

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/323760442>

Trajectory Tracking and VFH Obstacle Avoidance for Differential Drive Mobile Robot

Conference Paper · March 2018

CITATIONS

0

READS

257

3 authors:



Salah Eddine Ghamri
University of Batna 2
3 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



N. Slimane
University of Batna 1
19 PUBLICATIONS 16 CITATIONS

[SEE PROFILE](#)



Fateh Nezzar
1 PUBLICATION 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Magistère [View project](#)



On the visual navigation [View project](#)

Trajectory Tracking and VFH Obstacle Avoidance for Differential Drive Mobile Robot

Salah Eddine Ghamri

Department of Electronics

University of Batna 2

Batna, Algeria

Email: s.ghamri@univ-batna2.dz

Noureddine Slimane

Advanced Electronics Laboratory

University of Batna 2

Batna, Algeria

Email: n.slimane@univ-batna2.dz

Fateh Nezzar

Department of Electronics

University of Batna 2

Batna, Algeria

Email: f.nezzar@univ-batna2.dz

Abstract—A concrete incorporation of mobile robots in more industrial applications depends mainly on their ability to navigate safely in industrial environments while fulfilling the assigned tasks. One of the important tasks in wheeled mobile robotics is trajectory tracking, which adds temporal constraints on the robot navigation. The employment of trajectory tracking strategies with obstacle avoidance capabilities is essential to avoid time waste and inefficiency in mobile robots industrial integration. In this paper, we present a navigation algorithm intended for differential drive mobile robots, that combines trajectory tracking with obstacle avoidance. Obstacle detection is accomplished using sonar range sensors and the navigation is found on kinematic control and vector field histogram approach. Demonstration with simulations in V-rep environment is presented where Pioneer-3DX successfully avoided obstacles while tracking a trajectory.

Keywords—Wheeled Mobile Robot, Trajectory Tracking , VFH Obstacle Avoidance, Collision Avoidance, Pioneer-3DX.

I. INTRODUCTION

Wheeled mobile robots (WMR) are being exploited more and more in industrial applications like transportation, warehousing, exploration and many more uses. This increasing use is driven by the great sense of automation robots bring to ordinary navigation routines, without any human intervention. However, navigation in partially known or unknown environments still not fully mastered, which hinders a tangible utilization of wheeled mobile robotics in industrial projects.

The primary difficulty facing WMRs in industrial environments is the presence of obstacles, which if not avoided, they cause damages and prevent tasks execution. A workaround to this problem is to generate an obstacle free paths using global path planning techniques[1], [2]. Only, the utilization of such techniques requires full knowledge of the working environment in order for them to select a path free of obstacles. Even if such knowledge is afforded, it can be easily lost due an unexpected occurrence of obstacles after environment data acquisition and path selection phase. This makes them suitable for unchangeable environments, which is rarely found in industrial applications. A more rigorous solution can be found using map updating techniques like simultaneous localization and mapping (SLAM)[3]. Yet such procedures result in a huge computational load that may effect navigation speed and production costs.

Local planning approaches, on the other hand, provide more simplistic solution for obstacle avoidance in unknown and partially-known environments. In the literature we find many obstacle avoidance approaches classified as local path planning, probably the most known methods are artificial potential field (APF)[4] and its derivative approaches like the vector field forces (VFF)[5] and the vector field histogram (VFH)[6]. The key idea behind them, is to assume the robot's motion in a virtual field of forces, the target is an attractive pole for the robot and the obstacles are repulsive surfaces. They use only information surging from range sensors mounted on the WMR. Their considerable simple implementation makes them suitable for real-time applications. The vector field histogram was introduced as a real-time obstacle avoidance for fast vehicles. It came as an amelioration on the APF and the VFF approaches. It looks for sectors free of obstacles, in one dimensional polar histogram extracted from a histogram grid built using range sensors information and robot position.

An implementation of a navigation algorithm that uses solely VFH obstacle avoidance would be set only through go-goal motion. This relaxes restraints on the chosen path which in this case will be an arbitrary instead of optimal path. An answer for this obligation under the assumption of full environment knowledge is to use the VFH as low level navigation control while using another global planning technique in a higher control level[6]. This is the motive behind this work, in which we use the VFH obstacle avoidance in cooperation with a trajectory tracking assuming that trajectory has been generated by a higher level planner. Such combination will be beneficial in cases where the trajectory is roughly selected or while operation in changeable working environments. The reason is that VFH obstacle avoidance will correct the robot's path for short time intervals and that only when to avoid obstacles. This eventually adds more restrictions on the robot's path and forces the robot to respect temporal constraints and to approach the optimal chosen path.

This work will be presented in five sections as follows: In the first section we start by presenting the Pioneer-3DX robot's kinematic model. In the second and third sections, we will present and develop respectively the used trajectory tracking and the vector field histogram approach. A discussion on the proposed navigation algorithm will be presented in section

four. In the fifth section a demonstration by simulation is presented.

II. ROBOT'S KINEMATIC MODEL

The Pioneer-3DX is a non-holonomic differential drive mobile platform designed for research purposes[7]. In this work, a kinematic model is used to represent the Pioneer-3DX robot. To study the WMR's motion we represent its movement in a plane using two frames, one global O_{XY} called the world frame, and the other local $P_{X'Y'}$, mobile and attached to point P on the robot. As can be seen in Fig. 1 the point P lays on the center of axis crossing the frontal wheels of the robot. The angle ϕ represents the robot orientation, and

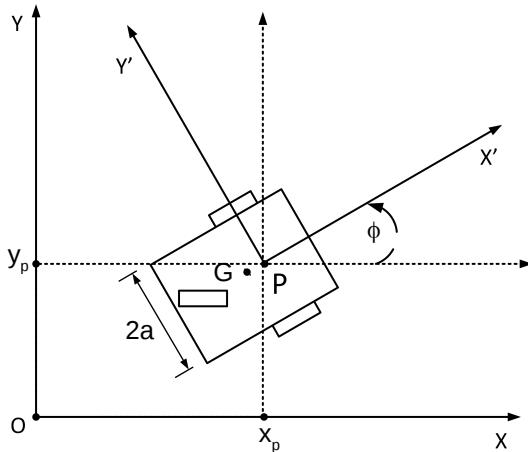


Fig. 1. The Pioneer-3DX Geometric Representation

the coordinates (x_p, y_p) of the point P in the global frame state the actual robot position in its working environment. In such configuration, the movement of a non-holonomic wheeled mobile robot is characterized by two non-holonomic constraints derived from the following assumptions[8]:

- No lateral movement is allowed in the robot's own frame $P_{X'Y'}$.
- Rolling of the robot's wheels is pure, which means wheels slippage and sliding is ignored.

These assumptions lead to the following non-holonomic constraints:

$$-\dot{x}_p \sin \phi + \dot{y}_p \cos \phi = 0 \quad (1)$$

$$\begin{cases} \dot{x}_p \cos \phi + \dot{y}_p \sin \phi + a\dot{\phi} = r\dot{\theta}_r \\ \dot{x}_p \cos \phi + \dot{y}_p \sin \phi - a\dot{\phi} = r\dot{\theta}_l \end{cases} \quad (2)$$

Where a is the distance between the wheels and the point P , Fig. 1, and r is the radius of the wheels. By adding and subtracting equations in (2) and using equation (1) we get the kinematic model of the robot represented by the following equations:

$$\begin{cases} \dot{x}_p = \frac{r}{2}(\dot{\theta}_r \cos \phi + \dot{\theta}_l \cos \phi) \\ \dot{y}_p = \frac{r}{2}(\dot{\theta}_r \sin \phi + \dot{\theta}_l \sin \phi) \\ \dot{\phi} = \frac{r}{2a}(\dot{\theta}_r - \dot{\theta}_l) \end{cases} \quad (3)$$

Expressing these equations in a matrix form, we get the following system:

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{r}{2a} & -\frac{r}{2a} \end{bmatrix} * \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (4)$$

This models takes $[\dot{\theta}_r \dot{\theta}_l]^T$ as input and gives the output vector $[\dot{x}_p \dot{y}_p \dot{\phi}]^T$. For control purposes, it is more convenient to replace the kinematic model in (4) with a simpler model, the unicycle model, where the robot is considered as a point moving in the plane. the unicycle model takes the linear and angular velocities v, ω of the point P as input, and it can be derived from (4) using the following mapping:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{a}{r} \\ \frac{1}{r} & -\frac{a}{r} \end{bmatrix} * \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5)$$

Hence the kinematic model of the robot expressed in terms of velocities vector $[v \omega]^T$ is the following:

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (6)$$

III. TRAJECTORY TRACKING

Trajectory tracking is one of the well studied mobile robotics navigation tasks. Its main goal is to drive the mobile robot to follow a given trajectory. A trajectory differs from a path with a temporal constraints added to it, which makes control objective not only to minimize the distance between the robot and the path, but to respect a proper timing as well. We define the mobile robot pose in the global frame as :

$$q = [x_p \ y_p \ \phi]^T \quad (7)$$

and the desired pose is a vector of coordinates as function of time presented :

$$q_d = [x_d(t) \ y_d(t) \ \phi_d(t)]^T \quad (8)$$

For realization purposes and since the robot has movement constraints, the trajectory must obey to the same kinematic and non-holonomic constraints. Thus the generated trajectory must be built based on the following kinematic relations:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\phi}_p \end{bmatrix} = \begin{bmatrix} v_d \cos \phi_d \\ v_d \sin \phi_d \\ \omega_d \end{bmatrix} \quad (9)$$

$$\dot{x}_d \sin \phi_d = \dot{y}_d \cos \phi_d \quad (10)$$

We define the tracking error vector as :

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\phi} \end{bmatrix} = \begin{bmatrix} x_d(t) - x_p \\ y_d(t) - y_p \\ \phi_d(t) - \phi \end{bmatrix} \quad (11)$$

where $\tilde{x}, \tilde{y}, \tilde{\phi}$ are the tracking errors in the global frame. We use the inverse of the universal rotation matrix to map the tracking errors into the local frame of the robot as follows:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\phi \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\phi} \end{bmatrix} \quad (12)$$

With e_x, e_y, e_ϕ are the tracking errors in the local frame of the robot. By differentiating equation (12) and taking into account the equations (9) and (10) we get the dynamics of tracking errors expressed by the following kinematic model:

$$\begin{cases} \dot{e}_x = v_d \cos \phi_d - v - e_y \omega \\ \dot{e}_y = v_d \sin \phi_d - e_x \omega \\ \dot{e}_\phi = \omega_d - \omega \end{cases} \quad (13)$$

where v, ω are respectively the robot's linear and angular velocities. The right choice of v and ω in (13) will ensure that e_p will tend to zero. By using the Lyapunov direct method and by choosing the following candidate Lyapunov function[9]:

$$f(e_p) = \frac{1}{2}(e_x^2 + e_y^2) + \frac{1}{K_y}(1 - \cos e_\phi) \quad (14)$$

with $K_y > 0$ and the following derivative :

$$\dot{f}(e_p) = (-v + v_d \cos e_\phi)e_x + \left(\frac{\omega}{-K_y} + v_d e_y + \frac{\omega_d}{K_y}\right) \sin e_\phi \quad (15)$$

we choose the following velocities so that $\dot{f}(e_p) < 0$ and the Lyapunov conditions are all satisfied :

$$v = K_x e_x + v_d \cos e_\phi \quad (16)$$

$$\omega = K_\phi \sin e_\phi + K_y v_d e_y + \omega_d \quad (17)$$

where K_x, K_y and K_ϕ are tuning gains. By substituting the chosen velocities in equation (15) we get :

$$\dot{f}(e_p) = -(K_x e_x^2 + \frac{K_\phi}{K_y} \sin^2 e_\phi) \quad (18)$$

It is clear that $\dot{f}(e_p)$ is always negative for $K_x > 0, K_y > 0$ and $K_\phi > 0$ which satisfies the Lyapunov stability conditions. Assuming that the chosen velocities in equations (16), (17) are attainable, and by taking them as control inputs for the robot kinematic model in (6), will ensure that the tracking errors will converge to a region near zero and the control will be asymptotically stable in the sense of Lyapunov[8]. Thus we add the notation c and we write the control law for the trajectory tracking as follows:

$$v_c = K_x e_x + v_d \cos e_\phi \quad (19)$$

$$\omega_c = K_\phi \sin e_\phi + K_y v_d e_y + \omega_d \quad (20)$$

IV. OBSTACLE AVOIDANCE USING VFH

The Vector Field Histogram, abbreviated VFH, is a local obstacle avoidance, which came as a development on the vector field force and potential field obstacle avoidance techniques. The VFH can determine with great certainty the angle that the robot should be headed with in order to take a path clear of

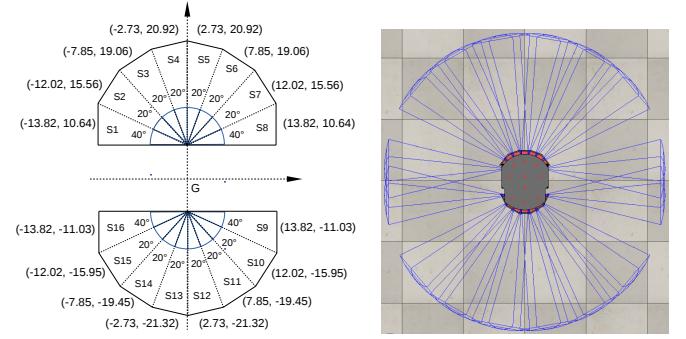


Fig. 2. (a) Arrangement of Sonar Sensors on Pioneer-3DX, (b) Coverage Zone of Pioneer-3DX in V-rep.

obstacles. In this section we will present the steps to follow in order to implement the VFH algorithm.

A. Obstacles Detection and Sensors Configuration

In this work, we used the sonar range sensors to detect obstacles. The sonar range sensors are very efficient sensors for proximity and obstacle detection. Their principle is the emission of a ultra-sonic waves and the reception of their reflection on obstacles, the distance to the obstacle is calculated using the waves traveling time. The Pioneer 3-DX has a ring of 16 sonar sensors divided on two, frontal and rear. This configuration ensures a coverage of 360° zone and a radius of 1 m around the robot. The arrangement and zone coverage of all sonar sensors mounted on the Pioneer-3DX is presented in Fig. 2 (extracted from V-rep software[10]).

B. Building the Histogram Grid

The first step in VFH implementation is to retrieve a description of the robot's environment. In this step, a Cartesian two dimensional histogram grid C is created which contains information streamed from the range sensors (In this work, C is chosen of size 51×51 and of resolution 0.1 meter/cell). The histogram grid is similar to the certainty grids, each grid cell $C[i, j]$ contains c_{ij} value which represents the confidence of an obstacle existence in the location of coordinates (i, j) . The grid is filled using the distances measured by the range sensors and that is by updating, each reading session, a single occupied cell for each sensor. This updating technique minimizes the computational cost for a real-time application and it is the only difference between certainty grids and the histogram grid. The cell, which to be updated for each sensor, corresponds to the measured distance d and lays on the acoustic axis of the sensor. Knowing the robot's position and orientation in the grid and using the arrangement of sensors presented in Fig. 2 (a), the coordinates of the concerned cell for each sensor is found using the dead reckoning and the universal rotation matrix, as follows:

$$x_{grid} = x_G + x_s \cos(\pi/2 - \phi) + y_s \sin(\pi/2 - \phi) \quad (21)$$

$$y_{grid} = y_G - x_s \sin(\pi/2 - \phi) + y_s \cos(\pi/2 - \phi) \quad (22)$$

while x_{grid}, y_{grid} are the coordinates of the detected cell for each sensor in the grid, x_G, y_G are the coordinates of the

center of mass G of the robot, ϕ is the robot's orientation in the world frame and x_s, y_s are coordinates of the detected cell in the frame presented in Fig. 2, by projection they are calculated as follows:

$$x_s = x_{sensor} + d \cos(ori_{sensor}) \quad (23)$$

$$y_s = y_{sensor} + d \sin(ori_{sensor}) \quad (24)$$

with x_{sensor} and y_{sensor} are the position coordinates of each sensor and ori_{sensor} is its orientation all referred to point G (Fig. 2).

C. Building the Polar Histogram

In second step, a window C^* of a fixed orientation, which its center is attached to the robot and moves along its movement, is selected (In this work, C^* is set of size 33×33). It is called the active region and it overlays the vicinity of the robot's momentary location in the histogram grid C . Each active region cell is then treated as an obstacle vector, its orientation β is the direction to the center of the active region (point G):

$$\beta_{i,j} = \tan^{-1}\left(\frac{y_j - y_0}{x_i - x_0}\right) \quad (25)$$

and its magnitude is calculated as follows :

$$m_{i,j} = (c_{i,j}^*)^2(a - bd_{i,j}) \quad (26)$$

where a, b are positive constants, $c_{i,j}^*$ is the certainty value of the active cell (i, j) , $d_{i,j}$ is the distance between the active cell and the center of the active region, $m_{i,j}$ is the magnitude of the obstacle vector at the cell (i, j) , (x_0, y_0) are the coordinates of the center of the active region, (x_i, y_j) are the coordinates of the active cell (i, j) , $\beta_{i,j}$ is the direction from the active cell (i, j) to the center of the active region.

At this stage, one polar histogram H is constructed by dividing the active region into n number of sectors k , where n is determined using an arbitrary angular resolution α (for our simulation $\alpha = 5^\circ$):

$$n = 360^\circ/\alpha \quad (27)$$

while each active region cell is related to a sector k using the equation (25) and the following mapping:

$$k = INT(\beta_{i,j}/\alpha) \quad (28)$$

For each sector k a polar density of obstacle is calculated by:

$$h_k = \sum_{i,j} m_{i,j} \quad (29)$$

more smoothing of H can be applied to avoid errors due the ragged nature of the histogram. the following smoothing functions is applied:

$$h'_k = \frac{h_{k-l} + 2h_{k-l+1} + \dots + lh_k + \dots + 2h_{k+l-1} + h_{k+l}}{2l + 1} \quad (30)$$

Where $l = 5$ gave a satisfactory results.

D. Decision and Control

The extracted smoothed polar histogram is divided to n sectors where each sector represents an angle, while the obstacle density associated to it reveals whether the angle is free of obstacle or cluttered. Thus the decision making in the VFH comes as a steering angle selection, meaning the selection of a sector with low obstacle density. A sector is considered free based on the polar density assigned to it and to its adjacent sectors and the target angle. For that matter, the sector selection is performed in three stages:

- The first, candidate valleys selection, the candidate valley is any consecutive sectors in the smoothed histogram with obstacle density lower than a certain threshold. The selected candidate valleys can be either wide or narrow based on the number of the consecutive sectors.
- As a second stage, a candidate valley with a proper width must be selected based on the size of the robot and its kinematic constraints. For that matter, any candidate valley with less than S_{max} consecutive sectors is eliminated, (For the Pioneer-3DX $S_{max} = 18$ is deemed to be sufficient).
- In the last stage of selection, the free valley , which to be selected is the candidate valley with the center sector the closest to the target angle sector, the target angle is the orientation angle that directly leads the robot to the given target point. The steering angle ϕ_{VFH} then is the angle of the center sector of the selected free valley and it is extracted by multiplying with the angular resolution α .

At this stage, we use a proportional controller to feed the selected steering angle to the robot in the form of an angular velocity :

$$\omega_{VFH} = K_p(\phi_{VFH} - \phi) \quad (31)$$

The difference between the selected steering angle ϕ_{VFH} and the robot orientation ϕ is the steering error. As regards the linear velocity, it can be fixed in arbitrary fashion or can be limited and set to vary according the obstacles presence. The mapping of equation (5) is then used to determine the wheels angular velocities, which steers the robot to the selected angle ϕ_{VFH} .

V. NAVIGATION ALGORITHM

We propose a navigation algorithm that compromises the use of both navigation techniques, trajectory tracking and obstacle avoidance using VFH. The default mode is the trajectory tracking but whenever an obstacle is detected in front of the robot, obstacle avoidance mode is enabled. The key element in the choice of the active mode is the vector field histogram, therefore, the proposed navigation algorithm starts firstly by setting up the VFH environment. During initialization, the software retrieves robot's position and orientation along with the sensors readings and builds the histogram grid and the smoothed polar histogram H' . Since the decision of the steering angle in VFH depends on the target location, the trajectory coordinates at each step will be considered as target

point coordinates. At this stage, the algorithm achieves the right mode choice based on three important sectors in the smoothed polar histogram H' :

- *sector1* corresponds to the retrieved momentary robot orientation.
- *sector2* corresponds to the direction at which temporally rest the target point.
- *sector3* is the sector located in the middle of the shortest distance between *sector1* and *sector2*.

The algorithm sets the robot on obstacle avoidance mode whenever an obstacle is detected ahead, this is interpreted as a high obstacle density in the valley centered by *sector1*. Otherwise, it will continue in trajectory tracking procedure as studied in the first section by calculating tracking errors and applying the kinematic control law afterwards. A flowchart illustrating this algorithm is depicted in Fig. 3.

In the first conditional statement, which is portrayed by $H'(5, \text{sector1}) \leq \text{threshold1}$, a valley of 5 sectors width, among which *sector1* is the center, is evaluated comparing an arbitrary threshold that represents the tolerated obstacle density and the admissible approaching distance to the ob-

stacles as well. If the obstacle density for only one sector of the valley is found to be higher than the *threshold1*, the obstacle avoidance mode will be enabled by assigning 0 to the variable *Mode*. The width of the valley is chosen to cover a rectangular region in front of the robot conforming the robot's dimensions and sensors range. The second conditional verification, $H'(28, \text{sector2}) \leq \text{threshold2}$, which is performed in the obstacle avoidance mode, guarantees the robot's return to the trajectory tracking mode whenever the obstacle density is low for all sectors in the valley centered by *sector2* of width of 28 sectors. This means that the path from the robot to the target is clear of obstacles. Safety measures can be seen through the choice of the valley width, which covers a region of 140 angle and the choice of the *threshold2*, which must be tuned to guarantee trajectory tracking activation only when the path is cleared. The third conditional statement, $H'(\text{mid}, \text{sector3}) \leq \text{threshold3}$, belongs to the obstacle avoidance mode and it works on driving the robot gradually closer to the target point while passing an obstacle. It concerns the target angle choice for the VFH steering angle selection. In the obstacle avoidance maneuver, the robot must continue traveling with the same orientation of the trajectory. However, when the middle valley between *sector1* and *sector2* shows a low obstacle density, the robot must navigate closer to the target point, this is performed by changing the target angle in the decision module of VFH to the target direction from the robot's point *G*. This will eventually guarantee that the robot gradually approaches the trajectory while maneuvering an obstacle. *mid* is the variable width of the middle valley between *sector1* and *sector2* and its value depends only on the location of *sector1* and *sector2* in the polar histogram H' .

The linear velocity under the obstacle avoidance mode is fixed and chosen in a fashion to keep up with the trajectory linear velocity, which will amend the high correction speed when trajectory tracking is activated.

VI. SIMULATION

The simulation is conducted using V-rep environment which is a robotic simulation software that facilitate simulation of various scenarios and testing robotic software. Fig. 4 shows the V-rep simulation environment. We have prepared two simulation scenes including the built-in Pioneer-3DX model and a combination of static primitive shapes with size of 0.5 meters for the first demonstration and 0.35 meters for the second, they serve as obstacles. The linear velocity of both trajectories is fixed to 0.1 m/s and the robot starting pose is [0, 0, π/4] for the first demonstration and [1, 3.10, π/2] for the second.

As regards the software, the program is written in Python language using Numpy package for scientific computing and Matplotlib library for plotting purposes. The algorithm parameters in the demonstration are chosen as follows:

Dimension parameters for the Pioneer-3DX are extracted from V-rep model and they have the following values $a = 0.1655$ m and $r = 0.0975$ m. For the kinematic controller in (19)

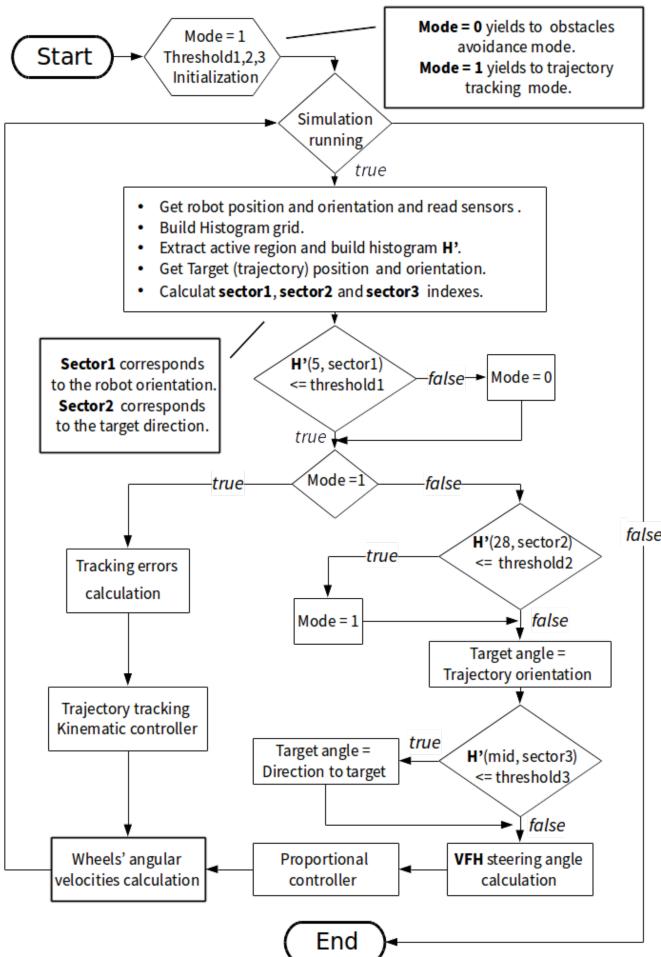


Fig. 3. Flowchart of the Navigation Algorithm.

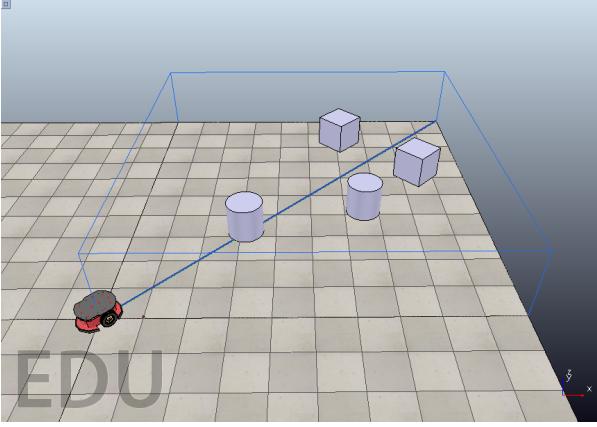


Fig. 4. V-rep simulation environment

and (20), the following gains give the best tracking results $k_x = 0.9$, $k_y = 2.0$, $k_\phi = 0.9$ for the first demonstration and $k_x = 0.9$, $k_y = 2.0$, $k_\phi = 1.5$ for the second. As for the VFH mode, the cells of the histogram grid are updated when an obstacle is detected by a certainty value of 0.36, and the threshold for the sector selection in the histogram is chosen 80. The gain of the proportional controller is chosen of $K_p = 0.25$ and the linear velocity for the robot is fixed at 0.11 m/s for both demonstrations. Concerning the proposed algorithm, *threshold1*, *threshold2* and *threshold3* are tuned for each demonstration. The results of the simulations are presented in Fig. 5 and Fig. 6 where scale is in *centimeters*.

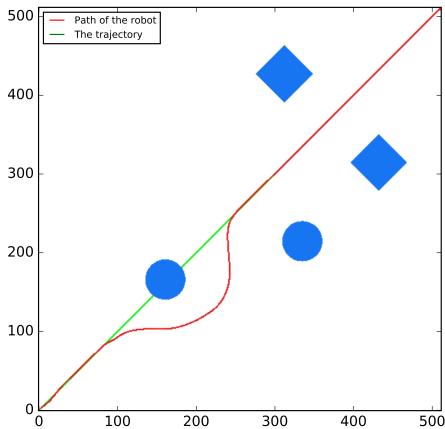


Fig. 5. Line Trajectory (threshold1 = 12, threshold2 = 12, threshold3 = 20)

VII. DISCUSSION AND CONCLUSION

The above simulation results show that the generated path is smooth, disregarding the shape of obstacles. Obstacle avoidance mode is activated only when an obstacle obstructs the trajectory, otherwise trajectory tracking will continue. The nearness to the obstacles during avoidance maneuver is governed by VFH threshold and the navigation algorithm

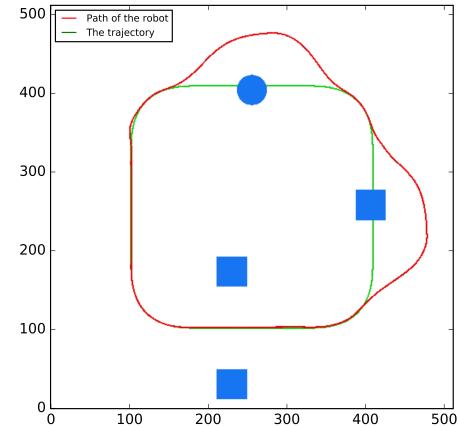


Fig. 6. Rectangular Trajectory (threshold1 = 6, threshold2 = 6, threshold3 = 6)

thresholds *threshold1*, *threshold2* and *threshold3*. In a more cluttered environments a tuning for all the thresholds is necessary, this is inherited from the VFH approach itself.

In summary, we have demonstrated by this work a successful integration of trajectory tracking method with obstacle avoidance using VFH approach. This binding can be viewed either as equipping the trajectory tracking navigation with obstacle avoidance capabilities, or as putting more restrictions of the path generated by the VFH, which in this case can be very beneficial if an optimal path was generated in a higher control level. the simulation results support the proposed algorithm and the outcome is adding a sense of safety to the robot's navigation in changeable environments.

REFERENCES

- [1] M. S. Ganeshmurthy and G. R. Suresh, "Path planning algorithm for autonomous mobile robot in dynamic environment," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, March 2015, pp. 1–6.
- [2] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [3] S. Thrun and J. J. Leonard, "Simultaneous localization and mapping," in *Springer handbook of robotics*. Springer, 2008, pp. 871–889.
- [4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, Mar 1985, pp. 500–505.
- [5] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sept 1989.
- [6] ———, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun 1991.
- [7] Mobile robots — robotic control platforms and applications. [Online]. Available: <http://www.mobilerobots.com>
- [8] S. G. Tzafestas, *Introduction to mobile robot control*. Elsevier, 2013.
- [9] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings. IEEE International Conference on Robotics and Automation*, May 1990, pp. 384–389 vol.1.
- [10] Coppelia robotics v-rep: Create. compose. simulate. any robot. [Online]. Available: <http://www.coppeliarobotics.com/>