2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Robot Crowd Navigation using Predictive Position Fields in the Potential Function Framework

Ninad Pradhan, Timothy Burg, and Stan Birchfield

Abstract— **A potential function based path planner for a mobile robot to autonomously navigate an area crowded with people is proposed. Path planners based on potential functions have been essentially static, with very limited representation of the motion of obstacles as part of their navigation model. The static formulations do not take into account the possibility of using predicted workspace configuration to augment the performance of the path planner. The use of an elliptical region signifying the predicted position and direction of motion of an obstacle is proposed in this paper. The repulsive potential caused by an obstacle is defined relative to this elliptical field. An analytic switch is made when the robot enters this predicted elliptical zone of the obstacle. The development of navigation functions makes it possible to design a potential-based planner which is guaranteed to converge to the target.**

I. INTRODUCTION

The motivation for our work is to develop a robust vision-based system for a mobile robot to be able to follow a human leader in crowded environments. There are many challenging sub-problems which need to be solved before such a system can be considered complete. One such sub-problem is avoiding moving obstacles while navigating to a goal position in the workspace. We propose a path planner which incorporates probabilistic information in the framework of traditional potential-based path planning to trace a more optimal path to the goal. Computer vision (not discussed in this paper) will be used to provide motion and position information as input to the path planner proposed in this paper.

The simple and effective idea behind a potential function driven path planner is to attract the robot toward the target while simultaneously repelling it from obstacles in its way. These opposing potentials create a topology for navigating the robot. The use of potential functions was proposed in a seminal work by Khatib [1] and has since gained widespread acceptance as a path planning technique for mobile robots. Various potential function based planners ([2], [3]) have been proposed to expand the initial concept.

Early potential field based path planners exhibited local minima, places in the topology where the robot "gets stuck" at a point other than the global minimum located at the destination. Rimon and Koditschek [4] introduced a special kind of a potential function, called the navigation function, to counter this problem. They proved that the structure of the navigation function guarantees a unique minimum at the goal configuration, thus allowing the robot to "roll down" the gradient of the field toward a guaranteed stop at the goal. Chen et al. [5] demonstrated the navigation function approach for multiple robots navigating to their respective goals in the presence of both stationary and moving obstacles.

Even with moving obstacles, motion decisions for these planners are taken without incorporating much, if any, information about how the obstacles are moving. This makes their formulation inherently static in nature. Moreover, even with convergence, the path is not guaranteed to be optimal.

There have been various attempts to maintain the framework of potential functions while improving their alertness to the behavior of objects in the workspace. Ge and Cui in [6], [7] add a velocity term to the system state to enhance the path planner with motion information, making it possible in to track a moving object [6] and to reach a goal with obstacles nearby [7]. In the latest of a series of papers [8], Melchior et al. approach the problem of making dynamic potential field calculations by employing the concept of fractional attractive and repulsive forces. This derives from fractional calculus and is a method of altering the potential of an obstacle based on its level of danger to the robot traversing a path to goal.

Despite their successes, all potential function based planners assume a near-complete knowledge of the position and velocity of the obstacles. Moreover, there is no framework within which these planners can currently incorporate a 'look-ahead' feature, which allows the potential to be shaped by extrapolating from the obstacle's current trajectory. Prediction and some modeling of uncertainty in prediction are vital characteristics for a path planner designed to work in practical, populated environments where objects will rarely move along precise and deterministic trajectories. Navigating through such populated environments or through crowds has seen a great deal of recent interest in path planning literature, where discrete path planners have been used. Techniques used include reinforcement learning [9] and building probabilistic maps based on obstacle motion prediction ([10], [11], [12]).

An extension to the navigation function based path planner [4] which allows obstacle positions as well as their motions to be characterized with a probabilistic region we call a *position field* is described in this paper. The shape of the field conveys information about the predicted position of the obstacle and the confidence in this prediction. The position field is formulated to maintain the simplicity of the navigation function based framework while enhancing its effectiveness in avoiding obstacles and reaching its target. We propose an elliptical position field and keep the current robot position at one focus of the ellipse, while specifying the predicted position to lie at the other focus and computing the potential relative to this predicted position. The size of the major and minor axes of the ellipse encapsulate information

978-1-4577-0079-8/11/$26.00 ©2011 AACC

about the predicted direction of motion and the confidence in this prediction respectively. In keeping with the properties of navigation functions, the proposed path planner ensures that the robot stays within its workspace, avoids collisions with obstacles, and reaches its destination.

II. PROBLEM FORMULATION

The robot is assumed to start inside the 2D workspace, which is circular. The position of the target within the workspace is known to the robot. The position and velocity of obstacles in the workspace are measurable, though their velocity at a future time instant may be known with a degree of uncertainty. This degree of uncertainty is influenced by two factors:

- Knowledge of the history of motion of an object
- Model of the object's motion within the workspace

It is assumed that the above information about uncertainty is conveyed to the path planner by a filtering or estimation algorithm, working with camera or laser range-finder inputs. Physical dimensions of both the robot and obstacles are known, though the dimensions of obstacles could be approximated by a circular zone which envelops all the points on each obstacle. The initial configuration is such that the robot is not in physical contact with either the workspace boundary or with any of the obstacles.

Given the above, the following are the path-planning objectives:

- **Objective 1** The robot should remain within its workspace at all times.
- **Objective 2** The robot should avoid collisions with moving and stationary obstacles.
- **Objective 3** The robot should reach its destination.

III. REVIEW OF PREVIOUS CONTROLLER DEVELOPMENT

To address the problem of robot navigation through an environment with obstacles, we will modify the navigation function method first proposed by Rimon et al. [4] and later modified by Chen et al. [5].

A. Model Development

Let the robot be defined by its position in the workspace $q(t)$ and a circular envelope of radius r which completely contains the robot, where $q(t) \in \mathbb{R}^2$, $r \in \mathbb{R}^+$, and t is time. It is assumed [5] that the robot can be described by the following kinematic model

$$\dot{q} = u, \tag{1}$$

where $u(t) \in \mathbb{R}^2$ is the control input to the robot. The static destination of the robot is represented by q^*, where $q^* \in \mathbb{R}^2$ and is assumed to be within the workspace.

Two functions, called beta functions, are designed to repel the robot from the boundary of the workspace and from obstacles inside the workspace. Both functions require the use of a curve called the smooth bump function, defined in [13].

Using the bump function, a boundary function is defined, the purpose of which is to repel the robot from the workspace boundary as it gets close to it. Denoted by $\beta_0 : \mathbb{R}^2 \to \mathbb{R}_0^+$, it is a function satisfying

$$\beta_0(q) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\pi \frac{f(q)-h}{1-h}\right)\right] & \text{if} \quad h \le f(q) < 1 \\ 0 & \text{if} \quad f(q) \ge 1 \\ 1 & \text{if} \quad f(q) < h. \end{cases} \tag{2}$$

The first condition implies that the robot has sensed the boundary but is not touching it, the second that the robot has touched the workspace boundary, and the final term indicates that the robot is far away from the workspace boundary. The function f is defined as

$$f(q) = \frac{1}{r_{o0} - r}\left\|q - q_{o0}\right\|, \tag{3}$$

where r is the radius of the robot, r_{o0} is the radius of the workspace, and $q_{o0} \in \mathbb{R}^2$ is the center of the workspace. The parameter h is defined as

$$h = \frac{r_{o0} - r_s}{r_{o0} - r}, \tag{4}$$

where r_s is the sensing range of the robot. Since $r < r_s < r_{o0}$, $0 \le h < 1$.

For each of the n obstacles, the repulsive potential with respect to the robot is called β_i and is defined as $\beta_i : \mathbb{R}^2 \to \mathbb{R}_0^+$, $i = 1, 2, \cdots, n$, a function such that $\beta_i(q) = 0$ means that the robot has made contact with the i^{th} obstacle. In earlier literature [4], [5], the obstacle beta is computed using a simple distance formula between the robot's position and the obstacle's position in the workspace:

$$\beta_i(q) = \left\|q - q_{oi}\right\|^2 - (r + r_{oi})^2, \tag{5}$$

where $q_{oi} \in \mathbb{R}^2$ is the center and $r_{oi} \in \mathbb{R}^+$ the radius of the i^{th} obstacle. When the robot and obstacle touch, the value of β_i goes to zero as per the requirement of the beta function. We discuss the weaknesses of this formulation and propose an improvement in Section IV.

B. Robot Navigation

The navigation function, which encapsulates the forces experienced by the robot, is defined in [4] as

$$\varphi(q) = \frac{K_s \left\|q - q^*\right\|^2}{\left[\left\|q - q^*\right\|^{2\kappa} + G(q)\right]^{1/\kappa}}, \tag{6}$$

where $\kappa \in \mathbb{R}^+$ is a positive constant parameter and $K_s \in \mathbb{R}^+$ is a scale factor used to establish correspondence between the geometry of the potential field and the units of the coordinate system occupied by the robot. $G(q) \triangleq G_0 G_1 \in \mathbb{R}$, and the scalar functions $G_0, G_1 \in \mathbb{R}$ are defined as follows

$$G_0\left(q\right) \quad = \quad \beta_0(q) \tag{7}$$

$$G_1\left(q\right) \quad = \quad \prod_{i=1}^{n} \beta_i\left(q\right), \tag{8}$$

where β_0 and β_i were defined in (2) and (5), respectively.

The convergent path planner was designed in [5] based on the kinematic model in (1)

$$u = -K\left(\frac{\partial\varphi}{\partial q}\right)^T, \tag{9}$$

where K is a a vector of gains and $\frac{\partial\varphi}{\partial q} \in \mathbb{R}^{1\times 2}$ is the partial derivative of $\varphi\left(q\right)$ from (6) with respect to $q\left(t\right)$.

IV. DEVELOPMENT OF ELLIPTICAL OBSTACLE FUNCTION

The original definition of β_i satisfies the requirements of the beta function and has the favorable property that beta changes quadratically as the robot moves toward the obstacle. This rate of change ensures that the robot's approach to an obstacle's current position is strongly repelled. However, this definition does not account for the manner in which an obstacle has been moving or is expected to move. It does not convey the level of threat posed by an obstacle to the robot's approach to the goal. For example, even if the current position of the obstacle is not between the robot and the goal, is there a chance that the obstacle will move in between the robot and target at a later instant?

A. *Using an Ellipse to Create a Position Field*

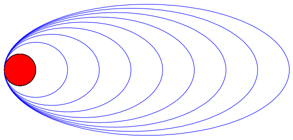


Fig. 1. Transition of the ellipse from the stationary assumption (red circle) to increasing estimates of the velocity of the object (blue ellipses). As the estimated speed increases, the ellipse begins to skew in the estimated direction of motion.

In our formulation, the original beta function for obstacles in (5) is modified to incorporate information about the motion and expected future state of an obstacle. As we shall see in the simulations, this new formulation makes the robot more responsive to the threat posed by the motion of an obstacle, and it skews the gradient of the navigation function in such a way that the region in which the obstacle may be expected to appear is avoided by the robot. To begin the discussion, consider a standard ellipse

$$\frac{\left(x - h_e\right)^2}{a^2} + \frac{\left(y - k_e\right)^2}{b^2} = 1 \tag{10}$$

centered at $\left(h_e, k_e\right)$ and fully containing the obstacle. In this work we assume the ellipse is aligned with the coordinate axes, but it could easily be rotated to make the approach general. The predictive position field is defined using the lengths of the major axis $2a$ and the minor axis $2b$ of the ellipse. When the obstacle is either known to be stationary or nothing is known about its motion, the ellipse collapses into a circle the size of the obstacle to indicate no motion information. As we learn (based on estimates from the vision system) the motion of the obstacle, the circle is skewed in the direction of motion. Therefore, the direction of the major axis indicates the estimated direction of motion, and the length of the major axis indicates the estimated speed. The length of the minor axis then indicates the uncertainty in the direction estimate. Thus, various scenarios are captured by the construction of this elliptical field, enabling it to explain the influence of predictive fields in potential field based path planning. Typical evolution of the elliptical field is illustrated in Fig. 1.

It is beyond the scope of this paper to discuss the methods of arriving at values of a and b. Without losing the generality of the approach, we can assume that sensors and algorithms working in parallel with the path planner can track objects and provide suitable values of a and b to guide the model development for this work.

B. *Constraints on the size of the ellipse*

The elliptical position field is a probabilistic estimate of where we expect the obstacle to be at a future time instant. This estimate should obviously contain the current position of the obstacle, so its radius should not extend outside the perimeter of the ellipse. If the obstacle of radius r_o is placed at the focus of the ellipse, then this means that the radius of the obstacle should be less than the periapsis (the smallest radial distance) of the ellipse:

$$r_o \leq a - \sqrt{a^2 - b^2}, \tag{11}$$

which rearranging terms yields a constraint on the length of the minor axis:

$$b \geq \sqrt{r_o\left(2a - r_o\right)}. \tag{12}$$

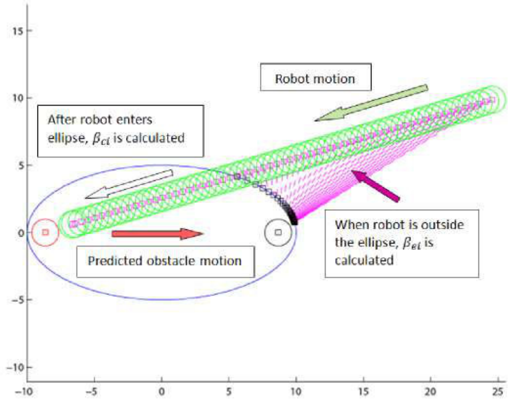
The limiting case of (11) is when the ellipse is a circle, i.e., $a = b$. This leads to the following constraint on the length of the major axis:

$$a \geq r_o. \tag{13}$$

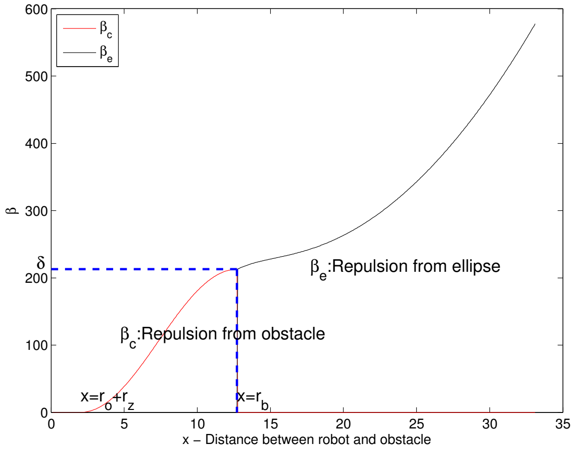
C. *Making the Elliptical Field Relevant to β_i*

Now that the elliptical field has been defined to capture motion trends of an object, the potential in (5) needs a redefinition to give the ellipse importance in this formulation. It needs to be noted that the ellipse is a *probabilistic* region for the presence of the obstacle, and it is possible that the robot finds itself this ellipse. This is not explicitly forbidden, as long as the robot does not touch the measured (or deterministic) position of the obstacle. β_i should go to zero on physical contact between the robot and obstacle, and the robot should be repelled from the obstacle both inside and outside the ellipse.

The requirements for the beta redefinition are



(a) Various positions of the robot (indicated by the green intersecting circles) as it approaches the obstacle along a straight line. The left focus of the ellipse (red) is the actual obstacle position, the right focus (black) is the most likely predicted position. Points of intersection with the ellipse are calculated and the point closer to the robot is selected for β_e computation.



(b) The analytical switch between the β_c and β_e curves takes place when the robot touches the elliptical field. 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. 2. The variation in obstacle beta (b) as the robot approaches an obstacle (a).

- The elliptical position field should provide the obstacle's repulsive force when the robot is outside the ellipse.
- The circular formulation from (5) should come into play only when the robot is inside the ellipse.

From the above list, a modified beta function is proposed as follows:

$$\beta_i = \begin{cases} 0 & \text{robot touches the boundary of an obstacle} \\ \beta_{c_i} & \text{robot is inside the ellipse} \\ \beta_{e_i} & \text{robot is outside the ellipse} \end{cases} \tag{14}$$

where β_{e_i} is the beta function for the robot with respect to the ellipse around the i^{th} obstacle. The obstacle is located at one focus of the ellipse defined in (10). Let this position be q_{o_i}. The obstacle is expected to move along the major axis in the direction of motion to arrive at its predicted position q'_{o_i} at a future time instant t'. Then β_{e_i} is defined as

$$\beta_{e_i}(q) = \left\|q - q'_{o_i}\right\|^2 - (r + d_{e_i})^2 + \delta, \tag{15}$$

where d_{e_i} is the distance from the predicted obstacle position q'_{o_i} to the point q_{re_i} where the line joining the robot position q and the predicted position of the obstacle q'_{o_i} intersects the ellipse. Note that if we set $\delta = 0$, this formula for β_{e_i} guarantees that it goes to zero when the robot touches the outside of the ellipse. The curve described by this formula (with a nonzero δ) can be seen in the right half of Fig. 2(b). We will see later how to define δ.

The requirement for the overall beta β_i is that it should be defined up to the point of contact with the obstacle. To satisfy this, the constant δ allows β_{e_i} to reduce to a non-zero minimum at the point where the robot touches the ellipse. This constant is also the value of β_{c_i} at the point where the robot touches the ellipse. As the robot continues to move into the ellipse toward the target, β_{c_i} reduces to zero as desired. The β_{c_i} curve should be continuous with respect to the β_{e_i} curve to make the resultant beta differentiable throughout its domain.

The requirements of the function β_{c_i} are:

- The function should reach its maximum value at the boundary of the ellipse, i.e., when $\left\|q - q'_{o_i}\right\| = (r + d_{e_i})$.
- The function should reach its minimum value of zero when the robot and the obstacle touch, i.e., when $\|q - q_{o_i}\| = (r + r_{o_i})$.
- The maximum value of the function should be given by the β_{c_i} value when the robot touches the ellipse along a straight line approach to the obstacle. Let this point be q_{re_i}. This gives δ a constant value relative to the line of approach $\delta = \|q_{re_i} - q_{o_i}\|^2 + (r + r_{o_i})^2$. This constant value is added to the ellipse beta when the robot is outside the ellipse, and accounts for the movement of the robot inside the elliptical position field.
- Additionally, the addition of delta to β_{e_i} ensures that the obstacle beta constraint, i.e., beta goes to zero only when the robot and obstacle physically touch, is satisfied even with the addition of the ellipse to the formulation.

This is accomplished by using a mirror image of the bump function described in [13], since the highest point on the curve needs to be further away from the x axis.

Given the above constraints, let the following terms be defined:

$$\begin{aligned} r_b &= \|q_{re_i} - q_{o_i}\| - (r_{o_i} + r) \\ h_c &= r_{o_i} + r \\ \delta &= \|q_{re_i} - q_{o_i}\|^2 - (r + r_{o_i})^2, \end{aligned}$$

where the notations represent:

- r_b - range of the bump function, or the x coordinate where it attains its maximum
- h_c - zero point of the bump function relative to distance of the robot from obstacle

- δ - maximum value of the bump function, added to the ellipse beta

An additional point has to be made about q_{re_i}. When we are outside the ellipse, the point of intersection of the robot and the ellipse is simply the point closest to the robot of the two possible points of intersection. When we calculate the bump function value, we are inside the ellipse and the definition of the point of intersection needs a slight change. As we move closer and closer to the obstacle, it is possible that the point of intersection on the other side of the obstacle is the nearer point of intersection. This changes δ for the bump function and the desired shape of the β curve is lost because of this. To ensure this does not happen, we introduce the unit vector from the obstacle to the robot, $\bar{n}_{re_i}$. The point of intersection is then defined as the one which is along the vector $\bar{n}_{re_i}$.

The bump function is then defined as:

$$\beta_{c_i}(x) = \begin{cases} 1 & r_b \leq x \\ 0 & 0 \leq x < h_c \\ \frac{\delta}{2}\left[1 - cos\left(\pi \frac{x-h_c}{r_b-h_c}\right)\right] & h_c \leq x < r_b \end{cases} \tag{16}$$

The bump function then gets the following values. At $x = h_c$, the obstacle and robot touch and β_{c_i} goes to zero. At $x = r_b$, the elliptical position field and the robot touch and β_{c_i} gets its maximum value of δ. Beyond r_b, the maximum value of the β_{c_i} term, δ, adds to the β_{e_i} term which begins to dominate the overall β function. Therefore the value of β_{e_i} approaches δ instead of 0 as the robot moves towards the ellipse.

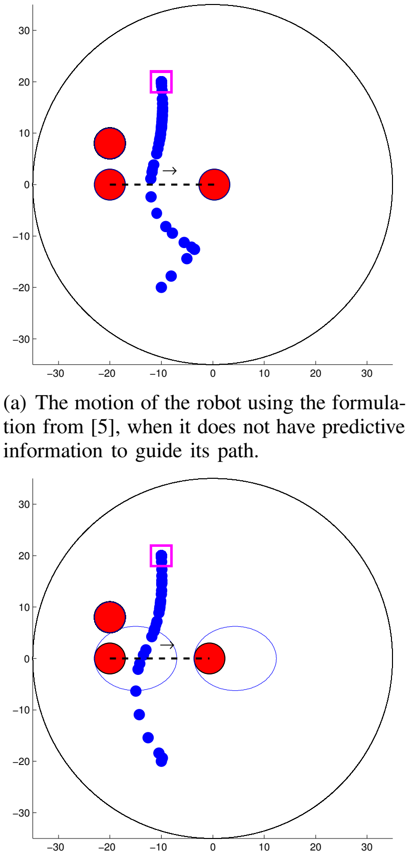
With these definitions for β_{c_i} (16) and β_{e_i} (15), the overall definition of β_i (14) is consistent with the requirements of the obstacle function. See Fig. 2.

V. SIMULATION RESULTS

We used Simulink (Mathworks Inc., Natick, MA) to simulate the proposed path planner in such a way that it would be possible to directly compare our method against previous results from Chen et al. [5]. This is made possible by the fact that when the position field is forced to a circle the size of the obstacle, the equation in (15) reduces to (5). Our hypothesis was that the predictive position fields would make it possible for the robot to converge to the target following a more optimal trajectory than that followed by purely static workspace information. Moreover, the predictive look-ahead should make it possible for the robot to stay away from the predicted path of the obstacle, thus lowering its chances of a collision with the obstacle.

We tested our approach using two hypothetical scenarios:

1) An obstacle initially obstructs the straight line path from robot to goal, but it begins to move out of the way as the simulation progresses.
2) An obstacle is initially at a distance from the straight line trajectory from robot to goal, but it moves to obstruct the path as the simulation progresses.



(a) The motion of the robot using the formulation from [5], when it does not have predictive information to guide its path.

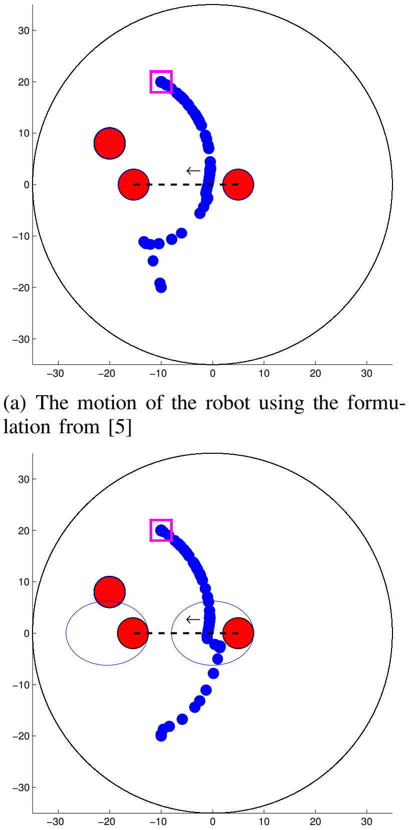
(b) The motion of the robot using our proposed approach with the elliptical beta function, which guides the robot behind the obstacle using the predicted position of the obstacle

Fig. 3. Two obstacles, one moving and one stationary, occupy the workspace as the robot moves from the start position to its goal at the top of the figure, marked by the rectangle. The moving obstacle moves from left to right (as marked by the arrow). Without predictive information (a), the robot first attempts to move in the direction of danger (in front of the moving obstacle) and takes a longer path. The path of the robot is represented by a succession of smaller circles. The path is more optimal as predictive information is added (b).

Both cases are tested with and without the predictive position field surrounding the obstacle. The setup of the workspace is described as follows:

- Robot with boundary sensing zone $r_s = 5$ and radius $r = 1$ is initially located at $(-10, -20)$.
- Goal is located at $(-10, 20)$.
- Stationary obstacle with radius $r_{o1} = 3$ is located at $(-20, 8)$. The stationary nature of the obstacle causes the predictive position field around it to shrink to a circle with the same radius as the obstacle.
- The workspace is centered at $(0, 0)$ with a radius of $r_{o0} = 35$.
- The predictive position field of the moving obstacle of radius $r_{o2} = 3$ is described by an ellipse with parameter $a = 8$ in both cases.

In Scenario 1, the obstacle starts at $(-20, 0)$ and travels 20 units in the workspace at a constant velocity. The sense of its motion is such that it is moving out of the way of the robot's



(a) The motion of the robot using the formulation from [5]

(b) The motion of the robot using our proposed approach with the elliptical beta function, which prevents the robot moving toward the path of the obstacle

Fig. 4. The motion of the robot in a setup similar to Fig. 3 except that this time, the obstacle moves from right to left (as marked by the arrow). Without predictive information (a), the robot first attempts to move in the direction of danger (in front of the obstacle) and takes a longer path. The robot goes around the obstacle as predictive information is added (b).

path to goal. Without the use of a predictive position field, we observe that the robot (in Fig. 3(a)) tries to move around the obstacle. This causes it to move toward the path of the obstacle and forces a correction in its path approximately midway through its trajectory. However, when the position field is added, the path planner is able to sense that the more optimal path to goal would actually be behind the obstacle, as seen in Fig. 3(b). The trajectory traced as a result is much more intuitive than the first case. A similar improvement was observed in Scenario 2 (see Fig. 4).

The scale factor K_s from the navigation function (6) is set to a constant value of $1e10$ for our simulations, resulting in velocities of around 1 unit/sec for the robot. The gains from the navigation (6, 9) are set at $\kappa = 4.5$ and $K = 1.2$. The gains κ and K remain unchanged regardless of the position of the goal and of the obstacles. However, they need to be tuned on changing the number of obstacles in the workspace for the robot to converge to goal.

Results from multiple trials corroborated the hypothesis that the robot was able to successfully converge to the goal while improving on its trajectory to goal after the elliptical position field was defined for obstacles in the workspace.

VI. CONCLUSIONS AND FUTURE WORK

We have shown that it is possible to encode probabilistic data and motion information into the conventional formulation of potential fields. A path planner with a predictive position field has been shown to work favorably for moving obstacles. However, it is essential to recognize that the elliptical shape of the position field is not a prerequisite for predictive path planners to work within the framework of navigation functions. It will be interesting to investigate other geometric representations of position fields which may be capable of more accurately representing the behavior of specific categories of obstacles. This will change the obstacle beta function, but the overall framework of the solution will stay unchanged. This flexibility to new geometric models is one of the strengths of potential field based planners which we will continue to investigate. One of the deficiencies of this method is sensitivity to parameter values in the system. With automatic parameter estimation and tuning, we will be able to make the planner more flexible to dynamic environments and get closer to realizing a mobile robot capable of navigating through crowds.

REFERENCES

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

[2] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, “Visual navigation and obstacle avoidance using a steering potential function.” *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 288–299, 2006.

[3] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, “Nonholonomic navigation and control of cooperating mobile manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 53–64, 2002.

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

[5] 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.

[6] 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.

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

[8] 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.

[9] P. Henry, C. Vollmer, B. Ferris, and D. Fox, “Learning to navigate through crowded environments,” in *International Conference on Robotics and Automation*, 2010.

[10] F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider, “Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps,” in *IROS*, 2007, pp. 1260–1265.

[11] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, “Learning motion patterns of people for compliant robot motion,” *I. J. Robotic Res.*, vol. 24, no. 1, pp. 31–48, 2005.

[12] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Probabilistic motion planning among moving obstacles following typical motion patterns,” in *IROS*, 2009, pp. 4027–4033.

[13] R. Saber and R. Murray, “Flocking with obstacle avoidance: cooperation with limited communication in mobile networks,” in *Proc. IEEE Conference on Decision and Control, 2003*, vol. 2, 2003, pp. 2022 – 2028 Vol.2.