The Report Committee for Apurva Patil
certifies that this is the approved version of the following report:

Analytical Bounds for End-to-End Risks in Stochastic Robot Navigation

SUPERVISING COMMITTEE:

Takashi Tanaka, Supervisor

Dongmei "Maggie" Chen, Co-supervisor

Luis Sentis

Analytical Bounds for End-to-End Risks in Stochastic Robot Navigation

by

Apurva Patil

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Mechanical Engineering

The University of Texas at Austin

December 2024

Abstract

Analytical Bounds for End-to-End Risks in Stochastic Robot Navigation

Apurva Patil,
The University of Texas at Austin, 2024

SUPERVISORS: Takashi Tanaka, Dongmei "Maggie" Chen

In this report, we present analytical methods to estimate the collision probability of motion plans for autonomous agents with both discrete-time and continuous-time dynamics operating under Gaussian uncertainties. Motion plans generated by planning algorithms cannot be perfectly executed by autonomous agents in reality due to the inherent uncertainties in the real world. Estimating end-to-end risk is crucial to characterize the safety of trajectories and plan risk optimal trajectories. In this report, we derive upper and lower bounds for end-to-end collision probability of motion plans using results from probability theory. Using ground robot navigation examples, we numerically demonstrate that our methods are considerably faster than the naïve Monte Carlo sampling method and the proposed bounds are significantly less conservative than Boole's bound commonly used in the literature.

Table of Contents

Chapter 1: Introduction

Motion plans for mobile robots in obstacle-filled environments can be generated by autonomous trajectory planning algorithms LaValle (2006). In reality, due to the presence of uncertainties, the robots cannot follow the planned trajectories perfectly, and collisions with obstacles occur with a non-zero probability, in general. To address this issue, risk-aware motion planning has received considerable attention over the years (e.g., Strawser and Williams (2018)). Optimal planning under set-bounded uncertainty provides some solutions against worst-case disturbances (e.g., Majumdar and Tedrake (2013)). However, in many cases, modeling uncertainties with unbounded (e.g. Gaussian) distributions has a number of advantages over a set-bounded approach Blackmore et al. (2011). In the case of unbounded uncertainties, it is generally difficult to guarantee safety against all realizations of noise. This motivates for an efficient risk estimation technique that can both characterize the safety of trajectories and be embedded in the planning algorithms to allow explicit trade-offs between control optimality and safety. In this report, we develop analytical frameworks of end-to-end risk estimation of motion plans for autonomous agents, with both discrete-time and continuous-time dynamics operating under Gaussian uncertainty.

1.1 Discrete-Time Dynamics Systems

We assume that a planned trajectory in a known configuration space is given and a robot follows this trajectory in finite time T. If E_t represents an event that the robot collides with the obstacles at time step t, then the end-to-end probability of failure can be formulated as $P\left(\bigvee_{t=0}^{T} E_t\right)$. In general, $\{E_t\}_{t=0}^{T}$ are statistically dependent events; hence the exact evaluation of this probability is challenging. In this report, we derive upper and lower bounds for the end-to-end failure probability using inequalities from probability theory.

Monte Carlo and other sampling-based methods (e.g., Blackmore et al. (2010), Janson et al. (2018)) provide estimates of end-to-end failure probability by computing the ratio of the number of simulated executions that collide with obstacles. However, these methods are often expensive in computation due to the need for a large number of simulation runs to obtain reliable estimates. Moreover, they provide no guarantee that they will not underestimate the failure probability. Various analytical approaches also have been proposed in the literature. Using the law of total probability, we can write

$$P\left(\bigvee_{t=0}^{T} E_t\right) = 1 - \prod_{t=0}^{T} P\left(E_t^c | E_{t-1}^c, \ldots, E_0^c\right) \tag{1.1}$$

where E_t^c represents the event that the robot is collision-free at time step t. It is challenging to compute (1.1) exactly because it requires evaluating integrals of multivariate distributions over non-convex regions. One approach to estimate (1.1) is to assume that the event E_t is independent of other events or depends only on E_{t-1} (Strawser and Williams (2018)), and subsequently approximate (1.1) as $1 - \prod_{t=0}^{T} P\left(E_t^c\right)$ or $1 - \prod_{t=0}^{T} P\left(E_t^c | E_{t-1}^c\right)$, respectively. However, these assumptions do not hold in general, and can result in overly conservative estimates or can even underestimate the failure probability.

Another popular approach is to use Boole's inequality to compute an upper bound to the end-to-end risk (Blackmore et al. (2011), Ono et al. (2015)). The inequality is given as

$$P\left(\bigvee_{t=0}^{T} E_t\right) \leq \sum_{t=0}^{T} P\left(E_t\right). \tag{1.2}$$

This bound is highly convenient due to its time-additive structure. The time-additivity allows the use of dynamic programming; hence (1.2) is commonly used in the risk-aware control problems (Frey et al. (2020)). However, this bound ignores the dependency among events $\{E_t\}_{t=0}^{T}$ and can result in overly conservative estimates, especially as the time discretization is refined. In contrast to the previous approaches, an approach presented in Patil et al. (2012) accounts for the fact that the distribution of the

state at each time step along the trajectory is conditioned on the previous time steps being collision-free. It truncates the estimated distributions of the robot's positions with respect to obstacles and approximates the truncated distributions as Gaussians. However, the Gaussianity is inexact and this approximation leads to an estimate that might not remain statistically consistent (Frey et al. (2020)).

1.1.1 Contributions

The contributions of our work are as follows: We account for the fact that the events of collision at different time-steps $\{E_t\}_{t=0}^{T}$ are statistically dependent. Unlike Patil et al. (2012), we compute the joint distribution of the entire robot trajectory without attempting to approximate the conditional state distributions. Using this joint trajectory distribution and leveraging inequalities from probability theory, we derive both upper and lower bounds for the end-to-end probability of failure. Our upper bounds are considerably tighter than the estimates obtained by Boole's inequality commonly used in the literature. The lower bounds, on the other hand, are useful for predicting how conservative the computed upper bounds are. Using a simulation study, we show the validity of our bounds. We also study the performance of our discrete-time bounds in the continuous-time setting as the underlying discretization is refined.

1.2 Continuous-Time Dynamics System

We also develop an analytical method of continuous-time risk estimation for autonomous agents with linear controlled Itô dynamics of the form (2.23). We assume that a planned trajectory with a finite length in a known configuration space $\mathcal{X} \subseteq \mathbb{R}^n$ is given and a robot tracks this trajectory in finite time T. If $\boldsymbol{x}^{sys}(t) \in \mathcal{X}$ represents the robot's position at time t, and $\mathcal{X}_{obs} \subset \mathcal{X}$ is the obstacle region, then the continuous-

time end-to-end risk $\mathcal{R}$ in the navigation of the given trajectory can be written as

$$\mathcal{R} = P\left(\bigcup_{t\in[0,T]} \boldsymbol{x}^{sys}(t) \in \mathcal{X}_{obs}\right). \tag{1.3}$$

Unfortunately, exact evaluation of (1.3) is a challenging task because all the states $\boldsymbol{x}^{sys}(t)$ across the time horizon $[0, T]$ are correlated with each other. In this report, we derive two upper bounds for $\mathcal{R}$ by leveraging properties of Brownian motion (also called a Wiener process) as well as Boole and Hunter's inequalities from probability theory.

Monte Carlo and other sampling based methods Blackmore et al. (2010), Janson et al. (2018) provide accurate estimates of (1.3) by computing the ratio of the number of simulated executions that collide with obstacles. However, these methods are often computationally expensive due to the need for a large number of simulation runs to obtain reliable estimates and are cumbersome to embed in planning algorithms.

The discrete-time risk estimation methods compute risks at the discretized time steps t_i, $i = 0, 1, \ldots, N$, and approximate the probability in (1.3) by

$$\mathcal{R} \approx P\left(\bigcup_{i=0}^{N} \boldsymbol{x}^{sys}(t_i) \in \mathcal{X}_{obs}\right). \tag{1.4}$$

Since the states $\{\boldsymbol{x}^{sys}(t_i)\}_{i=0,1,\ldots,N}$ are correlated with each other, evaluating the joint probability (1.4) exactly is computationally expensive Patil and Tanaka (2021). Several approaches have been proposed in the literature to upper bound this joint probability Blackmore et al. (2011), Patil and Tanaka (2021), Ono et al. (2015). The commonly used approach is to use *Boole's inequality* (a.k.a. union bound) which states that for any number of events $\mathcal{E}_j$, we have

$$P\left(\bigcup_{j=1}^{N} \mathcal{E}_j\right) \leq \sum_{j=1}^{N} P\left(\mathcal{E}_j\right). \tag{1.5}$$

Using this inequality, the probability in (1.4) can be decomposed the over timesteps as Blackmore et al. (2011):

$$\mathcal{R} \leq \sum_{i=0}^{N} P\left(\boldsymbol{x}^{sys}(t_i) \in \mathcal{X}_{obs}\right). \tag{1.6}$$

While the discrete-time risk estimation approaches can be applied for continuous-time systems, their performance is highly sensitive to the chosen time discretization. They may underestimate the risk when the sampling rate is low or may produce severely conservative estimates when the sampling rate is high Patil and Tanaka (2021).

Various continuous-time risk estimation approaches also have been proposed in the literature such as the approaches based on stochastic control barrier functions Santoyo et al. (2021), Yaghoubi et al. (2020), cumulative lyapunov exponent Oguri et al. (2019), and first-exit times Shah et al. (2011), Frey et al. (2020), Ariu et al. (2017), Chern et al. (2021). The analyses presented by Shah et al. Shah et al. (2011) and Chern et al. Chern et al. (2021) give the exact continuous-time collision probability as the solution to a partial differential equation (PDE). Shah et al. Shah et al. (2011) presents an analytic solution of this PDE for a simple case; namely that of a constrained spherical environment with no internal obstacles. However, such a closed-form solution is generally not tractable for complicated configuration spaces. Frey et al. Frey et al. (2020) uses an interval-based integration scheme to approximate the collision probability by leveraging classical results in the study of first-exit times. Ariu et al. Ariu et al. (2017) proposes an upper-bound for the continuous-time risk using the *reflection principle* of Brownian motion and Boole's inequality (1.5). In this report, we extend the results presented in Ariu et al. (2017) and derive tighter continuous-time risk bounds.

1.2.1 Contributions

The contributions of this work are summarized as follows: We first use the *Markov property* of Brownian motion, and tighten the risk bound derived in Ariu

et al. (2017). We then further reduce the conservatism of this bound by leveraging *Hunter's inequality* of the probability of union of events. Both our bounds possess the time-additive structure required in several optimal control techniques (e.g. dynamic programming) Ono et al. (2015), Van Den Berg et al. (2012), making these bounds useful for risk-aware motion planning. Finally, using a ground robot navigation example, we demonstrate that our method requires considerably less computation time than the naïve Monte Carlo sampling method. We also show that compared to the discrete-time risk bound (1.6), our bounds are tighter, and at the same time ensure conservatism (i.e. safety).

Chapter 2: Analytical Bounds for Discrete-Time End-to-End Risks in Stochastic Robot Navigation

In this Chapter, we present an analytical method to estimate the collision probability of motion plans for autonomous agents with discrete-time dynamics operating under Gaussian motion and sensing uncertainty. Motion plans generated by planning algorithms cannot be perfectly executed by autonomous agents in reality due to the inherent uncertainties in the real world. Estimating end-to-end collision probability is crucial to characterize the safety of trajectories and plan risk-optimal trajectories. In this work, we derive upper and lower bounds for end-to-end collision probability of motion plans using results from probability theory. Using a ground robot navigation example, we demonstrate that our method is considerably faster than the naïve Monte Carlo sampling method and the proposed bounds are significantly less conservative than Boole's bound commonly used in the literature.

2.1 Preliminaries

In this section, we formulate the risk analysis problem and review probability inequalities.

2.1.1 Problem Formulation

Suppose $\mathcal{X} \subseteq \mathbb{R}^d$, $d \geq 2$ is a known configuration space. Let $\mathcal{X}^{\text{obs}} \subset \mathcal{X}$, $\mathcal{X}^{\text{free}} = \mathcal{X} \backslash \mathcal{X}^{\text{obs}}$ and $\mathcal{X}^{\text{goal}} \subset \mathcal{X}^{\text{free}}$ be the obstacle region, obstacle-free region, and goal region, respectively. Given an initial position $x_0^{\text{plan}} \in \mathcal{X}^{\text{free}}$ of the robot, a planning algorithm generates a trajectory $\{x_t^{\text{plan}}\}_{t=0}^{T}$ by designing a finite, optimal sequence of control inputs $\{u_t^{\text{plan}}\}_{t=0}^{T-1}$ such that $x_T^{\text{plan}} \in \mathcal{X}^{\text{goal}}$. We call the finite sequence $\{x_t^{\text{plan}}\}_{t=0}^{T}$ the *planned trajectory*, satisfying

$$x_{t+1}^{\text{plan}} = A_t x_t^{\text{plan}} + B_t u_t^{\text{plan}}, \qquad t \in \{0, 1, \ldots, T-1\}.$$

Let $\boldsymbol{x}_t^{\text{sys}}$ be the actual position of the robot during the execution of the plan and $\boldsymbol{u}_t^{\text{sys}}$ be the control input applied at time step t. We call the finite sequence $\{\boldsymbol{x}_t^{\text{sys}}\}_{t=0}^{T}$ the *executed trajectory*. Suppose the executed trajectory satisfies

$$\boldsymbol{x}_{t+1}^{\text{sys}} = A_t \boldsymbol{x}_t^{\text{sys}} + B_t \boldsymbol{u}_t^{\text{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{x}_0^{\text{sys}} = x_0^{\text{plan}} \tag{2.1}$$

where $\boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$ is a Gaussian white noise that models the motion uncertainty. Suppose $\boldsymbol{x}_t \coloneqq \boldsymbol{x}_t^{\text{sys}} - x_t^{\text{plan}}$ represents the deviation of the robot from the planned trajectory and $\boldsymbol{u}_t \coloneqq \boldsymbol{u}_t^{\text{sys}} - u_t^{\text{plan}}$ be the control input deviation. Then, we can write

$$\boldsymbol{x}_{t+1} = A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t + \boldsymbol{w}_t, \quad \boldsymbol{x}_0 = 0, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, W_t) \tag{2.2}$$

Remark 1. For the analysis purpose, in this report, the system dynamics (2.1) are assumed to be linear. However, the presented methodology of risk estimation can also be applied to nonlinear Gaussian system by linearizing it around the nominal trajectory $\{x_t^{\text{plan}}, u_t^{\text{plan}}\}_{t=0}^{T}$.

Suppose the sensor model is given as

$$\boldsymbol{y}_t = C_t \boldsymbol{x}_t + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, V_t), \tag{2.3}$$

where $\boldsymbol{v}_t$ is a Gaussian white noise that models the noise in the measurements. If E_t represents the event that the robot collides with the obstacles at time step t, defining $E \coloneqq \bigvee_{t=0}^{T} E_t$, the end-to-end probability of failure in the trajectory tracking phase can be formulated as

$$P\left(\bigvee_{t=0}^{T} \boldsymbol{x}_t^{\text{sys}} \in \mathcal{X}^{\text{obs}}\right) = P\left(\bigvee_{t=0}^{T} E_t\right) = P(E). \tag{2.4}$$

2.1.2 Probability Bounds

In this section, we summarize first and second-order probability inequalities. These inequalities require computation of the terms $P(E_t)$ and $P(E_s \bigwedge E_t)$, which

are often easy to calculate. Define:

$$p_t \coloneqq P\left(E_t\right), \quad p_{s,t} \coloneqq P\left(E_s \bigwedge E_t\right),$$

$$S_1 \coloneqq \sum_{0 \le t \le T} p_t, \quad S_2 \coloneqq \sum_{0 \le s < t \le T} p_{s,t}. \tag{2.5}$$

Following are the upper bounds for $P\left(E\right)$:

- Kwerel's upper bound (Kwerel (1975)):

$$P\left(E\right) \le S_1 - \frac{2}{T+1} S_2. \tag{2.6}$$

  This is the closest upper bound for the probability of the union of events based on knowledge of S_1 and S_2.

- Kounias' upper bound:

$$P\left(E\right) \le S_1 - \max_{0 \le s \le T} \sum_{0 \le t \le T,\, t \ne s} p_{s,t}. \tag{2.7}$$

- Hunter's upper bound:

$$P\left(E\right) \le S_1 - \max_{\tau} \sum_{(s,t):e_{s,t} \in \tau} p_{s,t}. \tag{2.8}$$

  Here, τ is a spanning tree of the graph whose vertices are $E_0, E_1, \ldots, E_T$, with E_s and E_t joined by an edge $e_{s,t}$ if $E_s \bigwedge E_t \ne \emptyset$. Kruskal's minimum spanning tree algorithm (Kruskal (1956)) can be used to find the τ which attains the maximum of $\sum p_{s,t}$. In this work, we also compute a suboptimal Hunter's bound by choosing a particular spanning τ having edges $e_{0,1}, e_{1,2}, \ldots, e_{T-1,T}$:

$$P\left(E\right) \le S_1 - \sum_{1 \le t \le T} p_{t-1,t}. \tag{2.9}$$

Remark 2. Compared to (2.8), the bound in (2.9) is cheaper in computation. It also possesses the time-additive structure similar to Boole's bound (1.2); hence, this bound could be embedded in the risk-aware motion planning framework.

Following are the lower bounds for $P(E)$:

- Fréchet's lower bound: $P(E) \geq \max_{0 \leq t \leq T} p_t$.
- Bonferroni's second-order lower bound (Prékopa (1988)):

$$P(E) \geq S_1 - S_2. \tag{2.10}$$

- Dawson and Sankoff's lower bound (Dawson and Sankoff (1967)): If $S_1 > 0$,

$$P(E) \geq \frac{2}{k+1} S_1 - \frac{2}{k(k+1)} S_2,$$

where $k-1$ is the integer part of $2S_2/S_1$. It is the closest lower bound for the probability of union of events based on the knowledge of S_1 and S_2 (Galambos (1977)).

2.2 End-to-End Risk Analysis

In this section, we present a method to estimate the end-to-end risks based on the bounds presented in Section 2.1.2.

2.2.1 Distribution of the Closed-Loop Trajectory

During an actual execution of the planned trajectory, the robot will likely deviate from the plan due to model uncertainties and external disturbances. In order to compensate for these uncertainties, we design a trajectory tracking controller using a linear feedback policy and Kalman filter (Stengel (1994)). It is well-known that the optimal state feedback controller for stochastic linear quadratic regulator (LQR) is given by

$$\boldsymbol{u}_t = F_t \hat{\boldsymbol{x}}_{t|t}, \qquad t \in \{0, 1, \ldots, T-1\}, \tag{2.11}$$

where F_t are the LQR gains and $\hat{\boldsymbol{x}}_{t|t}$ are the state estimates based on the measurements $\{\boldsymbol{y}_s\}_{s=0}^{t}$. The *a priori* state estimates $\hat{\boldsymbol{x}}_{t|t-1}$ and the *a posteriori* state estimates $\hat{\boldsymbol{x}}_{t|t}$ evolve according to

$$\begin{aligned} \hat{\boldsymbol{x}}_{t|t-1} &= A_{t-1}\hat{\boldsymbol{x}}_{t-1|t-1} + B_{t-1}\boldsymbol{u}_{t-1}, \quad \hat{x}_{0|0} = 0 \\ \hat{\boldsymbol{x}}_{t|t} &= \hat{\boldsymbol{x}}_{t|t-1} + G_t\left(\boldsymbol{y}_t - C_t\hat{\boldsymbol{x}}_{t|t-1}\right), \end{aligned} \tag{2.12}$$

where G_t are the Kalman gains. See Stengel (1994) for the details of the derivation of (2.11) and (2.12). Combining (2.2), (2.11) and (2.12), the state deviation $\boldsymbol{x}_t$ and its *a priori* estimate $\hat{\boldsymbol{x}}_{t|t-1}$ jointly evolve as:

$$\overline{\boldsymbol{x}}_{t+1} = \overline{A}_t\overline{\boldsymbol{x}}_t + \overline{\boldsymbol{w}}_t, \qquad \overline{\boldsymbol{w}}_t \sim \mathcal{N}\left(0, \overline{W}_t\right)$$

where

$$\overline{\boldsymbol{x}}_t \coloneqq \begin{bmatrix} \boldsymbol{x}_t \\ \hat{\boldsymbol{x}}_{t|t-1} \end{bmatrix}, \quad \overline{\boldsymbol{w}}_t \coloneqq \begin{bmatrix} B_tF_tG_t\boldsymbol{v}_t + \boldsymbol{w}_t \\ (A_t + B_tF_t)\,G_t\boldsymbol{v}_t \end{bmatrix},$$

$$\overline{A}_t \coloneqq \begin{bmatrix} A_t + B_tF_tG_tC_t & B_tF_t\,(I - G_tC_t) \\ (A_t + B_tF_t)\,G_tC_t & (A_t + B_tF_t)\,(I - G_tC_t) \end{bmatrix},$$

$$\overline{W}_t \coloneqq \begin{bmatrix} B_tF_tG_tV_tG_t^\top F_t^\top B_t^\top + W_t & B_tF_tG_tV_tG_t^\top (A_t + B_tF_t)^T \\ (A_t + B_tF_t)G_tV_t^\top G_t^\top F_t^\top B_t^\top & (A_t + B_tF_t)G_tV_tG_t^\top (A_t + B_tF_t)^T \end{bmatrix}.$$

Now, we stack $\overline{\boldsymbol{x}}_t$ for all time steps, and via recursive substitution obtain

$$\overline{\boldsymbol{x}}^{\text{traj}} = M\overline{x}_0 + N\overline{\boldsymbol{w}}^{\text{traj}}, \qquad \overline{\boldsymbol{w}}^{\text{traj}} \sim \mathcal{N}\left(0, \operatorname*{diag}_{0\le t\le T-1} \overline{W}_t\right),$$

where

$$\overline{\boldsymbol{x}}^{\text{traj}} \coloneqq \begin{bmatrix} \overline{\boldsymbol{x}}_0 \\ \overline{\boldsymbol{x}}_1 \\ \overline{\boldsymbol{x}}_2 \\ \vdots \\ \overline{\boldsymbol{x}}_T \end{bmatrix}, \overline{\boldsymbol{w}}^{\text{traj}} \coloneqq \begin{bmatrix} \overline{\boldsymbol{w}}_0 \\ \overline{\boldsymbol{w}}_1 \\ \overline{\boldsymbol{w}}_2 \\ \vdots \\ \overline{\boldsymbol{w}}_{T-1} \end{bmatrix}, M \coloneqq \begin{bmatrix} I \\ \overline{A}_0 \\ \overline{A}_1\overline{A}_0 \\ \vdots \\ \overline{A}_{T-1}\dots\overline{A}_0 \end{bmatrix},$$

$$N \coloneqq \begin{bmatrix} 0 & 0 & \dots & 0 \\ I & 0 & \dots & 0 \\ \overline{A}_1 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \overline{A}_{T-1}\dots\overline{A}_1 & \overline{A}_{T-1}\dots\overline{A}_2 & \dots & I \end{bmatrix}.$$

Assuming $\overline{x}_0 = 0$, the distribution of $\overline{\boldsymbol{x}}^{\text{traj}}$ can be written as

$$\overline{\boldsymbol{x}}^{\text{traj}} \sim \mathcal{N}\left(0, \overline{X}^{\text{traj}}\right), \quad \overline{X}^{\text{traj}} = N\left(\operatorname*{diag}_{0 \le t \le T-1} \overline{W}_t\right) N^\top.$$

We can now obtain the distribution of the *closed-loop trajectory* $\boldsymbol{x}^{\text{traj}} \coloneqq \begin{bmatrix}\boldsymbol{x}_0 & \boldsymbol{x}_1 & \dots & \boldsymbol{x}_T\end{bmatrix}^\top$ by marginalizing $\overline{X}^{\text{traj}}$. Suppose $\mu_{\boldsymbol{x}^{\text{traj}}}\left(x^{\text{traj}}\right)$ denotes the distribution of the closed-loop trajectory, then the end-to-end probability of failure (2.4) can be calculated exactly by integrating $\mu_{\boldsymbol{x}^{\text{traj}}}\left(x^{\text{traj}}\right)$ over the obstacle-free region $\mathcal{X}^{\text{free}}$ as

$$P(E) = 1 - \int_{\mathcal{X}^{\text{free}}} \mu_{\boldsymbol{x}^{\text{traj}}}\left(x^{\text{traj}}\right) dx^{\text{traj}}. \tag{2.13}$$

However, $\mathcal{X}^{\text{free}}$ can possibly be a nonconvex region, and as stated earlier, obtaining an integral in a high dimensional space over a non-convex region is a computationally expensive problem. Therefore, we make use of the probability inequalities listed in Section 2.1.2 to obtain bounds for the end-to-end probability of failure (2.4).

2.2.2 Computation of the Bounds

The main task in evaluating the bounds presented in Section 2.1.2 is to compute the univariate probabilities p_t, and bivariate joint probabilities $p_{s,t}$. These can be computed from the distribution of $\boldsymbol{x}_t^{\text{sys}}$ and $\begin{bmatrix}\boldsymbol{x}_s^{\text{sys}} & \boldsymbol{x}_t^{\text{sys}}\end{bmatrix}^\top$. Using the distribution of the closed loop trajectory, we can obtain the distributions of $\boldsymbol{x}_t^{\text{sys}}$ and $\begin{bmatrix}\boldsymbol{x}_s^{\text{sys}} & \boldsymbol{x}_t^{\text{sys}}\end{bmatrix}^\top$ as

$$\mu_{\boldsymbol{x}_t^{\text{sys}}}\left(x_t^{\text{sys}}\right) = \mathcal{N}\left(x_t^{\text{plan}}, X_t\right)$$
$$\mu_{\boldsymbol{x}_s^{\text{sys}}\boldsymbol{x}_t^{\text{sys}}}\left(x_s^{\text{sys}}, x_t^{\text{sys}}\right) = \mathcal{N}\left(\begin{bmatrix}x_s^{\text{plan}} & x_t^{\text{plan}}\end{bmatrix}^\top, X_{st}\right)$$

where X_t and X_{st} are obtained by marginalizing $\overline{X}^{\text{traj}}$. Now, p_t and $p_{s,t}$ can be evaluated as

$$p_t = \int_{\mathcal{X}^{\text{obs}}} \mu_{\boldsymbol{x}_t^{\text{sys}}}\left(x_t^{\text{sys}}\right) dx_t^{\text{sys}}, \tag{2.14}$$

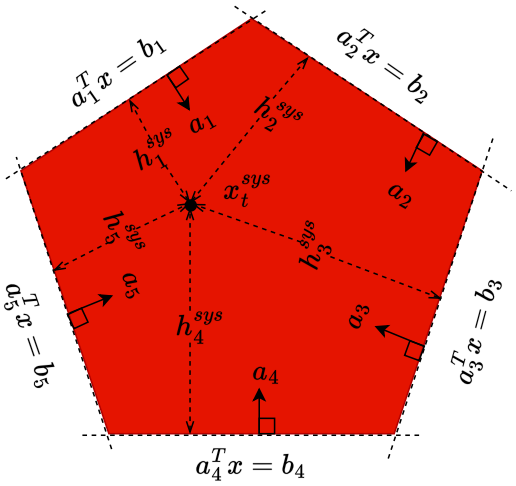

Figure 2.1: A polytopic obstacle, $\mathcal{X}_{obs_l}$ composed of five linear constraints (facets). For notational compactness, the subscript l is removed from $a_{i,l}$, $b_{i,l}$ and $h_{i,l}^{\text{sys}}$.

$$p_{s,t} = \int_{\mathcal{X}^{\text{obs}}} \int_{\mathcal{X}^{\text{obs}}} \mu_{\boldsymbol{x}_s^{\text{sys}} \boldsymbol{x}_t^{\text{sys}}} \left(x_s^{\text{sys}}, x_t^{\text{sys}}\right) dx_s^{\text{sys}} dx_t^{\text{sys}}. \tag{2.15}$$

In this report, we assume that the obstacles are $d-$ dimensional convex polytopes and develop a formulation to compute p_t and $p_{s,t}$ numerically. Suppose the obstacle region $\mathcal{X}^{\text{obs}}$ contains L disjoint convex polytopes and the l^{th} polytope $\mathcal{X}_l^{\text{obs}}$ is represented by a conjunction of I_l linear constraints as follows:

$$\mathcal{X}_l^{\text{obs}} = \bigwedge_{i=1}^{I_l} \left\{ x \in \mathbb{R}^d : a_{i,l}^\top x \geq b_{i,l} \right\}, \quad l \in \{1, 2, \ldots, L\}. \tag{2.16}$$

The vector $a_{i,l}$ is the unit normal of the facet i of the polytope l, pointing inside the polytope as shown in Figure 2.1. Let $\boldsymbol{h}_{i,l}^{\text{sys}}$ be a univariate random variable representing the perpendicular distance between the facet i of the polytope l and $\boldsymbol{x}_t^{\text{sys}}$ as shown in Figure 2.1. It can be shown that $\boldsymbol{h}_{i,l}^{\text{sys}} \sim \mathcal{N}\left(h_{i,l}^{\text{plan}}, H_{i,l}\right)$ where $h_{i,l}^{\text{plan}} \coloneqq a_{i,l}^\top x_t^{\text{plan}} - b_{i,l}$ and $H_{i,l} \coloneqq a_{i,l}^\top X_t a_{i,l}$.

2.2.2.1 Computation of p_t.

We can write p_t as

$$p_t = \sum_{l=1}^{L} p_{t,l}, \quad p_{t,l} = P\left(\boldsymbol{x}_t^{\text{sys}} \in \mathcal{X}_l^{\text{obs}}\right). \tag{2.17}$$

Using (2.16), $p_{t,l}$ can be written as

$$p_{t,l} = P\left(\bigwedge_{i=1}^{I_l} a_{i,l}^{\top} \boldsymbol{x}_t^{\text{sys}} \geq b_{i,l}\right) = P\left(\bigwedge_{i=1}^{I_l} \boldsymbol{h}_{i,l}^{\text{sys}} \geq 0\right).$$

Defining, $a_l = \begin{bmatrix} a_{1,l} & a_{2,l} & \dots & a_{I_l,l} \end{bmatrix}$, $\boldsymbol{h}_l^{\text{sys}} = \begin{bmatrix} \boldsymbol{h}_{1,l}^{\text{sys}} & \boldsymbol{h}_{2,l}^{\text{sys}} & \dots & \boldsymbol{h}_{I_l,l}^{\text{sys}} \end{bmatrix}^{\top}$, we obtain the distribution of $\boldsymbol{h}_l^{\text{sys}}$ as $\mu_{\boldsymbol{h}_l^{\text{sys}}}\left(h_l^{\text{sys}}\right) = \mathcal{N}\left(h_l^{\text{plan}}, H_l\right)$, where

$$h_l^{\text{plan}} \coloneqq \begin{bmatrix} h_{1,l}^{\text{plan}} & h_{2,l}^{\text{plan}} & \dots & h_{I_l,l}^{\text{plan}} \end{bmatrix}^{\top}, \quad H_l \coloneqq a_l^{\top} X_t a_l.$$

Therefore, $p_{t,l}$ can be computed as

$$p_{t,l} = \int_0^{\infty} \mu_{\boldsymbol{h}_l^{\text{sys}}}\left(h_l^{\text{sys}}\right) dh_l^{\text{sys}}. \tag{2.18}$$

In this work, MATLAB's `mvncdf` function is utilized for computing (2.18) numerically.

2.2.2.2 Computation of $p_{s,t}$.

Similar to (2.17), we write $p_{s,t}$ as

$$p_{s,t} = \sum_{l=1}^{L} \sum_{m=1}^{L} p_{st,lm},$$

where

$$p_{st,lm} = P\left[\left(\boldsymbol{x}_s^{\text{sys}} \in \mathcal{X}_l^{\text{obs}}\right) \bigwedge \left(\boldsymbol{x}_t^{\text{sys}} \in \mathcal{X}_m^{\text{obs}}\right)\right]. \tag{2.19}$$

Notice that (2.19) can be written as

$$p_{st,lm} = P\left[\left(\bigwedge_{i=1}^{I_l} \boldsymbol{h}_{i,l}^{\text{sys}} \geq 0\right) \bigwedge \left(\bigwedge_{i=1}^{I_m} \boldsymbol{h}_{i,m}^{\text{sys}} \geq 0\right)\right]. \tag{2.20}$$

Defining $K_{ss} \coloneqq X_s,\ K_{tt} \coloneqq X_t,\ K_{st} \coloneqq \operatorname{cov}\left(\boldsymbol{x}_s^{\text{sys}}, \boldsymbol{x}_t^{\text{sys}}\right)$,

$$\mu_{\boldsymbol{h}_l^{\text{sys}}\boldsymbol{h}_m^{\text{sys}}}\left(h_l^{\text{sys}}, h_m^{\text{sys}}\right) = \mathcal{N}\left(\begin{bmatrix} h_l^{\text{plan}} & h_m^{\text{plan}} \end{bmatrix}^\top,\ H_{lm}\right) \text{ where}$$

$$H_{lm} \coloneqq \begin{bmatrix} a_l^\top K_{ss} a_l & a_l^\top K_{st} a_m \\ a_m^\top K_{st}^\top a_l & a_m^\top K_{tt} a_m \end{bmatrix}.$$

Therefore, $p_{st,lm}$ can be computed as

$$p_{st,lm} = \int_0^\infty \int_0^\infty \mu_{\boldsymbol{h}_l^{\text{sys}}\boldsymbol{h}_m^{\text{sys}}}\left(h_l^{\text{sys}}, h_m^{\text{sys}}\right) dh_l^{\text{sys}} dh_m^{\text{sys}}. \tag{2.21}$$

MATLAB's `mvncdf` function can be utilized for computing (2.21) numerically.

2.3 Simulation Results

In this section, we demonstrate, in simulation, the validity of our bounds for the end-to-end risks. The configuration space is $\mathcal{X} = [0,1] \times [0,1]$ and the planned trajectory is assumed to satisfy

$$x_{t+1}^{\text{plan}} = x_t^{\text{plan}} + u_t, \qquad t \in \{0, 1, \ldots, T-1\}.$$

The executed trajectory $\{\boldsymbol{x}_t^{\text{sys}}\}_{t=0}^T$ satisfies the linearized robot dynamics,

$$\boldsymbol{x}_{t+1}^{\text{sys}} = \boldsymbol{x}_t^{\text{sys}} + \boldsymbol{u}_t^{\text{sys}} + \boldsymbol{w}_t, \quad \boldsymbol{x}_0^{\text{sys}} = x_0^{\text{plan}} \tag{2.22}$$

where $\boldsymbol{w}_t \sim \mathcal{N}(0, W_t)$, $W_t = \|x_{t+1}^{\text{plan}} - x_t^{\text{plan}}\| Z$, with $Z = 10^{-3} \times I$ (I is a 2×2 identity matrix). (2.22) is a natural model for ground robots whose location uncertainty grows linearly with the distance traveled. $\boldsymbol{u}_t^{\text{sys}}$ is computed using the state feedback policy, as explained in Section 2.2.1. We demonstrate two experiments:

2.3.0.1 Experiment 1.

In this experiment, we plan trajectories using the algorithm in Pedram et al. (2021). This algorithm plans trajectories via RRT* with the instantaneous safety criterion (i.e., at every time step, the confidence ellipse with a fixed safety level is

collision-free). For a given configuration space, four planned trajectories with 25%, 50%, 75%, and 99% instantaneous safety levels are shown in Figure 2.2 and our bounds for the *end-to-end* failure probabilities vs instantaneous safety levels are plotted in Figure 2.3. We validate our bounds by comparing them with the failure probabilities

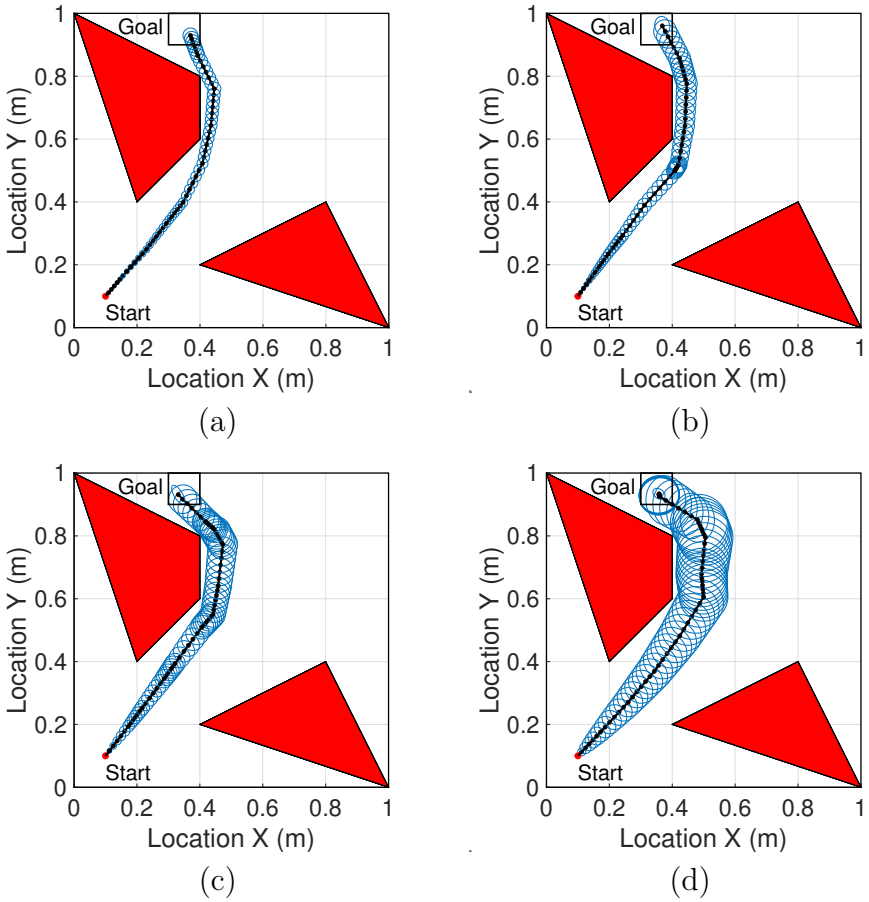

Figure 2.2: Trajectories planned with the instantaneous safety criterion. Trajectories are shown with (a) 25%, (b) 50%, (c) 75% and (d) 99% confidence ellipses.

computed using 10^5 Monte Carlo simulations (shown in black). Bonferroni's second-order lower bounds (shown with red dashed line) are trivial for all the paths in this example. As evident from the graph, Hunter's upper bound or its suboptimal version and Dawson and Sankoff's lower bound together provide close approximation to the Monte Carlo estimates of the end-to-end probability of failure. The graph shows that

the bounds presented in this work are significantly less conservative than Boole's bound (shown in solid green).

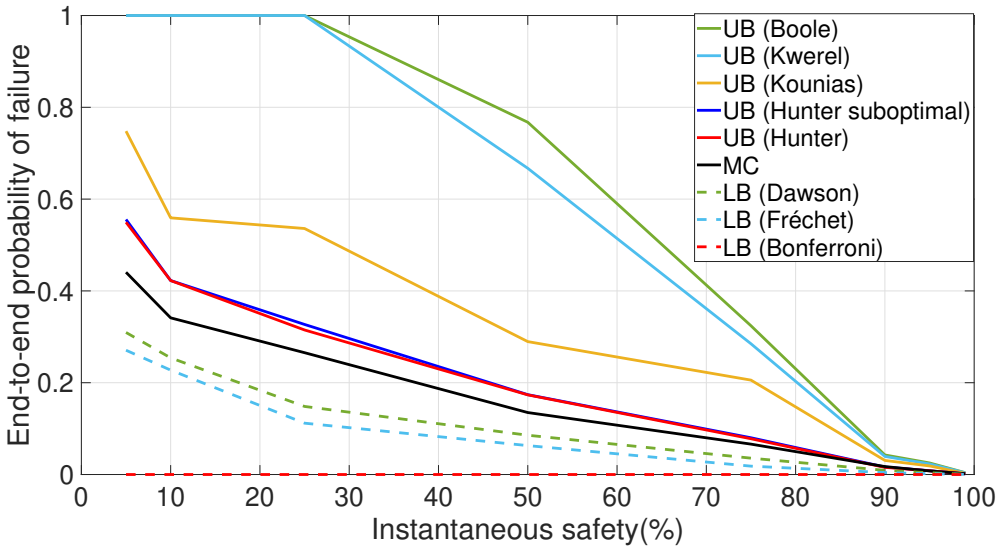

Figure 2.3: Upper (UB) and lower bounds (LB) of the end-to-end probabilities of failure for the trajectories with different instantaneous safety levels.

2.3.0.2 Experiment 2.

In this experiment, we demonstrate a larger statistical evaluation over 100 trajectories planned using RRT* (Karaman and Frazzoli (2011)) in randomly-generated environments (random initial, goal and obstacle positions). These trajectories are nominally safe i.e., only the planned positions $\{x_t^{\text{plan}}\}_{t=0}^{T}$ are ensured to be collision-free. Table 2.1 compares the mean absolute errors of different bounds with respect to 10^5 Monte Carlo simulations. The computation times for our MATLAB implementation are also reported. From the data presented, we can draw the following conclusions. First, the bounds presented in this work require significantly less computation time as compared to the Monte Carlo method. Second, our upper bounds provide considerably tighter estimates than Boole's bound at the expense of some additional computational overhead. Dawson and Hunter's estimates provide respec-

Table 2.1: Comparison of different bounds

Estimates	**Mean Abs. Error**	**Avg. Time [s]**
Monte Carlo	0	46.83
Upper bounds		
Boole	40.59	0.01
Kwerel	38.15	2.43
Kounias	13.34	2.42
Hunter	8.63	2.41
Hunter suboptimal	10.25	0.18
Lower bounds		
Bonferroni	54.88	2.44
Fréchet	40.08	0.01
Dawson	16.74	2.44

tively the best lower and upper bounds of risk among all. Finally, Hunter's suboptimal bound even though slightly more conservative, is computationally cheaper than Hunter's bound. As it also possesses the time-additive structure, this bound could be embedded in the risk-aware motion planning framework.

2.4 Discussion

2.4.1 Risk Bounds for Continuous-Time Systems

Although our results so far are restricted to discrete-time systems, in practice we are often interested in the safety of continuous-time systems. Consider a continuous-time robot dynamics governed by the following Itô process for $t \in [0, T]$:

$$d\boldsymbol{x}^{\text{sys}}(t) = \boldsymbol{v}^{\text{sys}}(t)dt + R^{\frac{1}{2}}(t)d\boldsymbol{b}(t), \quad \boldsymbol{x}^{\text{sys}}(0) = x_0^{\text{plan}} \tag{2.23}$$

where $\boldsymbol{v}^{\text{sys}}(t)$ is the velocity input command, $\boldsymbol{b}(t)$ is the d-dimensional standard Brownian motion, and $R(t)$ is given positive definite matrix for all $t \in [0, T]$. Suppose the time horizon $[0, T]$ is discretized in n time steps. Let $\mathcal{T} = (0 = t_0 < t_1 < \ldots < t_n = T)$ be a partition of the time horizon, and $\Delta t_j = t_{j+1} - t_j$. Time-discretization of (2.23)

under $\mathcal{T}$ based on Euler-Maruyama discretization (Kloeden and Platen (1992)) yields:

$$\boldsymbol{x}^{\text{sys}}_{t_{j+1}} = \boldsymbol{x}^{\text{sys}}_{t_j} + \boldsymbol{u}^{\text{sys}}_{t_j} + \boldsymbol{w}_{t_j} \quad \boldsymbol{x}^{\text{sys}}_0 \quad = x^{\text{plan}}_0 \tag{2.24}$$

where $\boldsymbol{u}^{\text{sys}}_{t_j} = \boldsymbol{v}^{\text{sys}}_{t_j} \Delta t_j$ and $\boldsymbol{w}_{t_j} \sim \mathcal{N}\left(0, \Delta t_j R_{t_j}\right)$. As we refine the time discretization of $[0, T]$ by increasing n, the discrete-time approximation (2.24) of the continuous-time system (2.23) becomes more accurate. Therefore, it is of our natural interest to study the impact of increased sampling rates on the aforementioned discrete-time risk bounds and how they can be used to imply the safety of continuous-time systems. Consider the configuration space and planned trajectories from Experiment 1 of Section 2.3. In (2.24), if we choose $R_{t_j} = \frac{\|x^{\text{plan}}_{tj+1} - x^{\text{plan}}_{tj}\|}{\Delta t_j} \times 10^{-3} I$ (I is an identity matrix), we retrieve the same discrete-time model as (2.22). For this system, we plot in Figure 2.4, 2.5 and 2.6 the probability bounds of Boole, suboptimal Hunter, and Dawson, respectively, at increasing rate of time discretization n. The probabilities obtained using Monte Carlo simulations for a high rate of time discretization (time steps $n = 206$) are plotted in black, in all three figures, and are assumed to be the ground truths of continuous-time risks. Figure 2.6 shows that Dawson and Sankoff's lower bound becomes tighter with the increase in the sampling rate. Similarly, it can be shown that Fréchet's bound also becomes tighter at the higher sampling rates. Note that the discrete-time lower bounds at all rates of time discretization are valid lower bounds for the continuous-time risks, whereas the upper bounds for a lower rate of time discretization might underestimate the continuous-time risks (see the suboptimal Hunter's bound at time steps 30 in Figure 2.5).

Figure 2.4 and 2.5 demonstrate that Boole's and Hunter's suboptimal upper bounds become less tight with the increase in the sampling rate. However, there is a significant difference in the rates at which they lose tightness. Boole's bound quickly diverges to the trivial probability of 1 as the sampling rate is increased, unlike the suboptimal Hunter's bound. It can be shown that Hunter's and Kounias' bounds also lose tightness at the higher sampling rates but they still perform better than Boole's bound. More investigation and comparison of our bounds at the high sampling rates

with the continuous-time risk estimates computed in the existing literature (e.g., Patil and Tanaka (2022), Patil et al. (2022)) are left for the future work.

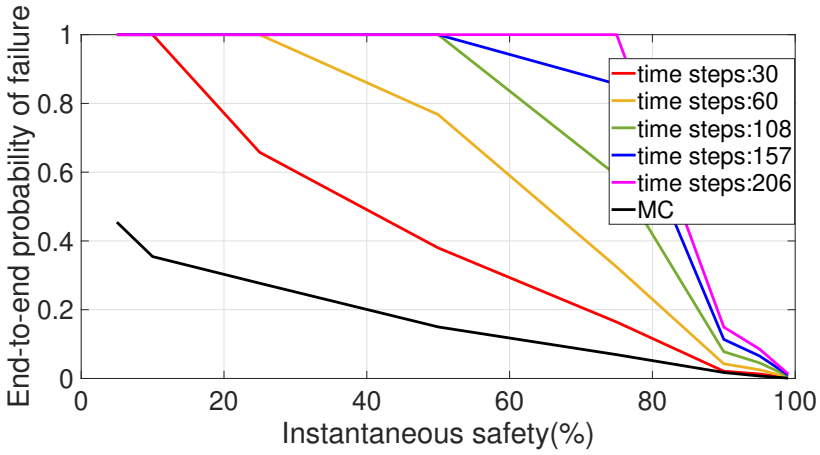

Figure 2.4: Boole's upper bounds of end-to-end probabilities of failure for different sampling rates.

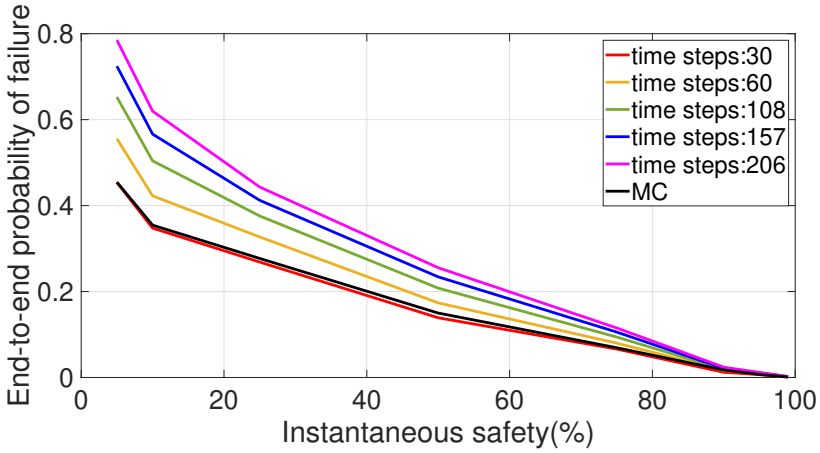

Figure 2.5: Hunter's suboptimal upper bounds of end-to-end probabilities of failure for different sampling rates.

2.4.2 Higher-Order Probability Bounds

In this work, we have implemented first and second-order probability bounds. The question naturally arises whether we can consider bounds of order higher than

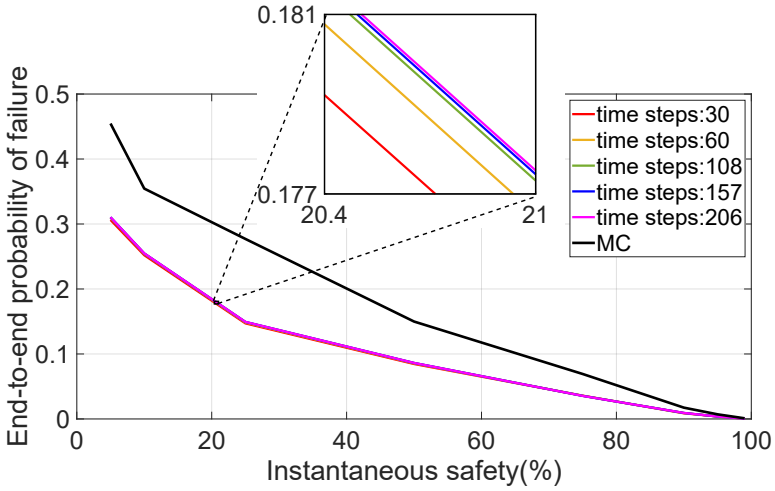

Figure 2.6: Dawson and Sankoff's lower bounds of end-to-end probabilities of failure for different sampling rates.

two. The classical inclusion-exclusion principle states that

$$P(E) = S_1 - S_2 + S_3 - S_4 + \ldots + (-1)^T S_{T+1}, \tag{2.25}$$

where S_r, $1 \leq r \leq T+1$, is defined as $S_r := \sum_{0 \leq j_1 < \ldots < j_r \leq T} P(E_{j_1} \bigwedge \ldots \bigwedge E_{j_r})$. The sum of the first r terms on the right side of (2.25) provides an upper bound to $P(E)$ when r is odd and a lower bound when r is even, producing Bonferroni's r^{th}-order bound. It is generally not true that Bonferroni's bounds increase in sharpness with the order (Schwager (1984)). Hence, Bonferroni's higher-order inequalities might not give tighter bounds than the ones considered in this work. A third-order upper bound computed using the Cherry Trees approach (Bukszár and Prekopa (2001)) can be tighter than Hunter's upper bound. Tighter higher-order upper and lower bounds can be computed using the linear programming algorithms (Prékopa (1988)). Of course, higher-order bounds are associated with higher computational complexities.

2.5 Conclusion

In this work, we presented an analytical method to compute upper and lower bounds for the collision probability of motion plans for systems with discrete-time

Gaussian dynamics. We made no independence assumptions on the events of collision at different time steps and computed the joint distribution of the entire robot trajectory. Using this joint trajectory distribution, we derived less conservative bounds for the failure probability. It was also demonstrated that the presented approach is considerably faster than the Monte Carlo sampling method. The future work includes the incorporation of these bounds in planning algorithms to generate risk-optimal trajectories. We also plan to conduct risk analysis for general robot dynamics that do not operate under Gaussian noises.

Chapter 3: Analytical Bounds for Continuous-Time End-to-End Risks in Stochastic Robot Navigation

In this Chapter, we present an analytical method to estimate the continuous-time collision probability of motion plans for autonomous agents with linear controlled Itô dynamics. Motion plans generated by planning algorithms cannot be perfectly executed by autonomous agents in reality due to the inherent uncertainties in the real world. Estimating end-to-end risk is crucial to characterize the safety of trajectories and plan risk optimal trajectories. In this Chapter, we derive upper bounds for the continuous-time risk in stochastic robot navigation using the properties of Brownian motion as well as Boole and Hunter's inequalities from probability theory. Using a ground robot navigation example, we numerically demonstrate that our method is considerably faster than the naïve Monte Carlo sampling method and the proposed bounds perform better than the discrete-time risk bounds.

3.1 Preliminaries and Problem Statement

3.1.1 Planned Trajectory

Let $\mathcal{X}_{free} = \mathcal{X} \backslash \mathcal{X}_{obs}$ be the obstacle-free region, and $\mathcal{X}_{goal} \subset \mathcal{X}$ be the target region. We assume that, for an initial position $x_0^{plan} \in \mathcal{X}_{free}$ of the robot, a trajectory planner gives us finite sequences of positions $\{x_j^{plan} \in \mathcal{X}_{free}\}_{j=0,1,\ldots,N}$ and control inputs $\{v_j^{plan} \in \mathbb{R}^n\}_{j=0,1,\ldots,N-1}$ such that $x_N^{plan} \in \mathcal{X}_{goal}$. Let $\mathcal{T} = (0 = t_0 < t_1 < \ldots < t_N = T)$ be the partition of the time horizon $[0, T]$, with $\Delta t_j = t_{j+1} - t_j$ satisfying

$$v_j^{plan} \Delta t_j = x_{j+1}^{plan} - x_j^{plan}, \qquad j = 0, 1, \ldots, N-1. \tag{3.1}$$

The *planned trajectory*, $x^{plan}(t)$, $t \in [0, T]$ is generated by the linear interpolations between x_j^{plan} and x_{j+1}^{plan}, $(j = 0, 1, \ldots, N-1)$.

3.1.2 Robot Dynamics

Assume that a robot following the planned path generates a trajectory defined by a random process $\boldsymbol{x}^{sys}(t)$, $t \in [0, T]$ with associated probability space $(\Omega, \mathcal{F}, P)$. We assume that the process $\boldsymbol{x}^{sys}(t)$ satisfies the following controlled Itô process:

$$d\boldsymbol{x}^{sys}(t) = \boldsymbol{v}^{sys}(t)dt + R^{\frac{1}{2}}d\boldsymbol{w}(t), \qquad t \in [0, T] \tag{3.2}$$

with $\boldsymbol{x}^{sys}(0) = x_0^{plan}$. Here, $\boldsymbol{v}^{sys}(t)$ is the velocity input command, $\boldsymbol{w}(t)$ is the n-dimensional standard Brownian motion, and R is a given positive definite matrix used to model the process noise intensity. We assume that the robot tracks the planned trajectory in open-loop using a piecewise constant control input:

$$\boldsymbol{v}^{sys}(t) = v_j^{plan} \qquad \forall\, t \in [t_j, t_{j+1}). \tag{3.3}$$

The time discretization of (3.2) under $\mathcal{T}$, based on the Euler-Maruyama method Kloeden and Platen (1992) yields:

$$\boldsymbol{x}^{sys}(t_{j+1}) = \boldsymbol{x}^{sys}(t_j) + \boldsymbol{v}^{sys}(t_j)\Delta t_j + \boldsymbol{n}(t_j) \tag{3.4}$$

where $\boldsymbol{n}(t_j) \sim \mathcal{N}(0, \Delta t_j R)$. Introducing $\boldsymbol{x}_j^{sys} \coloneqq \boldsymbol{x}^{sys}(t_j)$, $\boldsymbol{u}_j^{sys} \coloneqq \boldsymbol{v}^{sys}(t_j)\Delta t_j$, $\boldsymbol{n}_j \coloneqq \boldsymbol{n}(t_j)$, and $\Sigma_{\boldsymbol{n}_j} \coloneqq \Delta t_j R$, (3.4) can be rewritten as

$$\boldsymbol{x}_{j+1}^{sys} = \boldsymbol{x}_j^{sys} + \boldsymbol{u}_j^{sys} + \boldsymbol{n}_j, \qquad \boldsymbol{n}_j \sim \mathcal{N}(0, \Sigma_{\boldsymbol{n}_j}), \tag{3.5}$$

for $j = 0, 1, \ldots, N-1$. Further, using (3.3) and (3.1), $\boldsymbol{u}_j^{sys}$ can be rewritten as

$$\boldsymbol{u}_j^{sys} = v_j^{plan}\Delta t_j = x_{j+1}^{plan} - x_j^{plan}. \tag{3.6}$$

Let

$$\boldsymbol{x}(t) \coloneqq \boldsymbol{x}^{sys}(t) - x^{plan}(t), \qquad t \in [0, T] \tag{3.7}$$

be the deviation of the robot from the planned trajectory during trajectory tracking. Defining $\boldsymbol{x}_j \coloneqq \boldsymbol{x}(t_j)$, from (3.5), (3.6) and (3.7), the dynamics of $\boldsymbol{x}_j$ can be written as

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \boldsymbol{n}_j, \qquad \boldsymbol{n}_j \sim \mathcal{N}(0, \Sigma_{\boldsymbol{n}_j}) \tag{3.8}$$

for $j = 0, 1, \ldots, N-1$ with $\boldsymbol{x}_0 = 0$. The state $\boldsymbol{x}_j$ is distributed as $\boldsymbol{x}_j \sim \mathcal{N}(0, \Sigma_{\boldsymbol{x}_j})$ where $\Sigma_{\boldsymbol{x}_j}$ propagates according to $\Sigma_{\boldsymbol{x}_{j+1}} = \Sigma_{\boldsymbol{x}_j} + \Sigma_{\boldsymbol{n}_j}, j = 0, 1, \ldots, N-1$, with the initial covariance $\Sigma_{\boldsymbol{x}_0} = 0$.

3.1.3 Problem Statement

The continuous-time end-to-end risk $\mathcal{R}$ over the time horizon $[0, T]$ is formulated as (1.3). Under $\mathcal{T}$, we reformulate $\mathcal{R}$ as follows:

$$\mathcal{R} = P\left(\bigcup_{j=1}^{N} \bigcup_{t \in \mathcal{T}_j} \boldsymbol{x}^{sys}(t) \in \mathcal{X}_{obs}\right) \tag{3.9}$$

where $\mathcal{T}_j = [t_{j-1}, t_j]$, $j = 1, 2, \ldots, N$. In the rest of the paper, we deal with formulation (3.9) in order to derive upper bounds for $\mathcal{R}$.

3.1.4 Properties of Brownian Motion

Definition 1 (Markov property)**.** Let $\boldsymbol{w}(t)$, $t \geq 0$ be an n-dimensional Brownian motion started in $z \in \mathbb{R}^n$. Let $s \geq 0$, then the process $\widetilde{\boldsymbol{w}}(t) \coloneqq \boldsymbol{w}(t+s) - \boldsymbol{w}(s)$, $t \geq 0$ is again a Brownian motion started in the origin and it is independent of the process $\boldsymbol{w}(t)$, $0 \leq t \leq s$.

Theorem 1 (Reflection principle)**.** If $\boldsymbol{w}(t)$, $t \geq 0$ is a one-dimensional Brownian motion started in the origin and $d > 0$ is a threshold value, then

$$P\left(\sup_{s \in [0,t]} \boldsymbol{w}(s) \geq d\right) = 2P\left(\boldsymbol{w}(t) \geq d\right). \tag{3.10}$$

Refer to Durrett (2019) and Mörters and Peres (2010) for the proof.

3.2 Continuous-Time Risk Analysis

In this Section, we first reformulate $\mathcal{R}$ in terms of one-dimensional Brownian motions and then use the properties from Section 3.1.4 to compute bounds for $\mathcal{R}$.

For the analysis in Sections 3.2.1 to 3.2.3, we assume that $\mathcal{X}_{obs}$ is convex. In Section 3.2.4, we explain how the analysis can be generalized when $\mathcal{X}_{obs}$ is non-convex.

3.2.1 $\mathcal{R}$ in terms of One-Dimensional Brownian Motions

Let $\mathcal{S}_j$ be the path segment connecting x_{j-1}^{plan} and x_j^{plan} or equivalently, $x^{plan}(t_{j-1})$ and $x^{plan}(t_j)$, $j = 1, 2, \ldots, N$. Now, we conservatively approximate $\mathcal{X}_{obs}$ with a half space, similar to Morgan et al. (2014), Zhu and Alonso-Mora (2019). Since $\mathcal{S}_j$ and $\mathcal{X}_{obs}$ are convex, bounded and disjoint subsets of $\mathbb{R}^n$, from the hyperplane separation theorem, we can guarantee the existence of a hyperplane that strictly separates $\mathcal{S}_j$ and $\mathcal{X}_{obs}$. Let $\mathcal{H}_j : a_j^T x - b_j = 0$, $a_j \in \mathbb{R}^n$, $b_j \in \mathbb{R}$, $\|a_j\| = 1$ be a hyperplane such that $\mathcal{X}_{obs} \subseteq \mathcal{H}_j^+$ and $\mathcal{S}_j \subset \mathcal{H}_j^-$ where the half spaces $\mathcal{H}_j^+$ and $\mathcal{H}_j^-$ are defined as

$$\mathcal{H}_j^+ \coloneqq \{x \in \mathbb{R}^n : a_j^T x - b_j \geq 0\}, \quad \mathcal{H}_j^- \coloneqq \mathbb{R}^n \backslash \mathcal{H}_j^+. \tag{3.11}$$

Since $\mathcal{H}_j^+$ is a conservative approximation of $\mathcal{X}_{obs}$, we can upper bound $\mathcal{R}$ in (3.9) as

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N} \bigcup_{t \in \mathcal{T}_j} \boldsymbol{x}^{sys}(t) \in \mathcal{H}_j^+\right). \tag{3.12}$$

To find a least conservative upper bound, each hyperplane $\mathcal{H}_j$ can be constructed using the solution (y_1^*, y_2^*) to the following optimization problem:

$$\begin{aligned} \min_{y_1, y_2 \in \mathbb{R}^n} \quad & \|y_1 - y_2\| \\ \text{s.t.} \quad & y_1 \in \mathcal{X}_{obs},\ y_2 \in \mathcal{S}_j. \end{aligned} \tag{3.13}$$

The least conservative hyperplane $\mathcal{H}_j$ will be perpendicular to the line segment connecting y_1^* and y_2^*, and passing through y_1^*. If $d_j \coloneqq \|y_1^* - y_2^*\|$, then d_j represents the minimum distance of $\mathcal{S}_j$ from $\mathcal{X}_{obs}$. Fig. 3.1 shows an example of an optimal hyperplane $\mathcal{H}_j$ for a given $\mathcal{X}_{obs}$ and $\mathcal{S}_j$.

Now, it can be shown that

$$\left(\bigcup_{t \in \mathcal{T}_j} \boldsymbol{x}^{sys}(t) \in \mathcal{H}_j^+\right) \subseteq \left(\bigcup_{t \in \mathcal{T}_j} a_j^T \boldsymbol{x}(t) \geq d_j\right) \tag{3.14}$$

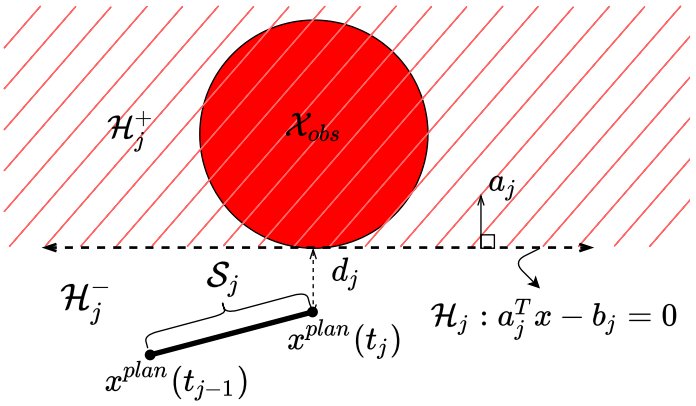

Figure 3.1: The least conservative hyperplane $\mathcal{H}_j$ approximating $\mathcal{X}_{obs}$ (shown in a red-faced circle) with a half space $\mathcal{H}_j^+$ (shown in red hatching). d_j is the minimum distance of $\mathcal{S}_j$ from $\mathcal{X}_{obs}$.

where $\boldsymbol{x}(t)$ is the deviation of the robot from the planned trajectory as defined in (3.7). Proof of (3.14) is presented in Appendix A. Using (3.12) and (3.14), $\mathcal{R}$ can be upper-bounded as

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N}\bigcup_{t\in\mathcal{T}_j} a_j^T\boldsymbol{x}(t) \geq d_j\right). \tag{3.15}$$

For the proposed robot dynamics (Section 3.1.2), it is trivial to show that $a_j^T\boldsymbol{x}(t)$ is a one-dimensional Brownian motion for $t \in [0, T]$ that starts in the origin. Let us denote $\boldsymbol{w}_j(t) \coloneqq a_j^T\boldsymbol{x}(t)$, $j = 1, 2, \ldots, N$. Now, (3.15) can be written as

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N}\max_{t\in\mathcal{T}_j}\boldsymbol{w}_j(t) \geq d_j\right). \tag{3.16}$$

Defining $\mathcal{E}_j \coloneqq \left(\max_{t\in\mathcal{T}_j}\boldsymbol{w}_j(t) \geq d_j\right)$, (3.16) can be rewritten as

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N}\mathcal{E}_j\right). \tag{3.17}$$

Since $\{\mathcal{E}_j\}_{j=1,2,\ldots,N}$ are non-independent events, computing (3.17) exactly is a challenging task. In the following sections, we derive bounds for $P\left(\bigcup_{j=1}^{N}\mathcal{E}_j\right)$.

3.2.2 First-Order Risk Bound

Define $p_j \coloneqq P(\mathcal{E}_j) = P\left(\max_{t\in[t_{j-1},t_j]} \boldsymbol{w}_j(t) \geq d_j\right)$. Applying Boole's inequality (1.5), the probability in (3.17) can be decomposed as

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N} \mathcal{E}_j\right) \leq \sum_{j=1}^{N} p_j. \tag{3.18}$$

This gives us a first-order risk bound for $\mathcal{R}$. p_j is the continuous-time risk associated with the time segment $\mathcal{T}_j = [t_{j-1}, t_j]$. Note that the bound in (3.18) possesses the time-additive structure which is helpful to use this bound in the risk-aware motion planning algorithms.

In order to take advantage of the reflection principle to compute p_j, Ariu et al. Ariu et al. (2017) proposes to compute an upper bound to p_j as

$$p_j \leq P\left(\max_{t\in[0,t_j]} \boldsymbol{w}_j(t) \geq d_j\right). \tag{3.19}$$

Using the reflection principle (3.10), the right side of (3.19) can be evaluated as

$$P\left(\max_{t\in[0,t_j]} \boldsymbol{w}_j(t) \geq d_j\right) = 2P(\boldsymbol{w}_j(t_j) \geq d_j) = 2P\left(a_j^T \boldsymbol{x}_j \geq d_j\right). \tag{3.20}$$

From (3.18), (3.19), and (3.20) we get

$$\mathcal{R} \leq 2\sum_{j=1}^{N} P\left(a_j^T \boldsymbol{x}_j \geq d_j\right). \tag{3.21}$$

The bound in (3.21) requires computing probabilities only at the discrete-time steps, simplifying the estimation of the continuous-time risk. However, the over-approximation in (3.19) introduces unnecessary conservatism that can be avoided using the Markov property of Brownian motion. Next, we present a way by which p_j can be computed exactly without any over-approximation.

For notational convenience, let us denote the random variables $\boldsymbol{w}_j(t_{j-1})$ and $\boldsymbol{w}_j(t_j)$ by $\boldsymbol{z}_j^s$ and $\boldsymbol{z}_j^e$ respectively:

$$\boldsymbol{z}_j^s \coloneqq \boldsymbol{w}_j(t_{j-1}) = a_j^T \boldsymbol{x}_{j-1}, \quad \boldsymbol{z}_j^e \coloneqq \boldsymbol{w}_j(t_j) = a_j^T \boldsymbol{x}_j \tag{3.22}$$

for $j = 1, 2, \ldots, N$. If $\mu_{\boldsymbol{\xi}}(\xi)$ denotes the probability density function (p.d.f.) of any random variable $\boldsymbol{\xi}$, then

$$\begin{aligned} \mu_{\boldsymbol{z}_j^s}(z_j^s) &= \mathcal{N}(0, \sigma^2_{\boldsymbol{z}_j^s}), \qquad \sigma^2_{\boldsymbol{z}_j^s} = a_j^T \Sigma_{\boldsymbol{x}_{j-1}} a_j, \\ \mu_{\boldsymbol{z}_j^e}(z_j^e) &= \mathcal{N}(0, \sigma^2_{\boldsymbol{z}_j^e}), \qquad \sigma^2_{\boldsymbol{z}_j^e} = a_j^T \Sigma_{\boldsymbol{x}_j} a_j. \end{aligned} \tag{3.23}$$

Let us define $\boldsymbol{z}_j \coloneqq \begin{bmatrix} \boldsymbol{z}_j^s & \boldsymbol{z}_j^e \end{bmatrix}^T \in \mathbb{R}^2$. It is straightforward to show that the joint distribution of $\boldsymbol{z}_j$ is

$$\mu_{\boldsymbol{z}_j}(z_j) = \mathcal{N}\left(0, \Sigma_{\boldsymbol{z}_j}\right), \qquad \Sigma_{\boldsymbol{z}_j} = \begin{bmatrix} \sigma^2_{\boldsymbol{z}_j^s} & \sigma^2_{\boldsymbol{z}_j^s} \\ \sigma^2_{\boldsymbol{z}_j^s} & \sigma^2_{\boldsymbol{z}_j^e} \end{bmatrix}. \tag{3.24}$$

Now, we compute p_j using the following theorem:

Theorem 2. If $\mu_{\boldsymbol{z}_j^s}(z_j^s)$ and $\mu_{\boldsymbol{z}_j}(z_j)$ are the distributions of the normal random variables $\boldsymbol{z}_j^s$ and $\boldsymbol{z}_j$, represented as (3.23) and (3.24) respectively, then p_j is given by:

$$p_j = \int_{z_j^s = d_j}^{\infty} \mu_{\boldsymbol{z}_j^s}(z_j^s) dz_j^s + 2 \int_{z_j^s = -\infty}^{d_j} \int_{z_j^e = d_j}^{\infty} \mu_{\boldsymbol{z}_j}(z_j) dz_j^e dz_j^s. \tag{3.25}$$

Proof. Let us define:

$$\begin{aligned} p_j^1 &\coloneqq P\left(\max_{t \in [t_{j-1}, t_j]} \boldsymbol{w}_j(t) \geq d_j,\ \boldsymbol{w}_j(t_{j-1}) \geq d_j \right), \\ p_j^2 &\coloneqq P\left(\max_{t \in [t_{j-1}, t_j]} \boldsymbol{w}_j(t) \geq d_j,\ \boldsymbol{w}_j(t_{j-1}) < d_j \right). \end{aligned}$$

Using the law of total probability, we can write p_j as

$$p_j = p_j^1 + p_j^2. \tag{3.26}$$

Since $(\boldsymbol{w}_j(t_{j-1}) \geq d_j) \subseteq \left(\max_{t \in [t_{j-1}, t_j]} \boldsymbol{w}_j(t) \geq d_j \right)$, p_j^1 can be computed as

$$p_j^1 = P\left(\boldsymbol{w}_j(t_{j-1}) \geq d_j\right) = \int_{z_j^s = d_j}^{\infty} \mu_{\boldsymbol{z}_j^s}(z_j^s) dz_j^s. \tag{3.27}$$

Now, we write p_j^2 as

$$\begin{aligned} p_j^2 &= P\left(\max_{t \in [t_{j-1}, t_j]} \boldsymbol{w}_j(t) \geq d_j,\ \boldsymbol{z}_j^s < d_j \right) \\ &= P\left(\max_{t \in [0, t_j - t_{j-1}]} \boldsymbol{w}_j(t + t_{j-1}) \geq d_j,\ \boldsymbol{z}_j^s < d_j \right). \end{aligned}$$

From Markov property of Brownian motion (Definition 1),

$$\widetilde{\boldsymbol{w}}_j(t) = \boldsymbol{w}_j(t + t_{j-1}) - \boldsymbol{w}_j(t_{j-1}), \quad t \in [0, (T - t_{j-1})] \tag{3.28}$$

is a one-dimensional Brownian motion that starts in the origin. Rewriting p_j^2 in terms of $\widetilde{\boldsymbol{w}}_j(t)$, we get

$$\begin{aligned} p_j^2 &= P\left(\max_{t\in[0,t_j-t_{j-1}]} \widetilde{\boldsymbol{w}}_j(t) \geq d_j - \boldsymbol{z}_j^s,\ \boldsymbol{z}_j^s < d_j\right) \\ &= \int_{-\infty}^{d_j} P\left(\max_{t\in[0,t_j-t_{j-1}]} \widetilde{\boldsymbol{w}}_j(t) \geq d_j - z_j^s\right) \mu_{\boldsymbol{z}_j^s}(z_j^s) dz_j^s. \end{aligned} \tag{3.29}$$

Since $d_j - z_j^s > 0,\ \forall\, z_j^s \in (-\infty, d_j)$, we can apply the reflection principle (3.10) and rewrite (3.29) as

$$p_j^2 = \int_{-\infty}^{d_j} 2P\big(\widetilde{\boldsymbol{w}}_j(t_j - t_{j-1}) \geq d_j - z_j^s\big)\, \mu_{\boldsymbol{z}_j^s}(z_j^s) dz_j^s. \tag{3.30}$$

Let us denote the random variable $\widetilde{\boldsymbol{w}}_j(t_j - t_{j-1})$ by $\boldsymbol{y}_j$. Using (3.28) and (3.22),

$$\boldsymbol{y}_j \coloneqq \widetilde{\boldsymbol{w}}_j(t_j - t_{j-1}) = \boldsymbol{w}_j(t_j) - \boldsymbol{w}_j(t_{j-1}) = \boldsymbol{z}_j^e - \boldsymbol{z}_j^s,$$

and the p.d.f. of $\boldsymbol{y}_j$ is $\mu_{\boldsymbol{y}_j}(y_j) = \mathcal{N}(0, \sigma_{\boldsymbol{y}_j}^2)$ where $\sigma_{\boldsymbol{y}_j}^2 = \sigma_{\boldsymbol{z}_j^e}^2 - \sigma_{\boldsymbol{z}_j^s}^2$. Now, (3.30) can be rewritten as

$$\begin{aligned} p_j^2 &= 2\int_{-\infty}^{d_j} \left(\int_{d_j - z_j^s}^{\infty} \mu_{\boldsymbol{y}_j}(y_j)\, dy_j\right) \mu_{\boldsymbol{z}_j^s}(z_j^s) dz_j^s \\ &= 2\int_{-\infty}^{d_j}\int_{d_j - z_j^s}^{\infty} \frac{1}{2\pi\sigma_{\boldsymbol{z}_j^s}\sigma_{\boldsymbol{y}_j}} exp\left\{-\frac{1}{2}\left(\frac{z_j^s}{\sigma_{\boldsymbol{z}_j^s}}\right)^2 - \frac{1}{2}\left(\frac{y_j}{\sigma_{\boldsymbol{y}_j}}\right)^2\right\} dy_j dz_j^s. \end{aligned} \tag{3.31}$$

The outside integral in right side of (3.31) is w.r.t. z_j^s and the inside is one is w.r.t. y_j. Substituting y_j with $z_j^e - z_j^s$, (3.31) can be rewritten as

$$\begin{aligned} p_j^2 =& 2\int_{z_j^s=-\infty}^{d_j}\int_{z_j^e=d_j}^{\infty} \frac{1}{2\pi\sigma_{\boldsymbol{z}_j^s}\sigma_{\boldsymbol{z}_j^e}\sqrt{1-\rho^2}} \cdot \\ & exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{z_j^s}{\sigma_{\boldsymbol{z}_j^s}}\right)^2 - \frac{2\rho\, z_j^s z_j^e}{\sigma_{\boldsymbol{z}_j^s}\sigma_{\boldsymbol{z}_j^e}} + \left(\frac{z_j^e}{\sigma_{\boldsymbol{z}_j^e}}\right)^2\right]\right\} dz_j^e\, dz_j^s \end{aligned} \tag{3.32}$$

where $\rho = \sigma_{\boldsymbol{z}_j^s}/\sigma_{\boldsymbol{z}_j^e}$. The expression inside the double integral of (3.32) is a bivariate normal distribution of $\boldsymbol{z}_j$. Hence,

$$p_j^2 = 2\int_{z_j^s=-\infty}^{d_j} \int_{z_j^e=d_j}^{\infty} \mu_{\boldsymbol{z}_j}(z_j) dz_j^e \, dz_j^s. \tag{3.33}$$

Combining (3.26), (3.27) and (3.33) we recover (3.25). □

MATLAB's `mvncdf` function can be utilized to compute the integrations (3.27) and (3.33) numerically.

3.2.3 Second-Order Risk Bound

The proposed first-order risk bound (3.18) can be tightened using a variant of Hunter's inequality that additionally considers the joint probability of consecutive events Prékopa (2003):

$$\mathcal{R} \leq P\left(\bigcup_{j=1}^{N} \mathcal{E}_j\right) \leq \sum_{j=1}^{N} p_j - \sum_{j=1}^{N-1} p_{j,\,j+1}$$

where $p_{j,\,j+1} \coloneqq P(\mathcal{E}_j \cap \mathcal{E}_{j+1})$ is the joint risk associated with the time segments $\mathcal{T}_j$ and $\mathcal{T}_{j+1}$. Computing $p_{j,\,j+1}$ exactly is challenging. In this work, we propose to compute a lower bound $p_{j,\,j+1}^{LB}$ of $p_{j,\,j+1}$ using the following theorem:

Theorem 3. If $t_{j-1} = \hat{t}_j^0 < \hat{t}_j^1 < \ldots < \hat{t}_j^{r_j} = t_j$ is a discretization of the time segment $\mathcal{T}_j$, and $\boldsymbol{z}_j^i$, $\mathcal{D}_j$ are defined as

$$\boldsymbol{z}_j^i \coloneqq \boldsymbol{w}_j(\hat{t}_j^i) = a_j^T \boldsymbol{x}(\hat{t}_j^i),$$

$$\mathcal{D}_j \coloneqq \left(\boldsymbol{z}_j^0 < d_j\right) \cap \left(\boldsymbol{z}_j^1 < d_j\right) \cap \ldots \cap \left(\boldsymbol{z}_j^{r_j} < d_j\right),$$

then $p_{j,\,j+1}$ is lower bounded by $p_{j,\,j+1}^{LB}$ given as

$$p_{j,\,j+1}^{LB} = 1 - P(\mathcal{D}_j) - P(\mathcal{D}_{j+1}) + P(\mathcal{D}_j \cap \mathcal{D}_{j+1}).$$

Proof. Introduce $\mathcal{C}_j$ as

$$\begin{aligned}\mathcal{C}_j =& \left(\boldsymbol{w}_j(\hat{t}_j^0) \geq d_j\right) \cup \left(\boldsymbol{w}_j(\hat{t}_j^1) \geq d_j\right) \cup \ldots \cup \left(\boldsymbol{w}_j(\hat{t}_j^{r_j}) \geq d_j\right) \\ =& \left(\boldsymbol{z}_j^0 \geq d_j\right) \cup \left(\boldsymbol{z}_j^1 \geq d_j\right) \cup \ldots \cup \left(\boldsymbol{z}_j^{r_j} \geq d_j\right) \\ =& \mathcal{D}_j^c.\end{aligned}$$

Now, since $\mathcal{C}_j \subset \mathcal{E}_j$

$$\begin{aligned}p_{j,\,j+1} &\geq P(\mathcal{C}_j \cap \mathcal{C}_{j+1}) \\ &= 1 - P(\mathcal{D}_j \cup \mathcal{D}_{j+1}) \\ &= 1 - P(\mathcal{D}_j) - P(\mathcal{D}_{j+1}) + P(\mathcal{D}_j \cap \mathcal{D}_{j+1}) \\ &= p_{j,\,j+1}^{LB}.\end{aligned}$$

□

$P(\mathcal{D}_j)$ can be computed by finding the joint distribution of $\begin{bmatrix}\boldsymbol{z}_j^0 & \boldsymbol{z}_j^1 & \ldots & \boldsymbol{z}_j^{r_j}\end{bmatrix}^T$ and $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$ by finding the joint distribution of $\begin{bmatrix}\boldsymbol{z}_j^0 & \boldsymbol{z}_j^1 & \ldots & \boldsymbol{z}_j^{r_j} & \boldsymbol{z}_{j+1}^0 & \boldsymbol{z}_{j+1}^1 & \ldots & \boldsymbol{z}_{j+1}^{r_{j+1}}\end{bmatrix}^T$. The computations of $P(\mathcal{D}_j)$ and $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$ are summarized in Appendix B. Now, we get our second-order risk bound as follows:

$$\mathcal{R} \leq \sum_{j=1}^{N} p_j - \sum_{j=1}^{N-1} p_{j,\,j+1}^{LB}. \tag{3.34}$$

Similar to the first-order risk bound (3.18), this bound also possesses the time-additive structure. Note that the higher sampling rates we choose to discretize the time segments $\mathcal{T}_j$ (a set of higher r_j's), the tighter the bound in (3.34) becomes.

3.2.4 Risk Analysis when $\mathcal{X}_{obs}$ is Non-Convex

As mentioned earlier, the analysis in Sections 3.2.1 to 3.2.3 assumes that $\mathcal{X}_{obs}$ is convex, which is sufficient to guarantee the existence of a set of separating hyperplanes $\{\mathcal{H}_j\}_{j=1,2,\ldots,N}$. When $\mathcal{X}_{obs}$ is non-convex, we partition it into M subregions $\mathcal{X}_{obs_m}$, $m = 1, 2, \ldots, M$ such that $\mathcal{X}_{obs} = \left(\bigcup_{m=1}^{M} \mathcal{X}_{obs_m}\right)$ and a set of separating hyperplanes

$\{\mathcal{H}_j\}_{j=1,2,\ldots,N}$ exists for each $\mathcal{X}_{obs_m}$. We then bound $\mathcal{R}$ as

$$\mathcal{R} \leq \sum_{m=1}^{M} \mathcal{R}_m, \qquad \mathcal{R}_m = P\left(\bigcup_{t\in[0,T]} \boldsymbol{x}^{sys}(t) \in \mathcal{X}_{obs_m} \right).$$

The first and second-order upper bounds for $\mathcal{R}_m$ can be computed using the analysis in Sections 3.2.1 to 3.2.3. In order to obtain tight upper-bounds for $\mathcal{R}$, the partitioning of $\mathcal{X}_{obs}$ can be optimized which is left for the future work.

3.3 Simulation Results

In this Section, we demonstrate the validity and performance of our continuous-time risk bounds via a ground robot navigation simulation. The configuration space is $\mathcal{X} = [0,1] \times [0,1]$. We assume that the robot dynamics are governed by the Itô process (3.2), with $R = 10^{-3} \times I$ (I is a 2×2 identity matrix), and it is commanded to travel at a unit velocity i.e., $\|\boldsymbol{v}^{sys}(t)\| = 1$, $t \in [0,T]$. As explained in Section 3.1, we discretize the dynamics (3.2) under the time partition $\mathcal{T}$. Due to the unit velocity assumption, $\Delta t_j = \|x_{j+1}^{plan} - x_j^{plan}\|$. Hence, our discrete-time robot dynamics are

$$\boldsymbol{x}_{j+1}^{sys} = \boldsymbol{x}_j^{sys} + \boldsymbol{u}_j^{sys} + \boldsymbol{n}_j, \quad \boldsymbol{n}_j \sim \mathcal{N}(0, \|x_{j+1}^{plan} - x_j^{plan}\| R), \tag{3.35}$$

where $\boldsymbol{u}_j^{sys}$ is defined as per (3.6). The model (3.35) is natural for ground robots whose location uncertainty grows linearly with the distance traveled.

First, we plan trajectories using RRT* with the instantaneous safety criterion Pedram et al. (2021) (i.e., at every time step, the confidence ellipse with a fixed safety level is collision-free). For a given configuration space, four planned trajectories with 95%, 75%, 50%, and 25% instantaneous safety levels are shown in Fig. 3.2. In each case, the confidence ellipses grow in size with the distance since the robot tracks these trajectories in open-loop. Fig. 3.3 plots the continuous and discrete-time risk bounds for these plans having different instantaneous safety (risk) levels. For validation, we compute failure probabilities using 10^5 Monte Carlo simulations at a high rate of time

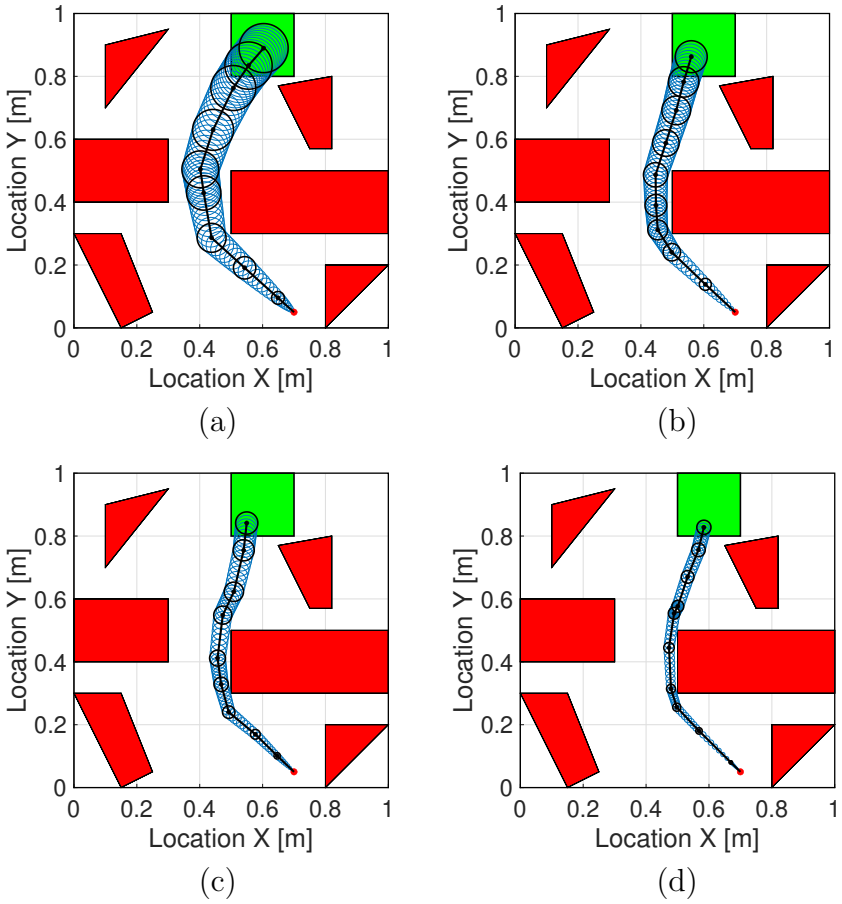

Figure 3.2: Trajectories planned with the instantaneous safety criterion Pedram et al. (2021) are shown in black. The black dots on the trajectories represent the planned positions $\{x_j^{plan}\}_{j=1,2,\ldots,N}$. The red dot represents the initial position x_0^{plan} of the robot. The red-faced polygons represent $\mathcal{X}_{obs}$ and the green faced rectangle represents $\mathcal{X}_{goal}$. The trajectories are shown with (a) 95%, (b) 75%, (c) 50% and (d) 25% confidence ellipses. The ellipses at the time steps $\{t_j\}_{j=1,2,\ldots,N}$ are shown in black and the ellipses at the intermediate time steps are shown in blue.

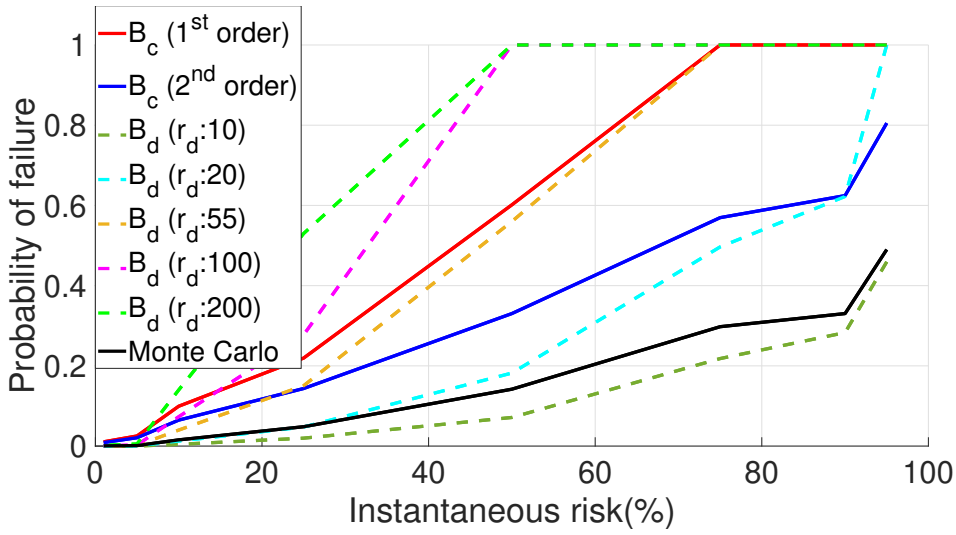

Figure 3.3: End-to-end probabilities of failure computed for the trajectories with different instantaneous risk levels. The solid red and blue graphs represent the first-order and second-order continuous-time risk bounds (B_c) respectively. The dotted graphs are discrete-time risk bounds (B_d) computed using (1.6) at different rates of time discretization (r_d). The Monte Carlo estimates of the same trajectories are shown in black.

discretization ($r_d = 100$) and assume them as the ground truths (shown in black). The dotted graphs are the discrete-time risk bounds (B_d) computed using (1.6) at different rates of time discretization (r_d). As is evident from the graph, the discrete-time risk bounds at a lower rate of time discretization underestimate the Monte Carlo estimates, and as the time-discretization rate increases, they become overly conservative. On the other hand, our continuous-time risk bounds (B_c) (shown with solid red and blue graphs) are tighter, and at the same time ensure conservatism.

Next, we demonstrate a larger statistical evaluation over 100 trajectories planned using RRT* in randomly-generated environments (random initial, goal and obstacle positions). These trajectories are generated with 5% instantaneous safety criterion Pedram et al. (2021). The average risk estimate of 10^5 Monte Carlo simulations (run at a high rate of time discretization $r_d = 100$) is 0.27. The statistics of the discrete-time and continuous-time risk estimates are shown in Table 3.1. The discrete-time

risk estimates are computed using (1.6) at increasing rates of time discretization (r_d). The continuous-time risk estimates are computed using the method proposed by Ariu et al. Ariu et al. (2017) and our approach. The **Bias** and **RMSE** columns lists respectively the mean (signed) difference and the root mean squared difference between the corresponding estimate and the Monte Carlo estimate. The **% Conservative** column reports the percentage of cases where the corresponding estimate was greater than (or within 0.1% of) the Monte Carlo estimate and the **Avg. Time** lists the average computation times for our MATLAB implementations. From the data presented,

Table 3.1: Comparison of different risk estimates over 100 trajectories. Computation is performed in MATLAB on a consumer laptop.

Risk Estimates	**Avg. Time**	**Bias**	**RMSE**	**%Conservative**
Monte Carlo	101.50 s	0	0	-
Discrete-time				
$r_d : 5$	0.14 s	-0.14	0.18	28%
$r_d : 10$	0.26 s	-0.002	0.16	59%
$r_d : 20$	0.52 s	0.31	0.57	82%
$r_d : 55$	1.53 s	1.50	2.33	100%
$r_d : 100$	2.87 s	2.98	4.53	100%
Continuous-time				
Ariu et al. Ariu et al. (2017)	1.39 s	0.97	1.33	100%
Our 1^{st} order	1.47 s	0.66	0.90	100%
Our 2^{nd} order	2.23 s	0.28	0.36	100%

following conclusions can be drawn: First, our risk bounds require significantly less computation time than the Monte Carlo method. Second, unlike the discrete-time risk bounds at the lower sampling rates, our bounds remain conservative (i.e., safe) in all the trials. Lastly, our bounds produce tighter estimates than the discrete-time risk bounds at the higher sampling rates and the continuous-time risk bound of Ariu et al. (2017).

3.4 Conclusion

In this paper, we conducted an analysis to estimate the continuous-time collision probability of motion plans for autonomous agents with linear controlled Itô dynamics. We derived two upper bound for the continuous-time risk using the properties of Brownian motion (Markov property and reflection principle), and probability inequalities (Boole and Hunter's inequality). Our method boils down to computing probabilities at the discrete-time steps, simplifying the analysis, yet providing risk guarantees in continuous-time. We show that our bounds outperform the discrete-time risk bound (1.6) and are cheaper in computation than the naïve Monte Carlo sampling method.

Our analysis motivates a number of future investigations. This paper assumes that the robot follows a linear controlled Itô process. Future work will focus on risk analysis for systems with generalized stochastic dynamics. Another direction we would like to explore is risk analysis by fusing sampling-based methods and methods from continuous stochastic processes as suggested in Frey et al. (2020). This hybrid approach may provide the best of both worlds: high accuracy as well as computational simplicity and compatibility with continuous optimization.

Chapter 4: List of Publications

- **A. Patil**, T. Tanaka, "Upper and Lower Bounds for End-to-End Risks in Stochastic Robot Navigation," *2023 IFAC World Congress*
- **A. Patil**, T. Tanaka, "Upper Bounds for Continuous-Time End-to-End Risks in Stochastic Robot Navigation," *2022 European Control Conference (ECC)*

Appendix

Proof of (3.14)

$$\left(\bigcup_{t\in\mathcal{T}_j} \boldsymbol{x}^{sys}(t) \in \mathcal{H}_j^+\right) = \left(\bigcup_{t\in\mathcal{T}_j} a_j^T \boldsymbol{x}^{sys}(t) \geq b_j\right) = \left(\bigcup_{t\in\mathcal{T}_j} a_j^T \boldsymbol{x}(t) \geq b_j - a_j^T x^{plan}(t)\right). \tag{D.1}$$

Two equalities of (D.1) follow from (3.11) and (3.7) respectively. Now, recall that d_j is the minimum distance of $\mathcal{S}_j$ from $\mathcal{X}_{obs}$ i.e., $d_j = b_j - a_j^T y_2^*$, where y_2^* is the solution to the optimization problem (3.13). Noting that $b_j - a_j^T x^{plan}(t) \geq d_j$, $\forall\, t \in \mathcal{T}_j$, from (D.1),

$$\left(\bigcup_{t\in\mathcal{T}_j} \boldsymbol{x}^{sys}(t) \in \mathcal{H}_j^+\right) \subseteq \left(\bigcup_{t\in\mathcal{T}_j} a_j^T \boldsymbol{x}(t) \geq d_j\right).$$

Computation of $P(\mathcal{D}_j)$ and $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$

Let us define: $\Delta\hat{t}_j^i \coloneqq \hat{t}_j^{i+1} - \hat{t}_j^i$, and $\hat{\boldsymbol{x}}_j^i \coloneqq \boldsymbol{x}(\hat{t}_j^i)$. From (3.8), we can write

$$\hat{\boldsymbol{x}}_j^{i+1} = \hat{\boldsymbol{x}}_j^i + \hat{\boldsymbol{n}}_j^i, \qquad \hat{\boldsymbol{n}}_j^i \sim \mathcal{N}(0, \Sigma_{\hat{\boldsymbol{n}}_j^i}) \tag{D.2}$$

where $\hat{\boldsymbol{x}}_j^0 = \boldsymbol{x}_{j-1}$, $\Sigma_{\hat{\boldsymbol{n}}_j^i} \coloneqq \Delta\hat{t}_j^i R$, for $i = 0, 1, \ldots, r_j - 1$, and $j = 1, 2, \ldots, N$. Multiplying both sides of (D.2) by a_j^T we get

$$a_j^T \hat{\boldsymbol{x}}_j^{i+1} = \boldsymbol{z}_j^{i+1} = a_j^T \hat{\boldsymbol{x}}_j^i + a_j^T \hat{\boldsymbol{n}}_j^i, \qquad \hat{\boldsymbol{n}}_j^i \sim \mathcal{N}(0, \Sigma_{\hat{\boldsymbol{n}}_j^i}).$$

Stacking all $\boldsymbol{z}_j^i$ for $i = 0, 1, \ldots, r_j$, we can write the dynamics for the entire time segment $\mathcal{T}_j$ as

$$\boldsymbol{z}_j^{seg} = M_j \hat{\boldsymbol{x}}_j^0 + K_j \hat{\boldsymbol{n}}_j^{seg}, \qquad \hat{\boldsymbol{n}}_j^{seg} \sim \mathcal{N}(0, \Sigma_{\hat{\boldsymbol{n}}_j^{seg}}) \tag{D.3}$$

where $\boldsymbol{z}_j^{seg} \coloneqq \begin{bmatrix}\boldsymbol{z}_j^0 & \boldsymbol{z}_j^1 & \dots & \boldsymbol{z}_j^{r_j}\end{bmatrix}^T$, $M_j = a_j^T \cdot \mathbb{1}$,
$\hat{\boldsymbol{n}}_j^{seg} = \begin{bmatrix}\hat{\boldsymbol{n}}_j^0 & \hat{\boldsymbol{n}}_j^1 & \dots & \hat{\boldsymbol{n}}_j^{r_j-1}\end{bmatrix}^T$, $\Sigma_{\hat{\boldsymbol{n}}_j^{seg}} = \underset{0\le i\le r_j-1}{\text{diag}}\ \Sigma_{\hat{\boldsymbol{n}}_j^i}$,

$$K_j = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_j^T & 0 & \dots & 0 \\ a_j^T & a_j^T & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_j^T & a_j^T & \dots & a_j^T \end{bmatrix}$$

Computation of $P(\mathcal{D}_j)$:

In order to compute $P(\mathcal{D}_j)$, we need to find the distribution of $\boldsymbol{z}_j^{seg}$. Since $\hat{\boldsymbol{x}}_j^0 = \boldsymbol{x}_{j-1}$, it is distributed as $\hat{\boldsymbol{x}}_j^0 \sim \mathcal{N}(0, \Sigma_{\boldsymbol{x}_{j-1}})$. Hence, from (D.3), the p.d.f. of $\boldsymbol{z}_j^{seg}$ can be written as $\mu_{\boldsymbol{z}_j^{seg}}(z_j^{seg}) = \mathcal{N}(0, \Sigma_{\boldsymbol{z}_j^{seg}})$ where $\Sigma_{\boldsymbol{z}_j^{seg}} = M_j \Sigma_{\boldsymbol{x}_{j-1}} M_j^T + K_j \Sigma_{\hat{\boldsymbol{n}}_j^{seg}} K_j^T$. Now, $P(\mathcal{D}_j)$ can be computed as

$$P(\mathcal{D}_j) = \int_{\mathcal{C}_j} \mu_{\boldsymbol{z}_j^{seg}}(z_j^{seg})\, dz_j^{seg} \tag{D.4}$$

where $\mathcal{C}_j$ is a hypercube of dimension $r_j + 1$, having its sides along each direction run from $-\infty$ to d_j.

Computation of $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$:

Let us define $\boldsymbol{z}_{j,\,j+1}^{seg} \coloneqq \begin{bmatrix}\boldsymbol{z}_j^{seg} & \boldsymbol{z}_{j+1}^{seg}\end{bmatrix}^T$. In order to compute $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$, we need to find the distribution of $\boldsymbol{z}_{j,\,j+1}^{seg}$. First, let us write $\hat{\boldsymbol{x}}_{j+1}^0$ in terms of $\hat{\boldsymbol{x}}_j^0$.

$$\hat{\boldsymbol{x}}_{j+1}^0 = \hat{\boldsymbol{x}}_j^0 + G_j \hat{\boldsymbol{n}}_j^{seg} \tag{D.5}$$

where $G_j = \begin{bmatrix} I & I & \dots & I \end{bmatrix}_{n \times n r_j}$, $j = 1, 2, \dots, N-1$, and I is an $n \times n$ identity matrix. We know that

$$\boldsymbol{z}_{j+1}^{seg} = M_{j+1} \hat{\boldsymbol{x}}_{j+1}^0 + K_{j+1} \hat{\boldsymbol{n}}_{j+1}^{seg}. \tag{D.6}$$

Substituting $\hat{\boldsymbol{x}}_{j+1}^0$ from (D.5) in (D.6), we get

$$\boldsymbol{z}_{j+1}^{seg} = M_{j+1} \hat{\boldsymbol{x}}_j^0 + M_{j+1} G_j \hat{\boldsymbol{n}}_j^{seg} + K_{j+1} \hat{\boldsymbol{n}}_{j+1}^{seg}. \tag{D.7}$$

Let $H_{j,\,j+1} \coloneqq cov(\boldsymbol{z}_j^{seg},\ \boldsymbol{z}_{j+1}^{seg})$. Using (D.3) and (D.7), we can show that

$$H_{j,\,j+1} = M_j \Sigma_{\boldsymbol{x}_{j-1}} M_{j+1}^T + K_j \Sigma_{\hat{\boldsymbol{n}}_j^{seg}} G_j^T M_{j+1}^T. \tag{D.8}$$

For computing (D.8) we use the fact that

$$cov(\hat{\boldsymbol{x}}_j^0, \hat{\boldsymbol{n}}_j^{seg}) = cov(\hat{\boldsymbol{x}}_j^0, \hat{\boldsymbol{n}}_{j+1}^{seg}) = cov(\hat{\boldsymbol{n}}_j^{seg}, \hat{\boldsymbol{n}}_{j+1}^{seg}) = 0$$

Now, the p.d.f. of $\boldsymbol{z}_{j,\,j+1}^{seg}$ can be written as $\mu_{\boldsymbol{z}_{j,\,j+1}^{seg}}\left(z_{j,\,j+1}^{seg}\right) = \mathcal{N}(0, \Sigma_{\boldsymbol{z}_{j,\,j+1}^{seg}})$ where

$$\Sigma_{\boldsymbol{z}_{j,\,j+1}^{seg}} = \begin{bmatrix} \Sigma_{\boldsymbol{z}_j^{seg}} & H_{j,\,j+1} \\ H_{j,\,j+1}^T & \Sigma_{\boldsymbol{z}_{j+1}^{seg}} \end{bmatrix}$$

and $P(\mathcal{D}_j \cap \mathcal{D}_{j+1})$ can be computed as

$$P(\mathcal{D}_j \cap \mathcal{D}_{j+1}) = \int_{\mathcal{C}_j} \int_{\mathcal{C}_{j+1}} \mu_{\boldsymbol{z}_{j,\,j+1}^{seg}}\left(z_{j,\,j+1}^{seg}\right)\, dz_{j+1}^{seg} dz_j^{seg}. \tag{D.9}$$

MATLAB's `mvncdf` function can be utilized for computing (D.4) and (D.9) numerically.

Works Cited

Kaito Ariu, Cheng Fang, Marcio Arantes, Claudio Toledo, and Brian Williams. Chance-constrained path planning with continuous time safety guarantees. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics*, 26(3):502–517, 2010.

Lars Blackmore, Masahiro Ono, and Brian C Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6): 1080–1094, 2011.

József Bukszár and Andras Prekopa. Probability bounds with cherry trees. *Mathematics of Operations Research*, 26(1):174–192, 2001.

Albert Chern, Xiang Wang, Abhiram Iyer, and Yorie Nakahira. Safe control in the presence of stochastic uncertainties. *arXiv preprint arXiv:2104.01259*, 2021.

DA Dawson and D Sankoff. An inequality for probabilities. *Proceedings of the American Mathematical Society*, 18(3):504–507, 1967.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Kristoffer M Frey, Ted J Steiner, and J How. Collision probabilities for continuous-time systems without sampling. *RSS, Oregon, USA*, 2020.

Janos Galambos. Bonferroni inequalities. *The Annals of Probability*, pages 577–581, 1977.

Lucas Janson, Edward Schmerling, and Marco Pavone. Monte Carlo motion planning for robot trajectory optimization under uncertainty. In *Robotics Research*, pages 343–361. Springer, 2018.

Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

Peter E Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*. Berlin: Springer, 1992.

Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

Seymour M Kwerel. Bounds on the probability of the union and intersection of M events. *Advances in Applied Probability*, 7(2):431–448, 1975.

Steven M LaValle. *Planning algorithms*. Cambridge University Press, 2006.

Anirudha Majumdar and Russ Tedrake. Robust online motion planning with regions of finite time invariance. In *Algorithmic Foundations of Robotics X*, pages 543–558. Springer, 2013.

Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 37(6):1725–1740, 2014.

Peter Mörters and Yuval Peres. *Brownian motion*, volume 30. Cambridge University Press, 2010.

Kenshiro Oguri, Masahiro Ono, and Jay W McMahon. Convex optimization over sequential linear feedback policies with continuous-time chance constraints. In *IEEE 58th Conference on Decision and Control*, pages 6325–6331, 2019.

Masahiro Ono, Marco Pavone, Yoshiaki Kuwata, and J Balaram. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Autonomous Robots*, 39(4):555–571, 2015.

Apurva Patil and Takashi Tanaka. Upper and lower bounds for end-to-end risks in stochastic robot navigation. *arXiv preprint arXiv:2110.15879*, 2021.

Apurva Patil and Takashi Tanaka. Upper bounds for continuous-time end-to-end risks in stochastic robot navigation. In *ECC*, pages 2049–2055, 2022.

Apurva Patil, Alfredo Duarte, Aislinn Smith, Fabrizio Bisetti, and Takashi Tanaka. Chance-constrained stochastic optimal control via path integral and finite difference methods. In *CDC*, pages 3598–3604, 2022.

Sachin Patil, Jur Van Den Berg, and Ron Alterovitz. Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty. In *ICRA*, pages 3238–3244, 2012.

Ali Reza Pedram, Jeb Stefarr, Riku Funada, and Takashi Tanaka. Rationally inattentive path-planning via RRT*. In *ACC*, pages 3440–3446, 2021.

András Prékopa. Boole-Bonferroni inequalities and linear programming. *Operations Research*, 36:145–162, 1988.

András Prékopa. Probabilistic programming. *Handbooks in operations research and management science*, 10:267–351, 2003.

Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. A barrier function approach to finite-time stochastic system verification and control. *Automatica*, 125:109439, 2021.

Steven J Schwager. Bonferroni sometimes loses. *The American Statistician*, 38 (3):192–197, 1984.

Shridhar K Shah, Chetan D Pahlajani, and Herbert G Tanner. Probability of success in stochastic robot navigation with state feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3911–3916, 2011.

Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 1994.

Daniel Strawser and Brian Williams. Approximate branch and bound for fast, risk-bound stochastic path planning. In *ICRA*, pages 7047–7054, 2018.

Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

Shakiba Yaghoubi, Keyvan Majd, Georgios Fainekos, Tomoya Yamaguchi, Danil Prokhorov, and Bardh Hoxha. Risk-bounded control using stochastic barrier functions. *IEEE Control Systems Letters*, 5(5):1831–1836, 2020.

Hai Zhu and Javier Alonso-Mora. Chance-constrained collision avoidance for MAVs in dynamic environments. *IEEE Robotics and Automation Letters*, 4(2): 776–783, 2019.