SCISPACE
formerly Typeset

Open access • Proceedings Article • DOI:10.1109/ROBIO.2014.7090699

Autonomous vehicle global navigation approach associating sensor based control and digital maps — Source link

Elisa C. M. Pereira, Danilo Alves de Lima, Alessandro Correa Victorino

Institutions: University of Technology of Compiègne

Published on: 01 Dec 2014 - Robotics and Biomimetics

Topics: Area navigation, Mobile robot navigation, Dead reckoning, Digital mapping and Wind triangle

Related papers:

- Visual navigation of autonomous on-road vehicle
- Map building with RGB-D camera for mobil robot
- Research on navigation system of an autonomous vehicle using RTK-DGPS
- Robot Navigation in a 3D World Mediated by Sensor Networks
- Vision based intersection navigation

Share this paper:

View more about this paper here: https://typeset.io/papers/autonomous-vehicle-global-navigation-approach-associating-3s5ongtm4k

HAL
open science

Autonomous vehicle global navigation approach associating sensor based control and digital maps

Elisa C. M. Pereira, Danilo Alves de Lima, Alessandro Corrêa Victorino

To cite this version:

Elisa C. M. Pereira, Danilo Alves de Lima, Alessandro Corrêa Victorino. Autonomous vehicle global navigation approach associating sensor based control and digital maps. IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), Dec 2014, Bali, Indonesia. pp.2404-2409. hal-01088393

HAL Id: hal-01088393

https://hal.archives-ouvertes.fr/hal-01088393

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.

Autonomous vehicle global navigation approach associating sensor based control and digital maps

Elisa C. M. Pereira, Danilo A. Lima and Alessandro C. Victorino

***Abstract*— This paper proposes a global navigation strategy for autonomous vehicle combining sensor based control and digital maps information. In order to avoid localization problems in urban environments, this approach intends to solve the global navigation focusing on two local navigation tasks: road lane following and road intersection maneuvers. For that, it is important to use digital maps, once they provide a rich database containing information about the environment structure, like speed limit, number of lanes, traffic directions, etc. Associating the data extracted from the digital map with local navigation approaches, the vehicle is able to perform its global navigation task. The experiments take into account real data and a simulated scenario, which show the viability of this approach.**

***Index Terms*— Digital maps, Global Navigation, Sensor Based Navigation.**

I. Introduction

Currently, the research on autonomous vehicle has focused on low cost systems, adapted to different urban environments, since these factors influence the viability of a future production of the vehicle in question. Navigation is a key aspect when developing an autonomous vehicle. To navigates, a car-like robot must be able to perceive its surroundings and localize itself. However, as the surrounding becomes more complex, so does the localization problem. Even with differential corrections, the GPS information fails when the urban environment has structures like tall buildings, overpasses and tunnels [1]. One way to solve this problem is to use a navigation system that do not depend on accurate localization information.

The structure of an urban environment can be divided into corridors and critical points. The corridors represent the streets which are connected by critical points, as road intersections and traffic lights. When the vehicle is in a corridor, the navigation task does not change given that there is only one possible road to follow. At critical points, on the other side, the navigation system must take a decision, like, for example, choosing the correct road to follow from the the current available options in a road intersection or stopping in a traffic light. In this scenario, the car-like robot must localize itself only when it is near a critical point, avoiding the localization problems explained above.

This work is funded in part by the Picardie Region VERVE project, in part by the ANR JCJC RECIF and carried out in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02). The authors are working at Heudiasyc Laboratory, UMR CNRS 7253, Université de Technologie de Compiègne, BP 20529, 60205 Compiègne, France. danilo.lima-de-alves@hds.utc.fr, alessandro.victorino@hds.utc.fr, elisacmpereira@gmail.com

In order to solve the navigation task, it is fundamental to use a database containing information about the environment structure. For this, digital maps provide rich information about streets, tracks, railways, waterways, points of interest detailing number of lanes and speed limits. There are many approaches using digital maps to improve the global navigation problem, like enhancing the vehicle localization as it was done in [2], [3], [4]. In this article, the focus on extracting information from digital maps is not to solve localization problems, but the difficulties of navigation as a whole [5].

Dealing with corridor navigation, there are approaches based on robot proprioceptive information which are able to control the vehicle without accurate position information [6], [7], [8]. These techniques use local environment information acquired from cameras and others sensors to accomplish the local task of road lane following. In addition to use the proprioceptive information, the approach presented in [9] reacts to the environment, thus interacting with obstacles during the robot movement. Even avoiding the localization problems during a local task, these approaches were not developed to deal with road intersections. In this case, road lane features can not be well detected, resulting in wrong movements for the car and failing to complete a global navigation task.

To perform global navigation task, including at road intersections, DARPA's Challenges participants [10] used a detailed predefined map of the path and the route network. With the same objective, [11] proposes a hierarchical world model and distinguishes between a low-level model for the trajectory planning. Differently from these techniques, which requires the vehicle localization to perform its movement, the focus here is to solve the global navigation in local tasks.

These local tasks are divided in corridor following and critical points. The technique presented in [9], which controls the vehicle by combining the road lane following with an obstacle avoidance, is used to solve the local navigation task at corridor. To improve this approach, a routing table will provide information about number of lanes and speed limit, enriching the route knowledge. Considering critical points as road intersections, our approach intends to artificially change the environment information, guiding the robot to the next local navigation task. The goal is to induce the techniques like [12] and [9] to provide the correct destination when there is more than one possibility. For that, it is fundamental to provide information about the critical points, as its structure and approximated geographic position.

The block diagram of the Figure 1 shows the present methodology, structured in two main layers: Global planning

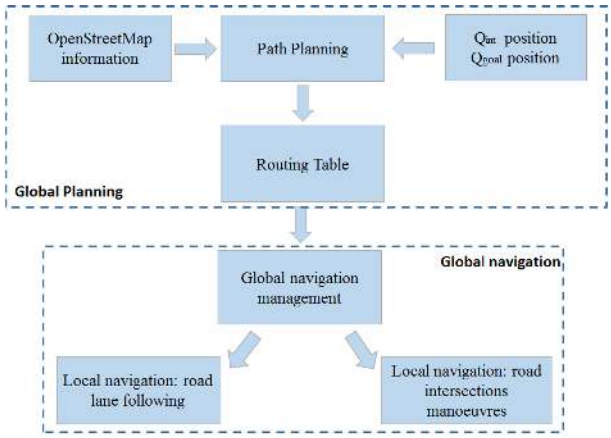

Fig. 1. Methodology block diagram.

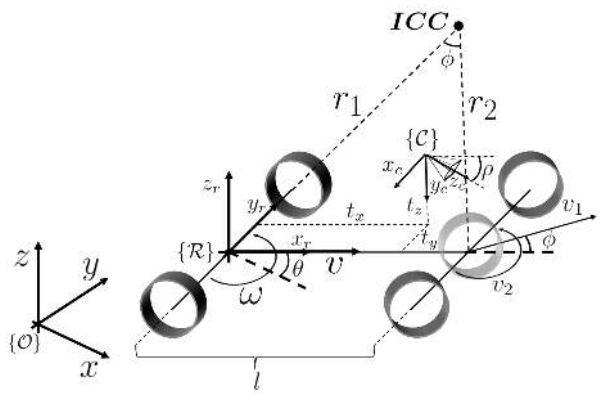

Fig. 2. Kinematic model diagram for a front wheel car-like robot. In this model the vehicle reference frame $\mathcal{R}$ performs circular trajectories related to the instantaneous center of curvature (ICC). The pinhole camera frame is also represented in $\mathcal{C}$.

and Global navigation. This article is organized as follow: Section II presents the robot model and the environment perception; Section III presents the global planning, describing the network of roads; the global navigation is presented in Section IV; Section V presents the experimental results and Section VI the conclusions.

II. General Definitions

The workspace perception (composed by on-boarded camera and laser scan) is responsible for provide the local features necessary to the navigation task. The construction of the workspace perception in 2D follows the order: features extraction, obstacle detection and occupancy grid representation. After extracting the environment information, an obstacle detection is applied with the purpose of characterizing the navigation area. The last step correspond to map the obstacles in a local occupancy grid [13]. Actualized with the robot movement, the occupancy grid contains only the local obstacles around the robot. For more details about the obstacle detection and occupancy grid layers, see [14]. This implementation considers only static environments for validation purposes, which does not restrain a future implementation with dynamic environments as presented in [15].

A low-cost GPS provides the vehicle position (longitude, latitude and altitude). Without any correction, its expected error can reach 6 meters. Even with this rude error, the technique proposed is able to perform the navigation.

To validate the navigation approach proposed in this work, a simulated car-like robot will be used. It moves in a planar workspace with a fixed pinhole camera and a laser scan, both directed to the front to perceive the environment. The kinematic model is based on a front wheel car [16] as:

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

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 [16]. The steering and orientation 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, for more information see [9].

III. Global Route Planning

The global route planning is the first step for the global navigation task, this step is calculated before the robot movements starts and can be actualized if any unexpected change happens on the way. The global route planning is based on OpenStreetMap [17] information.

The OpenStreetMap data include three basic geometric elements: *nodes*, *ways* and *relations*. The *nodes* are geometrical points interconnected by *ways* used to represent the geometry of the *way*, in case of roads, they also represent the critical points, e.g. the road intersections. To fit the geometry, the *ways* represent a set of *nodes* characterizing a specified object like roads, buildings, railways, etc. The element *relations* constructs a relationship of geoobjects, the relation may contain *nodes* and *ways*. To classify the different types of *nodes*, the term key tag is applied according the origin of the *node* (building, railways, highways). In our case, it is necessary to filter the data considering only the *ways* with the key tag "highways", once they correspond to the navigable area.

For path planning, the optimal path from the current q_{init} position to q_{goal} position is calculated using the A* search algorithm [18]. This search algorithm minimizes the cost function, computing the path with the lowest cost. The Euclidean distance between two *nodes* correspond to the cost of the path, in this way, the shortest path is calculated choosing the least-cost paths to reach the q_{goal} position. The result is a list of *nodes* in which the robot must pass (see Figure 3).

From this list, the local information is extracted and organized as a routing table, as shown in the Table I. Four classifications are considered according to the traffic directions at the critical point. If all the *ways* arrive at the *node*, the critical point is classified as arrival point (P_a);

TABLE I

NAVIGATION INFORMATION TO THE PATH PRESENTED IN FIGURE 3

Node	Lon	Lat	P_a	P_d	P_ad	P_r	Angles (degrees)			Directions			Speed (km/h)	N. lanes
1	2,800	49,392	0	0	1	0	0,0	180,0	250,0	0	2	4	50	2
2	2,798	49,380	0	0	1	0	0,0	180,0	290,0	0	2	4	30	2
3	2,805	49,380	0	0	1	0	0,0	90,0	270,0	0	4	3	30	2
4	2,805	49,370	1	0	0	1	0,0	120,0	240,0	0	3	4	30	2
5	2,810	49,365	0	0	1	1	0,0	90,0	240,0	0	2	4	30	2

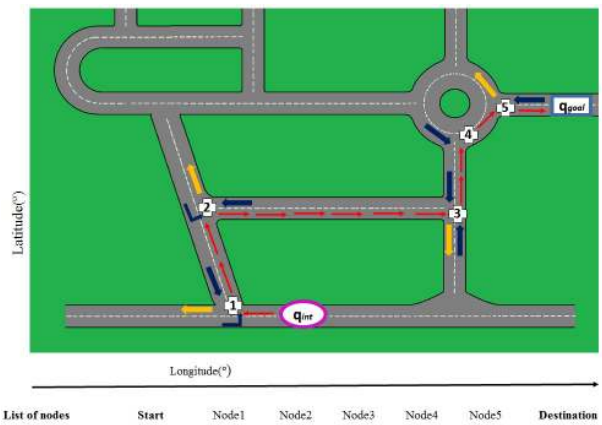

Fig. 3. Optimal path planning from q_{init} position to q_{goal} position with the list of *nodes* representing critical points. The arrows represent the traffic directions at each node, exiting in yellow and arriving in blue.

similarly, the critical point is classified as a departure point (P_d) if all connected *ways* exit. It is also important to take into account the case in which both are at the same *node*, the classification is called arrival and departure (P_ad). The roundabouts must be considered as a special critical point, once the perception requirements increase as well as the risks of accident. Therefore, there is a classification called roundabout point (P_r) which represents this case. The Figure 3 shows traffic directions at critical points using arrows with two different colors: yellow is representing the right way and blue is representing the wrong way. In this way, critical points with only yellow arrows may be considered as departure points, as well as points with only blue arrows may be considered arrival points.

The last two columns of the table focus on the geometry. It is desirable to guide the local navigation task at critical points, thus preventing the car to take wrong ways. The connected *ways* have an angle and may have three directions: path to follow(4), wrong way(3) and right way(2) (see columns angles and directions in Table I). Only the *way* represented as path to follow(4) must be considered as free space.

IV. GLOBAL NAVIGATION

Global navigation takes into account the problem of guide the robot from an initial configuration q_{init} to a final configuration q_{goal}. Focusing on local navigation approaches to avoid localization problems [1], the global navigation task will be divided in two local tasks: road lane following and road intersection maneuvers. At this section, both navigation will be presented using a routing table, as the one of the Section III, to solve and improve the navigation problems.

A. Global navigation management

Given the routing table from the q_{init} position to q_{goal} position, the role of navigation management is to determine which local navigation must be performed. For that, it is necessary to know the critical points around the vehicle during its movement. If the robot is between two critical points, the local navigation "Road lane Following" is performed. The information about number of lanes and speed limit from the last node is used to improve this local task (as explained in the Section IV-B bellow). When the robot is near to a critical point, the information about angles and directions are extracted from the table and used to solve the local navigation task as "Road intersection maneuver" (as explained in the Section IV-C bellow).

B. Local navigation: road lane following

There are several approaches based on road lane following [6], [7], [8], which are able to control the vehicle without accurate position information. These approaches require a linear velocity setpoint for the control law, as well as a good perception system to detect the road lane marks. With the previous knowledge of the road speed limits and the number of lane, the speed setpoint can be adjusted and the lane detection algorithms optimized for each road. In addition, with the number of lanes it is possible to reduce the number of false positives in the final result.

This work will apply the road lane following algorithm proposed by [9], called Image Based Dynamic Window Approach (IDWA), to accomplish the current local navigation task. The IDWA was based on the integration of a line following Visual Servoing equations, adapted to the road lane following problem, in a Dynamic Window Approach for reactive obstacle avoidance. Details of its implementation and validation are in [9]. Beyond the advantages above, this method can use the number of lanes to restrict the robot movement during an obstacle avoidance.

C. Local navigation: road intersection manoeuvres

The approaches based on "Road lane Following" were not developed to deal with road intersection maneuvers, thus requiring different methods to accomplish this task. Our approach intends to complement the vehicle control proposed by [9] with an algorithm that will artificially change the environment information contained at the local occupancy

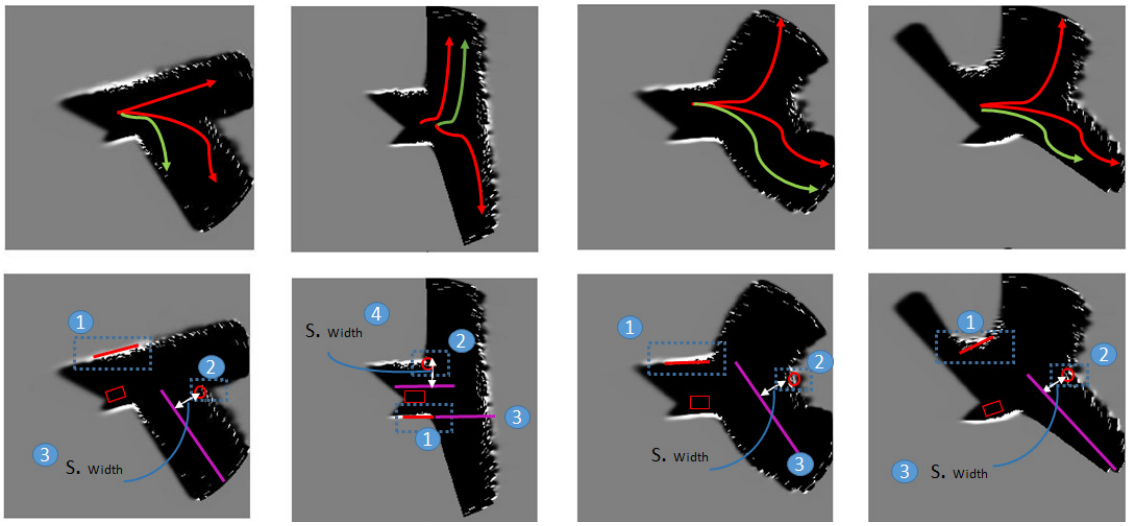

Fig. 4. Application of local navigation task: road intersection maneuver. The intersections shown correspond to the critical points *1,3,4* and *5* of the Figure 3.

grid [13]. This algorithm is based on a routing table which contains the geometry of intersections making possible to previously know the directions that the robot should take and the approximated angle between the current road and the next candidate road.

Considering the previous "Road Lane Following" approach, it is possible to force the vehicle to take right directions at intersections limiting the navigable area. The Figure 4 illustrates the technique proposed, the map shows the path to be crossed by the robot considering the points where exist more than one path to follow. As the vehicle gets closer to these points, it uses the geometry information (given by the routing table) to determine the pink line in the middle of the free space. This line is considered as an obstacle by the perception limiting the next decisions. The arrows represent all the possible decisions but only the green ones are corrects.

To adapt the occupancy grid with virtual obstacles, different types of critical points must be considered: right turn, left turn and roundabout. Each type requires a different algorithm to calculate the pink line (considered as obstacle) given that the obstacle geometry around the robot changes according to the critical point. Once calculated, the pink line is actualized with robot movement.

1) Right turn: When the vehicle gets closer to a critical point where it must turn right, the pink line is calculated following three steps, as the first critical point in the Figure 4 shows. The first step corresponds to searching the points of the occupancy grid above the vehicle, the blue dotted box represents the area where the points are sought and the red line shows the set of points found. From this set of points, it is possible to calculate the window inclination around the robot. The second step searches the point which corresponds to the street corner, the dotted box represents the search area and the red circle the point found. To find the pink line slope, the inclination of the occupancy grid (calculated at the first step) and the intersection angle (extracted from the routing table) are combined. If it is a two-way street, the street width must be considered.

2) Left turn: The steps to adapt the occupancy in a left turn are similar to the ones mentioned above, see the second critical point at Figure 4. The set of points below the vehicle is sought using a search area represented by a blue dotted box at the first step. This set is used for two reasons: estimate the window inclination and extend the red line giving rise to the pink line (represented at third step). The step two corresponds to finding the street corner which is used to localize the pink line represented in the step four.

3) Roundabout: The set of points representing the obstacles at the occupancy grid is geometrically different when the vehicle arrives and leaves the roundabout. In this way, there are two algorithms used to guide the robot in a roundabout. The first represents the arrival case, where the steps are the same as described in the section Section IV-C.1, only changing the way to perform the second step given that at this case there is no street corner. To leave the roundabout, the steps used to close the free space are equals to the ones described in Section IV-C.1.

V. Experimental Results

To demonstrate the application of digital maps for global navigation on autonomous vehicle, two experiments have been performed. The first experiment intends to create a global route planning at a real environment. The second experiment focuses on simulating the global navigation technique proposed using an environment created by Matlab.

A. Global route planning

The experiment was conducted at the city of Compiègne, in France. The urban area was around the campus of Université de Technologie de Compiègne and included all the types of intersection: arrival point (P_a), departure point (P_d), arrival and departure point (P_{ad}) and roundabout point (P_r),

TABLE II

NAVIGATION INFORMATION TO THE PATH PRESENTED IN FIGURE 6

Node	Lon	Lat	P. a	P. l	P. al	P. r	Angles (degrees)			Directions			Speed (km/h)	N. lanes
1	2,781	49,376	0	1	0	0	0,0	180,0	271,3	0	4	2	30	1
2	2,787	49,389	1	0	0	0	92,8	176,8	278,1	3	4	3	30	1
3	2,790	49,392	0	0	1	0	0,0	88,7	164,8	0	3	4	30	1
4	2,777	49,408	1	0	0	0	0,0	97,7	278,2	0	4	3	50	2
5	2,775	49,409	0	1	0	0	0,0	74,7	188,2	0	2	4	50	2
6	2,774	49,409	1	0	0	1	0,0	88,4	268,3	0	4	3	30	2
7	2,776	49,410	0	1	0	1	0,0	114,0	180,0	0	4	2	30	2

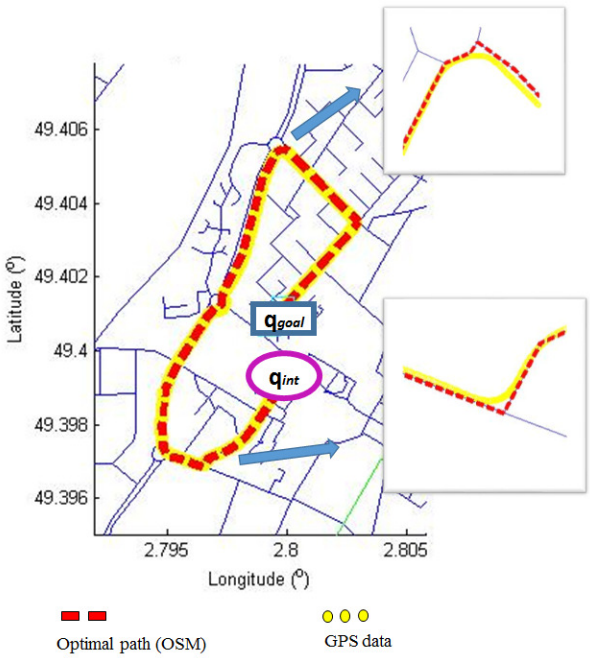

Fig. 5. Predefined path from a initial position q_{init} to final position q_{goal}. The red dash line is the differential GPS data and the yellow line the data from the OpenStreetMap. The average error is around 6 to 10 meters.

as seen in Section III. Based on a predefined path, from the initial position q_{init} and to the final position q_{goal}, the algorithm based on OpenStreetMap (OSM) data was applied to extract high level information of this path (see Table II). The next step was driving the vehicle thought this path to extract the localization data generated by a differential GPS.

The Figure 5 compares the OSM data (red) with GPS data (yellow) during the predefined path. It shows the consistency of the data from the OSM database, comparing with the GPS data. The error was around 6 to 10 meters, due to the nature of the OSM data, provided by final users with low-cost GPS. For a global navigation application based on it, an error over than 10 meters must be considered to deal with critical points.

To generate the routing table presented at Table II only a part of the path was used, as presented at Figure 6. The critical points are ordered from 1 to 7, where the traffic directions are presented by arrows with two different colors: yellow for the right way and blue is to the wrong way.

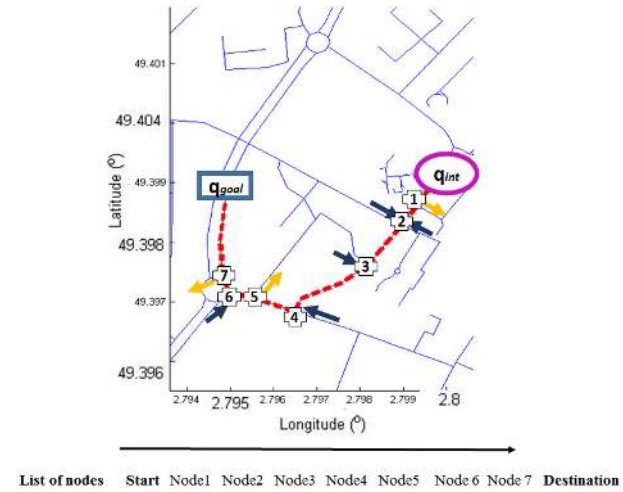

Fig. 6. Part of the predefined path of Figure 5 with the list of critical points ordered from 1 to 7. The arrows represent the traffic directions of exiting (yellow) and arriving (blue) for each node.

B. Global navigation

In the second experiment, the global navigation technique based on a routing table was simulated with Matlab. 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 and a laser sensor as described in [9], where more details about the simulation environment can be found. The detected obstacles were stored in an occupancy grid [13] by a bidimensional Gaussian model, locally constructed around the robot. The relative movement of the robot frame updates the grid information, using its proprioceptive data, like odometry, velocities and steering angle, which is enough for the simulation purposes and low speed experiments.

The Figure 7 presents the simulation environment considering the two cases of local navigation: road lane following and road intersection maneuver. At each local navigation task, the information extracted from the routing table are shown according to the nodes around. To visualize the changes at local occupancy grid according to the distance to the critical points, the Figure 8 presents some steps during the road intersection maneuvers. The distance considered to change the local navigation tasks was 15 meters, due to the average error calculated in the Subsection V-A. The localization system simulates a low-cost GPS with a Gaussian error

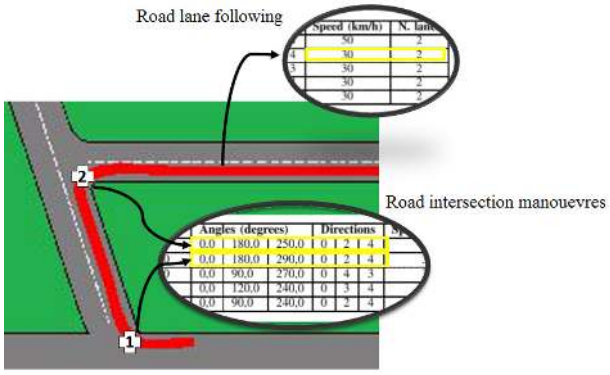

Fig. 7. Simulation of a global navigation emphasizing the two local tasks and their respective information extracted from the Table II.

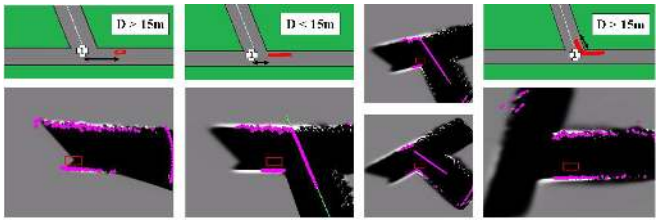

Fig. 8. Steps during the road intersection maneuvers considering limit distance of 15 meters.

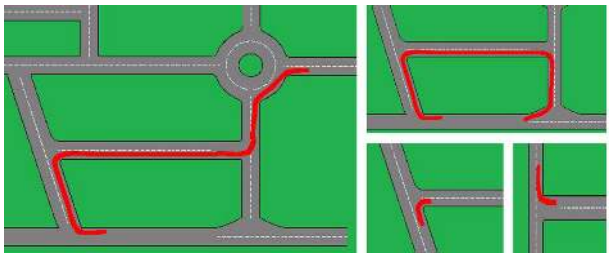

Fig. 9. Global navigation simulation for different paths.

of 6 meters. Finally, the Figure 9 shows the global navigation for different paths.

VI. Conclusions and Future Works

This paper proposes a global navigation approach independent of an accurate localization system and based on digital map data. The information extracted from this digital map had two main objectives: to solve the navigation task at a critical point and to complement the techniques of road following. The difference between the OpenStreetMap and the real data from differential GPS was close enough to guarantee the viability of the current approach. According to the results, the car-like robot was able to solve the global navigation task with a low-cost localization system, given that it was validated in several circuits conditions with different critical points. Future works will consider real environments for complete validation of the navigation task. In this case, more robust techniques must be applied to change artificially the environment in the occupancy grid and detect the lane to follow, with a low-cost localization system. Also, to improve the technique proposed, the deterministic analysis used in this work must be replaced by a probabilistic analysis taking different event sequences into account.

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

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

[4] A. Schindler, "Vehicle self-localization with high-precision digital maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 141–146.

[5] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, Sept 2010, pp. 1645–1650.

[6] 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, Jun 1992.

[7] 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.

[8] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino, L. Bianco, and A. Piazzi, "The argo autonomous vehicle's vision and control systems," *International Journal of Intelligent Control and Systems*, pp. 409–441, 1999.

[9] 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.

[10] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohb, C. Dupont, L. erik Jendrossek, C. Koelen, C. Markey, C. Rummel, J. V. Niekerk, E. Jensen, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "The robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, pp. 661–692, 2006.

[11] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," in *Autonomic and Autonomous Systems, 2008. ICAS 2008. Fourth International Conference on*, March 2008, pp. 94–99.

[12] A. C. Victorino, P. Rives, and J.-J. Borrelly, "Safe navigation for indoor mobile robots. part I: a sensor-based navigation framework," *International Journal of Robotics Research*, 2003.

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

[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] 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.

[16] 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.

[17] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, Oct 2008.

[18] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, July 1968.