

Probabilistic Collision Checking with Chance Constraints

Noel E. Du Toit, *Member, IEEE*,
and Joel W. Burdick, *Member, IEEE*,

Abstract—Obstacle avoidance, and by extension collision checking, is a basic requirement for robot autonomy. Most classical approaches to collision checking ignore the uncertainties associated with the robot and obstacle’s geometry and position. It is natural to use a probabilistic description of the uncertainties. However, constraint satisfaction cannot be guaranteed in this case and collision constraints must instead be converted to chance constraints. Standard results for linear probabilistic constraint evaluation have been applied to probabilistic collision evaluation, but this approach ignores the uncertainty associated with the sensed obstacle. An alternative formulation of probabilistic collision checking that accounts for robot and obstacle uncertainty is presented which allows for dependent object distributions (e.g., interactive robot-obstacle models). In order to efficiently enforce the resulting collision chance constraints, an approximation is proposed and the validity of this approximation is evaluated. The results presented here have been applied to robot motion planning in dynamic, uncertain environments.

Index Terms—Chance Constraints, Probabilistic Collision Checking, Collision Avoidance

I. INTRODUCTION

Obstacle avoidance is a basic requirement for robots operating in practical environments. This is especially true for robots operating in *Dynamic, Uncertain Environments* (DUEs). In such environments, robots must work in close proximity to many other moving agents whose future actions and reactions are often not well known. The robot is plagued by noise in its own state measurements and in the sensing of static and moving obstacles. An example of a DUE application is a service robot which must move through a swarm of humans in a cafeteria during a busy lunch hour in order to deliver food items. This paper presents an obstacle-robot collision checking method that accounts for the probabilistic uncertainties associated with the robot and obstacle position states.

Classical approaches to obstacle avoidance react locally to the obstacles, but often ignore state uncertainty [1], [2], [3]. Approaches include the Velocity Obstacle approach [4], the Dynamic Window approach [5], and Inevitable Collision States (ICS) [6]. ICS differs from collision checking in that the goal of ICS is to identify robot configurations for which no sequence of actions exist that allows the robot to avoid a collision. Recent extensions of the ICS problem to the uncertain case have been proposed [7], [8].

In this work, the collision checking problem is considered as part of a robot motion planning implementation [9], [10]. Obstacle avoidance is obtained by recursively solving the planning problem while enforcing collision constraints along the trajectory. The specific problem considered in this work is probabilistic collision checking between uncertain configurations for two objects, referred to as collision chance constraints. Blackmore et al. [11] extended results for linear chance constraints to the problem of polygonal obstacle avoidance, but the uncertainty associated with the obstacle is ignored. The main contribution of this paper is an alternative formulation of the probability of collision which accounts for both robot and obstacle uncertainty. The resulting formulation is a generalization of the formulation presented by Lambert et al. [12]. Finally, the motion planning problem in dynamic, uncertain environments is inherently non-convex. Computational considerations for this optimization problem motivate the development of efficient constraint evaluation routines.

N. Du Toit and J. Burdick are with the Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA, 91125 USA e-mail: {ndutoit, jwb}@robotics.caltech.edu.

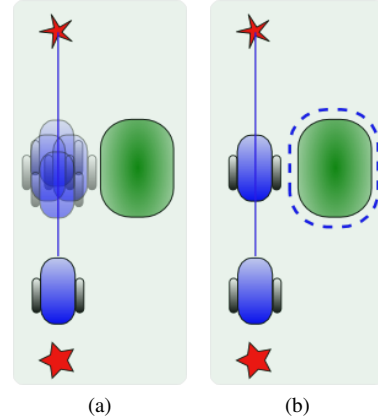


Fig. 1. (a) The robot’s position (blue) is propagated, resulting in an uncertain state (represented by multiple transparent robots). The static obstacle (green) is to be avoided. (b) The constraint is tightened (dashed blue) by some amount that accounts for the level of confidence and the predicted state covariance.

An approximation scheme is presented to efficiently enforce the collision chance constraints and the validity of the approximation is investigated.

The layout of the paper is as follows: previous results on probabilistic linear constraints and the application of these results to probabilistic collision checking is discussed in Section II. An alternative formulation of the problem is given in Section III. An approximation is introduced and specialized for systems with normally distributed states (for independent and dependent object distributions). The validity of the approximation is investigated for disk objects with Gaussian distributed position states in Section IV. The application of the results in this paper to the robot motion planning problem in dynamic, uncertain environments [9], [10] is discussed in Section V.

II. CHANCE CONSTRAINTS

It is often desirable to impose constraints on the states of a robotic system when solving a motion planning problem (e.g., velocity constraints, and collision constraints between the robot and other objects). In general, these constraints are described as inequality functions, $c(x) \leq 0$, of the system state, $x \in \mathbb{X} \subset \mathbb{R}^{n_x}$. The system state may refer to the robot state, obstacle state, or the augmented state (robot and obstacle). However, in a probabilistic formulation, the system states are represented by unbounded distributions (e.g., normal distributions) and constraint satisfaction can not be guaranteed for all realizations of the states. It is instead necessary to introduce *chance constraints*, or probabilistic state constraints, which are of the form:

$$P(x \notin \mathbb{X}_{free}) \leq \delta \text{ or } P(x \in \mathbb{X}_{free}) \geq \alpha. \quad (1)$$

The probability of constraint violation is bounded. α and δ are *levels of confidence* and \mathbb{X}_{free} denotes the set of states where the exact, deterministic constraints are satisfied (e.g., $\mathbb{X}_{free} = \{x : c(x) \leq 0\}$).

Chance constraints have been commonly applied in Stochastic Control and Stochastic Receding Horizon Control (SRHC) (e.g., [13], [14], [15], [16], [17]). To date, two approaches have been proposed to evaluate chance constraints: (i) assume *linear constraints* on *Gaussian-distributed* state variables and convert the chance constraints into constraints on the means of the states (e.g., [11], [16]) and (ii) evaluate the constraints by Monte Carlo simulation (e.g., [18]). Approach (i) is restrictive, whereas approach (ii) can handle non-Gaussian distributions and non-linear constraints, but it is computationally intensive. Standard results for linear constraints on

Gaussian-distributed system states (approach (i)) are reviewed next to give intuition about the effects of chance constraints.

A. Individual Linear Chance Constraints

For Gaussian-distributed state variables, linear chance constraints on state space $\mathbb{X} \subset \mathbb{R}^{n_x}$ take the form $P(a^T x \leq b) \geq \alpha$, where $a^T x \in \mathbb{R}$, $x \in \mathbb{X}$, $b \in \mathbb{R}$. Let $\hat{x} \triangleq E[x]$ be the mean of the state, and let the covariance matrix of x be $\Sigma \triangleq E[(x - \hat{x})(x - \hat{x})^T]$.

The chance constraint $P(a^T x \leq b) \geq \alpha$ is satisfied iff

$$a^T \hat{x} + F^{-1}(\alpha) \times \sqrt{a^T \Sigma a} \leq b \quad (2)$$

where $F^{-1}(\alpha)$ is the inverse of the cumulative distribution function for a *standard* scalar Gaussian variable (zero mean and unit standard deviation [19]). The chance constraint is converted into a constraint on the mean of the robot state. The constraint is *tightened*¹ by some amount that is a function of the level of confidence and covariance (illustrated in Figure 1).

B. Using Linear Chance Constraints for Collision Checking

When using linear chance constraints to impose collision constraints between the robot and polygonal objects, sets of linear constraints must be satisfied simultaneously (or jointly) [11]. Two approaches for imposing joint constraints have emerged: (i) probability mass is allocated to the different components of the joint constraints, each of which is then implemented using the individual constraint results, and (ii) use an ellipsoidal approximation (for Gaussian-distributed variables).

1) *Probability Mass Allocation*: To impose the set of constraints, it is convenient to formulate the problem in terms of the probability of not being in free space: $P(x \notin \mathbb{X}_{free}) = 1 - P(x \in \mathbb{X}_{free}) \leq 1 - \alpha \triangleq \delta$. A part of the confidence, δ , is then assigned to each of m individual constraints that are to be satisfied [11], [16], [20]. Define the constraint violation event as A_j , then $x \in \mathbb{X}_{free} \Leftrightarrow \bigcap_{j=1}^m \overline{A_j}$ where negation is represented by the overbar. The negation of this expression is given by De Morgan's law:

$$\overline{\bigcap_{j=1}^m A_j} = \bigcup_{j=1}^m \overline{A_j}. \quad (3)$$

From Boole's inequality, the probability of any constraint violation is bounded by:

$$P\left(\bigcup_{j=1}^m A_j\right) \leq \sum_{j=1}^m P(A_j). \quad (4)$$

Thus, if a portion, ϵ_j , of δ is assigned to each component of the constraint ($P(A_j) \leq \epsilon_j$), then:

$$\sum_{j=1}^m \epsilon_j = \delta \implies P(x \in \mathbb{X}_{free}) \geq \alpha. \quad (5)$$

It is necessary to specify the level of certainty, ϵ_i , for each of the constraint components. A naive approach is to assign a constant value, $\epsilon_i = \frac{\delta}{m}$, but this has been shown to be very conservative [20]. Ono and Williams [21] and Blackmore [20] introduced the notion of risk allocation, where more probability mass is assigned to parts of the trajectory where collisions are more likely.

2) *Ellipsoidal Approximation*: The probability mass contained in the free space can be approximated by the probability mass contained in the uncertainty ellipse associated with the jointly Gaussian distributed states. Then, the uncertainty ellipse is confined to the free space [19]. Again, the ellipsoidal approximation is very conservative [20].

¹Constraint tightening refers to growing the constraint to account for uncertainty.

C. Discussion

These standard results for linear constraints have been applied to probabilistic collision checking, but uncertainty in obstacle position or shape is ignored. However, it is desirable to evaluate collision chance constraints when *both* the obstacle and robot locations are uncertain. An alternative formulation of the probability of collision is presented next.

III. PROBABILISTIC COLLISION CHECKING

Probabilistic collision checking can be formulated as a chance constraint of the form $P(C) \leq \delta$, where C is the collision condition (defined below). Let $\mathbb{W} \subset \mathbb{R}^{n_x}$ be the *workspace* of the robot. Assume a rigid-body² robot. Assign a body-fixed reference frame to the robot, located at x_R in the global reference frame. Let $\mathbb{X}_R(x_R) \subset \mathbb{W}$ be the set of points occupied by the robot. In a similar manner, assume a rigid-body obstacle (the obstacle might be moving) with a body-fixed reference frame located at x_A in the global frame. Let $\mathbb{X}_A(x_A) \subset \mathbb{W}$ be the set of points occupied by the obstacle. The *collision condition* is defined as $C(x_R, x_A) : \mathbb{X}_R(x_R) \cap \mathbb{X}_A(x_A) \neq \{\emptyset\}$. The probability of collision is defined as:

$$P(C) = \int_{x_R} \int_{x_A} I_C(x_A, x_R) p(x_R, x_A) dx_R dx_A \quad (6)$$

where I_C is the indicator function, defined as:

$$I_C(x_A, x_R) = \begin{cases} 1 & \text{if } \mathbb{X}_R(x_R) \cap \mathbb{X}_A(x_A) \neq \{\emptyset\} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

For obstacle avoidance, collision chance constraints must be simultaneously enforced over the trajectory. The results from Section II-B1 can be applied directly where the individual collision constraint violations are the events A_j . A fraction of the total level of confidence is assigned at each stage to the individual collision constraints.

The resulting formulation is a generalization of the result³ from Lambert et al. [12] to account for dependence between the robot and obstacle distributions (e.g., when the object and robot motions are dependent, such as in interactive scenarios). This integral function is difficult to evaluate for general robot and obstacle geometries. One strategy is to use Monte-Carlo Simulation, but that method can be computationally intensive [12] (see also Section IV-A). As an alternative, an approximate solution to the probability of collision is presented next.

A. Approximate Probability of Collision

For simplicity, assume a spherical robot of radius ϵ ($\mathbb{X}_R \triangleq \mathcal{S}(x_R, \epsilon)$) and a point obstacle⁴. The collision condition is defined as $C : x_A \in \mathcal{S}(x_R, \epsilon)$. Let V_S be the volume of this sphere. The probability of collision, eq. (6), is rewritten as:

$$P(C) = \int_{x_R} \left[\int_{x_A \in \mathcal{S}(x_R, \epsilon)} p(x_A | x_R) dx_A \right] p(x_R) dx_R. \quad (8)$$

Assuming that the robot has a small radius ($\epsilon \ll 1$), the inner integral can be approximated with a constant value of the conditional

²An idealized collection of points with a fixed relative position

³The motivating formulation of the probability of collision in Section III, eq. (5) of [12] is incorrect: the probability of collision requires a double integral over the space and the obtained result does not account for object sizes, as noted by the authors. Also, the resulting function is not a probability density function since there is no uncertain parameter. Instead, it is simply a deterministic function in the form of a Gaussian distribution. Later results (e.g., eq. (10) in Section IV of [12]) are correct.

⁴This analysis can readily be extended to the case where both the robot and obstacle have spherical geometries by considering the configuration space.

distribution of the obstacle evaluated at the robot location, multiplied by the volume occupied by the robot, V_S :

$$\int_{x_A \in \mathcal{S}(x_R, \varepsilon)} p(x_A|x_R) dx_A \approx V_S \times p(x_A = x_R|x_R). \quad (9)$$

The approximate probability of collision for small objects is therefore:

$$P(C) \approx V_S \times \int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R. \quad (10)$$

This integral can be evaluated in closed form for systems where the robot and obstacles have Gaussian-distributed position states which are (i) independent and (ii) dependent.

B. Objects with Independent, Gaussian-distributed States

Lemma 1: Assume that the robot's and object's position uncertainties are described with *independent* Gaussian distributions: $x_R \sim N(\hat{x}_R, \Sigma_R)$ and $x_A \sim N(\hat{x}_A, \Sigma_A)$, then:

$$\int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R = \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}(\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1}(\hat{x}_R - \hat{x}_A)\right] \quad (11)$$

where $\Sigma_C \triangleq \Sigma_R + \Sigma_A$ is the combined position covariance.

Proof: Using the independence assumption:

$$\int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R = \int_{x_R} N_{x_R}(\hat{x}_A, \Sigma_A) N_{x_R}(\hat{x}_R, \Sigma_R) dx_R. \quad (12)$$

It is known that the product of independent Gaussian distributions is a weighted Gaussian (e.g., [22]):

$$N_{x_R}(\hat{x}_A, \Sigma_A) N_{x_R}(\hat{x}_R, \Sigma_R) = \beta N_{x_R}(m_c, \Sigma_c) \quad (13)$$

where

$$\beta \triangleq \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}(\hat{x}^R - \hat{x}^A)^T \Sigma_C^{-1}(\hat{x}^R - \hat{x}^A)\right] \quad (14)$$

and

$$\begin{aligned} m_c &= \Sigma_c (\Sigma_R^{-1} \hat{x}_R + \Sigma_A^{-1} \hat{x}_A) \\ \Sigma_c &= (\Sigma_R^{-1} + \Sigma_A^{-1})^{-1}. \end{aligned} \quad (15)$$

$\Sigma_C \triangleq \Sigma_R + \Sigma_A$ is the combined position covariance. Eq. (12) becomes:

$$\int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R = \beta \int_{x_R} N_{x_R}(m_c, \Sigma_c) dx_R \quad (16)$$

and since the integral evaluates to one, the result is obtained. \blacksquare

This analysis provides intuition about the appropriate combination of the covariances of the two distributions. The probability of collision is approximately given by:

$$P(C) \approx V_S \times \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}(\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1}(\hat{x}_R - \hat{x}_A)\right]. \quad (17)$$

The objective is to convert the constraint $P(C) \leq \delta$ into a constraint on the robot mean state in terms of the obstacle mean state. From (17), the equivalent constraint is:

$$\begin{aligned} (\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1}(\hat{x}_R - \hat{x}_A) &\geq -2 \ln\left(\frac{\delta}{\sqrt{\det(2\pi\Sigma_C)} V_S}\right) \\ &\triangleq \kappa \end{aligned} \quad (18)$$

where κ is a function of the level of certainty, the sizes of the robot and obstacle, and the combined covariance of the position states. This constraint is in the form of an ellipsoid around the obstacle that the robot has to avoid to ensure that the chance collision constraint is satisfied.

C. Objects with Dependent, Gaussian-distributed States

Lemma 2: Assume that the robot and agent state (augmented state) are jointly Gaussian (e.g., when modeling interaction between the robot and the agent [9]):

$$p(x) = N\left(\begin{bmatrix} \hat{x}_R \\ \hat{x}_A \end{bmatrix}, \begin{bmatrix} \Sigma_R & \Sigma_M \\ \Sigma_M^T & \Sigma_A \end{bmatrix}\right). \quad (19)$$

then

$$\int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R = \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}(\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1}(\hat{x}_R - \hat{x}_A)\right] \quad (20)$$

where $\Sigma_C \triangleq \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T$.

Proof: The dependence condition implies that $\Sigma_M \neq 0$. Then, the marginal distribution of the robot state is $p(x_R) = N(\hat{x}_R, \Sigma_R)$ and the conditional distribution of the agent state is $p(x_A|x_R) = N(\bar{x}_A(x_R), \bar{\Sigma}_A)$, where $\bar{x}_A(x_R) = \hat{x}_A + \Sigma_M^T \Sigma_R^{-1}(x_R - \hat{x}_R)$ and $\bar{\Sigma}_A = \Sigma_A - \Sigma_M^T \Sigma_R^{-1} \Sigma_M$.

The product of the two Gaussian distributions has the functional form:

$$N(\hat{x}_R, \Sigma_R) \times N(\bar{x}_A(x_R), \bar{\Sigma}_A) = \eta \exp(-L(x_R, \hat{x}_R, \hat{x}_A)) \quad (21)$$

where

$$\eta = \frac{1}{\sqrt{\det(2\pi\Sigma_R)}} \frac{1}{\sqrt{\det(2\pi\bar{\Sigma}_A)}} \quad (22)$$

and

$$\begin{aligned} L(x_R, \hat{x}_R, \hat{x}_A) &= \frac{1}{2}(x_R - \hat{x}_R)^T \Sigma_R^{-1}(x_R - \hat{x}_R) + \\ &\frac{1}{2}(x_A - \bar{x}_A(x_R))^T \bar{\Sigma}_A^{-1}(x_A - \bar{x}_A(x_R)). \end{aligned} \quad (23)$$

Partition this function as $L(x_R, \hat{x}_R, \hat{x}_A) = L_1(x_R, \hat{x}_R, \hat{x}_A) + L_2(\hat{x}_R, \hat{x}_A)$ so that the integral can be evaluated efficiently. With this partition we obtain:

$$\int_{x_R} p(x_A = x_R|x_R) p(x_R) dx_R = \eta \times \left[\int_{x_R} \exp(-L_1(x_R, \hat{x}_R, \hat{x}_A)) dx_R \right] \exp(-L_2(\hat{x}_R, \hat{x}_A)). \quad (24)$$

By evaluating the first and second partial derivatives⁵ with respect to x_R , $L_1(x_R, \hat{x}_R, \hat{x}_A)$ can be written as a normal distribution in terms

⁵For a generic normal distribution of random variable z , the mean, μ , and the covariance, Γ , of the distribution can be recovered from the first and second derivatives of the function $L(z)$ [23]:

$$\begin{aligned} N(z; \mu, \Gamma) &= \frac{1}{\sqrt{\det(2\pi\Gamma)}} \exp(-L(z)) \\ L(z) &= \frac{1}{2}(z - \mu)^T \Gamma^{-1}(z - \mu) \end{aligned}$$

then

$$\begin{aligned} \frac{\partial L(z)}{\partial z} &= \Gamma^{-1}(z - \mu) \quad \text{and} \quad \frac{\partial L(z)}{\partial z} = 0 \Rightarrow \bar{z} = \mu \\ \frac{\partial^2 L(z)}{\partial z^2} &= \Gamma^{-1}. \end{aligned}$$

of x_R :

$$\frac{\partial L(x_R, \hat{x}_R, \hat{x}_A)}{\partial x_R} = 0 \Rightarrow \tilde{x}_R = \Psi \bar{x}_R \quad (25)$$

where

$$\bar{x}_R \triangleq \left(\Sigma_R^{-1} - \bar{\Sigma}_A^{-1} \Sigma_M^T \Sigma_R^{-1} + \Sigma_R^{-1} \Sigma_M \bar{\Sigma}_A^{-1} \Sigma_M^T \Sigma_R^{-1} \right) \hat{x}_R + \left(\bar{\Sigma}_A^{-1} - \Sigma_R^{-1} \Sigma_M \bar{\Sigma}_A^{-1} \right) \hat{x}_A \quad (26)$$

and

$$\Psi \triangleq \left(\frac{\partial^2 L(x_R, \hat{x}_R, \hat{x}_A)}{\partial x_R^2} \right)^{-1} = \Sigma_R - (\Sigma_R - \Sigma_M) \left(\Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T \right) \left(\Sigma_R - \Sigma_M^T \right). \quad (27)$$

Define $L_1(x_R, \hat{x}_R, \hat{x}_A) \triangleq \frac{1}{2}(x_R - \bar{x}_R)^T \Psi^{-1}(x_R - \bar{x}_R)$, and note that

$$\int_{x_R} \exp(-L_1(x_R, \hat{x}_R, \hat{x}_A)) dx_R = \sqrt{\det(2\pi\Psi)} \quad (28)$$

so that (24) becomes:

$$\int_{x_R} p(x_A = x_R | x_R) p(x_R) dx_R = \eta \sqrt{\det(2\pi\Psi)} \exp(-L_2(\hat{x}_R, \hat{x}_A)). \quad (29)$$

Recall that $L_2(\hat{x}_R, \hat{x}_A) = L(x_R, \hat{x}_R, \hat{x}_A) - L_1(x_R, \hat{x}_R, \hat{x}_A)$, which is quadratic in \hat{x}_R and \hat{x}_A . A similar strategy is used to show that (29) is in the form of a normal distribution, $N_{\hat{x}_R}(m, \Phi)$, where $m = \hat{x}_A$ and $\Phi = \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T$. Since $\eta \sqrt{\det(2\pi\Psi)} = \frac{1}{\sqrt{\det(2\pi\Phi)}}$, and $L_2(\hat{x}_R, \hat{x}_A) = \frac{1}{2}(\hat{x}_R - m)^T \Phi^{-1}(\hat{x}_R - m)$, the result is obtained. ■

Again, the objective is to convert the constraint $P(C) \leq \delta$ into a constraint on the robot mean state in terms of the obstacle mean state, resulting in (18), with the exception that $\Sigma_C \triangleq \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T$.

IV. APPROXIMATION VALIDATION

The approximation introduced in Section III-A assumes a constant probability of collision over the volume occupied by the robot. This approximation is only valid for a small robot, and it is of interest to evaluate the validity of this approximation as the size of the robot is increased. A scheme to evaluate the true probability of collision is presented before a comparison with the approximation is performed by investigating the effect of the relative size of the robot and the spread of the distribution, as well as the shape of the distribution on the accuracy of the approximation.

A. True Probability of Collision Evaluation

Consider the integral in eq. (6). By introducing the augmented state, $x \triangleq [x_R \ x_A]^T$, the equation is rewritten:

$$P(C) = \int_x I_C(x) p(x) dx \quad (30)$$

Using the Central Limit Theorem, this integral can be approximated by drawing N *iid* samples from the joint distribution $p(x_R, x_A)$ and then evaluating the function $I_C(x_R, x_A)$:

$$P(C) \approx \frac{1}{N} \sum_{i=1}^N I_C(\tilde{x}^i) \quad (31)$$

where \tilde{x}^i is the sample drawn from $p(x_R, x_A)$. This approach is known as Monte-Carlo Simulation (MCS). By writing the integral in terms of the joint distribution and augmented state, a generalization

of the algorithm introduced by Lambert et al. [12] to dependent distributions is obtained. By assuming independent distributions, Lambert et al. formulate the collision probability in terms of the individual distributions. Using this formulation, a naive MCS approach with a double summation is obtained⁶. In contrast, by formulating the probability of collision in terms of the joint distribution between the robot and the obstacle position states, dependent distributions can be handled and the resulting algorithm requires evaluation of a single summation. The accuracy of the estimate of the true probability of collision as a function of number of samples is investigated in simulation in [12]. For small probabilities of collision, it is concluded that 1000 samples are sufficient. A more formal analysis is presented here, based on the expected accuracy of the estimate in terms of the coefficient of variation. The coefficient of variation is a normalized measure of the dispersion. For the *estimate* of the probability of collision, and noting that $I_C^2(x) = I_C(x)$, it is defined as:

$$\delta \triangleq \frac{\sigma}{\sqrt{N}\mu} \quad (32)$$

where

$$\mu = \int_x I_C(x) p(x) dx \quad (33)$$

and

$$\begin{aligned} \sigma &= \int_x I_C^2(x) p(x) dx - \mu^2 \\ &= \mu(1 - \mu). \end{aligned} \quad (34)$$

The coefficient of variation simplifies to:

$$\delta = \sqrt{\frac{1 - \mu}{N\mu}} \quad (36)$$

Assuming that a coefficient of variation of 0.1 (i.e., the standard deviation of the estimate is 10% of the mean) and a target probability of collision of 1% ($\mu = 0.01$), then at least $N = 9900$ samples are required. For a C++ implementation⁷ of the MCS algorithm and using $N = 10000$ samples, the average runtime is 4.56 milli seconds, which is on the same order as presented by Lambert et al. [12].

B. Validity of Approximation

When considering the validity of the approximation presented in Section III-A, the *shape* and *spread* of the distribution (which is governed by the covariance) and the size of the object are relevant. The approximation assumes a constant valued joint probability density function over the space occupied by the robot, multiplied by the volume of the object. If the object is large relative to the spread of the distribution, then the approximation will be inaccurate. As a result, the effects of the spread and robot size are connected. Two cases are considered in 2D state space: the covariance is fixed (with covariances resulting in circular and oblong uncertainty ellipsoids, respectively), and the size of the robot is varied.

1) *Effect of Relative Scaling*: The covariance for both the robot and obstacle are fixed at $\Sigma = \text{diag}(1, 1)$. The radius of the disk robot, r , is varied from 0.3 to 1 m. The robot position is fixed at the origin, and the obstacle position is varied over the space $x \in [-5, 5]$ and $y \in [-5, 5]$. The true (eq. (6)) and approximate (eq. (17)) probabilities of collision are calculated (e.g., the values of the calculated probabilities of collision are plotted in Figure 2 for a robot of radius 0.4m).

⁶In an attempt to increase computational efficiency, the algorithm is modified to use a single summation without justification.

⁷2.66 GHz Intel Core 2 Duo processor, 4GB RAM, and using GSL for random number generation

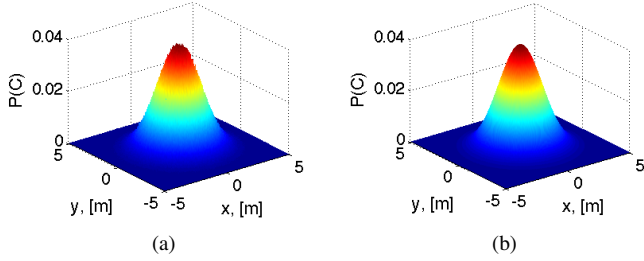


Fig. 2. Values of the probability of collision at different obstacle locations for a robot of radius 0.4m as calculated from the true (a) and approximated (b) results.

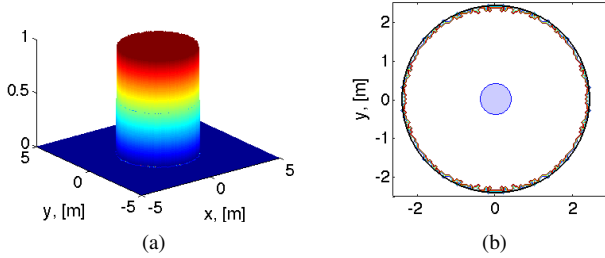


Fig. 3. The collision chance constraint is evaluated for a robot of radius 0.4 m, using the true probability of collision (a) and the scaling of an equivalent elliptical constraint, κ_t , is determined, as illustrated in (b), together with a contour plot of the data in (a).

TABLE I
ELLIPSOIDAL SCALINGS TO ENFORCE COLLISION CHANCE CONSTRAINTS FOR VARYING ROBOT SIZES.

Radius	κ_a	κ_t	$\frac{\sqrt{\det(\Sigma_C)}}{\pi r^2}$	Area error
0.3	1.62	1.67	7.07	3.0%
0.4	2.77	2.92	3.98	5.1%
0.5	3.67	3.87	2.55	5.2%
0.6	4.39	4.39	1.77	6.4%
0.7	5.01	5.41	1.30	7.4%
0.8	5.55	6.05	0.99	8.3%
1.0	6.44	7.14	0.64	9.8%

In order to compare the validity of the approximation, the chance constraint is evaluated ($P(C) > \delta$ for $\delta = 0.01$). From eq. (18), an ellipsoidal constraint is expected where the size of the ellipsoid scales with $\sqrt{\det(\Sigma_C)}/\pi r^2$. By plotting the collision chance constraint, as evaluated from the true probability of collision, the scaling, κ_t , of the ellipsoid required to enforce the true constraint can be obtained (see Figure 3). These values are compared to the approximation value, κ_a , in Table I.

The threshold for the validity of the approximation is hard to quantify. The areas of the resulting ellipses from the true and approximate results are compared. For $r = 0.7m$, the error is 7.4%, which is deemed as the maximum allowable (from visual inspection, Figure 4). For larger robot radii, the approximation is not accurate and it is concluded that the approximation is valid for $\sqrt{\det(\Sigma_C)}/\pi r^2 > 1.3$.

2) *Effect of Distribution Shape*: In the previous section it was determined that the approximation is valid for $\sqrt{\det(\Sigma_C)}/\pi r^2 > 1.3$. However, the shape of the distribution is expected to affect the accuracy of the approximation since the size of the robot relative to the major/minor axes of the resulting uncertainty ellipse is different. To evaluate the effect of shape, the covariance matrices of the robot and obstacle are fixed at $\Sigma = \text{diag}(1, 0.1)$. The size of the robot is chosen to yield $\sqrt{\det(\Sigma_C)}/\pi r^2 \approx 1.3$, giving $r = 0.4m$. From Figure 5 it is concluded that the effect of the shape is small for the

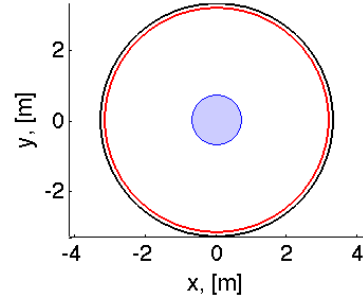


Fig. 4. Graphical comparison of the elliptical constraints obtained from the true (black) and approximate (red) probabilities of collision for a robot of radius 0.7m.

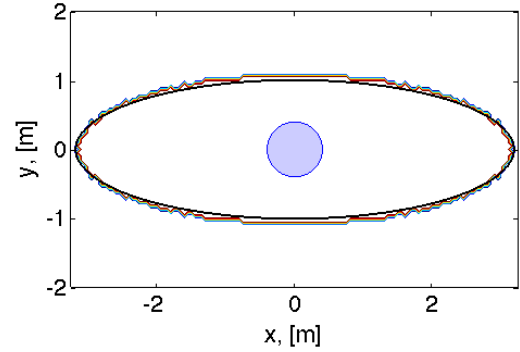


Fig. 5. The collision chance constraint for a robot of radius 0.4 m is plotted for the true probability of collision (contour plot).

region where the approximation is valid. For larger robot sizes, the effect of the shape is more pronounced.

V. APPLICATION TO ROBOT MOTION PLANNING

The results presented in this paper can be used to evaluate collision chance constraints as they arise in the robot motion planning problem in dynamic, uncertain environments [9], [10]. This type of motion planning problem can be posed in a receding horizon (RH) framework where an approximate problem is solved over a finite planning horizon and a small part of the plan is executed before new information is incorporated and the problem is re-solved. In this RH framework, the above results can be used to model the constraints posed by dynamic and static obstacles.

One requirement of the RH approach is real-time solution of the approximate planning problem. In Section IV-A it was noted that the MCS implementation can evaluate constraints in a few milliseconds. However, the general optimal motion planning problem formulation results in a non-convex optimization problem [9]. These problems are notoriously hard to solve efficiently, regardless of the constraint checking implementation. General approaches to non-convex optimization problems can require thousands of constraint evaluations per planning stage. As a result, the MCS implementation of probabilistic collision checking (as presented in Section IV-A) can be insufficient for real-time implementation *when combined with a naive non-convex optimization algorithm*. This motivates the need for approximate probabilistic collision constraint evaluation.

The approximation presented in this work offers a viable method for efficient constraints enforcement, noting that small errors arising from approximations are compensated by the information feedback when the problem is re-solved in the RH framework. The range

of covariances where the approximation is valid can be extended in certain applications (e.g., the motion planning problem with disk objects and random walk models [9], [10]). This extension is possible when the dynamic- and measurement models for the robot and dynamic obstacles do not alter the shape of the uncertainty as it is propagated through the system. The functional form of the constraints, as obtained from the approximation, can be applied directly by adjusting the scaling factor to account for the non-point geometry. A look-up table of scaling factors can be constructed offline for different covariances by recording the required scaling from the true probability of collision, κ_t , as obtained from MCS in Section IV-A. By using the look-up table, the constraints can be evaluated very efficiently. Additionally, since the constraints are based on the true probability of collision, the enforced constraints lack the conservatism that plagued previous approaches.

VI. CONCLUSIONS

Probabilistic collision checking is a basic requirement for a robot operating in an a priori unknown environment. Standard approaches ignore uncertainty entirely, or account only for robot state uncertainty. In this work, the state uncertainties of both the robot and object are accounted for. Probabilistic collision checking is formulated for robots and obstacles of arbitrary geometry with arbitrary probability distributions for the robot and obstacle position states. Evaluation of the resulting formula is cumbersome, but intuition is gained when using a ‘small objects’ approximation. Closed-form solutions are obtained for this approximate problem for the cases where the robot and obstacle states are described by independent and dependent Gaussian-distributed variables, respectively. The validity of the approximation is analyzed in terms of the relative scale of object size and distribution spread, and the shape of the distribution. The result has been applied to the robot motion planning problem in [9], [10]).

REFERENCES

- [1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. MIT Press, 2007.
- [2] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Press, 1991.
- [3] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [4] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *Int. J. Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [5] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [6] T. Fraichard and H. Asama, “Inevitable collision states - a step towards safer robots?” *Advanced Robotics*, vol. 18, no. 10, p. 1001, 2004.
- [7] A. Bautin, L. Martinez-Gomez, and T. Fraichard, “Inevitable collision states: A probabilistic perspective,” in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2010, pp. 4022–4027.
- [8] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, “Probabilistic collision state checker for crowded environments,” in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2010, pp. 1492–1498.
- [9] N. E. Du Toit, “Robot motion planning in dynamic, cluttered, uncertain environments,” Ph.D. dissertation, Dept. of Mechanical Engineering, California Institute of Technology, 2010.
- [10] N. E. Du Toit and J. W. Burdick, “Robotic motion planning in dynamic, cluttered, uncertain environments,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2010, pp. 966–973.
- [11] L. Blackmore, H. Li, and B. Williams, “A probabilistic approach to optimal robust path planning with obstacles,” in *Proc. American Control Conf.*, 2006.
- [12] A. Lambert, D. Gruyer, and G. St. Pierre, “A fast monte carlo algorithm for collision probability estimation,” in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, Dec 2008, pp. 406–411.
- [13] L. Blackmore, “Robust path planning and feedback design under stochastic uncertainty,” in *Proc. AIAA Guidance, Navigation and Control Conf.*, no. AIAA-2008-6304, 2008.
- [14] P. Li, M. Wendt, and G. Wozny, “A probabilistically constrained model predictive next term controller,” *Automatica*, vol. 38, no. 7, pp. 1171–1176, 2002.
- [15] A. T. Schwarm and M. Nikolaou, “Chance-constrained model predictive control,” *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999.
- [16] D. van Hessem and O. Bosgra, “Towards a separation principle in closed-loop predictive control,” *Proc. American Control Conf.*, vol. 5, pp. 4299–4304, 2003.
- [17] J. Yan and R. R. Bitmead, “Incorporating state estimation into model predictive control and its application to network traffic control,” *Automatica*, vol. 41, no. 4, pp. 595–604, 2005.
- [18] L. Blackmore and B. C. Williams, “Optimal, robust predictive control of nonlinear systems under probabilistic uncertainty using particles,” in *Proc. American Control Conf.*, 2007, pp. 1759–1761.
- [19] D. van Hessem, “Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation,” Ph.D. dissertation, Delft University, 2004.
- [20] L. Blackmore and M. Ono, “Convex chance constrained predictive control without sampling,” in *Proc. AIAA Guidance, Navigation and Control Conf.*, 2009.
- [21] M. Ono and B. Williams, “An efficient motion planning algorithm for stochastic dynamic systems with constraints on the probability of failure,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2008, pp. 1376–1382.
- [22] K. B. Petersen and M. S. Pedersen, “Matrix cookbook,” www.matrixcookbook.com, November 2008.
- [23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.