Indoor Navigation for Mobile Robots using Predictive Fields

Ninad Pradhan*, Timothy Burg*, Stan Birchfield*, Ugur Hasirci†
*Dept. of Electrical and Computer Engineering, Clemson University
†Dept. Of Electrical and Electronics Engineering, Duzce University

***Abstract*— A predictive field based path planner for mobile robot navigation in indoor areas is described. Predictive fields are used for incorporating moving obstacle information into the navigation function framework. Navigation functions have been limited by geometric restrictions for robot workspaces and cannot easily be used in everyday environments. A methodical description of indoor workspaces such that they follow the constraints of a navigation function based path planner is proposed. Typical use of navigation function based path planners as a gradient based control input requires large scaling factors in practical use. A direction based control input, which eliminates the scaling factor altogether, is proposed and its stability and convergence is proved using Lyapunov-type analysis. The proposed algorithm improves the practicability of navigation functions and makes it possible to envisage a person following mobile robot operating in indoor environments using predictive field navigation.**

I. INTRODUCTION

Using a mobile robot to follow a person around indoor environments is an interesting application of service robotics. Such robots can potentially be used in hospitals and homes for assisting the elderly or the infirm. In workplaces, robotic carts could be used to follow the workers to transport supplies and instruments. The major challenges for such an application are identification and tracking of the leader, and planning a collision-free path behind the human or robotic leader in the presence of stationary and moving obstacles. We seek to tackle the problem of person following by addressing its primary components, i.e. path planning and sensing, separately.

In this paper, we present a path planning approach for person-following mobile robots operating in indoor environments. This approach builds on the classic Rimon-Koditschek path planner. Rimon and Koditschek, in their seminal work [1], [2], proposed an integrated path planning, motion planning, and control approach which they called the navigation function method. A navigation function is a mathematical description of the workspace which guarantees a collision-free and singularity-free path for the mobile robot. In spirit, it resembles Khatib's work [3], which is the widely used potential function method. Similar to the potential function approach, the navigation functions create a topology in which the robot is attracted to goal and repelled from obstacles. However, the navigation function differs from the potential function method in that it ensures a unique minimum at the goal and the absence of local minima as long as certain conditions are met while formulating the navigation function.

Navigation functions are attractive solutions for path planning because of their relative ease of formulation, their packaging of various aspects of robot navigation into a single framework, and the fact that they prove convergence to the goal. Thus, they have attracted attention [4], [5], [6], [7] since they were first formulated. Some improvements have been made to the original approach. Tanner et al. [7] showed navigation functions could be applied to complex robotic manipulators by systematically decomposing the object into spherical regions. Chen [6] demonstrated that they work with moving obstacles in the workspace. Most recently, Filippidis [4] proposed an algorithm to automatically tune the navigation function gain. In their original formulation, navigation functions did not allow for incorporating an element of uncertainty which is inevitable to obstacles moving in practical environments. A solution to this was proposed in [8], in which moving obstacles were represented using elliptical envelopes around them called predictive fields.

Even with these structural improvements in place, navigation functions are still limited by their lack of practicability. They require the use of arbitrary scaling factors to be applied to the dimensions of a given workspace. Mapping from the navigation function world, which is circular or spherical, to the real world (or vice versa) requires estimating a transformation called the star-world transformation [1], which is difficult to realize in practice. The attractiveness of its mathematical guarantees of convergence do make a convincing case for navigation functions to be used in applications; however, due to these constraints on implementation, there has not been widespread adoption. Discrete or cell based path planners [9] are relatively easier to implement.

In this paper, we seek to move towards a practical navigation function based person following robot. A workspace generation approach which applies the spherical world constraint from Rimon and Koditschek is presented. This approach takes into account sensor feedback which is available to the robot. The scale factor is eliminated through the use of a gradient direction controller, and the stability of this controller is proved. Finally, the predictive fields method from our previous work [8] is applied to account for motion, and uncertainty thereof, of moving obstacles.

II. REVIEW - PREDICTIVE POSITION FIELDS

Predictive position fields were used previously [8] to describe the motion of obstacles within the navigation function framework. Navigation functions were not originally used with moving obstacles in the workspace. The workspace was assumed to be completely static until it was shown [6] that the rules of navigation functions did permit convergence to the goal when the obstacles were in motion. However, obstacles in practical environments move with an inherent uncertainty, and this important property was not part of the moving obstacles solution in [6]. On the discrete path planning side, there have been various efforts, such as the use of reinforcement learning [10] or of probabilistic maps based on predicted obstacle motion [11]. Analogous approaches on the potential fields side include Ge and Cui [12], and Melchior et al. [13].

The use of elliptical position fields was an effort to incorporate obstacle motion information in navigation functions. The significance of the ellipse was in capturing the speed and motion uncertainty of the obstacles. Navigation functions require the definition of an obstacle avoidance β term, which is an indirect measure of the repulsion experienced by the robot due to the obstacle. The requirements for an obstacle β are quite simple: it should reach 0 when the robot and obstacle contact each other, and should be a non-decreasing function of the distance elsewhere.

These minimal constraints provide an opportunity for shaping this repulsive term according to the predicted motion of the obstacle. An ellipse is generated such that the shape of the ellipse has physical significance with respect to obstacle motion. The major axis represents the direction of predicted obstacle motion, its length is representative of the speed of the obstacle, and the minor axis indicates the degree of certainty of this prediction. The obstacle sits at one focus.

To incorporate a sense of the ‘danger’ posed by an obstacle based on its predicted motion, the β term is comprised of two components: β_e outside the ellipse, and β_c inside the ellipse. Outside the ellipse, the robot is repelled by the projected position of the obstacle along the major axis. When the robot enters into the elliptical field, an analytic switch causes the β_c term to act as the repulsive term, and the robot is repelled from the measured position of the obstacle. Fig. 1(a) and Fig. 1(b) illustrate the variation in the repulsive term as the robot approaches an obstacle’s elliptical field.

It was found that the use of elliptical position fields caused the robot to swerve away from the projected path of the obstacle, thus away from a potential collision with it. Further, since the geometry of the workspace was not being changed, the obstacle itself was still circular in shape when the repulsive term was computed. Retaining the spherical world constraint from [1] ensured that none of the mathematical constraints for navigation functions were violated, while a measure of motion uncertainty was added to the formulation.

III. Control Development

Navigation functions solve the problem of driving a robot from its given position in the workspace to the goal position while constantly navigating free configuration space. A navigation function φ is defined to have the following properties, according to Rimon and Koditschek [1], [2].

Let $q \in \Re^{1\times 2}$ denote robot position. Let $q_d \in \Re^{1\times 2}$ be the goal point in the interior of a robot free configuration space F. A map $\varphi : F \to [0,1]$ is a navigation function if it is

- analytic on F,
- polar, with a unique minimum at q_d,
- admissible on F, and
- a Morse function.

Mathematically, this function is described as

$$\varphi(q) = \frac{K_s \left\| q - q_d \right\|^2}{\left[\left\| q - q_d \right\|^{2k} + G(q) \right]^{1/k}}, \quad (1)$$

where $k \in \mathbb{N}$ and the term G is a composite of workspace boundary and interior obstacle avoidance functions. The Rimon-Koditschek [1] definitions for boundary avoidance are retained and the predictive field β from [8] is used to define obstacle repulsion.

The controller introduced here differs from previous controllers [6], [1]. The preceding controller [6] requires the use of an arbitrarily high scale factor, or gain, K_s to drive the robot to goal. The resultant system was found to be extremely gain-sensitive and it was difficult to empirically estimate the scaling required for the robot to be successfully driven to goal for a particular configuration. Thus, a modification is made to the control input, driving the robot using the direction of the navigation gradient rather than the gradient itself. The modified control input to drive the robot to q_d is:

$$\dot{q} = u. \quad (2)$$

where

$$u = -K \frac{\left(\frac{\partial \varphi}{\partial q}\right)^T}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon}$$

$K \in \Re^{2\times 2}$ is a matrix of positive gain values and ϵ is a small positive constant.

A. Stability Analysis

Consider a Lyapunov candidate function, same as the one used in [6],

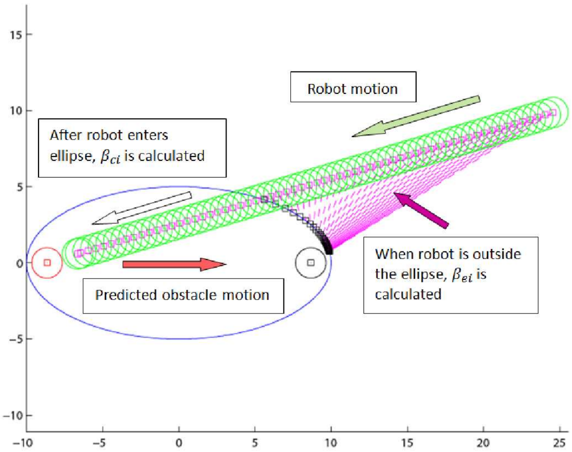
$$V(q) = \varphi(q). \quad (3)$$

First differentiating (3) with respect to time and then substituting the right hand side of (2) yields

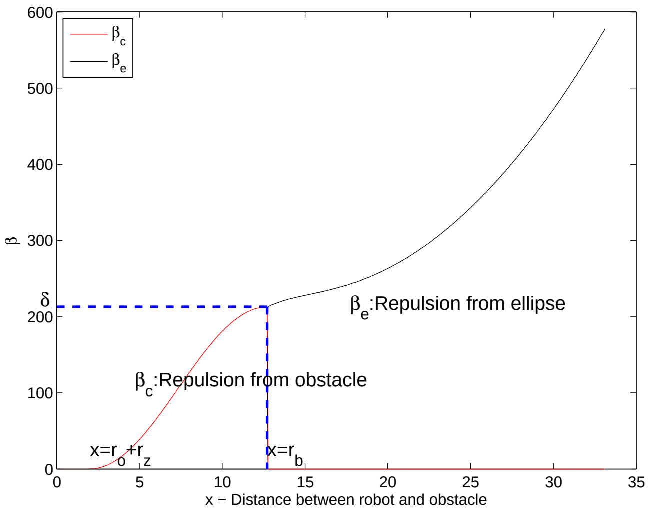
$$\begin{aligned} \dot{V} &= \frac{\partial \varphi}{\partial q} \cdot \dot{q} \\ &= -\frac{\partial \varphi}{\partial q} \cdot K \cdot \frac{\left(\frac{\partial \varphi}{\partial q}\right)^T}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon} \\ &= -f(t) \end{aligned}$$

where $f(t)$ denotes a non-negative function as follows

$$\begin{aligned} f(t) &= \begin{bmatrix} \frac{\partial \varphi}{\partial x} & \frac{\partial \varphi}{\partial y} \end{bmatrix} \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \begin{bmatrix} \frac{\partial \varphi / \partial x}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon} \\ \frac{\partial \varphi / \partial y}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon} \end{bmatrix} \\ &= \frac{1}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon} K_x \left(\frac{\partial \varphi}{\partial x}\right)^2 + \frac{1}{\left\| \frac{\partial \varphi}{\partial q} \right\| + \epsilon} K_y \left(\frac{\partial \varphi}{\partial y}\right)^2 \end{aligned}$$



(a) Intersecting circles indicate the approach of the robot towards an obstacle. The left focus of the ellipse is the actual obstacle position, the right being the predicted position. Predicted position is used for computing the obstacle repulsion term.



(b) When the robot is outside the elliptical field, the quadratic β_e curve determines the overall β. Inside the field, only the circle β, caused by the current position of the obstacle, takes effect.

Fig. 1. The obstacle beta curve as the robot approaches an obstacle [8] [1].

Each of the terms in $f(t)$ is positive. So we can conclude that

$$\dot{V} \leq 0. \tag{4}$$

Therefore $V(q)$ is non-increasing. To establish convergence, we invoke the corollary of Barbalat's Lemma used for stability analysis from [14] (Lemma 4.3) which states that:

- If $V(t)$ is a non-negative function of time on $[0, \infty)$,
- If $\dot{V}(t) \leq -f(t)$, $f(t)$ being non-negative,
- If $\dot{f}(t) \in L_\infty$

then

$$\lim_{t \to \infty} f(t) = 0. \tag{5}$$

It is clear that the first condition is satisfied by the basic requirement of the navigation function as a mapping onto $[0, 1]$, and the second by the proof for $f(t)$ being non-negative. For the third, we consider that the navigation function is analytic on the free configuration space, which establishes that $\frac{\partial \varphi}{\partial q}, \frac{\partial^2 \varphi}{\partial q^2} \in L_\infty$. Thus $\dot{f}(t) \in L_\infty$ is satisfied, and the lemma can be applied.

From the lemma and from the equation for $f(t)$, $\left\|\frac{\partial \varphi}{\partial q}\right\| \to \infty$ as $t \to 0$. The properties of the navigation function imply that $\frac{\partial \varphi}{\partial q} \to 0$ only at the goal configuration q_d.

Hence it is proven that

$$q(t) \to q_d \tag{6}$$

within a specific workspace.

The control input described in this section uses the gradient of the navigation function. The gradient vector is normalized to unit magnitude, but its x and y components vary according to the direction of the gradient. This unit-length directional vector is multiplied by the gain K to create a constant magnitude, velocity control, input to the robot.

IV. Workspace Generation

Navigation functions operate in spherical worlds, or in transformed star-worlds. Each of these geometries is prohibitive for different reasons. Practical indoor environments do not resemble spherical worlds, and finding a mapping from a practical environment (star-world) to a spherical world is non-trivial. These constraints force us to look for another option, i.e. to describe a practical environment as a spherical world. However, this representation has to be practicable; failing this, it will not register as a contribution to the understanding or application of navigation functions to the real-world.

For our proposed method of workspace generation, we make the following reasonable assumptions:

- The robot's sensing range is limited,
- Sensing technology (e.g. Microsoft Kinect) permits gathering all information necessary to implement a navigation function, and
- The person to be followed is always within the sensing range of the robot.

The limited sensing range of the robot makes it possible to define workspace boundaries for the navigation function. The area of the hallway visible to the robot is enveloped by a circular workspace. Intersecting segments of the wall with this sensing envelope are computed. Since the wall is an obstacle in this setup, non-overlapping circular obstacles are generated out of the wall segments. Moreover, each of the stationary and moving obstacles are encompassed by a circular envelope, thus defining non-wall obstacles for the generated workspace. Thus, when the robot begins its person following task, the workspace defined for its motion has been computed to be fully compatible with the requirements of navigation functions. The generated workspace is shown in Fig. 2.

The setup described previously will work well for a single instance. But it is required to extend it over the duration of the person following task. For accomplishing this, an important navigation function constraint needs to be considered. The goal position is not allowed to move inside a workspace. However, this constraint will render navigation functions unusable for person following applications, where the goal is generally moving. Thus, it is necessary to come up with a description of workspaces in which this conflict is resolved.

This is done by using the tracked motion of the leader to generate waypoints for the robot to follow. The setup is easy to visualize. As the robot moves to the goal position in the i^{th} workspace, its sensing algorithm tracks the leader.

[1]To be reprinted with permission from IEEE

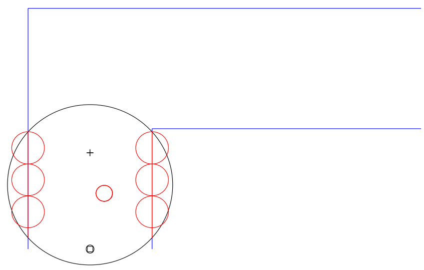


Fig. 2. Generation of workspace compatible with navigation functions. The robot is at the bottom, moving towards the goal marked at the top. Wall segments are enveloped by circular obstacles and internal obstacles are generated in the same manner.

On converging to the i^{th} goal, the sensing algorithm has identified the $i+1^{th}$ goal position. Each workspace contains obstacles which become part of the navigation function formulation for that workspace. This process continues until the person stops or the robot is commanded to stop following the leader. A sample setup, generated waypoints, and obstacles for each workspace are seen in Fig. 3.

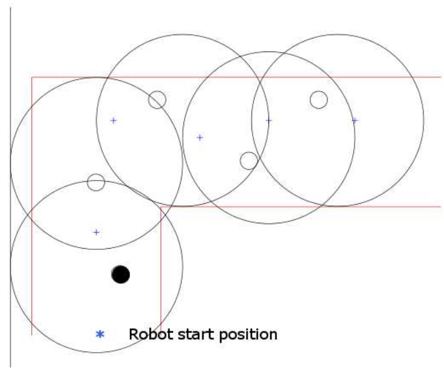


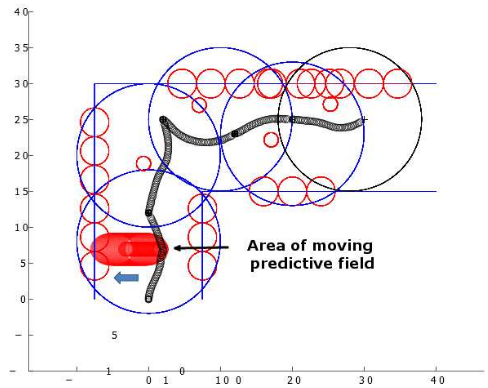
Fig. 3. Generation of leader waypoints as the robot follows the leader. Workspaces are generated so that the navigation function solution to each workspace is that of robot convergence to a static goal position, shown as a "+". Moving obstacles are represented as solid circles.

This completes the description of workspaces such that the robot can follow a person without violating any of the constraints of navigation functions, and without requiring geometric transformations for arbitrarily shaped hallways.

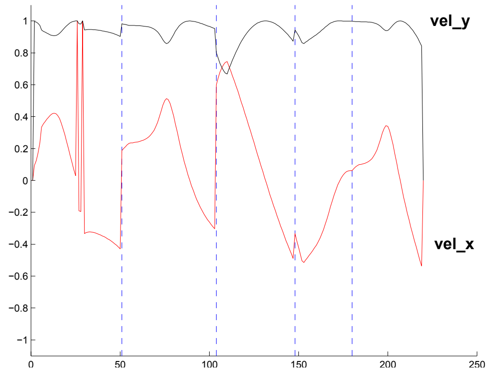
V. Simulation Results

MATLAB Simulink (Mathworks Inc., Natick, MA) was used for simulating the predictive fields path planner and controller. The waypoint generation system for workspace goal positions was simulated by arbitrary selection of waypoints by mouse clicks inside the L-shaped hallway figure. Initial position of the robot was selected using a mouse click; beyond that, the end position of the robot in the current workspace became its start position in the next workspace. Interior obstacles were also manually positioned in the hallway, and their direction of motion was input by the user. Moving obstacles, marked with solid circles in Fig. 3, and stationary obstacles, marked with empty circles, were positioned in the workspace. For interior obstacles, motion prediction had to be simulated using variable sized ellipses, the variable size being another user input. The workspace generation method described in Section IV would then automatically determine the wall segments intersecting with the robot's sensing range for a given workspace and generate wall obstacles for a given workspace. It needs to be emphasized that the process of manual selection and labeling was necessary only in the absence of real-world sensing data. A sensing system will render a completely autonomous person following robot for indoor environments.

Decomposing a hallway into n workspaces worked as expected. This process was designed so that the task was the same as solving n independent navigation function problems. Obstacles were correctly assigned to the individual workspaces by the algorithm, and the robot converged to its goal position in every single workspace, using the velocity control inputs as seen in Fig. 4(b).



(a) Use of the elliptical field allows the robot to avoid the path of the obstacle. This is clearly seen in the first workspace with the overlapping ellipses.



(b) A plot of the individual components of the unit velocity vector being used as the control input. The vertical dotted lines indicate a transition for the robot from one workspace to another.

Fig. 4. Path taken by the robot in the presence of elliptical fields and distance to waypoints in each workspace.

Fig. 5 and Fig. 4(a) illustrate the effect of predictive fields on the path planned for the robot. In the absence of an elliptical field (Fig. 5) the robot moves towards the path of the obstacle before the navigation function guides it away from it. This increases the chances of a collision in uncertain environments, and of moving along a path which is less optimal temporally or spatially. In contrast, the elliptical field provides a path in which the robot moves away from the projected obstacle path much earlier, as seen in Fig. 4(a).

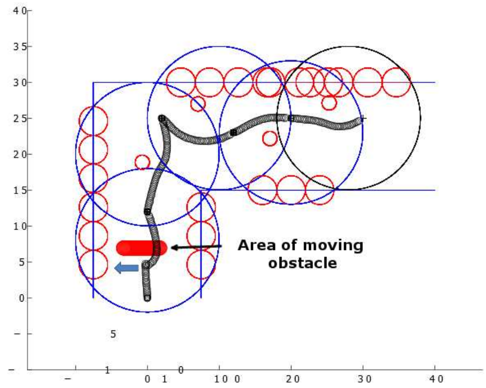


Fig. 5. When the robot navigates to goal in the absence of predictive information, it tends to react to the position of the obstacle and can move towards the path of danger. This is seen clearly in the first workspace at the left of the figure.

It can be seen that, for stationary obstacles, the elliptical field is absent and the predictive formulation reduces to that seen in the classical navigation function systems [6], [1]. An example of this can be seen in Fig. 6.

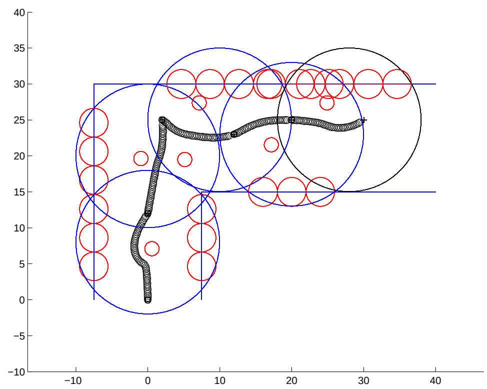


Fig. 6. Predictive field navigation in the absence of obstacle motion is the same as the original navigation function method. The robot converges to goal past stationary obstacles using the unit velocity control input.

The use of a direction based controller allows for a very standard gain value to be used in the setup. The navigation function only has a single gain which needs to be tuned; however, this gain is critical to the working of navigation function based path planners. We found that a gain value of $k = 10$ in Eq. 1 was adaptable to a wide variety of scenarios in our simulated indoor hallway environment. The system scale factor K_s, typically a very large gain, was completely eliminated by our choice of controller.

VI. Conclusions and Future Work

Steps which enhance the applicability of navigation functions to practical indoor environments have been demonstrated. Predictive fields, which were shown in previous work to be effective at moving the robot away from the projected path of moving obstacles, have been used. A new control input, based on navigation function gradient direction rather than the gradient vector, has been proposed. Its stability and convergence have been demonstrated. However, an important part of the proposed approach is the sensing algorithm on-board the robot. Navigation functions require large amounts of workspace information, and we have assumed that this is available to the path planner via a robust sensing system. While this is a valid assumption for the scope of this paper, the usability of navigation functions can be conclusively demonstrated only after experimental testing with such a sensing system in place. Given the useful properties of navigation function based planning and control, experimental verification will be a desirable next step for indoor person following systems using predictive fields.

REFERENCES

[1] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[2] D. Koditschek and E. Rimon, “Robot navigation functions on manifolds with boundary,” *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.

[3] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotic Research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] I. Filippidis and K. Kyriakopoulos, “Adjustable navigation functions for unknown sphere worlds,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, dec. 2011, pp. 4276 –4281.

[5] J. L. Piovesan and H. G. Tanner, “Randomized model predictive control for robot navigation,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1817–1822.

[6] J. Chen, D. Dawson, M. Salah, and T. Burg, “Cooperative control of multiple vehicles with limited sensing,” *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 2-3, pp. 115–131, 2007.

[7] H. Tanner, S. Loizou, and K. Kyriakopoulos, “Nonholonomic navigation and control of cooperating mobile manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53 – 64, feb 2003.

[8] N. Pradhan, T. C. Burg, and S. T. Birchfield, “Robot crowd navigation using predictive position fields in the potential function framework,” in *Proc. American Control Conference*, June 2011, pp. 4628–4633.

[9] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[10] P. Henry, C. Vollmer, B. Ferris, and D. Fox, “Learning to navigate through crowded environments,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[11] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Probabilistic motion planning among moving obstacles following typical motion patterns,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 4027–4033.

[12] S. S. Ge and Y. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[13] P. Melchior, B. Metoui, S. Najar, M. Abdelkrim, and A. Oustaloup, “Robust path planning for mobile robot based on fractional attractive force,” in *Proc. American Control Conference*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1424–1429.

[14] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.