Exact and Bounded Collision Probability for Motion Planning under Gaussian Uncertainty

Antony Thomas, Fulvio Mastrogiovanni and Marco Baglietto

Abstract—Computing collision-free trajectories is of prime importance for safe navigation. We present an approach for computing the collision probability under Gaussian distributed motion and sensing uncertainty with the robot and static obstacle shapes approximated as ellipsoids. The collision condition is formulated as the distance between ellipsoids and unlike previous approaches we provide a method for computing the exact collision probability. Furthermore, we provide a tight upper bound that can be computed much faster during online planning. Comparison to other state-of-the-art methods is also provided. The proposed method is evaluated in simulation under varying configuration and number of obstacles.

I. INTRODUCTION

Safe and reliable navigation of robots is of vital importance while operating in real-world environments. To this end, it is required to foresee collisions to compute collision free trajectories online. However, the robot state estimates are uncertain due to motion and sensing errors. Similarly, environmental errors due to imperfect sensing, partial knowledge and estimation errors render obstacle location uncertain. As a result, rather than resorting to deterministic approaches that avoid collisions, collision avoidance algorithms need to incorporate such estimation uncertainties and compute the probability for encountering collisions.

Similarly to other existing approaches [1]–[6], in this work we model the uncertainties using Gaussian distributions. The robot and obstacle locations are thus parametrized as Gaussian probability distribution functions (pdfs). Given the robot and obstacle location, the probability of collision can be computed by marginalizing the joint distribution of the robot and the obstacle positions over the set of all robot and obstacle locations that lead to collisions. For example, if we assume the robot and the obstacle to be spheres, a collision occurs whenever the magnitude of the difference in robot and obstacle location is less than the sum of the radius of the spheres. The integral is thus marginalized over this collision constraint. However, there is no closed form solution to this marginalization. Most existing approaches [1], [3], [5], [7] therefore compute an approximation of the integral. These approximations can be overly conservative as the computed collision probabilities tend to be loose upper bounds. To circumvent the issue of non existence of a closed form solution, in this paper we show that the probability of collision is equivalent to the cumulative distribution function (cdf) of a quadratic form in random variables of the difference in robot and obstacle locations. We further compute a tighter bound for the probability of a quadratic form to be below a user-specified threshold.

Contributions: In this paper, we present a novel approach to compute the exact collision probability when the uncertainties are modeled using Gaussian distributions. We formulate the collision constraint as the distance between L"owner-John ellipsoids (Section IV) of the robot and the obstacles. This results in a quadratic form in random variables of the difference in robot and obstacle locations and we show that the required collision probability is the cdf of the quadratic form. An exact expression for the collision probability is thus obtained as a converging infinite series for the cdf. Furthermore, we derive a tighter upper bound with respect to previous works for fast approximation of the collision probability for 3D motion planning.

II. RELATED WORK

Uncertain environments are such that they often preclude the existence of guaranteed collision free trajectories [8]. Therefore, one can only provide probabilistic formal guarantees for safe navigation. Bounding volume approaches enlarge the robot and obstacle shapes, most often by their 3-σ uncertainty ellipsoids or spheres. In [9], the uncertainty bound is used to enlarge the robot into a sphere. The robot shape is enlarged with the 3-σ uncertainty ellipsoids in [10]. Exact collision checking with enlarged bounding volumes is used in [11]. Sigma hulls are used to compute the signed distance to the obstacles in [12]. Rectangular bounding box for both the robot and the obstacle are used to obtain an upper bound in [13]. Patil et al. [2] truncate the Gaussian robot state distributions [14], propagating collision free samples. Truncated Gaussian distributions is also employed in [15] to compute risk-aware and asymptotically optimal trajectories.

Several approaches compute the collision probability by marginalizing the joint distribution between the robot and obstacle locations. The marginalization is computed over the set of robot and obstacle locations that satisfy the collision constraint. However, there is no closed form solution to such a formulation and Monte Carlo (MC) techniques are employed [16]. In [7], the resulting double summation from the MC integration is approximated to a single summation. Another related MC approach and is presented in [17]—solves a deterministic motion planning problem with inflated obstacles, and then refines the inflation to compute a path that is exactly as safe as desired. However, MC approaches tend to be computationally expensive and hard to model within an optimization framework. Assuming that the robot size is
negligible, the marginalization can be approximated to obtain an expression in terms of the joint distribution multiplied by the volume occupied by the robot. Du Toit and Burdick [1] compute the joint density at the robot center, whereas in [3] the maximum value (upper bound) is computed by evaluating the density at the surface of the robot.

Chance-constrained approaches find optimal sequence of control inputs subject to the constraint that the collision probability must be below a user-specified threshold [18]. Zhu et al. compute an approximate upper bound linearizing the collision condition. They further employ chance constraints to compute bounded collision-free trajectories with dynamic obstacles. An upper bound is computed in [19] by employing Gaussian chance-constraints. They use Newton’s method to compute the point on the boundary of the obstacle that is closest to the robot configuration. In [8], a Gaussian Process (GP) based technique is employed to learn motion patterns (a mapping from states to trajectory derivatives) to predict possible future obstacles trajectories. First-exit times for Brownian motions are extended to continuous nonlinear dynamics to compute collision probabilities in [20]. Axelrod et al. [4] focus exclusively on obstacle uncertainty and formalize a notion of shadows, which are the geometric equivalent of confidence intervals for uncertain obstacles. The shadows fundamentally give rise to loose bounds but are used as polytopes with Gaussian-distributed faces in [21]. Planning a collision-free path in the presence of risk zones is considered in [22] by penalizing the time spent in these zones. Jasour et al. [23] employ risk contours map that takes into account the risk information (in uncertainties in location, size and geometry of obstacles) to obtain safe paths with bounded risks. A related approach for randomly moving obstacles is introduced in [24]. Collision avoidance in the context of human-robot interaction are presented in [25], [26]. Formal verification methods have also been used to construct safe plans [27], [28]. On the one hand, existing approximations tend to overestimate collision probabilities 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.

### III. Preliminaries

Throughout this paper vectors will be assumed to be column vectors and will be denoted by bold lower case letters, $\mathbf{x}$ and its components will be denoted by lower case letters. Transpose of $\mathbf{x}$ will be denoted by $\mathbf{x}^T$ and its Euclidean norm by $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$. The mean of a random vector, $\mathbb{E}(\mathbf{x})$ will be denoted by $\mathbf{\mu}$, the corresponding covariance, $\text{Cov}(\mathbf{x})$ by $\Sigma$. Matrices will be denoted by capital letters, $\mathbf{M}$, with its trace denoted by $\text{tr}(\mathbf{M})$. By $\text{vec}(\mathbf{M})$ we will denote the vector formed used the columns of $\mathbf{M}$. The identity matrix will be denoted by $\mathbf{I}$ or $\mathbf{I}_n$ (if dimension needs to be stressed). A diagonal matrix with diagonal elements $\lambda_1, \ldots, \lambda_n$ will be denoted by $\text{diag}(\lambda_1, \ldots, \lambda_n)$. Sets will be denoted using mathcal fonts, $\mathcal{S}$. Unless otherwise mentioned, subscripts on vectors/matrices will be used to denote time indexes. The notation $P(\cdot)$ will be used to denote the probability of an event and the pdf by $p(\cdot)$.

**Definition 1.** Given a positive definite $n \times n$ matrix $\mathbf{A}$ and a vector $\mathbf{a} \in \mathbb{R}^n$, an $n$-dimensional ellipsoid $\mathcal{E}_n(\mathbf{A}, \mathbf{a})$ is defined as

$$\mathcal{E}_n(\mathbf{A}, \mathbf{a}) = \{ \mathbf{x} \in \mathbb{R}^n | (\mathbf{x} - \mathbf{a})^T \mathbf{A} (\mathbf{x} - \mathbf{a}) \leq 1 \}$$

where the superscript $n$ will be avoided when there is no cause for confusion. At any time $k$, we denote the robot state by $\mathbf{x}_k$, the acquired measurement from objects is denoted by $\mathbf{z}_k$ and the applied control action is denoted as $\mathbf{u}_k$. By objects we refer to the landmarks and the obstacles in the environment. We make the following assumptions: (1) the uncertainties are modeled using Gaussian distributions, (2) the robot and obstacle geometry are known to us and are assumed to be non-deformable objects, (3) the collision probability at each time-step is treated to be independent with respect to the previous time-step. To describe the dynamics of the robot, we consider a standard motion model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + n_k \quad n_k \sim \mathcal{N}(0, \mathbf{R}_k)$$

where $n_k$ is the random unobservable noise, modeled as a zero mean Gaussian. Objects are detected through the robot’s sensors and assuming known data association, the observation model can be written as

$$\mathbf{z}_k = h(\mathbf{x}_k) + v_k \quad v_k \sim \mathcal{N}(0, \mathbf{Q}_k)$$

### IV. Collision Probability

For computing collision-free paths it is imperative that the distance between the robot and its environment is known. The geometry of robots and other objects in the environment can be expressed using polyhedrons or a combination of polyhedrons. It is known that polyhedrons can be approximated using suitable ellipsoids. In particular, in [29] it was shown that for every convex polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$, there exits a unique ellipsoid $\mathcal{E}$ of minimal volume that contains $\mathcal{P}$, called the Löwner-John ellipsoid $\mathcal{J}(\mathcal{P})$. Thus, there exists a unique minimum volume ellipsoid that encloses every $\mathcal{P}$ and it is sufficient to compute the distance between the Löwner-John ellipsoids of the robot and the obstacle, respectively. A convex optimization approach for computing Löwner-John ellipsoids is presented in [30]. In this paper we assume that these ellipsoids are known to us. For an ellipsoid $\mathcal{E}(\mathbf{A}, \mathbf{a})$, $\mathbf{a}$ is the center of the ellipsoid and $\mathbf{A}$ determines its geometry. Thus our assumption of known ellipsoids mean that both these quantities are known at the initial time instant for robot and obstacles. Typically, $\mathbf{a}$ is the center of the robot or the objects that the ellipsoid encloses. It is noteworthy that while planning for future control commands, the robot state (location and orientation) are often estimated using a motion model and by simulating possible future observations. As a result, the estimates for $\mathbf{a}$ as well as the orientation are computed at each planning instance. Since we assume Gaussian distributed pdfs for state estimation, $\mathbf{a}$ is typically taken to be mean of the location estimate and $\mathbf{A}$ is obtained
from the orientation estimate using an appropriate rotation matrix. In this paper we will only consider static obstacles and their L"owner-John ellipsoids remain constant throughout.

A. Distance Between Two Ellipsoids

For two ellipsoids $E_1, E_2$, we begin by computing a point $x^* \in E_2$ such that the ellipsoid level surfaces surrounding $E_1$ first touch $E_2$ at $x^*$. The result is stated below without proof (see [30] for proof).

**Lemma 1.** Given two ellipsoids $E_1 = E^n(B, b)$ and $E_2 = E^n(C, c)$, the point $x^* \in E_2$ at which the ellipsoid level surfaces surrounding $E_1$ first touch the ellipsoid $E_2$ is given by

$$x^* = b + \lambda_0(M')B^{-1/2} \left[ \lambda_0(M')I - \hat{C} \right]^{-1} \hat{c}$$

where $\lambda_0(M')$ is the minimal eigenvalue of a $2n \times 2n$ matrix $M'$ given by

$$M' = \begin{bmatrix} \hat{C} & -I \\ -\hat{c}\hat{c}^T & \hat{C} \end{bmatrix}$$

with $\hat{C} = \bar{C}^{-1}$ and $\hat{c} = \bar{C}^{-1/2}\bar{c}$, where $\bar{C} = B^{-1/2}C B^{-1/2}$ and $\bar{c} = B^{1/2}(c - b)$.

For two ellipsoids $E_1 = E^n(B, b)$ and $E_2 = E^n(C, c)$, a collision between them occurs if $E_1 \cap E_2 \neq \{\emptyset\}$. We will denote this collision condition by

$$C_{b, c} = \{E_1, E_2 | E_1 \cap E_2 \neq \{\emptyset\} \}$$

Using Lemma 1 for two ellipsoids $E^n(B, b)$, $E^n(C, c)$, we can compute the point $x^* \in E_2$ at which the ellipsoid level surfaces surrounding $E_1$ first touch $E_2$. Now, suppose that the two ellipsoids touch or intersect each other, then $x^*$ satisfies the equation of $E_1$. Thus we have

$$(x^* - b)^T B(x^* - b) \leq 1$$

Substituting for the value of $x^*$ from Lemma 1 expanding and rearranging, it follows that

$$y^T D^T BDy \leq 1/\lambda_0^2(M')$$

where $y = c - b$ and $D = B^{-1/2}(\lambda_0(M')I - \hat{C})^{-1}B^{1/2}$ and $M'$ is a $2n \times 2n$ matrix as defined in [5].

B. Collision Condition

We first define a quadratic form in random variables, and later show that the collision condition can be written in terms of the quadratic form of the difference in robot and obstacle locations.

**Definition 2.** For a random vector $x = (x_1, \ldots, x_n)^T$, the quadratic form in the random variables $x_1, \ldots, x_n$ associated with an $n \times n$ symmetric matrix $A = (a_{ij})$ is

$$Q(x, A) = Q(x_1, \ldots, x_n) = x^T Ax = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

Let us now consider a random vector $x$ with mean value $\mu$ and a positive definite covariance matrix $\Sigma > 0$. Then for $y = \Sigma^{-1/2}x$, where $\Sigma^{1/2}$ is the symmetric square root, we have $E(y) = \Sigma^{-1/2}\mu$ and $Cov(y) = I$. Let us define a random vector $z = (y - \Sigma^{-1/2}\mu)$ such that $E(z) = 0$ and $Cov(z) = I$. Then

$$Q(x, A) = x^T Ax = y^T \Sigma^{1/2} A \Sigma^{1/2} y$$

$$= (z + \Sigma^{-1/2}\mu)^T \Sigma^{1/2} A \Sigma^{1/2} (z + \Sigma^{-1/2}\mu)$$

Let $P$ be an orthogonal matrix, that is, $PP^T = I$ which diagonalizes $\Sigma^{1/2} A \Sigma^{1/2}$, then $PP^T \Sigma^{1/2} A \Sigma^{1/2} P = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of $\Sigma^{1/2} A \Sigma^{1/2}$. The quadratic form can now be written as

$$Q(x, A) = (z + \Sigma^{-1/2}\mu)^T \Sigma^{1/2} A \Sigma^{1/2} (z + \Sigma^{-1/2}\mu)$$

$$= (u + b)^T \text{diag}(\lambda_1, \ldots, \lambda_n) (u + b)$$

where $u = P^T z = (u_1, \ldots, u_n)^T$ and $b = P^T \Sigma^{-1/2}\mu = (b_1, \ldots, b_n)^T$. The expression in (11) can be written concisely,

$$Q(x) = x^T Ax = \sum_{i=1}^{n} \lambda_i (u_i + b_i)^2$$

We recall from [5] that the collision condition is given by

$$y^T D^T BDy \leq 1/\lambda_0^2(M')$$

where $A = D^T BD$. In motion planning, the collisions correspond to robot-obstacle or robot-robot collisions. While planning under motion and sensing uncertainty, the robot and obstacle states can only be estimated in probabilistic terms. This renders $b, c$ as random vectors. As discussed before, in this paper we assume them to have Gaussian pdfs. As a result, the difference vector $y = c - b$ is also a Gaussian random vector. Since $B$ and $C$ are positive definite, the matrix $A$ is symmetric and therefore $y^T Ay$ is in a quadratic form. Thus the collision probability can be written as

$$P \left( y^T Ay \leq 1/\lambda_0^2(M') \right)$$

Let $v = y^T Ay$, then

$$P \left( v \leq 1/\lambda_0^2(M') \right) = F_v \left( 1/\lambda_0^2(M') \right)$$

where $F_v$ is the cdf of $v$. For convenience, we state the theorem that computes an exact expression of $F_v$ for quadratic forms. The proof may be found in [31].

**Theorem 1.** The cdf of $Q(y, A) = v = y^T Ay$ with $A = A^T > 0, y \sim N(\mu, \Sigma), \Sigma > 0$ is

$$F_v(v) = P(v \leq v) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{v^{\frac{\nu}{2} + k}}{\Gamma\left(\frac{\nu}{2} + k + 1\right)}$$

and its pdf is given by

$$p_v(v) = P(v = v) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{v^{\frac{\nu}{2} + k - 1}}{\Gamma\left(\frac{\nu}{2} + k\right)}$$

where $\Gamma$ denotes the gamma function and $c_0 = \exp(-\frac{n}{2} \sum_{i=1}^{n} b_i^2) \prod_{i=1}^{n} (2\lambda_i)^{-\frac{\nu}{2}}$, $c_k = \frac{1}{k} \sum_{i=0}^{k-1} d_{k-i} c_i$ and $d_k = \frac{\nu}{2} + k - 1$.
Let the cdf of the chi-square distribution by table look-up. We provide the theorem without proof
experience, suitable convergence is often obtained within the
validated in Theorem 1 gives the exact value of collision
probability. Given the current robot state $x$, the quadratic form is
bounded by $\lambda_{\text{max}}$. Note that in our case $x$ is a random
vector which is obtained from the difference between robot and obstacle locations. Since the environment in
which the robot operates is bounded, without any loss of
generality we can assume that $x$ lies within a sphere of
certain radius $\mathcal{S}$. Thus, $\beta = \max(x^T A x) = \lambda_{\text{max}} \epsilon^2$.
Since $x^T A x$ is a real number, we may thus write $x \in [\mathbb{E}(x^T A x) - \eta \sqrt{\text{Var}(x^T A x)}], [\mathbb{E}(x^T A x) + \eta \sqrt{\text{Var}(x^T A x)}]$, where $\eta \in \mathbb{R}^+$. Based on several experimental analysis, $\eta = 1$
provides a tighter bound and works well in practice in all
our scenarios. We thus use $\beta = \mathbb{E}(x^T A x) + \sqrt{\text{Var}(x^T A x)}$.

**Definition 3.** A robot configuration $x_k$ is an $\epsilon$-safe config-
uration with respect to an obstacle configuration $s_k$, if the
probability of collision is such that $P(C_{x_k, s_k} \leq \epsilon)$

Let us consider the case where the ellipsoids enclose a robot currently at location $x_k$ and an obstacle at location $s_k$, respectively. Since the position of robots and obstacles are Gaussian distributed random variables, there exists a nonzero probability for both the robot and the obstacle to be found anywhere within the environment. Therefore, trajectories are computed such that the probability of collision with the obstacles is less than a specified bound. We thus look for robot positions $x_k$ such that the probability of collision is at most $\epsilon$, that is, $P(C_{x_k, s_k} \leq \epsilon)$. From (14), we have

$$P\left(\mathbb{E}(y^T A y) \leq 1/\lambda_0^2(M')\right) \leq \epsilon \quad (22)$$

Using Lemma 3 the bounded collision constraint is obtained

$$\beta - \mathbb{E}(y^T A y) \leq \epsilon \left(\beta - 1/\lambda_0^2(M')\right) \quad (23)$$

**D. Comparison to Other Methods**

We provide a comparison with several state-of-the-art methods using a robot and a close-by obstacle. The robot
mean is located at $(0.95, 0.95, 0)$ with its Löwner-John ellipsoid semi-principle axes $(0.18, 0.18, 0.22)$ m and
covariance $\text{diag}(0.41, 0.41, 0.21)$ m$^2$. The obstacle is located at the origin with semi-principle axes $(0.6, 0.6, 1.2)$ m. The
collision probability values can be seen in Table 1. Most
approaches approximate an integral of the joint distribution
between the robot and the obstacle to compute the collision
probability. Given the current robot state $x_k$ and the obstacle
state $s_k$, the collision probability is given by

$$P(C_{x_k, s_k}) = \int_{x_k} \int_{s_k} I_c(x_k, s_k) p(x_k, s_k) \quad (24)$$

where $I_c$ is an indicator function defined as

$$I_c(x_k, s_k) = \begin{cases} 1 & \text{if } \mathcal{R} \cap \mathcal{S} \neq \{\phi\} \\ 0 & \text{otherwise.} \end{cases} \quad (25)$$

and $p(x_k, s_k)$ is the joint distribution of the robot and the obstacle. The numeric integral of (24) gives the exact value
and we compute the same to validate our approach. The

This means truncating $x$ at the boundary of the workspace. However, we assume the workspace to be large enough so that the truncated $x$ is approximately Gaussian distributed.
Table I: Comparison of collision probability methods

| Methods                        | Collision probability | Computation time (s) | Feasible |
|--------------------------------|-----------------------|----------------------|----------|
| Numerical integral            | 0.0568                | 4.3619 ± 1.184       | Yes      |
| Approximate numerical integral [1] | 0.1577                | 1.9002 ± 1.1607      | No       |
| Bounding volume [10, 11]       | 0.0013                | 0.0000               | No       |
| Center point approximation [1] | 0.1027                | 0.0004 ± 0.0001      | No       |
| Maximum probability approximation [2] | 0.1198                | 0.0000 ± 0.0006      | No       |
| Unbiased constraint [3]        | 0.1894                | 0.0001 ± 0.0000      | No       |
| Rectangular white noise sequence with | 0.1582                | 0.0006 ± 0.0006      | No       |
| Our approach– exact           | 0.0672                | 0.0044 ± 0.0002      | Yes      |
| Our approach– upper bound     | 0.0680                | 0.0009 ± 0.0003      | Yes      |

Theorem 3. The innovations process is a zero-mean Gaussian white noise sequence with

\[
\mathbb{E} \left[ \left( z_k - h(\mu_k) \right) \left[ z_k - h(\mu_k) \right]^T \right] = H_k \Sigma_k H_k^T + Q_k
\]

Proof. The proof may be found in [37], page 234. □

From Theorem [3] the stochastic belief state dynamics can be written as

\[
b_k(x_{k-1}, u_{k-1}) = g \left( b(x_{k-1}, u_{k-1}) + W b(x_{k-1}, u_{k-1}) w_{k-1} \right)
\]

where

\[
g \left( b(x_{k-1}, u_{k-1}) \right) = \begin{bmatrix} \mu_k \\ s_k \end{bmatrix} = \begin{bmatrix} f(\mu_{k-1}, u_{k-1}) \\ vec \left( (I - K_k H_k) \Sigma_k \right) \end{bmatrix}
\]

\[
W b(x_{k-1}, u_{k-1}) = \begin{bmatrix} K_k \\ 0 \end{bmatrix}
\]

Thus the innovation term \( K_k (z_k - h(\bar{\mu}_k)) \) is distributed according to

\[
N(0, K_k H_k \Sigma_k H_k^T + Q_k)
\]

The maximum likelihood observation at time \( k \) is \( z_k = h(\bar{\mu}_k) \), which reduces the innovation term in (27) to zero and thus eliminating the stochasticity from the belief state dynamics. The assumption of maximum likelihood observation was first relaxed in [34]. However, a first-order approximation is assumed rendering the innovation term to be distributed according to \( N(0, K_k H_k \Sigma_k) \). Other approaches that relax this assumption either simulate future measurements or treat them as random variables [38–40].
The expectation is discarded from the first term as it does not depend on the future observations. Let us now proceed by evaluating the term with expectation. Using (37), we have

\[ \|x_{k+L} - x^g\|^2_{M_y} = \|\bar{\mu}_{k+L} + K_{k+L} (z_{k+L} - h(\bar{\mu}_k)) - x^g\|^2_{M_y} \]

\[ = \|\bar{\mu}_{k+L} - x^g + K_{k+L} (z_{k+L} - h(\bar{\mu}_k))\|^2_{M_y} \]

\[ = \|\bar{\mu}_{k+L} - x^g\|^2_{M_y} + \|K_{k+L} (z_{k+L} - h(\bar{\mu}_k))\|^2_{M_y} \]

\[ + (\bar{\mu}_{k+L} - x^g)^T M_y (K_{k+L} (z_{k+L} - h(\bar{\mu}_k))) \]

\[ + \left( K_{k+L} (z_{k+L} - h(\bar{\mu}_k)) \right)^T M_y (\bar{\mu}_{k+L} - x^g) \] (38)

For any random vector \( y \) and a matrix \( A \) of appropriate dimension, we have \( \mathbb{E} [y^T A y] = tr \left( A \mathbb{E} [y] \mathbb{E} [y]^T \right) \). Exploiting this and the fact that \( \mathbb{E} \left[ (z_{k+L} - h(\bar{\mu}_k)) \right] = 0 \), the expression in (38) simplifies to

\[ \mathbb{E} \left[ \|x_{k+L} - x^g\|^2_{M_y} \right] = \|\bar{\mu}_{k+L} - x^g\|^2_{M_y} + tr \left( K_{k+L} \mathbb{E} \left[ (z_{k+L} - h(\bar{\mu}_k)) \right] \right) \] (39)

where the expression for the variance is obtained from (29).

The optimization problem can be formally stated now as

\[ \min_{\beta} \mathbb{E} \left[ \|x_{k+L} - x^g\|^2_{M_y} \right] \]

s.t.

\[ b(x_{k+l}) = g(b(x_{k+l-1}), u_{k+l-1}) \]

\[ + W(b(x_{k+l-1}), u_{k+l-1})w_{k+l-1} \]

\[ u_{k+l} \in U \] (40b)

\[ P (C_{x_{k+l}, s_{k+l}}) \leq \epsilon \] (40c)

\[ \beta - \mathbb{E} \left[ (s_{k+l} - x_{k+l})^T A_{k+l} (s_{k+l} - x_{k+l}) \right] \]

\[ \leq \epsilon (\beta - 1/\lambda_0(M'_{k+l+1})) \] (41)

where \( M'_{k+l} \) is computed using (5) and \( \lambda_0(M'_{k+l+1}) \) is the minimal eigenvalue of \( M'_{k+l+1} \).

VI. RESULTS

In this section we describe our implementation and then evaluate the capabilities of our proposed approach. Simulations are performed in Gazebo with a quadruplet of semi-principle axes \((0.18, 0.18, 0.06)\) m. We refer the readers to [41] for the quadruplet dynamics. The ground truth odometry from Gazebo is used to measure the robot pose, mimicking a motion capture system. This measurement is then corrupted with noise which is zero mean with covariance \( \Sigma = diag(0.05m^2, 0.05m^2, 0.05m^2, 0.1deg^2, 0.1deg^2) \). The optimization of (40) is performed using the model predictive control approach developed in [41] and modified to meet our requirements. In all the experiments we use \( \epsilon = 0.05 \) and a look-ahead horizon of two seconds with a discretization of 0.1 seconds. The performance is evaluated on an Intel® Core i7-6500U CPU@2.50GHz×4 with 8GB RAM under Ubuntu 16.04 LTS. The results reported are averaged over 10 different simulation runs.

Comparison to bounding volume approaches: We compare our approach to bounding volume methods [10], [11] which enlarge robots and obstacles with their 3-σ confidence ellipsoids. Computation time and complexity are greatly reduced with bounding volumes. However, plans tend to be overly conservative and suboptimal. In the experiment, we consider a quadrotor at \((0, 0, 1.4)\) m moving to the goal location at \((0, 13, 1.4)\) m. An obstacle of semi-principle axes \((1, 1, 1)\) m is located in between at \((0, 3, 3)\) m. We conduct the experiment with varying measurement noise of \( \Sigma, 2\Sigma, 3\Sigma, \) and \( 4\Sigma \).

To compare the efficiency of our approach we define the compute the following metrics: (a) \( d \)– minimum distance between the quadrotor and the obstacle, (b) \( l \)– total trajectory length, and (c) \( T \)– total trajectory duration.

The results can be seen in the first two rows of Table II. In all the cases, our approach is more efficient as can be seen from the shorter average trajectory length and duration, which is more evident as the measurement noise increases. We also provide a comparison to center point approximation [1], which is also computationally less intensive. However, as recognized in [3], if the covariance is small, the approximated probability can be much smaller than the exact value. Moreover, the approach works well only when the sizes of objects are relatively very small compared with their position uncertainties [5]. This is seen in the last row of Table II. For measurement noise \( \Sigma \) and \( 2\Sigma \) the approach resulted in collision for all the runs. For measurement noise \( 3\Sigma \) and \( 4\Sigma \), the approach succeeded in 55% and 35% of the runs, respectively. This reduced success percentages are due to lower values of the collision probabilities computed. The executed trajectories for all the three approaches in Table II are shown in Fig. 1. For a given \( \epsilon \), the metric \( d \) allows us
to define a measure of risk—distance to closest obstacle. However, increasing $\epsilon$, increases risk as we solicit controls such that the collision probability is at most $\epsilon$. For example, in the scenario considered in Fig. 1 (b), an $\epsilon \geq 0.2$ lead to collision in 80% of the experiments.

**Four obstacles:** In this experiment, we consider four obstacles which are placed (a) far apart from each, (b) two obstacles are placed close to each other, and (c) all obstacles are placed close to each other. In each of the cases, the quadrotor starting from $(0, 0, 1.4)$ m has to reach the goal location at $(0, 13, 1.4)$ m. The respective trajectories in each of the three cases can be seen in Fig. 2a-2c. The change in configuration of the obstacles affects the collision probability computation which is reflected in the respective executed trajectories. The results are shown in Table III.

**Column domain:** We test the adaptability of our approach to cluttered and challenging environments. Fig. 3 shows a snapshot of the column domain which consists of 19 cylindrical columns. We consider two different scenarios where the quadrotor needs to avoid collision with the columns to reach the goal location. In scenario 1, the quadrotor starting from $(-2.5, 0, 1.4)$ m has to reach the goal location at $(2.5, 12, 1.4)$ m. The trajectory followed is seen in Fig. 2d with an average trajectory length $l=13.72$ m, and trajectory time $T=13.14$ (±0.02) seconds. The average minimum distance between the quadrotor and the obstacles $d=0.21$ m. In scenario 2, the quadrotor starting from $(0, 0, 1.4)$ m has to reach the goal location at $(1, 12, 1.4)$ m. The trajectory followed is seen in Fig. 2e with $l=13.42$ m, $T=13.37$ (±0.05) seconds and $d=0.66$ m. Note that since there are 19 columns, there are as many collision avoidance constraints. Our method is computationally less intensive and hence solvable in real time.

**VII. Conclusion**

In this paper, we present a novel approach to compute an exact as well as a tighter bound for collision probability under Gaussian motion and sensing uncertainties. The collision constraint is formulated as the distance between Löwner-John ellipsoids of the robot and the obstacles. Efficiency of our approach with respect to trajectory length and duration is tested in simulation by comparing with bounding volume approaches. Trajectory variation due to change in obstacle configurations have also been tested and it is seen that our method readily adapts to the same. Finally, we also test our approach in a heavily cluttered column domain.

Our approach has several limitations. First, we assume a Gaussian parametrization for the belief states. This may not be an admissible approximation in some cases, for example, when obstacles are not known a-priori, one may need to consider multi-modal beliefs. In such applications one may resort to nonlinear filters (say particle filters) for modeling non-Gaussian beliefs. Yet, in aerial as well as mobile robotics, EKF is extensively used for state estimation since Gaussian distributions are pertinent in a variety of applications. Gaussian distribution is thus a reasonable assumption in any such application. Second, we represent the robot and obstacles using their minimum volume enclosing ellipsoids. For a convex polygon $P \subseteq \mathbb{R}^n$, $J(P)$ provides an $n$-rounding of $P$, that is, $n^{-1}J(P) \subseteq P \subseteq J(P) \cap \mathbb{R}^n$ for symmetric $P$. Although the tightness of the bound is still an open problem, for a large class of problems, the approximation of $P$ using $J(P)$ gives a reasonable first (and sometimes optimal) bound [42].

**Acknowledgment**

The authors are grateful to the reviewers for their helpful suggestions.

**References**

[1] N. E. Du Toit and J. W. Burdick, “Probabilistic collision checking with chance constraints,” IEEE Transactions on Robotics, vol. 27, no. 4, pp. 809–815, 2011.
[2] S. Patil, J. Van Den Berg, and R. Alterovitz, “Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty,” in IEEE International Conference on Robotics and Automation, 2012, pp. 3238–3244.
[3] C. Park, J. S. Park, and D. Manocha, “Fast and bounded probabilistic collision detection for high-DOF trajectory planning in dynamic environments,” IEEE Transactions on Automation Science and Engineering, vol. 15, no. 3, pp. 980–991, 2018.
[4] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, “Provably safe robot navigation with obstacle uncertainty,” The International Journal of Robotics Research, vol. 37, no. 13-14, pp. 1760–1774, 2018.
[5] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 776–783, 2019.
[6] A. Thomas, F. Mastrogiovanni, and M. Baglietto, “An Integrated Localization, Motion Planning and Obstacle Avoidance Algorithm in Belief Space,” Intelligent Service Robotics, vol. 14, no. 2, pp. 235–250, 2021.
[7] A. Lambert, D. Gruyer, and G. Saint Pierre, “A fast Monte Carlo algorithm for collision probability estimation,” in 10th IEEE International Conference on Control, Automation, Robotics and Vision, 2008, pp. 406–411.
[8] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, “Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns,” Autonomous Robots, vol. 35, no. 1, pp. 51–76, 2013.
Fig. 2: (a)-(c) Top view (x-y) of four obstacles in different locations. (d)-(e) Top view of the column domain. The solid blue lines represent the trajectories executed by the quadrotor and the red blobs represent the obstacles.