A visual servoing approach for road lane following with obstacle avoidance

Danilo Alves de Lima, Alessandro Corrêa Victorino

A visual servoing approach for road lane following with obstacle avoidance

Danilo Alves de Lima and Alessandro Corrêa Victorino

***Abstract*— This paper presents a local navigation strategy with obstacle avoidance applied to autonomous robotic automobiles in urban environments, based on the validation of a Visual Servoing controller in a Dynamic Window Approach. Typically, Visual Servoing applications do not consider velocity changes to stop the robot in danger situations or avoid obstacles, while performing the navigation task. However, in several urban conditions, these are elements that must be deal with to guarantee the safe movement of the car. As a solution for this problem, in this work a line following Visual Servoing controller will be used to perform road lane following tasks and its control outputs will be validated in an Image-Based Dynamic Window Approach. The final solution is a validation scheme for the Visual Servoing velocities which allows the obstacle avoidance, taking into account the car kinematics and some dynamics constraints. Experiments in simulation and with a full-sized car show the viability of the proposed methodology.**

***Index Terms*— Visual Servoing, Dynamic Window Approach, Local Navigation, Obstacle Avoidance.**

I. Introduction

Traditionally, a car-like robot performs its movement based on a trajectory planned from its perception and localization sensor's data. The localization is common related to GPS data which must deal with noise and signal losses, some described by many participants of the DARPA Grand Challenges competitions held by the American's Defense Advanced Research Projects Agency (DARPA) between 2004 and 2007 [1]. For global navigation tasks, a localization system is important to the task accomplishment, similarly when the human driver uses the GPS information or his previous knowledge to navigate in a city. However, in a local navigation task, following GPS points results in many path following problems that must be avoided. In recent works, the use of exteroceptive sensors (like LIDAR and vision systems) for automatic vehicle guidance has considerably raised, specially in urban environments, once there are useful features available [2].

Visual Servoing is one of the many ways to deal with this guidance problem using visual features in a sensor-based navigation. It can be divided in two main approaches: the *Position-Based Visual Servoing* (PBVS) and the *Image-Based Visual Servoing* (IBVS) [3]. They are related to where the control objective is expressed, which means in the robot's Cartesian Space or in the image space directly. Based on this principle, several control laws can be defined to allow a vehicle to converge and follow different features and primitives, like points, lines, and ellipses [4], [5]. It is important to mention that there are many others Visual Servoing approaches for nonholonomic robots, combining the both IBVS and PBVS information, as in [6], or without prior knowledge of the scene [7], [8].

Although these tasks can guide the vehicle, they do not directly perform velocities changes to avoid obstacles, essential in urban environments navigation. Considering the obstacle avoidance problem, some approaches define control laws which combine the Visual Servoing task with some obstacle avoidance methodology (e.g., potential field and tentacles) [9], [10], [11]. Normally, these tasks are merged in the control level by some switching methodology that changes the task weight and, consequently, the control law.

This work address the car-like robot local navigation problem using an IBVS as deliberative control for road lane following with a reactive obstacle avoidance, compounding a hybrid control strategy. Once that the IBVS is independent of the vehicle localization and does not consider the presence of obstacles to perform its task, an obstacle avoidance technique must be incorporated to safely perform the local navigation. Using a reduced feature set, based on the one presented in [5], the Visual Servoing equations were integrated to the Dynamic Window Approach (DWA) [12], compounding a reactive obstacle avoidance called by Image-Based Dynamic Window Approach (IDWA) [13]. With the IDWA, the original IBVS velocities are validated, taking in to account the obstacles and the vehicle dynamic/kinematic constraints in the control loop framework. The methodology used diverges from the one presented in [14], where a Velocity Vector Field was integrated to the DWA to perform the navigation of a car-like robot, once it required a path planning and global localization of the vehicle, becoming unsuited for visual local navigation. This work also diverges from the previous ones [9], [10], [11] based on Visual Servoing, once the obstacle avoidance proposed with the DWA incorporates the path following and the velocity control in its calculations. Moreover, differently from [13] that uses the reactive control of the IDWA to guide the car, the present work combines the advantages of both deliberative/reactive controllers in a hybrid control. The objective in mind is to allow electric vehicles, like the one from the project VERVE[1], to perform local navigation in road lanes with a safe behavior.

The proposed methodology was structured in two layers: Workspace Perception and Navigation Control, as show in

The authors are with Heudiasyc UMR CNRS 7253 Université de Technologie de Compiègne. Danilo Alves Lima holds a Ph.D scholarship from Picardi region. Contact authors danilo.alves-de-lima@hds.utc.fr

[1]The project VERVE stands for *Novel Vehicle Dynamics Control Technique for Enhancing Active Safety and Range Extension of Intelligent Electric Vehicles*.

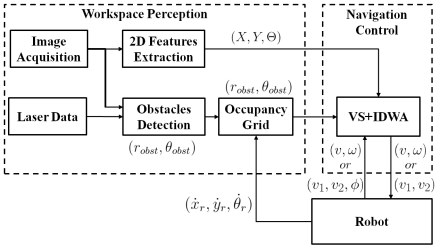

Fig. 1. Methodology block diagram.

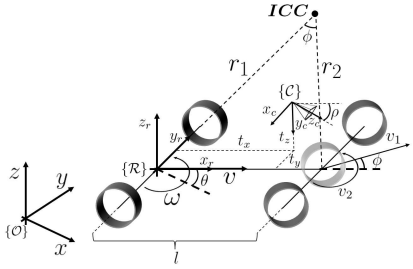

Fig. 2. Kinematic model diagram of a front wheel drive car-like robot centered in the reference frame $\mathcal{R}$. The pinhole camera frame is represented in $\mathcal{C}$.

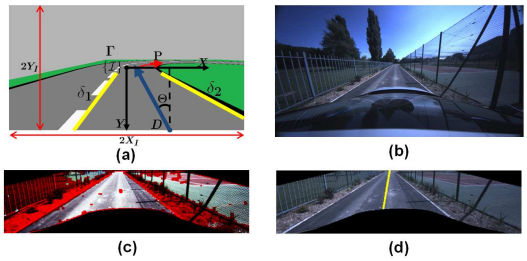

Fig. 3. Acquired image frame $\{\mathcal{I}\}$ from the simulation environment (a) with the road lane center projection P (in red) related to the boundaries δ_1 and δ_2 (in yellow), its tangent Γ (in blue) at the point D and the angle offset Θ of Γ to the axis $-Y$. (b) is a sample image of the real environment, (c) is the road segmentation approach based in [16], and (d) the fitted road center line by the RANSAC approach.

the block diagram of the Figure 1. To present its concepts, this article is structured as follow: Section II presents the robot model used and some definitions; Section III presents the workspace perception layer, describing the strategy used to features extraction and obstacle detection; the navigation control layer, that proposes a validation scheme for the Visual Servoing controller in the Image-Based Dynamic Window Approach, is presented in the Section IV; an experimental analysis and validation of the method, using a simulated and real autonomous vehicle, is in Section V; and Section VI presents some conclusions and perspectives for future works.

II. General Definitions

The robot is considered to move in a planar workspace, similar to the one described in [5], executing an Image-Based Visual Servoing task (IBVS) with a fixed pinhole camera. It follows the road lane center which defines a path once differentiable in $\mathbb{R}^2$. The vehicle is considerate to be over the road surface, and able to always see the road lane. For validation purposes at low speed, a kinematic model of a front wheel drive car was considered, represented as [15]:

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

where the vehicle configuration is given by $q = [x_r\ y_r\ \theta\ \phi]^T$, with the position ($x_r$, y_r) and orientation (θ) of the car's reference frame $\{\mathcal{R}\}$ in relation to a static world reference frame $\{\mathcal{O}\}$, and ϕ is the average steering angle of each front wheel by the Ackerman approximation [15]. The orientation and steering angles (θ and ϕ) are positive counter-clockwise, with $\theta \in]-\pi, \pi]$ and $\phi \in [-\phi_{max}, \phi_{max}]$. The Figure 2 illustrates these variables. Note that 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), and the approximation for the steering angle ϕ is related to the x_r axis, pointed to the front of the vehicle. The control input for the vehicle of the model (1) is $u = [v_1\ v_2]^T$, where v_1 is the linear velocity of the front wheels and v_2 is the steering velocity. In this model, the robot linear velocity v is related to the front wheels velocity by $v = v_1\cos(\phi)$, and the angular velocity $\dot{\theta} = v_1\cos(\phi)/r_1 = \omega$ is directed related to the steering angle (see the Figure 2), which allows to chose the robot control input as $u_r = [v\ \omega]^T$.

The Figure 2 also represents the camera frame $\{\mathcal{C}\}$ with optical center position in $(x_c, y_c, z_c) = (t_x, t_y, t_z)$ in the robot frame and a constant tilt offset $0 < \rho < \frac{\pi}{2}$ related to the x_r axis. In this work the camera was positioned in the robot sagittal plane ($t_y = 0$), which is not a limitation, and must be with a certain height from the floor ($t_z > 0$). Finally, the camera's image frame $\{\mathcal{I}\}$ is represented in the Figure 3a, with defined size as ($2X_I$, $2Y_I$).

III. Workspace Perception

As presented in the Figure 1, the workspace perception is the first step for the navigation task proposed, responsible to provide all environment information (calculated by on-boarded camera and laser scan) required to perform the visual servoing task and obstacle avoidance. It was divided in the 2D features extraction, the obstacle detection and representation in the occupancy grid.

For the visual servoing task, Cherubini et al. [5] implemented a path reach and following strategy using a small set of path features to navigate a nonholonomic robot. The path was defined by the projection in the image plane of a visible white line on the floor, with its features calculated in the image frame $\{\mathcal{I}\}$ for an IBVS scheme. These features, as shown in the Figure 3a, are related to the tangent Γ of the path P (according to its direction) at D, with an angular offset $\Theta \in]-\pi, \pi]$ from Γ to the axis $-Y$ (positive counterclockwise). In this work, the path P was defined as the center of the road surface between the boundaries δ_1 and δ_2, which are on the limit of the most right visible lane or,

$$L_s = \begin{bmatrix} \frac{-\sin\rho - Y\cos\rho}{t_z} & 0 & \frac{X(\sin\rho + Y\cos\rho)}{t_z} & XY & -1-X^2 & Y \\ 0 & \frac{-\sin\rho - Y\cos\rho}{t_z} & \frac{Y(\sin\rho + Y\cos\rho)}{t_z} & 1+Y^2 & -XY & -X \\ \frac{\cos\rho\cos^2\Theta}{t_z} & \frac{\cos\rho\cos\Theta\sin\Theta}{t_z} & -\frac{\cos\rho\cos\Theta(Y\sin\Theta + X\cos\Theta)}{t_z} & -(Y\sin\Theta + X\cos\Theta)\cos\Theta & -(Y\sin\Theta + X\cos\Theta)\sin\Theta & -1 \end{bmatrix} \quad (2)$$

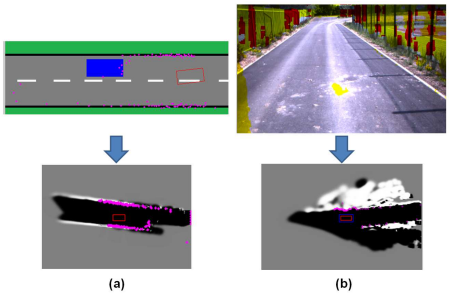

Fig. 4. Obstacle detection for the simulation (a) and real environments (b).

in case of non lane marks, are on the road limits. With this defined path, the Γ features were defined similarly to the ones presented by Cherubini et al. [5]. These features can be better visualized in the Figure 3a.

Since the simulation environment can be easily structured and environment perception was not the focus, the features detection was readily accomplished. However, the simulated camera uses the same intrinsic parameters of the real camera of the final experiments. For the real scenario (Figure 3b), the features detection was performed in a road segmented image, based on [16], which the final result is in the Figure 3c. The final image features were calculated by detecting the road boundaries in the previous segmentation and then fitting a line by the RANSAC approach, giving the line of the Figure 3d.

To guarantee the robot surrounding perception and perform the obstacle avoidance maneuvers, the occupancy grid [17] is the tool for this task. It is a probabilistic technique which maps the workspace using sensors data acquired during the robot movement. Considering that no entire environment information must be on the grid, the occupancy grid can be reduced to a local window around the robot, actualized with its movement (see Figure 1). The occupancy grid was filled with road surface and obstacle information from a laser sensor (for simulation experiments) and a stereo camera (for real experiments), as presented in the Figures 4a and 4b. The obstacle detection for the real experiments were based on a combination of the U/V disparity maps [18], with the results represented in the occupancy grid as described in [19].

IV. Navigation Control

The present controller was divided in two base schemes, one to perform the Image-Based Visual Servoing (IBVS) [5] to follow the road path on the floor, as a deliberative controller, and other to perform the obstacle avoidance maneuvers, as a reactive controller, with the validation of the Visual Servoing velocities in the Image-Based Dynamic Window Approach (IDWA) [13]. This section will describe this controller, represented by the block VS+IDWA of Figure 1.

A. Visual Servoing Controller

This subsection describes the formulation used in Cherubini et al. [5] for the IBVS approach to follow a line path projected in the image frame. Let us consider the image plane presented in the Figure 3a, and the features set $s = [X \;\; Y \;\; \Theta]^T$ defined by the tangent Γ, which is the same features base used by Cherubini et al. [5], as explained in the Section III. 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 road. It starts defining a constant linear velocity $v = v_d > 0$ regarding the road speed limits, and applying a nonlinear feedback control law in the angular velocity ω.

Cherubini et al. [5] describes two primitive controllers (row and column), which have the same principle. The controller must relate the image features velocities $\dot{s} = [\dot{X} \;\; \dot{Y} \;\; \dot{\Theta}]^T$ to the robot velocities $u_r = [v \;\; \omega]^T$. First of all, the image features velocities must be written in therms of the camera frame velocities $u_c = [v_{c,x} \; v_{c,y} \; v_{c,z} \; \omega_{c,x} \; \omega_{c,y} \; \omega_{c,z}]^T$. Using the interaction matrix $L_s(X, Y, \Theta)$ (2), expressed for a normalized perspective camera model, thus:

$$[\dot{X} \; \dot{Y} \; \dot{\Theta}]^T = L_s(X, Y, \Theta) u_c. \quad (3)$$

Note that each line of the matrix L_s are related to its respective image feature (L_X, L_Y and L_Θ). The robot velocities u_r can be expressed in the camera frame $\{C\}$ by (4) using the homogeneous transformation (5):

$$u_c = {}^C T_R u_r, \quad (4)$$

$${}^C T_R = \begin{bmatrix} 0 & -t_x \\ -\sin\rho & 0 \\ \cos\rho & 0 \\ 0 & 0 \\ 0 & -\cos\rho \\ 0 & -\sin\rho \end{bmatrix} \quad (5)$$

Each column of the transformation ${}^C T_R$ are related to the robot velocity, named by T_v and T_ω. The row controller must drive (X,Θ) to a desired set point (X^*,Θ^*), regulating the error $e = [X - X^* \;\; \Theta - \Theta^*]^T$ to zero. This task is performed under the constraint $Y = const = Y^*$. Under this constraint the system state equations are:

$$[\dot{X} \; \dot{\Theta}]^T = A_r v + B_r \omega, \quad (6)$$

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:

$$\omega = -B_r^+(\lambda e + A_r v), \tag{7}$$

where $\lambda = [\lambda_X \ \ \lambda_\Theta]^T$ are positive gains. The column controller can be analogously defined changing the row controller definitions from X to Y. For more details of the implementation and stability analysis see [5].

B. VS+IDWA

The Visual Servoing (VS) methodology presented in the previous subsection do not guarantee safeness to the car movement, once the obstacle information and the vehicle constraints and dimensions are not considered. Due to this, the VS velocities must be validated before being applied in the robot. The Dynamic Window Approach (DWA) is a reactive obstacle avoidance technique proposed originally by [12], with a modification for car-like robots presented in [20], which selects in the velocity space the best control input to the robot regarding some conditions. It takes into account the weighted sum of three functions based on the goal position (*heading*), the obstacle collision distance (*dist*) and the final linear velocity (*velocity*), compounding an objective function (8) to be optimized.

$$\begin{aligned} DWA(v,\omega) =& \alpha \cdot heading(v,\omega) + \beta \cdot dist(v,\omega) \\ &+ \gamma \cdot velocity(v). \end{aligned} \tag{8}$$

When the VS velocities are invalid, the DWA objective function is performed to return a new valid velocity input. Although the resulted velocities are valid, it is necessary to guarantee the same VS goal in the DWA to continue with the road lane following in focus while avoiding the obstacle. For that, in our previous work [13] a new approach for the DWA was presented, which considers 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) and the main change in the objective function (8) is concerned to the function *heading*, as explained bellow.

1) The IDWA Functions: The first function $heading(v,\omega)$, in the original formulation of the DWA [12], is responsible to guide the vehicle to a desired goal. It calculates high weights to the velocity inputs which lead the vehicle to a final orientation closer to the goal position. Based on the robot localization, the goal is precisely known in the world. However, in the present VS application the robot localization is not required, and the goal is leading the robot to the condition were the image features error is closer to zero (see Sub-Section IV-A).

Recalling the VS features (the tangent Γ in the Figure 3) for the row (X, Θ) and column (Y, Θ) controllers, to allow the IDWA to reduce the features errors, the function $heading(v,\omega)$ was divided in: $vs1(v,\omega)$, responsible for the row/column error (X or Y); and $vs2(v,\omega)$ with the Θ error. The main point for both functions is to estimate the position of the tangent Γ, formed in the image frame $\mathcal{I}_{t+\triangle t}$, if applied the control input (v, ω). This can be acquired using the equations (3) and (4) to estimate the features velocity $\dot{s}$ and integrating the computed values over the time (see [13] for further details). The final values are calculated as:

$$vs1 = \begin{cases} 1 - \frac{|e_X|}{e_{Xmax}}, & \text{if row controller,} \\ \\ 1 - \frac{|e_Y|}{e_{Ymax}}, & \text{otherwise.} \end{cases} \tag{9}$$

$$vs2 = 1 - \frac{|e_\Theta|}{\pi}. \tag{10}$$

where e_X, e_Y, and e_Θ are the features errors in the image frame $\mathcal{I}_{t+\triangle t}$ and e_{Xmax} and e_{Ymax} are the maximum measurable errors in X and Y. The final value is defined as:

$$heading(v,\omega) = \alpha_1 vs1(v,\omega) + \alpha_2 vs2(v,\omega). \tag{11}$$

The next function $dist(v,\omega)$ is the normalized distance to collision, calculated for polygonal robots as proposed by [21]. The last function, $velocity(v)$ is calculated based on the desired robot linear velocity v_d from the VS approach of the Subsection IV-A (which is constant regarding to the road speed limit), as follow:

$$velocity = \begin{cases} \dfrac{v}{(v_d - v_{min})} & \text{if } v \leq v_d, \\ \\ \dfrac{(v_{max} - v)}{(v_{max} - v_d)} & \text{if } v > v_d. \end{cases} \tag{12}$$

2) The IDWA Search Space and VS validation: Initially, for the robot current velocity (v_a, ω_a), the Dynamic Window V_d is defined for all reachable velocities in a time interval $\triangle t$ as:

$$\begin{aligned} V_d = \{(v,\omega) |\ & v \in [v_a - \dot{v}\triangle t, v_a + \dot{v}\triangle t], \\ & \omega \in [\omega_a - \dot{\omega}\triangle t, \omega_a + \dot{\omega}\triangle t]\}, \end{aligned} \tag{13}$$

with the robot input set $u_r = [v\ \omega]^T$ (see section II) and the robot accelerations $(\dot{v}, \dot{\omega})$.

Following, each reachable velocity must be classified in admissible or not due to the obstacle collision distance (function $dist(v,\omega)$ defined previously and proposed by [21]) and the robot maximum breaking accelerations ($\dot{v}_b$, $\dot{\omega}_b$). If the distance to the obstacle in a circular trajectory is bigger than the distance required to stop safely the vehicle, then the velocity is admissible. The resulting set is defined as:

$$\begin{aligned} V_a = \{(v,\omega) |\ & v \leq \sqrt{2 \cdot dist(v,\omega) \cdot \dot{v}_b}, \\ & \omega \leq \sqrt{2 \cdot dist(v,\omega) \cdot \dot{\omega}_b}\}. \end{aligned} \tag{14}$$

Finally, the Dynamic Window search space considering the current speed of the vehicle, its accelerations/physical limits, and the obstacles in the workspace is computed as:

$$V_{DW} = V_d \cap V_a \cap V_s, \tag{15}$$

where V_s is the set of points that satisfy the maximum acceleration constraints $\dot{v}_{max}$ and $\dot{\omega}_{max}$. By discretization of the search space V_{DW}, a velocity must be selected following

Fig. 5. Car-like robot from the project ROBOTEX.

the criteria presented by the objective function (8). The resulted search space V_{DW} is the one used to validate the control input calculated from the VS approach. For the case where this input are not allowed, a new control input is calculated by the objective function defined previously.

V. Experimental Results

To validate the navigation methodology proposed, both simulation and real environment experiments were performed. In these situations, the vehicle moves based on the kinematic model of the equation 1, respecting its kinematics constraints and some actuators dynamics. They were defined to be equal to those from our experimental electrical car-like robot from the project ROBOTEX in Figure 5. For the workspace perception, the vehicle uses a monocular camera with a focal length of $1.8mm$ and large field of view ($\simeq 140°$) to detect the road lane center, as described in Section III. The camera is in a rigid structure with tilt offset $\rho \simeq 7°$ and $(t_x, t_y, t_z) = (2.0, 0.0, 2.0)m$. It also detect the obstacles from a laser sensor with $180°$ of coverage (simulation experiments) or a stereo vision camera (real experiments). The detected obstacles data are represented in an occupancy grid [17], locally constructed around the robot. This occupancy grid is updated by a bidimensional Gaussian model to each sensor measurement and the relative movement of the robot frame, using its proprioceptive information, like odometry, velocity and steering angle, which is enough for low speed experiments.

The simulation environment was codified using Matlab with the road surface approximated as the Figure 6a. It implements the methodology presented in Figure 1 to adjust the Visual Servoing (VS) controller parameters and the Image-Based Dynamic Window Approach (IDWA) validation during the robot navigation. Using the right road lane center as reference for the Image-Based Visual Servoing (IBVS), similar to the Figure 3a, the vehicle trajectory without considering the obstacles is performed as the Figure 6b. In this figure, the result presented in [5] is confirmed showing that the IBVS is not robust for path reaching which results in large overshoots on the final path. However, for the path following, when the lateral error is small, the robot can track the road lane with better precision than a position-based controller, not presented here.

To avoid this path reaching problem, the VS+IDWA controller was applied considering the road limits, obstacles and linear velocity variations. The gains α, β, and γ from

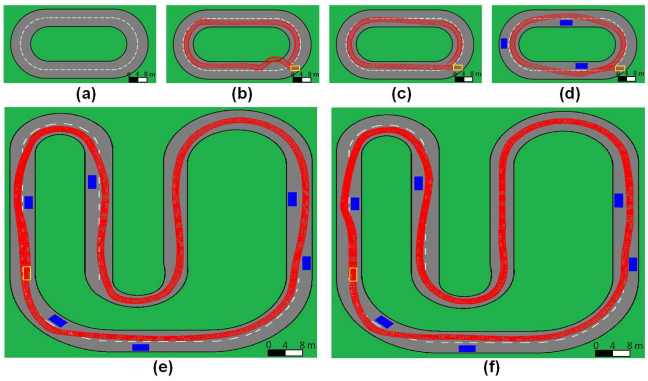

Fig. 6. Simulation environment (a), with the Image-Based Visual Servoing (IBVS) controller performing the lane following with $\lambda = 0.7$ (b). The VS+IDWA for the same task is in (c) and with some static obstacles is in (d), where $\lambda = 0.7$, $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. For comparison, the car trajectory using only the reactive controller IDWA is in (e) and the complete solution VS+IDWA integrating the deliberative and reactive controllers are in (f). The car initial pose is represented in yellow, and in red are the car instantaneous positions for a clockwise movement.

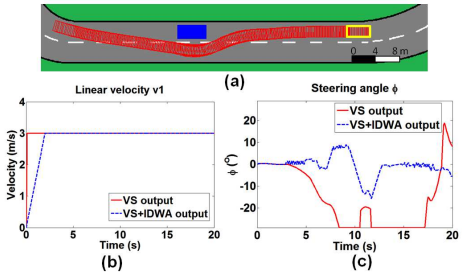

Fig. 7. VS and the VS+IDWA output for the simulated path represented in (a). In (b) is the linear velocity and (c) is the steering angle ϕ.

equation 8, were adjusted similarly to the result presented in [13]. The complete adjustment of the VS+IDWA can be seen at Figures 6c, 6d, and 6f, where the gains were set to $\lambda = 0.7$, $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$ to guarantee a movement that follows the VS features and avoid smoothly the obstacles. To compare the reactive controller IDWA alone and the hybrid controller VS+IDWA, the Figures 6e and 6f were created. The vehicle was able to avoid the obstacles in both cases, but when there are no frontal obstacles, the hybrid solution presents a better performance to guide the robot fast to the reference condition, which means the robot in the road lane center.

To verify the VS and VS+IDWA outputs command, the Figure 7 shows a 20 seconds of simulation during an obstacle avoidance maneuver. As expected in Figure 7(b-c), when the VS outputs are admissible to be applied on the robot (see Section IV), its values are kept by the IDWA. However, after 3 seconds the IDWA starts to modify the VS output to guarantee the obstacle avoidance.

The real size car-like robot experiments were performed in a non-structured circuit, as the one of Figure 8a, which reproduces the road center following experiment. In this case, the robot must move respecting the road boundaries limits and the desired linear velocity of $1m/s$. As expected,

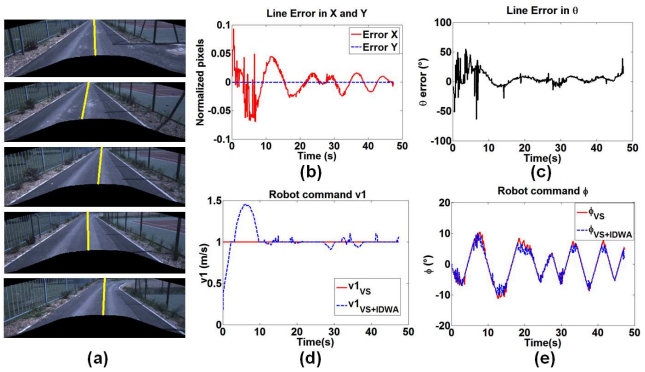

Fig. 8. Road center following applying the VS+IDWA, where (a) presents some detected image features sequence during the experiment, with the evolution of the X/Y errors in (b) and the θ error in (c). The final output calculated by VS and the VS+IDWA are in (d) and (e).

the robot was able to correct the detected features errors (Figures 8b-c) and converge to the road center, even in the presence of considerable variations in the features detection. The validation of the VS velocities in the IDWA can be observed in the Figures 8d-e, where the linear velocity and steering angle were limited by the current approach.

VI. Conclusions and Future Works

This work presented a local navigation approach for car-like robots in urban environments, combining an Image-Based Visual Servoing with an Image-Based Dynamic Window Approach in a hybrid controller called VS+IDWA. Using only the IBVS controller, the accomplishment of the road lane following task was not guaranteed, due to some limitations like: the path reaching problems, the constant linear velocity, and no obstacle avoidance. The hybrid VS+IDWA controller was validated in simulation, performing the road lane following with obstacles avoidance in different scenarios. A full-sized car-like robot experiment also showed the viability of the proposed methodology. In the future, more experiments using the full-sized car-like robot must be performed, considering a dynamic model to increase the vehicle speed and some improvements in the workspace perception layer to reduce the features and obstacles detection variations.

Acknowledgement

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 the European Fund of Regional Development FEDER. The authors wish to thank the helpful assistance of Mr. Pierre Hudelaine during the experiments.

References

[1] 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. [Online]. Available: http://dx.doi.org/10.1002/rob.v25:9

[2] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10846-008-9235-4

[3] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[4] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313–326, 1992.

[5] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, pp. 1037–1048, 12 2011. [Online]. Available: http://journals.cambridge.org/article_S0263574711000221

[6] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 2, pp. 238–250, 1999.

[7] P. Rives, "Visual servoing based on epipolar geometry," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 1, 2000, pp. 602–607 vol.1.

[8] G. Silveira, E. Malis, and P. Rives, "Visual servoing over unknown, unstructured, large-scale scenes," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 4142–4147.

[9] V. Cadenat, R. Swain, P. Soueres, and M. Devy, "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*, vol. 2, 1999, pp. 775–780 vol.2.

[10] D. Folio and V. Cadenat, "A redundancy-based scheme to perform safe vision-based tasks amidst obstacles," in *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, 2006, pp. 13–18.

[11] A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 189–205, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/2/189.abstract

[12] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, 1997.

[13] D. A. Lima and A. C. Victorino, "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*, Juin 2014, pp. 120–125.

[14] D. Lima and G. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *Journal of Control, Automation and Electrical Systems*, pp. 1–11, 2013. [Online]. Available: http://dx.doi.org/10.1007/s40313-013-0006-5

[15] A. D. Luca, G. Oriolo, A. De, and C. Samson, *Robot Motion Planning and Control*. Springer Berlin / Heidelberg, 1998, vol. 229, ch. Feedback Control Of A Nonholonomic Car-Like Robot, pp. 171–253.

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

[17] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

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

[19] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M. Meinecke, "Stereo-camera-based urban environment perception using occupancy grid and object tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 154–165, March 2012.

[20] K. Rebai, O. Azouaoui, M. Benmami, and A. Larabi, "Car-like robot navigation at high speed," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 2053–2057.

[21] K. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-time obstacle avoidance for polygonal robots with a reduced dynamic window," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 2002, pp. 3050–3055.