Autonomous mobile robot navigation based on an integrated environment representation designed in dynamic environments

Faten Cherni* and Maissa Boujelben

Control and Energy Management Laboratory (CEM Lab),
National Engineering School of Sfax,
University of Sfax, Tunisia
Email: fatin.cherni@gmail.com
Email: boujelben.maissa@gmail.com
*Corresponding author

Lotfi Jaiem

Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM),
University of Montpellier, France
Email: jaiem@lirmm.fr

Yassine Boutereaa, Chokri Rekik and Nabil Derbel

Control and Energy Management Laboratory (CEM Lab),
National Engineering School of Sfax,
University of Sfax, Tunisia
Email: yassineboutereaa@gmail.com
Email: chokri.rekik@enis.rnu.tn
Email: n.derbel@enis.rnu.tn

Abstract: The collision avoidance concept is widely applied for developing and designing in autonomous robotic applications. In an unknown environment, the challenge in mobile robot navigation or path planning is to find the path from the starting point to the target avoiding obstacles. This paper investigates the mobile robot navigation based on an integrated environment representation of the environment. It presents a developed algorithm for solving the obstacle avoidance problem in dynamic environment. The proposed algorithm ensures that the mobile robot whenever senses the obstacle, changes his direction without being collided and moves smoothly to the designed target when the path is free. The simulation results and experimental ones are presented to show performances and the effectiveness of the presented approach. These results illustrate that the developed algorithm can be well applied in the mobile robot navigation.

Keywords: obstacle avoidance; mobile robot; path planning; unknown dynamic environment; integrated environment representation.

Biographical notes: Faten Cherni received her Electrical Engineering degree from National Engineering School of Sfax, Tunisia in 2013. She is working in Control and Energy Management Laboratory, University of Sfax, as a PhD student doing research in path planning and navigation in mobile robots. Her research interests include mobile robot control and fuzzy logic.

Maissa Boujelben received her Engineering Diploma in 2010, Master degree in Automatic Control in 2011, and PhD in Electrical Engineering in 2016, all from the Ecole Nationale d’Ingénieurs de Sfax (ENIS). She is a member in Control and Energy Management Lab (CEM Lab) in ENIS. Her research interests include mobile robot control, fuzzy logic, neural networks and genetic algorithms.

Lotfi Jaiem received his Engineering degree in Mechatronics in 2012 from the National School of Engineers of Sousse (Tunisia) and his Master degree in Robotics in 2013 from the University of Blaise Pascal (France). He is currently working towards his PhD in Robotics at the University of Montpellier, Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM), France. His research interests include autonomous mobile robot.

Yassine Boutereaa received his National Engineering degree in Electrical Engineering from the National School of Engineers of Sfax in June 2006, and Master of Science degree in Control and Computer Science in 2007. He received his PhD in Electrical Engineering from the University of Sfax and the University of Orleans in 2011. Currently, he is an Assistant Professor in the Higher Institute of Biotechnology of Sfax. His interest concerns robotic control, medical robots, mechatronics design of robots, cooperative control and synchronisation of Lagrangian systems, embedded systems in robotics and real-time implementation.

Chokri Rekik received his Engineering Diploma in 1999, the Diplome d’Etudes Approfondies in Automatic Control in 2001 and a PhD degree in 2006, all from the Ecole Nationale d’Ingénieurs de Sfax (ENIS). He has been an Assistant Professor in the Electrical Engineering Department of the Ecole Nationale d’Ingénieurs de Sfax in 2006. Currently, he is a Full Professor of Automatic Control at the Ecole Nationale d’Ingénieurs de Sfax and carrying his research activities in Control and Energy Management Lab (CEM Lab) in ENIS. His research interests are advanced control of complex nonlinear systems, and robotic control.

Nabil Derbel received his Engineering Diploma from the Ecole Nationale d’Ingénieurs de Sfax in 1986, and Diplôme d’Etudes Approfondies in Automatic Control from the Institut National des Sciences Appliquées de Toulouse in 1986, his Doctorat d’Université degree from the Laboratoire d’Automatique et d’Analyse des Systèmes de Toulouse in 1989 and his Doctorat d’Etat degree from the Ecole Nationale d’Ingénieurs de Tunis. He joined the Tunisian University in 1989, where he held different positions involved in research and education. Currently, he is a Full Professor of Automatic Control at the Ecole Nationale d’Ingénieurs de Sfax. His current interests include: optimal control, complex systems, fuzzy logic, neural networks and genetic algorithms. He is the author and the co-author of more than 40 papers published.
in international journals and of more than 250 papers published in international conferences. He is a member in Control and Energy Management Lab (CEM Lab) in ENIS.

This paper is a revised and expanded version of a paper entitled ‘Autonomous mobile robot navigation algorithm for planning collision-free path designed in dynamic environments’ presented at the 2015 9th Jordanian International Electrical and Electronics Engineering Conference (JIEEEC), Amman, Jordan, 12–14 October 2015.

1 Introduction

Robot navigation still represents a challenge for several applications in the real world. The task of the mobile robot navigation is to find the path from the starting point to the goal avoiding collisions. Several researchers have worked in this field which is mainly about avoiding (static or dynamic) obstacles in known or unknown environments. Moreover, they have focussed on developing approaches allowing the robot to travel smoothly to the designed target without collision.

The entire process that enables a robot to memorise its environment and then move to reach a goal, can be divided into three interdependent phases: mapping, localisation and planning. These three phases can meet the three fundamental questions for navigation tasks: Where am I? Where are the other places compared to me? How can I reach my goal?

- Mapping is the phase which allows the construction of a map reflecting the spatial structure of the environment from different information collected by the robot.
- Such a map is available, the localisation is the way of determining the orientation and the position of the mobile robot in the map with respect to its surrounding. The robot needs to know the objects (target or obstacle) around it (Filliat and Mayer, 2003).
- Finally, knowing the environment map and the current position of the robot, planning is the phase that allows to decide the movement in order to reach a goal in the environment. Indeed, the mobile robot needs to find a collision free path between any two points (from its beginning to its end). To be able to find this path, the mobile robot should run an adequate path planning algorithm. Several research works, for path planning of mobile robots, have been proposed in the literature (Cai and Peng, 2002; Liang and Lee, 2015; Nishitani and Matsumura, 2015; Liu and Arimoto, 1991; Zhong et al., 2014).

Developed navigation approaches can be generally classified into local and global navigation.

In the global navigation, the environment surrounding the mobile robot is known and the path leading to the goal avoiding obstacles is selected.

Concerning the local navigation, the environment surrounding the mobile robot is unknown. The aim is to detect obstacles using sensors and to avoid the collision of obstacles close to the robot while ensuring a fluid and a reactive movement.
Several researchers have developed reactive methods based on sensor inputs. The three most known algorithms for local navigation are: the artificial potential field (APF) method (Khatib, 1985; Ge and Cui, 2002; Masehian and Amin-Naseri, 2004; Amada et al., 2009), the vector field histogram (VFH) method (Borenstein and Koren, 1991; Kim et al., 2001), and the dynamic windows approach (Fox et al., 1997; Brock and Khatib, 1999). Additionally, there are other approaches like curvature velocity method (Simmons, 1996), lane curvature method (Ko and Simmons, 1998) and the beam curvature method (Fernández et al., 2004), and other methods (Jamdagni and Patra, 2014; Park and Zhang, 2011).

Several classical approaches dedicated to static environments are extended to dynamic ones (Ge and Cui, 2002; Hsu et al., 2012; Fujimura and Samet, 2005). However, the problem of avoiding collisions in dynamic environment is much harder. Several works have been developed for dynamic environments like velocity obstacles (Fiorini and Shiller, 1998; Large et al., 2005), collision cones (Chakravarthy and Ghose, 1998), the rolling window method (Zhang and Xi, 2003), inevitable collision state (Fraichard and Asama, 2004), the prediction model for beam curvature method (Shi et al., 2010), and other methods (Jaradata et al., 2011; Yaonana and Yimin, 2011; Zhu and Hua, 2011; Zhong et al., 2014; Bis et al., 2012).

This paper is organised as follows. In the next section, the problem formulation is presented. Section 3 contains the navigation algorithm. The simulation results are given in Section 4. Section 5 presents experiments with a real robot Pioneer 3-DX. Finally, conclusions are given in Section 5.

2 Problem formulation

We consider a wheeled mobile robot which takes as an input the angular velocity. The kinematic model of the mobile robot is given by:

\[ \begin{align*}
\dot{X}_R &= V \cos \alpha_R \\
\dot{Y}_R &= V \sin \alpha_R \\
\dot{\alpha}_R &= \omega
\end{align*} \]  

(1)

where \((X_R, Y_R)\) are the robot’s Cartesian coordinates. The heading direction \(R\) is taken counter-clockwise from the OX-axis. \(V\) and \(\omega\) are the translational and rotational velocities, respectively. The robot mobile is moving with a constant speed \(V\) taking into account a maximum angular velocity equal to \(\alpha_{\text{max}}\) which translates the robot limits. The purpose of this paper is to produce a reliable and a smooth trajectory in a static and a dynamic environment and to guide the robot towards the target direction without hitting obstacles, taking into account physical constraints of the robot.

3 Navigation algorithm

Several navigation approaches can be founded in the literature. The basic idea of this work has been developed by Savkin and Wang (2014). In this section, we introduce the proposed algorithm dedicated to the robot path planning in presence of static and
Autonomous mobile robot navigation

Dynamic obstacles. The developed approach, based on an integrated environment representation, is efficient and very easy to implement.

In order to reach the objective, the designed strategy is working to solve several various issues such as: How the robot can detect and separate obstacles, how to privilege situations of obstacle avoidance to overcome, how to treat the situation with more than one obstacle... To this end, the proposed algorithm uses several assumptions to solve the matter mentioned above.

3.1 Assumption 1

The initial condition of the mobile robot satisfies \( \alpha_R(0) = \alpha_g \) which \( \alpha_g \in \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right] \) is the desired goal direction and assumed to be known to the robot and \((X_g, Y_g)\) are the goal Cartesian coordinates. Then:

\[
\alpha_g = \arctan\left( \frac{Y_g - Y_R}{X_g - X_R} \right)
\]

Let \((X_R(0), Y_R(0))\) the initial condition of the robot and let \(r\) the sampling rate.

3.2 Assumption 2

To facilitate the present work, we assume that all obstacles are circles. We define the disc \(C\) of the radius \(R\) centred at the point \(O\) that is ahead of the mobile robot’s position as shown in Figure 1. The geometric sense of the disc \(C\) choice is to ensure an efficient detection of obstacles. Indeed, the geometric shape covers the entire area in front of the robot. Actually, the developed obstacle avoidance approach computes the intersection points between virtual circle and real obstacles [Figure 2(a)]. In Figure 2(b), we have two intersection points \(N_1(X_{N1}, Y_{N1})\) and \(N_2\). The angle \(\theta_1\) is given by:

\[
\theta_1 = \arctan\left( \frac{Y_{N1} - Y_R}{X_{N1} - X_R} \right)
\]

Figure 1 Representation of the disc (see online version for colours)
The idea is to compute the angle that makes the disc $C$ with two intersection points. Based on the computation angle, a new direction $\gamma$ close to $\alpha_R$ will be provided. The objective behind the calculation of the new direction is to change the robot’s heading in order to avoid obstacles detected ahead. That remains now is to move towards the goal. To this end, the angular velocity, that guides the robot to the target direction without hitting obstacles, should be determined.

### 3.3 Assumption 3

In the following, the developed algorithm to determine the goal direction, is explained.

**Step 1** Let $\theta_T$ the set which contains all values of $\theta$ calculated by 3.

$$\theta_T = \{\theta_1, \theta_2, \ldots \theta_j, \ldots\}$$

where $j \in \{1, 2, \ldots, 2n\}$ and $n$ is the number of obstacles.

**Step 2** We note $I_{ad}$ the index of the angle that is closest to the robot current heading $\alpha_R$.

$$I_{ad} = \arg \min (|\theta_T - \alpha_R|)$$ \hspace{1cm} (4)

Moreover, we note:

$$S_1 = \alpha_R - \frac{\pi}{2}; S_2 = \theta_T (I_{ad} - 1)$$

$$S_2 = \alpha_R - \frac{\pi}{2}; S_4 = \theta_T (I_{ad} + 1)$$

$$S_3 = \theta_T (I_{ad})$$
Step 3 If the robot is in front of obstacles, there are four cases depending on the value of the index $I_{nd}$:

- If $I_{nd}$ is odd:
  - Case 1 If $I_{nd} = 1$ then $\varphi = S_1$
  - Case 2 If $I_{nd} \neq 1$ then $\varphi = S_2$.

- If $I_{nd}$ is even:
  - Case 3 If $I_{nd} = 2n$ then $\varphi = S_1$
  - Case 4 If $I_{nd} \neq 2n$ then $\varphi = S_2$.

Step 4 We compute the new direction:

$$\gamma = \frac{S_k + \varphi}{2}$$  \hspace{1cm} (5)

The direction $\gamma$ represents the middle of the interval closest to $\alpha_R$.

Step 5 We calculate the angular velocity $\omega$:

$$\omega = \begin{cases} 
\alpha_{\text{max}} \text{ sign}(\gamma - \alpha_R) & \text{if \ length(} \theta_{\gamma} \text{) \neq 0} \\
\alpha_{\text{max}} \text{ sign}(\alpha_R - \alpha_R) & \text{elsewhere}
\end{cases}$$  \hspace{1cm} (6)

where sign is the standard sign function.

Figure 3 illustrates the distribution of different intervals and the direction $\gamma$. For example, if $\alpha_R \in [\theta_3, \theta_4]$ and if:

- $\alpha_R$ is very close to $\theta_4$, so $I_{nd} = 4$ then $\varphi = \theta_5$ and $\gamma = \frac{\theta_5 + \theta_6}{2}$.

- $\alpha_R$ is very close to $\theta_3$, so $I_{nd} = 3$ then $\varphi = \theta_2$ and $\gamma = \frac{\theta_3 + \theta_2}{2}$.

Figure 3 Illustration of the intervals (see online version for colours)

The algorithm flowchart presented in Figure 4 explains how to determine the direction $\gamma$. 
4 Simulation results

To show the basic ability of the proposed navigation strategy, we assume, in a first simulation, an arbitrarily environment including static obstacles.

4.1 Navigation with static obstacles

4.1.1 First scenario

Eleven static obstacles, with different values of the radius, have been placed in an arbitrary positions between the robot initial position \([X_R(0), Y_R(0), \alpha_R(0)] = [0, 0, \alpha_g]\) and the desired goal \((x_g, y_g) = (7, 9)\). The translational velocity and the maximum angular velocity have been chosen respectively as \(V = 0.2\, \text{ms}^{-1}\), \(\alpha_{\text{max}} = 0.22\, \text{rad.s}^{-1}\). The sampling rate is set to be \(\tau = 0.1\, \text{s}\). Figure 5 illustrates the mobile robot trajectory depicted with small circles (interrupted lines) and Figure 6 shows the same navigation showing depicted the disc (dashed line). Once the mobile robot senses the obstacle, it changes its heading towards a safe direction in order to avoid collision with obstacles. As we can see, the robot is able to move around static obstacles or to pass through the middle of the gap.
between them. The simulation results show that the robot not only achieves the obstacle avoidance but also ensures to reach efficiently the goal with a high speed navigation.

**Figure 5** Robot navigation with static obstacles: simulation without showing the disc ($R = 1$) (see online version for colours)

![Figure 5](image1)

**Figure 6** Simulation showing the disc (see online version for colours)

![Figure 6](image2)

### 4.1.1.1 Influence of the radius

In this part, we try to change the disc radius value and to prove that the variation of the radius $R$ of the disc affects the nature of the robot path and the travelling time. Thus, we summarised in Table 1, the results of varying the radius $R$. We define $\Delta S = s_2 - s_1$, the covered distance between the starting and the ending points and the period $\Delta t = t_2 - t_1$. 
Table 1  Variation of the radius

<table>
<thead>
<tr>
<th></th>
<th>$\Delta t$ (s)</th>
<th>$\Delta S$ (m)</th>
<th>Reach the goal</th>
<th>Avoid the obstacle</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R = 0.2$</td>
<td>50.9</td>
<td>10.18</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$R = 0.5$</td>
<td>53.2</td>
<td>10.64</td>
<td>Yes</td>
<td>Very near to obstacle</td>
</tr>
<tr>
<td>$R = 1$</td>
<td>53.7</td>
<td>10.74</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$R = 3$</td>
<td>72.7</td>
<td>14.54</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If the radius is small ($R \leq 1$) the robot passes between obstacles to reach the goal. The path is optimised and the robot spends a short time. However, when the disc is very small (case ($R \leq 0.5$)), the robot cannot sense obstacles ahead that is why it risks to be very close to obstacles or to hit them.

If we increase the radius ($R \geq 3$) the robot moves too much away from obstacles in order to avoid collision. Consequently, the robot travels a much longer path to attain the target and spends a long time.

These radius values were determined through a number of empirical trials and were resulted in the best safe and efficient robot movement. Thus, for the best choice $R = 1$ m, the travelled distance to the goal is $\Delta S = 10.74$ m and the robot spends $\Delta t = 53.7$ s to reach the goal. Figure 7 illustrates different scenarios showing the influence of varying the radius.

Figure 7  Influence of the radius (see online version for colours)
4.1.2 Second scenario

In order to prove that the developed approach is robust and efficient in partially known environment, we have constructed a virtual environment containing static obstacles. Figure 8 shows the scene supplied to the mobile robot. In this scene, six static obstacles are placed with an arbitrary way in front of the robot path and between the robot initial position and its goal. The robot should begin at point $S$ and finish at point $G$. In such a crowd environment the developed algorithm is used to guide the mobile robot to fulfil the task.

![Figure 8 Static environment (see online version for colours)](image)

For this reason, the robot has to move and bypass between obstacles. In order to have an efficient and best safe robot movement, the value of the radius of the disk is chosen equal to 1.25 m. Also, the linear velocity is set 0.2 ms$^{-1}$ and the maximum angular velocity is 0.22 rad.s$^{-1}$. Figure 8 shows that the robot accomplishes the navigation mission and spends 133.5 s to achieve the target.

4.2 Navigation with moving obstacles

In Figure 9, the robot is navigating with three dynamic obstacles starting its motion as mentioned in Table 2. The robot initial position is $[X_R(0), Y_R(0), \alpha_R(0)] = [0, 0, \alpha_g]$ and the desired goal $(x_g, y_g) = (12, 12)$. We set the translational and the angular velocities $(V, \omega) = (0.3, 0.22)$. In the following simulations, we show different scenarios illustrating the mobile robot moving towards the goal in a dynamic environment. As it can be seen in Figure 10, the robot tries to detour the moving obstacles from its front and changes its direction when it detects the obstacle. Finally, the mobile robot accomplishes successfully the navigation mission and reaches the stationary goal.
Figure 9  Dynamic environment (see online version for colours)

Table 2  Initial condition of obstacles

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>$x_{obs}$</th>
<th>$y_{obs}$</th>
<th>$\theta_{obs}$</th>
<th>$r_{obs}$</th>
<th>$V_{obs}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle 1</td>
<td>2</td>
<td>4</td>
<td>$-\frac{\pi}{2}$</td>
<td>0.5</td>
<td>0.05</td>
</tr>
<tr>
<td>Obstacle 2</td>
<td>6</td>
<td>10</td>
<td>$-\frac{\pi}{2}$</td>
<td>0.3</td>
<td>0.04</td>
</tr>
<tr>
<td>Obstacle 3</td>
<td>11</td>
<td>4</td>
<td>$\frac{\pi}{2}$</td>
<td>0.2</td>
<td>0.04</td>
</tr>
</tbody>
</table>

In order to prove that there are no collision between the robot and obstacles, we illustrate in Figure 11, the curves of the robot’s Cartesian coordinates ($X_R$, $Y_R$) depicted with continuous line and the curves of Cartesian coordinates of different obstacles represented by dotted lines. If there is a collision between the robot and obstacles, they will have the same Cartesian coordinates at the same time. Observing Figure 11, it is easy to conclude that there is no collision between the robot curves and obstacles. This proves that the robot moves away from mobile obstacles and does not collide them.

4.3 Navigation with static and moving obstacles

In this part, we examine the developed algorithm by a series of navigation simulations in known environment with static and moving obstacles at the same time. The task of the mobile robot is to know which path leads to the goal and ensuring a free collision.
Figure 10  The mobile robot environment containing moving obstacles (see online version for colours)

Figure 11  Cartesian coordinates curves of the robot and obstacles (see online version for colours)
Seven obstacles are placed in the environment as it is shown in Figure 12. Obstacles 1 and 2 are dynamic obstacles and the others are stationary obstacles. In the following numerical values are expressed in international system of units: \( V \) in m/s\(^{-1} \) and \( \omega \) in rad.s\(^{-1} \). Obstacle 1 moves at linear and angular velocities (0.01, 0) from point (5, 9) and Obstacle 2 moves at linear and angular velocities (-0.01, 0) from point (14, 10). The others obstacles are located at points (5, 5), (7, 9), (5, 3), (11, 12), (12, 7).

**Figure 12** Path for the robot with static and moving obstacles (see online version for colours)

In the beginning, the robot moves away from the influence of static obstacles. Then, when the robot detects the moving obstacle 1, it changes its direction and continues its way while avoiding the second obstacle 2. Finally, the robot accomplishes the navigation mission and succeeds to reach the goal. To this end, we conclude that the developed strategy is robust and efficient in both static and dynamic environments.

### 4.4 Target tracking

The target tracking is to make the robot asymptotically to follow a required trajectory. Chunyu and Qu (2010) and Masehian and Katebi (2007) have presented methods of target tracking in a dynamic environment. In the first part of this work, we prove the effectiveness of our algorithm and its ability to reach a stationary given target. In this section, we prove also that the developed algorithm is able to track a moving target in a dynamic environment.

**Assumption:** We consider another robot as the moving target. The target robot has the same kinematic model presented in Section 2. We assume that only one moving target exists.
We note $V_T$ and $\theta_T$ the target linear and angular velocities (both known by the mobile robot at the beginning of the motion planning phase).

Figure 13 shows the mobile robot path in a dynamic environment containing three moving obstacles. Obstacles 1, 2 and 3 move at linear and angular velocities (0.05, 0), (0.02, 0) and (0.02, 0), respectively and starting from points (2, 4), (8, 4) and (10, 15), respectively. In the following simulation, the robot motion path, from starting point towards the target, is shown. The moving target and the mobile robot trajectories are depicted with small circles and care, respectively. At first, the robot sets off from point (0, 0) and it is affected only by the target. In the beginning, the mobile robot moves away from the influence of moving obstacles 1 and 2. Then, it detects the obstacle 3 so it changes its direction and tries to detour the moving obstacle from its front. After that, the robot turns again and it follows the target.

The simulations show that the robot successfully catches the moving target and simultaneously avoids the moving obstacles, which proves the performances of the developed algorithm.

5 Experimental results

Experiments with the developed navigation algorithm are examined on a differential drive robot called Pioneer-3DX robot. This robot controlled by a real time control architecture implemented in an embedded laptop as shown in Figure 14. The Pioneer is used for applications and research involving mapping, localisation, monitoring, autonomous navigation, etc. The robot characteristics are illustrated in Table 3. We set the translational velocity and the maximum angular velocity $V = 0.2$ m.s$^{-1}$ and $\alpha_{\text{max}} = 0.22$ rad.s$^{-1}$. The sampling rate is set to be $\tau = 0.1$ s as in simulations.
Figure 14  Differential drive mobile robot (see online version for colours)

Table 3  Pioneer specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot length</td>
<td>237 mm</td>
</tr>
<tr>
<td>Robot width</td>
<td>455 mm</td>
</tr>
<tr>
<td>Wheel radius</td>
<td>195 mm</td>
</tr>
<tr>
<td>Swing radius</td>
<td>267 mm</td>
</tr>
<tr>
<td>Turn radius</td>
<td>0 mm</td>
</tr>
<tr>
<td>Axe length</td>
<td>381 mm</td>
</tr>
<tr>
<td>Max forward/backward speed</td>
<td>1.2 m/s</td>
</tr>
<tr>
<td>Rotation speed</td>
<td>5.23 rad/s</td>
</tr>
</tbody>
</table>

Figure 15  The mobile robot environment containing static obstacles (see online version for colours)
Figure 15 shows a static environment with two obstacles. The mobile robot follows the same path depicted with small black circles showed in Figure 16 and reaches the target.

As we can see in Figures 15 and 16, the mobile robot is able to move around static obstacles. When it detects the two obstacles, the robot changes its heading towards safe direction in order to avoid collisions and it reaches efficiently and safely the goal.

Figure 16  Mobile robot navigation among static obstacles (see online version for colours)

6 Conclusions

In this paper, we have proposed a developed algorithm based on an integrated representation of the environment. The aim is to make the robot able to find a collision-free path between the starting point and the goal in cluttered environment containing static and moving obstacles. The development of a computer simulations for navigation of wheeled mobile in static and dynamic environments has been described. The simulation results prove that the developed algorithm proves a high effectiveness for obstacle avoidance and convergence to the target. The performance of the developed real-time navigation strategy has been confirmed by experimentation on a Pioneer-3DX robot.

Acknowledgements

The authors would like to thank Professor Ahmed Chemori for his advices and successful discussions. Experiments with the developed navigation algorithm have been examined in the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM), France.
References


Autonomous mobile robot navigation


