



A Hybrid Controller for Vision-Based Navigation of Autonomous Vehicles in Urban Environments

Danilo Alves de Lima, Alessandro Corrêa Victorino

► To cite this version:

Danilo Alves de Lima, Alessandro Corrêa Victorino. A Hybrid Controller for Vision-Based Navigation of Autonomous Vehicles in Urban Environments. IEEE Transactions on Intelligent Transportation Systems, 2016, 17 (8), pp.2310-2323. 10.1109/TITS.2016.2519329 . hal-01305733

HAL Id: hal-01305733

<https://hal.science/hal-01305733>

Submitted on 21 Apr 2016

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

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

A hybrid controller for vision-based navigation of autonomous vehicles in urban environments

Danilo Alves de Lima and Alessandro Corrêa Victorino

Abstract—This paper presents a new hybrid control approach for vision-based navigation applied to autonomous robotic automobiles in urban environments. It is composed by a Visual Servoing (VS) for road lane following (as deliberative control) and a Dynamic Window Approach (DWA) for obstacle avoidance (as reactive control). Typically, VS applications do not change the velocities to stop the robot in dangerous situations or avoid obstacles while performing the navigation task. However, in several urban conditions, these are elements that must be dealt with to guarantee the safe movement of the car. As a solution for this problem, in this study a line following VS controller will be used to perform road lane following tasks with obstacle avoidance, validating its control outputs in a new Image-Based Dynamic Window Approach (IDWA). The final solution combines the benefits of both controllers (VS+IDWA) for optimal lane following and fast obstacle avoidance, taking into account the car kinematics and some dynamics constraints. Experiments in a challenging scenario with both simulated and real experimental car show the viability of the proposed methodology.

Index Terms—Hybrid Controller, Visual Servoing, Dynamic Window Approach, Obstacle Avoidance.

I. INTRODUCTION

IN the last few decades, autonomous robotic automobiles have been increasingly in evidence, given the fact several gains for security, power consumption, efficiency, etc. are involved. 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 [1], that the potentiality of these vehicles were tested. Nowadays, there are vehicles capable of driving in different situations, for long distances and respecting traffic laws [2]. However, these vehicles use high cost sensors, some of them impractical for final commercial cars, limiting the target customers. In addition, they must deal with some problems caused by the environment where the car is inserted, for example, the localization problems common related to GPS signal losses, as described by many DARPA participants [1]. Hence, this leaves us several possibilities for new navigation approaches based on low cost sensors, better suited for the environment where the vehicle is inserted.

Sensor-based control is a useful strategy based on exteroceptive sensors data (such as 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, is rich of perceptible

information. Commercially, some manufacturers have already used exteroceptive sensors in their drive assistance systems (ADAS) for parking, lane keeping, collision alert, etc.

Focusing on local navigation strategies for road following, vision systems are a reliable low cost alternative which concentrate a large number of environment data in a single image [3]. The viability of implementing many computer vision algorithms in hardware also increases the processing speed and reduces the power consumption, common problems 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 [4]. However, the approaches based on VS do not directly change the velocities of the vehicle to perform the obstacle avoidance, which is essential in urban environments navigation.

In this context, we proposed a new reactive controller named as Image-based Dynamic Window Approach (IDWA) [5], integrating the VS control methodology for road lane following in the Dynamic Window Approach for obstacle avoidance [6]. However, following the desired road lane and avoiding obstacles are opposite tasks for the 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, in this paper we present a hybrid control solution (VS+IDWA), where the IDWA works as a validation method for the VS control outputs. The performances of the proposed IDWA and VS+IDWA are validated experimentally in simulation and in a real electric car. Initial results of our methodology were presented in [5] and [7]. This paper presents the complete methodology and a fully analysis of the implemented controller results.

This article is organized as follows: before introducing our hybrid control for visual navigation, a brief state of the art is presented in the Section II; Section III presents the statement of the problem, with some important definitions about the robot model and the image features used; Section IV describes the deliberative (VS) and reactive (IDWA) controllers and their combination in the hybrid solution VS+IDWA; a discussion about the controller convergence is presented in the Section V; the experimental analysis and validation of the method, using a simulated and a real autonomous vehicle, are detailed in Section VI; and some conclusions and perspectives for future works are given in the Section VII. Complementary information about the DWA methodology is given in Appendix A.

II. RELATED WORK

Vision systems play an important role in autonomous cars application, due to the richness and versatility of the information that they supply [3]. These applications include monovision and stereovision cameras. Focusing on the navigation

The authors are with Federal University of Lavras and Heudiasyc UMR CNRS 7253 University of Technology of Compiègne. During this study, Danilo Alves Lima was supported by a Picardie region (French) Ph.D. scholarship and a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brazil) Post-Doc scholarship. Contact the authors: {daniloalvesdelima}@yahoo.com.br and {alessandro.victorino}@utc.fr.

control area, VS is a robust way to deal with the vehicle guidance in urban environments, since it avoids car localization problems, such as those from [1]. VS can be divided in two main approaches: the Position-Based Visual Servoing (PBVS) and the Image-Based Visual Servoing (IBVS) [4]. 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. On the other hand, in the IBVS 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 [8].

Beyond these techniques, several control laws can be defined to allow a vehicle to converge and follow perceived features and primitives in the image, for example, points, lines, and ellipses [9]. These are called *model-based* approaches, which require some geometrical knowledge of the environment. These controllers normally combine a features tracking with the robot forward movement by means of task functions. These approaches can also be adapted for pose stabilization oriented by landmarks [10]–[12] and path reach and following [13]. 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 [14], [15]. It is important to mention that there are many others VS approaches for nonholonomic robots, such as: the hybrids approaches that combine the both IBVS and PBVS information in a 2 1/2D visual servoing [16], and the methods where no previous knowledge about the scene are required [17], [18].

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) [15], [19], [20]. These tasks are often merged in the control level by some switching strategy which changes the task weight in the presence of obstacles, which consequently changes the control law. Moving obstacles may also be considered in this kind of approach, as presented in [21].

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 [22] was presented a method for car-like robot navigation based on the validation of a Velocity Vector Field in a Dynamic Window Approach (DWA) [6]. 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, such as the tentacles [15], [23], 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, whereas the DWA calculates the reachable velocities

around the current ones, to choose among then the best one related to some conditions (which may be similar to those from the tentacles).

In this study, 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 feature set 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 IDWA¹ [5]. This combination results in a hybrid controller (VS+IDWA) regardless the vehicle localization, diverging from the methodology presented in [22], which required a global path planning and a localization system. This work also diverges from the previous ones [15], [19], [20] based on VS, because 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. Likewise [21], the usage of the DWA also enables us to deal with moving obstacles, as described in [24], which will be left for future research.

III. STATEMENT OF THE PROBLEM

Focusing on the vision-based navigation of autonomous vehicles, the proposed methodology is mainly situated in two robotic research areas: Perception of the Environment and Local Navigation Control. Perception of the Environment performs features extraction for the VS control and obstacles detection for reactive obstacle avoidance. In this study, the features and obstacles are considered to be correctly detected for all situations, since the perception is not our main objective. However, the features used are described in the Subsection III-B. For the local navigation control, the robot model used and some constraints are described in the Subsection III-A as follows.

A. Robot definitions

The vehicle control is performed by the input set defined as $u = [v_1 \ v_2]^T$, where v_1 is the linear velocity of the front-wheels and v_2 is the steering velocity. The robot movement is related to its body frame $\{\mathcal{R}\}$ in relation to an inertial frame $\{\mathcal{O}\}$, and can be recovered by $\dot{q} = [\dot{x}_r \ \dot{y}_r \ \dot{\theta}]^T$. These elements are related by the kinematic model of a front wheel drive car, with the Ackerman's approximation for the steering angle ϕ , given by:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \sin \phi / l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2. \quad (1)$$

¹Once 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.

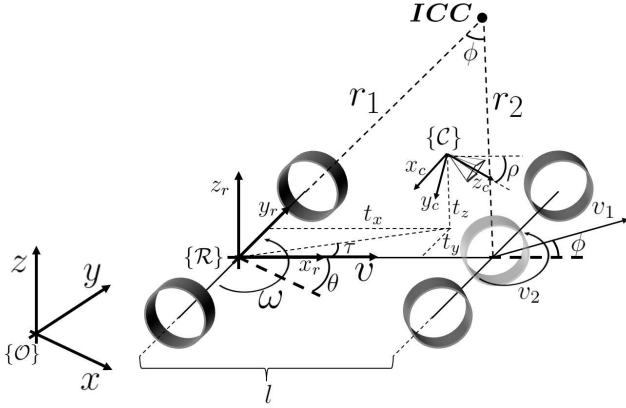


Fig. 1. Kinematic model diagram of a front-wheel drive car centered in the body frame $\{\mathcal{R}\}$. The pinhole camera frame is represented in $\{\mathcal{C}\}$.

The vehicle orientation and steering angle (θ and ϕ) are positive counter-clockwise, with $\theta \in [-\pi, \pi]$ and $\phi \in [-\phi_{max}, \phi_{max}]$ (see the Fig. 1 for a complete illustration of these variables). The origin of $\{\mathcal{R}\}$ is located at the midpoint of the two rear wheels, which performs circular trajectories defined by the instantaneous center of curvature (ICC). Note that the robot linear velocity v is related to the front wheel velocity by $v = v_1 \cos(\phi)$, and the angular velocity $\dot{\theta} = v_1 \cos(\phi)/r_1 = \omega$ is directly related to the steering angle (see the Fig. 1). These considerations allow us to choose the robot control input as $u_r = [v \ \omega]^T$.

This model is an approximation used only for fast validation of our navigation method, because it is valid only for low speed applications. However, this low speed consideration was a security constraint due to the extension of the test track used during the experimental results (Section VI) and its sharp turns. For high speed applications, a dynamic model must be considered [23]. Moreover, 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 studies.

B. 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. In order to achieve that, the robot is considered to move in a planar surface and also includes a fixed pinhole camera, represented by the reference frame $\{\mathcal{C}\}$ at the Fig. 1. The optical center position is given by $(x_c, y_c, z_c) = (t_x, t_y, t_z)$ in the robot frame, 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 Fig. 2, where (u_{max}, v_{max}) and $(2X_I, 2Y_I)$ 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, the image features are the line $(X, Y, \text{ and } \Theta)$ formed by the tangent Γ of the path P at the point

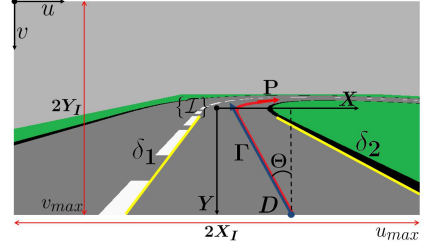


Fig. 2. Image frame $\{\mathcal{I}\}$ representation for the camera frame $\{\mathcal{C}\}$. The road lane center P (in red) is related to the boundaries δ_1 and δ_2 (in yellow). Its tangent Γ (in blue), at the point D and angle offset Θ from Γ to the axis $-Y$, defines the image features X, Y , and Θ .

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

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

$$\begin{cases} X = \frac{u}{u_{max}-1} - c_x = \frac{x_c f_x}{z_c} \\ Y = \frac{v}{v_{max}-1} - c_y = \frac{y_c f_y}{z_c} \end{cases}, \quad (2)$$

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

$$\begin{cases} x_c = \frac{X t_z}{\sin \rho + Y \cos \rho} \\ y_c = \frac{Y t_z}{\sin \rho + Y \cos \rho} \\ z_c = \frac{t_z}{\sin \rho + Y \cos \rho} \end{cases}. \quad (3)$$

There are no singularities in these equations caused by a zero on the denominator, because 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.

IV. CONTROL DESIGN

In this paper we propose a new vision-based navigation approach combining visual servoing and obstacle avoidance in a hybrid control. The focus is on the optimality of deliberation for road lane following and real-time reaction to environment changes. The deliberative control will be performed by an IBVS approach, adapted from [25] 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. The goal 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. 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 camera calibration, as described in [8]. 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 section describes both controllers, compounding the hybrid controller called by VS+IDWA.

A. Deliberative control: VS

This subsection recalls the formulation used by Cherubini *et al.* [25] for the IBVS approach of following a line path projected on the image frame. Considering the image frame depicted in the Fig. 2, the features set $s = [X \ Y \ \Theta]^T$ defined by the tangent Γ , and the camera installed on the sagittal plane of the car ($t_y = 0$), 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 limit 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 Fig. 3. 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, as seen in the Subsection III-A. 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

$$\dot{s} = L_s u_c. \quad (4)$$

This expression uses the interaction matrix L_s for the normalized perspective camera model (2), derived from [9] as:

$$L_s(X, Y, \Theta) = \begin{bmatrix} \frac{1}{z_c} & 0 & \frac{X}{z_c} & XY & -1-X^2 & Y \\ 0 & \frac{-1}{z_c} & \frac{Y}{z_c} & 1+Y^2 & -XY & -X \\ \frac{C\rho C^2\Theta}{t_z} & \frac{C\rho C\Theta S\Theta}{t_z} & -\frac{\zeta C\rho C\Theta}{t_z} & -\zeta C\Theta & -\zeta S\Theta & -1 \end{bmatrix}, \quad (5)$$

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 (3).

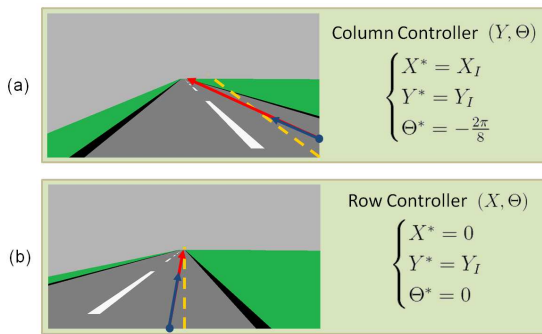


Fig. 3. Feature configuration to apply the column (a) and row (b) controllers and their corresponding setpoints in dashed yellow.

Thus, u_r may be expressed in the camera frame $\{C\}$ by (6) using the homogeneous transformation (7):

$$u_c = {}^C T_R u_r, \quad (6)$$

$${}^C T_R = \begin{bmatrix} 0 & -t_x \cos \tau \\ -\sin \rho & t_y \cos \rho \\ \cos \rho & -t_y \sin \rho \\ 0 & 0 \\ 0 & -\cos \tau \cos \rho \\ 0 & -\cos \tau \sin \rho \end{bmatrix} \quad (7)$$

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 $Y = \text{const} = Y^*$. This led us to the system state equations, obtained from combining (4), (6), and (7):

$$[\dot{X} \ \dot{\Theta}]^T = A_r v + B_r \omega, \quad (8)$$

with $A_r = \begin{bmatrix} L_X \\ L_\Theta \end{bmatrix} T_v$ and $B_r = \begin{bmatrix} L_X \\ L_\Theta \end{bmatrix} T_\omega$. When $B_r \neq 0$, the control law is:

$$\omega = -B_r^+ (\lambda e + A_r v), \quad (9)$$

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 $X = \text{const} = X^*$. It is analogously defined changing the row controller definitions from X to Y . The controllers are selected regarding the features' coordinates in the image frame (see Fig. 3). 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 [25] 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 position-based approach (PBVS) [13]. For more details about the implementation and stability analysis see [13], [25]. Some studies could be performed in order to change the control law and improve the controller convergence. Instead of that, here we propose 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.

B. Reactive Control: IDWA

The Dynamic Window Approach is a reactive obstacle avoidance technique proposed originally by Fox *et al.* [6]. It selects, between all available velocities in a Dynamic Window search space (V_{DW}), the best control input by optimizing the objective function (10):

$$DWA(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v). \quad (10)$$

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 [26], [27] and different robot types, like car-like robots [28], as well as to dynamic environments [24]. However, the robot and goal relative position to the world were known in these studies, 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 (10), as explained below.

The IDWA functions: The *heading*(v, ω) function is responsible to guide the vehicle to a desired goal configuration. In the original DWA formulations [6], *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 IV-A) the robot localization is not required, and the goal is to guide the errors of the image features to zero. Extending to the DWA, the *heading* function must estimate the error

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

in the next frame $\mathcal{I}_{t+\Delta t}$, considering (X^*, Y^*, Θ^*) as setpoint. This is illustrated in the Fig. 4. Thus, high weights are given to the sets of control inputs $(v_i, \omega_j) \in V_{DW}$ which reduce the final error $e_{t+\Delta t}$.

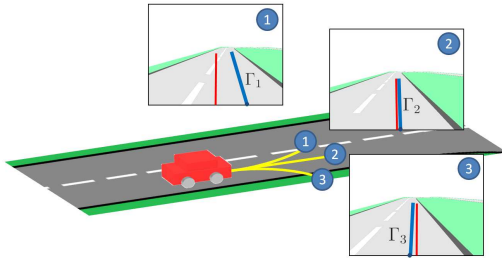


Fig. 4. Estimation of the image features set Γ_i (blue line) in the frame $\mathcal{I}_{t+\Delta t}$ applying the control inputs (v_1, ω_1) , (v_2, ω_2) and (v_3, ω_3) . The reference position is also represented in red, which means the vehicle in the center of the road lane.

Recalling (4) and (6), 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, by integrating each set of velocities over the time, we estimate the features configuration in the frame $\mathcal{I}_{t+\Delta t}$. As the row/column controllers are applied independently, the function *heading*(v, ω) 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:

$$XY_{error}(v, \omega) = \begin{cases} 1 - \frac{|e_X(t+\Delta t)|}{e_{Xmax}}, & \text{if row controller,} \\ 1 - \frac{|e_Y(t+\Delta t)|}{e_{Ymax}}, & \text{otherwise.} \end{cases} \quad (11)$$

$$\Theta_{error}(v, \omega) = 1 - \frac{|e_\Theta(t+\Delta t)|}{\pi}. \quad (12)$$

The errors in the image frame $\mathcal{I}_{t+\Delta t}$ 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, ω) as the sum of these previous functions:

$$heading(v, \omega) = \alpha_1 XY_{error}(v, \omega) + \alpha_2 \Theta_{error}(v, \omega). \quad (13)$$

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.* [29] 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. 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. This is the same consideration used in [6], [23], 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 IV-A). It is constant and respects the speed limit of the road. 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:

$$velocity(v) = \begin{cases} \frac{v}{(v_d - v_{min})} & \text{if } v \leq v_d, \\ \frac{(v_{max} - v)}{(v_{max} - v_d)} & \text{if } v > v_d, \end{cases} \quad (14)$$

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 Δt as:

$$V_d = \{(v, \omega) \mid v \in [v_a - \dot{v}\Delta t, v_a + \dot{v}\Delta t], \omega \in [\omega_a - \dot{\omega}\Delta t, \omega_a + \dot{\omega}\Delta t]\}, \quad (15)$$

with the robot input set $u_r = [v \ \omega]^T$ (see Subsection III-A) 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, ω), and the robot maximum decelerations $(\dot{v}_b, \dot{\omega}_b)$. By (16), 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.

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot d_{max} \cdot \dot{v}_b}, \\ \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot d_{max} \cdot \dot{\omega}_b}\}. \quad (16)$$

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 (17).

$$V_{DW} = V_d \cap V_a \cap V_s, \quad (17)$$

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 (10). The discretization considers the smallest $\Delta\phi$ and Δv which significantly contributes to the vehicle navigation. In addition, the stabilization time required for the vehicle actuators are observed when defining the final Δt . An example of V_{DW} is shown in the Fig. 5, where w_a was converted in ϕ_a for visualization purposes.

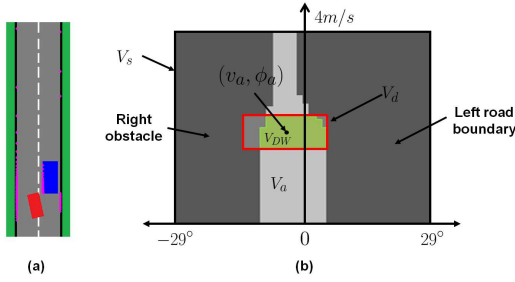


Fig. 5. Example of a Dynamic Window V_{DW} (b) calculated for a certain vehicle state in (a). The vehicle is illustrated by the red rectangle, where the obstacle (blue) and road boundaries are detected by a laser (pink dots) projected in an occupancy grid [30]. In (b), v_a and ϕ_a are the current linear velocity and steering angle of the vehicle, resulting the V_{DW} in green.

C. Hybrid Control: VS+IDWA

The VS methodology presented in the Subsection IV-A does not guarantee safety to the car movement, for 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 study. In this controller, the VS output is validated before being applied in the vehicle.

The equation (17) defines the IDWA search space V_{DW} , with all reachable velocities available in a time interval Δt . 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 (10). 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:

$$(v_{VS}, \omega_{VS}) \in V_{DW} \text{ and } \text{dist}(v_{VS}, \omega_{VS}) > \frac{d_{VS}}{d_{max}}. \quad (18)$$

V. CONVERGENCE ANALYSIS

To analyze the convergence of the current approach, it is necessary to define the expected objectives of the local navigation approach. Since 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 available options.

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

The first main task (1) is guaranteed by the deliberative control (VS), for its stabilization is proved by Cherubini *et al.* [25]. 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 [26], 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 available options, since 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.

VI. EXPERIMENTAL RESULTS

The proposed controller was implemented in both simulation and real car-like robot. The experiments setup and results are shown next.

A. Experiments setup

To validate the navigation methodology proposed, both simulation and real environment were adjusted with the same parameters. They use the kinematic model (1) to represent the vehicle movement and the same kinematics constraints, and some actuators dynamics from our full actuated electric car APACHE of the Fig. 6. For the perception of the environment, the vehicle uses a monocular camera with a focal length of 1.8 mm and large field of view ($\simeq 140^\circ$) to detect the road lane center features, as described in Subsection III-B. The camera is on the robot sagittal plane in a rigid structure with $(t_x, t_y, t_z) = (1.54, 0.0, 1.62)$ m and tilt offset $\rho \simeq 9.5^\circ$. It also detects obstacles with a laser sensor with 180° of coverage

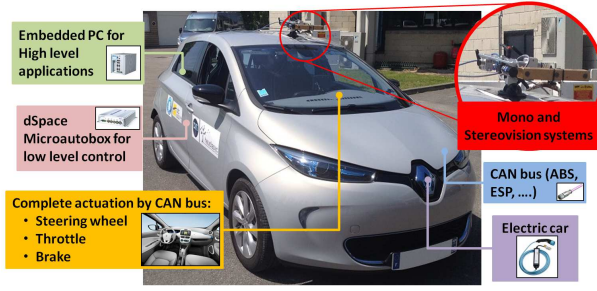


Fig. 6. Fully actuated electric car APACHE with some available resources.

(for 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 [30]. 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.

B. Simulation experiments

Simulation is an important resource that aggregates several possibilities of experiments, which could be hard to execute in a real environment. It also allows an easy visualization of the VS control parameters influence and the IDWA validation during the robot navigation. The robot speed was limited to 3m/s due to the kinematic model approximations. The reference was always set to the center of the right lane, similar to the Fig. 2. With this setup, we checked the influence of the VS and IDWA gains in the final VS+IDWA result, and then we performed a quantitative comparison between IDWA and the VS+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 not a priori path planning.

Influence of the VS gain: The Fig. 7 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 obstacles on the way. In this figure it is possible to see the errors decreasing when increasing λ , which is evidenced by the Table I 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.

Moreover, this figure shows that the VS approach is not robust for path reaching, because 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

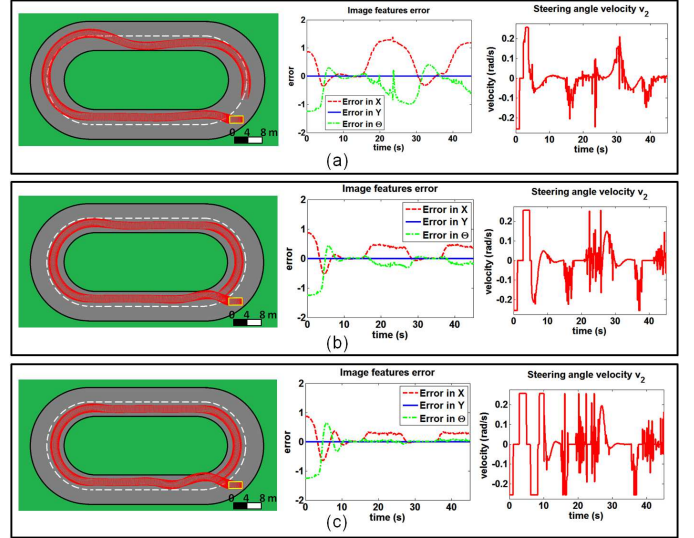


Fig. 7. Simulation result for the VS controller performing the road lane following with $\lambda = 0.3$ (a), $\lambda = 0.5$ (b), and $\lambda = 0.7$ (c). The left column illustrates the car movement in this environment, where in yellow is the initial pose and in red the instantaneous positions for a clockwise movement. In the column of the middle is the error evolution of the image features and in the column of the right is the steering wheel velocity output (v_2) during the experiments.

TABLE I
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

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 in [25], 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 Fig. 7(a). 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 Fig. 1).

Instead of dynamic adjusting the gain, it is possible to reduce the overshoot by validating the VS control outputs in the IDWA. In order to do that, we considered the final $\lambda = 0.5$, to take advantage of the path following capabilities of the VS controller (Fig. 7(b)). It is important to mention that drifting away the tangent estimation point D (see the Fig. 2) can also be used to anticipate the vehicle reaction to different path curvatures and reduce the problem observed in the Fig. 7(a).

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, because it is a user's choice to define what the best reaction to the robot is (e.g. overtaking or just stopping near 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 Fig. 8(a) 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 an obstacle or in a situation where the error cannot be reduced, as shown in this figure. The function *dist* (Fig. 8(b)), on the other hand, 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 Fig. 8(c). All functions together with no adjustment result in the Fig. 8(d). With this configuration, the robot gives priority to reduce the error in the image features until stop near to an obstacle.

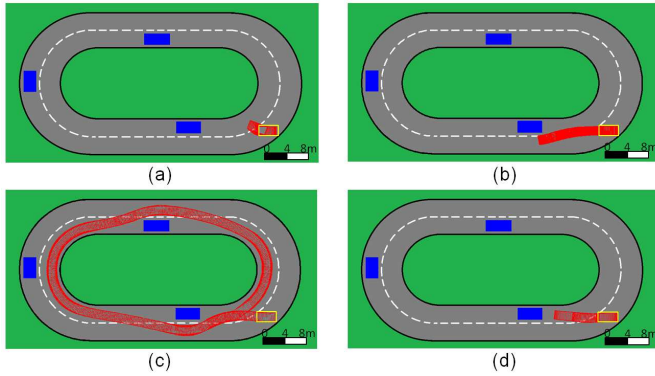


Fig. 8. Influence of the gains associated with the IDWA when performing a reactive obstacle avoidance. Initially, only the functions XY_{error} and Θ_{error} were enabled in (a), *dist* in (b), and *velocity* (c). Finally, all the gains were enabled in (d), with $\alpha_1 = \alpha_2 = \beta = \gamma = 1.0$. 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 Fig. 8, 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 close to v_d , and avoiding the collision with the obstacle situated on the lane. Since the main objective for the IDWA is the smooth obstacle overtaking with high velocities, in the Fig. 9 the previous gains were adjusted one by one regarding this final objective.

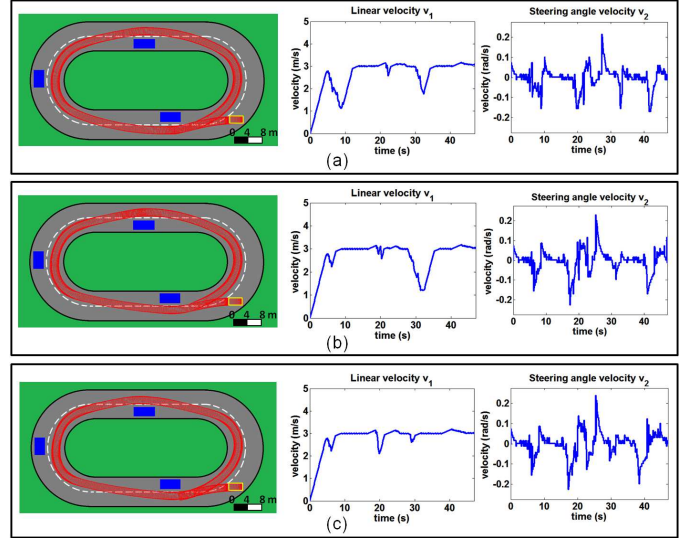


Fig. 9. Tuning process for the gains associated with the IDWA, where: (a) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 1.0$, and $\gamma = 1.0$; (b) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 1.0$, and $\gamma = 3.0$; and (c) $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, and $\gamma = 3.0$. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

Initially, we reduced the XY_{error} and Θ_{error} gains $\alpha_1 = \alpha_2 = 0.1$ (Fig. 9(a)) to avoid the native opposition between *heading* and the other functions of the IDWA (Section V). Although avoiding obstacles, this configuration reduced considerably the car linear velocity (v_1) during the maneuvers (after 10 seconds and 30 seconds of simulation). In this context, we increased the *velocity* gain $\gamma = 3.0$ in the Fig. 9(b). However, the distance to the obstacles were very close during the overtaking, forcing the velocity to be reduced around 30 seconds of simulation. Thus, the distance to the obstacles was enhanced by the gain $\beta = 2.0$, resulting in the Fig. 9(c). The final configuration was set to $\alpha_1 = \alpha_2 = 0.1$, $\beta = 2.0$, 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 Fig. 10 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 Fig. 10). This difference can also be observed in several points of the Fig. 11, 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.

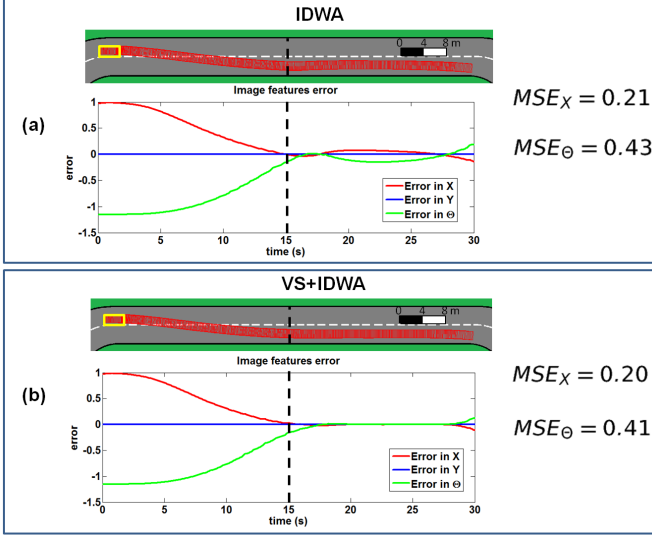


Fig. 10. Comparison between the vehicle lane convergence and following using the IDWA (a) and the VS+IDWA (b) controllers, with the error of the image features and the MSE. The car initial pose is represented in yellow and the car instantaneous positions are in red for a left to right movement.

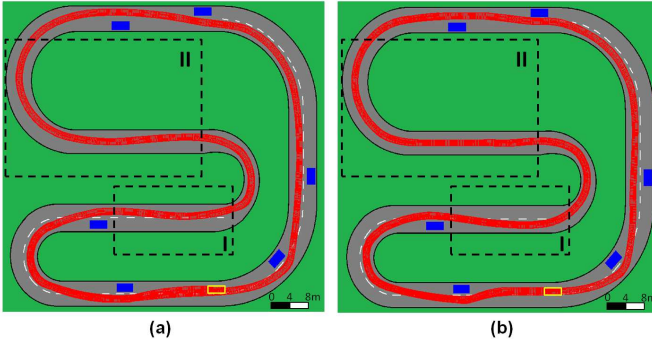


Fig. 11. Comparative movement using only the reactive controller IDWA (a) and the complete solution VS+IDWA (b). The difference is mainly observed for the road lane following task in the dashed box I and II. The car initial pose is represented in yellow, the obstacles are in blue, and the car instantaneous positions are in red for a clockwise movement.

C. Real car-like robot experiments

Three experiments (Exp. I-III) were performed with the real car-like robot APACHE (Fig. 6) 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 Fig. 12. In all cases, the vehicle must 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 images. These experiments and the APACHE system are explained as follow.

The experimental car-like robot: The experimental car is a full actuated Renault Zoe, hereby appointed APACHE and presented in the Fig. 6. This figure illustrates some available resources in this vehicle, like CAN bus access for low level control (steering, acceleration, and brake) and access to proprioceptive information, exteroceptive sensors for perception



Fig. 12. Robot course (red line) for the local navigation experiments at the SEVILLE test track. All experiments start in the arrow.

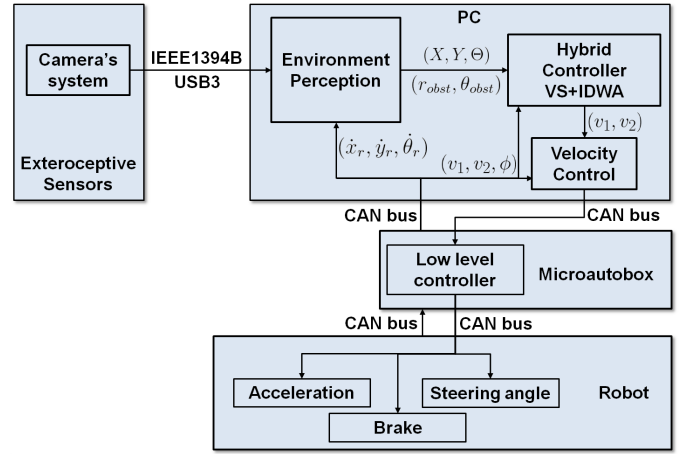


Fig. 13. System diagram for the experimental vehicle APACHE.

of the environment, and computers for data processing and control. In addition, this is an electric car which aggregates some benefits in the linear speed control, since it does not requires a gearbox.

With these available resources, the solution was structured in the diagram of Fig. 13. The box PC represents an embedded computer which implements the environment perception [31], with the setup presented in the Subsection VI-A, the hybrid controller VS+IDWA, and a controller for the linear and steering velocities. The entire system runs in approximately 10 Hz with an Intel Core I7-3610QE CPU (2.30 GHz).

Exp. 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 Fig. 14(a). The desired linear velocity v_d was set to 3.61 m/s (or 13 km/h). The Fig. 14(b)-(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 Fig. 14(a)). Note that, even with large variations in the features set (first 7 seconds of Fig. 14(b)-(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 Fig. 14(d)-(e). In these figures, the VS curve represents only the deliberative control calculations, on the other hand the VS+IDWA curve is the final result after

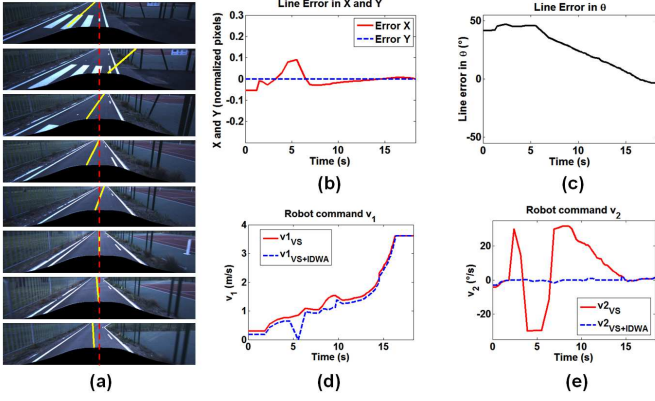


Fig. 14. Road center tracking experiment (I) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment, (b) and (c) show the evolution of the X , Y , and θ errors. The vehicle control inputs calculated by VS and the VS+IDWA are in (d) and (e).

validation in equation (18). 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, the errors in the features set result in large control outputs for the VS, invalidating these outputs. Thus, the IDWA reacts changing the v_1 and v_2 values in order to slowly converge the car to the road center. After 15 seconds of experiment, the VS control is valid and guarantees a better error reduction than the IDWA approach, as proved in the simulation results of Figs. 10 and 11. Similarly to a human driver behavior, the vehicle control inputs were performed smoothly during the experiment, even in the presence of features variations like in the second image top-down of Fig. 14(a), validating the gains tuned in simulation.

Exp. 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 Fig. 15(a) 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).

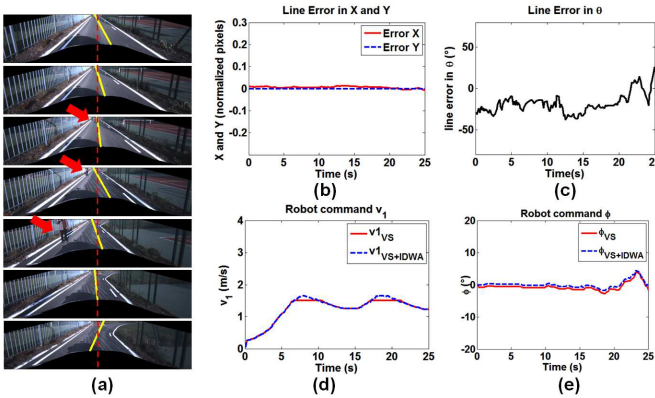


Fig. 15. Road center following with obstacle avoidance experiment (II) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment with the obstacle indicated by the red arrow, (b) and (c) show the evolution of the X , Y , and θ errors. The vehicle control inputs calculated by the VS and VS+IDWA are in (d) and (e).

In the Fig. 15(b)-(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 always provided smooth vehicle control inputs, improving the comfort sensation for the passengers (Fig. 15(d)-(e)).

1) Exp. 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 Fig. 16(a), the car follows the road lane center until stop just in front of the person.

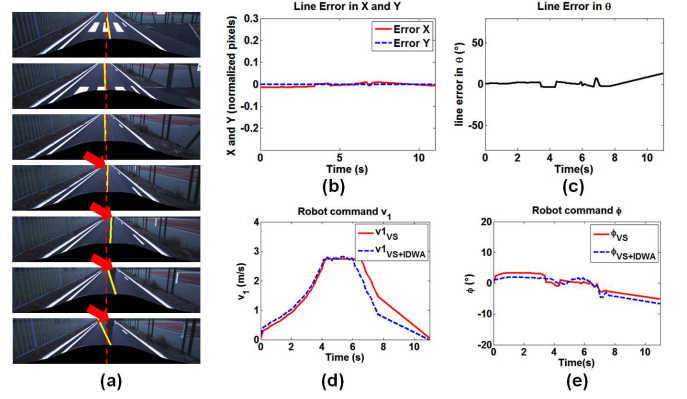


Fig. 16. Road center following with collision avoidance experiment (III) applying the VS+IDWA, where (a) presents some detected image features in sequence during the experiment with the obstacle indicated by the red arrow blocking the road, (b) and (c) show the evolution of the X , Y , and θ errors. The vehicle control inputs calculated by the VS and VS+IDWA are in (d) and (e).

Once the road center is obstructed, the image features detection vary considerably during the vehicle movement (Fig. 16(b)-(c)), but the VS+IDWA always avoided hard movements which could lead the vehicle out of the road. We can note the acceleration and deceleration applied in the linear velocity control, taking almost the same time to go from 0 to 2.7 m/s and vice-versa (Fig. 16(d)-(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 [32].

VII. CONCLUSIONS AND FUTURE WORKS

This study presented a new hybrid controller for vision-based local navigation of car-like robots in urban environment. By combining a Visual Servoing (VS) approach, as deliberative control, and the Image-based Dynamic Window Approach (IDWA), as reactive control, it allowed the road lane following with obstacle avoidance by our experimental vehicle. By using only the VS controller it is not possible to guarantee the

accomplishment of the local navigation task, caused by some limitations such as: the path reaching problems, the constant linear velocity, and the obstacles in the environment. However, validating the VS outputs in the IDWA assured the safety of the VS approach. In addition, if the VS outputs are not allowed, the IDWA optimization function gives 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 VS techniques could be integrated with the present solution to allow, e.g., vehicle platooning and navigation in different scenarios. Due to the nature of this methodology, it can be applied with low-cost sensors and independent of high precision localization system. Based on these benefits, future studies will integrate this controller in a global navigation methodology, as the one proposed in [33], using low-cost sensors and our experimental vehicle. To do so, more accurate experiments must be performed, considering a dynamic model to increase the vehicle speed and some improvements in the environment perception layer to allow dynamic environments and reduce the features and obstacles detection variations.

APPENDIX

APPENDIX A. DISTANCE TO COLLISION CALCULATION

This Appendix presents the Distance to Collision (d_{coll}) formulation proposed by [29] for polygonal robots, such as 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 (19) and variables (20) are defined as in the Fig. 17.

$$\begin{aligned}\vec{P_C} &= \vec{P} - \vec{C}, \\ \vec{O_C} &= \vec{O} - \vec{C}, \\ \vec{R_C} &= \vec{R} - \vec{C},\end{aligned}\quad (19)$$

$$\begin{aligned}|r| &= |\vec{C}|, \\ \omega &= v_1 \cos(\phi)/r = \dot{\theta}, \\ v &= v_1 \cos \phi, \\ r &= v/\omega, \\ r_O &= |\vec{O_C}|.\end{aligned}\quad (20)$$

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.

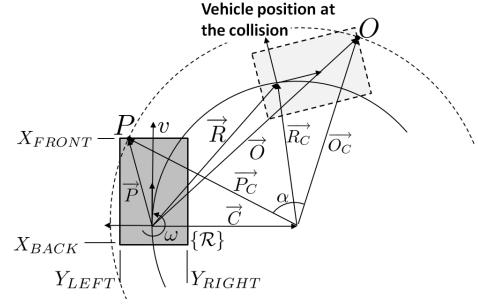


Fig. 17. Circular trajectory of a point O (dashed line) described in the robot frame $\{\mathcal{R}\}$ (continuous line). If P is on the robot contours, the distance to collision is the arc length between $\vec{P_C}$ and $\vec{O_C}$.

Accordingly to the Fig. 17, the obstacle point O describes a circular trajectory (dashed line) with the same center of the robot frame trajectory (continuous line), defined as:

$$r_O^2 = x_{coll}^2 + (y_{coll} - r)^2. \quad (21)$$

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

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

$$\left. \begin{aligned} x_{coll}^2 + (y_{coll} - r)^2 &= r_O^2 \\ x_{coll} &= X_{FRONT} \end{aligned} \right\} \Rightarrow \begin{aligned} y_{coll} &= r \pm \sqrt{r_O^2 - X_{FRONT}^2} \\ x_{coll} &= X_{FRONT} \end{aligned} \quad (22)$$

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

$$\left. \begin{aligned} x_{coll}^2 + (y_{coll} - r)^2 &= r_O^2 \\ y_{coll} &= Y_{LEFT} \end{aligned} \right\} \Rightarrow \begin{aligned} x_{coll} &= \pm \sqrt{r_O^2 - (Y_{LEFT} - r)^2} \\ y_{coll} &= Y_{LEFT} \end{aligned} \quad (23)$$

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

$$\left. \begin{aligned} x_{coll}^2 + (y_{coll} - r)^2 &= r_O^2 \\ y_{coll} &= Y_{RIGHT} \end{aligned} \right\} \Rightarrow \begin{aligned} x_{coll} &= \pm \sqrt{r_O^2 - (Y_{RIGHT} - r)^2} \\ y_{coll} &= Y_{RIGHT} \end{aligned} \quad (24)$$

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

$$\left. \begin{aligned} x_{coll}^2 + (y_{coll} - r)^2 &= r_O^2 \\ x_{coll} &= X_{BACK} \end{aligned} \right\} \Rightarrow \begin{aligned} y_{coll} &= r \pm \sqrt{r_O^2 - X_{BACK}^2} \\ x_{coll} &= X_{BACK} \end{aligned} \quad (25)$$

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 $\vec{P_C}$ and $\vec{O_C}$ as:

$$d = \alpha \cdot r. \quad (26)$$

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:

$$d_{coll} = \min(d_{FRONT}, d_{LEFT}, d_{RIGHT}, d_{BACK}). \quad (27)$$

ACKNOWLEDGEMENT

This study 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 the European Fund of Regional Development FEDER. The authors wish to thank the helpful assistance of G rald Dherbomez, Giovanni B. Vitor, Pierre Hudelaine, and Thierry Monglon during the experiments.

REFERENCES

- [1] M. Buehler, K. Lagnemma, and S. Singh, "Special issue on the 2007 DARPA Urban Challenge, part I-III," in *J. Field Robot.*, M. Buehler, K. Lagnemma, and S. Singh, Eds., 2008, vol. 25, no. 8–10, pp. 423–860.
- [2] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knöppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.
- [3] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *J. Intell. Robot. Syst.*, vol. 53, no. 3, pp. 263–296, 2008.
- [4] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [5] D. A. Lima and A. C. Victorino, "An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments," in *ICRA Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots*, Juin 2014, pp. 120–125.
- [6] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.
- [7] D. Lima and A. Victorino, "A visual servoing approach for road lane following with obstacle avoidance," in *Proc. 17th IEEE Int. Conf. ITSC*, Oct 2014, pp. 412–417.
- [8] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*, ser. Lecture Notes in Control and Information Sciences, D. Kriegman, G. Hager, and A. Morse, Eds. Springer London, 1998, vol. 237, pp. 66–78.
- [9] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, 1992.
- [10] F. Jurie, P. Rives, J. Gallice, and J. Brame, "High-speed vehicle guidance based on vision," *Control Eng. Pract.*, vol. 2, no. 2, pp. 289–297, 1994.
- [11] S. Lee, K. Boo, D. Shin, and S. Lee, "Automatic lane following with a single camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 1998, pp. 1689–1694 vol.2.
- [12] K. Usher, P. Ridley, and P. Corke, "Visual servoing of a car-like vehicle - an application of omnidirectional vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Sept 2003, pp. 4288–4293 vol.3.
- [13] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, pp. 1037–1048, 12 2011.
- [14] J. Courbon, Y. Mezouar, and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 392–402, Sept 2009.
- [15] A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *Int. J. Robot. Res.*, vol. 32, no. 2, pp. 189–205, 2013.
- [16] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, 1999.
- [17] P. Rives, "Visual servoing based on epipolar geometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 1, 2000, pp. 602–607 vol.1.
- [18] G. Silveira, E. Malis, and P. Rives, "Visual servoing over unknown, unstructured, large-scale scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 4142–4147.
- [19] V. Cadenat, R. Swain, P. Soueres, and M. Devy, "A controller to perform a visually guided tracking task in a cluttered environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, 1999, pp. 775–780 vol.2.
- [20] D. Folio and V. Cadenat, "A redundancy-based scheme to perform safe vision-based tasks amidst obstacles," in *Proc. IEEE Int. Conf. Robot. Bio.*, 2006, pp. 13–18.
- [21] A. Cherubini, F. Spindler, and F. Chaumette, "Autonomous visual navigation and laser-based moving obstacle avoidance," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2101–2110, Oct 2014.
- [22] D. A. Lima and G. A. S. Pereira, "Navigation of an Autonomous Car Using Vector Fields and the Dynamic Window Approach," *J. Control Autom. Electr. Syst.*, vol. 24, no. 1–2, pp. 106–116, 2013.
- [23] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, Sep. 2008.
- [24] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1986–1991.
- [25] A. Cherubini, F. Chaumette, and G. Oriolo, "An image-based visual servoing scheme for following paths with nonholonomic mobile robots," in *Proc. IEEE Int. Conf. Control, Autom., Robot., Vision*, Dec 2008, pp. 108–113.
- [26] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, pp. 341–346.
- [27] P. Saranrittichai, N. Niparnan, and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on dynamic window approach," in *Int. Conf. Electr. Eng./Electron., Comp., Telecommun. Info. Tech.*, May 2013, pp. 1–4.
- [28] K. Rebai, O. Azouaoui, M. Benmami, and A. Larabi, "Car-like robot navigation at high speed," in *Proc. IEEE Int. Conf. Robot. Bio.*, 2007, pp. 2053–2057.
- [29] K. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-time obstacle avoidance for polygonal robots with a reduced dynamic window," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2002, pp. 3050–3055.
- [30] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [31] D. Lima and A. Victorino, "VERVE and ROBOTEX – 2D/3D image processing for road and obstacles detection," https://youtu.be/rS4ejXJC2_w?list=PLqKAHZQ-ozS-kPvkcUoyxnV7xTBi0w5Z, 2015.
- [32] —, "VERVE and ROBOTEX – VS+IDWA for local navigation," <http://youtu.be/FZxUAU4-jCcA?list=PLqKAHZQ-ozS-kPvkcUoyxnV7xTBi0w5Z>, 2014.
- [33] E. C. M. Pereira, D. A. Lima, and A. C. Victorino, "Autonomous vehicle global navigation approach associating sensor based control and digital maps," in *Proc. IEEE Int. Conf. Robot. Bio.*, Dec 2014, pp. 2404–2409.



Danilo Alves de Lima received the B.S. degree in control and automation engineering and M.S. degree in electrical engineering from Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2008 and 2010, and a Ph.D. degree in information and systems technologies from the University of Technology of Compiègne (UTC), Compiègne, France, in 2015. Currently, he is a post-Ph.D. researcher at the Federal University of Lavras (UFLA), Lavras, Brazil.

He is a member of the Terrestrial Mobility Laboratory (LMT), UFLA, since 2015. He also worked with the Group for Research and Development of Autonomous Vehicles (PDVA), UFMG, and Heudiasyc UMR 7253, a common research laboratory between UTC and CNRS. His research interests include robotic navigation, intelligent vehicles development, and computer vision.



Alessandro Corrêa Victorino received the B.S. degree in Mechanical Engineering from Federal University of Espírito Santo (UFES) in 1996, the M.S. in Mechanical Engineering from State University of Campinas (UNICAMP) in 1998, and a Ph.D. in Robotics Automation and Control from National Institute for Research in Computer Science and Control (INRIA) in 2002. He defended a "Habilitation à Diriger des Recherches" (HDR) at University of Technology of Compiègne (UTC), Compiègne, France, in 2012.

Since 2006, he is an Associate Professor in the Computer Engineering Department at the UTC and member of the Heudiasyc laboratory UMR CNRS 7253. His research interests are in the field of robotic and mechatronic systems, more precisely in the area of intelligent and robotic vehicles, the development of autonomous navigation and driver assistance systems, integrating advanced control techniques, fusion of sensor data, observers of dynamic states, and exteroceptive perception of the environment.