Data-driven Polytopic Approximation for n-Dimensional Probabilistic Reachable Set

Pengcheng Wu, Jun Chen, *Member, IEEE*

***Abstract*—In this work, we first propose an algorithm to find a probabilistic reachable set that bounds system states given a prescribed confidence level. Then, we establish an optimization framework using mixed integer linear programming, where the solution identifies a convex polytope that approximates the probabilistic reachable set. Utilizing this formulation, we have devised a heuristic algorithm aimed at efficiently determining its solution without compromising significant accuracy. Through case studies, we have tested this heuristic algorithm, showcasing its simultaneous benefits in terms of efficiency, accuracy, near-optimality, and robustness. The positive outcomes of this research lay the foundation for potential applications in the real-time, safety-critical motion planning of dynamic systems under uncertainties.**

***Index Terms*—Probabilistic Reachable Set, Convex Approximation, Uncertain Dynamic System, Mixed Integer Linear Programming**

I. Introduction

How to guarantee safety plays a pivotal role in achieving successful operations for autonomous systems under uncertainties [1]–[4]. Reachability analysis is a popular set-based approach which computes reachable sets to guarantee the operational safety of uncertain dynamic systems [5]. In previous literature, many researchers have proposed different algorithms to leverage system details to compute reachable sets. However, such algorithms no longer apply when it comes to some challenges.

For an autonomous system under uncertainties not limited to bounded ones, it is usually impossible to absolutely avoid collision and guarantee operation safety. Alternatively, researchers tried to identify a trajectory for the uncertain dynamic system following which collision probability is bounded by an acceptable threshold. One method involves the evaluation of collision probability for a candidate trajectory [6], [7]. Nevertheless, a significant challenge in implementing this method is the absence of a closed-form expression for evaluating collision probability. Alternatively, another method involves the computation of a probabilistic reachable set (PRS) rather than a traditional reachable set [8]. The introduction of PRS can help convert probabilistic constraint which imposes an upper bound on collision probability, to a deterministic constraint of the PRS not intersecting with the candidate trajectory [9].

Pengcheng Wu is with the Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, and also with the Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182 pcwupat@ucsd.edu, pwu@sdsu.edu

Jun Chen is with the Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182 jun.chen@sdsu.edu

Researchers usually assumed Gaussian uncertainties in previous literature. However, this may not make sense in real scenarios [10], [11]. Instead, it is a natural idea to fit the non-Gaussian uncertainties into Gaussian ones. However, it is not accurate and may compromise operation safety [12]. Wu et al. applied the Monte Carlo method in the process of collision-free motion planning [13]. However, a large number of samples is usually required in pure sampling methods like Monte Carlo, which is infeasible and computationally inefficient in practice [14], [15]. Some researchers turned to employing the moment-based methods to capture non-Gaussian PRS [12]. However, it is not realistic to assume that the moment information is known *a priori*. Such limitations motivate the study of data-driven reachability analysis, which is dedicated to approximating reachable sets using data obtained from experiments or simulations. It combines the advantages of both sampling-based methods and analytical methods. In this paper, kernel density estimator (KDE), a data-driven approach, is used to capture unknown uncertainties not limited to Gaussian ones. The level set of KDE can be taken as the PRS of the states.

The shape of the probabilistic reachable set is usually irregular, which is not convenient in practice. For example, when it comes to collision detection, it is difficult to determine whether a point lies inside a probabilistic reachable set whose shape is irregular. Alternatively, researchers tried to find a convex polytopic approximation of PRS. Compared with an irregular probabilistic reachable set, it is much easier to determine the relative position between a point and a convex polytope. Different methods have been proposed to provide convex approximation. The bounding box is a popular way to find a convex approximation of PRS [16]. Hwang et al. provided a polytopic approximation for reachability analysis [17]. However, because of lacking quantification, they could not provide tight and accurate approximations of PRS. Wu et al. discovered an efficient zonotopic approximation for data-driven PRS; however, zonotopic approximation is not general when compared with polytopic approximation [9], [18]. Instead, Wu et al. also found convex polygons to serve as the approximation of PRS; however, its applicability is restricted to 2-dimensional spaces [19]. This study presents a novel approach where a mixed integer linear programming (MILP) optimization problem is constructed based on Kernel Density Estimation (KDE) outcomes. The solution to this problem determines a convex polytopic approximation for PRS in n-dimensional space. Additionally, a heuristic algorithm is devised to effectively tackle the MILP problem.

This paper focuses on offering an accurate and efficient convex approximation for PRS of uncertain dynamic systems

in n-dimensional space. The main contributions of this paper can be outlined as follows:

1) Formulating a MILP optimization problem, we aim to compute a convex polytopic approximation of the PRS in n-dimensional space. Unlike conventional methods that directly utilize data samples, our approach employs grid points weighted by the results of KDE, rendering it tractable. This strategy stands in contrast to existing methods like Gaussian fit and bounding box [18], [20], [21], yielding a convex polytopic approximation which is more tight and accurate. Consequently, this enhancement contributes to a planning space that is more feasible.

2) An efficient heuristic algorithm is devised to address the MILP problem. In contrast to the conventional approach, which seeks a globally optimal solution to the MILP problem formulated [19], [22], this algorithm can efficiently identify a near-optimal solution that is robust and accurate.

The remainder of this paper is organized as follows. In Section II, we formally formulate the problem under consideration. Section III presents the formulation of a MILP problem, the solution to which determines a convex polytopic approximation of the PRS. In Section IV, we introduce a heuristic algorithm devised to tackle the MILP problem. Case studies illustrating the feasibility and efficacy of the proposed heuristic algorithm are presented in Section V. Finally, Section VI provides a conclusion to this paper along with a future vision.

Some Notations:

$\mathbb{M} = \{1, \cdots, M\}$: the index set of M data samples

$\mathbb{N} = \{1, \cdots, N\}$: the index set of N grid points in each dimension

$\boldsymbol{x}_t \in \mathbb{R}^n$: the system state at time step t

$\boldsymbol{u}_t \in \mathbb{R}^m$: the control input at time step t

$\boldsymbol{\theta}_t \in \mathbb{R}^p$: the uncertain parameter

$\boldsymbol{w}_t \in \mathbb{R}^q$: the external disturbance

α: confidence level

N_o: the number of obstacles

$\|\cdot\|$: Euclidean norm

$\boldsymbol{x}_{it}$: the state of the i-th obstacle at time t

δ: the safe separation distance

$\mathcal{X}_t$: the PRS of the system

$\mathcal{X}_{it}$, $i \in \{1, \cdots, N_o\}$: the PRS of N_o obstacles

II. Problem Statement

In this section, we examine a discrete-time dynamic system under uncertainties and introduce the notion of an n-dimensional probabilistic reachable set (PRS) of the system state. Additionally, we outline the objective of this paper.

Consider a discrete-time dynamic system given by the form

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t, \boldsymbol{w}_t\right), \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the system state at time step t, $\boldsymbol{u}_t \in \mathbb{R}^m$ is the control input at time step t, the uncertain parameter $\boldsymbol{\theta}_t \in \mathbb{R}^p$ and the external disturbance $\boldsymbol{w}_t \in \mathbb{R}^q$ represent two random vectors obeying unknown probability distributions and they are independent and identically distributed across time. The initial condition $\boldsymbol{x}_0$ may obey an unknown probability distribution.

Collision avoidance plays a pivotal role in the operations of the system. It is not possible to figure out a bounded reachable set for unbounded uncertainties which can guarantee definite collision avoidance without violating the reachable set of the system state. Alternatively, we turn to guarantee that the collision probability is under a prescribed threshold $1-\alpha$, i.e.,

$$\forall t,\ \Pr\Big(\bigvee_{i=1}^{N_o} \|\boldsymbol{x}_t - \boldsymbol{x}_{it}\| \le \delta\Big) \le 1-\alpha, \tag{2}$$

where N_o represents the quantity of obstacles, $\|\cdot\|$ represents 2-norm, $\boldsymbol{x}_{it}$ represents the i-th obstacle's state at time t, δ represents the safe separation distance.

The equation in Eq. (2) represents a chance constraint, and our aim is to transform it into an equivalent deterministic constraint. Given that $\boldsymbol{x}_t$ and $\boldsymbol{x}_{it}, , i \in 1, \cdots, N_o$ are influenced by unknown uncertainties that could be unbounded, we seek to establish a Probabilistic Reachable Set (PRS) for each of them, individually. The formal definition of an n-dimensional PRS is provided as follows [8].

Definition 1 (Probabilistic Reachable Set)**.** At time t, given a confidence level α, a bounded set $\mathcal{X}_t \subseteq \mathbb{R}^n$ is an n-dimensional *probabilistic reachable set* (PRS) of the system state $\boldsymbol{x}_t \in \mathbb{R}^n$ in Eq. (1) if and only if

$$\Pr(\boldsymbol{x}_t \in \mathcal{X}_t) \ge \alpha.$$

With the help of PRS, we can convert Eq. (2) to

$$\forall t,\ \mathcal{X}_t \cap \bigcup_{i=1}^{N_o} \mathcal{X}_{it} = \emptyset.$$

The shape of the PRS $\mathcal{X}_t$ and $\mathcal{X}_{it}$, $i \in \{1, \cdots, N_o\}$ is usually nonconvex and irregular. In this paper, we are going to find a convex polytope serving the convex approximation of the PRS, which satisfies: i) *Convexity*: The polytope is convex; ii) *Efficiency*: The algorithm to find the polytope is computationally efficient; iii) *Accuracy*: The probability of the system state $\boldsymbol{x}_t$ in the polytope is close to the confidence level α; iv) *Optimality*: The hypervolume of the polytope is as small as possible (not too conservative) while ensuring accuracy.

III. MILP Formulation

We can use FFT-based KDE to capture an unknown n-variate probability distribution not limited to Gaussian ones [23]. We implement the algorithm proposed by [6] to compute a PRS of the system state in real-time. Please refer to the Online Appendix for more details of that algorithm.

In this section, we establish an optimization framework using mixed integer linear programming (MILP), where the solution determines a convex polytope to approximate the PRS.

A. Weight Assignment

Consider a mesh of N^n *grid points* $\{(x_{1i_1}, \cdots, x_{ni_n}) \in \mathbb{R}^n : i_1, \cdots, i_n \in \mathbb{N}\}$ evenly spaced on the space $\mathbb{R}^n$. The mesh of grid points can be viewed as a map $\boldsymbol{g} : \{(i_1, \cdots, i_n) :$

$i_1, \cdots, i_n \in \mathbb{N}\} \mapsto \{(x_{1i_1}, \cdots, x_{ni_n}) \in \mathbb{R}^n : i_1, \cdots, i_n \in \mathbb{N}\}$ given by

$$(x_{1i_1}, \cdots, x_{ni_n}) = \boldsymbol{g}(i_1, \cdots, i_n),\ i_1, \cdots, i_n \in \mathbb{N},$$

where $(i_1, \cdots, i_n)$ is the index of a grid point whose coordinate is $(x_{1i_1}, \cdots, x_{ni_n})$. Throughout the remainder of this paper, a coordinate $\boldsymbol{g}(i_1, \cdots, i_n)$ is abbreviated as $\boldsymbol{g}_{i_1 \cdots i_n}$.

Consider M data samples $\boldsymbol{x}_k = (x_{k1}, \cdots, x_{kn}) \in \mathbb{R}^n$, $k \in \mathbb{M}$ of a random vector $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$ and a mesh of N^n grid points $\boldsymbol{g}$ aligning with the axes of a global coordinate system. Evaluating KDE, it follows that each grid point $\boldsymbol{g}_{i_1 \cdots i_n}$, $i_1, \cdots, i_n \in \mathbb{N}$ has a binned KDE value $\tilde{f}_{i_1 \cdots i_n}$, $i_1, \cdots, i_n \in \mathbb{N}$, respectively. A normalized weight matrix $\boldsymbol{w} = [w_{i_1 \cdots i_n}]$ can be defined. Its element is

$$w_{i_1 \cdots i_n} = \frac{\tilde{f}_{i_1 \cdots i_n}}{\sum\limits_{i_1, \cdots, i_n = 1}^{N} \tilde{f}_{i_1 \cdots i_n}}, \quad i_1, \cdots, i_n \in \mathbb{N}. \tag{3}$$

B. Objective Function

The objective of the MILP optimization problem is to efficiently find a convex polytope which serves an accurate approximation of the PRS, through deciding which grid points should be selected to lie inside the polytope desired. To make it, we introduce three different kinds of decision variables:

- $2^n n$ continuous variables $a_{k1}, \cdots, a_{kn} \in \mathbb{R}$, $k \in \{1, \cdots, 2^n\}$;
- $2^n N^n$ binary variables $p^k_{i_1 \cdots i_n} \in \{0, 1\}$, $i_1, \cdots, i_n \in \mathbb{N}$, $k \in \{1, \cdots, 2^n\}$;
- N^n binary variables $z_{i_1 \cdots i_n} \in \{0, 1\}$, $i_1, \cdots, i_n \in \mathbb{N}$,

where n is the dimension of space, N is the number of grid points in each dimension, $a_{k1}, \cdots, a_{kn}$ are the coefficients of a hyperplane p^k which is the extension of a facet (also called hyperface) of the convex polytope, $p^k_{i_1 \cdots i_n} = 1$ if and only if $\boldsymbol{g}_{i_1 \cdots i_n}$ lies on the specified side of the hyperplane p^k, and $z_{i_1 \cdots i_n} = 1$ if and only if $\boldsymbol{g}_{i_1 \cdots i_n}$ lies within the polytope.

To get the smallest convex polytope by hypervolume, we formulate the objective function of the MILP problem as

$$\min \sum_{i_1, \cdots, i_n = 1}^{N} z_{i_1 \cdots i_n}. \tag{4}$$

C. Constraints on Continuous Variables

In an n-dimensional space $\mathbb{R}^n$, all the hyperplanes which do not pass through the origin are

$$\{a_1 x_1 + \cdots + a_n x_n = 1 : a_1, \cdots, a_n \in \mathbb{R} \wedge a_1^2 + \cdots + a_n^2 \neq 0\},$$

and different tuples of coefficients $(a_1, \cdots, a_n)$ determine different hyperplanes.

Given 2^n hyperplanes not passing through the origin in $\mathbb{R}^n$, we can define a set S determined by these hyperplanes as

$$S = \{(x_1, \cdots, x_n) \in \mathbb{R}^n : \bigwedge_{k=1}^{2^n} a_{k1} x_1 + \cdots + a_{kn} x_n \leq 1\}, \tag{5}$$

which is a convex polyhedron containing the origin. Alternatively, Eq. (5) can be concisely written as

$$S = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{1}\}, \tag{6}$$

where $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$, $\boldsymbol{1} = (1, \cdots, 1) \in \mathbb{R}^{2^n}$, and $\boldsymbol{A} = \begin{bmatrix} a, & \cdots, & a_{1n} \\ a_{2^n 1}, & \cdots, & a_{2^n n} \end{bmatrix} \in \mathbb{R}^{2^n \times n}$.

The set S given in Eq. (6) must contain the origin, but may be unbounded. If bounded, S is a convex polytope containing the origin. The following lemma introduced in [24] gives the condition under which S must be bounded.

Lemma 1. *The convex polyhedron S in Eq. (6) is unbounded if and only if $\exists \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0} \wedge \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{0}$.*

Next, the following theorem gives the constraints on the coefficients of the hyperplanes, which suffices to ensure the convex polyhedron S given in Eq. (6) is bounded.

Theorem 1. *Given 2^n hyperplanes $p^k : a_{k1} x_1 + \cdots + a_{kn} x_n = 1$, $k \in \{1, \cdots, 2^n\}$ which do not pass through the origin in $\mathbb{R}^n$, if the coefficients of these hyperplanes satisfy*

$$\begin{gathered} a_{11} > 0,\ a_{12} > 0,\ \cdots,\ a_{1n} > 0; \\ a_{21} < 0,\ a_{22} > 0,\ \cdots,\ a_{2n} > 0; \\ \vdots \\ a_{2^n 1} < 0,\ a_{2^n 2} < 0,\ \cdots,\ a_{2^n n} < 0, \end{gathered} \tag{7}$$

then the convex polyhedron S given in Eq. (6) is bounded, namely, S is a convex polytope containing the origin.

Proof. See Online Appendix. ■

That S in Eq. (6) is bounded doesn't necessarily mean the quantity of the hyperfaces (also called facets) of the convex polytope S is exactly 2^n. Indeed, the number of the facets of S is $\leq 2^n$.

Theorem 1 shows that the convex polytope S determined by the hyperplanes whose coefficients satisfy Eq. (7) must contain the origin. The following corollary applies to the scenarios where an arbitrary point must be contained.

Corollary 1. *Given 2^n hyperplanes $p^k : a_{k1}(x_1 - \hat{x}_1) + \cdots + a_{kn}(x_n - \hat{x}_n) = 1$, $k \in \{1, \cdots, 2^n\}$ not passing through a point $(\hat{x}_1, \cdots, \hat{x}_n) \in \mathbb{R}^n$, if these hyperplanes' coefficients satisfy Eq. (7), then the convex polyhedron S given by*

$$S = \{(x_1, \cdots, x_n) \in \mathbb{R}^n : \bigwedge_{k=1}^{2^n} a_{k1}(x_1 - \hat{x}_1) + \cdots + a_{kn}(x_n - \hat{x}_n) \leq 1\} \tag{8}$$

is bounded, namely, S is a convex polytope containing the point $(\hat{x}_1, \cdots, \hat{x}_n)$.

Proof. See Online Appendix. ■

In summary, Eq. (7) gives the constraints on $2^n n$ continuous variables $a_{k1}, \cdots, a_{kn} \in \mathbb{R}$, $k \in \{1, \cdots, 2^n\}$. Given 2^n hyperplanes not passing through a point in $\mathbb{R}^n$, if Eq. (7) holds, then S in Eq. (8) is a convex polytope containing that point.

D. Constraints of Discrete Variables

We denote the index of the grid point which has the greatest normalized weight by $(\hat{i}_1, \cdots, \hat{i}_n) = \arg\max(\boldsymbol{w})$, and its coordinate by $(\hat{x}_1, \cdots, \hat{x}_n) = \boldsymbol{g}_{\hat{i}_1 \cdots \hat{i}_n}$. We enforce the grid point $\boldsymbol{g}_{\hat{i}_1 \cdots \hat{i}_n}$ lies within the convex polytope. In other words,

$$z_{\hat{i}_1 \cdots \hat{i}_n} = 1. \tag{9}$$

The purpose of the convex polytope is to provide an approximation of the PRS. To ensure a confidence level α of the PRS, we enforce that the normalized weights' sum of the grid points lying within the polytope exceeds α. That is,

$$\sum_{i_1, \cdots, i_n = 1}^{N} w_{i_1 \cdots i_n} z_{i_1 \cdots i_n} \geq \alpha. \tag{10}$$

Eq. (9) and Eq. (10) are what we establish for $z_{i_1 \cdots i_n} \in \{0, 1\},\ i_1, \cdots, i_n \in \mathbb{N}$.

We enforce $z_{i_1 \cdots i_n} = 1$ if and only if $\boldsymbol{g}_{i_1 \cdots i_n}$ lies within the convex polytope. That is,

$$z_{i_1 \cdots i_n} = 1 \implies \bigwedge_{k=1}^{2^n} a_{k1}(x_{1 i_1} - \hat{x}_1) + \cdots + a_{kn}(x_{n i_n} - \hat{x}_n) \leq 1,\ i_1, \cdots, i_n \in \mathbb{N}, \tag{11}$$

$$z_{i_1 \cdots i_n} = 0 \implies \bigvee_{k=1}^{2^n} a_{k1}(x_{1 i_1} - \hat{x}_1) + \cdots + a_{kn}(x_{n i_n} - \hat{x}_n) > 1,\ i_1, \cdots, i_n \in \mathbb{N}. \tag{12}$$

Next, a new type of binary variables $p^k_{i_1 \cdots i_n} \in \{0, 1\},\ i_1, \cdots, i_n \in \mathbb{N}, k \in \{1, \cdots, 2^n\}$ is introduced to transform the logical constraints to algebraic constraints. We enforce $p^k_{i_1 \cdots i_n} = 1$ if and only if $\boldsymbol{g}_{i_1 \cdots i_n}$ lies on the same side as the point $(\hat{x}_1, \cdots, \hat{x}_n)$ of the hyperplane p^k. That is,

$$p^k_{i_1 \cdots i_n} = 1 \implies a_{k1}(x_{1 i_1} - \hat{x}_1) + \cdots + a_{kn}(x_{n i_n} - \hat{x}_n) \leq 1,\ i_1, \cdots, i_n \in \mathbb{N},\quad k \in \{1, \cdots, 2^n\},$$

$$p^k_{i_1 \cdots i_n} = 0 \implies a_{k1}(x_{1 i_1} - \hat{x}_1) + \cdots + a_{kn}(x_{n i_n} - \hat{x}_n) > 1,\ i_1, \cdots, i_n \in \mathbb{N},\quad k \in \{1, \cdots, 2^n\}.$$

Further, the constraints above can be reformulated as algebraic constraints using big-M method [25], which are

$$a_{k1}(x_{1 i_1} - \hat{x}_1) + \cdots + a_{kn}(x_{n i_n} - \hat{x}_n) - 1 \leq M_{\text{big}}(1 - p^k_{i_1 \cdots i_n}),\quad i_1, \cdots, i_n \in \mathbb{N},\ k \in \{1, \cdots, 2^n\}, \tag{13}$$

$$-a_{k1}(x_{1 i_1} - \hat{x}_1) - \cdots - a_{kn}(x_{n i_n} - \hat{x}_n) + 1 < M_{\text{big}} p^k_{i_1 \cdots i_n},\quad i_1, \cdots, i_n \in \mathbb{N},\ k \in \{1, \cdots, 2^n\}, \tag{14}$$

where M_{big} is a constant which is large enough.

The logical connection between $p^k_{i_1 \cdots i_n}$ and $z_{i_1 \cdots i_n}$ is

$$z_{i_1 \cdots i_n} = 1 \implies \sum_{k=1}^{2^n} p^k_{i_1 \cdots i_n} = 2^n,\quad i_1, \cdots, i_n \in \mathbb{N},$$

$$z_{i_1 \cdots i_n} = 0 \implies \sum_{k=1}^{2^n} p^k_{i_1 \cdots i_n} \leq (2^n - 1),\quad i_1, \cdots, i_n \in \mathbb{N},$$

and these constraints can be further reformulated as the following algebraic constraints

$$\sum_{k=1}^{2^n} p^k_{i_1 \cdots i_n} \geq 2^n z_{i_1 \cdots i_n},\quad i_1, \cdots, i_n \in \mathbb{N}, \tag{15}$$

$$\sum_{k=1}^{2^n} p^k_{i_1 \cdots i_n} \leq (2^n - 1) + z_{i_1 \cdots i_n},\quad i_1, \cdots, i_n \in \mathbb{N}. \tag{16}$$

E. Formulation of MILP Optimization Problem

We have established the objective function $\min \sum_{i_1, \cdots, i_n = 1}^{N} z_{i_1 \cdots i_n}$ in Eq. (4), and introduced the following three types of decision variables. Collecting the formulated constraints Eq. (7), Eq. (9), Eq. (10), Eq. (13), Eq. (14), Eq. (15), Eq. (16), we formulate the problem proposed in Section II as an MILP problem Eq. (17) on Page 5.

To determine the convex polytope S as defined in Eq. (8), we need to figure out the coefficients $a_{k1}, \cdots, a_{kn} \in \mathbb{R},\ k \in \{1, \cdots, 2^n\}$. They can be found through solving the above MILP problem.

$$
\begin{aligned}
\min \quad & \sum_{i_1,\cdots,i_n=1}^{N} z_{i_1\cdots i_n} \\
\text{s.t.} \quad & a_{11}>0,\ a_{12}>0,\ \cdots,\ a_{1n}>0, \\
& a_{21}<0,\ a_{22}>0,\ \cdots,\ a_{2n}>0, \\
& \qquad\qquad \vdots \\
& a_{2^n1}<0,\ a_{2^n2}<0,\ \cdots,\ a_{2^nn}<0; \\
& a_{k1}(x_{1i_1}-\hat{x}_1)+\cdots+a_{kn}(x_{ni_n}-\hat{x}_n)-1 \le \\
& M_{\text{big}}(1-p^k_{i_1\cdots i_n}), \quad i_1,\cdots,i_n\in\mathbb{N},\ k\in\{1,\cdots,2^n\}; \\
& -a_{k1}(x_{1i_1}-\hat{x}_1)-\cdots-a_{kn}(x_{ni_n}-\hat{x}_n)+1< \\
& M_{\text{big}}p^k_{i_1\cdots i_n}, \quad i_1,\cdots,i_n\in\mathbb{N},\ k\in\{1,\cdots,2^n\}; \\
& \sum_{k=1}^{2^n} p^k_{i_1\cdots i_n} \ge 2^n z_{i_1\cdots i_n}, \quad i_1,\cdots,i_n\in\mathbb{N}; \\
& \sum_{k=1}^{2^n} p^k_{i_1\cdots i_n} \le (2^n-1)+z_{i_1\cdots i_n}, \quad i_1,\cdots,i_n\in\mathbb{N}; \\
& z_{\hat{i}_1\cdots\hat{i}_n}=1; \\
& \sum_{i_1,\cdots,i_n=1}^{N} w_{i_1\cdots i_n} z_{i_1\cdots i_n} \ge \alpha; \\
& a_{k1},\cdots,a_{kn}\in\mathbb{R}, \quad k\in\{1,\cdots,2^n\}; \\
& p^k_{i_1\cdots i_n}\in\{0,1\}, \quad i_1,\cdots,i_n\in\mathbb{N},\ k\in\{1,\cdots,2^n\}; \\
& z_{i_1\cdots i_n}\in\{0,1\}, \quad i_1,\cdots,i_n\in\mathbb{N}.
\end{aligned} \tag{17}
$$

F. Convex Approximation through Transformation

The constraints Eq. (7) on the coefficients of the hyperplanes of a convex polytope enforce that different hyperplanes are located at different orthants (also called hyperoctants) of a coordinate system. Therefore, the convex polytope which is determined by the solution to the MILP problem Eq. (17) is dependent on the selection of the coordinate system. Instead of the original global coordinate system, we hope to establish a new global coordinate system, especially one that aligns with the principal axes of the original data samples, to help determine the convex polytope of desired.

Theorem 2. *Consider a random vector $\boldsymbol{x}=(x_1,\cdots,x_n)\in\mathbb{R}^n$ following an unknown n-variate probability distribution with M data samples $\boldsymbol{x}_k=(x_{k1},x_{k2},\cdots,x_{kn})\in\mathbb{R}^n$, $k\in\mathbb{M}$ extracted from this distribution. Let $\boldsymbol{e}_1,\cdots,\boldsymbol{e}_n\in\mathbb{R}^n$ denote n unit vectors indicating the direction of the principal axes of these M data samples. Define $\boldsymbol{u}=(u_1,\cdots,u_n)\in\mathbb{R}^n$ as a new random vector obtained through a special affine transformation of $\boldsymbol{x}$, which is*

$$
\boldsymbol{u}=\boldsymbol{T}(\boldsymbol{x})=\bar{\boldsymbol{x}}+\boldsymbol{A}(\boldsymbol{x}-\bar{\boldsymbol{x}}),\ \boldsymbol{x}\in\mathbb{R}^n, \tag{18}
$$

where $\boldsymbol{A}=[\boldsymbol{e}_1,\cdots,\boldsymbol{e}_n]^{-1}$ is an orthogonal matrix and $\bar{\boldsymbol{x}}=(\bar{x}_1,\cdots,\bar{x}_n)=\frac{1}{M}\sum_{k=1}^{M}\boldsymbol{x}_k=\frac{1}{M}\sum_{k=1}^{M}(x_{k1},\cdots,x_{kn})$ is a fixed point under $\boldsymbol{T}$. Then we determine a convex polytopic approximation

$$
\boldsymbol{T}^{-1}(S)=\left\{\boldsymbol{T}^{-1}(\boldsymbol{u})\in\mathbb{R}^n:\boldsymbol{u}\in S\right\} \tag{19}
$$

of PRS for the random vector $\boldsymbol{x}$, which depends on its principal axes. Here S, as defined in Eq. (8), is a convex polytope determined by the solution to a MILP optimization problem Eq. (17) formulated for the random vector $\boldsymbol{u}$.

Proof. See Online Appendix. ■

IV. Solution Method

In this section, we develop an algorithm of Heuristic MILP which can efficiently find a near-optimal solution to the formulated MILP optimization problem.

Throughout the rest of this paper, we refer to the way of solving the MILP problem Eq. (17) through the algorithm of cutting planes and branching, which is a standard solver of MILP problems, as *Optimal MILP algorithm* [22]. The optimal solution found in this way determines an optimal convex polytopic approximation of the PRS. However, Optimal MILP algorithm is often intractable. Instead, we develop Algorithm 1, *Heuristic MILP algorithm*, which can efficiently find a near-optimal solution of $a_{k1},\cdots,a_{kn},\ k\in\{1,\cdots,2^n\}$, instead of an optimal solution found by Optimal MILP algorithm. This near-optimal solution determines a near-optimal convex polytopic approximation of the PRS.

Algorithm 1 conducts weighted sampling to choose some grid points $\boldsymbol{g}'_{\boldsymbol{u}}$ as representatives from the entire set of grid points $\boldsymbol{g}_{\boldsymbol{u}}$. Then, we can formulate a revised MILP problem using $\boldsymbol{g}'_{\boldsymbol{u}}$. Subsequently, the Optimal MILP can be utilized to solve the revised MILP problem with $\boldsymbol{g}'_{\boldsymbol{u}}$. The optimal solution obtained from this new MILP problem utilizing $\boldsymbol{g}'_{\boldsymbol{u}}$ serves as an approximation to the optimal solution of the original MILP problem using $\boldsymbol{g}_{\boldsymbol{u}}$. This trade-off between efficiency and optimality arises due to the reduction in the size of $\boldsymbol{g}'_{\boldsymbol{u}}$ compared to $\boldsymbol{g}_{\boldsymbol{u}}$, leading to a decrease in the number of constraints and decision variables in the new MILP problem, thus significantly reducing computational time.

In Algorithm 1, we denote N_s as the count of $\boldsymbol{g}'_{\boldsymbol{u}}$. In lines 1 through 6, we employ the AES algorithm, given in [26] to select N_s $\boldsymbol{g}'_{\boldsymbol{u}}$ from N^n $\boldsymbol{g}_{\boldsymbol{u}}$ without replacement randomly, based on their normalized weights $\boldsymbol{w}$. Subsequently, from lines 7 through 9, we re-normalize the original weights $\boldsymbol{w}'$ of $\boldsymbol{g}'_{\boldsymbol{u}}$ to $\hat{\boldsymbol{w}}'$, ensuring their sum equals one. This step of re-normalization plays a pivotal role to ensure the solution derived from Algorithm 1 is accurate.

V. Main Results

In this section, we undertake case studies to compare the efficacy of Heuristic MILP algorithm with three other algorithms: Optimal MILP, Gaussian fit [9], and the bounding box method for PRS approximation [16]. Each algorithm aims to provide a convex set for approximating the PRS. The

Algorithm 1 Heuristic MILP Algorithm

function RESAMP($\boldsymbol{g_u}$, $\boldsymbol{w}$, N_s)
  $u_{i_1\cdots i_n} = \text{random}(0,1)$
  $k_{i_1\cdots i_n} = u_{i_1\cdots i_n} \frac{1}{w_{i_1\cdots i_n}}$
  Pick up first N_s grids from $\boldsymbol{g_u}$ in descending order of $k_{i_1\cdots i_n}$ as $\boldsymbol{g'_u}$
  Pick up N_s grids' weights from $\boldsymbol{w}$ as $\boldsymbol{w}'$
  return $\boldsymbol{w}'$, $\boldsymbol{g'_u}$

function RENORM($\boldsymbol{w}'$, N_s)
  $\hat{\boldsymbol{w}}' = \text{ones}(N_s) \cdot (1 - \text{sum}(\boldsymbol{w}'))/N_s + \boldsymbol{w}'$
  return $\hat{\boldsymbol{w}}'$

function GUROBI(α, $\boldsymbol{g_u}$, $\boldsymbol{g'_u}$, $\hat{\boldsymbol{w}}'$)
  $\hat{i}_1, \cdots, \hat{i}_n = \arg\max(\hat{\boldsymbol{w}}')$
  $(\hat{x}_1, \cdots, \hat{x}_n) = \boldsymbol{g_u}_{\hat{i}_1\cdots\hat{i}_n}$
  Establish Eq. (17) on $\boldsymbol{g'_u}$ weighted by $\hat{\boldsymbol{w}}'$ at confidence level α
  Use Gurobi to solve Eq. (17) for $a_{k1}, \cdots, a_{kn}$, $k \in \{1, \cdots, 2^n\}$
  return $\hat{x}_1, \cdots, \hat{x}_n$, $a_{k1}, \cdots, a_{kn}$, $k \in \{1, \cdots, 2^n\}$

function HEURISTICPOLYTOPE(α, $\boldsymbol{x}_k$, N_s)
  Get n orthonormal vectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n$ representing the orientation of M data samples $\boldsymbol{x}_k$
  Get N^n grid points $\boldsymbol{g_x}$ aligning with $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n$
  $\boldsymbol{A} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n]^{-1}$
  $\bar{\boldsymbol{x}} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{x}_k$
  Get a new random vector $\boldsymbol{u} = \boldsymbol{T}(\boldsymbol{x}) = \bar{\boldsymbol{x}} + \boldsymbol{A}(\boldsymbol{x} - \bar{\boldsymbol{x}})$, $\boldsymbol{x} \in \mathbb{R}^n$
  Get N^n grid points $\boldsymbol{g_u} = \boldsymbol{T} \circ \boldsymbol{g_x}$ aligning with the axes of the coordinate system
  Evaluate KDE $\hat{f}_{\boldsymbol{u}}$ on $\boldsymbol{g_u}$ and get the normalized weight matrix $\boldsymbol{w}$
  $\boldsymbol{g'_u}$, $\boldsymbol{w}'$ = RESAMP($\boldsymbol{g_u}$, $\boldsymbol{w}$, N_s)
  $\hat{\boldsymbol{w}}'$ = RENORM($\boldsymbol{w}'$, N_s)
  $\hat{x}_1, \cdots, \hat{x}_n$, $a_{k1}, \cdots, a_{kn}$, $k \in \{1, \cdots, 2^n\}$ = GUROBI(α, $\boldsymbol{g_u}$, $\boldsymbol{g'_u}$, $\hat{\boldsymbol{w}}'$)
  Determine the heuristic polytope S for $\boldsymbol{u}$ by the solution $\hat{x}_1, \cdots, \hat{x}_n$, $a_{k1}, \cdots, a_{kn}$, $k \in \{1, \cdots, 2^n\}$
  Determine the heuristic polytope $\boldsymbol{T}^{-1}(S)$ for $\boldsymbol{x}$
  return $\boldsymbol{T}^{-1}(S)$

HEURISTICPOLYTOPE(α, $\boldsymbol{x}_k$, N_s)

experiments were conducted using Python 3.9 on an Intel(R) Core(TM) i9-12900KF processor running at 3187 Mhz, with 16 physical cores and 24 logical processors, on a desktop equipped with 64GB of RAM.

A. Case Settings

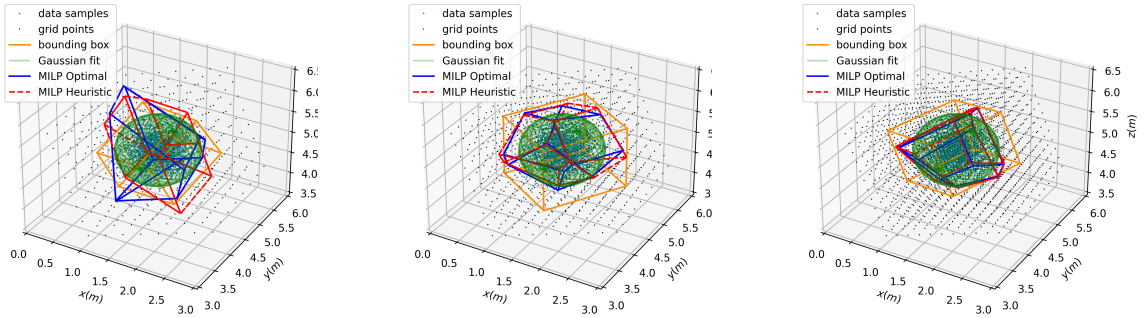
We consider Astrobee, a free-flyer robot with six degrees of freedom operating on the International Space Station, subject to uncertainties [27]. Its state is $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}) \in \mathbb{R}^{13}$, input is $\boldsymbol{u} = (\boldsymbol{F}, \boldsymbol{M}) \in \mathbb{R}^6$, and dynamics are $\dot{\boldsymbol{p}} = \boldsymbol{v}$, $m\dot{\boldsymbol{v}} = \boldsymbol{F}$, $\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q}$, $\boldsymbol{J}\dot{\boldsymbol{\omega}} = \boldsymbol{M} - \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{J}\boldsymbol{\omega}$ where $\boldsymbol{J} = \text{diag}([J_x, J_y, J_z])$. We discretize the dynamics with $\Delta t = 1\,\text{s}$ such that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k)\Delta t + \boldsymbol{w}_k$ where $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_w})$ are i.i.d. disturbances. The mass and inertia are unknown with known bounds $m \in [7.1, 7.3]$, $J_i \in [0.065, 0.075]$, $|w_{ki}| \leq 10^{-4}$ for $i = 1, \cdots, 13$, and $|w_{ki}| \leq 5 \times 10^{-4}$ for $i = 4, 5, 6$. We will show the projection $\boldsymbol{p} \in \mathbb{R}^3$ of $\boldsymbol{x}$ at $k = 20\,\text{s}$ later.

In addition, a common constant 10^4 is taken for the parameter M_{big} for big-M. The dataset comprises $M = 1000$ samples. The baseline setup encompasses a confidence level of $\alpha = 95\%$, with Optimal MILP utilizing $N^n = 10^3$ grid points and Heuristic MILP using $N_s = 200$ grid points. In the subsequent figures and tables, the baseline configuration is denoted by the symbol "*".

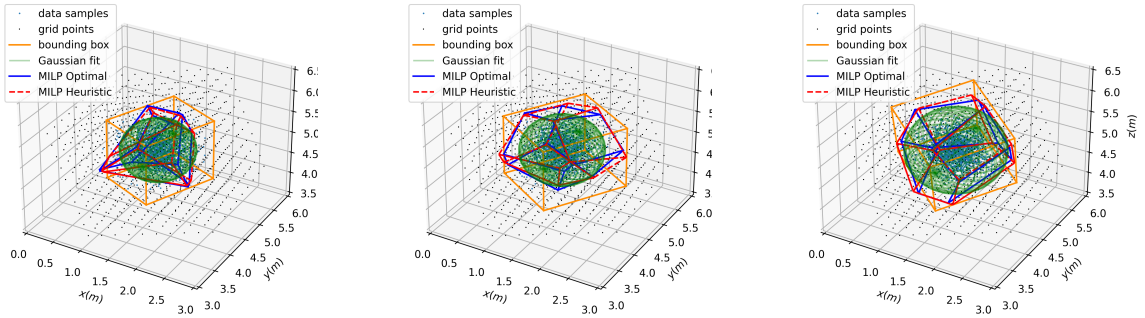
B. Impact of Different Numbers of Grid Points

In this part, we undertake a performance evaluation of various algorithms across different grid point configurations. Optimal MILP employs N^n grid points, while Heuristic MILP utilizes N_s. The variations in the quantity of grid points are represented by the ratios N^n/N_s, specifically $8^3/150$, $10^3/200$, and $12^3/250$. All other parameters remain unchanged from the baseline configuration.

The results are displayed in Fig. 1a, Fig. 1b, Fig. 1c, and Table I. The outcomes yielded by various algorithms exhibit disparities. Notably, there can be a substantial distinction between the convex polytopes derived from Optimal MILP and those from Heuristic MILP, particularly noticeable when the quantity of grid points is limited, as illustrated in Fig. 1a. This divergence arises due to Heuristic MILP typically achieving a near-optimal solution, whereas Optimal MILP ensures an optimal one. A smaller gap means a more accurate ratio. The volume of the polytope obtained by Optimal MILP or Heuristic MILP is usually much smaller than bounding box or Gaussian fit, and yet the ratio of the polytope of Optimal MILP or Heuristic MILP owns better accuracy.For example, when $N^n/N_s = 10^3/200$, the gap 0.8% of Optimal MILP or 0.9% of Heuristic MILP is smaller than 3.2% of bounding box. This supports that Optimal MILP or Heuristic MILP enjoys accuracy. For each algorithm, as the quantity of grid points increases from $8^3/150$ to $12^3/250$, the performance in terms of ratio and volume improves. For example, the gap 0.4% (resp. 0.7%) of Optimal MILP (resp. Heuristic MILP) is smaller than 3.8% (resp. 4.6%) of Optimal MILP (resp. Heuristic MILP). The number of grid points greater than 12^3 does not make significant difference in terms of ratio and volume. However, with Optimal MILP, the increase in computational time is notable due to the augmented quantity of constraints and decision variables resulting from the expanding grid point count, escalating from $1.90\,\text{s}$ to $2077\,\text{s}$. Consequently, the pursuit of optimality with Optimal MILP is accompanied by a sacrifice in computational efficiency. Conversely, the computational time for Heuristic MILP exhibits a gradual increase, from $0.08\,\text{s}$ to $1.21\,\text{s}$, while the resulting polytope closely resembles that obtained by Optimal MILP. These findings indicate that despite the compromise in optimality, Heuristic MILP ensures near-optimal solutions alongside accuracy and efficiency. Compared



(a) # grid points = $8^3/150$ (b) # grid points = $10^3/200$ * (c) # grid points = $12^3/250$



(d) confidence level = 90% (e) confidence level = 95% * (f) confidence level = 99%

Fig. 1. Impact of different quantities of grid points N^n/N_s and different confidence levels α

TABLE I
IMPACT OF OF DIFFERENT QUANTITIES OF GRID POINTS N^n/N_s AND DIFFERENT CONFIDENCE LEVELS α

Algorithm	**Impact of # Grid Points N^n/N_s**				**Impact of Different Confidence Levels α**			
	# Grid Points	**Ratio/Gap**	**Volume** (m^3)	**Time** (s)	**Confidence Level**	**Ratio/Gap**	**Volume** (m^3)	**Time** (s)
Optimal MILP	$8^3/150$	91.2%/3.8%	1.41	1.90	90%	89.5%/5.5%	1.03	93.2
Heuristic MILP		90.4%/4.6%	1.31	0.08		90.0%/5.0%	1.08	0.74
Gaussian Fit		95.1%/0.1%	1.38	0.02		90.1%/4.9%	0.99	0.02
Bounding Box		87.9%/7.1%	1.18	0.01		90.9%/4.1%	1.51	0.01
Optimal MILP	$10^3/200$*	95.8%/0.8%	1.35	167.2	95%*	95.8%/0.8%	1.35	167.2
Heuristic MILP		95.9%/0.9%	1.36	0.60		95.9%/0.9%	1.36	0.60
Gaussian Fit		95.1%/0.1%	1.38	0.02		95.1%/0.1%	1.38	0.02
Bounding Box		98.2%/3.2%	2.40	0.01		98.2%/3.2%	2.40	0.01
Optimal MILP	$12^3/250$	95.4%/0.4%	1.27	2077	99%	98.7%/3.7%	1.99	377.4
Heuristic MILP		95.7%/0.7%	1.33	1.21		98.9%/3.9%	2.07	0.56
Gaussian Fit		95.1%/0.1%	1.38	0.02		99.1%/4.1%	2.42	0.02
Bounding Box		97.5%/2.5%	2.14	0.01		99.5%/4.5%	3.67	0.01

with Gaussian Fit and Bounding Box algorithms, Heuristic MILP can obtain results which are near-optimal, tighter, and accurate. Although it runs slower than Gaussian Fit and Bounding Box, the expense in terms of computational time is acceptable while greatly improving performance in near-optimality, tightness, and accuracy. Thus, Heuristic MILP surpasses the other three algorithms by effectively balancing computational efficiency and accuracy.

The polytope obtained by MILP Optimal is optimal in the sense that it has a minimum volume without violating constraints. However, this does not mean the volume of MILP Heuristic must be greater than MILP Optimal. For example, when grid points are $8^3/150$, the volume obtained by Heuristic MILP algorithm (e.g. $1.31\,\mathrm{m}^3$) is indeed less than that of MILP Optimal (e.g. $1.41\,\mathrm{m}^3$). The polytope derived from Heuristic MILP serves as the optimal solution to a novel MILP problem utilizing the N_s representative grid points. Given the disparate grid configurations between the original and new MILP problems, there exists a possibility that the hypervolume of the polytope generated by Heuristic MILP is smaller than that of Optimal MILP.

C. Impact of Different Confidence Levels

In this part, we conduct a performance evaluation of different algorithms across three distinct confidence levels: $\alpha = 90\%$, 95%, and 99%. All other parameters remain unchanged from the baseline configuration.

As depicted in Fig. 1 and Table I, when the confidence level is constant, both Optimal MILP and Heuristic MILP yield tighter polytopes compared to the results of the bounding box and Gaussian fit. Moreover, the ratios achieved by Optimal MILP or Heuristic MILP are more accurate. For instance, at $\alpha = 95\%$, the volume is $1.35\,\mathrm{m}^3$ for the polytope derived from Optimal MILP, or $1.36\,\mathrm{m}^3$ for the polytope obtained from Heuristic MILP, which are both smaller than the volume of $1.38\,\mathrm{m}^3$ for the ellipsoid from Gaussian fit and $2.40\,\mathrm{m}^3$ for the bounding box. Additionally, the disparity between the ratio (95.8%) and the confidence level (95%) for Optimal MILP or 95.9% for Heuristic MILP) is notably narrower than the difference between $\alpha = 95\%$ and 98.2% for the bounding box. Therefore, both Optimal MILP and Heuristic MILP outperform bounding box and Gaussian fit from the aspects of accuracy and optimality (or near-optimality), as they generate polytopes that offer a convex approximation for the PRS.

As the confidence level rises, the volume of the polytope derived from Optimal MILP or Heuristic MILP also grows, with computational time exhibiting slight fluctuations. Maintaining a constant confidence level, the volume of the polytope of Heuristic MILP closely aligns with that of Optimal MILP. However, Heuristic MILP significantly beats Optimal MILP in the sense that Heuristic MILP is more computationally efficient. For instance, at a confidence level of 95%, the computational time is $0.60\,\mathrm{s}$ for Heuristic MILP, considerably shorter than the $167.2\,\mathrm{s}$ required by Optimal MILP, while the volumes are $1.35\,\mathrm{m}^3$ and $1.36\,\mathrm{m}^3$ respectively, with ratios of 95.8% and 95.9%. Thus, Heuristic MILP offers computational efficiency without sacrificing accuracy.

D. Polytopic Approximation versus Convex Hull

We also compared the method proposed in this paper with the method of Convex Hull. The parameters are: the number of grid points N^n/N_s = $10^3/200$, and the confidence level α = 95%.

Our proposed algorithm of Heuristic MILP beats the method of convex hull in terms of convenience while still enjoying accuracy and computational efficiency. For instance, Table II shows that the polytope obtained by Heuristic MILP is better than convex hull in the sense that the ratio / volume of Heuristic MILP 95.9% / $1.36\,\mathrm{m}^3$ is more tight and accurate than that of convex hull 96.0% / $1.43\,\mathrm{m}^3$. In addition, although Heuristic MILP $0.02\,\mathrm{s}$ runs a little bit slower than convex hull $0.01\,\mathrm{s}$, it is still very computationally efficient. Also, as illustrated in Fig. 2, the number of facets (also called hyperfaces) of the convex hull is undetermined and a convex hull usually has too many facets leading to too many constraints, which is not convenient for the operations in practice. In contrast, Heuristic MILP can overcome this limitation. In n-dimensional space, the number of facets of the polytope obtained by Heuristic MILP is always 2^n.

E. Robustness of Heuristic MILP Algorithm

We've demonstrated that Heuristic MILP algorithm is computationally efficient at the expense of optimality compared to the Optimal MILP algorithm when approximating the convex

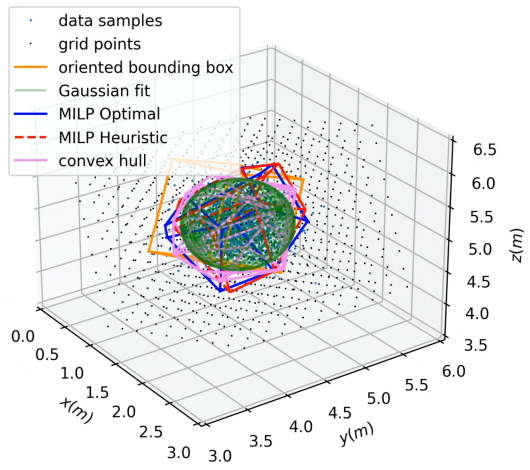


Fig. 2. Polytopic approximation versus convex hull of PRS

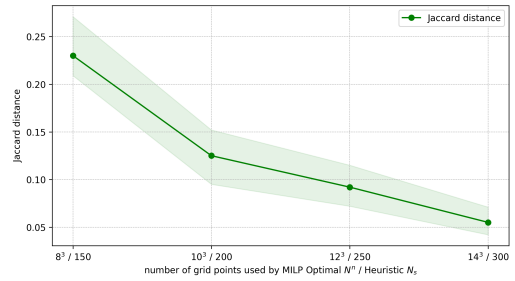
TABLE II
POLYTOPIC APPROXIMATION VERSUS CONVEX HULL OF PRS

Algorithm	**Case 3D ***		
	Ratio / Gap	**Volume** (m^3)	**Time** (s)
Optimal ILP	95.8% / 0.8%	1.35	167.2
Heuristic ILP	95.9% / 0.9%	1.36	0.60
Gaussian Fit	95.1% / 0.1%	1.38	0.02
Bounding Box	98.2% / 3.2%	2.40	0.01
Convex Hull	96.0% / 1.0%	1.43	0.01

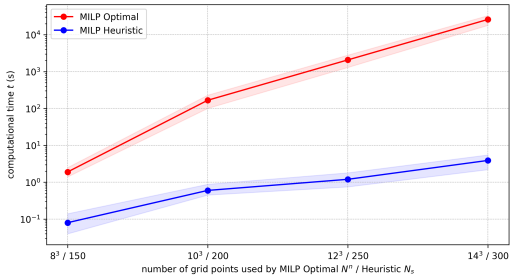
PRS. However, differences can exist between the convex polytopes obtained by these two algorithms. Furthermore, even with identical data samples and parameters, the randomness inherent in weighted sampling during the implementation of Heuristic MILP can result in distinct convex polytopes each time it is applied.

The Jaccard distance defined in [6] is evaluated between the convex polytope of Heuristic MILP and of Optimal MILP, with respect to the increasing quantity of grid points taken by Heuristic MILP N_s and Optimal MILP N^n. All the parameters, except for N_s and N^n, remain consistent with its respective baseline. Given a constant number of grid points used by Heuristic MILP and Optimal MILP, we implement Optimal MILP once to obtain an optimal convex polytope which is constant, and implement Heuristic MILP 20 times to find 20 near-optimal convex polytopes which may differ across different times. Then, we compute the Jaccard distance between the optimal polytope and every near-optimal polytope to quantify the differences. Also, we figure out the computational time of implementing Heuristic MILP every time. The above procedure is repeated for different N^n as well as N_s, respectively.

The results are depicted in Fig. 3. On one hand, Fig. 3a illustrates that when N_s and N^n are too small, various Jaccard distances exhibit significant fluctuations around the average value, indicating poor robustness of Heuristic MILP in terms of space under such circumstances. As the ratio N^n/N_s increases from $8^3/150$ to $14^3/300$, the average Jaccard distance decreases from 0.230 to 0.055 respectively. This suggests that



(a) space robustness



(b) time robustness

Fig. 3. Robustness analysis of Heuristic MILP algorithm

with increasing N^n/N_s, the near-optimal polytope obtained by Heuristic MILP more closely approximates the optimal polytope acquired by Optimal MILP. Additionally, the variance in Jaccard distances also diminishes with the rise in N^n/N_s, indicating improved robustness of Heuristic MILP in terms of space. On the other hand, Fig. 3b demonstrates that as N^n/N_s increases, the running time of Heuristic MILP gradually increases from $0.08\,\mathrm{s}$ to $3.91\,\mathrm{s}$. Conversely, during the same period, the computational time for Optimal MILP dramatically escalates from $1.90\,\mathrm{s}$ to $26\,000\,\mathrm{s}$. This further underscores the superiority of Heuristic MILP over Optimal MILP in terms of computational time. Moreover, as N^n/N_s grows, the variation in computational time of Heuristic MILP increases but not significantly. Hence, Heuristic MILP offers time robustness and efficiency. Overall, the robustness of Heuristic MILP is achieved at the expense of heightened computational effort. An appropriate parameter of the quantity of grid points utilized by Heuristic MILP is expected to be tuned to ensure robustness in both space and time. By doing so, Heuristic MILP can be near-optimal, accurate, efficient, and robust simultaneously.

VI. Conclusion

In this paper, we initially define a MILP optimization problem aimed at determining an n-dimensional convex polytopic approximation of the PRS. Subsequently, we introduce a heuristic algorithm designed to effectively address the formulated MILP problem. Numerical experiments are conducted to evaluate the performance of the Heuristic MILP algorithm in comparison to other algorithms concerning convex PRS approximations. The numerical findings demonstrate the superiority of the proposed Heuristic MILP algorithm, as it achieves computational efficiency without compromising accuracy. Additionally, robustness analysis indicates that the Heuristic MILP algorithm efficiently identifies near-optimal solutions while it is accurate and robust. These advantages of our proposed heuristic algorithm pave the way for potential applications in the real-time, safety-critical motion planning of dynamic systems under uncertainties.

However, the approach proposed in this paper does not provide assurance in handling non-stationary uncertainties that may evolve over time. Another limitation of our work arises in environments where obstacles are densely distributed; in such scenarios, data samples from different obstacles may become intertwined. As a result, our current approach fails to distinguish between them, thereby hindering the ability to derive polytopic approximations of PRS for individual obstacles. Addressing these limitations will be a focus of our future work. In addition, we just focused on the systems whose state is fully measurable. We will extend to consider the systems whose state is not fully measurable in the future.

Acknowledgment

The authors would like to acknowledge the support from the National Science Foundation under Grants CMMI-2138612. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not reflect the views of NSF.

References

[1] H. Zhou, W. Chen, L. Cheng, J. Liu, and M. Xia, "Trustworthy fault diagnosis with uncertainty estimation through evidential convolutional neural networks," *IEEE Transactions on Industrial Informatics*, 2023.

[2] J. Kafunah, M. I. Ali, and J. G. Breslin, "Uncertainty-aware ensemble combination method for quality monitoring fault diagnosis in safety-related products," *IEEE Transactions on Industrial Informatics*, 2023.

[3] B. Wang, J. Xie, K. Lu, Y. Wan, and S. Fu, "Learning and batch-processing based coded computation with mobility awareness for networked airborne computing," *IEEE Transactions on Vehicular Technology*, 2022.

[4] W. Zhang and H. Wen, "Motion planning of a free-flying space robot system under end effector task constraints," *Acta Astronautica*, vol. 199, pp. 195–205, 2022.

[5] P. Wu, L. Li, J. Xie, and J. Chen, "Probabilistically guaranteed path planning for safe urban air mobility using chance constrained rrt*," in *AIAA AVIATION 2020 FORUM*, 2020, p. 2914.

[6] P. Wu and J. Chen, "Online evaluation for chance-constrained geofences under data-driven uncertainties," in *AIAA AVIATION 2022 Forum*, 2022, p. 3613.

[7] H. Zhang, J. Zhang, G. Zhong, H. Liu, and W. Liu, "Multivariate combined collision detection for multi-unmanned aircraft systems," *IEEE Access*, vol. 10, pp. 103 827–103 839, 2022.

[8] M. Fiacchini and T. Alamo, "Probabilistic reachable and invariant sets for linear systems with correlated disturbance," *Automatica*, vol. 132, p. 109808, 2021.

[9] P. Wu and J. Chen, "Efficient box approximation for data-driven probabilistic geofencing," *Unmanned Systems*, 2023.

[10] J. Zhao and L. Mili, "A framework for robust hybrid state estimation with unknown measurement noise statistics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 1866–1875, 2017.

[11] H. Zhang, T. Tian, O. Feng, S. Wu, and G. Zhong, "Research on public air route network planning of urban low-altitude logistics unmanned aerial vehicles," *Sustainability*, vol. 15, no. 15, p. 12021, 2023.

[12] H. Zhu, X. Zhu, and W. Yao, "Probabilistic multi-robot collision avoidance using chance-constrained safety barrier certificates," in *2023 42nd Chinese Control Conference (CCC)*. IEEE, 2023, pp. 5817–5822.

[13] P. Wu, X. Yang, P. Wei, and J. Chen, "Safety assured online guidance with airborne separation for urban air mobility operations in uncertain environments," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[14] M. Xue, J. Jung, and J. Homola, "Intent modeling and conflict probability calculation for operations in upper class e airspace," in *AIAA SCITECH 2022 Forum*, 2022, p. 1508.

[15] P. Wu, J. Xie, Y. Liu, and J. Chen, "Risk-bounded and fairness-aware path planning for urban air mobility operations under uncertainty," *Aerospace Science and Technology*, vol. 127, p. 107738, 2022.

[16] W. Zhao and R. Wen, "The algorithm of fast collision detection based on hybrid bounding box," in *2012 International Conference on Computer Science and Electronics Engineering*, vol. 3. IEEE, 2012, pp. 547–551.

[17] I. Hwang, D. M. Stipanović, and C. J. Tomlin, "Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems," in *Advances in Control, Communication Networks, and Transportation Systems: In Honor of Pravin Varaiya*. Springer, 2005, pp. 3–19.

[18] P. Wu and J. Chen, "Data-driven zonotopic approximation for n-dimensional probabilistic geofencing," *Reliability Engineering & System Safety*, vol. 244, p. 109923, 2024.

[19] P. Wu, S. Martinez, and J. Chen, "Fine-tuned convex approximations of probabilistic reachable sets under data-driven uncertainties," *IEEE Transactions on Automation Science and Engineering*, 2024.

[20] W. K. Härdle and L. Simar, *Applied multivariate statistical analysis*. Springer Nature, 2019.

[21] C. Ericson, *Real-time collision detection*. Crc Press, 2004.

[22] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, vol. 1, no. 1, pp. 65–77, 2002.

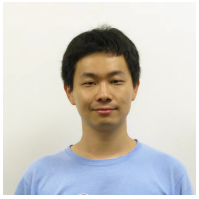
[23] M. P. Wand and M. C. Jones, *Kernel smoothing*. CRC press, 1994.

[24] B. Grünbaum, V. Klee, M. A. Perles, and G. C. Shephard, *Convex polytopes*. Springer, 1967, vol. 16.

[25] I. Griva, S. G. Nash, and A. Sofer, *Linear and nonlinear optimization*. Siam, 2009, vol. 108.

[26] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Information processing letters*, vol. 97, no. 5, pp. 181–185, 2006.

[27] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, and E. Smith, "Astrobee: A new platform for free-flying robotics on the international space station," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, no. ARC-E-DAA-TN31584, 2016.



Pengcheng Wu received both Master and Bachelor degree from Department of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. He is now a joint Ph.D. student with Department of Mechanical and Aerospace Engineering, University of California San Diego, and with Department of Aerospace Engineering, San Diego State University. He has extensive research interests in dynamics, guidance and control of unmanned vehicles. Currently, he is working on the path planning and control of multi-agent systems under uncertainty.

Jun Chen received the B.S degree in Aeronautics Engineering from Beihang University, China, the M.S. and Ph.D degree in Aerospace Engineering from Purdue University. He is currently an Assistant Professor of Aerospace Engineering at San Diego State University. His research interests include control and optimization for large-scale networked dynamic systems, with applications in mechanical and aerospace engineering such as air traffic control, traffic flow management, and autonomous air/ground vehicle systems.