments in a simulation environment and in a real autonomous car.

5.3 Experimental Results

The proposed controller was implemented in the simulation environment and in the real car APACHE presented in the Chapter 3. The experiments setup and results are shown next.

5.3.1 Experiments setup

To validate the navigation methodology proposed, the simulation environment was adjusted with the same parameters from the vehicle APACHE (Figure 3.1). This means the use of the kinematic model from equation (3.2) to represent the vehicle movement, the same kinematics constraints, and some actuators dynamics (as described in the Section 3.5. The perception of the environment was implemented the solution proposed in the Chapter 4, where the monocular camera is installed on the robot sagittal plane with a rigid structure at $(t_x, t_y, t_z) = (1.54, 0.0, 1.62)$ m and tilt offset $\rho \simeq 9.5^\circ$ to detect the road lane center features, as described in the Subsection 4.4.1. It also detects obstacles with a laser sensor with 180° of coverage (for the simulation experiments) or a stereo vision camera (for real experiments). The laser is located in front of the vehicle at $(t_x, t_y) = (3.43, 0.0)$ m and the stereo camera is at $(t_x, t_y, t_z) = (1.60, 0.29, 1.60)$ m with the same tilt offset of the monocular one. To extend, filter, and fuse the perception data, all the detected elements are represented in an occupancy grid locally constructed around the robot (Elfes, 1989), as detailed in the Section 4.5. The grid range is up to d_{max} = 30 m and the maximal distance to collision for the VS velocities was defined to d_{VS} = 20 m, which gives enough space for reactive maneuvers in low speed.

5.3.2 Simulation experiments

Simulation is an important resource that aggregates several experiments possibilities, which could be hard to execute in real environments. It also allows an easy visualization of the influence of the VS control parameters and the IDWA validation during the robot navigation. The robot speed was limited to 3 m/s due to the kinematic model approximations. The reference was always set to the center of the right lane, similar to the Figure 4.10a. With this setup, we checked the influence of the VS and IDWA gains in the final VS+IDWA result, then we performed a quantitative comparison between IDWA and the VS+IDWA, and finally we verified the VS control output validation in the IDWA. For a better visual explanation, the vehicle pose (red rectangles) were left in all figures to illustrate its movement during these experiments. These rectangles also give a speed notion during the vehicle movement, where more spaced they are, higher is the final speed. It is important to note that in our approach there is no a priori path planning.

Influence of the VS gain

The Figure 5.4 was created to understand the influence of the gain on the VS approach. It shows the simulation results for some λ values to track the road lane center, without considering any obstacle on the way. In this figure it is possible to see the errors decreasing when increasing λ, which is evidenced by the Table 5.1 with the mean square error (MSE) of the image features. However, high λ values also mean large control output variations, mainly when the features present significant errors, as illustrated in the same figure. Increasing the λ, the steering velocity v_2 reached several times the maximum value ($|v_{2max}|$) and frequently changed the signal, resulting in uncomfortable behavior for a human driver.

Table 5.1 – VS gain λ evaluation.

λ	MSE_X	MSE_Θ
0.3	0.512	0.302
0.4	0.175	0.197
0.5	0.117	0.156
0.6	0.096	0.146
0.7	0.090	0.152

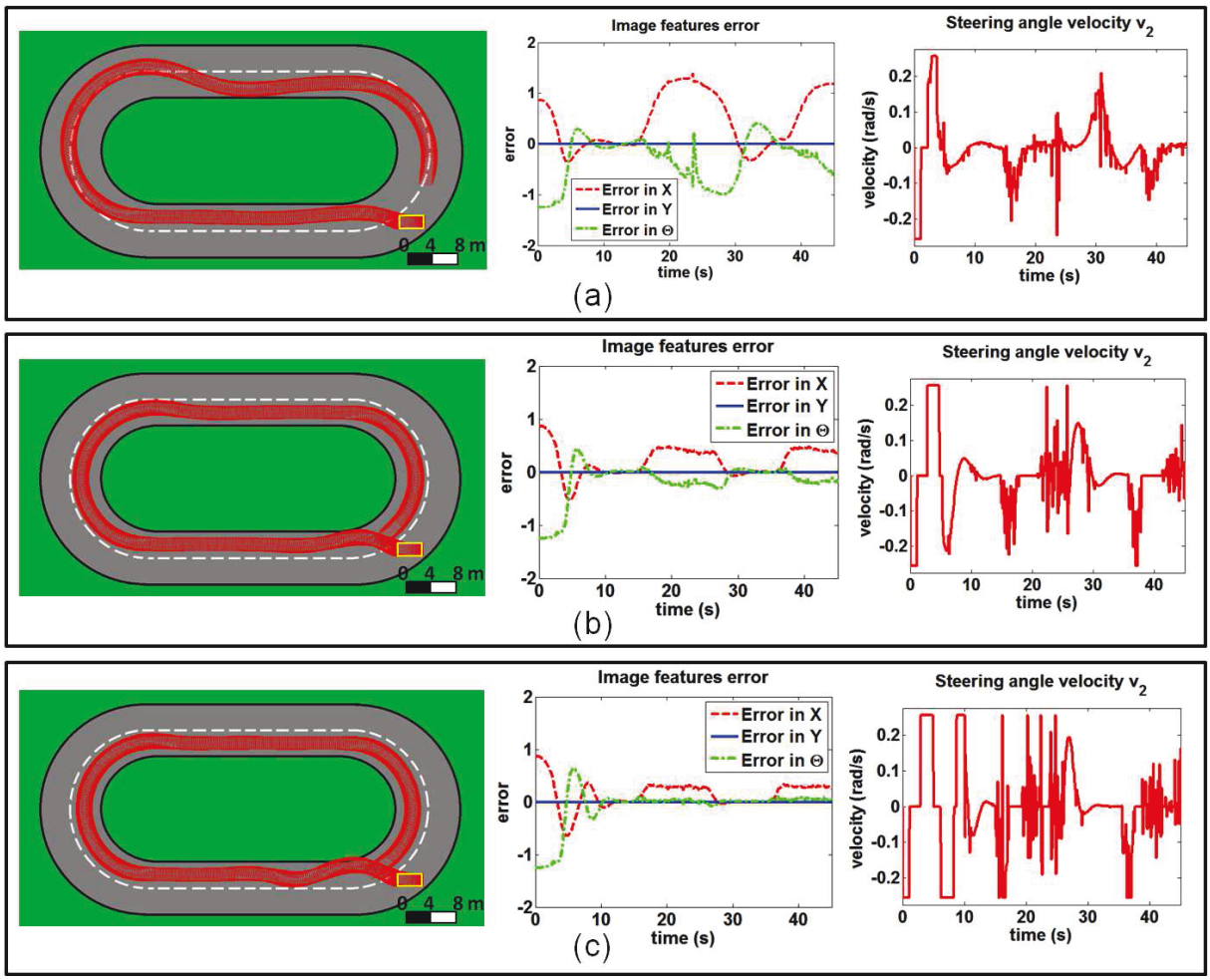


Figure 5.4 – Simulation result for the VS controller performing the road lane following with $\lambda = 0.3$ (a), $\lambda = 0.5$ (b), and $\lambda = 0.7$ (c). The left column illustrates the car movement in this environment, where in yellow is the initial pose and in red the instantaneous positions for a clockwise movement. In the column of the middle is error evolution of the image features and in the column of the right is the steering wheel velocity output (v_2) during the experiments.

Moreover, this figure shows that the VS approach is not robust for path reaching, once it results in large overshoots on the final movement. Differently for path following, where the lateral error is reduced and the robot can track the road lane with better precision. The convergence problem is mainly caused by the large v_2 commands, which do not consider the road boundaries and obstacles, neither the linear velocity changing to reduce the overshoot.

There are many possibilities to adjust the gain for this VS approach. The one used by (Cherubini et al., 2008), e.g., considered a variable exponential gain related to the error of the image features, reducing the gain when the higher is the error. Although

applying this gain resulted in small overshoots while converging to the center of the lane, it presents some problems when following the lane center in the curves. This is a similar result to the one illustrated in the Figure 5.4a. They also showed that there is a relation between the gain and the maximum curvature of the path, which is limited by the robot kinematic constraints (see the ICC in the Figure 3.7).

Instead of dynamic adjusting the gain, it is possible to reduce the overshoot by validating the VS control outputs in the IDWA. For that, we considered the final $\lambda = 0.5$, to take advantage of the path following capabilities of the VS controller (Figure 5.4b). It is important to mention that drifting away the tangent estimation point D (see the Figure 4.10) can also be used to anticipate the vehicle reaction to different path curvatures and reduce the problem observed in the Figure 5.4a.

Influence of the IDWA gains

Before applying the complete VS+IDWA solution, the gains α, β, and γ associated with the IDWA must be tuned. However, there are no metrics to evaluate these gains during a reactive obstacle avoidance maneuver, once it is a user choice to define what is the best reaction to the robot (e.g. overtaking or just stopping near to the obstacle). The IDWA functions control the final robot reaction, and then by enabling these gains one by one it is possible to understand their influence and finally adjust them.

The vehicle movement of the Figure 5.5a is acquired from adding some obstacles to the simulation environment and using only the $heading$ function, composed by XY_{error} and Θ_{error}. This function prioritize only the road lane following until stop near to an obstacle or in a situation where the error cannot be reduced, as shown in this figure. The function $dist$ (Figure 5.5b), otherwise, results in a movement over the regions free of obstacles but without any goal or speed definition. This leads the vehicle to stop in a region equidistant from other obstacles or moves with low velocities. The effect associated to the function $velocity$ is to emphasize the linear velocities commands near to v_d. This results in a movement closer to the obstacles, with abrupt maneuvers to avoid them, as shown in the Figure 5.5c. All functions together with no adjustment result in the Figure 5.5d. With this configuration, the robot gives priority to reduce the error in the image features until stop near to an obstacle.

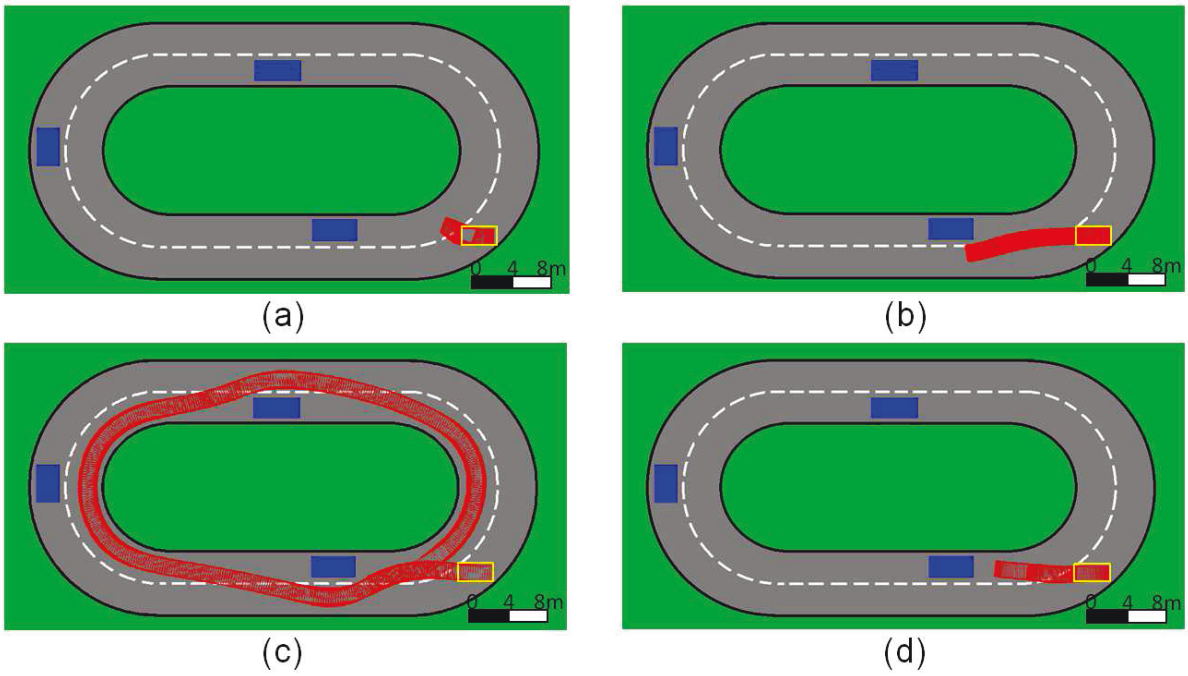


Figure 5.5 – Influence of the gains associated with the IDWA when performing a reactive obstacle avoidance. Initially, only the functions XY_{error} and Θ_{error} were enabled in (a), $dist$ in (b), and $velocity$ (c). Finally, all the gains were enabled in (d), with $\alpha_1 = \alpha_2 = \beta = \gamma = 1.0$. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

By comparing the movements presented in the Figure 5.5, we can see that the $heading$ function showed an opposite result to those from $dist$ and $velocity$. This is caused by the opposition between follow the road lane and avoid the obstacles on the same road lane. However, when these functions were combined with similar gains, the vehicle was able to perform the road lane following, with the linear velocity near to v_d, and avoiding the collision with the obstacle situated on the lane. Once the main objective for the IDWA is the smooth obstacle overtaking with high velocities, in the Figure 5.6 the previous gains were adjusted one by one regarding this final objective.

Initially, we reduced the XY_{error} and Θ_{error} gains $\alpha_1 = \alpha_2 = 0.1$ (Figure 5.6a) to avoid the native opposition between $heading$ and the other functions of the IDWA (Section 5.2). Although avoiding obstacles, this configuration reduced considerably the car linear velocity (v_1) during the maneuvers. In this context, we increased the $velocity$ gain $\gamma = 3.0$ in the Figure 5.6b. Thus, the distance to the obstacles was enhanced by the gain $\beta = 2.0$, resulting in the Figure 5.6c. The final configuration was set to

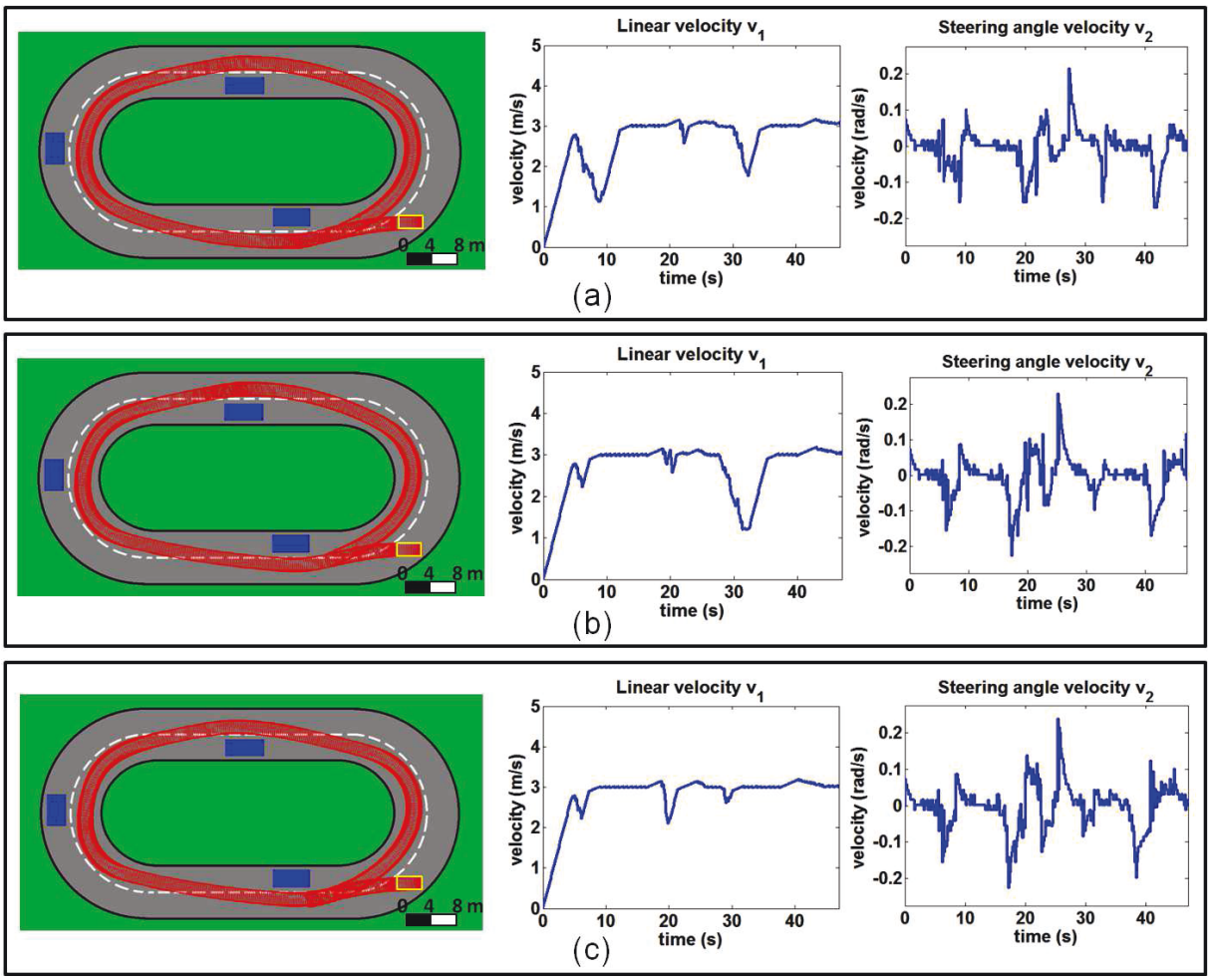


Figure 5.6 – Tuning process for the gains associated with the IDWA, where: (a) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 1.0$, and $\gamma = 1.0$; (b) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 1.0$, and $\gamma = 3.0$; and (c) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

$\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$. Note that the steering velocity v_2 varies smoothly in all configurations presented in this figure.

VS+IDWA versus IDWA

With the previous selected gains ($\lambda = 0.5$, $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$), an important difference between the results from VS+IDWA and IDWA may be observed, as shown in the Figure 5.7 and explained as follows. When performing the reach and following, both controllers are able to converge to the right lane center in the same time (0 to 15 seconds of simulation). However, following the right lane center (15 to 30 seconds of simulation), the deliberative control part of the VS+IDWA results in a

small error in the image features than the IDWA controller. This is only possible because the VS controller has no opposition following the road lane center, different from the IDWA which always regards the distance to collision and the robot velocity to calculate the vehicle control input. This is proved calculating the MSE for both simulations (see Figure 5.7). This difference can also be observed in several points of the Figure 5.8, mainly on the dashed boxes I and II. In all cases, the setpoint was defined as the road lane center on the right side, when there are visible marks, or the road center, otherwise.

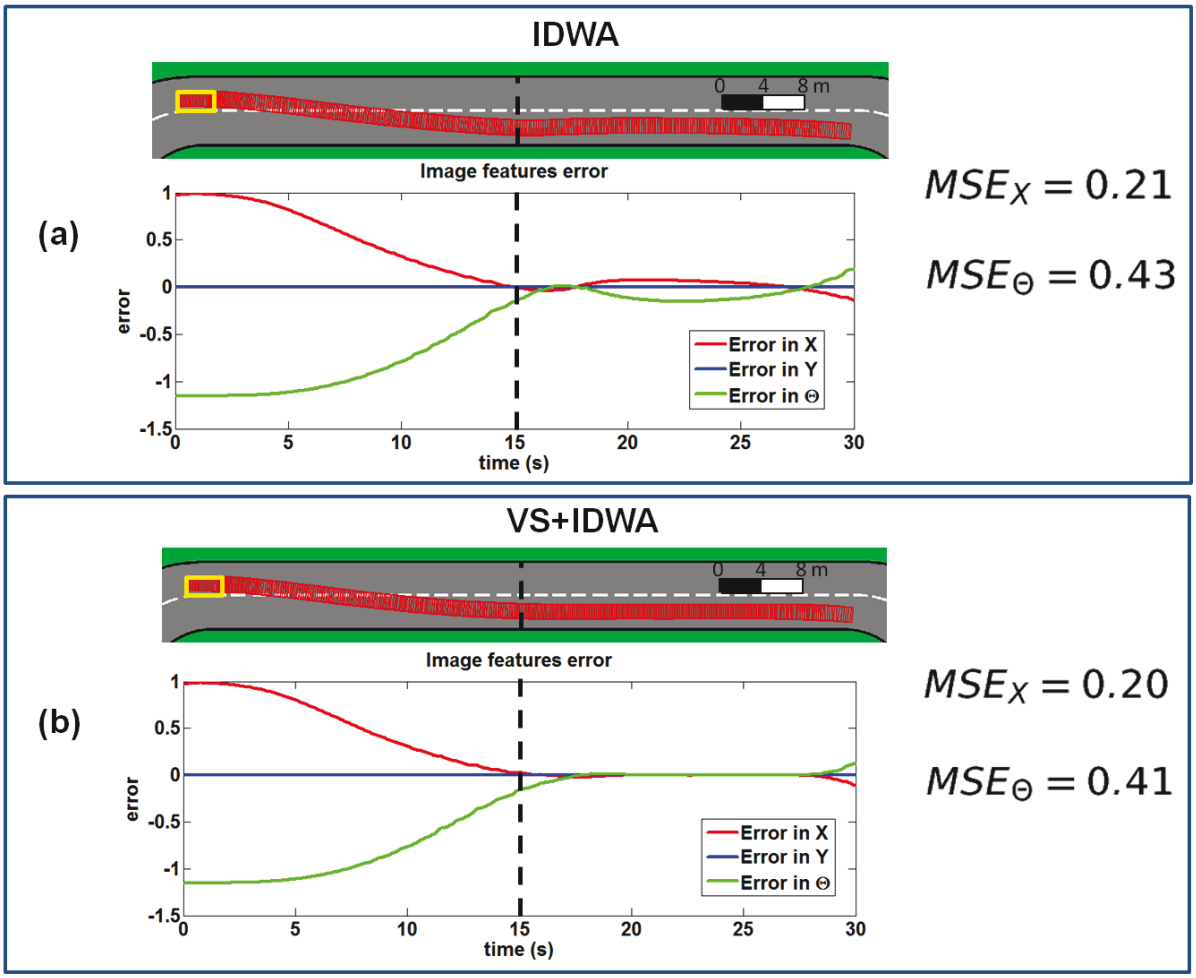


Figure 5.7 – Comparison between the vehicle lane convergence and following using the IDWA (a) and the VS+IDWA (b) controllers, with the error of the image features and the MSE. The car initial pose is represented in yellow and the car instantaneous positions are in red for a left to right movement.

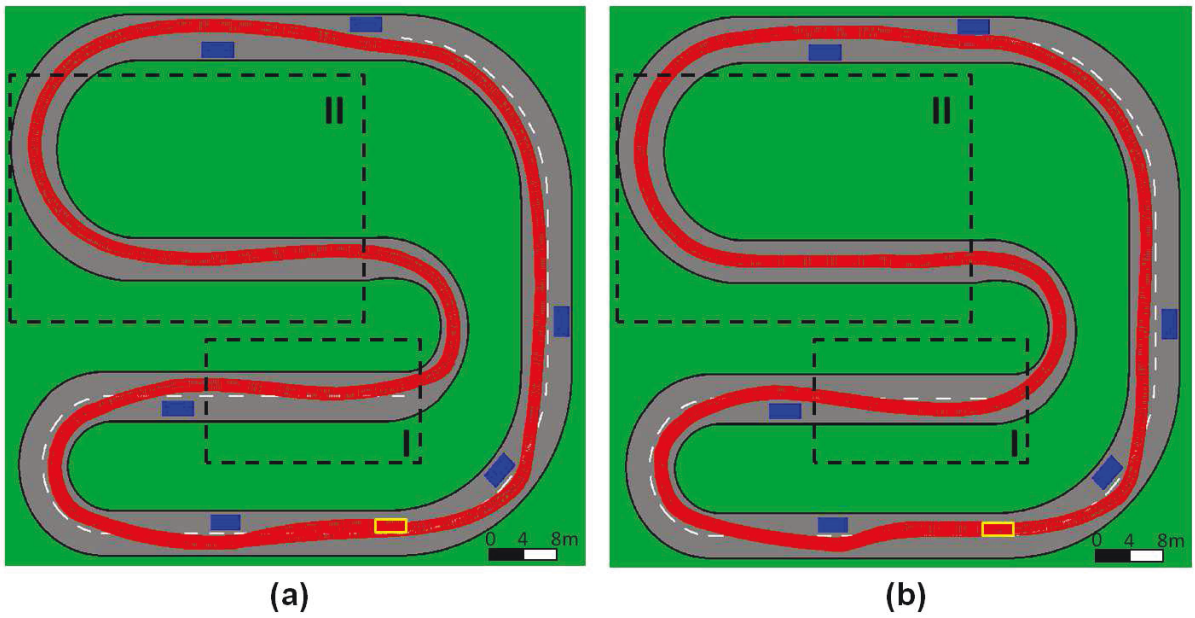


Figure 5.8 – Comparative movement using only the reactive controller IDWA (a) and the complete solution VS+IDWA (b). The difference is mainly observed for the road lane following task in the dashed box I and II. The car initial pose is represented in yellow, the obstacles are in blue, and the car instantaneous positions are in red for a clockwise movement.

The VS validation

By focusing on the VS validation in the IDWA, the experiment illustrated in the Figure 5.9 was performed for the car starting in the center of the road lane. It presents 30 seconds of simulation during the road lane following and obstacle avoidance maneuver. This simulation used the previous gains setup of $\lambda = 0.5$, $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$. For better visualizing and understanding the car movement, the Figure 5.9a-c is vertically in the same time scale and the outputs displayed are the linear velocity v_1 and the steering angle ϕ.

When the condition expressed by (5.16) is valid, the VS output is allowed as the robot input and its values are confirmed by the IDWA. This is represented in the first 10 seconds of simulation of the Figure 5.9. After that, the obstacle prevents the movement in the road lane and the IDWA starts to modify the VS control output to guarantee the obstacle avoidance. At the end of the road, the obstacles completely block the roadway and, as expected, the IDWA makes the vehicle stop safely before the collision. During the whole procedure, the steering angle performed only smooth variations, without

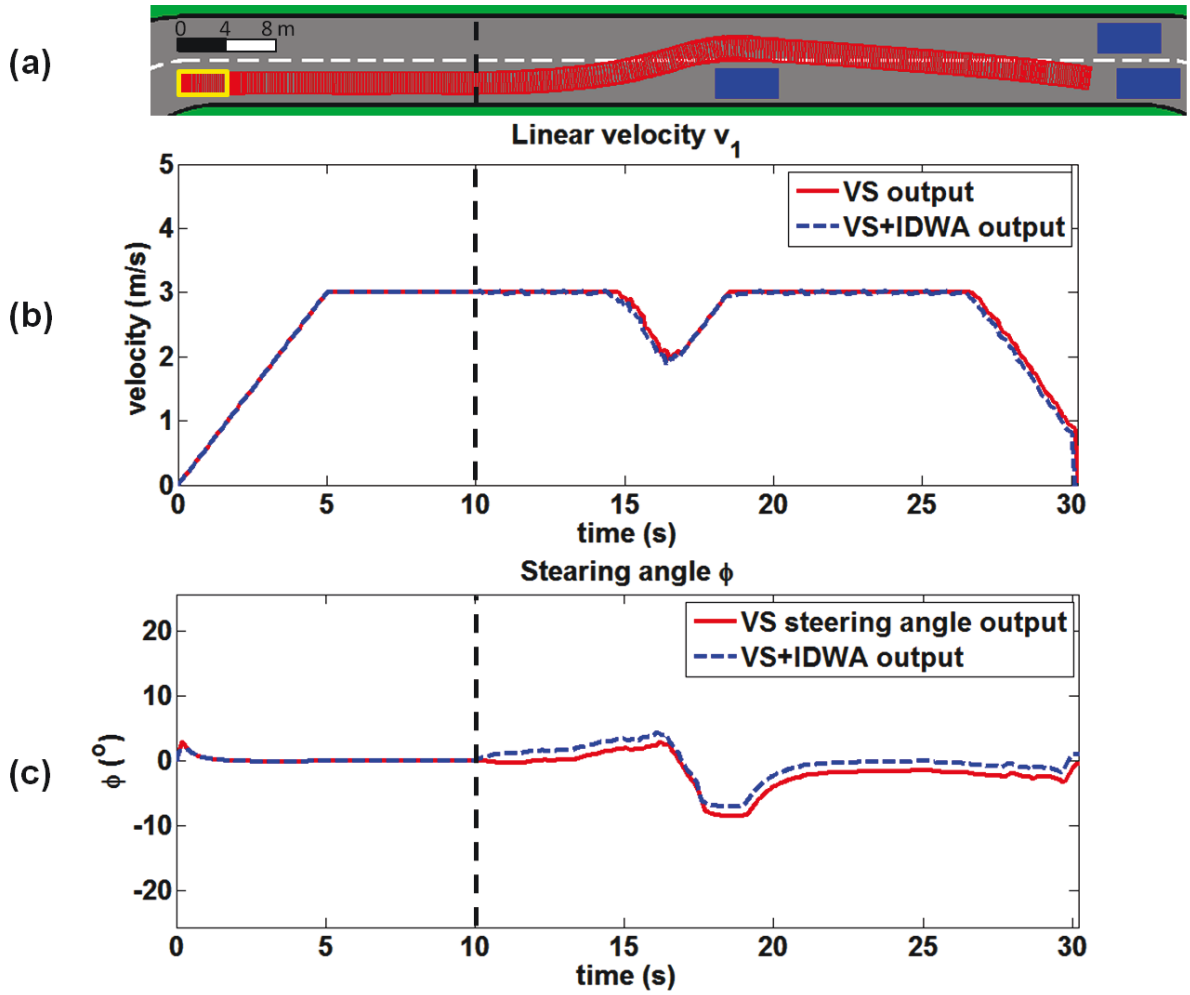


Figure 5.9 – VS and the VS+IDWA outputs for the simulation result represented in (a), with the car's linear velocity v_1 in (b) and steering angle ϕ in (c). The car initial pose is represented in yellow, the obstacles are in blue, and the car instantaneous positions are in red for a left to right movement.

reach values higher than $\pm 10°$, as seen in the Figure 5.9c. For passengers on the cockpit, this is an important indicative for the comfort on the car.

5.3.3 Real car-like robot experiments

Three experiments (I-III) were performed with the real car-like robot APACHE (Figure 3.1) in order to validate the VS+IDWA control approach. They were performed with the same setup of the previous simulation results ($\lambda = 0.5$, $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$), at the SEVILLE test track illustrated in the Figure 5.10. In all cases, the vehicle must to move in the road center respecting the road boundaries and the desired linear velocity v_d (limited for security reasons). The visual servoing task is defined by a yellow

line and the reference by a dotted red line in the follow images. These experiments where performed as explained below.

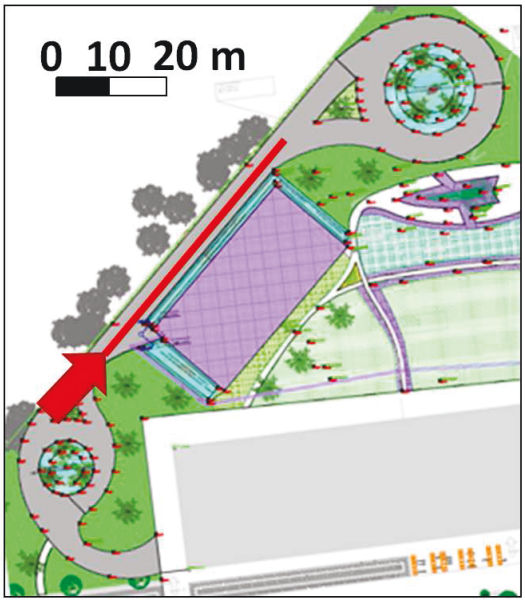


Figure 5.10 – Robot course (red line) for the local navigation experiments at the SEVILLE test track. All experiments start in the arrow.

Experiment I - Road lane center tracking

The experiment I considered the road lane center reach and following, where the robot starts away from the final objective, as presented in the image sequence of the Figure 5.11a. The desired linear velocity v_d was set to 3.61 m/s (or 13 km/h). The Figure 5.11b-c shows the evolution of the error in the image features, converging to the zero condition at the center of the image (red dotted line in the Figure 5.11a). Note that, even with large variations in the features set (first 7 seconds of Figure 5.11b-c), the robot was able to accomplish its task in security.

During the car navigation, the VS validation in the IDWA is observed in the Figure 5.11d-e. In these figures, the VS represents the deliberative control calculations alone, on the other hand the VS+IDWA is the final result after validation in the equation (5.16). When the curves are different, it means that the vehicle control input is given by the reactive part of the VS+IDWA controller. In the first 15 seconds, once the errors in the features set result in large control outputs for the VS, the IDWA changes the v_1 and v_2 values in order to slowly converge the car to the road center. Then, after

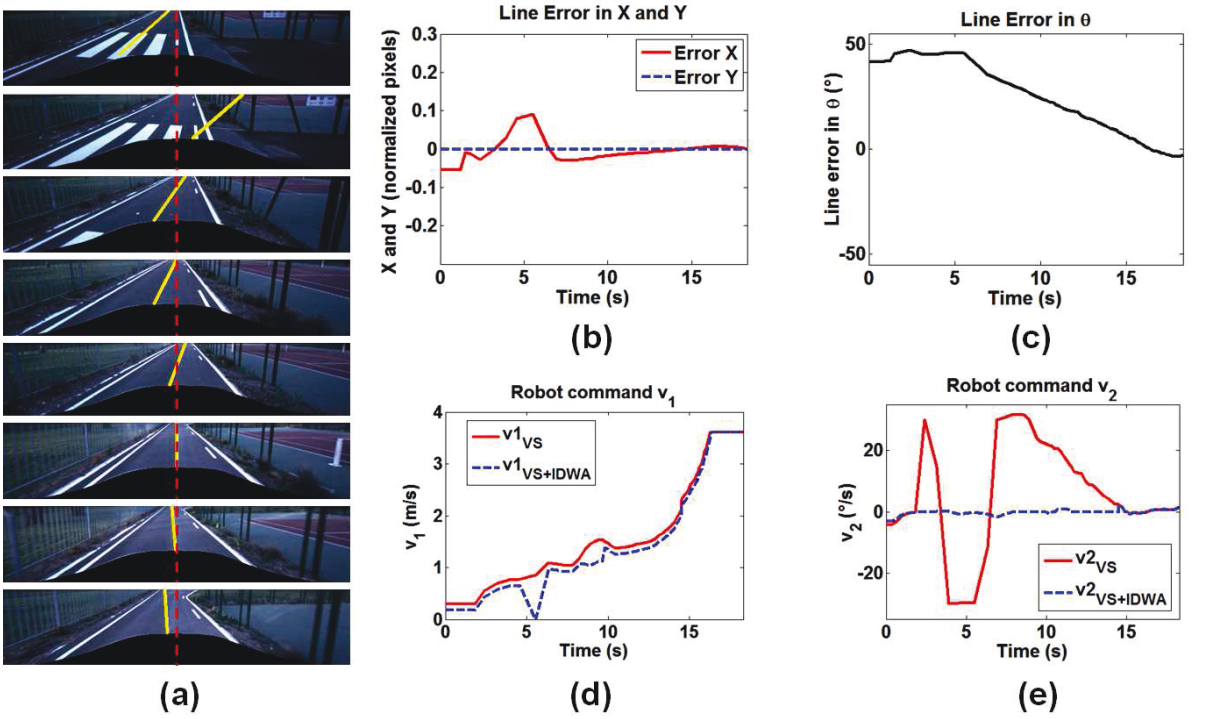


Figure 5.11 – Road center tracking experiment (I) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment, (b) and (c) show the evolution of the X, Y, and θ errors. The vehicle control inputs calculated by VS and the VS+IDWA are in (d) and (e).

15 seconds, the VS control is valid and guarantees a better error reduction than the IDWA approach, as proved in the simulation results of Figures 5.7 and 5.8. Just as the human driver behavior, the vehicle control inputs were performed smoothly during the experiment, validating the gains tuned in simulation.

Experiment II - Road following with obstacle avoidance

In the second experiment (II) the reactive capability of the controller was analyzed during an obstacle avoidance maneuver. For this purpose, a person was placed on the way to force the reactive obstacle avoidance. In the image sequence of the Figure 5.12a it is possible to see the car avoiding the person and moving safely at the maximum speed of $v_d = 1.5$ m/s (or 5.4 km/h).

In the Figure 5.12b-e, the obstacle avoidance is related to the first 23 seconds, where the VS outputs are invalid. The last 2 seconds were free from frontal obstacles and the VS+IDWA returned the valid VS outputs. Similarly to the experiment I, the VS+IDWA

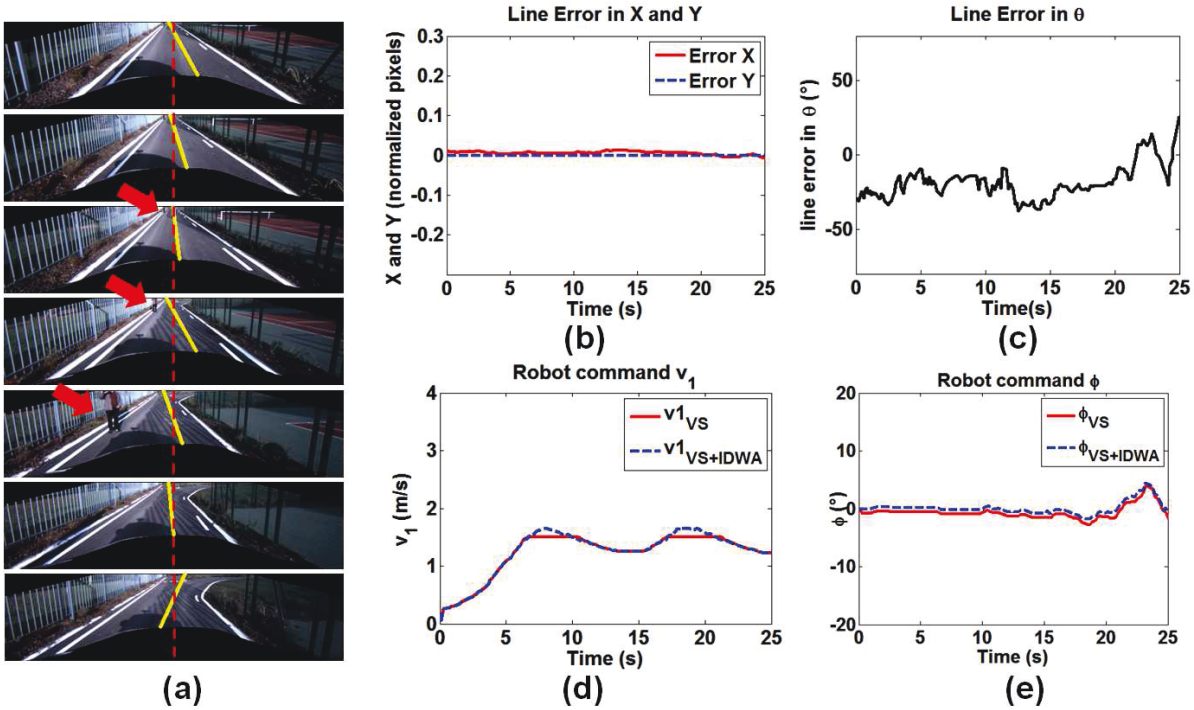


Figure 5.12 – Road center following with obstacle avoidance experiment (II) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment with the obstacle indicated by the red arrow, (b) and (c) show the evolution of the X, Y, and θ errors. The vehicle control inputs calculated by the VS and VS+IDWA are in (d) and (e).

always provided smooth vehicle control inputs, improving the comfort sensation for the passengers (Figure 5.12d-e).

Experiment III - Road following with collision avoidance

The third experiment (III) was performed to check the collision avoidance by the reactive system when there is no other possibility to go. In addition, the desired linear velocity was increased to 2.7 m/s (or 10 km/h) to check the final acceleration/deceleration applied to the car. The person was so positioned in the middle of the road, blocking completely the robot way to force it to stop. In the image sequence of the Figure 5.13a, the car follows the road lane center until stop just in front of the person.

Once the road center is obstructed, the image features detection vary considerably during the vehicle movement (Figure 5.13b-c), but the VS+IDWA always avoided hard movements which could lead the vehicle out of the road. We can note the acceleration

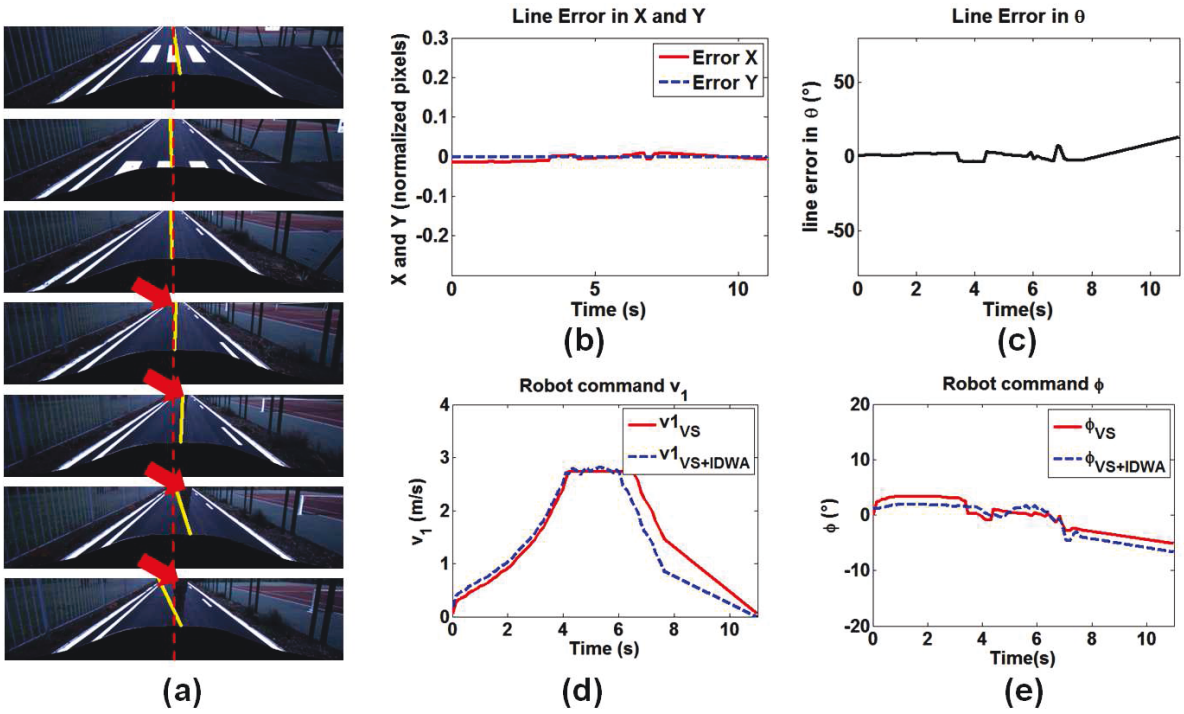


Figure 5.13 – Road center following with collision avoidance experiment (III) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment with the obstacle indicated by the red arrow, (b) and (c) show the evolution of the X, Y, and θ errors. The vehicle control inputs calculated by the VS and VS+IDWA are in (d) and (e).

and deceleration applied in the linear velocity control, taking almost the same time to go from 0 to $2.7m/s$ and vice-versa (Figure 5.13d-e). In practical situations, the robot reaction depends on the actuator dynamics considered during the VS+IDWA calculation.

For a better visualization of the local navigation control capabilities, a complete video sequence, including some experiments described here, is available in (Lima and Victorino, 2014a).

5.4 Conclusions

This chapter presented a new hybrid controller for vision-based local navigation of car-like robots in urban environments. By combining a VS approach, as deliberative control, and the IDWA, as reactive control, it allowed the road lane following with obstacle avoidance by our experimental vehicle. Using only the VS controller it is not possible to

guarantee the accomplishment of the local navigation task, caused by some limitations like: the path reaching problems, the constant linear velocity, and the obstacles in the environment. However, validating the VS outputs in the IDWA assured the safeness of the VS approach. In addition, if the VS outputs are not allowed, the IDWA optimization function give us a reactive alternative to safely complete the navigation task. Moreover, the VS control avoided the inherent opposition presented in the IDWA functions while following the road lane center.

Several tests were performed in the simulation environment over Matlab considering the car kinematics constraints, and some dynamics and sensors limitations, which provided a solid validation for the proposed solution. In this environment, a complete analysis about the controller gains was performed for a better setup.

Experiments in the real electric car-like robot APACHE showed the viability of the proposed methodology for local navigation with smooth control behavior. The complete strategy have integrated several resources from the car and environment perception systems.

It is important to mention that other Visual Servoing techniques could be integrated with the present solution to accomplish different tasks. Due to the nature of the presented approach, it can be applied with low-cost sensors and independent of high precision localization system. Based on these benefits, in the Chapter 6 this controller was integrated to a global navigation methodology using low-cost sensors and our autonomous vehicle.

Chapter 6

Global Navigation Management

Full autonomous systems must be able to accomplish the desired user tasks with no human intervention. The Chapter 5 presented an approach for local tasks accomplishment, guiding our experimental vehicle APACHE during road following navigation with the perceived information from the environment (Chapter 4). However, a full autonomous car must equally deal with the global navigation problems of go from a initial position (q_{init}) to a final destination (q_{goal}), passing through different roads and other scenarios conditions.

In the Section 2.3 several global navigation approaches were listed. They used waypoints to describe the robot path and, some of them, are highly dependent on GPS based localization to guide the robot to q_{goal}. But GPS sensors are very susceptible to errors, mainly in urban environments, where there are tall buildings, overpasses and tunnels (von Hundelshausen et al., 2008). This chapter presents a global navigation approach based on the topological representation of the path, divided on corridors and critical points (CPs). During the navigation on the corridors (the roads), which are connected by a topological representation, the local autonomous navigation strategy is used without explicitly localization of the robotic vehicle. When the vehicle arrives at a critical point (road intersections and roundabouts) a global localization phase takes place in a global navigation strategy, object of this chapter. The topological representation was constructed after a global path planning from q_{init} to q_{goal} based on a user-generated digital map. This chapter will present the methodology for our global autonomous navigation contribution (Pereira et al., 2014), generalizing and testing the navigation strategy in real scenarios with the experimental vehicle APACHE.

Following the system proposition of Figure 3.2, this navigation combines the environment information (Environment Perception block) surrounding the robot and its current estimated position (EKF block) to provide the correct setpoints for the Local Navigation Control block. Although we are using a GPS based localization for fast validation purposes, this is not a limitation for the approach, once digital maps can help with different localization possibilities, as well as vision cameras (Comport et al., 2011) and vehicle-to-infrastructure (V2I) communication (Hassan et al., 2013). However, the focus here is not to solve localization problems, but the car global navigation as a whole. The digital map applied in this work and its resources are described next.

6.1 OpenStreetMap: user-generated street maps

OpenStreetMap (OSM) is a free user-generating mapping platform for digital map with several detailed maps worldwide proposed by (Haklay and Weber, 2008). The OSM data include three basic geometric elements: *nodes*, *ways* and *relations*. The *nodes* are geometrical points interconnected by *ways* used to represent the geometry of the *way*. In case of roads, they also represent the intersections and roundabouts. To fit the geometry, the *ways* represent a set of *nodes* characterizing a specified object like road, building, railway, etc. The element *relations* constructs a relationship of geoobjects, where the relation may contain *nodes* and *ways*. The Figure 6.1 presents an example of the OSM data in a portion of the city of Compiègne, France. In Figure 6.1a it is shown the region of interest, extracted from the OSM using the global GPS coordinates (latitude and longitude). The OSM data is represented by a metadata file containing all the geometric elements, as shown in figure 6.1d. The geometric elements are represented by different types of nodes, identified in the file by keytags according to the origin of the node (building, railways, highways). In our case, it is necessary only the *ways* with the keytag “highway", as they correspond to the navigable area. The Figures 6.1b and 6.1c presents the respective drivable area filtered from a research on the metadata file (corresponding to the portion of the map in 6.1a) using the keytag "highway". Note that, the data from the OSM are rich in details, as many commercial digital maps, and provides some information about the road, like the number of lanes, maximum speed, and the road direction (see Figure 6.1d).

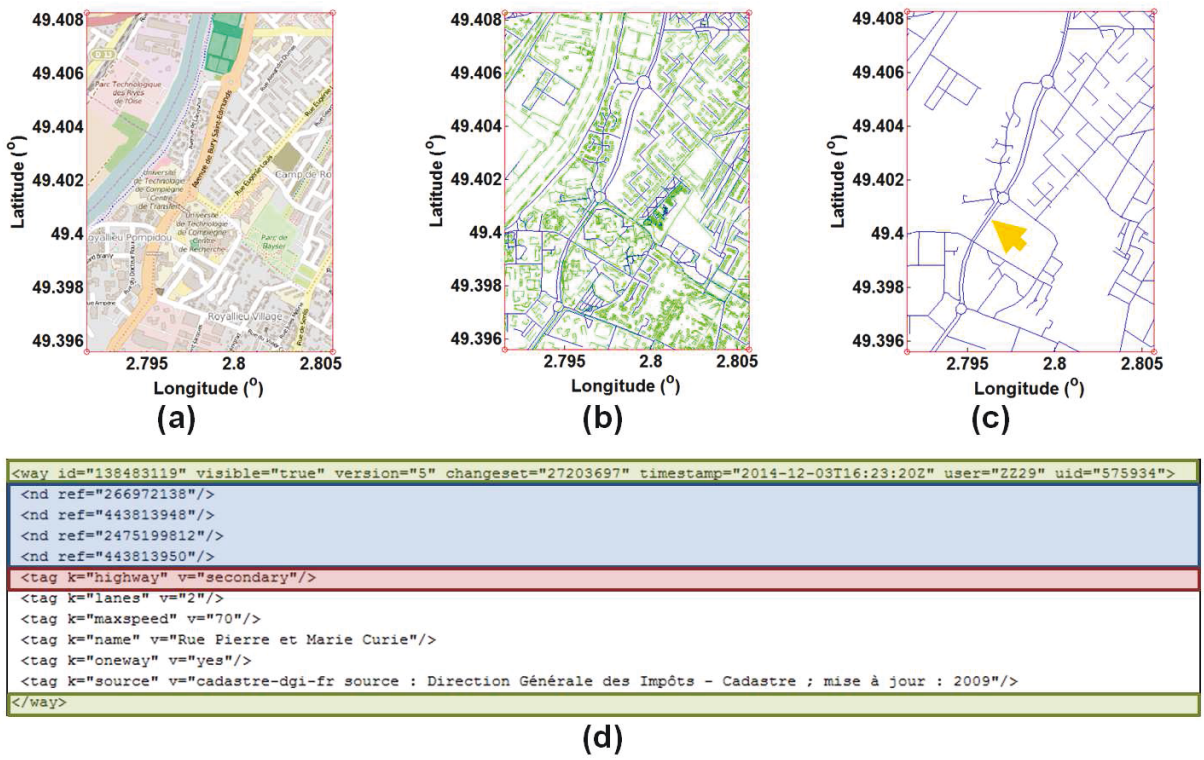


Figure 6.1 – Example of the OpenStreetMap data of a urban environment (a), all the extracted *nodes* (b), and the resulting *ways* with the tag "highway". A piece of the OSM metadata file for the road indicated by the yellow arrow is illustrated in (d), with the way definition (green), the keytag (red), the nodes (blue), and some parameters of the road (white).

In the Figure 6.2 both OSM (red) and differential GPS (DGPS) data (yellow) were plotted during a car navigation, in order to verify how far from the real road network are the OSM data. It shows the consistency of the data from the OSM database, comparing with the DGPS data. The error was around 5 to 10 meters, mainly in the road intersections, due to the nature of the OSM data provided by final users. The users' data are normally added by personal GPS measurements or line segments approximation in a drawing interface. For a global navigation application based on them, an error smaller than 10 meters should be considered to better deal with the road intersections. Otherwise, the points must be prerecorded by high precision devices and then inserted in the map (Ziegler et al., 2014), which are not the case for this current work.

The focus here is on the benefits of the OSM data for the geometric representation of the environment. This information will be combined in a routing table for the vehicle guidance, where the global navigation is acquired managing the vehicle state (following

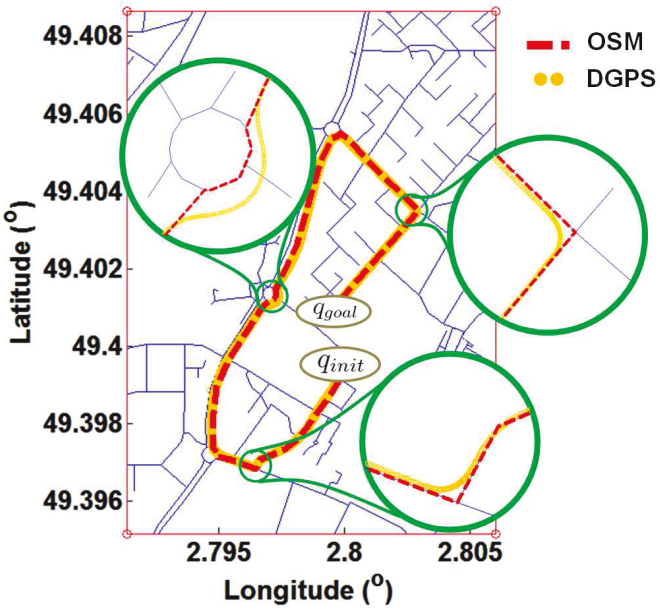


Figure 6.2 – OSM and GPS data comparison during the vehicle navigation.

the road or turning into the next road) in the routing table. The next section will describe this routing table principle and its construction after a global path planning.

6.2 Global route planning

In a simplified description of the urban environment, it can be divided into corridors and critical points (CPs) in a topological representation similar to the one applied to indoor places (Victorino et al., 2003b). The corridors are synonym for streets, avenues, etc., where the navigation task does not change and there are only one possible way to go until arrive into a CP. This task can be accomplished by some road following task (Dickmanns and Zapp, 1987; Kosecka et al., 1998; Lima and Victorino, 2014b), which does not obligatorily requires a localization system. The CPs represent the position where the local navigation task must take a decision to continue or change to a new task. This is commonly represented by the road intersections, traffic lights, pedestrian crossing, etc. By this assumption, every corridor must be connected to a CP and vice-versa. In this scenario, the car-like robot must localize itself only when it is near to a CP, avoiding several localization problems (von Hundelshausen et al., 2008).

To validate the present navigation approach, roads were considered as corridors and their intersections as CPs. Based on digital maps, provided by the OSM, a routing

table is created by structuring the global navigation from q_{init} to q_{goal}. For this, the first step is to perform the path planning, where the optimal path from q_{init} to q_{goal} is calculated. There are many techniques dealing with the path planning problem (Choset et al., 2005). Since dealing with road networks from digital maps, the optimal path search model was approximated to a graph traversal problem. In this case, we applied the A* algorithm (Hart et al., 1968), an evolution from the Dijkstra's algorithm with logarithmic time complexity. This search algorithm minimizes a cost function, defined by the Euclidean distance between two *nodes* and the car maximum speed, computing the path with the lowest driving time. The result is a list of *nodes*, or waypoints, in which the robot must pass. The nodes representing the road intersections are so filtered out forming the list of CPs (see Figure 6.3).

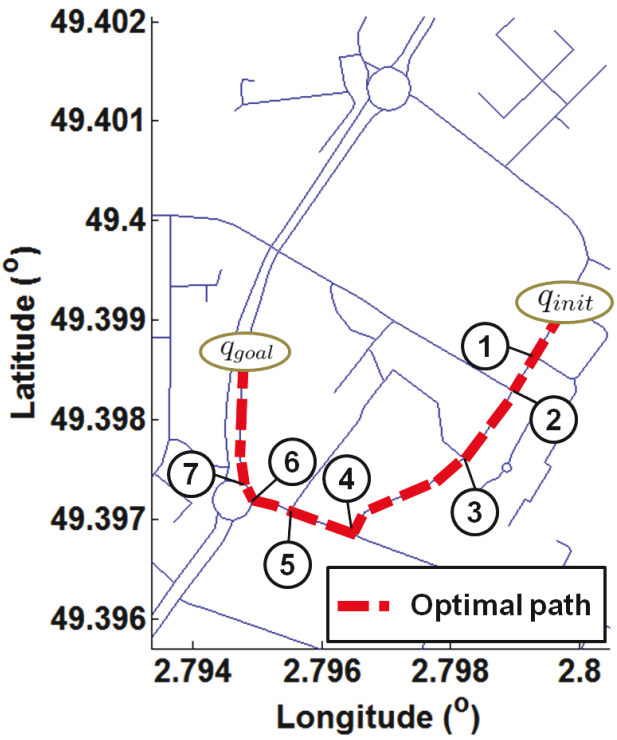


Figure 6.3 – Optimal path planning from q_{init} to q_{goal} with the list of *nodes* representing critical points. This used the OSM data in a portion of the city of Compiègne, France.

From this list, the local information is extracted and organized as a routing table. The Table 6.1 presents the routing table parameters for the path (red dotted line) illustrated in the Figure 6.4. This table was divided in two parts, Road description and CP geometry, where the first one is related to the road (*way*) leaving this CP, and the sec-

ond part represents the CP geometry with the angle and direction of every intersecting road. The first part has a constant number of columns, once the possible road parameters are always the same, and the second one has a number columns depending on the number of roads intersecting the CP. During the robot navigation, first comes the CP (with its associated geometry) and them the road leaving CP (with its description). The routing table parameters are:

Node is the list of CPs of the path ordered from q_{init} to q_{goal};

Lat and Long are the latitude and longitude coordinates of the CPs in degrees;

$\mathbf{V_{max}}$ defines the maximum speed limit of the road leaving the CP (in km/h);

$\mathbf{R_{lanes}}$ indicates the number of available lanes in the road leaving the CP (even if it is a two way road);

$\mathbf{R_{way}}$ classifies the road leaving the CP in one way (1) or not (2);

$\mathbf{R_a}$ tells if the CP is a roundabout point or not;

$\mathbf{N_{roads}}$ enumerates the roads connected to the CP;

$\mathbf{\Theta_{1..N_{roads}}}$ **and** $\mathbf{D_{1..N_{roads}}}$ form the sets of angle and direction for each road at the CP, ordered counterclockwise. $\Theta_{1..N_{roads}}$ are formed angles between the current road and the next one in the intersection. $D_{1..N_{roads}}$ indicate if the road is the path to follow(4), a wrong way (arrival) road (3), a right way (departure) road(2), an arrival/departure road(1), or if it is the goal point(0).

Table 6.1 – Routing table for the path represented in the Figure 6.4.

Road description						
Node	**Lat**	**Lon**	$\mathbf{V_{max}}$	$\mathbf{R_{lanes}}$	$\mathbf{R_{way}}$	$\mathbf{R_a}$
q_{init}	49.392	2.798	30	1	1	0
1	49.400	2.800	50	2	2	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

CP geometry					
Node	$\mathbf{N_{roads}}$	$\mathbf{\Theta_1}$	$\mathbf{D_1}$	$\mathbf{\Theta_2}$	$\mathbf{D_2}$
q_{init}	1	180	4	-	-
1	2	110	4	290	1
⋮	⋮	⋮	⋮	⋮	⋮

A complete routing table based on the OSM data is presented in the Table 6.2, for the optimal path (red dotted line) of Figure 6.3. A global navigation, based on the

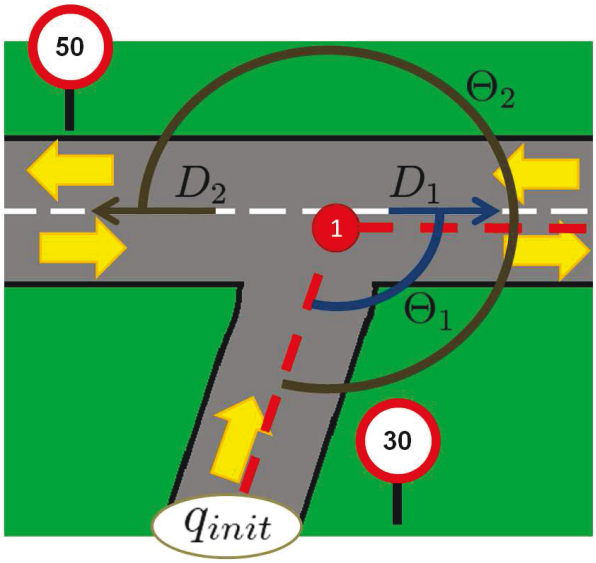


Figure 6.4 – Example of a road intersection to illustrate the routing table definitions, where the yellow arrows indicates the traffic way.

routing table, can be performed by following the directions indicated as path to follow (4) at each CP. The management of this table is presented next.

6.3 Global navigation

The global navigation takes into account the problem of guiding the robot from q_{init} to q_{goal}. Focusing on local navigation approaches to solve the entire problem, the global navigation was divided in two local tasks: road lane following (corridors) and road intersection maneuvers (at CPs). At this section, both navigation problems were solved based on the routing table, previously defined in the Section 6.2. However, it is necessary to switch between these two tasks, managing the car's motion when navigating in a CP.

6.3.1 Navigation management

For the car moving from q_{init} to q_{goal}, the role of the navigation management is to determine which local navigation task must be performed (road lane following or road intersection maneuver), regarding the routing table for its planned path. The selected task defines the image setpoint and all control parameters based on the robot

Table 6.2 – Routing table for the optimal path presented in the Figure 6.3.

Road description						
Node	**Lat**	**Lon**	$\mathbf{V_{max}}$	$\mathbf{R_{lanes}}$	$\mathbf{R_{way}}$	$\mathbf{R_a}$
q_{init}	49.399	2.800	30	1	2	0
1	49.399	2.799	30	1	2	0
2	49.398	2.799	30	1	2	0
3	49.398	2.798	30	1	2	0
4	49.397	2.797	50	1	2	0
5	49.397	2.796	50	1	2	0
6	49.397	2.795	50	2	1	1
7	49.397	2.795	50	2	1	1
q_{goal}	49.398	2.795	-	-	-	-

CP geometry							
Node	$\mathbf{N_{roads}}$	$\mathbf{\Theta_1}$	$\mathbf{D_1}$	$\mathbf{\Theta_2}$	$\mathbf{D_2}$	$\mathbf{\Theta_3}$	$\mathbf{D_3}$
q_{init}	1	180	4	-	-	-	-
1	2	180	4	271	1	-	-
2	3	93	1	177	4	278	1
3	2	89	1	165	4	-	-
4	2	97	4	278	1	-	-
5	2	75	1	188	4	-	-
6	2	88	4	268	3	-	-
7	2	114	4	268	2	-	-
q_{goal}	1	180	0	-	-	-	-

perception. Considering the localization method of the vehicle APACHE, based on a low-cost GPS (see Section 3.4), the world related position of the robot is provided with an error around $\pm 5m$. Thus, the navigation management is performed estimating the car position related to the CPs around during the movement, for an initial position near to q_{init}.

If the robot is located between two CPs (far away from both), this means that it is situated in a corridor. At corridors, there are only one direction to go and the local navigation *Road lane Following* is performed, as described in the subsection 6.3.2. When the robot is close enough to a CP, the information about the angles and directions are extracted from the routing table and the local navigation task is switched to *Road intersection maneuver* (as explained in the Section 6.3.3 bellow).

By analyzing these conditions, we observed that the management objective is to define how near/far the robot is from a CP, then switch between the local tasks. To do so, we performed the APACHE movement in the SEVILLE test track, defining the CPs in all road intersections, as seen in the Figure 6.5. The resulting trajectory, provided by its

localization system, always intersects the black circles containing the CPs. The circles consider the maximum localization and OSM errors of 10 meters. Thus, the vehicle is near to a CP when the distance between them is lower than 10 meters and faraway otherwise. Approaching a CP, the maximum vehicle speed can be reduced to improve the navigation behavior and also apply some localization technique based on the local information (Comport et al., 2011; Hassan et al., 2013).

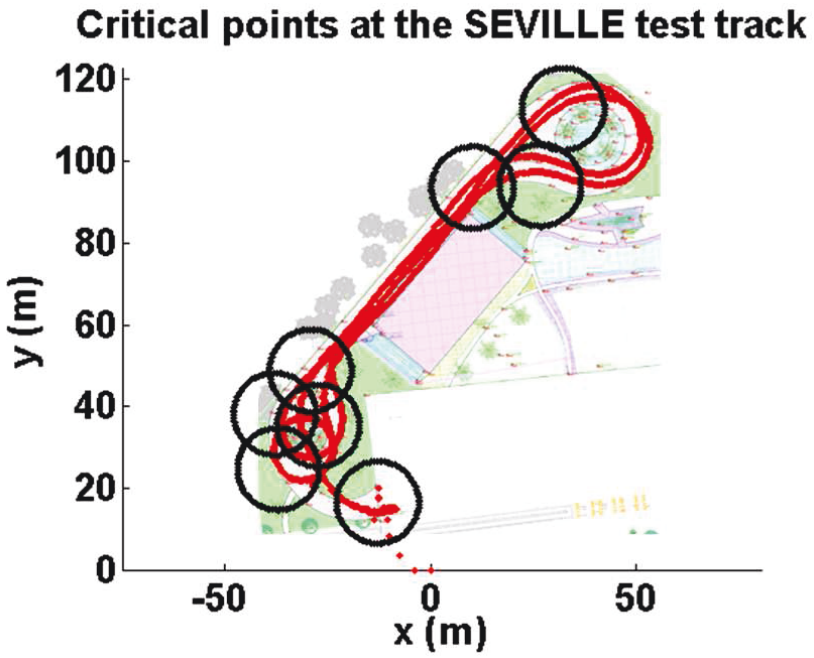


Figure 6.5 – Critical points for the test track SEVILLE, circumscribed by the black circles with radius equal to 10 meters, considering the localization and OSM errors. The vehicle trajectory is in red, always contained in the CPs.

6.3.2 Local navigation: road lane following

There are several approaches based on road lane following (Espiau et al., 1992; Lee et al., 1998; Broggi et al., 1999), which are able to control the vehicle without accurate position information. These approaches require a linear velocity setpoint for the longitudinal control, as well as a robust perception system to detect the road limits and lane marks for the lateral control. This work applies the VS+IDWA algorithm proposed in the Chapter 5 to accomplish the current local navigation task. The environment perception of Chapter 4 will provide the controller setpoint defined as the road lane center.

With the previous knowledge about road speed limit (V_{max}), number of lanes (R_{lanes}), and traffic way (R_{way}), provided by the routing table, the speed setpoint can be adjusted and the lane detection algorithm optimized for each road. In addition, with the traffic way, the obstacle overtaking can be better controlled, and movements in discordance with the traffic laws are as well avoided.

6.3.3 Local navigation: road intersection maneuvers

Approaches for road lane following do not traditionally deal with road intersection maneuvers, requiring some intervention at these situations. With this problem in mind, we propose a complement for the vehicle control, based on the previous VS+IDWA (Chapter 5), with an algorithm to artificially change the local environment information perceived at the local occupancy grid (see Section 4.5). This algorithm is based on the road intersection geometry, retrieved from the CPs at the routing table. It includes the directions that the car should take and the formed angle between the current and next roads.

The VS+IDWA provides safe movement to the vehicle by means of the reactive controller IDWA and the perceived obstacles in the navigable area. Artificially changing the navigable area with virtual obstacles, it is possible to force the IDWA to guide the vehicle to take right directions at intersections. In addition, providing the image features closer to the final destination make the IDWA to prioritize the velocities which lead the vehicle to this direction.

The first part of this approach is described in the Algorithm 6.1, with the detection of the road intersection corners. The main steps are illustrated in the Figure 6.6. It was inspired by the laser detection of tunnel intersections (Larsson et al., 2008), where a similar laser reading in the local occupancy grid defines the obstacles position in polar coordinates (r_{obst}, θ_{obst}) in a maximum perceived distance d_{max} (see the pink dots in the Figure 6.6b). It also uses the car's width C_{width}, to eliminate false positives, and the routing table information of N_{roads}, with the number of roads at the intersection and their geometry (Figure 6.6c). These parameters define the path possibilities for the current robot state (Figure 6.6a), represented by the arrows (red and yellow colors) in the navigable area (black pixels) of the occupancy grid.

Algoritmo 6.1 Find the road intersections based on the routing table information. Return the (row, θ) pairs representing the intersecting roads.

find_intersection_corners($r_{obst}, \theta_{obst}, d_{max}, C_{width}, N_{roads}$**)**

i_l, i_r = **find_main_road_limit(**$r_{obst}, \theta_{obst}, d_{max}$**)**
$\mathcal{I}_{polar}$ = Plot(r_{obst}, θ_{obst})
for all $\mathcal{I}_{polar}$ row between the columns i_l and i_r **do**
  find the n columns pairs $(\theta_{obst,l}[1..n], \theta_{obst,r}[1..n])$ in the row, representing the grey areas limited by the white pixels
  if $n == N_{roads}$ **then**
    if $E_{dist}(Cart(row, \theta_{obst,l}[1..n]), Cart(row, \theta_{obst,r}[1..n])) > C_{width}$ **then**
      return $row, \theta_{obst,l}[1..n], \theta_{obst,r}[1..n]$
    end if
  end if
end for
return false

Following this algorithm, initially, when the vehicle is close enough to the CP, the main road limit is calculated by the Algorithm 6.2, which return i_l and i_r. They represent the left and right θ_{obst} indexes with the smallest distance to the vehicle. These points reduce the search algorithm to the region defined between them. The final search is performed in the polar coordinates graphic $\mathcal{I}_{polar}$ (Figure 6.6d), with the radius measurements linked by the white line dividing the area among the vehicle (grey color) and faraway (black color). Thus, the algorithm search for the row discontinuities, which represent the white-grey and grey-white variations between i_l and i_r. These points are the candidates for the road intersection corners and must be checked by the following two filters:

— $\mathbf{N_{roads}}$ - verify if the number of n candidates is equal to the amount of expected roads at the intersection. One candidate is equal to the pair formed by the white-grey $(row, \theta_{obst,l}[k])$ and grey-white $(row, \theta_{obst,r}[k])$ variation, for $k = [1..n]$;

— $\mathbf{E_{dist}}$ - test if the Euclidean distance of each candidate, in Cartesian coordinates $(Cart)$, is higher than the car width (C_{width}).

The valid candidates are illustrated by dots in the Figure 6.6d. The main road limits and the intersections found are represented in the grid point-of-view in the Figure 6.6e. By the intersection geometry, the wrong vehicle directions are obstructed with virtual obstacles (also known as geofencing) at the distance equal to row, which results in the Figure 6.7a. Once the VS+IDWA requires an image features set for the control

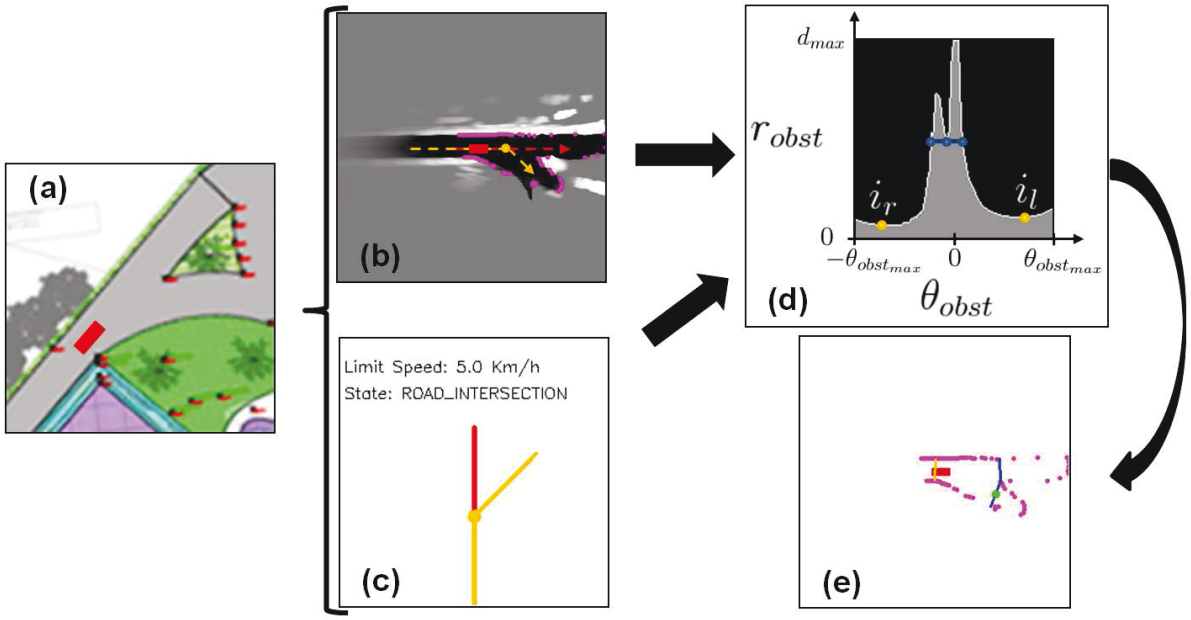


Figure 6.6 – Steps to find the road intersection corners for the car pose represented by the red rectangle (a). The respective local occupancy grid (b) and routing table state (c) illustrate the end of the navigable area given by pink dots with polar coordinates (r_{obst}, θ_{obst}) and the road structure at the intersection. These polar coordinates were used to detect the main road limits and to generate the graph (d), which defines the road intersections corners (in blue). They are better visualized in the grid point of view in (e), with the main road in yellow, the intersections in blue, and the destination road center in green.

calculation, the line connecting the vehicle to the next valid road center (green point at the Figure 6.7a) was projected in the image plane. This was performed by means of the equations 4.4 and 4.5, defining the new image feature line in the Figure 6.7b.

This approach is always executed when the vehicle is on the road intersection. Thus, the center of the destination road (green point) can be used as a condition to check if the robot has reached the destination road. If this point is closer enough to the front of the car, the road intersection maneuver was successfully performed and we can switch to the next local navigation method. This entire technique will be illustrated next in the experimental results at the real car APACHE.

6.4 Experimental results

To validate the proposed global navigation management, it was completely implemented in the embedded PC of the experimental car APACHE (see Section 3.1, using the

Algoritmo 6.2 Find the main road limit. Return the corners indexes (i_l, i_r) closer to the robot.

find_main_road_limit($r_{obst}, \theta_{obst}, d_{max}$**)**

1: $l_{d_{min}} = r_{d_{min}} = d_{max}$
2: **for** i = 1 to $SizeOf(\theta_{obst})/3$ **do**
3: **if** $l_{d_{min}} > r_{obst}\,[i]$ **then**
4: $l_{d_{min}} = r_{obst}\,[i]$
5: $i_l = i$
6: **end if**
7: **if** $r_{d_{min}} > r_{obst}\,[SizeOf(\theta_{obst}) - i + 1]$ **then**
8: $r_{d_{min}} = r_{obst}\,[SizeOf(\theta_{obst}) - i + 1]$
9: $i_r = i$
10: **end if**
11: **end for**
12: **return** i_l, i_r

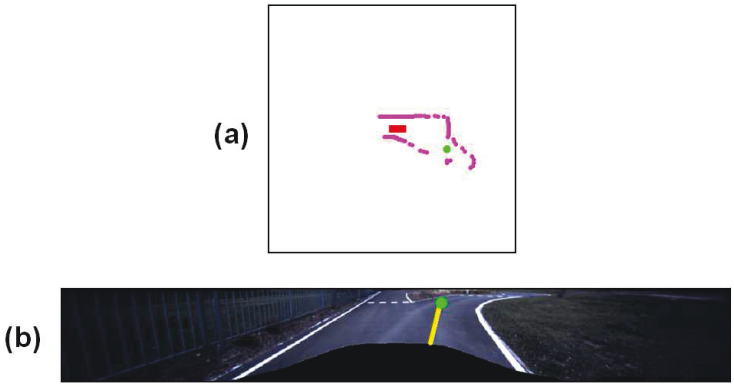


Figure 6.7 – Final r_{obst} and θ_{obst} with the virtual obstacles defined by the method described in the Algorithm 6.1 (a) and the new image features (b), oriented to the center of the next road (green point).

previous approaches for environment perception (Chapter 4) and local navigation control (Chapter 5). For the entire system running at 10 Hz, two navigation experiments were performed in order to present the global navigation management capabilities during the road lane following and the road intersection maneuvers. Both experiments were realized in the SEVILLE test track in full autonomous mode and their respective routing table constructed using the OSM data. Once this is a closed circuit, the path planning was not performed, requiring only the order of the CPs where the vehicle must pass. The localization tolerance for each CP was set to 10 m, regarding the GPS and OSM errors. These experiments are presented as follow.

6.4.1 Management at road lane following

Considering the results for the local navigation control based on the VS+IDWA (previously discussed in the Chapter 5), the experiment for road lane following management focused on the road description provided by the routing table during the car movement. The main information extracted here will be the maximum speed limit v_d for the roads and intersections. The experiment used the Table 6.3 to result the robot motion plotted in the Figure 6.8. This figure also presents some screenshots from the environment perception system to illustrate the car navigation during the road lane following tasks. Note that the localization variations make path following applications using GPS points unfeasible, but they are well fitted in the CPs tolerance radius.

Table 6.3 – Routing table for the global navigation management experiment I presented in the Figure 6.8.

Road description						
Node	**Lat**	**Lon**	$\mathbf{V_{max}}$	$\mathbf{R_{lanes}}$	$\mathbf{R_{way}}$	$\mathbf{R_a}$
q_{init}	49.4021061	2.7948685	13	1	1	0
1	49.4024539	2.7953365	7	1	2	0
2	49.4024597	2.7955631	5	1	2	1
3	49.4026169	2.7955953	7	1	1	1
4	49.4024800	2.7953713	13	1	2	0
q_{goal}	49.4021041	2.7948731	-	-	-	-

CP geometry					
Node	$\mathbf{N_{roads}}$	$\mathbf{\Theta_1}$	$\mathbf{D_1}$	$\mathbf{\Theta_2}$	$\mathbf{D_2}$
q_{init}	1	180	4	-	-
1	2	135	4	180	3
2	2	180	4	330	3
3	2	180	4	210	2
4	2	180	4	315	2
q_{goal}	1	180	0	-	-

On this previous movement, the global navigation management performs the speed limitation for each local road. The Figure 6.9 shows the linear velocity profile for the complete movement with the limitation performed for each routing table row. In this figure, the global management is evidenced during the CP transitions, where the road nominal speed (defined as 5, 7 and 13 km/h) is limited to 5 km/h to allow the intersection maneuvers. In order to mimic a human driver behavior and preparing the vehicle for the arrival of an intersection, we anticipate the speed limitation to 5 m before the CPs. This distance allows a smooth arrival in the intersection for the current speed

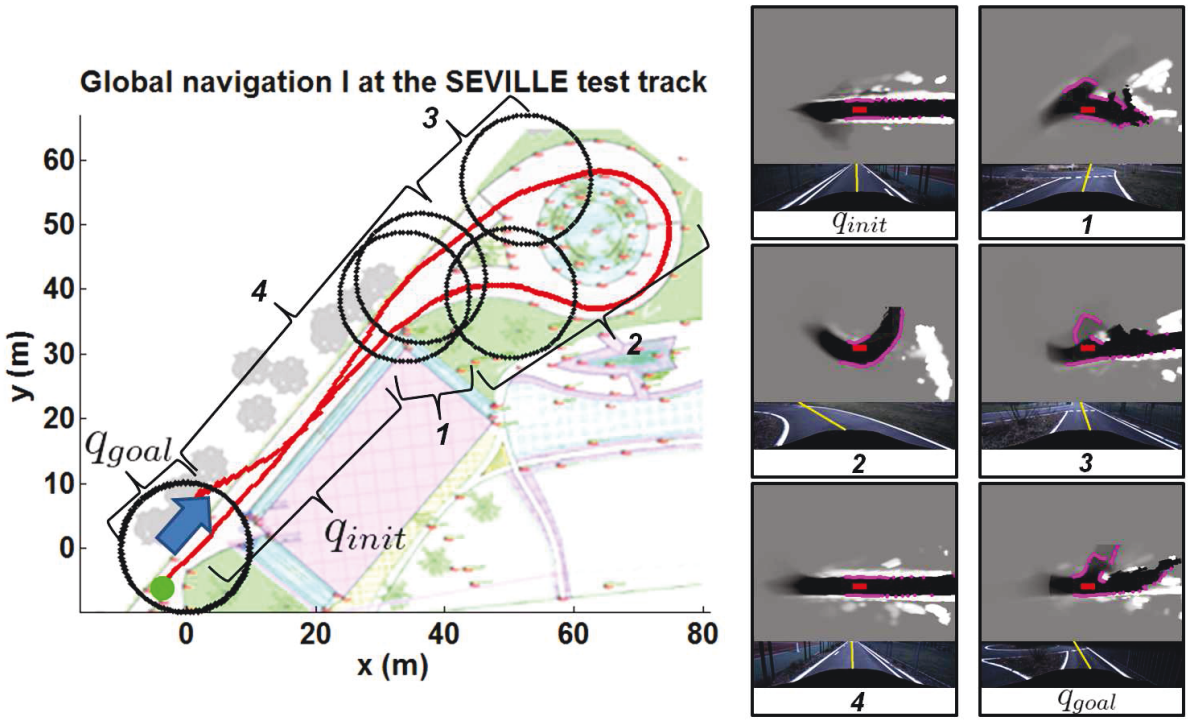


Figure 6.8 – Global navigation experiment I focusing on the road lane following for the routing table defined on Table 6.3. The initial position is defined by the blue arrow and the destination by the green point. The robot positions are illustrated in red and the localization tolerance around each CP are represented by the black circles.

limitations. However, for high speed applications, this distance must be related to the vehicle maximum deceleration for a smooth speed reduction.

6.4.2 Management at road intersection maneuvers

Extending the previous movement for the entire figure-eight circuit, we verify here the vehicle reaction at the road intersections. The Table 6.4 describes this movement, observed in the Figure 6.10 with some screenshots of the environment perception systems at the intersections. In these situations, the strategy proposed in the Subsection 6.3.3 modifies the occupancy grid information with virtual obstacles (blue dots). These obstacles combined with the new image feature (estimated from the green dot in the occupancy grid), allow the global navigation of the vehicle.

In addition, the Figure 6.11 presents in detail one of these maneuvers, with some sequential frames. In these frames it is possible to see the robustness of this approach during the movement, detecting the intersecting road with low variations as well as the

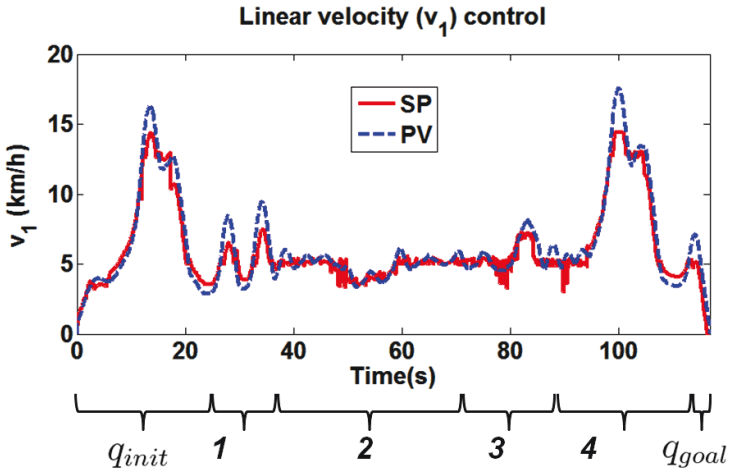


Figure 6.9 – Velocity profile for the global navigation experiment I of Figure 6.8.

next road center. Note that the roads geometry at the intersection do not matter on fact, but can be considered to enhance the detection capabilities. The most important element here is a good representation of the road area by the perception system. If the next road surface is not well detected for example, or it is not on the sensor's FOV, the vehicle will not be able to exit from the intersection situation.

6.5 Conclusions

Global navigation is an important task for many robots, and for car-like robots it is one of the main frontiers between a full and semi-autonomous system. In this chapter we presented a new sensor-based approach to deal with the global navigation problem in urban environments, using only low-cost sensors. We focused on two main tasks: road lane following and road intersection maneuvers. To accomplish it, the robot path was divided in critical points (CPs) for a topological representation of the global navigation task. These CPs were structured in a routing table, which describes the main road (lane, way, speed limit) and its geometry at the intersection. All the information required for it were retrieved from digital maps, provided by the OpenStreetMap (OSM). The management of this information allowed us to perform the global navigation of our vehicle.

The final system has integrated all the available resources of our experimental vehicle APACHE, detailed in the Chapters 3, 4 and 5, for localization, low-level control,

Table 6.4 – Routing table for the global navigation management experiment II presented in the Figure 6.10.

Road description						
Node	**Lat**	**Lon**	$\mathbf{V_{max}}$	$\mathbf{R_{lanes}}$	$\mathbf{R_{way}}$	$\mathbf{R_a}$
q_{init}	49.4021061	2.7948685	13	1	1	0
1	49.4024539	2.7953365	7	1	2	0
2	49.4024597	2.7955631	5	1	2	1
3	49.4026169	2.7955953	7	1	1	1
4	49.4024800	2.7953713	10	1	2	0
5	49.4020526	2.7947947	7	1	2	0
6	49.4019573	2.7946756	5	1	1	1
7	49.4018369	2.7946884	5	1	2	1
8	49.4019316	2.7948211	5	1	2	1
9	49.4020526	2.7947947	7	1	2	0
q_{goal}	49.4021061	2.7948685	-	-	-	-

CP geometry					
Node	$\mathbf{N_{roads}}$	$\mathbf{\Theta_1}$	$\mathbf{D_1}$	$\mathbf{\Theta_2}$	$\mathbf{D_2}$
q_{init}	1	180	4	-	-
1	2	135	4	180	3
2	2	180	4	330	3
3	2	180	4	210	2
4	2	180	4	315	2
5	2	180	4	225	3
6	2	180	4	330	2
7	2	180	3	245	4
8	2	180	4	225	3
9	2	135	4	315	2
q_{goal}	1	180	0	-	-

environment perception, and local navigation control. The two main tasks were successfully accomplished, following the image reference and avoiding obstacles, in a simulation environment and in real experiments. The results showed the viability of the solution, even in the presence of a reduced number of sensors for environment perception. Nevertheless, increasing the FOV and, consequently, the number of sensors used is fundamental to assure safe navigation in more situations.

Future works must deal with these perception limitations, enhancing the actual perception configuration for better intersection detection. Moreover, better localization techniques, like those presented by (Comport et al., 2011; Hassan et al., 2013) should be tested to avoid GPS problems in urban canyons when approaching an intersection. In addition, more road constraints must be considered, like road lanes, stop signs, the

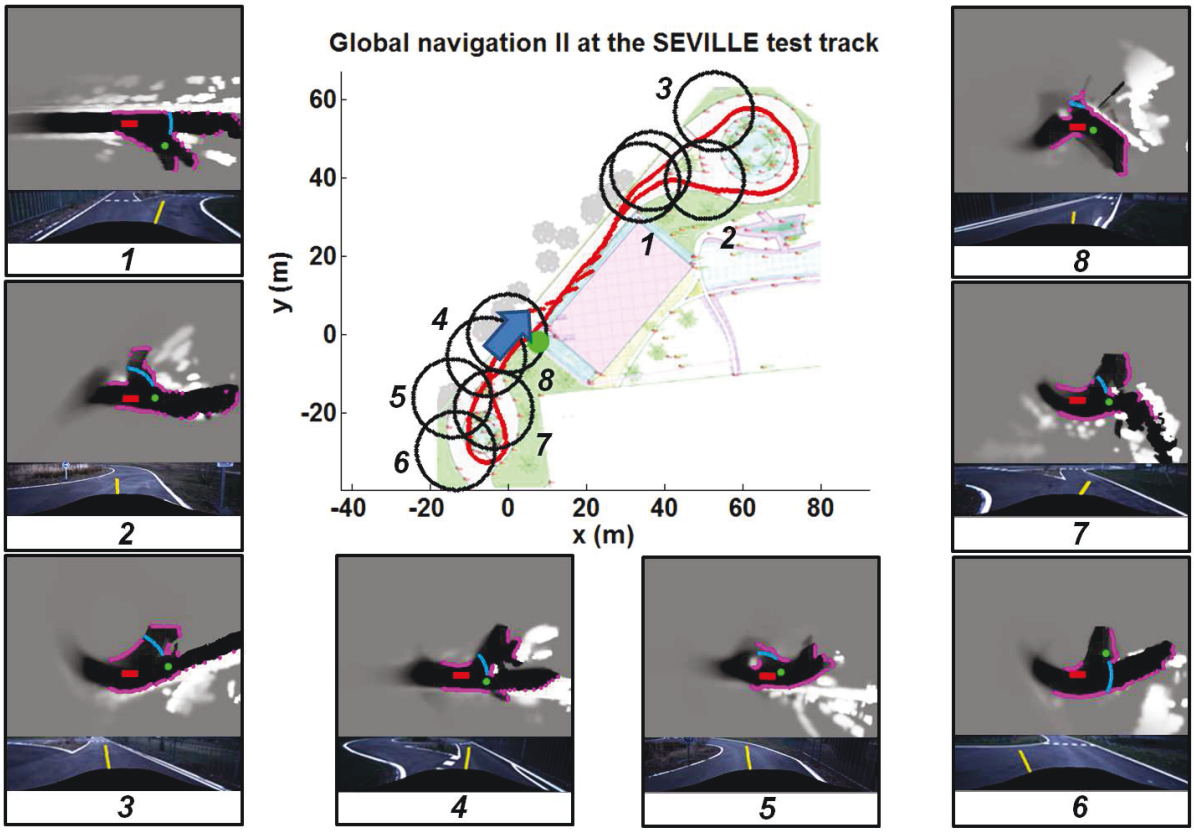


Figure 6.10 – Global navigation experiment II focusing on the road intersection maneuvers, for the routing table defined on Table 6.4. The initial position is defined by the blue arrow and the destination by the green point. The robot positions are illustrated in red and the localization tolerance around each CP are represented by the black circles.

road priority, etc., to allow road following and intersection maneuvers closer to real situations.

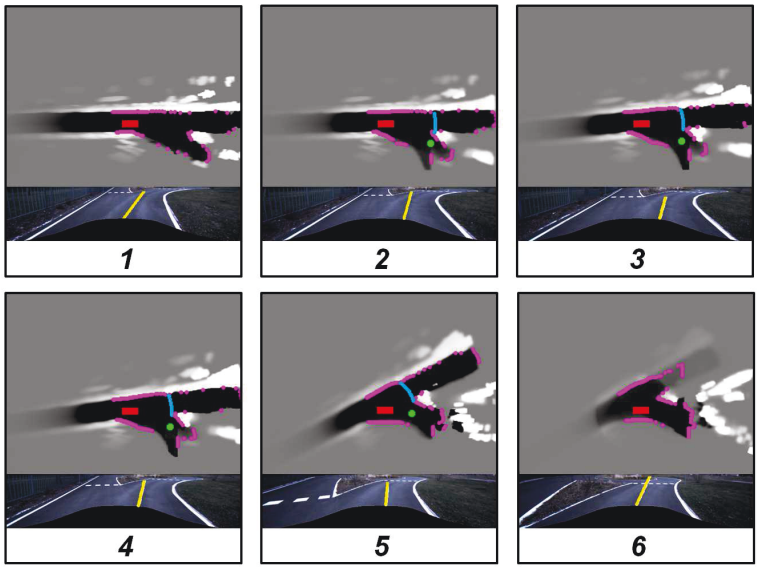


Figure 6.11 – Vehicle movement during a road intersection maneuver, emphasizing the virtual obstacles (blue dots) and the destination road center (green dot) used to define the new image feature in these conditions (yellow line).

Chapter 7

Conclusions and Future Works

This work presented some contributions for the sensor-based navigation problem applied to autonomous robotic automobiles. Beyond the several exiting contributions, described in the related work, we proposed a solution less dependent on high cost sensors and without any proprietary information prerecorded. The problem was so divided in three big areas: the environment perception, the local navigation control, and the global navigation management. In order to experimentally validate all techniques, a real electric vehicle was prepared with several components, like embedded computers, exteroceptive sensors, localization system, and low-level velocity controller. Moreover, a simulation environment was also developed to allow fast experiments in different scenarios.

In the environment perception, we presented contributions in the 2D/3D image processing context. They were developed to deal with common problems associated to the stereo vision cameras, like shadows, light reflections, and low texture variations. The first contribution was a road detection algorithm (Vitor et al., 2013), based on 2D image segmentation and classification in an artificial neural networks (ANN). This approach enhanced the 3D image data (from the stereo camera) with features extracted from the reference image (2D), for each formed segment, to perform a robust road detection. Once that many stereo vision problems are associated to sparse disparity map information, caused by wrong stereo matching, we avanced in this direction proposing another contribution, a disparity map refinement in (Lima et al., 2013). This refinement combined the previous 2D image segmentation with a RANSAC plane fitting to enhance the valid disparity information and filter out the variation of the data.

Although these were useful approaches, the execution time associated to these techniques was not enough to be used in our navigation system. In addition, they would have required a hard effort and working time to speed up these algorithms, which could compromise the development of the entire navigation system. Instead of that, we proposed another contribution, by using the 2D/3D image processing idea to improve and combine some existing techniques in a fast application for road and obstacle detection. All algorithms were tested in different urban scenarios and compared between them, with relation to a ground truth. The final solution was embedded in the experimental vehicle during the autonomous navigation experiments, and presented a precision of 89.7% and an accuracy of 80.6%.

For the local navigation control, we proposed an original contribution, that is a new hybrid control strategy called VS+IDWA (Lima and Victorino, 2014b). It combined a Visual Servoing (VS) approach as deliberative control and an Image-Based Dynamic Window Approach (IDWA) (Lima and Victorino, 2014c) as reactive control. The VS velocities directly applied in the vehicle do not guarantee the safe navigation of the vehicle, but their validation in the IDWA allowed a better convergence to the setpoint (compared with the IDWA control alone) and reactive obstacle avoidance. This controller was applied for road lane following with obstacle avoidance, where several validation experiments in simulation and real scenarios were performed. This solution considered the car kinematic and some dynamics constraints, only low-cost sensors, and no localization system. It is important to mention that several VS approaches could be applied in this context.

However, the previous local navigation control was not able to deal with road intersections, where there is more than one road available to follow. Therefore, we proposed a new global navigation management to artificially change the environment information and provide the right setpoint for the local navigation control. Starting from an initial contribution (Pereira et al., 2014), the routing table representation was improved and the navigation management was generalized and tested in a real scenario. Two navigation situations were considered: the road lane following and the road intersection maneuvers. In these situations, the control laws were performed in the sensors space.

The final system integrates all these capabilities in a modular scheme, user friendly and in a context of sustainable environment (an nowadays problem of the humanity).

In the practical context, it is easy to change and adapt each component of the proposed system to improve the final result. Nevertheless, there are several opportunities left for future works in each one of those presented areas.

In the environment perception it is important to consider more sensors in the vehicle surrounding, increasing the system FOV. Some improvements in the recall and accuracy rate could also be performed, integrating the benefits of each 2D/3D image processing techniques in a final solution. In a real scenario, moving obstacles must be considered by the perception system.

Once perceived the moving obstacles, they must be integrated in the final local navigation control, where techniques like (Seder and Petrovic, 2007) should be studied. In order to increase the vehicle speed, the car's dynamic model must be observed during the control. Other VS approaches could also be tested to allow, e.g., vehicle platooning and navigation in different scenarios. In addition, the influence of the path curvature in the features estimation must be studied to improve the lane tracking.

For the global navigation, all the elements listed previously must be integrated into the management. This means to increase the routing table application in different traffic scenarios, improving the vehicle overtaking and other behaviors. Alternatives for the localization system used can also be studied, avoiding completely GPS problems with a V2I communication scheme (Hassan et al., 2013). This may include problems associated with multi-vehicle navigation and system-to-system interaction, improving the decision-taking mechanisms in road intersections. Finally, navigation experiments in a real urban scenario must be considered for the complete validating of these methods.

Chapitre 8

Conclusions et Travaux Futurs

Ce travail a présenté des contributions pour le problème de navigation référencée capteurs appliqué aux voitures robotisées. Au-delà des plusieurs contributions déjà existantes, décrites dans les travaux connexes, nous avons proposé une solution moins dépendante des capteurs à coût élevé et sans aucune information propriétaire préenregistrée, normalement réalisée avec des systèmes de haute précision. Pour cela, le problème a été divisé en trois grandes axes : la perception de l'environnement, le contrôle de la navigation locale, et la gestion globale de la navigation. Afin de valider expérimentalement toutes les techniques, un véhicule électrique a été préparé avec plusieurs composants, comme des ordinateurs embarqués, des capteurs extéroceptifs, un système de localisation et un régulateur de vitesse de bas niveau. En outre, un environnement de simulation a également été développé pour permettre des expériences rapides dans différents scénarios.

Dans le volet perception de l'environnement, nous avons présenté des contributions dans le contexte du traitement d'image 2D/3D. Ils ont été conçus pour résoudre les problèmes communs liés aux caméras de vision stéréo, comme l'ombre, des reflets de lumière, et des faibles variations de texture. La première contribution est un algorithme de détection de la route (Vitor et al., 2013), basée sur la segmentation d'images 2D et une classification avec un réseaux de neurones artificiels (ANN). Cette approche ajoute des caractéristiques extraites dans l'image de référence (2D) aux données d'image 3D (de la caméra stéréo), pour chaque segment formé, pour effectuer une détection de la route d'une façon plus robuste. Une fois que nombreux problèmes de vision stéréo sont associés aux informations clairsemées de la carte de disparité, causées par une mauvaise correspondance stéréo, nous avons avancé dans cette direction en proposant

une autre contribution, un raffinement de la carte de disparité dans (Lima et al., 2013). Ce raffinement a combiné la segmentation d'images 2D précédente avec la méthode RANSAC, pour l'estimation des plans sur chaque segment. Cela a amélioré l'information de disparité valide et a filtré la variation sur les données.

Bien que ces approches fussent utiles, le temps d'exécution associé à ces techniques n'était pas suffisant pour être utilisé dans le système de navigation de notre véhicule. En outre, ils auraient exigé un grand effort et un temps important de travail pour accélérer ces algorithmes, ce qui pourrait compromettre le développement du système de navigation complet. Comme alternative, nous avons proposé une autre contribution, en utilisant le traitement d'image 2D/3D pour améliorer et combiner des techniques existantes dans une application plus rapide pour la détection de la route et des obstacles. Tous les algorithmes ont été testés dans différents scénarios urbains et comparés entre eux, par rapport à une vérité terrain. La solution finale a été intégrée dans le véhicule expérimental au cours des essais de navigation autonomes, et elle a présenté une fidélité de 89,7% et une justesse de 80,6%.

Une nouvelle stratégie de commande hybride appelé VS+IDWA (Lima and Victorino, 2014b) a été proposé, pour le contrôle de navigation locale. Elle a combiné une approche d'asservissement visuel (VS) pour le contrôle délibératif et l'approche de la fenêtre dynamique référencée image (IDWA) (Lima and Victorino, 2014c) pour le contrôle réactif. Les vitesses calculées par le VS ne garantissent pas la navigation sécurisée du véhicule. Toutefois, leur validation dans l'IDWA a permis une meilleure convergence vers la consigne (par rapport à la commande de générée seulement par l'IDWA) et un évitement d'obstacle réactif. Ce contrôleur a été appliqué pour le suivi de la voie routière avec évitement d'obstacles, où plusieurs essais de validation en simulation et en scénarios réels ont été réalisés. Cette solution a bien considéré des contraintes cinématiques et dynamiques de la voiture, aussi comme des capteurs à faible coût et aucun système de localisation. Il est important de mentionner que plusieurs méthodes de VS pourraient être appliquées dans ce contexte.

Cependant, le contrôle de navigation locale précédente n'était pas capable de traiter le mouvement dans des intersections de la route, où il y a plus d'une route disponible à suivre. Par conséquent, nous avons proposé une nouvelle gestion globale de la navigation pour changer artificiellement les informations de l'environnement perçu et fournir

la bonne consigne pour le contrôle de navigation locale. En partant de notre contribution décrite en (Pereira et al., 2014), la représentation de la table de routage a été améliorée et la gestion de la navigation a été généralisée et testé dans un scénario réel. Deux situations de navigation ont été testées : le suivi de la route et les manœuvres dans les intersections des routes. Dans ces situations, les lois de contrôle ont été effectuées dans l'espace des capteurs.

Le système final intègre toutes ces fonctionnalités dans un système modulaire et facile à utiliser. Dans le contexte pratique, il est possible de modifier et d'adapter chaque composant du système proposé pour améliorer le résultat final. Néanmoins, il y a plusieurs possibilités pour des travaux futurs dans chacun de ces domaines ici présentés.

Dans la perception de l'environnement, il est important d'augmenter la quantité des capteurs autour du véhicule, ce qui accroît le champ de vision FOV du système. Quelques améliorations dans les taux de rappel et précision pourraient également être effectuées, en intégrant les avantages de chaque technique de traitement d'image 2D/3D dans une solution finale. Dans un scénario réel, des obstacles mobiles doivent être pris en considération par le système de perception.

D'autres techniques, telles que (Seder and Petrovic, 2007), devraient être étudiées afin de considérer des obstacles mobiles dans l'approche de navigation locale. Afin d'augmenter la vitesse du véhicule, le modèle dynamique de la voiture doit être considéré dans la génération de la commande. Autres méthodes de VS pourraient également être testées pour permettre, par exemple, les platooning de véhicules et la navigation dans des différents scénarios.

Tous les éléments énumérés précédemment doivent être intégrées dans la gestion d'une navigation globale. Cela signifie augmenter l'application de la table de routage dans différents scénarios de circulation, pour améliorer le dépassement des véhicules et d'autres comportements. D'autres alternatives pour le système de localisation utilisé peuvent aussi être étudiés, en évitant les problèmes de GPS avec les solutions de communication entre véhicules et infrastructure (V2I) (Hassan et al., 2013). Cela peut inclure des problèmes associés à la navigation multi-véhicule et l'interaction système-à-système, améliorant les mécanismes de décision dans les intersections de routes. Enfin, plusieurs essais de navigation dans un scénario urbain réel doivent être considéré pour la validation complète de ces méthodes.

References

AdasWorks (2015). Software for self-driving cars. `http://adasworks.com/`.

Alvarez, J. and Lopez, A. (2011). Road detection based on illuminant invariance. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):184 –193.

Arras, K., Persson, J., Tomatis, N., and Siegwart, R. (2002). Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 3050–3055.

Audigier, R. and de Alencar Lotufo, R. (2010). Relationships between some watershed definitions and their tie-zone transforms. *Image Vision Comput*, 28(10):1472–1482.

Banz, C., Hesselbarth, S., Flatt, H., Blume, H., and Pirsch, P. (2010). Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *Embedded Computer Systems (SAMOS), 2010 International Conference on*, pages 93–101.

Bertozzi, M., Broggi, A., and Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1–16.

Beyeler, M., Mirus, F., and Verl, A. (2014). Vision-based robust road lane detection in urban environments. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4920–4925.

Bilodeau, M., Yu, X., and Beucher, S. (1990). *Road segmentation by watersheds algorithms*. CMMMM. ENSMP, Fontainebleau.

Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296.

Braid, D., Broggi, A., and Schmiedel, G. (2006). The TerraMax autonomous vehicle. *Journal of Field Robotics*, 23(9):693–708.

Brock, O. and Khatib, O. (1999). High-speed navigation using the global dynamic window approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 341–346.

Broggi, A., Bertozzi, A., Fascioli, A., and Guarino, C. (1999). The argo autonomous vehicles vision and control systems. *International Journal on Intelligent Control Systems*, 3(4):409–441.

Broggi, A., Caraffi, C., Fedriga, R. I., and Grisleri, P. (2005). Obstacle detection with stereo vision for off-road vehicle navigation. In *Proceedings of the International IEEE Workshop on Machine Vision for Intelligent Vehicles*, pages 1–8.

Broggi, A., Medici, P., Zani, P., Coati, A., and Panciroli, M. (2012). Autonomous vehicles control in the VisLab Intercontinental Autonomous Challenge. *Annual Reviews in Control*, 36(1):161–171. ISSN: 1367-5788.

Buehler, M., Lagnemma, K., and Singh, S. (2008). Special issue on the 2007 DARPA Urban Challenge, part I-III. In Buehler, M., Lagnemma, K., and Singh, S., editors, *Journal of Field Robotics*, volume 25, pages 423–860.

Cadenat, V., Swain, R., Soueres, P., and Devy, M. (1999). A controller to perform a visually guided tracking task in a cluttered environment. In *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 2, pages 775–780 vol.2.

Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In Kriegman, D., Hager, G., and Morse, A., editors, *The confluence of vision and control*, volume 237 of *Lecture Notes in Control and Information Sciences*, pages 66–78. Springer London.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. i. basic approaches. *Robotics Automation Magazine, IEEE*, 13(4):82–90.

Chen, Y.-L., Sundareswaran, V., Anderson, C., Broggi, A., Grisleri, P., Porta, P., Zani, P., and Beck, J. (2009). Terramax: Team oshkosh urban robot. In Buehler, M., Iagnemma, K., and Singh, S., editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 595–622. Springer Berlin Heidelberg.

Cherubini, A. and Chaumette, F. (2013). Visual navigation of a mobile robot with laser-based collision avoidance. *The International Journal of Robotics Research*, 32(2):189–205.

Cherubini, A., Chaumette, F., and Oriolo, G. (2008). An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 108–113.

Cherubini, A., Chaumette, F., and Oriolo, G. (2011). Visual servoing for path reaching with nonholonomic robots. *Robotica*, 29:1037–1048.

Cherubini, A., Spindler, F., and Chaumette, F. (2014). Autonomous visual navigation and laser-based moving obstacle avoidance. *Intelligent Transportation Systems, IEEE Transactions on*, 15(5):2101–2110.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA.

Comport, A., Meilland, M., and Rives, P. (2011). An asymmetric real-time dense visual localisation and mapping system. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 700–703.

Courbon, J., Mezouar, Y., and Martinet, P. (2009). Autonomous navigation of vehicles from a visual memory using a generic camera model. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):392–402.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339.

Dias, J., Pereira, G., and Palhares, R. (2014). Longitudinal model identification and velocity control of an autonomous car. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1–11.

Dickmanns, E., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Thomanek, F., and Schiehlen, J. (1994). The seeing passenger car 'vamors-p'. In *Intelligent Vehicles '94 Symposium, Proceedings of the*, pages 68–73.

Dickmanns, E., Mysliwetz, B., and Christians, T. (1990). An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(6):1273–1284.

Dickmanns, E. D. and Zapp, A. (1987). Autonomous high speed road vehicle guidance by computer vision. In *10th* IFAC *World Congress Munich*, pages 232–237.

Dixon, J. and Henlich, O. (1997). Mobile robot navigation. Technical report, Imperial College, London.

Dornaika, F., Alvarez, J., Sappa, A., and Lopez, A. (2011). A new framework for stereo sensor pose through road segmentation and registration. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):954 –966.

Einicke, G. A. (2012). *Smoothing, Filtering and Prediction – Estimating The Past, Present and Future*, chapter Nonlinear Prediction, Filtering and Smoothing. InTech.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.

Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *Robotics and Automation, IEEE Transactions on*, 8(3):313–326.

Faugeras, O. (1993). *Three-dimensional computer vision: A geometric view point*. MIT Press, Cambridge.

Fernandes, L. C., Souza, J. R., Pessin, G., Shinzato, P. Y., Sales, D., Mendes, C., Prado, M., Klaser, R., Magalhães, A. C., Hata, A., Pigatto, D., Branco, K. C., Jr., V. G., Osorio, F. S., and Wolf, D. F. (2014). Carina intelligent robotic car: Architectural design and applications. *Journal of Systems Architecture*, 60(4):372 – 392.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Flórez, S. A. R., Frémont, V., Bonnifait, P., and Cherfaoui, V. (2014). Multi-modal object detection and localization for high integrity driving assistance. *Machine Vision and Applications*, 25(3):583–598.

Folio, D. and Cadenat, V. (2006). A redundancy-based scheme to perform safe vision-based tasks amidst obstacles. In *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 13–18.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, 4(1):23–33.

Frazzoli, E., Dahleh, M. A., and Feron, E. (2000). Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129.

Gao, Y., Ai, X., Wang, Y., Rarity, J., and Dahnoun, N. (2011). U-v-disparity based obstacle detection with 3d camera and steerable filter. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 957–962.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361.

Geiger, A., Roser, M., and Urtasun, R. (2011a). Efficient large-scale stereo matching. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part I*, ACCV'10, pages 25–38.

Geiger, A., Ziegler, J., and Stiller, C. (2011b). Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968.

Gonzalez, R. C. and Woods, R. E. (2001). *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

Guizzo, E. (2011). How Google's Self-Driving Car Works. Online.

Gupta, R. and Cho, S. Y. (2010). A color-based approach for disparity refinement. In *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on*, pages 664 –667.

Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18.

Halterman, R. and Bruch, M. (2010). Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection. In *Proceedings of the SPIE: Unmanned Systems Technology XII*, pages 76920D–76920D–8.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107.

Hassan, O., Adly, I., and Shehata, K. (2013). Vehicle localization system based on ir-uwb for v2i applications. In *Computer Engineering Systems (ICCES), 2013 8th International Conference on*, pages 133–137.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 2 edition.

Hentschel, M. and Wagner, B. (2010). Autonomous robot navigation based on openstreetmap geodata. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1645–1650.

Hiroshi, T. K., Kano, H., Kimura, S., Yoshida, A., and Oda, K. (1995). Development of a video-rate stereo machine. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3*, page 3095, Washington, DC, USA. IEEE Computer Society.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814–.

Hu, Z. and Uchimura, K. (2005). U-v-disparity: an efficient algorithm for stereovision based scene analysis. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 48–54.

Irie, K. and Tomono, M. (2012). Localization and road boundary recognition in urban environments using digital street maps. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4493–4499.

Jamasmie, C. (2009). Lonely trucks in a lonely place: Autonomous trucks debut in Chile's desert. MINING.COM *a mine of information*, 2:23–24.

Ji, Z. and Prokhorov, D. (2008). Radar-vision fusion for object classification. In *Automation Congress, 2008. WAC 2008. World*, pages 1–6.

Jurie, F., Rives, P., Gallice, J., and Brame, J. (1994). High-speed vehicle guidance based on vision. *Control Engineering Practice*, 2(2):289 – 297.

Kang, Y., Alves de Lima, D., and Correa Victorino, A. (2014a). Driving behavior correction in human-vehicle interaction based on dynamic window approach. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, pages 485–490.

Kang, Y., Alves de Lima, D., and Victorino, A. (2014b). An approach of human driving behavior correction based on dynamic window approach. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 304–309.

Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515.

Kosecka, J., Blasi, R., Taylor, C., and Malik, J. (1998). A comparative study of vision-based lateral control strategies for autonomous highway driving. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 1903–1908 vol.3.

Kurdej, M., Moras, J., Cherfaoui, V., and Bonnifait, P. (2015). Map-aided evidential grids for driving scene understanding. *Intelligent Transportation Systems Magazine, IEEE*, 7(1):30–41.

Labayrade, R., Aubert, D., and Tarel, J. P. (2002). Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Proceedings of the IEEE Symposium on Intelligent Vehicles*, volume 2, pages 646–651.

Lamiraux, F. and Laumond, J. P. (2001). Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation*, 17:498–502.

Larsson, J., Broxvall, M., and Saffiotti, A. (2008). Laser based intersection detection for reactive navigation in an underground mine. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2222–2227.

Lee, S., Boo, K., Shin, D., and Lee, S. (1998). Automatic lane following with a single camera. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1689–1694 vol.2.

Lima, D. and Victorino, A. (2014a). VERVE and ROBOTEX – VS+IDWA for local navigation. `http://youtu.be/FZxAU4-jCcA?list=PLqKAHZQ-ozS-kPvkcUoyxnV7xTBit0w5Z`.

Lima, D. and Victorino, A. (2014b). A visual servoing approach for road lane following with obstacle avoidance. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 412–417.

Lima, D. A. and Pereira, G. A. S. (2013). Navigation of an Autonomous Car Using Vector Fields and the Dynamic Window Approach. *Journal of Control, Automation and Electrical Systems*, 24(1–2):106–116.

Lima, D. A. and Victorino, A. C. (2014c). An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments. In *Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (WMEPC), 2014 IEEE ICRA Workshop on*, pages 120–125.

Lima, D. A., Vitor, G. B., Victorino, A. C., and Ferreira, J. V. (2013). A disparity map refinement to enhance weakly-textured urban environment data. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6.

Luca, A. D., Oriolo, G., De, A., and Samson, C. (1998). *Robot Motion Planning and Control*, volume 229, chapter Feedback Control Of A Nonholonomic Car-Like Robot, pages 171–253. Springer Berlin / Heidelberg.

Luettel, T., Himmelsbach, M., and Wuensche, H. J. (2012). Autonomous ground vehicles - concepts and a path to the future. *Proceedings of the IEEE*, 100(Special Centennial Issue):1831–1839.

Malis, E., Chaumette, F., and Boudet, S. (1999). 2 1/2 d visual servoing. *Robotics and Automation, IEEE Transactions on*, 15(2):238–250.

Mattern, N., Schubert, R., and Wanielik, G. (2010). High-accurate vehicle localization using digital maps and coherency images. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 462–469.

Miranda Neto, A., Correa Victorino, A., Fantoni, I., and Ferreira, J. (2013). Real-time estimation of drivable image area based on monocular vision. In *Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE*, pages 63–68.

Mitsou, N. and Tzafestas, C. (2007). Temporal occupancy grid for mobile robot dynamic environment mapping. In *Control Automation, 2007. MED '07. Mediterranean Conference on*, pages 1–8.

Moras, J., Cherfaoui, V., and Bonnifait, P. (2014). Evidential grids information management in dynamic environments. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–7.

Moras, J., Rodriguez, F., Drevelle, V., Dherbomez, G., Cherfaoui, V., and Bonnifait, P. (2012). Drivable space characterization using automotive lidar and georeferenced map information. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 778 –783.

Nedevschi, S., Danescu, R., Marita, T., Oniga, F., Pocol, C., Sobol, S., Tomiuc, C., Vancea, C., Meinecke, M., Graf, T., To, T. B., and Obojski, M. (2007). A sensor for urban driving assistance systems based on dense stereovision. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 276 –283.

Nguyen, T.-N., Michaelis, B., Al-Hamadi, A., Tornow, M., and Meinecke, M. (2012). Stereo-camera-based urban environment perception using occupancy grid and object tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):154–165.

Ogren, P. and Leonard, N. (2005). A convergent dynamic window approach to obstacle avoidance. *Robotics, IEEE Transactions on*, 21(2):188–195.

Ohishi, K., Miyazaki, M., Fujita, M., and Ogino, Y. (1992). Force control without force sensor based on mixed sensitivity H^∞ design method. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 1356–1361 vol.2.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66.

Park, W.-J., Kim, B.-S., Seo, D.-E., Kim, D.-S., and Lee, K.-H. (2008). Parking space detection using ultrasonic sensor in parking assistance system. In *Proceedings of the IEEE Symposium on Intelligent Vehicles*, pages 1039–1044.

Pereira, E. C. M., Lima, D. A., and Victorino, A. C. (2014). Autonomous vehicle global navigation approach associating sensor based control and digital maps. In *Robotics and Biomimetics, 2014. ROBIO '14. IEEE International Conference on*, pages 2404–2409.

Pereira, G. A. S., Pimenta, L. C. A., Fonseca, A. R., Corrêa, L. d. Q., Mesquita, R. C., Chaimowicz, L., de Almeida, D. S. C., and Campos, M. F. M. (2009). Robot navigation in multi-terrain outdoor environments. *The International Journal of Robotics Research*, 28(6):685–700.

Ploeg, J., Shladover, S., Nijmeijer, H., and van de Wouw, N. (2012). Introduction to the special issue on the 2011 grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):989–993.

Pomerleau, D. and Jochem, T. (1996). Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert*, 11(2):19–27.

Rahman, A., Verma, B., and Stockwell, D. (2012). An hierarchical approach towards road image segmentation. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1 –8.

Rebai, K. and Azouaoui, O. (2009). Bi-steerable robot navigation using a modified dynamic window approach. In *Proceedings of the 6th International Symposium on Mechatronics and its Applications*, pages 1–6.

Rebai, K., Azouaoui, O., Benmami, M., and Larabi, A. (2007). Car-like robot navigation at high speed. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 2053–2057.

Regele, R. (2008). Using ontology-based traffic models for more efficient decision making of autonomous vehicles. In *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on*, pages 94–99.

Rives, P. (2000). Visual servoing based on epipolar geometry. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, pages 602–607 vol.1.

Saranrittichai, P., Niparnan, N., and Sudsang, A. (2013). Robust local obstacle avoidance for mobile robot based on dynamic window approach. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTICON), 2013 10th International Conference on*, pages 1–4.

Schindler, A. (2013). Vehicle self-localization with high-precision digital maps. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 141–146.

Seder, M. and Petrovic, I. (2007). Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1986–1991.

Sengupta, S., Sturgess, P., Ladicky, L., and Torr, P. (2012). Automatic dense visual semantic mapping from street-level imagery. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 857–862.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton.

Shinzato, P. Y. and Wolf, D. F. (2011). A road following approach using artificial neural networks combinations. *J. Intell. Robotics Syst.*, 62(3-4):527–546.

Silveira, G., Malis, E., and Rives, P. (2006). Visual servoing over unknown, unstructured, large-scale scenes. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 4142–4147.

Soquet, N., Aubert, D., and Hautiere, N. (2007). Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In *Proceedings of the IEEE Symposium on Intelligent Vehicles*, pages 160–165.

Soualmi, B., Sentouh, C., Popieul, J., and Debernard, S. (2014). Automation-driver cooperative driving in presence of undetected obstacles. *Control Engineering Practice*, 24(0):106 – 119.

Souza, A. and Goncalves, L. (2012). 2.5-dimensional grid mapping from stereo vision for robotic navigation. In *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, pages 39–44.

Svestka, P., Overmars, M. H., Overmars, M. H., and Overmars, M. H. (1995). Motion planning for car-like robots using a probabilistic learning approach. *International Journal of Robotics Research*, 16.

Tagne, G., Talj, R., and Charara, A. (2015). Design and validation of a robust immersion and invariance controller for the lateral dynamics of intelligent vehicles. *Control Engineering Practice*, 40(0):81–92.

Tao, Z., Bonnifait, P., Fremont, V., and Ibanez-Guzman, J. (2013). Lane marking aided vehicle localization. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 1509–1515.

Theuwissen, A. J. P. (1995). *Solid-State Imaging with Charge-Coupled Devices*. Springer Netherlands.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W. R., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., and Ferguson, D. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466.

Usher, K., Ridley, P., and Corke, P. (2003). Visual servoing of a car-like vehicle - an application of omnidirectional vision. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 4288–4293 vol.3.

Victorino, A. C., Rives, P., and Borrelly, J.-J. (2003a). Safe Navigation for Indoor Mobile Robots, PartI: A Sensor-Based Navigation Framework. *International Journal of Robotic Research*, 22:1005–1019.

Victorino, A. C., Rives, P., and Borrelly, J.-J. (2003b). Safe navigation for indoor mobile robots,partii: Exploration, self-localization and map building. *International Journal of Robotics Research*, 22(12):1019–1041.

Vitor, G. B. (2014). *Urban environment perception and navigation using robotic vision: conception and implementation applied to automous vehicle*. PhD thesis, Universidade Estadual de Campinas and Université de Technologie de Compiègne.

Vitor, G. B., Lima, D. A., Victorino, A. C., and Ferreira, J. V. (2013). A 2d/3d vision based approach applied to road detection in urban environments. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 952–957.

von Hundelshausen, F., Himmelsbach, M., Hecker, F., Mueller, A., and Wuensche, H.-J. (2008). Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9):640–673.

Wei, J., Snider, J., Kim, J., Dolan, J., Rajkumar, R., and Litkouhi, B. (2013). Towards a viable autonomous driving research platform. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 763–770.

Wei, X., Phung, S. L., and Bouzerdoum, A. (2011). Pedestrian sensing using time-of-flight range camera. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 43–48.

Xu, L., Au, O., Sun, W., Li, Y., and Li, J. (2012). Hybrid plane fitting for depth estimation. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1 –4.

Yang, J., Hou, E., and Zhou, M. (2013). Front sensor and gps-based lateral control of automated vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 14(1):146–154.

Yu, X., Beucher, S., and Bilodeau, M. (1992). Road tracking, lane segmentation and obstacle recognition by mathematical morphology. *IEEE Transactions on Intelligent Vehicles Symposium.*, 1:166–172.

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C., Kaus, E., Herrtwich, R., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knöppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., and Zeeb, E. (2014). Making bertha drive–an autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20.

Appendix A

Distance to collision calculation

This Appendix presents the Distance to Collision (d_{coll}) formulation proposed by (Arras et al., 2002) for polygonal robots, like car-like robots, moving in circular trajectories. Considering the robot body frame $\{\mathcal{R}\}$, the objective is to describe the trajectory that an obstacle point O performs in $\{\mathcal{R}\}$. Thus, the expected collision point P into the robot can be estimated by the intersection of the robot dimensions with the trajectory of O. To calculate this trajectory, some vectors and variables are defined as in the Figure A.1:

$$\begin{aligned}
\overrightarrow{P_C} &= \overrightarrow{P} - \overrightarrow{C}, \\
\overrightarrow{O_C} &= \overrightarrow{O} - \overrightarrow{C}, \\
\overrightarrow{R_C} &= \overrightarrow{R} - \overrightarrow{C}, \\
|r| &= \left|\overrightarrow{C}\right|, \\
\omega &= v_1 \cos(\phi)/r = \dot{\theta}, \\
v &= v_1 \cos\phi, \\
r &= v/\omega, \\
r_O &= \left|\overrightarrow{O_C}\right|.
\end{aligned} \tag{A.1}$$

All vectors are represented in the initial frame $(x, y, \theta, \phi) = (0, 0, 0, \phi)$ and depend on the velocity of the front wheel v_1, the steering angle ϕ, and the obstacles' data.

Accordingly to the Figure A.1, the obstacle point O describes a circular trajectory (dashed line) with the same center of the robot frame trajectory (continuous line). Its equation can be defined as:

$$r_O^2 = x_{coll}^2 + (y_{coll} - r)^2. \tag{A.2}$$

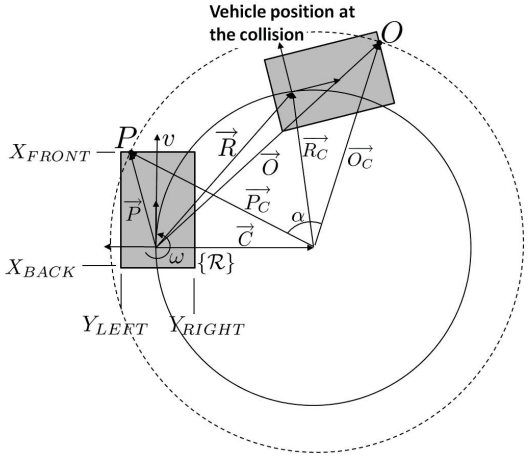


Figure A.1 – Circular trajectory of a point O (dashed line) described in the robot frame $\{\mathcal{R}\}$ (continuous line). If P is on the robot contours, the distance to collision is the arc length between $\overrightarrow{P_C}$ and $\overrightarrow{O_C}$.

where r_O is the circle's radius given by the length of the vector $\overrightarrow{O_C}$. For each side of the car, the collision point $P = (x_{coll}, y_{coll})$ is acquired by the following equation systems:

— Front side, with $y_{coll} \in [Y_{RIGHT}, Y_{LEFT}]$

$$\left.\begin{array}{l} x_{coll}^2 + (y_{coll} - r)^2 = r_O^2 \\ x_{coll} = X_{FRONT} \end{array}\right\} \Rightarrow \begin{array}{l} y_{coll} = r \pm \sqrt{r_O^2 - X_{FRONT}^2} \\ x_{coll} = X_{FRONT} \end{array} \tag{A.3}$$

— Left side, with $x_{coll} \in [X_{BACK}, X_{FRONT}]$

$$\left.\begin{array}{l} x_{coll}^2 + (y_{coll} - r)^2 = r_O^2 \\ y_{coll} = Y_{LEFT} \end{array}\right\} \Rightarrow \begin{array}{l} x_{coll} = \pm\sqrt{r_O^2 - (Y_{LEFT} - r)^2} \\ y_{coll} = Y_{LEFT} \end{array} \tag{A.4}$$

— Right side, with $x_{coll} \in [X_{BACK}, X_{FRONT}]$

$$\left.\begin{array}{l} x_{coll}^2 + (y_{coll} - r)^2 = r_O^2 \\ y_c = Y_{RIGHT} \end{array}\right\} \Rightarrow \begin{array}{l} x_{coll} = \pm\sqrt{r_O^2 - (Y_{RIGHT} - r)^2} \\ y_{coll} = Y_{RIGHT} \end{array} \tag{A.5}$$

— Back side, with $y_{coll} \in [Y_{RIGHT}, Y_{LEFT}]$

$$\left.\begin{array}{l} x_{coll}^2 + (y_{coll} - r)^2 = r_O^2 \\ x_{coll} = X_{BACK} \end{array}\right\} \Rightarrow \begin{array}{l} y_{coll} = r \pm \sqrt{r_O^2 - X_{BACK}^2} \\ x_{coll} = X_{BACK} \end{array} \tag{A.6}$$

Note that, only the real solutions represent a collision in the point P. The distance of each collision point can be calculated by the angle α formed between the vectors $\overrightarrow{P_C}$ and $\overrightarrow{O_C}$ as:

$$d = \alpha \cdot r. \tag{A.7}$$

Once the obstacle point O can collide in more than one point in the robot contour, the final distance to collision d_{coll} is the smallest distance between all possible ones:

$$d_{coll} = \min\left(d_{FRONT}, d_{LEFT}, d_{RIGHT}, d_{BACK}\right). \tag{A.8}$$