Trajectory Tracking of a Differentially Driven Wheeled Mobile Robot in the Presence of Obstacles

Raphael Grech and Simon G. Fabri†
Department of Electrical Power and Control Engineering
University of Malta
Msida MSD 06, Malta
† *E-mail: sgfabr@eng.um.edu.mt*

Abstract

A trajectory following and obstacle avoidance mechanism for a mobile robot is presented for situations where the robot has to follow a specific target trajectory but the task might not be completely possible due to obstacles in the way, which the robot must avoid. After avoiding an obstacle, the robot should catch up with the target trajectory. In the proposed system, this objective is reached by combining a nonlinear control method with an Artificial Potential Function method, leading to trajectory tracking control with obstacle avoidance capabilities.

1. Introduction

In the mobile robotics literature, one could distinguish between two main problem domains that have been studied separately, namely (i) *trajectory tracking* and (ii) *obstacle avoidance (path planning)* [1, 2, 3]. This paper proposes a method for integrating the solutions to these two separate domains so as to generate a trajectory tracking mechanism for a mobile robot, that is also able to avoid obstacles along the way.

A trajectory defines a specific path with an associated time law. Therefore, in trajectory tracking, the robot is asked to be at a given point on the path at a given time. On the other hand, obstacle avoidance is the ability to devise a suitable path for the robot to reach some goal position without colliding into obstacles that may be present between the initial position and the goal. In contrast to trajectory tracking, where no obstacles are assumed and the main aim is to guarantee that the robot follows the target trajectory, obstacle avoidance is concerned with generating a path to reach the goal position without collisions, and no constraints are imposed by any target trajectory.

This paper amalgamates the two scenarios, leading to an obstacle avoidance capability within trajectory tracking schemes. The solution to the trajectory tracking problem is well-known as long as there are no obstacles along the way [1]. In the presence of obstacles, a new path will have to be formed around the obstacle such that the robot will temporarily depart from the target trajectory and follow the new path around the obstacle. In this paper, the new path is formed using Artificial Potential Functions originally proposed by Khatib [2] for obstacle avoidance. However, in the scheme being proposed, after an obstacle is circumvented, the robot should follow once again the target trajectory until another obstacle is detected. This paper proposes three methods of achieving this objective.

The paper is organised as follows: Section 2 will present the mathematical background and briefly review conventional trajectory tracking and obstacle avoidance techniques as two separate domains. The three techniques being proposed for achieving trajectory tracking in the presence of obstacles are developed in Section 3. Section 4 illustrates some simulation results and finally conclusions are presented in Section 5.

2. Background

The robot considered in this report is a differentially driven wheeled mobile robot (WMR). This type of robot is driven using two independently controlled wheels. The kinematic structure of the vehicle prohibits certain vehicle motions and imposes nonholonomic constraints [4]. The control inputs are the robot linear velocity (V) and the angular velocity (ω). The kinematic state of the robot is given by vector $[x\ y\ \theta]^T$, where (x, y) represent the Cartesian position of the robot on a plane with respect to some reference frame and θ is the robot orientation. The triplet (x, y, θ) is often referred to as the pose of the robot. Using the kinematic model for a differentially driven WMR, the pose of the robot can be determined by Equation (1) from the control parameters V_L and V_R, which represent the linear velocities of the left and right wheel respectively. $2r$ denotes the distance between the wheels.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{2} & \frac{\cos\theta}{2} \\ \frac{\sin\theta}{2} & \frac{\sin\theta}{2} \\ \frac{1}{2r} & -\frac{1}{2r} \end{bmatrix} \begin{bmatrix} V_R \\ V_L \end{bmatrix} \tag{1}$$

V and ω are related to the left and right wheel velocities V_L and V_R by the equations:

$$V = \frac{V_L + V_R}{2} \quad ; \quad \omega = \frac{V_R - V_L}{2r}$$

In this paper, the following assumptions are taken:

A1) The obstacles are assumed to be circular.

A2) The positions of the robot and the obstacles can be measured accurately.

A3) The desired position, orientation and velocity are known; with the velocity not exceeding the robot's maximum velocity.

A4) The obstacles are sparse in the environment and not cluttered.

A5) The robot's initial position is not within an obstacle.

A6) The target trajectory is smooth and continuous.

2.1. Trajectory tracking

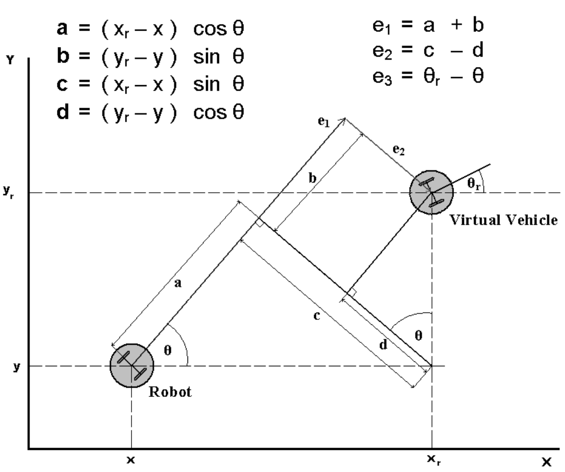

Figure 1. Virtual vehicle following error

The conventional trajectory tracking problem for a WMR of the unicycle type is formulated with the introduction of a *virtual reference vehicle* to be tracked by the robot. The virtual reference vehicle defines the desired target trajectory. The controller should asymptotically reduce down to zero the coordinate error $[e_1\ e_2\ e_3]^T$, detailed in Figure 1, between the real robot and the virtual vehicle. In this paper the nonlinear control design proposed by De Luca *et al.* in [1] will be used for trajectory tracking purposes. This design is briefly reviewed next, where all terms with a subscript r will have the usual definition but referred to the virtual vehicle rather than the mobile robot.

Given that the robot must follow the Cartesian trajectory of the virtual vehicle, the desired reference coordinates $x_r(t)$ and $y_r(t)$ of the path traced by the virtual vehicle at any given time t must be provided to the controller. The calculation of the linear velocity V_r and angular velocity ω_r of the virtual reference vehicle from $x_r(t)$ and $y_r(t)$ is called feedforward command generation. Using the kinematic model Equation (1), the virtual reference vehicle orientation is given by

$$\theta_r = \mathrm{atan2}\left(\dot{y}_r, \dot{x}_r\right) \tag{2}$$

where $\mathrm{atan2}$ is the four-quadrant inverse tangent function of $(\dot{y}_r/\dot{x}_r)$ (undefined only if both arguments are zero), and the linear velocity is

$$V_r = \pm\sqrt{\dot{x}_r^2 + \dot{y}_r^2} \tag{3}$$

The angular velocity ω_r is obtained by differentiating Equation (2) with respect to time yielding,

$$\omega_r = \frac{\ddot{y}_r \dot{x}_r - \ddot{x}_r \dot{y}_r}{\dot{x}_r^2 + \dot{y}_r^2} \tag{4}$$

By applying Lyapunov stability analysis, the following control law ensuring that the robot will asymptotically track the reference vehicle was derived in [1]:

$$V = V_r \cos\left(\theta_r - \theta\right) + k_1 \left[\cos\theta \left(x_r - x\right) + \sin\theta \left(y_r - y\right)\right] \tag{5}$$

$$\begin{aligned} &\omega = \omega_r + \\ &k_2 V_r \frac{\sin\left(\theta_r - \theta\right)}{\theta_r - \theta} \left[\cos\theta \left(y_r - y\right) - \sin\theta \left(x_r - x\right)\right] + \\ &k_3 \left(\theta_r - \theta\right) \end{aligned} \tag{6}$$

where

$$\begin{aligned} k_1 &= k_3 = 2\zeta\sqrt{\omega_r^2 + \beta V_r^2} \\ k_2 &= \beta \end{aligned}$$

with $\beta > 0$ and $\zeta \in (0, 1)$ being design parameters.

Simulation results for the above system are shown in Figure 2. The reference trajectory is a circle defined by $x_r = 20\cos(t)$; $y_r = 20\sin(t)$. Note that the robot converges to the reference trajectory very quickly, although it is to be noted that the dynamics of the robot were not considered in this simulation.

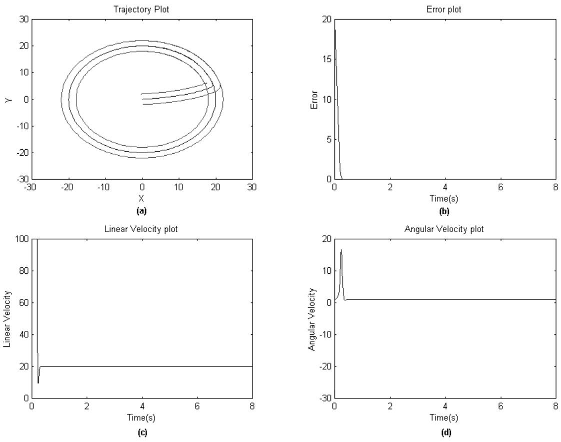

Figure 2. Trajectory tracking control: (a) robot trajectory (b) Euclidean norm of positional error (c) linear velocity, (d) angular velocity

2.2. Obstacle avoidance

In the Artificial Potential Field (APF) method for obstacle avoidance, the idea of imaginary forces acting on a robot is used [2]. The underlying concept is to fill the robot's workspace with an artificial potential field where the goal position to be reached by the robot is an attractive global minimum, and the obstacles are repulsive maxima [5]. The robot follows the gradient of this potential towards its minimum. A major disadvantage of the method is the possibility of local minima in which the robot might get stuck for certain compositions of targets and obstacles.

Letting $\mathbf{q} = (x, y)$ be the current position of the robot, $\mathbf{q}_g = (x_g, y_g)$ be the position of the goal and $\mathbf{q}_o = (x_o, y_o)$ be the position of a unique obstacle, the APF "felt" by the robot at point $\mathbf{q}$ has the form

$$U(\mathbf{q}) = U_g(\mathbf{q}) + U_o(\mathbf{q}) \tag{7}$$

where $U(\mathbf{q})$ is the resultant potential, $U_g(\mathbf{q})$ is the attractive potential produced by the goal at $\mathbf{q}$ and $U_o(\mathbf{q})$ is the repulsive potential produced by the obstacle at $\mathbf{q}$. The resultant force F acting on the robot is then set to:

$$F(\mathbf{q}) = -\nabla[U_g(\mathbf{q})] - \nabla[U_o(\mathbf{q})] \tag{8}$$

where ∇ denotes the gradient vector. $-\nabla[U_g]$ is an attractive force which guides the control point to the goal and $-\nabla[U_o]$ is a repulsive force exerted by the obstacle. In the case of more than one obstacle, the repulsive force is given by $\sum_{i=1}^{n} F_{o_i}$ where n is the number of obstacles and F_{o_i} is the repulsive force generated by the ith obstacle. The attractive and repulsive potentials in Equation (7) can take various mathematical forms [5]. The system could be simplified considerably by ignoring the dynamics of the robot and setting its velocity, rather than the force, directly proportional to $F(\mathbf{q})$ [6].

3. Trajectory tracking with obstacle avoidance

This section develops the main results of this paper, consisting of three schemes for amalgamating trajectory tracking with obstacle avoidance.

3.1. Scheme 1: The modified APF method

The reference vehicle in trajectory tracking could be regarded as a *moving* goal in the APF method. One way of modifying the classical APF approach to this scenario involving a moving target, would be that of vectorially adding target velocity to the negative gradient of the potential function of Equation (8). This way the robot could catch up with the desired trajectory. When there are no obstacles, the repulsive forces F_{o_i} would be zero. When the robot is at some distance d away from the goal, the APF would produce an attractive force proportional to the distance between the robot and the goal. If the goal were stationary, the force would decrease as the robot approaches it.

Since in this study we are relating the velocity of the robot to the force exerted by the APF in a proportional way, the robot would start decreasing its velocity down to zero as it gets nearer to the goal. However, in the case of a moving goal, we would like the robot to track the target with zero distance error. By adding the velocity of the target to the velocity obtained from the negative gradient of the APF, the robot would initially move faster than the target in order to catch up. Once it reaches the target, the robot velocity becomes equal to the target velocity and the robot follows the desired trajectory with zero error. Note that in the case of obstacles blocking the trajectory, this method will behave similarly to the classical APF method by finding a suitable path around the obstacle. The resultant velocities calculated by this modified APF method are fed into the trajectory tracking controller of Section 2.1. The Modified APF Method therefore provides the velocity components $\dot{x}_d$ and $\dot{y}_d$ along the x and y directions respectively in the world reference frame. To get the desired velocities and orientation in Equations (5) and (6) we set

$$V_r = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2} \tag{9}$$

$$\theta_r = \operatorname{atan2}(\dot{y}_d, \dot{x}_d) \tag{10}$$

Angular velocity ω_r would then be the rate of change of θ_r given by a first order backward difference approximation in a digital implementation as follows:

$$\omega_r(n) = \frac{[\theta_r(n) - \theta_r(n-1)]}{T} \tag{11}$$

where n is the sample number and T the sampling period.

3.2. Scheme 2: switching method A

The nonlinear control system described in Section 2.1 is guaranteed to drive the robot to the desired coordinates on the trajectory in the absence of obstacles. Using the Modified APF Method (i.e. adding the target velocity to the negative gradient of the APF) there is no theoretical guarantee that once the robot is on the target it would keep moving with the target velocity. This is mainly due to the fact that the outcome of the APF is not known *a priori* but is generated in real time. Being so, some of the control parameters would have to be calculated experimentally in real time since there would be no prior, known mathematical equation related to the desired trajectory to represent these parameters. Another drawback of using the modified APF method is the need for careful gain selection. Ideally, gains should not be unnecessarily large but just right for the system to react well. Choosing small gains would make the system slow to react in the case where an obstacle is to be avoided and a moving goal to be reached. On the other hand, large gains would demand large velocities on the robot. In practice, a robot is bound not to exceed a maximum velocity. If the robot is not able to deliver the requested velocity, the trajectory tracking might not be smooth and in some extreme cases might even go unstable due to convergence problems.

In order to avoid these problems of the Modified APF Method, a different approach is proposed next. A switching mechanism is used to select between the Modified APF Method when the robot is close to an obstacle, and the trajectory tracking algorithm of Section 2.1 when no obstacles are present. In the latter case, the robot controller is fed only with the desired trajectory's co-ordinates and velocities, thus removing redundant computation related to APF and, above all, ensuring that the robot will surely converge to the desired trajectory as proven in [1]. When an obstacle is detected, the system switches onto the Modified APF Method and it generates an alternative trajectory for the robot to follow. Once the robot surpasses the obstacle and moves a certain distance away, the system switches back to the ideal trajectory tracking scheme. The switching is dependent on the vicinity of the robot to the obstacle as shown in Figure 3. In a Region A surrounding the obstacle, the Modified APF scheme is used. Outside Region A, the trajectory tracking algorithm is used. The size of Region A is chosen such that it is large enough to allow for the robot to manoeuvre around the obstacle in order to avoid it. V_r, ω_r and θ_r in Equations (5) and (6) are generated by the Modified APF Equations (9), (10) and (11) when close to the obstacle, or from the ideal trajectory Equations (2), (3) and (4) when outside the obstalce's region of influence.

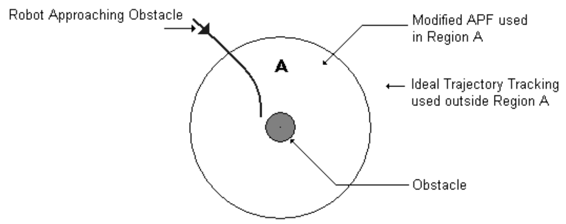

Figure 3. Hard switching mechanism

The advantage of Scheme 2 over Scheme 1 is that when there are no obstacles along the way, the APF is not used at all. If the error between the robot and the goal trajectory is large (e.g. initially when the robot does not start on target) and there are no obstacles, the robot will try to follow the desired trajectory directly and relatively faster, with guarantees of error convergence.

3.3. Scheme 3: switching method B

The Switching Method A of Scheme 2 involved *hard* switching between the modified APF and the trajectory tracking schemes. There is an imminent possibility of a control discontinuity during the switch, causing velocity spikes to occur. These velocity spikes could make the robot behave unpredictably, since large accelerations are demanded. This would make the robot jitter, oscillate or slip in practice. Such undesirable effects could have adverse consequences on the system. So, to remove these spikes, soft switching is used. This consists in setting a *boundary layer* of thickness b just outside Region A surrounding the obstacle as illustrated in Figure 4.

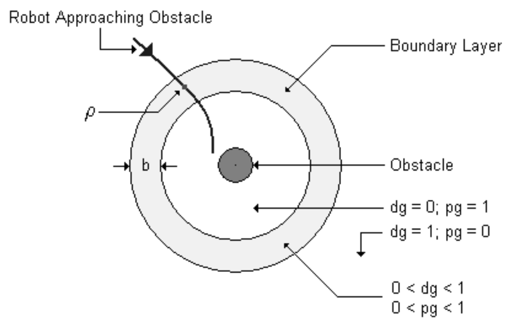

Figure 4. Soft switching mechanism

In the boundary layer, the influence of the APF control is gradually phased in and the trajectory tracking control phased out as the robot position approaches Region A, and *vice-versa*. This is achieved by weighting the control inputs from the two schemes whenever the robot is inside the boundary layer, using appropriate gains calculated as follows:

$$d_g = 1 - \frac{\rho}{b} \quad ; \quad p_g = \frac{\rho}{b}$$

where d_g is the trajectory tracking control gain, p_g is the modified APF control gain, ρ is the radial distance of the robot inside the boundary layer ($\rho = 0$ at the outer perimeter and increases linearly up to a maximum $\rho = b$ at the inner perimeter of the boundary layer).

Inside the boundary layer, as the robot approaches the obstacle, more weight is given to the modified APF control and less weight to the trajectory tracking control. Conversely, when moving further away from the obstacle, the gain of the modified APF control is reduced and that of the trajectory tracking control is increased. This way, the large velocity spikes of Scheme 2 at the boundary of Region A are smoothened, without any drastic change in robot acceleration.

As in Scheme 2, in the ideal case when no obstacles are present, the feedforward command generation is derived directly from the trajectory tracking scheme only. This would lead to using Equations (2) to (6). Within Region A, the Modified APF Method of Scheme 1 is used with Equations (9), (10) and (11) applied into the control law Equations (5) and (6). Within the boundary layer, V and ω from both schemes are linearly combined after weighting with gains d_g and p_g.

4. Simulation results

The three control schemes described in Section 3, namely

***A*)** the Modified APF Method

***B*)** Switching Method A

***C*)** Switching Method B

are now simulated and compared. The same trajectory as in the simulation of Section 2.1 is used. However, this time, three fixed circular obstacles are placed on the trajectory at points $[-15, 15]$, $[15, -15]$ and $[-10, -15]$. The latter obstacle has a radius of nine units, and the former both have a radius of four units. The APF parameters were all set the same for the three situations. The influence of the obstacle was set to start at 2 units away from the obstacle. The robot's maximum velocity was limited to 100 units/s.

4.1. The modified APF method

Figure 5 shows the trajectory, error and robot velocity when using the Modified APF Method. In this simulation, the robot successfully tracks the trajectory and overcomes

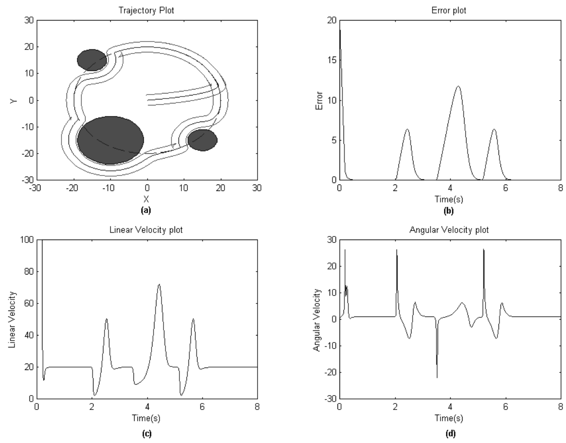

Figure 5. Results for the modified APF method: (a) trajectory following (b) positional error (c) linear velocity (d) angular velocity

all three obstacles. However, there is no theoretical guarantee (as yet) that once the robot is on the target it would keep moving with the target velocity. This was the reason for proposing the switching methods A and B where, in effect, the Modifed APF Method is used only when required *i.e.* when an obstacle is sensed. Otherwise, trajectory tracking is applied.

4.2. Switching method A

Figure 6 shows the results for the hard switching method. Switching occurs at 4 units away from obstacle.

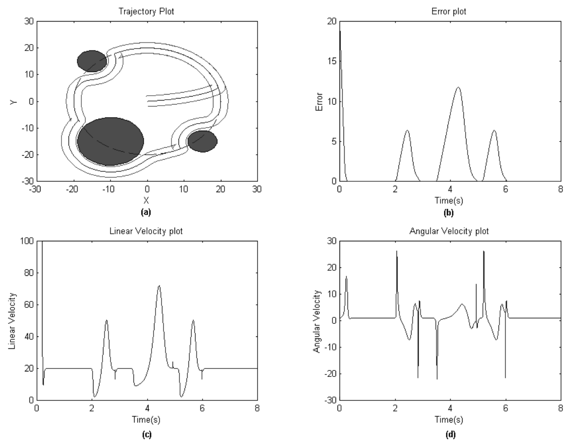

Figure 6. Results for the hard switching method: (a) trajectory following (b) positional error (c) linear velocity (d) angular velocity

The results show once again that the robot successfully avoids the obstacles. This time we are also ascertained that the robot will successfully track the desired trajectory due to the convergence properties of trajectory tracking control. However, hard switching has the problem of velocity spikes occurring during switching, as evident clearly in Figures 6c and 6d. These spikes might cause the robot to jitter and not behave well. In order to avoid this, soft switching is proposed, the simulations of which are shown next.

4.3. Switching method B

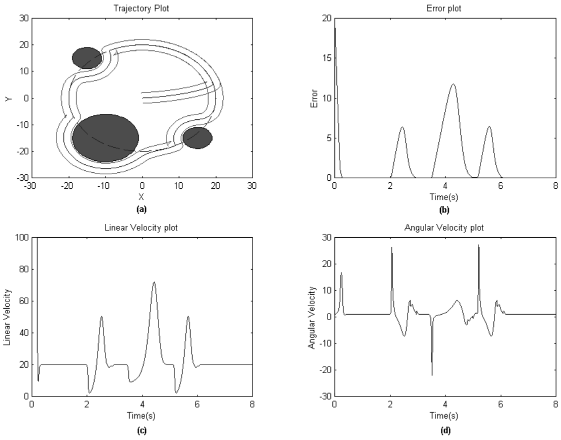

Figure 7. Results for the soft switching method: (a) trajectory following (b) positional error (c) linear velocity (d) angular velocity

Simulation results for soft switching between trajectory tracking and Modified APF control are shown in Figure 7. The boundary layer was set between 2 and 6 units away from the obstacle. Once again, the trajectory is followed and all obstacles are avoided, with the velocity spikes occurring on switching clearly dampened (plots 7c and 7d). One can say that the effect of switching is hardly noticeable and the velocities are practically similar to the Modified APF Method. Moreover, in situations when the gains of the Modified APF scheme were not optimally tuned, the Soft Switching method proved to be even superior due to the fact that once the robot avoided an obstacle, it would track back to the target trajectory relatively faster.

5. Conclusion

By the amalgamation of two concepts, namely trajectory tracking and APF path planning, this paper has presented novel control schemes that are able to track a trajectory very accurately even in the presence of obstacles. The appeal of this study is the possibility of having the desired trajectory passing through an obstacle and the robot successfully avoiding it, whilst keeping loyal to the desired trajectory once the obstacle is surpassed.

From the simulations presented, it was shown that by using the switching mechanisms proposed in this paper, the robot successfully circumvents the obstacles and gets back on target with the guarantee that the robot will converge to the target trajectory, once away from the obstacle. As yet this cannot be theoretically guaranteed in the Modified APF Method, also proposed in the paper. Hard and Soft Switching schemes were tested, the latter exhibiting smoother transitions during switching.

This research could be developed by testing out the proposed schemes on an actual WMR. Also, different types of potential functions or other obstacle avoidance mechanisms can be studied in order to make the system more robust and eliminate problems associated with local minima in APF. Analysis of the convergence properties of the Modified APF Method is another important issue that requires investigation.

References

[1] A. De Luca, G. Oriolo, and M. Vendittelli, *Control of Wheeled Mobile Robots: An Experimental Overview*. Springer-Verlag Heidelberg, 2001, vol. 270/2001, pp. 181–226.

[2] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *The Inter. Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[3] S. Ge and Y.J.Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots 13*, pp. 207–222, 2002.

[4] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2000.

[5] M. Khatib, "Sensor-based motion control for mobile robots," Master's thesis, Laboratoire d'Automatique et d'Analyse des Systèmes, Toulouse France, 1996.

[6] M. Frixione, G. Vercelli, and R. Zaccaria, "Diagrammatic reasoning for planning and intelligent control," *IEEE Control Systems Magazine*, vol. 21, no. 2, April 2001.