Abstract

This paper introduces a novel motion planning algorithm for stochastic scenarios. We extend the concept of a navigation function to such scenarios. Our main idea is to consider both the Gaussian distribution probabilities of the players’ locations and disc (or star sets) geometry of the objects operating in the work space. We do so by formulating a probability density function that encloses both. We use the PDF to define a metric between the robot, the obstacles and the configuration space boundary. In order to define the navigation function we formulate a safe probability value for collision. By analytically investigating the PDF we find a convenient approximation for a safe distance in the sense of that metric. We prove that the resulting map is a navigation function and demonstrate our algorithm for various scenarios.

1 Introduction

Motion planning for a moving robot has been extensively studied in the last three decades [1], [2], [3]. Ideally one can assume that properties describing the robot movement, the environment and the obstacles are perfectly known. However, those parameters often possess a substantial random factors (these are referred to as random variables) due to measurement noises, physical process etc. A common line of action in order to handle uncertainties involved in robotic path planning is to modify deterministic algorithms [4], [5], [6]:

Potential Field Method [7], [8] and Navigation Function (NF) path planning [9] are some of the most known methods due to their mathematical elegance and simplicity [10], [11], [12]. Both methods are closely connected and yield a closed loop control law, namely, the path and the control signal (motion direction and speed) are given simultaneously. Conversely, other methods provide these in a two folded manner, (e.g. [2]). One more important advantage of NF algorithm is its asymptotic convergence property [9], [8]. Thus, we shall focus our attention in this paper on the NF algorithm.

NF path planning algorithm was first published by Koditschek and Rimon [13]. A navigation function is a continuous smooth function which take a zero value at the target point while on the environment’s and obstacles’ boundaries the NF gets a unity value. Moreover, in order to ensure a solution, the NF critical points are all non-degenerated (i.e. there are no Plateau areas in which the gradient of the NF vanishes). Note that other concepts of navigation functions have been proposed for the purpose of robot path planning (see for example [14]) which numerically solve a discrete differential equation to find a NF. In this paper we shall modify Rimon an Koditschek’s algorithm to stochastic scenarios.

2 Problem Formulation

Let $C$ be a robot configuration space. We assume that $C$ is a subset of a smooth Euclidean manifold which we shall assume here to be $\mathbb{R}^n$. We denote the robot location by $x_r \in C$ and the set of obstacles with fixed locations by $x_i \in O \subset C$, where $O$ denotes the set of obstacles. Nevertheless, we do not have a deterministic knowledge about the $x_r, x_i \in O$, rather
then a set of probabilities \( \{ Pr_i : \mathbb{R}^n \to \mathbb{R} \}_{i \in \mathcal{O},r} \) where \( Pr_i(x) = Pr(x_i \in \mathcal{O} = x) \) is the probability for the \( i \)-th obstacle to be at \( x \in \mathbb{R}^n \). Similarly \( Pr_r(x) = Pr(x_r = x) \) is the probability for the robot to be at \( x \in \mathbb{R}^n \). We assume that \( x_r,x_i \in \mathcal{O} \) are generated using a sub-optimal nonlinear filter, which therefore is the source for uncertainties. For example a GPS navigation system in clear areas yields an average of 10 meters errors [15]. So, our main problem is formulated as follows:

Generate an optimal robot path, reducing the probability for obstacles-robot collision and pursuing the shortest path to the target configuration (\( q_d \)). Furthermore, the probability for collision should not exceed the highest allowable probability- \( \Delta \).

Here, \( \Delta \) indicates the highest allowable probability for collision: It is expected that in some scenarios the robot would follow the shortest path on the expense of the collision avoidance probability. Thus, in order to limit the probability for collision we introduce the user-determined \( \Delta \).

3 A Probability density function (pdf) for collision

We shall apply a modified NF in order to incorporate the robot and the obstacles uncertainty. We call such a function the Probability Navigation Function or the Stochastic Navigation Function (PNF). The PNF describes the probability for the robot and an obstacle to collide at a given point as well as the distance to the goal position. Since we would like the algorithm to be realistic as possible, the robot and obstacles are given a finite disc shape.

Remark: [9] show how one can generalize an NF to the case of star-shaped obstacles located in a star-world. This is done by defining a diffeomorphism on the configuration space, transforming the discs into star-shapes. Next the authors prove that composing any NF \( \varphi \) with the diffeomorphism map results with a NF defined on the deformed space.

Thus, we shall incorporate the robot’s and obstacles’ shapes to a probability map. The path is then chosen to be the PNF gradient. A common technique taken when dealing with path planning problems (see for example [16], [17]), is to define the free configuration space \( C_{\text{free}} \) (i.e. a subset of \( C \subset \mathbb{R}^n \) where the robot can travel without colliding with obstacles, excluding the boundary itself). To this end, a prevalent method is to define \( C_{\text{free}} \) as the complement of \( C_{\text{obs}} \), the union of the Minkowski sums of the robot with the set of obstacles. Intuitively, the obstacles are expended by the robot’s volume, while the robot is taken as a point mass. Explicitly, denoting by \( A \) the set of vectors defining the robot’s volume, which are the vectors measured from its center of mass to any point on the robot body, and setting \( B \) to be the set of vectors defining the geometry of all obstacles measured from the origin to their body points, we can write:

\[
C_{\text{obs}} = B * (-A) = \{ b - a | a \in A, b \in B \}
\]

(1)

(we use Astrix to denote both Minkowski sum and Convolution operation). Note that in order to measure the distance of a point inside the robot from point inside the obstacle, one should first rotate the robot at 180° (the minus sign in Eq.1). The sets \( A,B \) are sub spaces of \( C \) making \( B * (-A) \) overwhelming large. One way to overcome this is to confine calculations to an intermediate time step (i.e. \( A,B \subset W \subset \mathbb{R}^n \)).

We shall now incorporate both the geometries of the robot and the obstacles together with their location probabilities. We shall do so, in three stages:

3.1 Convolution of obstacle’s geometry with the robot’s geometry.

Let us define the \( i \)-th obstacle by a disc function:

\[
\text{obs}_i(x) = \begin{cases} 1 \: \text{if} \: \| x - \hat{x}_{\text{obs}} \| \leq r_{\text{obs}} \\
0 \: \text{otherwise} \end{cases}
\]
where \( \hat{x}_{\text{obs}} \) and \( r_{\text{obs}} \) are the estimated location of the obstacle’s center and his radius respectively. Similarly we define the robot position (as a \( r_{\text{rob}} \) radius disc, located at \( \hat{x}_{\text{rob}} \)) by:

\[
\text{rob}(x) = \begin{cases} 
1; & \|x - \hat{x}_{\text{rob}}\| \leq r_{\text{rob}} \\
0; & \text{otherwise}
\end{cases}
\]

The Minkowski sum of \( \text{obs}_i(x) \) and \( \text{rob}(x) \) (also known in the literature as Convolution) yields:

\[
\text{obs}_i(x) = \text{rob}(x) * \text{obs}_i(x)
\]

\[
\text{obs}_i(x, R) = \begin{cases} 
1; & \|x - \hat{x}_{\text{obs}}^i\| \leq R \\
0; & \text{otherwise}
\end{cases}
\]

where: \( R = r_{\text{obs}} + r_{\text{rob}} \). This describes the \( i \)-th obstacle while the robot ”transforms” into a point-mass located at the estimated location of the robot:

\[
\text{rob}(x) = \delta(x - \hat{x}_{\text{rob}})
\]

here \( \delta \) stands for Dirac’s delta function.

### 3.2 Convolution of Gaussian functions.

We would like to implement the above on a stochastic scenario. Thus, for simplicity, let us first consider a point-mass obstacles and a point-mass robot with given probability density functions embedded in \( \mathbb{R}^n \) for arbitrary configuration space dimension \( n \). Eq.\[\text{1}\] defines a map \( \varphi_i : C \to \mathbb{R} \). That is, we replaced the Minkowski sum by a convolution of the probability functions. So, the binary nature of \( C_{\text{obs}} \), which indicates if a configuration is free or occupied, is replaced by a continuous function - the probability for a grid cell (with a non zero volume) in the work space to be free of obstacles at a given time step. Note that the minus sign in the integrand \( \mathbb{1} \) can be replaced by a plus sign since we assume a (symmetric) Gaussian distribution functions for both the robot and obstacles locations:

\[
f_i(x) = p(x|\hat{x}_{\text{obs}}^i, \Sigma_{\text{obs}}^i) * p(x|0, \Sigma_r) = \\
= \int_{-\infty}^{\infty} p(x|\hat{x}_{\text{obs}}^i, \Sigma_{\text{obs}}^i) p(x-x'|0, \Sigma_r) dx' \tag{3}
\]

According to Eq.\[\text{3}\] the probability for a collision at \( x \) is simply the probability for the obstacles and the robot to be simultaneously at the same location while the expected value of the robot’s position is fixed at the origin: Eq.\[\text{3}\] can explicitly be written as:

\[
f_i(x) = \int_{M} \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{\text{obs}}^i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\hat{x}_{\text{obs}}^i)^T(\Sigma_{\text{obs}}^i)^{-1}(x-\hat{x}_{\text{obs}}^i)} \cdot \\
\cdot \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_r|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-x')^T\Sigma_r^{-1}(x-x')} dx' =
\]

following \[\text{18}\] Eq.\[\text{3}\] concludes to give the probability function of the \( i \)-th obstacle location:

\[
f_i(x) = \frac{(2\pi)^{-\frac{n}{2}}}{|\Sigma_{\text{obs}}^i + \Sigma_r|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\hat{x}_{\text{obs}}^i-x')^T(\Sigma_{\text{obs}}^i + \Sigma_r)^{-1}(x-\hat{x}_{\text{obs}}^i-x)}
\]

with expectation \( \hat{x}_{\text{obs}}^i \) and covariance \( \Sigma_i = \Sigma_{\text{obs}}^i + \Sigma_r \). We denote the distributions for the robot’s and obstacles’ locations by:

\[
x_{\text{Robot}} \sim \mathcal{N}(\hat{x}_{\text{Robot}}, 0) = \delta(x - \hat{x}_{\text{rob}}) \tag{4}
\]

\[
x_{\text{obs}}^i \sim \mathcal{N}(\hat{x}_{\text{obs}}^i, (\Sigma_r + \Sigma_{\text{obs}}^i)) = p(x|\hat{x}_{\text{obs}}^i, \Sigma_i) \tag{5}
\]

### 3.3 Convolution of probability density function and geometry functions.

Note that in both Eq.\[\text{1}\] and Eq.\[\text{2}\] the robot is represented by a point-mass. We shall therefore investigate the probability for collision of a disc shaped obstacle (see Eq. \[\text{2}\]) with the point mass robot (as is often done in motion planning problems). The location of any point \( \hat{v} \) of the obstacle relative to a fixed point is a deterministic value which can be defined as random variable by: \( p_{\hat{v}}(x|\hat{x}_{\text{obs}}^i) = \delta(x - (\hat{x}_{\text{obs}}^i + x_{\hat{v}})) \). Note that the \( i \)-th obstacle center location \( x_{\text{obs}}^i \) measured at global coordinate system, and the \( x_{\hat{v}} \) measured at the local coordinates system, are independent. The convolution of these functions, which yields the probability distribution function for an infinitely small portion \( \hat{v} \in \text{obs}_i(x) \), is:

\[
p_{\hat{v}}(x) = p_{\hat{v}}(x|\hat{x}_{\text{obs}}^i) * f_i(x) = \\
= \int_{-\infty}^{\infty} \delta \left(x - (\hat{x}_{\text{obs}}^i + x_{\hat{v}} - \tau)\right) f_i(\tau) d\tau
\]

\[3\]
Applying (19) (cf. Pg. 53) yields:

\[ p_{\vec{v}}(x) = \int_{-\infty}^{\infty} \delta (\tau - (x - (\hat{x}_{\text{obs}}^i + x_\vec{v}))) f_i(\tau) \, d\tau \]

\[ p_{\vec{v}}(x) = f_i(x - (\hat{x}_{\text{obs}}^i + x_\vec{v})) \]

Thus:

\[ p_{\vec{v}}(x) = \frac{1}{|2\pi \Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2} (x - (\hat{x}_{\text{obs}}^i + x_\vec{v}))^T \Sigma_i^{-1} (x - (\hat{x}_{\text{obs}}^i + x_\vec{v}))} \]

Since we would like to avoid collision with any of the obstacle’s points, the density function for such a collision is given by the integral:

\[ p_{\text{tot}}(x, R) = \frac{1}{A} \int_{\mathbb{R}^n} \tilde{\omega}_i(x_\vec{v}, R) p_{\vec{v}}(x_\vec{v} - x) \, dx_\vec{v} \]

Where \( A \) is the total volume of \( p_{\vec{v}} \). It is easy to see that:

\[ p_{\text{tot}}(x, R) = \frac{1}{A} \tilde{\omega}_i(x_\vec{v}, R) \ast f_i(x) \quad (6) \]

### 3.4 Convolution of n dimensional disk with Gaussian distribution.

Consider Eq.6 - a convolution of a normal distribution \( G(\vec{r}) \) (with a diagonal covariance of the form: \( \Sigma = \sigma I \)) and a disc \( D(\vec{r}) \):

\[ C(\vec{r}) = D(\vec{r}) \ast G(\vec{r}) \quad (7) \]

For an arbitrary Gaussian, \( \Sigma \) can be taken as a diagonal matrix with all its entries equal to the maximal eigenvalue of the covariance matrix.

Assume the disc is centered at the origin and the Gaussian is at \( \vec{g} \in \mathbb{R}^n \):

\[ D(\vec{r}) = \begin{cases} 1, ||\vec{r}|| \leq R \\ 0, \text{otherwise} \end{cases} \]

\[ G(\vec{r} - \vec{g}) = \left( \frac{1}{2\pi \sigma} \right)^{\frac{n}{2}} e^{-\frac{1}{2\sigma} ||\vec{r} - \vec{g}||^2} \]

We shall now generalize Plessor [20], for arbitrary Euclidean ambient space and arbitrary \( \sigma \). Eq.7 may be formulated as:

\[ C(\vec{g}) = \int_{\mathbb{R}^n} G(\vec{r} - \vec{g}) D(\vec{r}) \, d\vec{r} \]

The Jacobian for the polar form of the above is \( J = r^{n-1} \prod_{k=1}^{n-2} \sin^k(\phi_{n-1-k}) \) (see [21], Pg. 65-66) and thus:

\[ C(\vec{g}) = \int_{0}^{R} \frac{2\pi r^{n-1}}{(2\pi \sigma)^{\frac{n}{2}}} \prod_{k=1}^{n-2} \sin^k(\phi_{n-1-k}) \, d\phi_{n-1-k} \]

\[ \cdot \int_{0}^{\pi} e^{-\frac{r^2 - a^2 + 2rg \cos(\phi)}{2\sigma}} \sin^{n-2}(\phi_1) \, d\phi_1 \, dr \]

which may be rewritten as:

\[ C(\vec{g}) = \frac{e^{-\frac{r^2}{2\sigma}}}{(2\pi \sigma)^{\frac{n}{2}}} \int_{0}^{R} 2\pi r^{n/2} r^{n-1} e^{-\frac{r^2}{2\sigma}} \]

\[ \cdot \left[ \frac{1}{\Gamma((n-1)/2)} \int_{0}^{\pi} e^{\frac{rg \cos(\phi)}{\sigma}} \sin^{n-2}(\phi_1) \, d\phi_1 \right] \, dr \]

where \( r = ||\vec{r}|| \) and \( g = ||\vec{g}|| \). Following Abramowitz ([22] Eqs. 9.6.10, 9.6.18) one have:

\[ C(\vec{g}) = (2\sigma)^{-\frac{n}{2}} e^{-\frac{r^2}{2\sigma}} \sum_{k=0}^{\infty} \left( \frac{g}{2\sigma} \right)^{2k} \frac{1}{k!} \]

\[ \cdot \left[ \frac{1}{\Gamma((k+n)/2)} \int_{0}^{R^2} (r^2)^{k+n-1} e^{-\frac{r^2}{2\sigma}} \, dr \right] \]

Recall that:

\[ P(a, b) = \frac{1}{\Gamma(a)} \int_{0}^{b} e^{-x} x^{a-1} \, dx \]

where \( P(a, b) \) is the Normalized Incomplete Lower Gamma Function. Rearranging terms results with the equality:

\[ \frac{1}{\Gamma(s)} \int_{0}^{x} e^{-t} t^{s-1} \, dt = \frac{a^{s-1}}{\Gamma(s)} \int_{0}^{x/a} e^{-z} z^{s-1} \, dz = P\left(s, \frac{x}{a}\right) a^s \]

Finally, for an \( n \)-dimensional disk-shaped obstacles
distributed normally, Eq 8 becomes:
\[
p_{tot}(\bar{x}, R, \sigma) = e^{-\|x\|^2/2\sigma} \sum_{m=0}^{\infty} \frac{1}{m!} P\left(2\frac{m+n}{2}, \frac{R^2}{2\sigma}\right)
\]
where \(x\) is the location vector.

Next, we will need to calculate the Gradient and the Hessian of \(p_{tot}\). We shall consider them now:

### 3.5 The Gradient and the Hessian of \(p_{tot}\)

\[
\nabla p_{tot}(\bar{q}, R, \sigma) = \frac{\partial p_{tot}(\bar{q}, R, \sigma)}{\partial \|q\|^2} \frac{\partial \|q\|^2}{\partial q} = 2\bar{q}e^{-\|\bar{q}\|^2/2\sigma} \sum_{m=1}^{\infty} \frac{\|q\|^2(m-1)}{(2\sigma)^m} P\left(m + \frac{n}{2}, \frac{R^2}{2\sigma}\right) - \sum_{m=0}^{\infty} (2\sigma)^{-m-1} \frac{\|q\|^{2m}}{m!} P\left(m + \frac{n}{2}, \frac{R^2}{2\sigma}\right) = 2\bar{q}R e^{-\|\bar{q}\|^2/2\sigma} I_0\left(\frac{\|q\| R}{\sigma}\right)
\]

The Hessian is:

\[
\nabla^2 p_{tot}(\bar{q}, R, \sigma) = e^{-\|\bar{q}\|^2/2\sigma} \frac{\bar{q}^T R^2}{2\pi \sigma^{n+2}} \left(I_0\left(\frac{\|q\| R}{\sigma}\right) - \frac{R}{\|q\|} I_1\left(\frac{\|q\| R}{\sigma}\right)\right)
\]

### 3.6 Minimal permitted collision probability

In order to ensure a reasonably safe movement, in the sense of probability for collision, we would like to limit the collision probability to a predefined value \(\Delta\). In other words, we are interested in a closed curve \(\Psi \subset \mathbb{C} \subseteq \mathbb{R}^n\) such that:

\[
\frac{1}{\Psi} \int_{\Psi} \tilde{p}_{obs}(x, R) * f_i(x) dx = \Delta
\]

We pursue a safety distance \(R_{\Delta}\) from \(\tilde{p}_{obs}\) that will ensure probability for collision of at most \(\Delta\):

\[
\Delta = \frac{1}{A} \int_{\mathbb{R}^n} C(\xi) d^n\xi
\]

Where \(\xi = |\bar{r}|\) is the distance from the origin and \(A\) is the normalization factor, which by expansion yields:

\[
\Delta = \frac{1}{A} \int_{\mathbb{R}^n} e^{-\|\bar{q}\|^2/2\sigma} \sum_{m=0}^{\infty} \frac{g^{2m}}{(2\sigma)^m m!} P\left(m + \frac{n}{2}, \frac{R^2}{2\sigma}\right) d\bar{q}
\]

Again, using [21] we get:

\[
\Delta = \frac{1}{A} \int_{\mathbb{R}^n} \pi^{(n-1)/2} \sum_{m=0}^{\infty} P\left(m + \frac{n}{2}, \frac{R^2}{2\sigma}\right) \frac{1}{(2\sigma)^m m!} \int_0^{R_d^2} (g^2)^{m+2\frac{n}{2}-1} e^{-\|\bar{g}\|^2/2} \sin^{n-2} \phi d\phi dg^2
\]

Denoting the Double Factorial by \(X!!\), one may easily compute the following equality:

\[
l(n) = \int_0^\pi \sin^n \phi d\phi = \frac{(n-1)!!}{n!!} \begin{cases} \pi, & n \text{ mod } 2 = 0 \\ 2, & n \text{ mod } 2 = 1 \end{cases}
\]

and the value of \(A\) can be simplified due to Fubini’s theorem:

\[
A = \int_{\mathbb{R}^n} D(r) * G(r) dx = \int_{\mathbb{R}^n} D(r) dx \int_{\mathbb{R}^n} G(r) dx = V_n(R) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} R^n
\]

Finally Eq [12] may be written as:

\[
\Delta = \frac{\Gamma(\frac{n}{2}+1)}{\sqrt{\pi R^n \Gamma((n-1)/2)}} \cdot \sum_{m=0}^{\infty} \frac{1}{m!} P\left(m + \frac{n}{2}, \frac{R^2}{2\sigma}\right) \gamma\left(m + \frac{n}{2}, \frac{R_d^2}{2\sigma}\right)
\]

(13)
where $\gamma(a, b) = \int_0^b e^{-t} t^{a-1} dt$, is the Lower Incomplete Gamma Function, which can be approximately solved for $R_\Delta$.

We can now calculate $p_\Delta = p_{tot}(x)$ for all $x \in \Psi$ (Eq. 8 above):

$$p_\Delta = p_{tot}(R_\Delta, R, \sigma)$$  \hspace{1cm} (14)

4 Probability Navigation Function

In this section we present the approach for optimal path planning in uncertain environments. The following discussion is based on the deterministic Navigation Function suggested by Rimon and Koditschek [13] which we shall extend to stochastic scenarios later on in Subsection 4.2.

4.1 Deterministic Navigation Function

Rimon and Koditschek formulated a computation scheme for static deterministic environment. Their aim was to formulate a Navigation Function (NF), that construct a safe trajectory to the target position (but compare with other NF [12], [14], [24]). Our scheme will show how to find such a function for path planning, coupled with a feedback control law calculation; the NF resulting path avoids obstacles and is confined to a given region; finally, convergence is guaranteed from any initial condition (subject to a correct parameter selection). Rimon and Koditschek define the NF to the target position $q_d$, at a point $q \in C$ as:

$$\varphi_k(q) = \frac{\gamma_d^2}{\left[ \gamma_d^2 + \beta(q) \right]^{\frac{k}{2}}}$$  \hspace{1cm} (15)

where $k$ is a constant. $\gamma_d(q) = ||q - q_d||^2$ and $\beta(q)$ is:

$$\beta(q) = \prod_{i=0}^{N_{obs}} \beta_i(q)$$  \hspace{1cm} (16)

where:

$$\beta_i(q) = \begin{cases} 0 & ; \; p_i(q) > p_{\Delta i} \\ p_{\Delta i} - p_i(q) & ; \; otherwise \end{cases}$$  \hspace{1cm} (17)

Here $q_0$ defines the center of the permissible area, which we consider of as the coordinates’ origin, while $q_i$ for all $i \in O$, is the center of the $i$-th obstacle.

The nominator at Eq. 15 is defined in such a way that the robot is attracted to the goal position, while the denominator ensures obstacle avoidance.

Considering a stochastic scenario, we would like to minimize the probability for a collision while maintaining the shortest path length. In the deterministic scenario, $\beta_i(q)$ is a function of the distance between $q$ and the obstacle’s boundary. Our main idea is to replace $\beta_i$ by a function which is based on the probability for collision at $q$. We would also require to set some threshold for collision probability, which we get by replacing the obstacles’ geometric edge by the edge $\Psi$ - discussed above (see Eq. 11).

4.2 Modification of $\beta_i(q)$ for the stochastic scenario.

We shall now modify $\beta$ to fit an uncertain environment. In the deterministic scenario, $\varphi_K$ decreases the distance to the goal position while keeping the robot away from the obstacles (up to sliding along their boundaries). However, under uncertainty conditions, the only information we have on the obstacles’ positions is statistical so a reasonable probabilistic analogue is to limit the probability for collision to predetermined value - $\Delta$.

In order to do so, we replace the original $\beta$ with the probabilistic value $p_i(q)$ - the probability density function at $q$ (discussed in Section 3.3). This equal to $p_{tot}(q - q_i)$ (see Eq. 8) computed for the $i$-th obstacle ($i \in O$):

$$\beta_i(q) = \begin{cases} 0 & ; \; p_i(q) > p_{\Delta i} \\ p_{\Delta i} - p_i(q) & ; \; otherwise \end{cases}$$  \hspace{1cm} (18)

where, $p_{\Delta i} = p_{tot}(R_{\Delta i}, R_i, \sigma_i)$ Thus, $\beta_i$ vanishes (and so is $\beta$) on the extended obstacle’s boundary.
defined by $R_\Delta$, i.e. where the probability for collision is less than $\Delta$ (see Figure 1).

As for the workspace boundary, $\beta_0$ is:

$$
\beta_0(q) = \begin{cases} 
0 & : \quad p_0(q) > p_{\Delta_0} \\
-\Delta_0 + p_0(q) & : \quad \text{otherwise}
\end{cases}
$$

Note that $p_0$ and $p_{\Delta_0}$ refer to the external boundary computed based on the probability density function of the robot only. Note also that $p_{\Delta_0}$ is computed as in Eq. 13 substituting $1 - \Delta$ instead of $\Delta$.

**Remark:** The resulting functions $\beta_i$ for $i \in \{0, O\}$ is positive and so is $\beta$.

### 5 Is $\varphi$ a Navigation Function?

A map is said to be a navigation function if satisfies the following conditions:

1. It is analytic in all $q \in \mathcal{C}_{free}$.
2. It is polar throughout $\mathcal{C}$, with single minimum at $q_d \in \mathcal{C}_{free}$.
3. It is morse on $\mathcal{C}_{free}$.
4. It is admissible on $\mathcal{C}_{free}$.

The navigation function is the composition:

$$
\varphi = \sigma_d \circ \sigma \circ \hat{\varphi}
$$

where: $\sigma_d(x) = (x)^{1/k}$ ; $\sigma(x) = \frac{1+x}{1+x^k}$ and $\hat{\varphi} = \frac{\varphi}{\Delta^3}$

In this paper we change only $\hat{\varphi}$. So, according to proposition 2.7 in [13] it suffices to verify conditions (1)-(4) only for $\hat{\varphi}$ (note that the forth requirement directly follows from the definitions).

We shall now prove that $\hat{\varphi}$ constitutes an Navigation Function. In Proposition 5.1 we shall prove that $\varphi$ attains a minimum at the destination $q_d$. In order for our motion planning scheme to converge, $\varphi$ must not have critical points on $\partial \mathcal{C}_{free}$ (i.e. points were the gradient vanishes), we shall prove that in Proposition 5.2 which leaves us with the interior of $\mathcal{C}_{free}$.

For convergence we require that all critical points in $\mathcal{C}_{free}$ will be non-degenerated. We refer to this region as "near the $i$-th obstacle" and denote it by $B_i(\epsilon) = \{q\mid 0 < \beta_i(q) < \epsilon\}$, so obviously (since the obstacles do not intersect) there exist $\epsilon > 0$ such that $\beta_i(q) \cap \beta_i(q) = \emptyset$ for all $i \neq j \in \mathcal{O}$ and $\beta_i(q) \cap q_d = \emptyset$ for all $i \in \mathcal{O}$. In other words, we need to prove that:

$$
\mathcal{C}_{free} = \{q|\beta_i(q) \geq \epsilon, \forall i \in \mathcal{O}\} \cup B_0(\epsilon) \cup \bigcup_{i=1}^{N_{obs}} B_i(\epsilon)
$$

has no non-degenerate critical points in either regions (indicated by the three components). Propositions 5.3, 5.5 respectively proves that the first and second regions have no critical points, while Proposition 5.4 proves that all critical points near the obstacles are not local minimum points. In proposition 5.6 we conclude that $\varphi$ is a Morse function by showing that near the obstacles it is non-degenerate as well.

**Proposition 5.1.** The destination region located at $q_d$ is a local minimum of $\varphi$. 

![Figure 1: The partition of the configuration space:](image)
Proposition 5.2. All the critical points of $\varphi$ are in the interior of $C_{\text{free}}$.

**Proof.** We focus our attention on some point $q' \in \partial C_{\text{free}}$. Obviously $\beta_i = 0$ for a certain $i \in O$, and $\beta_j > 0$ for the rest $j \neq i$. Differentiating yields:

$$
\nabla \varphi(q') = \frac{1}{\gamma_d} \left( \nabla \gamma_d - \frac{1}{k} \gamma_d^{-1-k} (k \gamma_d^{-1} \nabla \gamma_d + \nabla \beta) \right)_{q'}
$$

which proves the proposition since:

$$
\nabla \beta_i(q') = -\nabla p_{\text{tot}}(q' - q_i, R_i, \sigma_i) \neq 0
$$

As $k$ increases the critical points of $\hat{\varphi}$ approach those of $\gamma_d$. We shall now show this by proving that there are no critical points far away from the obstacles:

**Proposition 5.3.** For every $\varepsilon > 0$ there exist $N(\varepsilon) \in \mathbb{R}$ such that for all $k \geq N(\varepsilon)$, $\varphi$ has no critical points in $\{q|\beta_i(q) \geq \varepsilon, \forall i \in O\}$.

**Proof.** Note that if $\hat{\varphi}$ has no critical points at a given region, so will $\varphi$. Thus we shall prove the proposition for $\hat{\varphi}$.

A critical point satisfies:

$$
\nabla (\hat{\varphi}) = \frac{\gamma_d^{k-1} (k \beta \nabla \gamma_d - \gamma_d \nabla \beta)}{\beta^2} = 0
$$

so:

$$
k \beta \nabla \gamma_d = \gamma_d \nabla \beta
$$

Taking the magnitude of the above yields:

$$
k \beta \|\nabla \gamma_d\| = \gamma_d \|\nabla \beta\|
$$

So, to avoid a critical point we require: $k > \frac{\gamma_d \|\nabla \beta\|}{\|\nabla \gamma_d\|}$

Since, $\nabla \beta = \sum_{i=0}^{N_{\text{obs}}} \nabla \beta_i \hat{\beta}_i$, $\|\nabla \gamma_d\| = 2\sqrt{\gamma_d}$ and $\hat{\beta}_i = \beta_i \geq \varepsilon$ $k$ must comply with the following constraint:

$$
k \geq \frac{1}{2\varepsilon} \max \{\sqrt{\gamma_d}\} \sum_{c} \max \{|\nabla \beta_i|\} \equiv N(\varepsilon) \quad(20)
$$

with $\max_{q} \{\gamma_d(q)\} = R_0 + \|q_d\|$. 

**Proposition 5.4.** There exist an $\varepsilon_0$ such that $\hat{\varphi}$ has no local minimum in the set $B_i(\varepsilon), i \in O$ (near the obstacles) for $\varepsilon \leq \varepsilon_0$.

**Proof.** Intuitively the NF must “flow” around the obstacles. We therefore, show that at least one eigenvalue of $\nabla^2 \hat{\varphi}$ is negative by calculating the projection onto the direction perpendicular to the gradient of $\beta_i$ at $q$.

Consider a critical point $q_c \in B_i(\varepsilon)$. The Hessian of $\hat{\varphi}$ is:

$$
\nabla^2 \hat{\varphi}(q) = \frac{1}{\beta^2} \left( \beta^2 \nabla^2 \gamma_d^k - \gamma_d^4 \nabla^2 \beta \right) = \frac{\gamma_d^{k-2}}{\beta^2} \left( k \beta \left( \gamma_d \nabla^2 \gamma_d + (k-1) \nabla \gamma_d \nabla \gamma_d^T \right) - \gamma_d^2 \nabla^2 \beta \right)
$$

Taking the tensor product of both sides of Eq.19 yields:

$$
(k \beta)^2 \nabla \gamma_d \nabla \gamma_d^T = \gamma_d^2 \nabla \beta \nabla \beta^T
$$

So, the Hessian of $\hat{\varphi}$ becomes:

$$
\nabla^2 \hat{\varphi}(q) = \frac{\gamma_d^{k-2}}{\beta^2} \left( k \beta \nabla^2 \gamma_d + \left( 1 - \frac{1}{k} \right) \frac{\gamma_d}{\beta} \nabla \beta \nabla \beta^T - \gamma_d^2 \nabla^2 \beta \right)
$$

(21)
Let us denote $A_s \equiv \frac{1}{2}(A + A^T)$ - the symmetric part of the matrix $A$, so we can write:

\[
\nabla^2 \hat{\varphi}(q) = \frac{\gamma_d}{\beta^2} \left( k_\beta \nabla^2 \gamma_d + \left( 1 - \frac{1}{k} \right) \right) \cdot \\
\cdot \frac{\gamma_d}{\beta} \left( \beta_1^2 \nabla \beta_1 \nabla \beta_1^T + 2 \beta_1 \beta_2 \left( \nabla \beta_1 \nabla \beta_2^T \right)_s + \beta_1^2 \nabla \beta_1 \nabla \beta_1^T \right) \\
- \gamma_d \left( \beta_1 \nabla^2 \beta_1 + 2 \left( \nabla \beta_1 \nabla \beta_1^T \right)_s + \beta_1 \nabla^2 \beta_1 \right)
\]

Note that $\nabla^2 \beta$, $\hat{\varphi}^T \nabla \beta = 0$, and $\nabla^2 \gamma_d = 2I$. Taking the quadratic form of $\hat{\varphi}$ by an arbitrary orthogonal vector to $\nabla \beta_i$:

\[
\hat{\varphi} \triangleq \frac{\nabla \beta_i (q_c)}{\| \nabla \beta_i (q_c) \|} T
\]

We can write:

\[
\hat{\varphi}^T \nabla^2 \hat{\varphi}(q) \hat{\varphi} = \\
= \beta_i \hat{\varphi}^T \left( 2k_\beta I + \left( 1 - \frac{1}{k} \right) \beta_1 \nabla \beta_1 \nabla \beta_1^T + \gamma_d \nabla^2 \beta_i \right) \hat{\varphi} + \\
\quad + \gamma_d \beta_1 \hat{\varphi}^T \nabla^2 \beta_i \hat{\varphi}
\]

(22)

It is hard to conclude whether the first component is positive or not. But note that the Hessian of $\beta_i$ (see Eq.10)

\[
\nabla^2 \beta_i = -\nabla^2 p_{pol} (q - q_i, R_i, \sigma_i)
\]

is negative definite since, $I_0(x) > I_1(x)$ $\forall x$ and $\| q - q_i \| > R_i \forall q \in B_1(\epsilon)$. Additionally, both $\gamma_d$ and $\beta_i$ are positive so, the second term is negative.

To ensure that Eq(22) is negative we can bound $\beta_i$ with $\epsilon$ by:

\[
\varepsilon < \varepsilon_0 \triangleq \\
\min_{q \in B_1(\epsilon)} \{ \gamma_d \beta_1 \hat{\varphi}^T \nabla^2 \beta_i \hat{\varphi} \}
\]

\[
\max_{q \in B_1(\epsilon)} \{ \hat{\varphi}^T \left( 2k_\beta I + \left( 1 - \frac{1}{k} \right) \beta_1 \nabla \beta_1 \nabla \beta_1^T + \gamma_d \nabla^2 \beta_i \right) \hat{\varphi} \}
\]

See Lemmas[13] and [2] in the Appendix for explicit expressions for the external terms.

**Proposition 5.5.** if $k \geq N(\epsilon)$ then there exists an $\varepsilon_1 > 0$ such that $\hat{\varphi}$ has no critical points near the workspace boundary, as long as $\varepsilon \leq \varepsilon_1$.

**Proof.** The inner product:

\[
\nabla \hat{\varphi}^T \nabla \gamma_d = \frac{\gamma_d}{\beta^2} (4k_\beta - \nabla \beta \nabla \gamma_d) > \\
\beta_0 \frac{\gamma_d}{\beta^2} (4k_\beta - \nabla \beta_0 \nabla \gamma_d)
\]

according to Eq.20 is

\[
\beta_0 \frac{\gamma_d}{\beta^2} (4k_\beta - \nabla \beta_0 \nabla \gamma_d) > 0.
\]

To estimate the second term define $\varepsilon_1$ as (loosely speaking) the probability for a robot located at $q_d$ to collide with the workspace boundary.

\[
\varepsilon_1 \equiv p_{\Delta o} - p_{tot}(q_d, R_0, \sigma_{rob})
\]

$\beta_0$ is restricted by:

\[
\beta_0 = p_{\Delta o} - p_{tot}(q_d, R_0, \sigma_{rob}) < \varepsilon_1
\]

i.e. all other possible locations of the robot near the workspace boundary are more likely to collide the latter. $\nabla \beta_0$ points away from the destination $q_d$ at any point $q$ in $B(\epsilon_0)$ since $\nabla \beta_0 = -\nabla p_{tot}(q, R_0, \sigma_{rob})$ (see Eq. [9]), and $\nabla \gamma_d = 2 \| q - q_d \| > 0$, so $\nabla \gamma_d \nabla \beta_0 < 0$.

This completes the proof.

We showed that near the obstacles there may be critical points of $\hat{\varphi}$. We also proved that such a point will have a negative gradient component directed tangentially to the obstacles. Yet, in order for $\hat{\varphi}$ to be a navigation function we need show that it is Morse function.

**Proposition 5.6.** $\varphi$ is a Morse function.

**Proof.** We would like to prove then, that the component of the gradient of $\hat{\varphi}$ in the radial direction to the obstacle is positive. This way $\nabla \hat{\varphi}$ will not have any degenerate direction as required.

Substituting [10] into Eq[21] and multiplying the sides of the equation by: $\hat{\varphi} \triangleq \nabla \beta_i \hat{\varphi}$ it becomes:

\[
\frac{\beta^2}{\gamma_d} \hat{\varphi}^T \nabla^2 \hat{\varphi} \hat{\varphi} = \\
\frac{-\gamma_d}{2k_\beta} \| \nabla \beta \|^2 + \left( 1 - \frac{1}{k} \right) \frac{\gamma_d}{\beta} (\nabla \beta \cdot \hat{\varphi})^2 - \gamma_d \hat{\varphi}^T \nabla^2 \beta \hat{\varphi}
\]
Algebraic manipulations leads to (compare with [Prop. 3.9,[13]]):
\[
\frac{\beta^2}{\gamma_i^k} - \tilde{v}^T \nabla^2 \tilde{v} \geq 0
\]
\[
\geq \frac{\gamma_i^d}{\beta_i} \left( \left( 1 - \frac{1}{k} \right) \beta_i \| \nabla \beta_i \|^2 - \tilde{v}^T (\beta_i^2 \nabla^2 \beta_i + \beta_i \tilde{v}^T \nabla^2 \beta_i) \tilde{v} \right)
\]
Since \( q \in B_i(\varepsilon) \), and assuming that \( k \geq 2 \) it can be rearranged as:
\[
\frac{\gamma_i^d}{\beta_i} \left( \left( 1 - \frac{1}{k} \right) \beta_i \| \nabla \beta_i \|^2 - \varepsilon \tilde{v}^T \nabla^2 \beta_i \tilde{v} \right)
\]
for the first term to be positive we require:
\[
\varepsilon \leq \min_{q \in B_i(\varepsilon)} \left\{ \| \nabla \beta_i \|^2 \right\} \equiv \varepsilon'
\]
and a sufficient condition for the second term to be positive we require:
\[
\varepsilon^2 \leq \frac{\tilde{v}^T \nabla^2 \beta_i \tilde{v}}{4 |\tilde{v}^T \nabla^2 \beta_i|} \leq \min \left\{ \sqrt{\frac{\beta_i}{\varepsilon}}, \| \nabla \beta_i \| \right\} \equiv \varepsilon''
\]
See Lemmas [1] and [2] in the Appendix for explicit expressions for the external terms. So, by restricting the distance of \( q \) to the obstacles such that \( \beta(q) < \min\{\varepsilon', \varepsilon''\} \) we guarantee that \( \varphi \) is a Morse function.

Finally, in order for \( \varphi \) to be a navigation function in all \( C \) we require that \( \varepsilon = \min\{\varepsilon_0, \varepsilon_1, \varepsilon', \varepsilon''\} \) which is required to determine \( k \) (constrained by \( k > N(\varepsilon) \)).

6 Some Examples

In this short section we shall exemplify how our PNF works. We implemented our PNF motion planning scheme using MATLAB on a Intel Core I5. We set the world’s radius to a unit length. Recall that to compute \( \beta_i \) (see Eq. [18]) we need \( p_{\Delta} \) which is given by
\[
p_{\Delta_i} = p_{\text{tot}} (R_{\Delta_i}, R_i, \sigma_i)
\]
In order to solve Eq[13] for \( R_{\Delta} \) we used an approximation. Following Bryson (cf. [25], §10.7) \( R_{\Delta} \) satisfies:
\[
\Delta = (2\pi)^{-n/2} \int_0^{R_{\Delta}} e^{-1/2r^2} f(r) \, dr
\]
where \( f(r) \, dr \) is the volume element in \( n \)-dimensional space. For the two dimensional case \( R_{\Delta} = -2ln(1 - \Delta) \). This can be thought of as the modified obstacle boundary. In other words, the obstacle is expended to include the probability \( \Delta \). Since \( (q - \mu_o)^T \Sigma^{-1} (q - \mu_o) = R_{\Delta}^2 \), the probability for a collision can be approximated as:
\[
p_{\Delta} \approx \frac{1 - \Delta}{\sqrt{2\pi} \Sigma} \] (24)

Figure 2 depicts a stochastic scenario where the obstacles radii from the top right c.w. are 4, 4 and 2, while the STD’s are 6, 5, and 4 respectively. the robot radius is 3 and its STD is 4. \( k \) is chosen as 5, and \( \Delta = 0.1 \). Figure 3 depicts a scenario where the obstacles have the same geometry, while the STD’s are 30, 5, and 4 respectively, \( k \) is chosen as 5 again, and \( \Delta \) remains 0.1.

In Figure 4 we used the same geometry and the same standard deviations. \( k \) is chosen as 0.5, and \( \Delta \) remains the same. We can see that the PNF seems more steep close to the obstacles.

Figure 5 depicts a poorly chosen constant \( k \) which resulted with the undesirable local minimum located.
at \([0,-17]\) and \([22, 1]\). In this case \(N(\varepsilon)\) is large since the obstacles are close to each other resulting with a small \(\varepsilon\) (see Prop. 5.4).

7 Summary

We have introduced a function \(\varphi\) defined on \(C \subset \mathbb{R}^n\). We showed that \(\varphi\) is a Navigation Function in the sense of [13]. The PNF is proved to converge for all stochastic scenarios. That is, given a disc robot, disc-shaped obstacles (located in a disc shaped world) with given uncertainties in their locations we show how to construct \(\varphi\). following the PNF’s gradient the robot will reach a predetermined destination while minimizing the probability for collision as well as bounding the maximal collision probability.

The discussion in this paper can be generalized to star worlds, as well, in exactly the same manner done in [9].

We used a Radial Gaussian to model the uncertainties. Future work will include a generalized Gaussian. Furthermore the authors intend to apply the above to the problem of stochastic-dynamic environment as well.

\textbf{proof for Prop 5.4.} At a critical point the first and second derivative of \(\varphi\) are:

\[
\nabla \varphi = \frac{1}{(\gamma d + \beta)^{2/k}} \left( (\gamma d + \beta)^{1/k} \nabla \gamma d - \gamma d \nabla (\gamma d + \beta)^{1/k} \right)
\]

\[
\nabla^2 \varphi = \frac{1}{(\gamma d + \beta)^{2/k}} \left( (\gamma d + \beta)^{1/k} \nabla^2 \gamma d - \gamma d \nabla^2 (\gamma d + \beta)^{1/k} \right)
\]

Since, \(\gamma d(q)\big|_{q_d} = \nabla \gamma d(q)\big|_{q_d} = 0\) we have:

\[
\nabla \varphi(q)\big|_{q_d} = 0.
\]

So:

\[
\nabla^2 \varphi(q)\big|_{q_d} = \beta^{-1/k} \nabla^2 \gamma d\big|_{q_d} = \beta^{-1/k} 2I \neq 0
\]

which proves the proposition. \(\square\)

Now, we shall prove some of the bounding \(\varepsilon\)'s we used in Section 5.

\textbf{Lemma 1.} \(\max_q \{ \|\nabla \beta_i\| \} \leq \frac{e^{-\frac{1}{2}}}{\sqrt{2\pi \sigma_i^2}}\) and,

\[
\max_q \{ \|\nabla^2 \beta_i\| \} \leq \sqrt{\frac{2}{\pi}} \frac{e^{-\frac{3}{2}}}{\sigma_i^3}.
\]

\textbf{Proof.} Throughout the paper \(\|\|\) denoted the Euclidean norm. Here we shall use \(\|\|_p\) to indicate the general p-norm (e.g. \(\|\|_2 = \|\|\)).
Recall that $\beta_i$ is based on the convolution of the disc with a Gaussian. Thus as a consequence of Young’s inequality [20], $\|\nabla \tilde{\beta}_i\|_2$ can be written as:

$$\|\nabla (D(r, R_i) \ast G(r, \sigma_i))\|_2 = \|D(r, R_i) \ast \nabla G(r, \sigma_i)\|_2$$

Again using Young’s inequality, this amounts to:

$$\|D(r, R_i) \ast \nabla G(r, \sigma_i)\|_2 \leq c_2,1 \|D(r, R_i)\|_2 \|\nabla G(r, \sigma_i)\|_1$$

where $c_2,1 < 1$. Since $D(r, R_i)$ is the disc with a unit height we have:

$$\max\{\|\nabla \beta_i\|\} \leq \max\{\|\nabla G(r, \sigma_i)\|_1\} = \frac{e^{-\frac{r^2}{2}}}{\sqrt{2\pi\sigma_i^2}}$$

Using the same logic:

$$\max\{\|\nabla^2 \beta_i\|\} \leq \max\{\|\nabla^2 G(r, \sigma_i)\|_1\} = \frac{\sqrt{2} e^{-\frac{r^2}{2}}}{\sigma_i^2}$$

Lemma .2.

$$\max_{q \in B_i(\varepsilon)} \{\tilde{\beta}_i\} = \prod_{j \in \{O-i\}} p_{\Delta_j} - p_{tot}(||q_j - q_i|| + R_\varepsilon, R_j, \sigma_j)$$

$$\min_{q \in B_i(\varepsilon)} \{\tilde{\beta}_i\} = \prod_{j \in \{O-i\}} p_{\Delta_j} - p_{tot}(||q_j - q_i|| - R_\varepsilon, R_j, \sigma_j)$$

Proof. Since $\tilde{\beta}_i = \prod_{j \in \{O-i\}} \beta_j$ we have

$$\max_{q \in B_i(\varepsilon)} \{\beta_j\} = p_{\Delta_j} - p_{tot}(||q_j - q_i|| + R_\varepsilon, R_j, \sigma_j)$$

where $R_\varepsilon$ is a scalar that satisfies $p_{tot}(||q_i + R_\varepsilon||, R_j, \sigma_j) = \varepsilon$. In the same way we obtain the second result. □

Lemma .3. $\max\{\|\nabla \tilde{\beta}_i\|\} \leq \frac{1}{\sqrt{2\pi\varepsilon}} \sum_{j \in \{O-i\}} \frac{1}{\sigma_j}$

Proof.

$$\nabla \tilde{\beta}_i = \sum_{j \in \{O-i\}} \nabla \beta_j \prod_{k \neq i,j} \beta_k.$$  

The result follows since $\max_{q} \{\beta_i\} = 1$ and by Lemma □

References

[1] S. Shoval, I. Ulrich, and J. Borenstein, “Navbelt and the guide-cane [obstacle-avoidance systems for the blind and visually impaired],” Robotics & Automation Magazine, IEEE, vol. 10, no. 1, pp. 9–20, 2003.

[2] S. Sundar and Z. Shiller, “Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation,” Robotics and Automation, IEEE Transactions on, vol. 13, no. 2, pp. 305–310, 1997.

[3] I. Filippidis and K. J. Kyriakopoulos, “Roadmaps using gradient extremal paths,” in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 370–375.

[4] A. Lazanas and J.-C. Latombe, “Landmark-based robot navigation,” Algorithmica, vol. 13, no. 5, pp. 472–501, 1995.

[5] R. Pepy and A. Lambert, “Safe path planning in an uncertain-configuration space using rrt,” in Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006, pp. 5376–5381.

[6] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid,” in Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007, pp. 1610–1616.

[7] J. Andrews and N. Hogan, “Impedance control as a framework for implementing obstacle avoidance in a manipulator.” Boston, Mass, USA, 1983, pp. 243 – 251, cONTROL SYSTEMS;PRODUCTION ENGINEERING;.

[8] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” Systems, Man and Cybernetics, IEEE Transactions on, vol. 19, no. 5, pp. 1179–1187, 1989.
[9] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[10] E. P. Lopes, E. P. Aude, J. T. Silveira, and H. Serdeira, “Obstacle avoidance strategy based on adaptive potential fields generated by an electronic stick,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 2626–2631.

[11] H. E. Espitia and J. I. Sofrony, “Path planning of mobile robots using potential fields and swarms of brownian particles,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 123–129.

[12] P. Konkimalla and S. M. LaValle, “Efficient computation of optimal navigation functions for nonholonomic planning,” in *Robot Motion and Control, 1999. RoMoCo’99. Proceedings of the First Workshop on*. IEEE, 1999, pp. 187–192.

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

[14] S. M. LaValle and P. Konkimalla, “Algorithms for computing numerical optimal feedback motion strategies,” *The International Journal of Robotics Research*, vol. 20, no. 9, pp. 729–752, 2001.

[15] R. Yozevitch, B. Ben Moshe, and H. Levy, “Breaking the 1 meter accuracy bound in commercial gnss devices,” in *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*. IEEE, 2012, pp. 1–5.

[16] O. Gal, Z. Shiller, and E. Rimon, “Efficient and safe on-line motion planning in dynamic environments,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 88–93.

[17] L. J. Guibas, L. Ramshaw, and J. Stolfi, “A kinetic framework for computational geometry,” in *FOCS*, vol. 24, 1983, pp. 100–111.

[18] S. Vinga, “Convolution integrals of normal distribution functions,” 2004.

[19] R. S. Strichartz, *A guide to distribution theory and Fourier transforms*. World Scientific, 2003.

[20] H. E. Plesser, “The Convolution of an Excentric Gaussian with a Disk.”

[21] L. Blumenson, “A derivation of n-dimensional spherical coordinates,” *American Mathematical Monthly*, pp. 63–66, 1960.

[22] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Courier Dover Publications, 1972, no. 55.

[23] W. Gautschi, “A note on the recursive calculation of incomplete gamma functions,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 25, no. 1, pp. 101–107, 1999.

[24] S. R. Lindemann and S. M. LaValle, “Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions,” *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 600–621, 2009.

[25] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[26] J. Fournier, “Sharpness in youngs inequality for convolution,” *Pacific Journal of Mathematics*, vol. 72, no. 2, pp. 383–397, 1977.