

Estimating Probability of Collision for Safe Planning under Gaussian Motion and Sensing Uncertainty

Sachin Patil

Jur van den Berg

Ron Alterovitz

Abstract—In this paper we present an analytical method to estimate the probability of collision of a plan for a mobile robot operating under the assumptions of Gaussian motion and sensing uncertainty. Estimating the probability of collision is an integral step in several motion planning under uncertainty algorithms for characterizing the safety of motion plans. In contrast to prior art, our work specifically accounts for the fact that the probabilities of collision at each stage along the plan are conditioned on the previous stages being collision free: we propose a novel method for truncating the a priori state distributions with respect to obstacles, correctly propagating them forward in time, and accurately estimating the collision probability of the plan. Our method can be directly applied to a variety of existing motion planners to improve their performance and quality of computed plans. We empirically demonstrate that our method is orders of magnitude faster than naïve sampling based methods. We also show that the mean error in estimating the probability of collision using our method is considerably lesser as compared to prior methods, while incurring negligible computational overhead.

I. INTRODUCTION

From self-driving cars [21] to steerable medical needles operating in the human body [4], many robots need to maneuver through highly unstructured environments to accomplish their objective. Motions computed for such robots must both consider the kinematic and dynamic constraints of the robot and the safety of the robot, objects, and humans in the environment. It is highly undesirable for an autonomous car to drive into a pole, or for a steerable needle to penetrate sensitive tissue such as nerves during a medical procedure.

Maintaining the safety of robot motions is complicated by real-world uncertainties; the motion of the robot may deviate unpredictably from the assumed dynamics model, and sensors might yield imperfect information about the robot state due to noisy and partial state measurements. For many applications, the motion plan chosen for execution should be as safe as possible so that the robot does not collide with obstacles in the environment. Prior approaches consider the uncertainty in the robot state to quantify plan safety in terms of the probability that the robot would collide with obstacles during execution [24], [27], [3]. However, they typically compute overly conservative estimates of the probability of collision, which may result in the robot not attempting a motion that is in fact very safe.

In this work, we present a fast, analytical method that estimates the probability of collision for a robot executing a

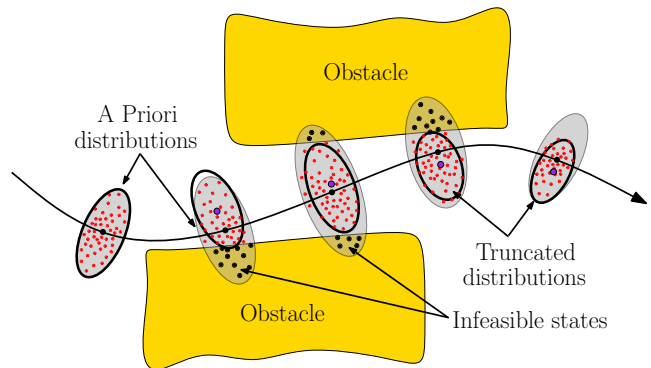


Fig. 1. We estimate the probability of collision based on a priori probability distributions of the robot state. The probabilities of collision at each stage of the plan are conditioned on the previous stages being collision free. We propose a novel method to truncate the a priori distributions with respect to obstacles to discount plan executions that collide with obstacles (black disks). Propagating the truncated distributions (black ellipses) accounts for only the collision free samples (red disks), resulting in accurate estimation of the probability of collision. Approximating the collision probability based on the unconditional distributions (gray ellipses), as is done in most prior work, results in a very conservative estimate of the collision probability.

motion plan under the assumptions of Gaussian motion and sensing uncertainty. Formally speaking, let $\mathbf{x}_t \in \mathcal{X}$ denote the state of the robot at stage t along the plan, and $\mathcal{X}_F \subset \mathcal{X}$ denote the feasible space not occupied by obstacles. The probability that a plan consisting of ℓ stages is collision free is then given by:

$$p\left(\bigwedge_{t=0}^{\ell} \mathbf{x}_t \in \mathcal{X}_F\right) = \prod_{t=0}^{\ell} p(\mathbf{x}_t \in \mathcal{X}_F \mid \bigwedge_{i=0}^{t-1} \mathbf{x}_i \in \mathcal{X}_F).$$

Hence, to accurately estimate the probability of collision, we need to compute the distribution of the state at each stage along the plan *conditioned* on the previous stages being collision free. This amounts to propagating the a priori distributions forward in time in such a way that instances that collide with obstacles are discounted from the propagation (Fig. 1). For this we propose a novel method to truncate the a priori distributions with respect to obstacles, approximate the truncated distributions by Gaussians, and propagate them forward in time. This results in an accurate estimate of the conditional distributions, and consequently, enables accurate estimation of the collision probability.

Existing approaches, in contrast, typically “approximate” the collision probability of a plan by assuming the probabilities of collision at stages along the plan are independent, i.e., $p(\bigwedge_{t=0}^{\ell} \mathbf{x}_t \in \mathcal{X}_F) \approx \prod_{t=0}^{\ell} p(\mathbf{x}_t \in \mathcal{X}_F)$ [24], [27], [3]. This yields an overly conservative estimate of the probability of collision (see Fig. 1), which might result in overly conserva-

Sachin Patil and Ron Alterovitz are with Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. {sachin, ron}@cs.unc.edu

Jur van den Berg is with the School of Computing, University of Utah, Salt Lake City, UT, USA. berg@cs.utah.edu

tive motion plans and, depending on the safety required by the motion planner, may result in failure to find a feasible plan even if one exists.

Our method can either be directly used to quantify the safety of a plan [24], [14], to accurately estimate collision chance constraints [3], [27], or to elegantly account for hard state constraints imposed by obstacles in optimization based [15], [7], [25] or inference based [23] planning methods. Our truncation approach is also directly applicable to the important problem of optimal state estimation with hard state constraints [19].

We present simulation results for a car-like robot with second-order dynamics, and a nonholonomic bevel-tip flexible needle, navigating with stochastic dynamics while operating under the guidance of partial and noisy state measurements. Our results suggest that our method is orders of magnitude faster than naïve sampling based methods, and the mean error in estimating the probability of collision is considerably lesser as compared to prior methods.

II. PREVIOUS WORK

Explicitly considering uncertainty in robot motion and sensing during motion planning enables the computation of safer plans. Motion planners that consider uncertainty have been developed recently for a variety of domains and settings [18], [1], [11], [9], [16], [15], [25], [14], [3], [27].

A key step of many motion planning methods that consider uncertainty is to estimate the probability of collision of a motion plan. These planners generally characterize the uncertainty of the robot's position by estimating the a priori probability distributions of the robot state along a given plan. A common approach is to check for collisions between these distributions and obstacles to compute an upper bound for the collision probability, which is used as a metric to evaluate the safety of the plan [24], [14]. This metric has also been used to penalize collisions with obstacles for computing optimal motion plans in an optimization based framework [25]. Vitus and Tomlin [27] use these distributions to compute a conservative probability bound using Boole's inequality for optimal motion planning using chance constrained optimization [2]. Bry and Roy [3] use an identical conservative estimate to compute edge costs for optimal motion planning in belief spaces. However, these methods incorrectly assume that the probabilities of collision are independent, which results in overly conservative plans or, in some cases, might result in failure to find a feasible plan even if one exists.

A naïve Monte-Carlo sampling strategy could also be used to accurately estimate the probability of collision. Lambert et al. [12] and Du Toit and Burdick [6] propose methods to accelerate the performance of such methods. This approach is general enough to consider all sources of uncertainty, including motion and sensing uncertainty, and uncertainty in sensing the obstacles in the environment. However, this estimation process can be computationally expensive if performed repeatedly, like in [25], [27], [3]. In contrast, we present a fast, analytical method for accurately estimating the conditional state distributions and the probability of collision

of a plan with Gaussian models of motion and sensing uncertainty.

The use of truncated Gaussian distributions [10] has been previously explored by Simon [19] in the context of optimal state estimation with state constraints, but this work does not consider motion uncertainty. Greytak [8] provides an analytical method to compute the probability of collision using truncated Gaussians but does not consider sensing uncertainty and only truncates the state distributions with respect to the obstacle closest to the robot state being considered. Toussaint [23] uses truncated Gaussians in an expectation-propagation framework for Bayesian inference under hard constraints and limits, but the truncation result is dependent on the order in which obstacles are processed, which leads to problems with convergence of the algorithm [22]. In contrast, we propose a novel order-independent algorithm for truncating Gaussian distributions with respect to hard state constraints.

III. PROBLEM STATEMENT

We consider a robot operating in an environment that may contain obstacles. The stochastic dynamics of the robot follow a given discrete-time model:

$$\mathbf{x}_t = \mathbf{f}[\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t], \quad \mathbf{m}_t \sim \mathcal{N}[\mathbf{0}, M_t] \quad (1)$$

where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the state of the robot at stage t , $\mathbf{u}_t \in \mathbb{R}^{n_u}$ is the applied control input, and \mathbf{m}_t is zero-mean Gaussian noise with variance M_t that models the motion uncertainty. During execution of a plan, partial and noisy sensor measurements of the robot state are available according to a given stochastic model:

$$\mathbf{z}_t = \mathbf{h}[\mathbf{x}_t, \mathbf{n}_t], \quad \mathbf{n}_t \sim \mathcal{N}[\mathbf{0}, N_t] \quad (2)$$

where $\mathbf{z}_t \in \mathbb{R}^{n_z}$ is the measurement obtained at time t and \mathbf{n}_t is the zero-mean Gaussian noise with variance N_t that models the sensing uncertainty.

Since our method serves as an evaluation metric, we assume the existence of a nominal plan computed by an RRT or other method as discussed in Sec. II. The nominal plan is defined by $[\mathbf{x}_0^*, \mathbf{u}_0^*, \dots, \mathbf{x}_\ell^*, \mathbf{u}_\ell^*]$ where $\mathbf{x}_t^* = \mathbf{f}[\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{0}]$ for $0 < t \leq \ell$, where ℓ is the number of discrete stages in the plan. The nominal plan is consistent with the dynamics model if there were no motion uncertainty. During actual execution of the plan, the robot will likely deviate from the nominal plan due to motion uncertainty and inaccurate estimation of the robot state due to sensing uncertainty.

To compensate for uncertainty, we assume the robot executes the plan in a closed-loop fashion using a feedback controller and state estimator framework [20]. We assume the existence of a linear feedback control law that operates on the estimate of the robot state and aims to keep the robot close to the nominal plan. Several feedback controllers, such as the linear quadratic regulator (LQR) and the proportional integral derivative (PID) controller fall in this category and are compatible with our method [20]. We also assume that a Kalman filter is used for state estimation during execution, which is the minimum mean square error estimator for linear,

Gaussian systems. Several variants of the Kalman filter such as the extended filter (EKF), unscented filter (UKF), and the ensemble filter (EnKF) [19] are compatible with our method.

Given Gaussian models of motion and sensing uncertainty, a description of the obstacles in the environment, a nominal motion plan, and associated feedback controller and state estimator, the objective is to compute the probability of collision of a given plan. It is important to note that our method can be directly used for estimating the a priori distributions of the robot's state over time with hard state constraints, which is a key step in several optimization based methods for motion planning under uncertainty.

IV. ESTIMATING PROBABILITY OF COLLISION

In principle, our approach applies to linear dynamics and measurement models \mathbf{f} and \mathbf{h} . However, since the robot will be controlled stay close to the nominal plan during execution, we can approximate the nonlinear dynamics and measurement models with local linearizations around the plan. The linear(ized) models can be expressed in terms of deviation from the true state $\bar{\mathbf{x}}_t = (\mathbf{x}_t - \mathbf{x}_t^*)$, control input deviation $\bar{\mathbf{u}}_t = (\mathbf{u}_t - \mathbf{u}_t^*)$, and deviation from the actual measurement $\bar{\mathbf{z}}_t = (\mathbf{z}_t - \mathbf{h}[\mathbf{x}_t^*, \mathbf{0}])$, as:

$$\bar{\mathbf{x}}_t = A_t \bar{\mathbf{x}}_{t-1} + B_t \bar{\mathbf{u}}_{t-1} + V_t \mathbf{m}_t, \quad \mathbf{m}_t \sim \mathcal{N}[\mathbf{0}, M_t] \quad (3)$$

$$\bar{\mathbf{z}}_t = H_t \bar{\mathbf{x}}_t + W_t \mathbf{n}_t, \quad \mathbf{n}_t \sim \mathcal{N}[\mathbf{0}, N_t]. \quad (4)$$

where the Jacobians matrices of \mathbf{f} and \mathbf{h} are given by:

$$A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{0}], \quad B_t = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}[\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{0}], \quad (5)$$

$$V_t = \frac{\partial \mathbf{f}}{\partial \mathbf{m}}[\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{0}], \quad H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}[\mathbf{x}_t^*, \mathbf{0}], \quad W_t = \frac{\partial \mathbf{h}}{\partial \mathbf{n}}[\mathbf{x}_t^*, \mathbf{0}].$$

The true state \mathbf{x}_t , and hence the true state deviation $\bar{\mathbf{x}}_t$, is not available during actual execution. We use a Kalman filter to keep track of an estimate of the state deviation $\hat{\mathbf{x}}_t = \mathbb{E}[\bar{\mathbf{x}}_t]$. The estimate of the state deviation evolves according to:

$$\hat{\mathbf{x}}_t = K_t \bar{\mathbf{z}}_t + (I - K_t H_t)(A_t \hat{\mathbf{x}}_{t-1} + B_t \bar{\mathbf{u}}_{t-1}), \quad (6)$$

where K_t is the Kalman gain matrix [19]. To compensate for the uncertainty, the robot is controlled using a linear feedback policy related to the estimate of the state deviation as:

$$\bar{\mathbf{u}}_t = L_{t+1} \hat{\mathbf{x}}_t, \quad (7)$$

where L_t is the control gain matrix determined by the choice of feedback controller [20].

A. A Priori State Distributions

Under the given assumptions, the probability distributions of the robot state can be characterized a priori i.e., before execution. Combining Eqns. (3), (4), (6), and (7), the true state deviation $\bar{\mathbf{x}}_t$, and the estimate $\hat{\mathbf{x}}_t$, jointly evolve as [24]:

$$\begin{bmatrix} \bar{\mathbf{x}}_t \\ \hat{\mathbf{x}}_t \end{bmatrix} = \begin{bmatrix} A_t & B_t L_t \\ K_t H_t A_t & A_t + B_t L_t - K_t H_t A_t \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_{t-1} \\ \hat{\mathbf{x}}_{t-1} \end{bmatrix} + \begin{bmatrix} V_t & 0 \\ K_t H_t V_t & K_t W_t \end{bmatrix} \begin{bmatrix} \mathbf{m}_t \\ \mathbf{n}_t \end{bmatrix}, \quad \begin{bmatrix} \mathbf{m}_t \\ \mathbf{n}_t \end{bmatrix} \sim \mathcal{N}[\mathbf{0}, \begin{bmatrix} M_t & 0 \\ 0 & N_t \end{bmatrix}]. \quad (8)$$

We can write this equation in shorthand (for appropriate definitions of \mathbf{y}_t , \mathbf{q}_t , F_t , G_t , and Q_t) as:

$$\mathbf{y}_t = F_t \mathbf{y}_{t-1} + G_t \mathbf{q}_t, \quad \mathbf{q}_t \sim \mathcal{N}[\mathbf{0}, Q_t]. \quad (9)$$

The mean $\hat{\mathbf{y}}_t \in \mathbb{R}^{2n_x}$ and associated variance $R_t = \text{Var}[\mathbf{y}_t]$, propagate according to:

$$\hat{\mathbf{y}}_t = F_t \hat{\mathbf{y}}_{t-1}, \quad \hat{\mathbf{y}}_0 = \mathbf{0}, \quad (10)$$

$$R_t = F_t R_{t-1} F_t^T + G_t Q_t G_t^T, \quad R_0 = \begin{bmatrix} \text{Var}[\bar{\mathbf{x}}_0] & 0 \\ 0 & 0 \end{bmatrix}. \quad (11)$$

The *unconditional* a priori distribution of the state \mathbf{x}_t at stage t is given by the marginal $\mathbf{x}_t \sim \mathcal{N}[(\mathbf{x}_t^* + \Lambda \hat{\mathbf{y}}_t), \Lambda R_t \Lambda^T]$, where $\Lambda = [I \ 0]$.

To accurately estimate the probability of collision, we need to estimate the a priori state distributions at each stage along the plan that are *conditioned* on the previous stages being collision free, i.e. the distribution $(\mathbf{x}_t \mid \bigwedge_{i=0}^{t-1} \mathbf{x}_i \in \mathcal{X}_F)$. To this end, we pursue a recursive approach similar as above to propagate the conditional distributions.

Let $\mathbf{y}_{t|s}$ denote the joint distribution of the true state deviation and its estimate at time t conditioned on the state being collision free for all stages $0, \dots, s$:

$$\mathbf{y}_{t|s} = \left(\begin{bmatrix} \bar{\mathbf{x}}_t \\ \hat{\mathbf{x}}_t \end{bmatrix} \right) \bigg| \bigwedge_{i=0}^s \mathbf{x}_i \in \mathcal{X}_F. \quad (12)$$

We then repeatedly, for each stage t of the plan, carry out the following steps. Assume we are given the joint conditional distribution $\mathbf{y}_{t|t-1}$ as approximated by a Gaussian distribution $\mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$. We then approximate the distribution $\mathbf{y}_{t|t} \sim \mathcal{N}[\hat{\mathbf{y}}_{t|t}, R_{t|t}]$ of all collision-free states at stage t by truncating the distribution $\mathbf{y}_{t|t-1}$ against the obstacles in the environment. Truncating the distribution effectively discounts all colliding states from the distribution (Fig. 1), and results in a shift of the mean and variance by $\Delta \mathbf{y}_t$ and ΔR_{t-1} (as detailed in Sec. IV-B), respectively:

$$\hat{\mathbf{y}}_{t|t} = \hat{\mathbf{y}}_{t|t-1} - \Delta \mathbf{y}_t \quad (13)$$

$$R_{t|t} = R_{t|t-1} - \Delta R_t \quad (14)$$

Using Eqns. (10) and (11), the conditional mean and variance are then propagated according to:

$$\hat{\mathbf{y}}_{t+1|t} = F_{t+1} \hat{\mathbf{y}}_{t|t}, \quad (15)$$

$$R_{t+1|t} = F_{t+1} R_{t|t} F_{t+1}^T + G_{t+1} Q_{t+1} G_{t+1}^T. \quad (16)$$

The recursion then continues. The initial conditions are set by defining $\hat{\mathbf{y}}_{0|-1} = \hat{\mathbf{y}}_0 = \mathbf{0}$ and $R_{0|-1} = R_0 = \begin{bmatrix} \text{Var}[\bar{\mathbf{x}}_0] & 0 \\ 0 & 0 \end{bmatrix}$.

At each stage of the recursion, the marginal $\mathbf{x}_{t|t-1} \sim \mathcal{N}[(\mathbf{x}_t^* + \Lambda \hat{\mathbf{y}}_{t|t-1}), \Lambda R_{t|t-1} \Lambda^T]$ of the joint distribution $\mathbf{y}_{t|t-1}$ gives the a priori distribution of the robot state \mathbf{x}_t , given that all the previous states $[\mathbf{x}_0, \dots, \mathbf{x}_{t-1}]$ are collision free. When this distribution is truncated against the obstacles, an accurate estimate is obtained of the conditional probability of collision at stage t (as detailed in Sec. IV-C).

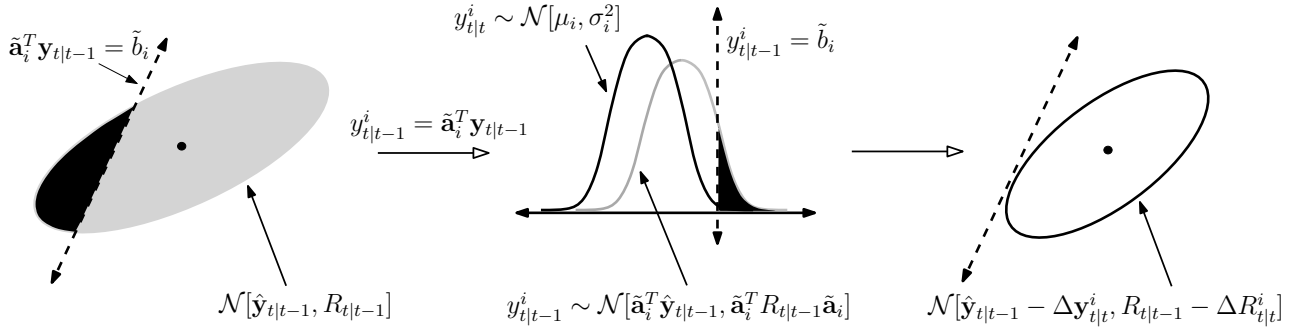


Fig. 2. The joint conditional distribution $\mathbf{y}_{t|t-1} \sim \mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$ (left), is truncated with respect to the i^{th} constraint $\tilde{\mathbf{a}}_i^T \mathbf{y}_{t|t-1} \leq \tilde{b}_i$, in \mathbb{R}^{2n_x} . Applying an affine transformation, $y_{t|t-1}^i = \tilde{\mathbf{a}}_i^T \mathbf{y}_{t|t-1}$, transforms the distribution to a 1D Gaussian $y_{t|t-1}^i \sim \mathcal{N}[\tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1}, \tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i]$ (middle). The area under the 1D Gaussian that lies beyond the constraint $y_{t|t-1}^i = \tilde{b}_i$ (shaded in black), gives the probability of collision of the robot with the i^{th} constraint. We estimate the truncated distribution in \mathbb{R}^{2n_x} by conditioning on the truncated 1D Gaussian $y_{t|t-1}^i \sim \mathcal{N}[\mu_i, \sigma_i^2]$ (right). The mean $(\hat{\mathbf{y}}_{t|t-1} - \Delta \mathbf{y}_{t|t-1}^i)$, and variance $(R_{t|t-1} - \Delta R_{t|t-1}^i)$, of the distribution $\mathbf{y}_{t|t}$ after truncation are obtained by accumulating the effects of truncation with respect to all constraints (order independent).

B. Truncating A Priori Distributions

At each stage t of the plan, we approximate the distribution of the feasible robot states with a truncated Gaussian distribution [10]. For the sake of brevity, we assume that the feasible region containing the state at each stage t is convex and is described by the conjunction of k linear inequality constraints as $\bigcap_{i=0}^k \mathbf{a}_i^T \mathbf{x}_t \leq b_i$. We later extend this analysis in Sec. V to non-convex regions by constructing a locally convex feasible region around the robot state.

Since the true state deviation and its estimate are correlated (Eqn. 8), it is important to truncate the joint conditional distribution $\mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$ in \mathbb{R}^{2n_x} , with respect to the k constraints. The i^{th} linear constraint is then represented in \mathbb{R}^{2n_x} as $\tilde{\mathbf{a}}_i^T \mathbf{y}_{t|t-1} \leq \tilde{b}_i$, where $\tilde{\mathbf{a}}_i = [\mathbf{a}_i^T; \mathbf{0}]$, and $\tilde{b}_i = (b_i - \mathbf{a}_i^T \mathbf{x}_t^*)$.

We truncate the joint conditional distribution with respect to each constraint in a sequential manner and then accumulate the effect of truncation over all the constraints. In contrast to prior methods that use truncated distributions [19], [22], we propose a novel truncation method that does not depend on the order in which the constraints are processed. Given the i^{th} constraint $\tilde{\mathbf{a}}_i^T \mathbf{y}_{t|t-1} \leq \tilde{b}_i$, we apply an affine transformation $y_{t|t-1}^i = \tilde{\mathbf{a}}_i^T \mathbf{y}_{t|t-1}$ to transform the conditional distribution $\mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$, to a 1D Gaussian $\mathcal{N}[\tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1}, \tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i]$ along an axis normal to the constraint (as shown in Fig. 2). The problem now simplifies to truncating the 1D Gaussian distribution at a specified upper bound given by $y_{t|t-1}^i = \tilde{b}_i$, which is well-known from standard statistical literature [10]. The mean, μ_i , and variance, σ_i^2 , of the truncated 1D Gaussian $y_{t|t-1}^i$ is given by:

$$\mu_i = \tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1} + \lambda(\alpha_i) \sqrt{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i}, \quad (17)$$

$$\sigma_i^2 = \tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i (1 - \lambda(\alpha_i)^2 + \alpha_i \lambda(\alpha_i)), \quad (18)$$

where

$$\alpha_i = \frac{(\tilde{b}_i - \tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1})}{\sqrt{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i}}, \quad \lambda(\alpha_i) = \frac{\text{pdf}(\alpha_i)}{\text{cdf}(\alpha_i)}. \quad (19)$$

$\lambda(\alpha_i)$ is the ratio of the standard Gaussian (mean 0 and variance 1) probability distribution function and the standard Gaussian cumulative distribution function evaluated at α_i .

Note that $(1 - \text{cdf}(\alpha_i))$ is the area under the Gaussian that lies beyond the constraint (shaded in black in Fig. 2), and is the probability that the robot lies in the infeasible region corresponding to the i^{th} constraint.

The mean and variance of the truncated distribution $\mathbf{y}_{t|t}$ can be found by conditioning the joint distribution $(\mathbf{y}_{t|t-1}, y_{t|t-1}^i)$, on the truncated 1D distribution $y_{t|t-1}^i: \mathbf{y}_{t|t} = (\mathbf{y}_{t|t-1} | y_{t|t-1}^i = y_{t|t-1}^i)$ (see appendix). The shift in the mean due to truncation with respect to the i^{th} constraint is given by:

$$\Delta \mathbf{y}_t^i = \frac{R_{t|t-1} \tilde{\mathbf{a}}_i}{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i} (\tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1} - \mu_i), \quad (20)$$

and the shift in variance is given by:

$$\Delta R_t^i = \frac{R_{t|t-1} \tilde{\mathbf{a}}_i}{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i} (\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i - \sigma_i^2) \frac{\tilde{\mathbf{a}}_i^T R_{t|t-1}}{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i}. \quad (21)$$

Given k constraints, the cumulative shift in the mean due to truncation is then given by $\Delta \mathbf{y}_{t|t} = \sum_{i=0}^k \Delta \mathbf{y}_{t|t}^i$, and the cumulative shift in variance is given by $\Delta R_{t|t} = \sum_{i=0}^k \Delta R_{t|t}^i$. The mean and variance of the truncated conditional distributions are then propagated recursively using Eqs. (15) and (16).

The truncation method described here is general enough to handle constraints on the entire robot state. Examples of such constraints include spatial constraints imposed on the position of the robot by obstacles in the environment or limits imposed on other state variables such as the velocity and acceleration of the robot. Our method had the added advantage that it is independent of the order in which constraints are processed, which eliminates convergence issues related to handling state constraints in several optimization [15], [7], [25] and inference based [23] planning methods.

C. Estimating the Probability of Collision

We use the truncated conditional distributions to estimate the overall probability of collision of the given plan, based on the conditional probabilities of collisions at each stage along the plan. Given the joint conditional distribution at stage t , $\mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$, and the set of k linear constraints that define the locally convex region of free space containing the robot,

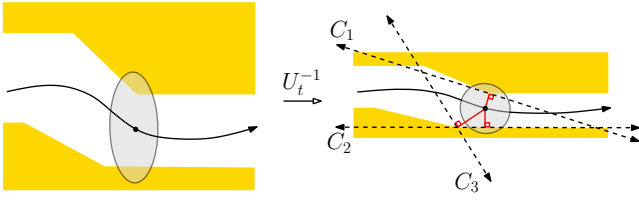


Fig. 3. We transform the environment such that the distribution of the robot position (left) is converted to a unit sphere (right). We then sequentially process the obstacle geometry in increasing order of distance from the origin. The linear constraints that define a locally convex region of the free space are determined by the normal to the vector of closest approach (shown in red). The locally convex region constructed using our approach for this example is defined by three constraints determined in order of their indices: C_1 , C_2 , and C_3 .

we compute a lower bound for the probability of the robot being collision free using Boole's inequality, as [27]:

$$\begin{aligned} p(\mathbf{x}_{t|t-1} \in \mathcal{X}_F) &\geq 1 - p\left(\bigvee_{i=0}^k \tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1} > \tilde{b}_i\right) \\ &\geq 1 - \sum_{i=0}^k (1 - \text{cdf}(\alpha_i)). \end{aligned} \quad (22)$$

The overall probability that the robot does not collide with any obstacle for the duration ℓ of the plan, is given by:

$$p\left(\bigwedge_{t=0}^{\ell} \mathbf{x}_t \in \mathcal{X}_F\right) = \prod_{t=0}^{\ell} p(\mathbf{x}_{t|t-1} \in \mathcal{X}_F), \quad (23)$$

and the overall probability of collision is provided by the complement $(1 - p(\bigwedge_{t=0}^{\ell} \mathbf{x}_t \in \mathcal{X}_F))$.

V. LOCAL CONVEXIFICATION OF FREE SPACE

We extend our analysis to non-convex regions by truncating the joint conditional distributions with respect to linear constraints that define a locally convex region of free space containing the robot. For the sake of simplicity, we assume that only the robot position is relevant for collision detection. Given the conditional distribution at stage t , $\mathcal{N}[\hat{\mathbf{y}}_{t|t-1} + [\mathbf{x}_t^*], R_{t|t-1}]$, the distribution of the robot position is given by the marginal $\mathcal{N}[\hat{\mathbf{p}}_{t|t-1}, \Sigma_{t|t-1}]$, computed over the state dimensions that describe the robot position. We outline a greedy method that computes a locally convex region of free space such that the probability that the distribution $\mathcal{N}[\hat{\mathbf{p}}_{t|t-1}, \Sigma_{t|t-1}]$ lies beyond the convex region is minimal.

Adopting the approach suggested in [24], we linearly transform the environment geometry by applying the transform U_t^{-1} , where $\Sigma_t = U_t U_t^T$ is the Cholesky decomposition. This transforms the uncertainty distribution of the robot position to a Gaussian distribution with zero mean and unit variance, which is a unit sphere in Euclidean space centered at the origin. The spherical symmetry simplifies the task of constructing a nonconservative convex region of free space around the distribution of the position of the robot.

We construct the convex region using a sequential process. We consider the closest point on the obstacle geometry from the origin. The linear truncation constraint $\mathbf{a}_i^T \mathbf{p}_{t|t-1} \leq b_i$, is defined by the normal to the vector of closest approach to

the obstacle. We then prune away all geometry that lies in the infeasible half space $\mathbf{a}_i^T \mathbf{p}_{t|t-1} > b_i$ of the constraint, and continue the process by considering the closest point on the remaining obstacle geometry to the origin. This procedure is repeated until all geometry has been pruned away. It is important to note that our convexification method works in a greedy fashion and is not guaranteed to find the least conservative convex bounding region.

In our implementation, we assume that the obstacles are defined in a piecewise linear fashion using either line segments in \mathbb{R}^2 , or polygons in \mathbb{R}^3 . We only consider obstacles that are contained within 3 standard deviations of the distribution (corresponding to a distance of 3 units from the origin in the transformed environment). Instead of performing a linear-time search by iterating over all obstacles, we use a BSP tree [5] data structure to determine obstacles within the given search radius, in logarithmic time.

VI. RESULTS

We present simulation results for two scenarios: a car-like robot with second order dynamics and a nonholonomic bevel-tip flexible needle. In both scenarios, the robot moves subject to stochastic dynamics under the guidance of partial and noisy state measurements. In each case, we initialize our method with a nominal plan computed using an RRT planner [13]. We empirically demonstrate that our method provides a more accurate estimation of the a priori probability distributions of the robot state and the probability of collision compared to prior methods. We also validate our method by comparing the estimated collision probability with the probability computed by Monte-Carlo simulations (considered as ground truth). We implemented our method in C++ and tested it on a 3.33 Ghz Intel® i7™ PC.

A. Car-like Robot with Second Order Dynamics

We apply our method to a nonholonomic car-like robot with second order dynamics, navigating in a 2D environment with obstacles. The state of the robot, $\mathbf{x} = [x, y, \theta, v]^T \in \mathbb{R}^4$, consists of its position $[x, y]$, its orientation θ , and its speed v . The control input $\mathbf{u} = [a, \phi]^T \in \mathbb{R}^2$, consists of the acceleration a , and steering angle ϕ , corrupted by motion noise $\mathbf{m} = [\tilde{a}, \tilde{\phi}]^T \sim \mathcal{N}[\mathbf{0}, \mathbf{M}]$. This gives the following stochastic dynamics model:

$$\mathbf{f}[\mathbf{x}, \mathbf{u}, \mathbf{m}] = \begin{bmatrix} x + \tau v \cos \theta \\ y + \tau v \sin \theta \\ \theta + \tau v \tan(\phi + \tilde{\phi})/d \\ v + \tau(a + \tilde{a}) \end{bmatrix}, \quad (24)$$

where τ is the time step, and d is the length of the car.

The robot localizes itself using noisy signal measurements from two beacons b_1 and b_2 , placed in the environment at locations $[\tilde{x}_1, \tilde{y}_1]$ and $[\tilde{x}_2, \tilde{y}_2]$ respectively. The strength of the signal decays quadratically with the distance to the beacon. The robot also measures its current speed using an on-board speedometer. The observation vector $\mathbf{z} \in \mathbb{R}^3$, consists of two readings of signal strengths from the beacons and a speed measurement from the speedometer, corrupted by sensing

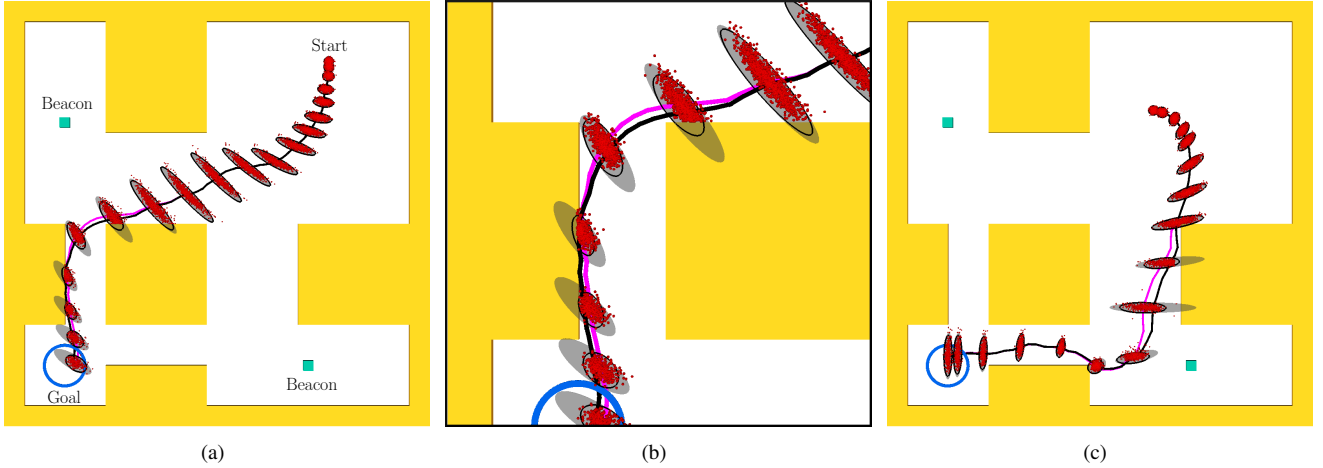


Fig. 4. Car-like robot with second order dynamics: (a) An initial plan (black) is computed using an RRT planner. The unconditional distributions (solid gray ellipses showing 3 standard deviations of the Gaussian distribution) provide an overly conservative approximation of the uncertainty. Samples of the true distribution computed by Monte Carlo simulations are shown by the red dots. Our method computes conditional distributions (black ellipses showing 3 standard deviations), which provide an accurate estimate of the true distributions. The probability of collision estimated by our method for this example is 68.87%, while the probability determined by Monte-Carlo simulations is 66.26%. Our method provides an accurate, yet conservative, approximation of the collision probability. (b) Zoomed in view of the conditional distributions in the narrow corridor in the environment. The mean of the conditional distribution (magenta) deviates from the nominal plan due to the truncation of the distributions due to obstacle collisions. (c) Conditional and unconditional distributions computed along a second plan. The probability of collision estimated by our method for this example is 44.78%, while the probability determined by Monte-Carlo simulations is 42.32%.

noise $\mathbf{n} \sim \mathcal{N}[\mathbf{0}, N]$. This gives us the following stochastic measurement model:

$$\mathbf{h}[\mathbf{x}, \mathbf{n}] = \begin{bmatrix} 1/((x - \check{x}_1)^2 + (y - \check{y}_1)^2 + 1) \\ 1/((x - \check{x}_2)^2 + (y - \check{y}_2)^2 + 1) \end{bmatrix}_v + \mathbf{n}. \quad (25)$$

We consider a non-convex environment with narrow corridors to evaluate the performance of our method (Fig. 4(a)). Fig. 4(b) shows the discrepancy between the unconditional and conditional distributions in the presence of obstacles. The conditional distributions computed using our method provide an accurate estimate of the distribution of the collision free robot states along the plan, thus providing an accurate estimate of the probability of collision. Fig. 4(c) shows how the mean of the conditional distribution can deviate significantly in the close vicinity of obstacles. Interestingly, the conditional and unconditional distributions become identical towards the end of the plan in the absence of obstacles.

To validate our method, we generated a set of 100 plans using the RRT planner using randomly initialized start states. We estimated the probability of collision using our method for each of these plans. This took 0.09 seconds. We also used a brute-force approach to estimate the ground truth probability of collision using Monte-Carlo sampling. We performed 10,000 simulations of executions of each plan using the given controller and Kalman filter, and with artificially generated motion and measurement noise, and counted the number of collision free executions. This took 271 seconds, which is orders of magnitude slower than our method.

Table I compares the collision probability estimate computed using our method with the probability computed using brute-force Monte-Carlo sampling, applying Boole's inequality to the unconditional distributions [27], and LQG-MP [24]. We use the mean error as a metric to compare the probability

estimates to the ground truth probability computed using sampling, across 100 randomly generated plans.

For this example, the mean error in estimation using our method as compared to the ground truth probability is 3.0%. The estimate computing using our method is much more accurate than the collision quality metric provided by LQG-MP (mean error of 52.2%) [24] and the collision probability computed using the unconditional distributions directly by assuming independent probabilities (mean error of 28.0%) [27], with negligible computational overhead. It is important to note that all the estimation methods, including ours, provide a conservative bound for the collision probability.

Robot	Sampling Time (secs)	Our method		Unconditional		LQG-MP	
		MAE prob(%)	Time (secs)	MAE prob(%)	Time (secs)	MAE prob(%)	Time (secs)
car	271	3.0 (± 1.9)	0.09	28.0 (± 15.1)	0.06	52.2 (± 15.4)	0.04
needle	388	5.0 (± 3.0)	1.4	20.7 (± 6.9)	1.2	61.7 (± 11.5)	1.0

TABLE I

COMPARISON OF OUR METHOD WITH EXISTING APPROACHES OVER 100 RANDOMLY GENERATED PLANS IN TERMS OF MEAN ERROR FROM GROUND TRUTH PROBABILITY ESTIMATED BY SAMPLING.

B. Nonholonomic Bevel-tip Flexible Needle

We also apply our method to a nonholonomic bevel-tip flexible needle [4], navigating in a 3D environment with obstacles. This class of needles offers improved mobility, enabling needles to reach previously inaccessible targets while maneuvering around sensitive or impenetrable areas.

The state of the needle \mathbf{x} , is described by the 4×4 matrix $X = \begin{bmatrix} R & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$, where $\mathbf{p} \in \mathbb{R}^3$ is the position of the needle tip, and $R \in SO(3)$ is the rotation matrix that encodes the orientation of the needle tip, relative to a world coordinate

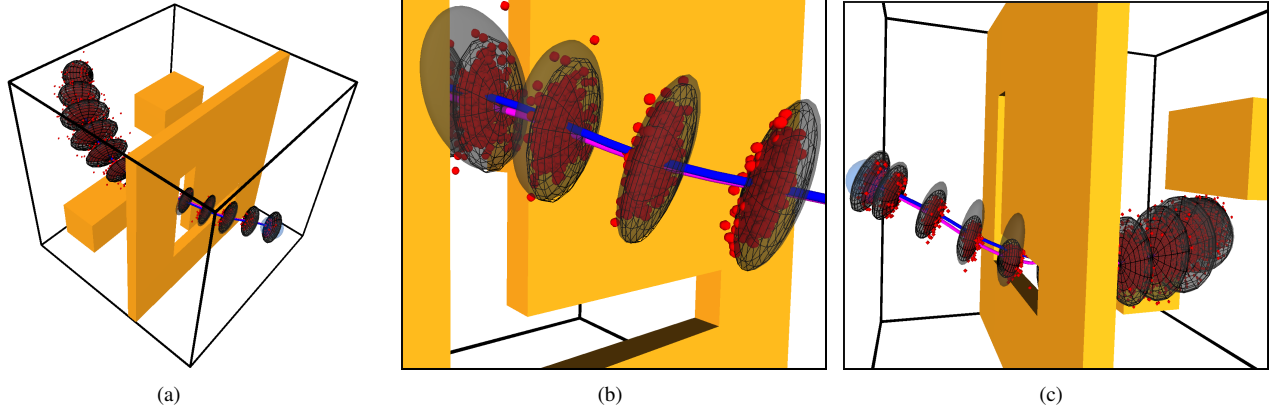


Fig. 5. Nonholonomic bevel-tip steerable needle: (a) The unconditional distributions (solid gray ellipsoids, 3 standard deviations) provide an overly conservative approximation of the uncertainty. Our method computes conditional distributions (black wireframe ellipsoids, 3 standard deviations), which provide an accurate estimate of the probability distributions of the feasible robot states (shown in red). The probability of collision estimated by our method for the given plan is 54.5%, while the probability determined by Monte-Carlo simulations is 50.4%. Our method provides an accurate, yet conservative approximation of the collision probability. (b) Zoomed in view of the conditional distributions in the narrow corridor in the environment. The mean of the conditional distribution (magenta) deviates from the nominal plan due to the truncation. (c) The probability of collision estimated by our method for a second plan is 43.9%, while the probability determined by Monte-Carlo simulations is 42.2%.

frame. The needle naturally moves along constant curvature paths when inserted into tissue, but the curvature of the needle motion can be varied by duty cycled spinning of the needle during insertion. Under these modeling assumptions [26], the control input $\mathbf{u} = [v, w, \kappa]^T \in \mathbb{R}^3$, consists of the insertion speed v , rotation speed applied at the base of the needle w , and the curvature κ .

It is convenient to describe the dynamics of the needle tip in terms of the instantaneous twist $U \in se(3)$ expressed in a local coordinate frame attached to the needle tip, given by:

$$U = \begin{bmatrix} [\mathbf{w}] & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{w} = [v\kappa \ 0 \ w]^T, \mathbf{v} = [0 \ 0 \ v]^T, \quad (26)$$

where the notation $[\mathbf{s}]$ for a vector $\mathbf{s} \in \mathbb{R}^3$ refers to the 3×3 skew-symmetric cross-product matrix. The instantaneous twist \tilde{U} that encodes the additive motion noise $\mathbf{m} = [\tilde{\mathbf{v}} \ \tilde{\mathbf{w}}]^T \sim \mathcal{N}[\mathbf{0}, \mathbf{M}]$, can be similarly expressed.

Given a time step τ , the stochastic discrete-time dynamics of the needle-tip is given by the following model:

$$\mathbf{f}[\mathbf{x}, \mathbf{u}, \mathbf{m}] = X \exp(\tau(U + \tilde{U})). \quad (27)$$

We also assume that we receive partial, noisy feedback on only the position of the needle tip \mathbf{p} , and not its orientation. This is a reasonable assumption since current medical imaging technologies such as ultrasound do not allow for measuring the full state of the needle tip (as the imaging resolution is often too low to infer its orientation). The noise in the sensor measurement is modeled as $\mathbf{n} \sim \mathcal{N}[\mathbf{0}, \mathbf{N}]$. This gives the following stochastic measurement model:

$$\mathbf{h}[\mathbf{x}, \mathbf{n}] = \mathbf{p} + \mathbf{n}. \quad (28)$$

We follow the approach suggested in [26] to approximate the given nonlinear dynamics and measurement models with local linearizations around the nominal plan.

We consider a non-convex environment with a L-shaped narrow corridor and other obstacles to evaluate the perfor-

mance of our method (Fig. 5(a)). Fig. 5(b) shows the discrepancy between the unconditional and conditional distributions in the presence of obstacles. The conditional distributions computed using our method provide an accurate estimate of the distribution of the collision free robot states along the plan, thus providing an accurate estimate of the probability of collision.

Similar to the analysis performed for the previous example, we generated a set of 100 plans using the RRT planner starting from the goal state. We estimated the probability of collision using our method for each of these plans. This took 1.4 seconds. We also used a brute-force approach to estimate the ground truth probability of collision using Monte-Carlo sampling. We performed 10,000 simulations of executions of each plan with artificially generated motion and measurement noise, and counted the number of collision free executions. This took 388 seconds, which is orders of magnitude slower than our method.

Table I shows the results of how our method compares to the ground truth and prior approaches. For the given example, the mean error in estimation using our method as compared to the ground truth probability is 5.0%. The estimate computed using our method is much more accurate than the collision quality metric provided by LQG-MP (MAE 61.7%) [24] and the collision probability computed using the unconditional distributions directly by assuming independent probabilities (MAE 20.7%) [27], while incurring negligible computational overhead.

VII. DISCUSSION AND FUTURE WORK

We have presented an analytical method to estimate a priori the probability of collision for a mobile robot operating under Gaussian motion and sensing uncertainty. We have shown that it is necessary to consider the correlations between the a priori probability distributions of the robot state, to accurately estimate the true distributions and consequently, the probability of collision. We have also proposed a novel

method for approximating the distribution of feasible states with truncated Gaussian distributions. Our method is directly applicable to a variety of motion planning under uncertainty methods [24], [27], [3], [25], [23] to improve the quality and safety of the computed plans.

We plan to integrate our method with other motion planning under uncertainty methods, including improving the safety of computed plans in deformable environments [14], and elegantly handle state constraints in optimization based approaches [25]. We also plan to extend our method to handle more general cases where the orientation of the robot is also relevant for collision detection. A natural extension is to investigate how our method can be extended to incorporate uncertainty in sensing obstacles in the environment [9]. Another interesting avenue for future research is to look at applications with unilateral contacts where the distribution of feasible states is no longer Gaussian [7]. It might be possible to approximate such non-Gaussian distributions using mixture of Gaussians and truncating them with respect to constraints on the robot state.

REFERENCES

- [1] R. Alterovitz, T. Simeon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proc. Robotics: Science and Systems (RSS)*, 2007.
- [2] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proc. AIAA Guidance, Navigation, and Control Conf.*, 2009.
- [3] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.
- [4] N. J. Cowan, K. Goldberg, G. S. Chirikjian, G. Fichtinger, R. Alterovitz, K. B. Reed, V. Kallem, W. Park, S. Misra, and A. M. Okamura, "Robotic needle steering: Design, modeling, planning, and image guidance," in *Surgical Robotics: System Applications and Visions*, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer, 2011, ch. 23, pp. 557–582.
- [5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag, 2008.
- [6] N. E. du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Trans. Robotics*, vol. 27, pp. 809–815, 2011.
- [7] T. Erez and W. D. Smart, "A scalable method for solving high-dimensional continuous pomdps using local approximation," in *Conf. on Uncertainty in Artificial Intelligence*, 2010.
- [8] M. Greytak, "Integrated motion planning and model learning for mobile robots with application to marine vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
- [9] L. Guibas, D. Hsu, H. Kurniawati, and E. Rehman, "Bounded uncertainty roadmaps for path planning," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2008.
- [10] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*. John Wiley & Sons, 1994, vol. 1.
- [11] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Science and Systems (RSS)*, 2008.
- [12] A. Lambert, D. Gruyer, and G. S. Pierre, "A fast monte carlo algorithm for collision probability estimation," in *Int. Conf. on Control, Automation, Robotics and Vision (ICARV)*, 2006, pp. 406–411.
- [13] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, Available at <http://planning.cs.uiuc.edu>.
- [14] S. Patil, J. van den Berg, and R. Alterovitz, "Motion planning under uncertainty in highly deformable environments," in *Proc. Robotics: Science and Systems (RSS)*, 2011.
- [15] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robotics: Science and Systems (RSS)*, 2010.
- [16] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1448–1465, 2009.
- [17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [18] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation - mobile robot navigation with uncertainty in dynamic environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1999, pp. 35–40.
- [19] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [20] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, 1994.
- [21] S. Thrun, "Toward robotic cars," *Communications of the ACM*, vol. 53, pp. 99–106, 2010.
- [22] M. Toussaint, "Pros and cons of truncated Gaussian EP in the context of Approximate Inference Control," in *NIPS Workshop on Probabilistic Approaches for Robotics and Control*, 2009.
- [23] —, "Robot trajectory optimization using approximate inference," in *Proc. Int. Conf. on Machine Learning*, 2009, pp. 1049–1056.
- [24] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," in *Int. Journal of Robotics Research*, vol. 30, no. 7, 2011, pp. 895–913.
- [25] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using differential dynamic programming in belief space," in *Int. Symp. on Robotics Research*, 2011.
- [26] J. van den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Goldberg, "LQG-based planning, sensing, and control of steerable needles," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2010, pp. 373–389.
- [27] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011.

APPENDIX

TRUNCATED GAUSSIAN DISTRIBUTIONS

Given a distribution $Y' \sim \mathcal{N}[\mu_Y', \Sigma_Y']$ and a joint Gaussian distribution (X, Y) :

$$(X, Y) \sim \mathcal{N} \left[\begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix}, \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix} \right], \quad (29)$$

the conditional distribution $(X|Y = Y')$ is given by [17]:

$$(X|Y = Y') \sim \mathcal{N}[\mu_X - L(\mu_Y - \mu_Y'), \Sigma_X - L(\Sigma_Y - \Sigma_Y')L^T] \quad (30)$$

where $L = \Sigma_{XY}\Sigma_Y^{-1}$.

We can construct the joint distribution of the conditional distribution $\mathbf{y}_{t|t-1} \sim \mathcal{N}[\hat{\mathbf{y}}_{t|t-1}, R_{t|t-1}]$, and the transformed 1D distribution $y_{t|t-1}^i \sim \mathcal{N}[\tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1}, \tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i]$, according to Eqn. (29) as:

$$(\mathbf{y}_{t|t-1}, y_{t|t-1}^i) \sim \mathcal{N} \left[\begin{bmatrix} \hat{\mathbf{y}}_{t|t-1} \\ \tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1} \end{bmatrix}, \begin{bmatrix} R_{t|t-1} & R_{t|t-1} \tilde{\mathbf{a}}_i \\ \tilde{\mathbf{a}}_i^T R_{t|t-1} & \tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i \end{bmatrix} \right]. \quad (31)$$

Using Eqn. (30), we reconstruct the truncated mean and variance of the joint distribution by conditioning on the truncated 1D distribution $y_{t|t}^i \sim \mathcal{N}[\mu_i, \sigma_i^2]$ (Eqn. 17, 18), according to:

$$(\mathbf{y}_{t|t-1} | y_{t|t-1}^i = y_{t|t}^i) \sim \mathcal{N}[\hat{\mathbf{y}}_{t|t-1} - L(\tilde{\mathbf{a}}_i^T \hat{\mathbf{y}}_{t|t-1} - \mu_i), R_{t|t-1} - L(\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i - \sigma_i^2)L^T], \quad (32)$$

where $L = \frac{R_{t|t-1} \tilde{\mathbf{a}}_i}{\tilde{\mathbf{a}}_i^T R_{t|t-1} \tilde{\mathbf{a}}_i}$, which is exactly as in Eqns. (20, 21).