Motion Planning with Environment Uncertainty
Antony Thomas, Fulvio Mastrogiovanni and Marco Baglietto

Abstract—As robots are becoming more pervasive and are increasingly used in close proximity to humans and other objects in factories, living spaces, elderly care, and robotic surgery, planning for collision free trajectories in real-time is imperative for safe and efficient operation. Yet, in the presence of noisy sensors, both the robot and world state cannot be estimated precisely. In this paper we investigate the problem of motion planning under environment uncertainty. To this end, we first incorporate object uncertainties in Belief Space Planning (BSP) and derive the resulting Bayes filter in terms of the Extended Kalman Filter (EKF) update equations. We further formulate the collision constraint as a quadratic form in random variables, thus computing an exact expression for collision probability. We further validate our approach using numerical integration and provide a comparison to other approaches.

Index Terms—Collision Probability, Belief Space Planning

I. INTRODUCTION

Belief Space Planning (BSP) and decision making under uncertainty has been researched extensively in the past with applications spanning a variety of areas including autonomous navigation, multi-modal planning, and active Simultaneous Localization and Mapping (SLAM) [1], [6], [10], [11], [15], [17], [20], [22], [23]. Uncertainties often arise due to insufficient knowledge about the environment, inexact robot motion or imperfect sensing. Planning is therefore done in the belief space, which corresponds to the set of all probability distributions over possible robot states. Further, the nature of uncertain environments are such that they often preclude the existence of collision free trajectories [2]. As such, for safe navigation, both the robot state uncertainty and the uncertainty in obstacle estimates need to be considered while computing collision probabilities. [13], [16] truncate the Gaussian state distributions to compute bounded collision-free trajectories. [5] and [14] compute the collision probability by marginalizing the joint distribution between the robot pose and the obstacle. However, since there is no closed-form solution to this formulation, an approximation is assumed. In [12], an approximation is computed using Monte Carlo Integration (MCI), which is nonetheless computationally intensive. Another relevant method that uses Monte Carlo approach and is real-time compatible is Monte Carlo Motion Planning (MCMP) [8]. In [2], a Gaussian Process (GP) based approach is used to learn motion patterns (a mapping from states to trajectory derivatives) to identify possible future obstacles trajectories. [3] focus exclusively on obstacle uncertainty. Uncertain obstacles are modeled as polytopes with Gaussian-distributed faces in [19]. Risk contours map are considered in [7], [9] to obtain safe paths with bounded risks. Formal verification methods have also been used to construct safe plans [4], [18]. Yet, most approaches leverage’s Boole’s inequality to compute collision probability along a path by summing or multiplying the probabilities along different waypoints in the path. Such approaches tend to be overly conservative. Moreover, in most approaches, the collision probability computed along each waypoint is an approximation of the true value. On the one hand, such approximations can overly penalize paths and could gauge all plans to be infeasible. On the other hand some approximations can be lower than the true collision probability values and can lead to synthesizing unsafe plans.

Localization is also a key aspect for safe and efficient navigation. However, most approaches assume that landmarks are known with high certainty. For example, given the map of the environment, while planning for future actions the standard Markov localization does not take into account the map uncertainty (that is, landmark locations are assumed to be perfect). However, this might not be true in practice. For example, landmarks locations obtained from a SLAM session may be uncertain. This landmark uncertainty directly translates to the fact that the viewpoints from which the landmark can be observed are uncertain. Therefore, one can only reason in terms of the probability of observing the object from the considered pose or the viewpoint. This results in a probability distribution function for the viewpoints, whose mean corresponds to observing the object with highest probability. Consequently, not considering this uncertainty can wrongly localize the robot, leading to inefficient plans causing catastrophes. We will use the term object uncertainty to refer to the notion of uncertainty in landmark location.

II. OBJECT UNCERTAINTY

We define the object space \( \mathcal{O} = \{ O^i | O^i \) is an object, and \( 1 \leq i \leq |\mathcal{O}| \} \) to be the set of all objects in the environment. Given an initial distribution \( p(x_0) \), and the motion and observation models \( p(x_k|x_{k-1}, u_k) \) and \( p(z_k|x_k, O_k^i) \), the posterior probability distribution at time \( k \) is the belief \( b[x_k] \) and can be written as \( p(x_k|z_k, O_k^i, z_{0:k-1}, u_{0:k-1}) \), where \( O_k^i \) is the object observed at time \( k \), \( z_{0:k-1} = \{ z_0, ..., z_{k-1} \} \) is the sequence of measurements up to \( k-1 \) and \( u_{0:k-1} = \{ u_0, ..., u_{k-1} \} \) is the sequence of control up to \( k-1 \). Using Bayes rule and theorem of total probability, \( b[x_k] \) can be expanded as
\[
p(x_k|z_k, O_k^i, z_{0:k-1}, u_{0:k-1}) = \eta p(z_k|x_k, O_k^i) p(O_k^i|x_k) \int_{x_{k-1}} p(x_k|x_{k-1}, u_{k-1}) b[x_{k-1}] \quad (1)
\]
where \( \eta = 1/p(z_k|z_{0:k-1}, u_{0:k-1}) \) is the normalization constant. The term \( p(O_k^i|x_k) \) denotes the probability of observing...
the object $O_k^i$ from the pose $x_k$ and models the object uncertainty. This term can be modeled given the environment and since we consider a Gaussian parametrization of the belief, in this work we model the object distribution as a Gaussian $p(O_k^i|x_k)\sim\mathcal{N}(\mu_{O_k^i},\Sigma_{O_k^i})$, where $\mu_{O_k^i}$ is the mean viewpoint/pose that corresponds to the maximum probability of observing $O_k^i$ and $\Sigma_{O_k^i}$ is the associated covariance. The EKF is used for state estimating and the posterior mean and covariance can be computed as (notations and complete derivation can be found in [21])

$$\mu_k = \tilde{\mu}_k + K_k + 1 (z_k - h(\tilde{\mu}_k)) + \Sigma_k^{-1}(\mu_{O_k^i} - \tilde{\mu}_k)$$ (2)

$$\Sigma_k = (I - K_k H_k) \Sigma_k (I + K_k H_k)^{-1} (I - K_k H_k)$$ (3)

We note that when no object uncertainty is considered the approach can be used for fast online planning. Let us denote by $w = x_k - s_k$ the difference between the two random variables. Then we know that $w$ is also a Gaussian, distributed as $w \sim \mathcal{N}(\mu_k - \mu_{s_k},\Sigma_k + \Sigma_{s_k})$. The collision constraint in (5) can now be written as

$$y = ||w||^2 = w^T w \leq (r_1 + s_1)^2$$ (6)

where $y$ is a random vector distributed according to the squared $L_2$-norm of $w$. Now, given the probability density function (pdf) of $y$, the collision constraint reduces to solving the integral

$$P(C_{x_k,s_k}) = \int_{0}^{(r_1 + s_1)^2} p(y)$$ (7)

where $p(y) = F_2'(y) = y$ is the pdf of $y$. It is noteworthy that the above integral is the cumulative distribution function (cdf) of $y$, that is, $P(C_{x_k,s_k}) = F_2(y)$, where $F_2(y)$ denotes the cdf. Moreover, the left hand side of (6), is a quadratic form in the random variables of $w$ and the cdf is computed as

$$F_2(y) = p(y \leq y) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{y^{2+k}}{\Gamma(\frac{2}{2} + k + 1)}$$ (8)

The above infinite series converges within the first few terms and we refer the readers to [21] for the definition of a quadratic form in random variables, its cdf and the proof of convergence. Further, it can be shown that to achieve a truncation error $E \leq \delta$, for an $\epsilon$-safe configuration$^1$, $k = O \left( \log \frac{\delta \rho}{\rho(1-\epsilon)} \right)$ iterations are required$^2$. We note that for each obstacle, the complexity is increased by this factor. As shown in the next section, the computation time is of the order of few milliseconds and hence the approach can be used for fast online planning.

IV. COMPARISON TO OTHER APPROACHES

In this section we compare our approach for collision probability computations with that of Park et al. [14] and [5]. Numerical integration of (4) gives the exact value and we use the same to validate our approach. Three different cases are considered (see Fig. 1). The solid green circle denotes an obstacle of radius 0.5m and its corresponding uncertainty contours are shown as green circles. The solid blue circle

\footnotesize

\[ 1 \text{ A robot configuration } x_k \text{ is an } \epsilon \text{-safe configuration with respect to an obstacle } s_k \text{ if the probability of collision is such that, } p(C_{x_k,s_k}) \leq 1 - \epsilon. \]

\[ 2 y = (r_1 + s_1)^2, \rho \text{ depends on robot and obstacle state covariance.} \]
denotes a robot of radius 0.3m with the blue circles showing the Gaussian contours. We define a collision probability threshold of 0.1 and the collision probability values and the computation times are provided in Table I. In Fig. 1(a), the robot position known with high certainty and our approach computes collision probability as 4.61% and hence the given configuration is feasible. The numerical integral provides the actual value and as seen in Table I, it is computed to be 4.62%, thus proving the exactness of our method. However, the collision probability computed as given in [14] is 33.26% (almost seven times our value), predicting the configuration to be unfeasible. The approach in [5] also gave a feasible value of 5.84%. In Fig. 1(b), there is robot uncertainty along the horizontal axis and the collision probability computed using our approach is 8.22%. The actual value is computed to be 8.25%. The values computed using the approaches in [14], [5] are 36.31% (4.5 times our value) and 14.20%, respectively. The approach of Park et al. [14] and [5] assumes that the robot radius is very small. We also compute the collision probabilities for a robot and an obstacle with radius 0.05m each, where the robot and the obstacle are touching each other (Fig. 1(c)). The obstacle location is also much more certain, with the uncertainty reduced by 97% as compared to cases in Fig. 1(a),(b). Actual value obtained using numerical integral is 14.82%. The probability of collision computed using our approach is 14.83%, rightly predicting it to be unfeasible. The approach in [14] computed the value as 0.61% and using [5] a value of 0.46% is obtained. Thus, using the approaches in [5], [14] would lead to collision as it predicts the configuration to be feasible. Our approach computes the exact probability of collision and outperforms the approaches in [5], [14].

V. CONCLUSION

We present a BSP framework that incorporate object uncertainties while localizing the robot and derive the EKF update equations for the same. Unlike previous approaches that compute approximate upper bounds we derive an exact expression for computing the collision probability. We validate our approach using numerical integration and provide comparison to other approaches.

REFERENCES


