



HAL
open science

An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments

Danilo Alves de Lima, Alessandro Corrêa Victorino

► To cite this version:

Danilo Alves de Lima, Alessandro Corrêa Victorino. An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments. IEEE ICRA Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots (WMEPC 2014), May 2014, Hong Kong, Hong Kong SAR China. <hal-01088389>

HAL Id: hal-01088389

<https://hal.science/hal-01088389v1>

Submitted on 27 Nov 2014

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.



HAL Authorization

An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments

Danilo Alves de Lima and Alessandro Corrêa Victorino

Abstract—This paper presents a local navigation strategy for autonomous vehicles in urban environments with an Image Based Dynamic Window Approach (IDWA). Differently from the global navigation techniques, which requires the vehicle localization to perform its movement, the focus here was to solve the navigation problem in local navigation steps. For that, the environment features will be used, performing the road lane following e.g. The DWA performs a reactive obstacle avoidance while trying to reach a goal destination. In this case, reach the goal destination is based on the Image Based Visual Servoing equations for road lane following, which were incorporated into the DWA. The final solution takes into account the car kinematics/dynamics constraints to allow the vehicle to follow the road lane while avoiding obstacles. The results show the viability of the proposed methodology.

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

I. INTRODUCTION

Traditionally, a car-like robot navigates based on its perception of the environment and its localization related to the path previously planned. However, in urban environments, the localization systems based in GPS information must deal with signal losses and several noises caused by urban canyons, as reported by many DARPA's Challenges participants between 2004 and 2007 [1]. Due to the GPS drift, this problem also increases locally when a vehicle tries to follow a road based on GPS points. One way to deal with it is using local navigation techniques, which do not use the vehicle global position to calculate the control action. These techniques are normally based on local features extracted from exteroceptive sensors (like vision systems), which in urban environments there are useful ones available [2]. Thus, a global navigation task in urban environments can be divided in road following (branches), representing the local tasks, and road intersections maneuvers (nodes), connecting the next local task. To accomplish the global task, the global localization system can be limited only to the nodes, e.g. using techniques based on vehicle-to-infrastructure (V2I) communication [3].

Focusing on the local navigation, there are many approaches to perform vehicle control using visual data [4], [5], [2]. In addition to follow the desired features the vehicle must also consider reactive techniques for obstacle avoidance. A well-known reactive technique is the Dynamic Window Approach (DWA) [6], which searches for an optimal input

command between all possible commands in a short time interval. Its optimization function takes into account the final goal position (*heading*), the obstacles distance (*dist*), and the maximum linear velocity (*velocity*) during the calculation. It also considers the kinematics and some dynamics constraints of the robot. Due to the nature of its optimization function, it can be adapted to several techniques [7], [8], [9], to different robot types, like car-like robots [10], [11], as well as to dynamic environments [12]. However, these works using DWA are conceived when the robot and goal positions are known in the world frame, recalling the localization problems previously mentioned.

To avoid this problem, the proposed work presents a local navigation strategy for autonomous vehicles in urban environments with an Image Based Dynamic Window Approach (IDWA). Differently from the global DWA techniques [7], [13], [9], which requires the vehicle localization to perform its movement, the focus here is to solve the navigation problem in local navigation steps using the environment features acquired from a camera, performing e.g. the road lane following. In this case, reaching the goal destination is a task guaranteed by the Image Based Visual Servoing equations [4], [14], [15], incorporated into the DWA functions. This work also differs from the vision navigation approach proposed by [16], based on the tentacles technique of [1], once the image based task and the obstacle avoidance are in the same controller on the robot velocity space. The objective in mind is to perform Image Based Visual Servoing control tasks while validating their velocity outputs in a obstacle avoidance methodology. In the near future, it will allow electric vehicles, like the one from the project VERVE¹, to perform local navigation in road lanes with a safe behavior.

In the block diagram of the Figure 1 it is shown the present methodology, structured in two general layers: workspace perception and robot control. Its concepts are presented in this article as follow: Section II presents the robot model used and the problem definition; Section III presents the workspace perception layer, describing the environment perception strategy to features extraction and obstacle detection; the navigation control layer, with the proposed Image Based Dynamic Window Approach, is presented in the Section IV; an experimental analysis and validation of the method, using a simulated car-like robot, is in Section V; and, finally, Section VI presents some conclusions and perspectives for future works.

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

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

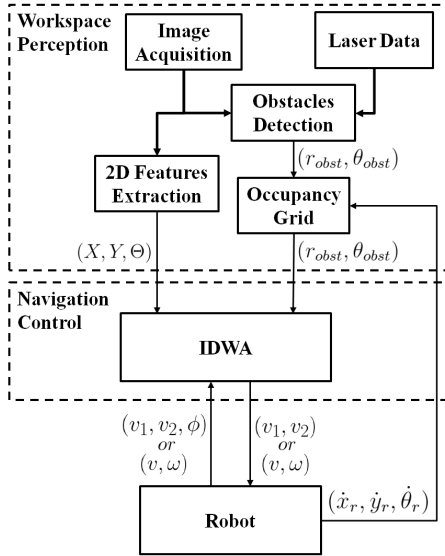


Fig. 1. Methodology block diagram.

II. GENERAL DEFINITIONS

The car-like robot used in this work is similar to the ones described in [15], [13]. It is considered to move in a planar workspace with a fixed pinhole camera directed to the front to perceive and follow the road lane center, which defines a path once differentiable in \mathbb{R}^2 . The vehicle is also considered to be over the road surface and able to always see a road lane. The kinematic model is based on a front wheel car, represented as [17]:

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

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 [17]. The orientation and steering angles $(\theta$ and $\phi)$ are defined as $\theta \in]-\pi, \pi]$ and $\phi \in [-\phi_{max}, \phi_{max}]$, both positive counter-clockwise. 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). The approximation for the steering angle ϕ is related to the x_r axis, pointed to the front of the vehicle.

For the vehicle model (1), the control input 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 $u_r = [v \ \omega]^T$. These inputs can be generalized for an unicycle robot, although the

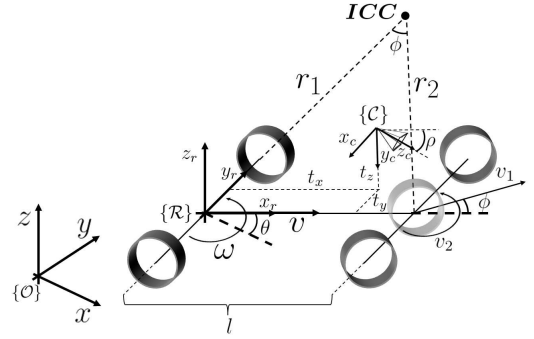


Fig. 2. Kinematic model diagram for a front wheel car-like robot. In this model the vehicle reference frame \mathcal{R} performs circular trajectories related to the instantaneous center of curvature (ICC). The pinhole camera frame is also represented in \mathcal{C} .

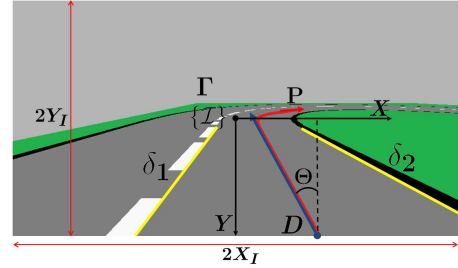


Fig. 3. Image frame $\{\mathcal{I}\}$ with the road lane center projection P (in red) related to the boundaries δ_1 and δ_2 (in yellow), its tangent Γ (in blue) at the point D and the angle offset Θ of Γ to the axis $-Y$.

robot frame $\{\mathcal{R}\}$, in the unicycle, be located in the projection of the wheel center on the ground. The main difference is regarding a constraint present in the car-like robot model, which limits the ICC (Figure 2) by ϕ , and that is not present in the unicycle robot.

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, required for the image based approach. The camera final position is in the robot sagittal plane ($t_y = 0$), which is not a limitation, but must be with a certain height from the floor ($t_z > 0$). Finally, the camera's image frame $\{\mathcal{I}\}$ is illustrated in the Figure 3, with a defined size of $(2X_I, 2Y_I)$.

III. WORKSPACE PERCEPTION

The workspace perception is the first step for the local navigation task proposed (Figure 1), that provides the environment information (calculated by on-boarded camera and laser scan) required to perform the Image Based Dynamic Window Approach (IDWA). It was divided in 2D features extraction, obstacle detection and occupancy grid representation.

The current implementation of the IDWA uses a similar features set presented by Cherubini et al [15], applied in a path reach and following strategy of a nonholonomic robot. It uses small path features set to navigate, defined as the projection in the image plane of a visible white line on the floor, with its features calculated in the image frame

$\{\mathcal{I}\}$ in an Image Based Visual Servoing scheme [14]. These features were adapted for the road lane following problem as described in Figure 3, where they are related to the tangent Γ of the path P (according to its direction) at the point $D = (X, Y)$, with an angular offset $\Theta \in]-\pi, \pi]$ from Γ to the axis $-Y$ (positive counterclockwise). P is 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, in case of non lane marks, are on the road limits.

An obstacle detection layer is also necessary in the IDWA to guarantee the right execution of obstacle avoidance maneuvers, and with an occupancy grid [18] the obstacles can be stored during the robot movement. Once 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). For more details about the obstacle detection and occupancy grid layers, see [13]. This implementation considers only static environments for validation purposes, which does not restrain a future implementation with dynamic environments as presented in [12].

IV. NAVIGATION CONTROL

The present controller was based on the integration of the Image Based Visual Servoing (IBVS) [15] equations with the Dynamic Window Approach (DWA) [6] to perform the obstacle avoidance while performing the road lane following. This technique was called by Image Based Dynamic Window Approach (IDWA), used by the navigation control layer of Figure 1, and will be presented in this section.

A. The Dynamic Window Approach

DWA is a reactive obstacle avoidance technique proposed originally by [6], adapted for car-like robots by [10], which selects in the velocity space an optimum control input around the current robot state. It regards some kinematics/dynamics conditions of the robot to construct a control search space, classified by the weighted sum of three functions. They are based on the goal position (*heading*), the obstacle distance (*dist*) and the final linear velocity (*velocity*), compounding the objective function (3):

$$DWA(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega), \quad (3)$$

to be optimized.

1) *The DWA Functions*: In the original formulation of the DWA [6], the function *heading*(v, ω) is responsible to guide the robot to a desired goal position, calculating high weights to the velocity inputs that lead the robot to a final orientation closer to the goal position in the world frame. It is frequently adapted when some specific navigation task is required [7], [13], [9]. Its improvements proposed for this work will be presented in the Subsection IV-B. The next function *dist*(v, ω) is the normalized distance to collision when performing circular movements, calculated for polygonal robots as proposed by [19]. It uses the obstacle information from the occupancy grid described in Section III.

To avoid unsafe conditions while performing the obstacle avoidance, a similar consideration from [1] was applied to expand the robot neighborhood in the *dist* evaluation. The last function, *velocity*(v, ω) is calculated based on the desired robot linear velocity v_d (which is constant regarding to the road speed limit), as follow:

$$\text{velocity} = \begin{cases} \frac{v}{(v_d - v_{min})} & \text{if } v \leq v_d, \\ \frac{(v_{max} - v)}{(v_{max} - v_d)} & \text{if } v > v_d. \end{cases} \quad (4)$$

The importance of these previous functions in the objective function is adjusted by the constant gains α , β and γ .

2) *The DWA Search Space*: Initially, for the current vehicle velocity (v_a, ω_a), the Dynamic Window V_d is defined for all reachable velocities in a time interval Δt as:

$$V_d = \{(v, \omega) \mid v \in [v_a - \dot{v}\Delta t, v_a + \dot{v}\Delta t], \omega \in [\omega_a - \dot{\omega}\Delta t, \omega_a + \dot{\omega}\Delta t]\}, \quad (5)$$

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

Once defined the reachable velocities, they must be classified in admissible or not due to the obstacle distance (function *dist*(v, ω) defined previously and proposed by [19]) and the robot maximum breaking accelerations ($\dot{v}_b, \dot{\omega}_b$). The resulting set is defined as:

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b}, \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b}\}. \quad (6)$$

Finally, the Dynamic Window search space is computed as:

$$V_{DW} = V_d \cap V_a \cap V_s, \quad (7)$$

where V_s is the set of points that satisfy the maximum acceleration constraints \dot{v}_{max} and $\dot{\omega}_{max}$. It considers the current speed of the vehicle, its accelerations/physicals limits, and the obstacles in the workspace. By discretization of the search space V_{DW} , a velocity must be selected following the criteria presented by the objective function (3).

B. The Image Based Dynamic Window Approach

Considering the objective function (3) and the search space (7), the main changes for the Image Based Dynamic Window Approach (IDWA) concern to the function *heading*(v, ω). As previously mentioned, it is responsible to guide the robot to a desired goal in the world frame. For the present formulation, the goal is to lead the features set $s = [X \ Y \ \Theta]^T$, defined in Section III by the tangent Γ in the image frame $\{\mathcal{I}\}$ (see Figure 3), to the final configuration $X^* = \Theta^* = 0$ and $Y^* = Y_I$, which means the vehicle in the center of the road.

Based on the Image Based Visual Servoing (IBVS) equations proposed by Cherubini et al [15], the *heading* function must estimate the features error:

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

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

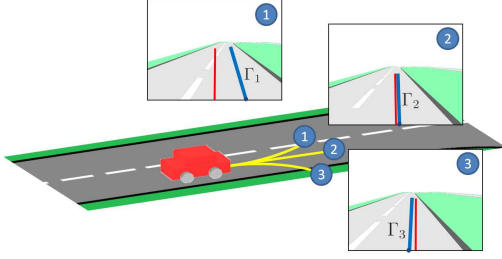


Fig. 4. Estimation of the features set Γ_i (blue line) in the frame $\mathcal{I}_{t+\Delta t}$ applying the control inputs (v_1, ω_1) , (v_2, ω_2) and (v_3, ω_3) . The reference position is also represented in red, which means the vehicle in the center of the road lane.

in the next image frame $\mathcal{I}_{t+\Delta t}$, considering (X^*, Y^*, Θ^*) as the set point. This is illustrated in the Figure 4. Thus, high weight are given to the inputs $(v_i, \omega_j) \in V_{DW}$ which reduce the final error e .

To this end, the controller must relate the image features velocity $\dot{s} = [\dot{X} \ \dot{Y} \ \dot{\Theta}]^T$ to the robot velocity $u_r = [v \ \omega]^T$. First of all, the image features velocity must be written in terms of the camera frame velocity $u_c = [v_{c,x} \ v_{c,y} \ v_{c,z} \ \omega_{c,x} \ \omega_{c,y} \ \omega_{c,z}]^T$. Using the interaction matrix $L_s(X, Y, \Theta)$ (2), expressed for a normalized perspective camera model, yields:

$$[\dot{X} \ \dot{Y} \ \dot{\Theta}]^T = L_s(X, Y, \Theta) u_c. \quad (8)$$

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 velocity u_r can be expressed in the camera frame $\{C\}$ by (9) using the homogeneous transformation (10):

$$u_c = {}^C T_R u_r, \quad (9)$$

$${}^C T_R = \begin{bmatrix} 0 & -t_x \\ -\sin \rho & 0 \\ \cos \rho & 0 \\ 0 & 0 \\ 0 & -\cos \rho \\ 0 & -\sin \rho \end{bmatrix}. \quad (10)$$

The final features configuration can be acquired using the equations (8) and (9) to estimate the features velocity \dot{s} and integrating them over the time interval Δt . Cherubini et al has defined a row and column controller, depending of the $D = (X, Y)$ point location in the image frame $\{\mathcal{I}\}$ (see Figure 3). The row controller is applied when $Y = const = Y^*$ or the column one otherwise. Under this constraint, the function $heading(v, \omega)$ was divided in: $XY_{error}(v, \omega)$, responsible for the row/column error (X or

Y); and $\Theta_{error}(v, \omega)$ with the Θ error. The final values are calculated as:

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

$$\Theta_{error} = 1 - \frac{|e_\Theta|}{\pi}. \quad (12)$$

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

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

V. EXPERIMENTAL RESULTS

To validate the navigation methodology proposed by the block diagram of Figure 1, a simulation environment using Matlab was created for some road configurations (see Figures 5a and 7a). The vehicle moves based on the kinematic model of the equation 1, respecting its kinematics constraints and some actuators dynamics. It simulates a monocular camera with a focal length $1.8mm$ and large field of view ($\simeq 140^\circ$). The camera tilt offset $\rho = 10^\circ$ and $(t_x, t_y, t_z) = (2.0, 0.0, 2.0)m$. The obstacles are detected by a simulated laser sensor with 180° of coverage in front of the vehicle and the information was stored in an occupancy grid [18], locally constructed around the robot. Each laser reading is represented in the occupancy grid by a bidimensional Gaussian model. The relative movement of the robot frame updates the grid information, using its proprioceptive data, like odometry (in real environments the visual odometry can be applied), velocity and steering angle, which is enough for the simulation purposes and low speed experiments.

The Image based Dynamic Window Approach (IDWA) must considers the road limits, obstacles and linear velocity variations when adjusting the gains α , β , and γ from equation 3. When no obstacle obstructs the robot path, the movement is similar to the one in Figure 5b. To visualize their influence in the final navigation, the Figure 6(a-c) combines them one by one. In the Figure 6a only the function $velocity$ was applied, resulting in a movement with no direction that stops at the first visible obstacle. Enabling the function $dist$, it gives the Figure 6b with a movement over regions free of obstacles but without any goal, better observed in the curves. Adding the $heading$ functions $vs1$ and $vs2$ to the previous ones, it results the Figure 6c. Here the robot avoid the obstacles and follow when possible the

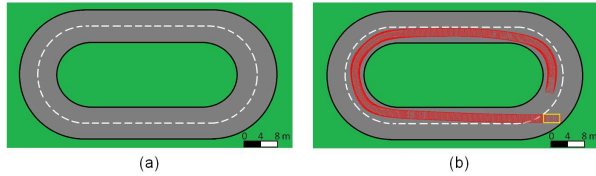


Fig. 5. Environment for car-like robot simulation (a) and its navigation in this condition (b). The car initial pose is represented in yellow, and in red are the car instantaneous positions for a clockwise movement.

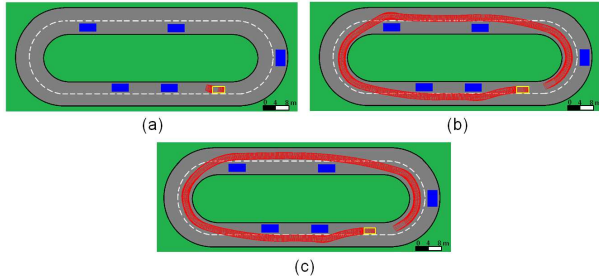


Fig. 6. Evaluation of the IDWA functions in the navigation task for: *velocity* (a), *velocity + dist* (b), and the complete objective function with the *heading* functions $vs1$ and $vs2$ (c). The gains were set to $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

right road lane, guaranteed by our image based task. For this final configuration: $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. It is important to observe that even using only one function in the objective function, the final movement will be safe from collisions, once we are using only the velocities from the DW search space in 7. An extended experiment is presented in Figure 7.

The influence of changing the road lane setpoint when overtaking an obstacle was verified in Figure 8. Comparing the Figures 8a, with the setpoint ways on the right lane, and 8b, where the setpoint changes due to the lane obstruction, the second one presents a smoothest trajectory and farther from the obstacles than the first one. This is better for real world applications, providing more safety for the vehicle movement.

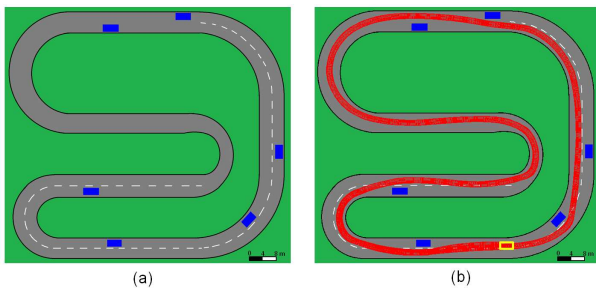


Fig. 7. Environment for car-like robot simulation (a) and its navigation in this condition (b), with $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. In the cases where do not have the road markers, the vehicle follows the road center. The car initial pose is represented in yellow and starts the movement to the left. The obstacles are in blue, and in red are the car instantaneous positions.

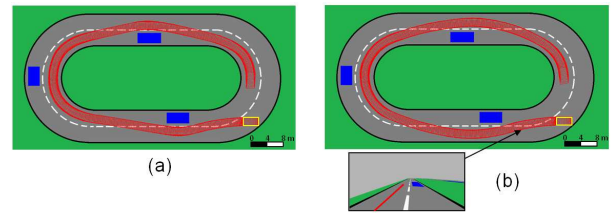


Fig. 8. Vehicle movement with the setpoint defined only on the right lane (a) and switching due to the lane obstruction (b), illustrated by the red line in the camera image from the car point of view. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

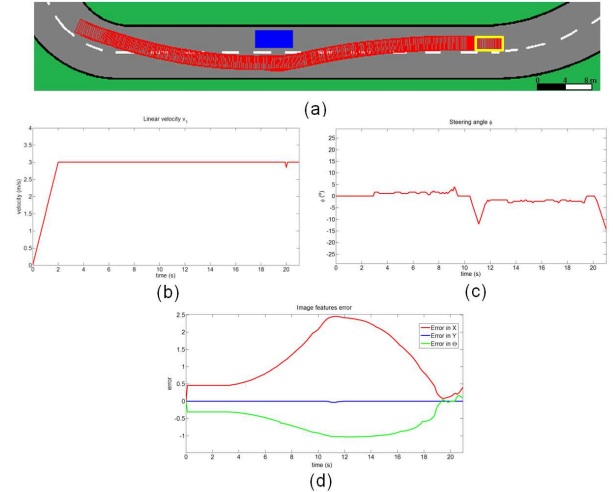


Fig. 9. IDWA controller outputs and image error evolution for the trajectory represented in (a). In (b) is the linear velocity and (c) the steering angle commands. The image features errors are in (d).

Finally, the Figure 9 shows the resulting commands and the image features error evolution during the first 21 seconds of simulation using the IDWA controller. Note that, even if the IDWA is a discrete technique, the output commands are smooth for real cars actuators. This can be better observed in the steering angle (Figure 9c), rarely reaching values higher than $\pm 10^\circ$ in the complete circuit, which results in more comfort for a final user. The image features error converge smoothly to zero when there are no obstacles preventing the vehicle movement, as seen in Figure 9d.

VI. CONCLUSIONS AND FUTURE WORKS

This work presented an image based local navigation approach applicable to car-like robots in urban environments among obstacles. For that, it was integrated the Image-Based Visual Servoing (IBVS) equations, originally conceived to follow lines on the floor with a small features set, in the Dynamic Window Approach (DWA), resulting in a local navigation system independent of the vehicle global localization and the final destination. The Image Based Dynamic Window Approach (IDWA), as it was called here, allowed the robot to perform the path reaching, to follow the road lane center avoiding obstacles. The methodology was tested in a simulation environment over Matlab considering the car

kinematics and some dynamics constraints, and sensors limitations, which provided a solid validation for the proposed solution in a static environment.

The optimization function of the DWA taken in to account the path reaching and following problem, the obstacles avoidance, and the linear velocities variations, in order to find the control inputs that best attend the gains setup. However, find the best adjustment for these gains depends of which element must be considered preferentially, if is to follow the road lane, to keep the distance from the surrounding obstacles, or moves with higher velocities. For the current configuration, the vehicle was able to complete its navigation task with smooth control inputs. The present approach also left us switch the setpoint between the lanes during the overpass maneuver. It is important to mention that other IBVS equations could be integrated in the present solution to allow different tasks.

Thanks to the nature of the elements analyzed, the application of this solution in a real car-like robot could be done with low cost sensors. The experiments are being prepared to the autonomous car Iris of the projects VERVE and ROBOTEX, from the Heudiasyc laboratory. Considering the real environments, more robust techniques must be applied to detect the image features, like the one proposed in [20], as well as the moving obstacles like in [12]. Other applications in mind are related to the human machine interface, improving the interaction between the vehicle and its conductor.

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] O. Hassan, I. Adly, and K. Shehata, "Vehicle localization system based on ir-uwb for v2i applications," in *Computer Engineering Systems (ICES), 2013 8th International Conference on*, Nov 2013, pp. 133–137.
- [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] S. Lee, K. Boo, D. Shin, and S. Lee, "Automatic lane following with a single camera," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, May 1998, pp. 1689–1694 vol.2.
- [6] 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.
- [7] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 341–346.
- [8] P. Ogren and N. Leonard, "A convergent dynamic window approach to obstacle avoidance," *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 188–195, April 2005.
- [9] P. Saranritichai, N. Niparnan, and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on dynamic window approach," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, May 2013, pp. 1–4.
- [10] 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.
- [11] K. Rebai and O. Azouaoui, "Bi-steerable robot navigation using a modified dynamic window approach," in *Proceedings of the 6th International Symposium on Mechatronics and its Applications*, 2009, pp. 1–6.
- [12] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 1986–1991.
- [13] 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>
- [14] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.
- [15] 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>
- [16] A. Cherubini, F. Spindler, and F. Chaumette, "A new tentacles-based technique for avoiding obstacles during visual navigation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 4850–4855.
- [17] 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.
- [18] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [19] 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.
- [20] G. B. Vitor, D. A. Lima, A. C. Victorino, and J. V. Ferreira, "A 2d/3d vision based approach applied to road detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 952–957.