Solving Chance Constrained Optimization under Non-Parametric Uncertainty Through Hilbert Space Embedding.

Bharath Gopalakrishnan, Arun Kumar Singh, K.Madhava Krishna and Dinesh Manocha

***Abstract*—In this paper, we present a computationally efficient algorithm for solving a class of chance constrained optimization under non-parametric uncertainty. Our algorithm is built on the possibility of representing arbitrary distributions as functions in Reproducing Kernel Hilbert Space (RKHS). We use this foundation to formulate chance constrained optimization as one of minimizing the distance between a desired distribution and the distribution of the constraint functions in the RKHS. We provide a systematic way of constructing the desired distribution based on the notion of scenario approximation. Furthermore, we use the kernel trick to show that the computational complexity of our reformulated optimization problem is comparable to solving a deterministic variant of the chance constrained optimization. We validate our formulation on two important robotic applications: (i) reactive collision avoidance of mobile robots in uncertain dynamic environments and (ii) inverse dynamics based path tracking of manipulators under perception uncertainty. In both these applications, the underlying chance constraints are defined over non-linear and non-convex functions of the uncertain parameters and possibly also decision variables. We also benchmark our formulation with the existing approaches in terms of sample complexity and the achieved optimal cost highlighting significant improvements in both these metrics.**

I. Introduction

Consider the following optimization problem in terms of a scalar variable u.

$$\min g(u). \tag{1a}$$
$$p_c(u) \geq \eta. \tag{1b}$$
$$u \in \mathcal{F}. \tag{1c}$$

$$p_c(u) = P(f(\mathbf{w}_1, \mathbf{w}_2, u) \leq 0), \tag{2}$$

Here, $g(u)$ is a user-defined cost function, $P(.)$ represents probability and $f(.)$ is the constraint function which depends on the decision variable u and uncertain parameters, $\mathbf{w}_1, \mathbf{w}_2$. The dependence of $f(.)$ on both $\mathbf{w}_1, \mathbf{w}_2$ and u could possibly be non-linear and non-convex. However, we assume that $f(.)$ is separable in u. The inequality (1b) can be generalized to include any number of uncertain parameters and multiple chance constraints. Nonetheless, the case with two uncertain parameters is rich enough to encompass many common robotic applications (see Section IV). Further, multiple optimization variables can also be accommodated. However, for easier exposition, we first restrict our analysis to the simple case described above. Extensions to a more general case are straightforward, and we discuss those later in the paper.

The set $\mathcal{F}$ represents the feasible space of u and is assumed to be convex for simplicity. Optimizations such as (1a)-(1c) are called chance constrained optimizations and are used extensively for decision making under uncertainty. In robotics and control applications, they form the backbone of the robust Model Predictive Control (MPC) frameworks. For example, see [1], [2], [3], [4].

In this paper, we consider two challenging motion planning/control applications. The first application shown in Fig.2(a)-2(b) involves navigating a mobile robot in dynamic and uncertain environments. Herein, we consider noise arising from both perception and ego-motion, and the chance constraints are formulated to ensure that the probability of collision avoidance is above the specified threshold. The motivation for this application stems from the fact the prediction in dynamic environments (e.g., neighboring vehicles in autonomous driving) would always have some uncertainty associated with it. Furthermore, autonomous vehicles like cars would have noise due to lateral and longitudinal slip; quadrotors would have wind disturbances, etc., leading to uncertainty in the motion model.

Our second application is shown in Fig.2(c)-2(d) is a stochastic variant of inverse dynamics based path tracking for manipulators. We assume that the manipulator has noise-less motions but noisy state estimation. Consequently, the manipulator should compute the necessary torque commands for path tracking while considering the state estimation uncertainty to ensure that the probability of exerting a torque that violates the specified bounds is under some threshold. This requirement can be naturally put in the form of chance constraints. The motivation for considering this example stems from cable-driven and soft inflatable manipulators like Raven [5] and [6] for which encoder readings do not provide a realistic estimate of the configuration of each link. Furthermore, for inflatable manipulators, physical properties like inertia may not also be

Manuscript submitted 30 Novemeber, 2018; revised December 31, 2019, July 1, 2020.

Bharath Gopalakrishnan(email:bharathg91@gmail.com) is a PhD student at Robotics Research Center, IIIT-Hyderabad, India. Arun Kumar Singh(email:aks1812@gmail.com) is an Associate Professor of Collaborative Robotics at University of Tartu, Estonia. K.Madhava Krishna (email:mkrishna@iiit.ac.in) is a Professor at IIIT-Hyderabad, India, affiliated with Robotics Research Center. Dinesh Manocha (email:dm@cs.umd.edu) is Paul Chrisman Iribe Professor of Computer Science and Professor of Electrical and Computer Engineering at University of Maryland College Park. The research was partly supported by IT Academy of Estonia grant (SLTTI19605T) to the second author.

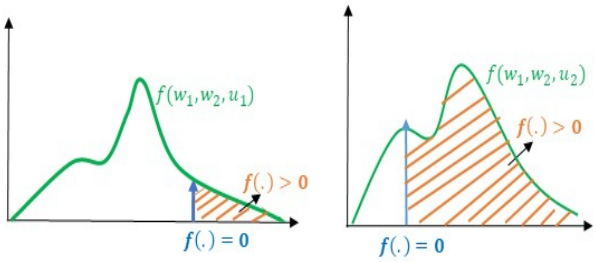


Fig. 1. An illustration of the observations made in Remark 1. The shape of the distribution can be manipulated by u. An appropriate shape is one where most of the mass lies to the left of $f(.) = 0$

known accurately. Our formulation can act as a building block for complex torque control with these manipulators.

Remark 1. *At an intuitive level, chance constrained optimizations can be interpreted as a problem of ensuring that a specific portion of the mass of the distribution $f(\boldsymbol{w}_1, \boldsymbol{w}_2, u)$ lie to the left of $f(.) = 0$ (refer to Fig. 1). For given uncertain parameters $\boldsymbol{w}_1, \boldsymbol{w}_2$, the distribution is parameterized by the decision variable u and therefore can be used to manipulate the location of a specified portion of its mass. However, each choice of u incurs a cost $g(u)$.*

Remark 2. *The chance constraint probability η has a direct correlation with the amount of mass of the distribution $f(\boldsymbol{w}_1, \boldsymbol{w}_2, u)$ lying to the left of $f(.) = 0$. A larger mass amounts to a higher η.*

A. Computational Challenge

Chance constrained optimizations are known to be very difficult to solve. The complexity increases further when the uncertainty is non-parametric, that is, the analytical, functional form of the probability distribution of $\mathbf{w}_1, \mathbf{w}_2$ are not known. Chance constraints are easy to solve when $\mathbf{w}_1$, $\mathbf{w}_2$ are assumed to have a Gaussian distribution and the constraint function $f(.)$ is affine with respect to u for given $\mathbf{w}_1$, $\mathbf{w}_2$ [7], [8]. However, in general, optimization problems where chance constraints are defined over non-linear and non-convex functions, and the underlying uncertainty cannot be represented in any parametric form are known to be computationally intractable. Thus, various approximations and reformulations are proposed in the existing literature to tackle chance constrained optimization problems.

Scenario Approximation: A popular approximation called the scenario approach [9], [10] starts with, drawing n samples (or scenarios) of $\mathbf{w}_1, \mathbf{w}_2$ from their distribution and then replaces (1b) with n^2 constraints of the form $f_i(\mathbf{w}_1^i, \mathbf{w}_2^j, u) \leq 0, \forall i, j$. The scenario approach has a very interesting set of pros and cons. On the one hand, it is conceptually simple and is applicable even when the parametric form of the distribution of uncertain parameters is not known, and just their samples are given. On the other hand, the naive implementation of the scenario approach is known to be overly conservative.

TABLE I
IMPORTANT SYMBOLS

$f(.)$	Constraint function
η	Chance constraint probability
$p_f(u)$	Distribution of the constraint function
p_f^{des}	Desired distribution
$\mathbf{w}_1, \mathbf{w}_2$	uncertain parameters
$\mathbf{w}_1^i, \mathbf{w}_2^j$	i^{th}, j^{th} sample of uncertain parameters
${}^i\mathbf{w}_2$	i^{th} variant of the uncertain parameter $\mathbf{w}_2$.
$k(.,.)$	Kernel function
μ_{p_f}	RKHS embedding of the distribution $f(\mathbf{w}_1, \mathbf{w}_2, u)$ or $p_f(u)$
$\mu_{p_f^{des}}$	RKHS embedding of the distribution p_f^{des}
$E[f(.)]$	Expectation of a function $f(.)$ with respect to its random arguments
$Var[f(.)]$	Expectation of a function $f(.)$ with respect to its random arguments

To be precise, the cost $g(u)$ increases with n, although the solution becomes more robust at the same time. Works like [11] provide algorithms for rejection sampling to reduce the conservativeness of the scenario approach.

Surrogates for Chance Constraints: [12], [13], [14], [15] proposed to replace chance constraints with the surrogate (3).

$$E[f(\mathbf{w}_1, \mathbf{w}_2, u)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u)]} \leq 0, \epsilon > 0, \tag{3}$$

where, $E[.]$, $Var[.]$ represent the mean and variance of $f(.)$, taken with respect to random variables $\mathbf{w}_1, \mathbf{w}_2$. Using Cantelli's inequality, it can be shown that the satisfaction of (3) ensures that chance constraints are satisfied with $\eta \geq \frac{\epsilon}{1+\epsilon^2}$. However, it should be noted that this bound can be rather loose. The attractive feature of (3) is that it is applicable for a wide class of chance constraints. However, its efficiency is predicated on how easy it is to compute analytical expressions for $E[.]$ and $Var[.]$. For example, if $f(.)$ is non-linear or/and the parametric form of $\mathbf{w}_1, \mathbf{w}_2$ is not known, then computing an accurate analytical expression for $E[.]$ and $Var[.]$ becomes a very challenging problem. A workaround has been proposed in works like [14], [16] [17] where the analytical expressions for $E[.]$ and $Var[.]$ are approximated through Monte Carlo sampling.

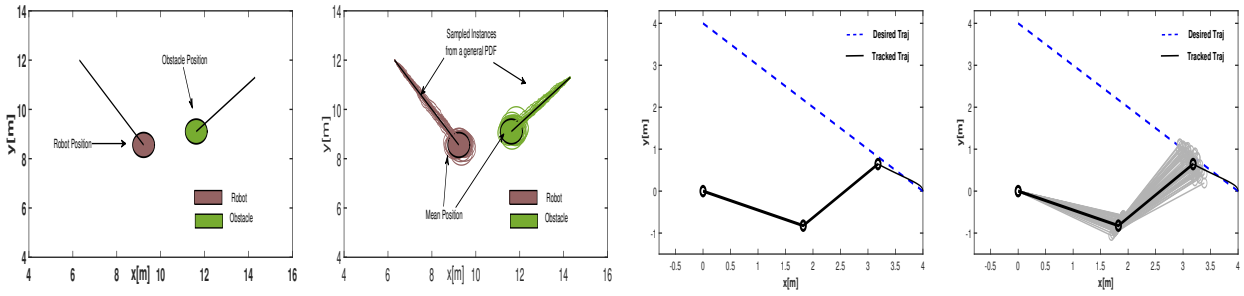
Sample Average Approximation: Another approach for reducing the conservativeness of the scenario approach is to reformulate the chance constraints as (4) using the so-called sample average approximation (SAA) given by (5) [18].

$$\widetilde{p}_c(u). \geq \delta \tag{4}$$

$$\widetilde{p}_c(u) \approx \frac{1}{n^2}\sum_i\sum_j \mathcal{I}_f. \tag{5}$$

$$\mathcal{I}_f = \begin{cases} 1, & \text{if } f(\mathbf{w}_1^i, \mathbf{w}_2^j, u) \leq 0. \\ 0, & \text{otherwise.} \end{cases}$$

Here, $\mathbf{w}_1^i, \mathbf{w}_2^j$ represent the i^{th}, j^{th} samples of $\mathbf{w}_1, \mathbf{w}_2$ and $\mathcal{I}_f$ represents an indicator function. The variable δ is similar but not necessarily the same as the chance constraint probability



(a) (b) (c) (d)

Fig. 2. Applications of chance-constrained optimization considered in this paper. Fig. (a) shows a robot avoiding collision with a dynamic obstacle in a deterministic noise-less setting. Fig. (b) represents the stochastic variant of the collision avoidance problem wherein the robot has noise in its motion commands as wells its perception of the movement of the dynamic obstacle. In this setting, the chance constraints model the probability of collision avoidance. Fig. (c): Problem set-up for inverse dynamics based path tracking for a two-link planar manipulator in a deterministic setting. Fig. (d): Inverse dynamics based path tracking under perception uncertainty leading to noisy estimates for the joint position (grey shaded configurations) and joint velocities. The torque that the manipulator can exert for path-tracking depends on the joint positions and velocities. Thus, noise in these entities means that the manipulator is unsure of the maximum torque it can generate without violating the specified bounds. We formulate chance-constraints to ensure that the torque bounds are satisfied with some specified probability.

η [18]. A strong advantage of SAA (5) is that it automatically allows for violation of $f(\mathbf{w}_1^i, \mathbf{w}_2^j, u) \leq 0$ for some appropriate subset of samples of $\mathbf{w}_1^i, \mathbf{w}_2^j$ (depending on specified η or δ) which in turn leads to solution with less conservative cost $g(u)$. A closer look would reveal that $\widetilde{p}_c(u)$ as given by (5) is in fact the sample approximation of $p_c(u)$ [18]. A key bottleneck in using SAA is that reformulation of (1a)-(1c) using (4) leads to a mixed integer optimization wherein the number of binary variables would be the product of the number of samples of $\mathbf{w}_1, \mathbf{w}_2$. Thus, even if we have 20 samples of each of them, we would have to deal with 400 binary variables which is computationally intensive if not practically intractable. Note that in [18], the uncertainty is clubbed in a single parameter which is different from the set-up considered in this paper.

B. Key Idea and Motivation for RKHS Embedding

Our main motivation is to perform distribution level reasoning as SAA but at the same time do away with the use of binary variables. We conjecture that this can be achieved by working with the distribution of $f(\mathbf{w}_1, \mathbf{w}_2, u)$ rather than the distribution of $f(\mathbf{w}_1, \mathbf{w}_2, u) \leq 0$ (as done in SAA [18]). Let, $p_f(u)$ represent the distribution of $f(\mathbf{w}_1, \mathbf{w}_2, u)$ parameterized in terms of u. As mentioned, earlier, for non-parametric $\mathbf{w}_1, \mathbf{w}_2$ and non-linear and non-convex $f(.)$, it is intractable to obtain an analytical expression of $p_f(u)$. However, it is possible to obtain a expression for the embedding of $p_f(u)$ in the Reproducing Kernel Hilbert Space (RKHS). Let μ_{p_f} represent the RKHS embedding of $p_f(u)$ given by the following expression [19]:

$$\mu_{p_f}(u) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \beta_j k(f(\mathbf{w}_1^i, \mathbf{w}_2^j, u), .), \tag{6}$$

where, $k(.) : \Re^n \times \Re^N \rightarrow \Re$ is a positive definite function called the Kernel. The α_i, β_j are the weights associated with $\mathbf{w}_1, \mathbf{w}_2$ respectively. For example, if the samples are i.i.d then, $\alpha_i, \beta_j = \frac{1}{n}$. An important thing to note from (6) is that for the given samples of $\mathbf{w}_1$, $\mathbf{w}_2$, the embedding given by (6) is dependent on the variable u.

The expression for μ_{p_f} is semi-analytic in nature in the sense that it is not possible to evaluate the right-hand side of (6) since the second argument of the kernel is not known. Nevertheless, the utility of (6) stems from a different perspective. To understand this further, assume that we have a known distribution p_f^{des} whose RKHS embedding is given by $\mu_{p_f^{des}}$. Now, it is straightforward to obtain an analytical expression for the distance between the two distributions in RKHS [19] [20].

$$\overbrace{\|\mu_{p_f}(u) - \mu_{p_f^{des}}\|^2}^{MMD} = \langle \mu_{p_f}(u) - \mu_{p_f^{des}}, \mu_{p_f}(u) - \mu_{p_f^{des}}. \rangle \tag{7}$$

The left hand side of (7) is called the maximum mean discrepancy (MMD) between the distributions $p_f(u)$ and p_f^{des} and can serve as the measure of similarity between the two distributions. The right hand side (7) involves computing the inner product of two functions embedded in RKHS and thus can be easily computed based on so-called "kernel trick". In the classic distribution matching setup of [19] [20], $p_f(u)$ and consequently MMD is independent of u and depends only on uncertain parameters $\mathbf{w}_1, \mathbf{w}_2$. In contrast, MMD as defined in (7) is clearly parameterized by u. That is, different u leads to different MMD measure. We exploit this precise feature in our formulation.

C. Contribution

In this paper we present the first result which provides a reformulation of chance constrained optimization as a problem of minimizing the cost $g(u)$ augmented with MMD of the distribution $p_f(u)$ with a certain given distribution p_f^{des}, which we will henceforth call as the desired distribution. We achieve this by connecting two existing results: (i) Matching the tail of

two distributions can be formulated as a moment[1] matching problem [21] and (ii) Moment matching can be formulated as minimizing MMD constructed with polynomial kernels [20], ([22], pp-15). As shown in Section III, the connecting link is derived from our interpretation of chance-constrained optimization in the form presented in Remark 1 and the way we construct p_f^{des}.

We show that for the class of constraint function $f(.)$ considered in this paper, the reformulated optimization problem takes the form of a non-linear optimization problem easily solvable through off-the-shelf gradient based optimizers. To be precise, if $f(.)$ is polynomial in u of order l, then the reformulated optimization consists of a cost which is polynomial of order $2l$ subject to convex feasibility constraints (1c). This is computationally significantly simpler than mixed-integer optimization encountered in SAA approach [18]. Furthermore, we present a detailed derivation of MMD constructed over $p_f(u)$ and p_f^{des} and put it in a form for which analytical gradients can be easily computed.

We benchmark our formulation with the existing approaches on the applications showed in Fig.2(a)-2(d) based on two metrics, namely sample complexity and obtained optimal cost. In particular, we highlight the following results: First, we show that our formulation significantly outperforms a baseline scenario approximation in both the metrics. Second, our formulation and the SAA approach based on reformulated chance constraints (5) result in a similar optimal cost. However, our formulation leads to a simpler optimization problem and enjoy better sample complexity. Finally, our formulation also outperforms reformulation of chance constraints based on expectation and variance of $p_f(u)$ proposed in works like [13], [14], [15].

II. Preliminaries

In this section, we summarize the pre-requisites needed to build our main results.

A. Moment Matching Problem

Our formulation relies on matching the tail of $p_f(u)$ with certain desired distribution p_f^{des} to be constructed later. Hypothesis: the Theorem 1 from [21] can allow us to interpret tail matching as a problem of matching (or making similar) the first d moments of $p_f(u)$ and p_f^{des}.

Theorem 1. $\|p_f(u) - p_f^{des}\| \leq B(d),\ B(d) \to 0,\ d \to \infty.$

where, $B(d) \geq 0$ is a non-negative function, the algebraic expression of which can be found in [21]. The above theorem suggests that the difference between two distributions can be bounded by the function $B(d)$ that decreases with increasing order of moment d. Authors in [21] also show that this bound is particularly tight near the tail end of the distribution.

[1]Moment here is used in the context of distributions and should not be confused with physical moment resulting from the application of force.

B. Moment Matching in the RKHS

Assume that the RKHS embedding μ_{p_f} and $\mu_{p_f^{des}}$ are computed through the following polynomial kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^T \mathbf{x}_2)^d. \tag{8}$$

Then the following theorem holds.

Theorem 2. *If* $\|\mu_{p_f}(u) - \mu_{p_f^{des}}\| \to 0$, *then moments of* $p_f(u)$ *and* p_f^{des} *upto order* d *become similar.*

Where, $\mu_{p_f}(u)$ and $\mu_{p_f^{des}}$ are respectively the kernel mean embedding of $p_f(u)$ and p_f^{des} constructed according to (6). Note that, $p_f(u)$ and consequently $\mu_{p_f}(u)$ is a function of u. However, since the desired distribution p_f^{des} is assumed to be known, $\mu_{p_f^{des}}$ is constant.
That is, decreasing the residual of MMD distance becomes a way of matching the first d moments of the distribution $p_f(u)$ and p_f^{des}. Theorem 2 suggests that the MMD distance can be used as a measure of similarity between the first d moments of the two distributions.

C. Reduced Set Methods

One of the strengths of RKHS embedding is that it opens up avenues for the use of established *reduced set methods* to achieve a good sample complexity. Intuitively, *reduced set method* provides a systematic way of choosing a subset of samples while still retaining as much information as possible from the original sample size by re-weighting the importance of those samples. In other words, the reduced set methods allow us to compute an optimal α_i, β_j for use in (7). Let $\hat{\mathbf{w}}_1^1, \hat{\mathbf{w}}_1^2..\mathbf{w}_1^N$ and $\hat{\mathbf{w}}_2^1, \hat{\mathbf{w}}_2^2..\mathbf{w}_2^N$ represent N i.i.d samples of $\mathbf{w}_1$, $\mathbf{w}_2$ respectively. Further, let $\mathbf{w}_1^1, \mathbf{w}_1^2..\mathbf{w}_1^n$ and $\mathbf{w}_2^1, \mathbf{w}_2^2..\mathbf{w}_2^n$ represent a subset (reduced set) of the i.i.d samples. It is implied that $n \ll N$. Now, intuitively, a reduced set method would re-weight the importance of each sample from the reduced set such that they retain as much as information of the original i.i.d samples. The weights α_i, β_j associated with $\mathbf{w}_1^i$ and $\mathbf{w}_2^i$ are computed through the following optimization problems.

$$\arg\min_{\alpha_i} \|\frac{1}{N}\sum_{i=1}^{i=N} k(\hat{\mathbf{w}_1}^i, .) - \frac{1}{n}\sum_{i=1}^{i=n} \alpha_i k(\mathbf{w}_1^i, .)\|_2, \sum \alpha_i = 1. \tag{9}$$

$$\arg\min_{\beta_j} \|\frac{1}{N}\sum_{j=1}^{i=N} k(\hat{\mathbf{w}_2}^i, .) - \frac{1}{n}\sum_{j=1}^{i=n} \beta_j k(\mathbf{w}_2^j, .)\|_2, \sum \beta_j = 1. \tag{10}$$

Note that the cost functions in (9) and (10) can be easily expressed as the inner product of two kernel functions, which eventually simplifies through kernel trick [19].

[1]Note that if we fix u in $\mu_{p_f}(u)$, then this embedding also becomes a constant which is exactly the trick that we use in the latter sections to construct a constant $\mu_{p_f^{des}}$

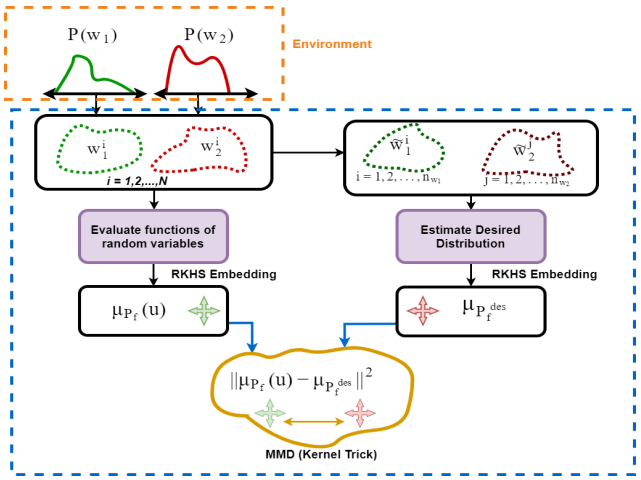


Fig. 3. Overview of our RKHS based formulation

III. Main Results

In this section, we derive our main result, which is a reformulation of chance-constrained optimization (1a)-(1c) into a much simpler minimization problem. The following are our key assumptions:

- We assume that the uncertainty is non-parametric, which, in our case, means that the probability distribution functions associated with $\mathbf{w}_1, \mathbf{w}_2$ are not known. Rather, we have access to their n discrete samples. These samples could come from a black box simulator, which mimics a generalized distribution with arbitrary order of moments.
- We assume that the analytical form for the constraint function $f(\mathbf{w}_1, \mathbf{w}_2, u)$ is known.

A. Overview

Fig. 3 gives an overview of our RKHS based formulation. We assume access to an environment that provides samples of uncertain parameters $\mathbf{w}_1, \mathbf{w}_2$. Physically, this can represent the possible states of the robot and the obstacles in the environment. Our algorithm begins by computing a subset of these samples denoted by $\widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2$. We use these subset samples to estimate a certain desired distribution. Next, we compute the RKHS embedding of the desired distribution p_f^{des} and $p_f(u)$ through polynomial kernels. Subsequently, we present the minimization of the distance between the two RKHS embedding as a tractable substitute for our robust MPC (1a)-(1c).

B. Algebraic Form of the Constraint Function

In this paper, we consider the chance constraints defined over the following class of constraint functions.

$$f(\mathbf{w}_1, \mathbf{w}_2, u) = \sum_{i=0}^{l} h_i(\mathbf{w}_1, \mathbf{w}_2) u^i, \tag{11}$$

where, $h_i(\mathbf{w}_1, \mathbf{w}_2), \Re^n \rightarrow \times\Re^n \rightarrow \Re$ is a generic possibly non-linear function of $\mathbf{w}_1, \mathbf{w}_2$, while u^i represents a monomial of order i. The definition (11) is very general and has the famous affine class of chance constraints as a special case with $l = 1$ and $h_0(\mathbf{w}_1, \mathbf{w}_2) = \mathbf{w}_2$, $h_1(\mathbf{w}_1, \mathbf{w}_2) = \mathbf{w}_1$. It can be seen that even if the uncertain parameters, $\mathbf{w}_1$, $\mathbf{w}_2$ are Gaussian, the chance constraints defined over $f(\mathbf{w}_1, \mathbf{w}_2, u)$ may still be too complex to get an analytical characterization for the distribution of $f(\mathbf{w}_1, \mathbf{w}_2, u)$.

For the class of constraint functions (11), the RKHS embedding (7) can be simplified in the following manner

$$\mu_{p_f}(u) = \sum_{i=0}^{i=l} \mu_{h_i} u^i. \tag{12}$$

$$\mu_{h_i} = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} \alpha_i \beta_j k(h_i(\mathbf{w}_1^i, \mathbf{w}_2^j), .). \tag{13}$$

C. Desired Distribution

The notion of desired distribution is derived from the observations made in Remark 1. To recap, we want to ensure that the distribution $f(\mathbf{w}_1, \mathbf{w}_2, u)$ achieves an appropriate shape. To this end, desired distribution acts as a benchmark for $f(\mathbf{w}_1, \mathbf{w}_2, u)$; in other words, a distribution that $f(\mathbf{w}_1, \mathbf{w}_2, u)$ should resemble as closely as possible for an appropriately chosen u. We formalize the notion of desired distribution with the help of the following definitions:

Definition 1. *u_{nom} refers to any solution of the optimization (1a)-(1c) that is associated with a low optimal cost $J(u_{nom})$.*

Definition 2. *Let $\widetilde{\boldsymbol{w}}_1$, $\widetilde{\boldsymbol{w}}_2$ be random variables which represent the same entity as $\boldsymbol{w}_1, \boldsymbol{w}_2$ but belong to some known distributions $p_{\boldsymbol{w}_1}^{des}$, $P_{\boldsymbol{w}_2}^{des}$. Further, when $\widetilde{\boldsymbol{w}}_1 \sim p_{\boldsymbol{w}_1}^{des}$ and $\widetilde{\boldsymbol{w}}_2, \sim p_{\boldsymbol{w}_2}^{des}$, then, $f(\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, u_{nom}) \sim p_f^{des}$. In such a case, p_f^{des} is called the desired distribution if the following holds:*

$$P(f(\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, u_{nom}) \leq 0) \approx 1.0, \widetilde{\boldsymbol{w}}_1 \sim P_{\boldsymbol{w}_1}^{des}, \widetilde{\boldsymbol{w}}_2 \sim P_{\boldsymbol{w}_2}^{des}. \tag{14}$$

Equation (14) suggests that if the uncertain parameters belong to the distribution $p_{\mathbf{w}_1}^{des}$, $p_{\mathbf{w}_2}^{des}$, then the entire mass of the distribution, $f(\widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2, u)$ can be manipulated to lie almost completely to the left of $f(.) = 0$ by choosing $u = u_{nom}$. This setting represents an ideal case because we have constructed uncertainties appropriately, so that we can manipulate the distribution of the chance constraints while incurring a nominal cost.

Constructing the Desired Distribution:
We now describe how distributions $p_{\mathbf{w}_1}^{des}$, $p_{\mathbf{w}_2}^{des}$ and p_f^{des} can be constructed. While exact computations may be intractable, in this section, we provide a simple way of constructing an approximate estimate of these distributions. The basic procedure is as follows.

Given n samples of $\mathbf{w}_1, \mathbf{w}_2$ we construct two sets $\mathcal{C}_{\widetilde{\mathbf{w}_1}}$, $\mathcal{C}_{\widetilde{\mathbf{w}_2}}$ respectively containing $n_{\mathbf{w}_1}$ samples of $\mathbf{w}_1$ and $n_{\mathbf{w}_2}$ samples of $\mathbf{w}_2$. For clarity of exposition, we choose $\widetilde{\mathbf{w}}_1$, $\widetilde{\mathbf{w}}_2$ to identify samples from set $\mathcal{C}_{\widetilde{\mathbf{w}_1}}$, $\mathcal{C}_{\widetilde{\mathbf{w}_2}}$. Now, assume that the following holds.

$$f(\widetilde{\mathbf{w}}_1^i, \widetilde{\mathbf{w}}_2^j, u_{nom}) \leq 0, \forall \widetilde{\mathbf{w}}_1^j \in \mathcal{C}_{\widetilde{\mathbf{w}_1}}, \widetilde{\mathbf{w}}_2^j \in \mathcal{C}_{\widetilde{\mathbf{w}_2}}. \quad (15)$$

By comparing (14) and (15), it can be inferred that the sets $\mathcal{C}_{\widetilde{\mathbf{w}_1}}, \mathcal{C}_{\widetilde{\mathbf{w}_2}}$ are in fact sample approximations of the distributions $p_{\mathbf{w}_1}^{des}$ and $p_{\mathbf{w}_2}^{des}$ respectively. Furthermore, a set $\mathcal{C}_f$ containing $n_{\mathbf{w}_1} * n_{\mathbf{w}_2}$ samples of $f(\widetilde{\mathbf{w}}_1^i, \widetilde{\mathbf{w}}_2^j, u_{nom})$ can be taken as the sample approximation of the desired distribution p_f^{des}.

One last piece of puzzle remains. We still do not know, however which $n_{\mathbf{w}_1}$ samples of $\mathbf{w}_1$ and $n_{\mathbf{w}_2}$ samples of $\mathbf{w}_2$ should be chosen to construct sets $\mathcal{C}_{\widetilde{\mathbf{w}_1}}, \mathcal{C}_{\widetilde{\mathbf{w}_2}}$. In particular, we need to ensure that the assumption (15) holds for the chosen samples. To this end, we follow the following process. We arbitrarily choose $n_{\mathbf{w}_1}$ samples of $\mathbf{w}_1$ and $n_{\mathbf{w}_2}$ samples of $\mathbf{w}_2$ and correspondingly obtain a suitable u_{nom} as a solution to the following optimization problem:

$$u_{nom} = \arg\min g(u). \quad (16a)$$
$$f(\mathbf{w}_1^i, \mathbf{w}_2^j, u) \leq 0 \forall i = 1, 2..n_{\mathbf{w}_1}, j = 1, 2..n_{\mathbf{w}_2}. \quad (16b)$$
$$u \in \mathcal{F}. \quad (16c)$$

Note that satisfaction of (16b) ensures that the assumption (15) holds. Few points are worth noting about the above optimization. First, it is a deterministic problem whose complexity primarily depends on the algebraic nature of $f(.)$. Second, the desired distribution can always be constructed if we have access to sets $\mathcal{C}_{\widetilde{\mathbf{w}_1}}$, $\mathcal{C}_{\widetilde{\mathbf{w}_2}}$. The construction of these two sets is guaranteed as long as we can obtain a feasible solution to (16a)-(16c). Third, the computational burden of solving the optimization problem can be significantly reduced by some clever sampling. For example, in our implementation, we compute the left hand side of (16b) for different combination of samples and then choose the set which leads to the least violation of the constraints (16b). Finally, (16a)-(16c) is precisely the so-called scenario approximation for chance constrained optimization (1a)-(1c). Conventionally, scenario approximation is solved with a large $n_{\mathbf{w}_1}, n_{\mathbf{w}_2}$ (typically 200 samples of each leading to a grid of $4 * 10^4$ and as many constraints) in order to obtain a solution that satisfy chance constraints (1b) with a high η (≈ 0.90). In contrast, we use (16a)-(16c) to estimate the desired distribution and thus, for our purpose, a small sample size in the range of $n_{\mathbf{w}_1} = n_{\mathbf{w}_2} \approx 20$ proves to be sufficient in practice.

The RKHS embedding of these distributions can be obtained in the following manner:

$$\mu_{p_f^{des}} = \sum_{i=1}^{i=n_{\mathbf{w}_1}} \sum_{j=1}^{j=n_{\mathbf{w}_2}} \lambda_i \xi_j k(f(\widetilde{\mathbf{w}}_1^i, \widetilde{\mathbf{w}}_2^j), .), \quad \widetilde{\mathbf{w}}_1^i, \widetilde{\mathbf{w}}_2^j \in \mathcal{C}^{\widetilde{\mathbf{w}}_1}, \mathcal{C}^{\widetilde{\mathbf{w}}_2}, \quad (17)$$

where, λ_i, ξ_j are constants derived from the reduced set methods described in Section II-C.

Notion of Low: As well known, the number of samples and optimal cost have an inverse relationship in scenario approximation. So our use of the term "low" signifies that we want to construct the desired distribution with as low number of samples as possible. Although we do not have a theoretical bound on the number of samples required, in our implementation, we empirically evaluate various sample sizes. All our implementation in the paper were constructed with a sample size of $5 - 20$.

D. Chance Constrained Optimization as a Moment Matching Problem

In this section, we reformulate the chance-constrained optimization (1a)-(1c) as a moment matching problem. Our key idea builds upon Theorem 1. Recall that almost the entire mass of p_f^{des} lies to the left of $f(.) = 0$. It is thus clear that as we make the tail of p_f^{des} and $p_f(u)$ similar by matching higher order moments, we ensure that more and more of the mass of $p_f(u)$ gets shifted to the left of $f(.) = 0$. This in turn would lead to the satisfaction of chance constraints (1b) with a higher η (see Remark 2). Theorem 1 lays the foundation for the following optimization problem which can act as a substitute for the original chance constrained optimization (1a)-(1c).

$$\arg\min \rho_1 \mathcal{L}_{mom}(P_f(u), P_f^d, d) + \rho_2 g(u). \quad (18a)$$
$$u \in \mathcal{F}. \quad (18b)$$

Here, $\mathcal{L}_{mom}(.)$ is a cost function that measures the similarity between the first d moments of $p_f(u)$ and p_f^{des}. That is, a low value of $\mathcal{L}_{mom}$ would imply that the first d moments of $p_f(u)$ and p_f^{des} are very similar.

Accommodating Chance Constraint Probability η: Optimization (18a)-(18b), accommodates the chance constraint probability η in an implicit manner. Thus, the process of obtaining solutions with different level of robustness based on η is more indirect and involved than the original optimization (1a)-(1c). In (18a)-(18b), the similarity between the tail of $p_f(u)$ and p_f^{des} not only depends on the residual of $\mathcal{L}_{mom}(.)$ but also on the moment order d used to construct $\mathcal{L}_{mom}(.)$. Fixing weights ρ_1 and ρ_2 and increasing d increases the similarity near the tail end and thus leads to the satisfaction of chance constraints with higher η. A similar goal can be achieved by fixing d and ρ_2 and increasing ρ_1.

E. Reformulating Distribution/Moment Matching through RKHS Embedding

The optimization (18a)-(18b) is still challenging to solve as it is not clear how to derive a suitable analytical form for $\mathcal{L}_{mom}(.)$. To the best of our knowledge, there is no mapping that directly quantifies the similarity between the first d moments of two given distributions. Here, we present a workaround based on the concept of RKHS embedding and Theorem 2. Essentially, if we construct the RKHS embedding of p_f, p_f^{des} through polynomial kernel, then decreasing the residual of MMD becomes a way of matching the first d moments of the distribution $p_f(u)$ and p_f^{des}. In other words, MMD with the polynomial kernel can act as a surrogate for $\mathcal{L}_{mom}(.)$. Using this insight, we present the following

optimization problem which can act as a surrogate for (18a)-(18b).

$$\arg\min \rho_1 \overbrace{\|\mu_{p_f}(u) - \mu_{p_f^{des}}\|^2}^{MMD} + \rho_2 g(u). \quad (19a)$$

$$u \in \mathcal{F}. \quad (19b)$$

F. Simplification Based on Kernel Trick

We now use the so called "kernel trick" to obtain a simplified form for the optimization (19a)-(19b). In particular, we put the cost (19a) in a form for which the gradient can be easily computed. For the ease of exposition, we consider a specific instance from the definition of constraint function (11) with $l = 2$ i.e. $f(.) = h_0(.) + h_1(.)u + h_2(.)u^2$.

We have,

$$\begin{aligned}\|\mu_{p_f}(u) - \mu_{p_f^{des}}\|^2 &= \langle \mu_{p_f}(u) - \mu_{p_f^{des}}, \mu_{p_f}(u) - \mu_{p_f^{des}} \rangle \\ &= \langle \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2, \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2 \rangle \\ &-2\langle \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2, \mu_{p_f^{des}} \rangle + \langle \mu_{p_f^{des}}, \mu_{p_f^{des}} \rangle. \quad (20)\end{aligned}$$

Expanding $\langle \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2, \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2 \rangle$, we get

$$\begin{aligned}&u^4 \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_2(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_2(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle \\ &+2u^3 \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_2(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_1(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle \\ &+u^2 \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_2(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_0(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle \\ &+u^2 \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_1(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_1(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle \\ &+2u \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_1(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_0(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle \\ &+ \langle \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_0(\mathbf{w}_1^i, \mathbf{w}_2^j), .), \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\beta_j k(h_0(\mathbf{w}_1^i, \mathbf{w}_2^j), .) \rangle. \quad (21)\end{aligned}$$

Using the kernel trick [19] on (21) reduces it to the following expression

$$\begin{aligned}&u^4 \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_2}\mathbf{c}_{\alpha\beta}^T + 2u^3 \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_1}\mathbf{c}_{\alpha\beta}^T + 2u^2 \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_0}\mathbf{c}_{\alpha\beta}^T \\ &+u^2 \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1h_1}\mathbf{c}_{\alpha\beta}^T + 2u\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1h_0}\mathbf{c}_{\alpha\beta}^T + \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_0h_0}\mathbf{c}_{\alpha\beta}^T, \quad (22)\end{aligned}$$

where,

$$\mathbf{c}_{\alpha\beta} = [\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, ...\alpha_n\beta_n]_{1Xn^2}. \quad (23)$$

$$\mathbf{K}_{h_ih_j} = \begin{pmatrix} \mathbf{K}_{h_i,h_j}^{11} & \mathbf{K}_{h_i,h_j}^{12} & \mathbf{K}_{h_i,h_j}^{13} & .. & .. & \mathbf{K}_{h_i,h_j}^{1n} \\ \mathbf{K}_{h_i,h_j}^{21} & \mathbf{K}_{h_i,h_j}^{22} & \mathbf{K}_{h_i,h_j}^{23} & .. & .. & \mathbf{K}_{h_i,h_j}^{2n} \\ . & . & . & .. & .. & . \\ . & . & . & .. & .. & . \\ \mathbf{K}_{h_i,h_j}^{n1} & \mathbf{K}_{h_i,h_j}^{n2} & \mathbf{K}_{h_i,h_j}^{n3} & .. & .. & \mathbf{K}_{h_i,h_j}^{nn} \end{pmatrix}. \quad (24)$$

$$\mathbf{K}_{h_ih_j}^{ab} = \begin{pmatrix} k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^1), h_j(\mathbf{w}_1^b, \mathbf{w}_2^1)), & ... & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^1), h_j(\mathbf{w}_1^b, \mathbf{w}_2^n)) \\ k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^2), h_j(\mathbf{w}_1^b, \mathbf{w}_2^1)), & ... & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^2), h_j(\mathbf{w}_1^b, \mathbf{w}_2^n)) \\ ., & .., & .. \\ k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^n), h_j(\mathbf{w}_1^b, \mathbf{w}_2^1)), & ... & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^n), h_j(\mathbf{w}_1^b, \mathbf{w}_2^n)) \end{pmatrix}_{n\times n}.$$

Following a similar process, the second term, $2\langle \mu_{h_0} + \mu_{h_1}u + \mu_{h_2}u^2, \mu_{p_f^{des}} \rangle$ reduces to

$$2(\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2f}\mathbf{c}_{\lambda\xi}^T u^2 + \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1f}\mathbf{c}_{\lambda\xi}^T u + \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_0f}\mathbf{c}_{\lambda\xi}^T), \quad (25)$$

where,

$$\begin{aligned}\mathbf{c}_{\alpha\beta} &= [\alpha_1\beta_1, \alpha_1\beta_2, \alpha_1\beta_3, ...\alpha_n\beta_n]_{1X(n*n)} \\ \mathbf{c}_{\lambda\xi} &= [\lambda_1\xi_1, \lambda_1\xi_2, \lambda_1\xi_3, ...\lambda_{n_{\mathbf{w}_1}}\xi_{n_{\mathbf{w}_2}}]_{1X(n_{\mathbf{w}_1}*n_{\mathbf{w}_2})}.\end{aligned}$$

$$\mathbf{K}_{h_if} = \begin{pmatrix} \mathbf{K}_{h_i,f}^{11} & \mathbf{K}_{h_i,f}^{12} & \mathbf{K}_{h_i,f}^{13} & .. & .. & \mathbf{K}_{h_i,f}^{1n_{\mathbf{w}_1}} \\ \mathbf{K}_{h_i,f}^{21} & \mathbf{K}_{h_i,f}^{22} & \mathbf{K}_{h_i,f}^{23} & .. & .. & \mathbf{K}_{h_i,f}^{2n_{\mathbf{w}_1}} \\ . & . & . & .. & .. & . \\ . & . & . & .. & .. & . \\ \mathbf{K}_{h_i,f}^{n1} & \mathbf{K}_{h_i,f}^{n2} & \mathbf{K}_{h_i,f}^{n3} & .. & .. & \mathbf{K}_{h_i,f}^{nn_{\mathbf{w}_1}} \end{pmatrix}_{n^2\times(n_{\mathbf{w}_1}*n_{\mathbf{w}_2})}. \quad (27)$$

Finally, the last term, $\langle \mu_{p_f^{des}}, \mu_{p_f^{des}} \rangle$ in (20) can be handled in a similar manner and thus, optimization (19a)-(19b) can be expressed as the following non-linear optimization problem

$$\min \rho_1(a_1u^4 + a_2u^3 + a_3u^2 + a_4u + a_5) + \rho_2 g(u). \quad (28a)$$

$$u \in \mathcal{F}. \quad (28b)$$

$$\begin{aligned}a_1 = \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_2}\mathbf{c}_{\alpha\beta}^T, a_2 &= \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_1}\mathbf{c}_{\alpha\beta}^T \\ a_3 = 2\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2h_0}\mathbf{c}_{\alpha\beta}^T + \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1h_1}\mathbf{c}_{\alpha\beta}^T &- 2\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_2f}\mathbf{c}_{\lambda\xi}^T \\ a_4 = 2\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1h_0}\mathbf{c}_{\alpha\beta}^T &- 2\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_1f}\mathbf{c}_{\lambda\xi}^T \\ a_5 = \mathbf{c}_{\alpha\beta}\mathbf{K}_{h_0h_0}\mathbf{c}_{\alpha\beta}^T - 2\mathbf{c}_{\alpha\beta}\mathbf{K}_{h_0f}\mathbf{c}_{\lambda\xi}^T &+ \mathbf{c}_{\lambda\xi}\mathbf{K}_{ff}\mathbf{c}_{\lambda\xi}^T.\end{aligned}$$

Computational Complexity The computational complexity of our proposed algorithm has two specific parts. The first part stems from the complexity of constructing the kernel matrix-like (24) used to formulate the cost function (28a). This in turn depends on the number of samples of the uncertain parameters $\mathbf{w}_1$, $\mathbf{w}_2$. In the worst case, we require n^2 samples. However, as explained in Section II-C, the value of n can be optimized using the reduced set methods.

The second part of the complexity stems from how difficult it is to solve the optimization (28a)-(28b). The optimization consists of a polynomial cost and convex feasibility constraints and thus can be solved with gradient based techniques such as projected gradient descent, sequential quadratic programming, etc. Furthermore, there exists a variety of software libraries like Scipy, which implement these optimization techniques. Contrast this with the mixed-integer optimizations obtained with SAA approach [18], which are not amenable to gradient based techniques. Furthermore, in [18], the number of binary variables is equal to the number of samples of the uncertainty parameters.

$$\mathbf{K}^{ab}_{h_i f} = \begin{pmatrix} k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^1), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^1, u_{nom})), & \dots & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^1), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^{n_{\mathbf{w}_2}}, u_{nom})) \\ k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^2), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^1), u_{nom})), & \dots & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^2), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^{n_{\mathbf{w}_2}}, u_{nom})) \\ \dots & \dots & \dots \\ k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^n), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^1, u_{nom})), & \dots & k(h_i(\mathbf{w}_1^a, \mathbf{w}_2^n), f(\widetilde{\mathbf{w}}_1^b, \widetilde{\mathbf{w}}_2^{n_{\mathbf{w}_2}}, u_{nom})) \end{pmatrix}_{n \times n_{w_2}}.$$

IV. Applications

In this section, we consider two robotic/control applications and model them in the form of the chance constrained optimization (1a)-(1c) and also present their RKHS reformulations. The motivation for this application have already been explained in Section I.

A. Dynamic Obstacle Avoidance along a Given Path

Here, we consider dynamic collision avoidance between a disk-shaped robot and non-reactive moving obstacles with similar shapes (Fig. 2(a)). Both the robot and the obstacles are assumed to have a single integrator motion model, i.e., they can instantaneously change their velocities. Further, we consider a variant of the problem where the path of the robot is fixed, and the robot achieves collision avoidance simply by varying the magnitude of its forward velocity. As shown in our earlier works [23], [24], the more general collision avoidance like [25], [26] can be conveniently built from this special case.

Let, (x, y) and $(\dot{x}, \dot{y})$ be the position and velocity vector of the robot at some specific time instant when the robot detects an imminent collision with the obstacles. Similarly, let (x_o, y_o) and $(\dot{x}_o, \dot{y}_o)$ represent similar vectors for the moving obstacle. It is clear that if the velocity vector of the robot is modified as $(u\dot{x}, u\dot{y})$, then it continues to move along its current path, although the magnitude of its forward velocity gets scaled by a factor u. For $u > 1$, the robot would increase its forward velocity while for $u < 1$, it would slow down to avoid collisions. Therefore the dynamic collision avoidance constraint can be written in the following form (refer to [24] for details).

$$\frac{(\mathbf{r}^T\mathbf{v})^2}{\|\mathbf{v}\|^2} - \|\mathbf{r}\|^2 + R^2 \leq 0. \tag{30a}$$

$$R = R + R_o. \tag{30b}$$

$$\mathbf{r} = \begin{bmatrix} x - x_o \\ y - y_o \end{bmatrix}, \mathbf{v} = \begin{bmatrix} u\dot{x} - \dot{x}_o \\ u\dot{y} - \dot{y}_o \end{bmatrix}. \tag{30c}$$

Here, R, R_o represent the radius of the footprint of the robot and the obstacle. Inequality (30a) can be put in the following more compact form, which resembles (11) with $l = 2$.

$$f(\mathbf{w}_1, \mathbf{w}_2, u) : h_0(\mathbf{w}_1, \mathbf{w}_2) + h_1(\mathbf{w}_1, \mathbf{w}_2)u + h_2(\mathbf{w}_1, \mathbf{w}_2)u^2 \leq 0 \tag{31}$$

where, $\mathbf{w}_1 = (x, y, \dot{x}, \dot{y})$ and $\mathbf{w}_2 = (x_o, y_o, \dot{x}_o, \dot{y}_o)$.

Uncertainty: Assume that the robot has both perception and ego-motion uncertainty in which case $\mathbf{w}_1, \mathbf{w}_2$ becomes random variables with some unknown distribution. The collision avoidance under uncertainty can be formulated through the following chance constrained optimization.

$$\min g(u) = (u-1)^2 \tag{32a}$$

$$P(f(\mathbf{w}_1, \mathbf{w}_2, u) \leq 0) \geq \eta \tag{32b}$$

$$u \geq 0 \tag{32c}$$

The cost (32a) minimizes the deviation from the current forward velocities. Optimization (32a)-(32c) fits in the form described by (1a)-(1c). After solving the above optimization problem or rather the RKHS embedding based reformulation of it, the robot draws a sample from its current velocity distribution $\dot{x}$, $\dot{y}$ and executes it after scaling by a factor u to avoid collisions.

Multiple moving obstacles: If there are multiple moving obstacles in the environment, then the parameter $\mathbf{w}_2$ needs to be computed specifically for each moving obstacle. That is, we have:

$${}^i\mathbf{w}_2 = ({}^i x_o, {}^i y_o, {}^i \dot{x}_o, {}^i \dot{y}_o)$$

Consequently, we will also have multiple collision avoidance constraints:

$$f(\mathbf{w}_1, {}^i\mathbf{w}_2, u) : h_0(\mathbf{w}_1, {}^i\mathbf{w}_2) + h_1(\mathbf{w}_1, {}^i\mathbf{w}_2)u + h_2(\mathbf{w}_1, {}^i\mathbf{w}_2)u^2 \leq 0. \tag{33}$$

The chance constrained optimization would now have multiple chance constraints and take the following form.

$$\min g(u) = (u-1)^2. \tag{34a}$$

$$P(f(\mathbf{w}_1, {}^i\mathbf{w}_2, u) \leq 0) \geq \eta, \forall, i = 1, 2..m. \tag{34b}$$

$$u \geq 0. \tag{34c}$$

Our RKHS embedding based reformulation would now have the following form:

$$\min \rho_1 \sum_i \overbrace{\|\mu_{p_{f_i}}(u) - \mu_{P^{des}_{f_i}}\|^2}^{MMD} + \rho_2 g(u). \tag{35a}$$

$$u \in \mathcal{F}. \tag{35b}$$

Here, $\mu_{p_{f_i}}(u)$ represents the KME of the i^{th} chance constraints (i.e with respect to each obstacle) and $\mu_{p^{des}_{f_i}}$ represents the KME of the desired distribution corresponding to the i^{th} chance constraints. Note that the first term in (35a) can be obtained using the derivations presented in Section III-F.

B. Inverse Dynamics based Path Tracking

In this application, we consider the task of tracking a reference trajectory $\mathbf{x}_d(t)$ by a manipulator (Fig. 2(c)), which can be framed as the following quadratic programming (QP) problem.

$$\arg\min_{\ddot{\mathbf{q}}(t)} \frac{1}{2}\|\mathbf{J}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \dot{\mathbf{J}}(\mathbf{q}(t), \ddot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) - \ddot{\mathbf{x}}(t)\|_2^2. \tag{36a}$$

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) \leq \tau_{max}. \quad (36b)$$

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) \geq -\tau_{max}. \quad (36c)$$

$$|\ddot{\mathbf{q}}(t)| \leq \ddot{\mathbf{q}}_{max}.. \quad (36d)$$

Here, $\ddot{\mathbf{x}}(t) = k_p(\mathbf{x}(t) - \mathbf{x}_d(t)) + 2\sqrt{k_p}(\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_d(t)) + \ddot{\mathbf{x}}_d(t)$ and k_p is a constant feedback gain. $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$ represents the joint angle and velocities at time t. Let the degree of freedom of the manipulator be m, i.e, $\mathbf{q}(t) = (q_1(t), q_2(t)...q_m(t))$. $\mathbf{J}$ is the manipulator Jacobian matrix. The inequalities (36b)-(36c) ensures that the resulting $\ddot{\mathbf{q}}(t)$ is achievable without violating the torque bounds. The QP (36a)-(36d) is solved in a one-step receding horizon setting for trajectory tracking. To be precise,the QP is solved for the joint accelerations at each instant considering the current joint position and velocities. The state is evolved with the current acceleration and the process is repeated for a specific time duration.

Constraints (36b)-(36c) represent $2m$ affine inequalities each of which can be represented in the following familiar form:

$$f_i(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2..u_n) = \sum_{j=1}^{j=m} h_i^j(\mathbf{w}_1.\mathbf{w}_2)u_j(t) + h_i(\mathbf{w}_1, \mathbf{w}_2) \leq 0 \quad \forall i = 1, 2..2m, \quad (37)$$

where, $\mathbf{w}_1 = (q_1(t), q_2(t)...q_m(t)), \mathbf{w}_2 = (\dot{q}_1(t), \dot{q}_2(t)...\dot{q}_m(t))$

$$(u_1, u_2..u_m) = (\ddot{q}_1(t), \ddot{q}_2(t)..\ddot{q}_m(t))$$

Trajectory Tracking under Perception Uncertainty As mentioned earlier, unlike industrial manipulators, the link configuration of cable-driven [5] and inflatable manipulators [6] cannot be precisely known by just the encoder readings. To perform torque based path tracking for these manipulators, we formulate a variant of inverse dynamics based path tracking where the manipulator has perfect motion capability but imperfect sensing for the joint angles $\mathbf{q}(t)$ and velocity $\dot{\mathbf{q}}(t)$ (Fig. 2(d)). In such a case, $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and functions $h_i^j(.)$ and $h_i(.)$ can be modeled as random variables. With this insight, we now formulate a stochastic variant of the inverse dynamics based path tracking problem as the following chance constrained optimization:

$$\arg\min_{\ddot{\mathbf{q}}(t)} \frac{1}{2}\|\mathbf{J}(\overline{\mathbf{q}}(t))\ddot{\mathbf{q}}(t) + \dot{\mathbf{J}}(\overline{\mathbf{q}}(t), \overline{\dot{\mathbf{q}}}(t))\dot{\mathbf{q}}(t) - \ddot{\mathbf{x}}(t)\|_2^2. \quad (38a)$$

$$P(f_i(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2..u_n) \leq 0) \geq \eta. \quad (38b)$$

$$|\ddot{\mathbf{q}}(t)| \leq \ddot{\mathbf{q}}_{max}. \quad (38c)$$

Here, $\mathbf{J}(\overline{\mathbf{q}}(t))$ and $\dot{\mathbf{J}}(\overline{\mathbf{q}}(t), \overline{\dot{\mathbf{q}}}(t))$ represents the Jacobian matrix formed with the mean variables $\overline{\mathbf{q}}(t)$ and $\overline{\dot{\mathbf{q}}}(t)$. The inequality (38b) ensures that the resulting $\ddot{\mathbf{q}}(t)$ can be achieved without violating the torque bounds with at least probability η. It can be seen that, (38a)-(38c) is an extended variant of the original chance constrained optimization (1a)-(1c). Specifically, we now have multiple decision variables along with multiple chance constraints.

Remark 3. *There is a subtle difference between the multiple chance constraints in optimization (34a)-(34c) and (38a)-(38c). In the former, multiple chance constraints arise because the parameters ${}^i\boldsymbol{w}_2$ were different for each obstacle while the function $f(.)$ remained the same for each constraint. In contrast, in the latter, the functions $f_i(.)$ were different for each constraint but the parameters $\boldsymbol{w}_1$, $\boldsymbol{w}_2$ remained same across different constraints.*

The RKHS embedding based reformulation of (38a)-(38c) takes the following form.

$$\min \rho_1 \sum_i \overbrace{\|\mu_{p_{f_i}}(u_1, u_2..u_n) - \mu_{p_{f_i}^{des}}\|^2}^{MMD} + \rho_2 g(u_1, u_2, ..u_n). \quad (39a)$$

$$|\ddot{\mathbf{q}}(t)| \leq \ddot{\mathbf{q}}_{max}. \quad (39b)$$

Here, $\mu_{p_{f_i}}(.)$ represents the KME of the i^{th} chance constraints and $\mu_{p_{f_i}^{des}}$ represents the KME of the desired distribution corresponding to the i^{th} chance constraint.

V. Results

In this section, we present simulations obtained by applying our formulation to the examples derived in the previous section. During each application, we also separately benchmark our formulation with some of the existing approaches for chance constrained optimization. Extra derivations and simulation videos for the results can be found in the supplementary material http://robotics.iiit.ac.in/uploads/Main/Publications/Bharath_journal/.

A. Collision Avoidance Results

1) Three Obstacle Benchmark with Non-Gaussian Uncertainty: Here we consider a benchmark where the robot needs to avoid collisions with three obstacles under Non-Gaussian perception and motion uncertainty. Fig.4(a) represents the configuration of the robot and the moving obstacles. At some specific time instant, Figs.4(b), 4(c) represent the uncertainty in the robot's and the obstacle's positions and velocities. Note the non-Gaussian nature of the position and velocity uncertainty. As shown in Section IV-A, the uncertainty in position and velocity can be mapped to uncertain parameters $\mathbf{w}_1, \mathbf{w}_2$ and consequently to functions $h_0(\mathbf{w}_1, \mathbf{w}_2)$, $h_1(\mathbf{w}_1, \mathbf{w}_2)$ and $h_2(\mathbf{w}_1, \mathbf{w}_2)$. We subsequently use this information to compute the collision avoidance velocity for the robot.

The solution process and results are summarized in Fig 5(a)-5(f). As described previously, the solution process starts with the construction of the desired distribution $p_{f_i}^{des}$ constructed corresponding to chance constraints formulated with respect to each obstacle [2]. Subsequently, we ensure that the tail of the distribution of $p_{f_i}(u)$ is similar to $p_{f_i}^{des}$ by choosing an appropriate u and the degree of the polynomial kernel d. The

[2]Recall that the parametric form for the desired distribution or even $p_f(u)$ is not known. But for illustration purposes, we can use the Kernel Density Estimation and empirical CDF methods to graphically represent the distribution in our plots

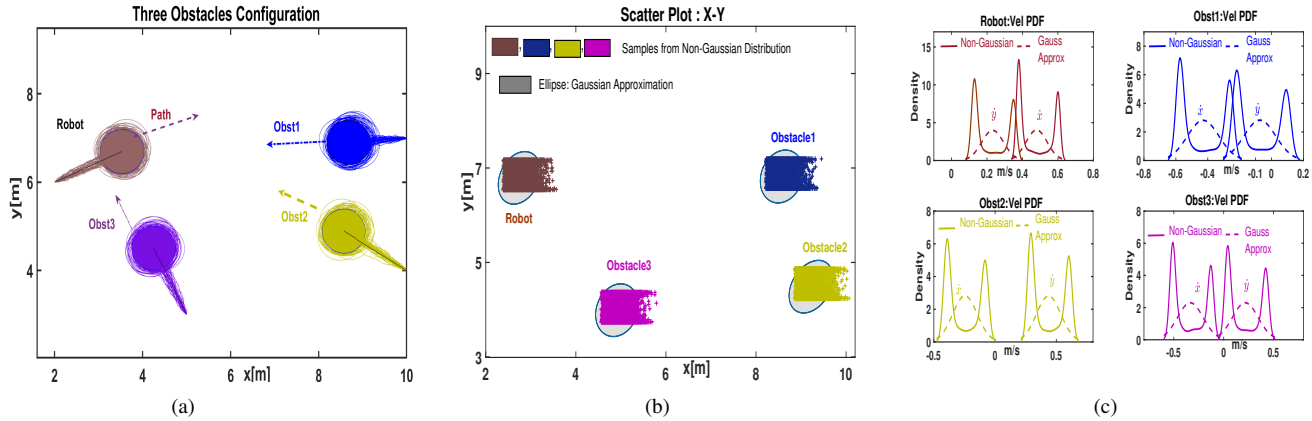


Fig. 4. (a): Collision avoidance scenario where the robot needs to avoid collision with three moving obstacles. (b): The position samples of robot and obstacles at some specific time instant when the robot detects an imminent collision with the obstacles. (c): The uncertainty in velocities of robot and obstacles. It is clear from the plots that they are non-Gaussian in nature, the Gaussian approximations of these distributions are also shown. The main intent of displaying the Gaussian distributions is to show how poorly they approximate the original non-Gaussian distributions.

following key points should be particularly noted from the plots. Fig.5(a)-5(c). As d increases more and more mass of $p_{f_i}(u)$ gets shifted to the right of $f_i(.) = 0$ leading to an increase of η. This can be further validated through the CDF plots. This validates our main idea of reformulating chance constrained optimization as a moment matching problem in RKHS (see section III-D).

The improvement in collision avoidance probability with an increasing value of d is also presented in Figs.5(d), 5(e), 5(f) via a comparison of the position samples from where the robot can either collide with (black) or avoid (red) the obstacles. Snapshots from collision avoidance simulations are shown in Figures 6(a)-6(h). It is easy to relate these snapshots to the position samples from figures 5(d), 5(e), 5(f). As the value of d increases, the robot chooses a velocity that results in more and more clearance with the obstacles. This is what results in reduction of colliding samples in Figs.5(d), 5(e), 5(f).

2) Cost-Robustness Trade-off: As mentioned in the Section III-D, ρ_1 and ρ_2 trades-off robustness with cost. For a given ρ_1, increasing ρ_2 would bias the optimization towards minimizing the primary cost $g(u)$ and thus decreasing η. This is shown in Fig.7(a)-7(c). Alternately, for a fixed ρ_2, increasing ρ_1 would prioritize minimizing the MMD cost which in turn improve the robustness or η value.

3) Quantitative Benchmarking on Collision Avoidance: Table II shows a comparison of the number of samples required by different approaches to compute an optimal solution such that the chance constraints are satisfied with a specified η.The following points can be noted from the table

- As expected, a naive implementation of the scenario approach shows the worst sample complexity. For $\eta \approx 0.7$, we required 200 samples each of $\mathbf{w}_1, \mathbf{w}_2$ leading to a grid of size $4 * 10^4$. For $\eta \approx 0.9$, we required 500 samples each of $\mathbf{w}_1, \mathbf{w}_2$.
- The SAA approximation proposed in [18] required a sample size almost half of that required by the scenario approach. For $\eta \approx 0.7$, we needed 100 samples each of $\mathbf{w}_1, \mathbf{w}_2$. This requirement increased to 200 for $\eta \approx 0.9$.
- The approach of [13], [14] which is based on surrogate constraints 3 shows an interesting trend. The sample complexity is worse than scenario approach for $\eta \approx 0.7$. However, the sample size does not vary with η. This is because the samples of the uncertain parameters are used to obtain an estimate of $E[f(\mathbf{w}_1, \mathbf{w}_2, u)]$ and $\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u)]}$ and importantly, this estimation is independent of η.
- As can be seen from Table II, our proposed formulation based on RKHS embedding has significantly better sample complexity than all the above-discussed approaches. It required 20 samples each of $\mathbf{w}_1, \mathbf{w}_2$ to construct a reasonable estimate of the desired distribution. An additional 20, 40 samples were required to construct the RKHS embedding based reformulations at $\eta \approx 0.7$ and $\eta \approx 0.9$ respectively.

Figs.8(a), 8(b) compare the optimal cost obtained through different formulations. The following important observations can be drawn from it

- Our proposed formulation results in lower cost solutions than approaches based on scenario approximation and surrogate constraints (3) [13], [14]. The difference is more pronounced for non-Gaussian uncertainty and at higher η. In fact, at a higher η, the approach based on (3) often runs into infeasibility.
- Interestingly, the SAA approach of [18] results in very similar costs to those of our proposed formulation for both Gaussian and non-Gaussian uncertainty. This is not surprising as SAA proposed in [18] is indeed a very tight approximation of the chance constraints.

Computational time: Tables III and IV show that our proposed formulation outperforms existing works in terms of computation time as well. The best factor of improvement comes with respect to the approach based on surrogate con-

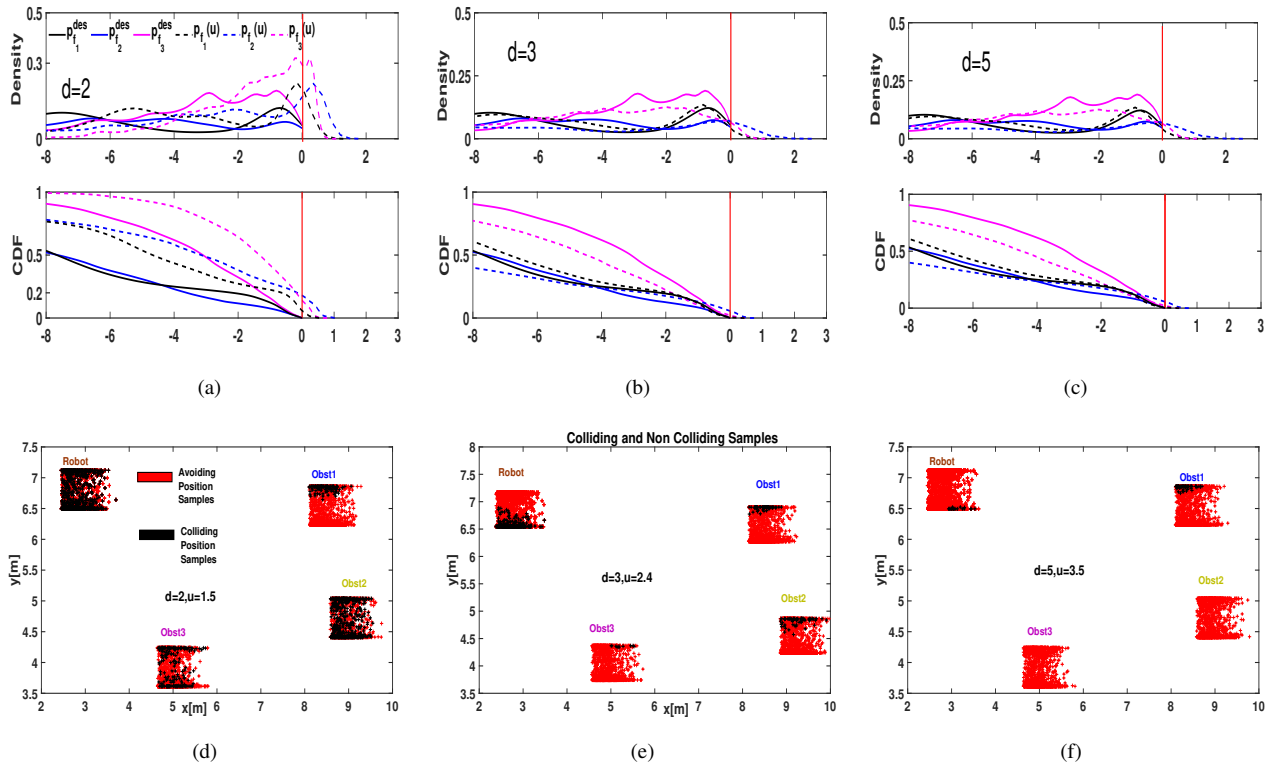


Fig. 5. Figures show the simulation results for collision avoidance with three moving obstacles shown in Fig.4(b) under non-Gaussian uncertainty. In this example, we have multiple chance constraints $P(f_i(\mathbf{w}_1, {}^i\mathbf{w}_2, u) \leq 0) \geq \eta$ because the uncertain parameter $\mathbf{w}_2$ was different for each obstacle. Thus, as shown in Figures (a), (b), (c), we need to construct three different desired distributions $p_{f_i}^{des}$ corresponding to chance constraints formulated with respect to each obstacle. As seen in previous examples, an increase in the degree of the polynomial kernel d leads to the increase in the portion of the mass of p_{f_i} to the left of $f(.) = 0$. Figures (d), (e), (f) validate the reduction of the colliding samples with an increase in d.

TABLE II
TABLE SUMMARIZING SAMPLE COMPLEXITY FOR COLLISION AVOIDANCE APPLICATION.

Approach	$P(f(\mathbf{w}_1, \mathbf{w}_2, u) < 0) \approx 0.7$	$P(f(\mathbf{w}_1, \mathbf{w}_2, u) < 0) \approx 0.9$
Scenario	$\mathbf{w}_1, \mathbf{w}_2 = 200$	$\mathbf{w}_1, \mathbf{w}_2 = 500$
SAA [18]	$\mathbf{w}_1, \mathbf{w}_2 = 100$	$\mathbf{w}_1, \mathbf{w}_2 = 200$
$E[f(\mathbf{w}_1, \mathbf{w}_2, u)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u)]} \leq 0$	$\mathbf{w}_1, \mathbf{w}_2 = 800$	$\mathbf{w}_1, \mathbf{w}_2 = 800$
Proposed RKHS embedding	$\mathbf{w}_1, \mathbf{w}_2 = 20$, $\widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 20$	$\mathbf{w}_1, \mathbf{w}_2 = 40$, $\widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 20$

straints (3). Furthermore, our formulation shows a moderate increase, with the number of obstacles. Note, that since the considered collision avoidance application has just one decision variable u (increase or decrease of forward speed), we sampled u within some fixed bounds in order to solve SAA based formulation. This allowed us to avoid the mixed-integer optimization proposed in [18].

B. Path Tracking Results for a 2 link Manipulator

Recall that in this application, we repeatedly solve the chance constrained optimization (38a)-(38c) or rather the reformulation of it (39a)-(39b) and evolve the joint angles and velocities according to the computed acceleration control input at each iteration. Moreover, we have multiple chance constraints $P(f_i(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2) \leq 0) \geq \eta$ and thus, a desired distribution $p_{f_i}^{des}$ needs to be constructed corresponding to each of them. Fig.9(a), 9(b) show the distributions $p_{f_i}^{des}$ and $p_{f_i}(.)$ (for one of the chance constraints) at iteration 60 and 69 for $d = 2$. Fig.9(c) shows the torque values obtained at each iteration. The lines in black represent the mean torque values while the cyan shows the uncertainty around it in the form of samples. Fig.9(d) shows the tracking performance in terms of path deviation and optimal cost values at each iteration. Comparing $p_{f_i}(.)$ and $p_{f_i}^{des}$ at both the iterations, it can be seen at iteration 60, the tails of the two distributions are more closely matched and as a result, a larger portion of $p_{f_i}(.)$ lies to the left of $f(.) = 0$. A direct consequence of this can be observed in the torque plots. At iteration 60, we observe fewer samples of torque that violate the torque bounds compared to what we observe at iteration 69.

1) Comparative Results for Path Tracking: We now compare our proposed RKHS based formulation with the scenario approach and the approach based on the surrogate constraint

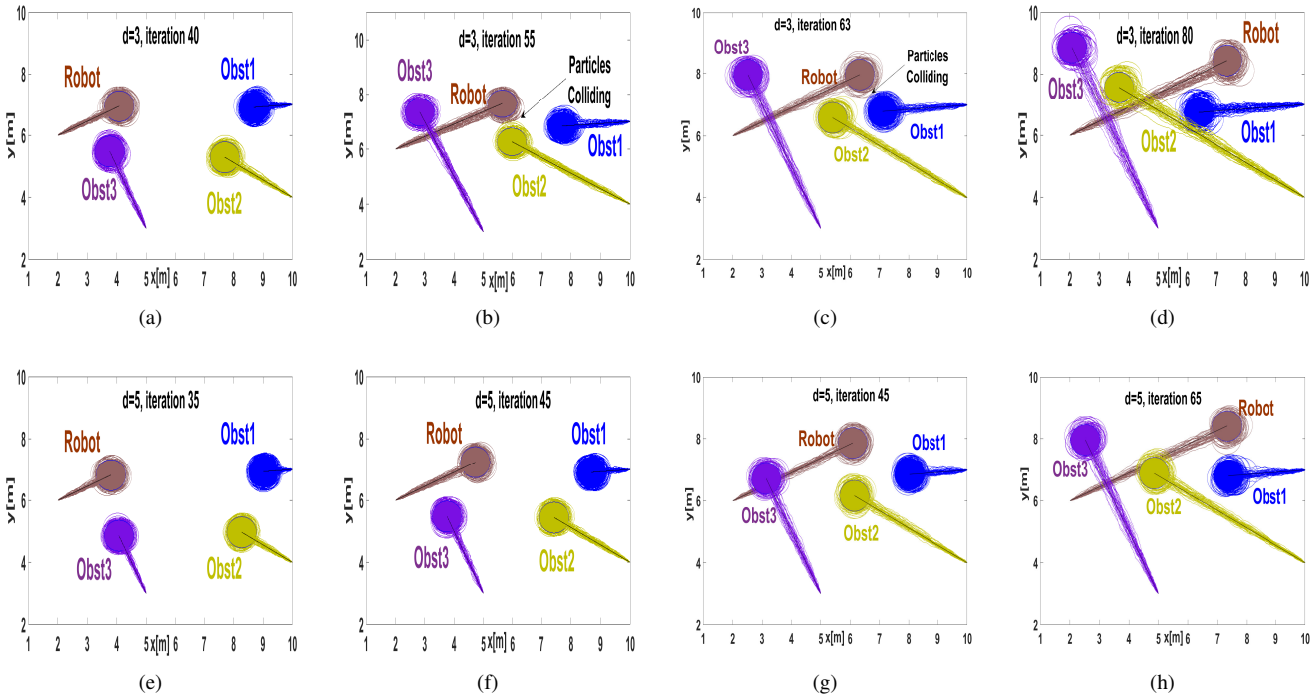


Fig. 6. Snapshots of collision avoidance simulation for $d = 3, 5$. Note how increase in d results in increase in clearance between the robot and the obstacles. The increased clearance translates to improvement in probability of collision avoidance.

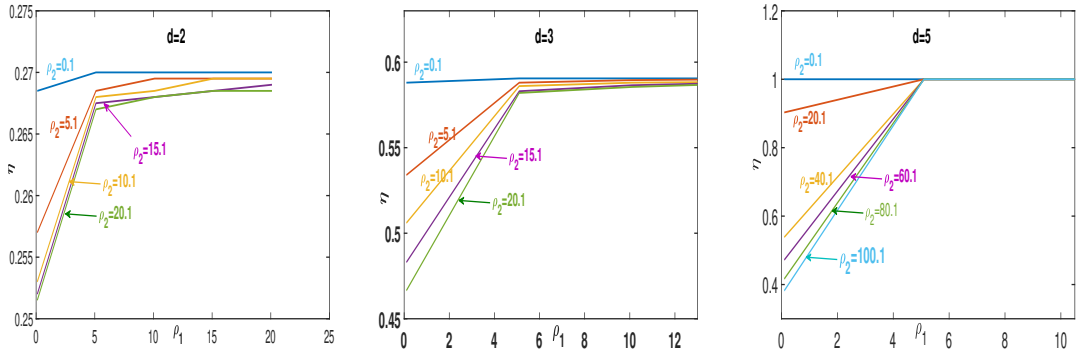


Fig. 7. Illustration of the effect of ρ_1, ρ_2 on η for collision avoidance application.

TABLE III
TABLE SUMMARIZING COMPUTATION TIME (S) FOR COLLISION AVOIDANCE APPLICATION: $P(f(\mathbf{w}_1, \mathbf{w}_2, u) < 0) \approx 0.7$

Approach	One Obstacle	Two Obstacles	Three Obstacles
Scenario	0.072	0.12	0.17
SAA [18]	0.11	0.13	0.2
$E[f(\mathbf{w}_1, \mathbf{w}_2, u)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u)]} \leq 0$	0.24	0.4	0.56
Proposed RKHS embedding	0.04	0.06	0.08

(3) in the context of the path tracking application. [3]. Table V and VI summarizes the sample complexity for $\eta \approx 0.7$ and $\eta \approx 0.9$ respectively for different values of τ_{max}. As can be seen, our RKHS based formulation enjoys better sample complexity than both the compared approaches in this

[3]We do not compare with the SAA approach of [18] here because its computational complexity on this application becomes too prohibitive. The collision avoidance application involved only one decision variable, and thus, we could do a brute force search to solve the SAA formulated problem. However, such an approach would not be suitable for the path tracking application. Authors in [18] suggest a mixed-integer reformulation, wherein the number of integer variables would be equal to the number of samples of the uncertain parameters. But, we remark that such a reformulation would be prohibitive for high dimensional robotic systems like manipulators.

TABLE IV
TABLE SUMMARIZING COMPUTATION TIME (S) FOR COLLISION AVOIDANCE APPLICATION: $P(f(\mathbf{w}_1, \mathbf{w}_2, u) < 0) \approx 0.9$

Approach	One Obstacle	Two Obstacles	Three Obstacles
Scenario	0.15	0.26	0.41
SAA [18]	0.15	0.41	0.6
$E[f(\mathbf{w}_1, \mathbf{w}_2, u)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u)]} \leq 0$	0.24	0.4	0.56
Proposed RKHS embedding	0.1	0.16	0.3

application too. The order of improvement increases with η. Moreover, at $\eta \approx 0.7$, an additional trend can be observed: the order of improvement also improves as the chance constrained optimization becomes tighter due to a decrease in τ_{max}. At $\eta \approx 0.9$, the order of improvement remains almost the same for various τ_{max}. It can also be noted that the sample complexity in this application is significantly lower than that observed in the previous collision avoidance application. We attribute this to the fact that $f_i(.)$ for path tracking application is affine in terms of decision variable (see (37)) while in collision avoidance application, it is a non-convex quadratic. Fig.10 shows the comparison of average optimal costs observed across 20 different problem instances. As can be seen, our RKHS based formulation produces significantly lower cost solutions, and the order of improvement increases with a decrease in τ_{max}.

Computational time: Tables VII and VIII compares the computation time our proposed RKHS based formulation with approaches based on scenario approximation and surrogate constraints (3). The improvement provided by our formulation is clearly visible.

C. Extension to three random variables

In this section, we show that our formulation can be easily extended to include additional random variables apart from $\mathbf{w}_1, \mathbf{w}_2$. In particular, we take the manipulator path tracking example and introduce additional random variable $\mathbf{w}_3$ to include the uncertainties in link lengths, mass, and inertia properties. Due to lack of space, we do not present the mathematical details, but it can be found in the supplementary material. We instead focus on the results which are summarized in the Fig.11(a)-11(b). In Fig.11(a), we show the increase in the optimal cost upon inclusion of more uncertainty in the problem through $\mathbf{w}_3$. Fig.11(b) shows that even with an increase with uncertainty, our RKHS embedding based approach outperforms existing methods like scenario approximation and approaches based on surrogate constraints (3).

D. Consistency and Sample Complexity

The RKHS embedding proposed in [19], [20], [27], [22] is consistent, i.e, it improves as the number of samples of $\mathbf{w}_1, \mathbf{w}_2$ increases. Further, these works relate estimation error with sample size. In this section, we show that we inherit the consistency guarantees in spite of the fact our RKHS embedding has an additional complexity of being parameterized in terms of control.

We begin by constructing a ground truth embedding in the following manner.

$$\overline{\mu_{p_f}(u)} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(f(\mathbf{w}_1^i, \mathbf{w}_2^j, u), .), \tag{40}$$

where, $\overline{\mu_{p_f}(u)}$ represents the same embedding as $\mu_{p_f}(u)$ (see 6) but is computed over a larger number of samples. That is, $N \gg n$. We can analyze the consistency by constructing the following error function from [19] for a fixed value of u.

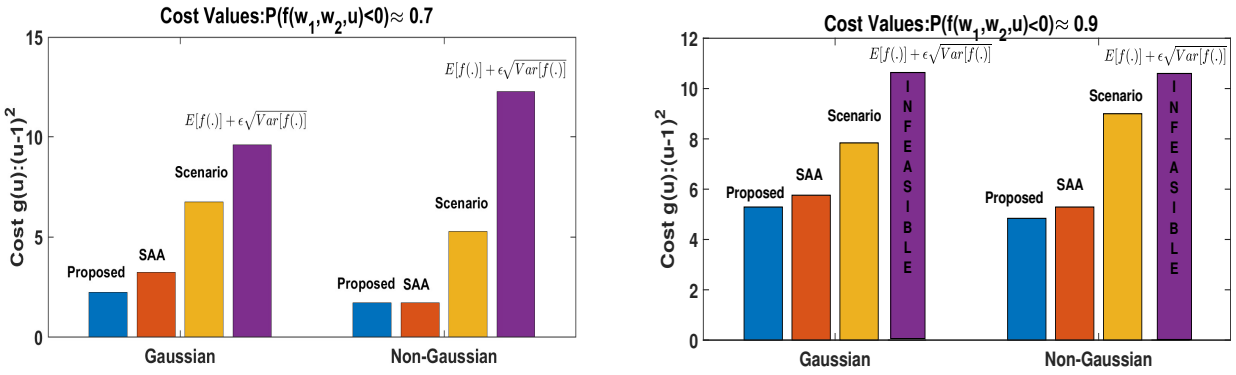
$$L = \|\mu_{p_f}(u) - \overline{\mu_{p_f}(u)}\|_2^2, \tag{41}$$

and analyzing its behavior for an increasing value of n. The results are summarized in Fig.12(a)-12(b) for (6)-(40) constructed with u_{nom}. As can be seen, the error reduces with an increase in the number of samples. For collision avoidance application (12(a)), we consistently get a very small error for samples as low as 40. For $d = 5$, the number of samples required for the same level of error is higher. The sample requirement for manipulator example (Fig.12(b)) follows a similar trend.

VI. CONCLUSION AND FUTURE WORK

Mathematical operations in RKHS have been the backbone for many of the modern machine learning algorithms. Examples of these span from kernel SVM to Gaussian Process. Recent trends in data science and programming languages widely advocate the use of probabilistic programming. Among the many existing approaches used in probabilistic programming, Hilbert space embedding of distributions has recently gained a lot of popularity. In fact literature along the lines of [19] even calls it Kernel Probabilistic Programming. We have adopted a series of recent papers [19], [20], [27] in this field that describes what a Hilbert space embedding of a function of random variables would actually mean. One of the key aspects of our work is connecting the theory of RKHS embedding of distributions to a widely studied problem of chance constrained optimization, which has applications in both robotics and control.

We formulated chance constrained optimization as a problem of matching higher order moments of two distributions. The eventual structure that our formulation takes is that of a non-linear optimization problem, which can be easily solved with the help of off-the-shelf solvers. We validated our formulation on applications like dynamic collision avoidance of mobile robots and path tracking of manipulators under torque



(a) (b)

Fig. 8. Average Optimal cost obtained with different methods for collision avoidance application observed across 20 different problem instances. Our RKHS formulation consistently results in lower cost solutions. The cost, in this case, models how much the robot needs to deviate from its current speed for collision avoidance. Furthermore, the approach based on surrogate constraints (3) often runs into in feasibility (inequality (3) infeasible) at higher η.

TABLE V
SAMPLE COMPLEXITY FOR PATH TRACKING APPLICATION CORRESPONDING TO $P(f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2) < 0) \approx 0.7$.

Approach	$\tau_{max} = \pm 8$	$\tau_{max} = \pm 5$	$\tau_{max} = \pm 3$
Scenario	$\mathbf{w}_1, \mathbf{w}_2 = 15$	$\mathbf{w}_1, \mathbf{w}_2 = 25$	$\mathbf{w}_1, \mathbf{w}_2 = 30$
$E[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)]} \leq 0$	$\mathbf{w}_1, \mathbf{w}_2 = 120$	$\mathbf{w}_1, \mathbf{w}_2 = 120$	$\mathbf{w}_1, \mathbf{w}_2 = 120$
Proposed RKHS formulation	$\mathbf{w}_1, \mathbf{w}_2 = 10, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 5$	$\mathbf{w}_1, \mathbf{w}_2 = 10, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 5$	$\mathbf{w}_1, \mathbf{w}_2 = 10, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 5$

TABLE VI
SAMPLE COMPLEXITY FOR PATH TRACKING APPLICATION CORRESPONDING TO $P(f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2) < 0) \approx 0.9$.

Approach	$\tau_{max} = \pm 8$	$\tau_{max} = \pm 5$	$\tau_{max} = \pm 3$
Scenario	$\mathbf{w}_1, \mathbf{w}_2 = 30$	$\mathbf{w}_1, \mathbf{w}_2 = 40$	$\mathbf{w}_1, \mathbf{w}_2 = 50$
$E[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)]} \leq 0$	$\mathbf{w}_1, \mathbf{w}_2 = 120$	$\mathbf{w}_1, \mathbf{w}_2 = 120$	$\mathbf{w}_1, \mathbf{w}_2 = 120$
Proposed RKHS formulation	$\mathbf{w}_1, \mathbf{w}_2 = 10, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 5$	$\mathbf{w}_1, \mathbf{w}_2 = 15, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 8$	$\mathbf{w}_1, \mathbf{w}_2 = 20, \widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2 = 8$

TABLE VII
COMPUTATION TIME (S) FOR PATH TRACKING APPLICATION CORRESPONDING TO $P(f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2) < 0) \approx 0.7$.

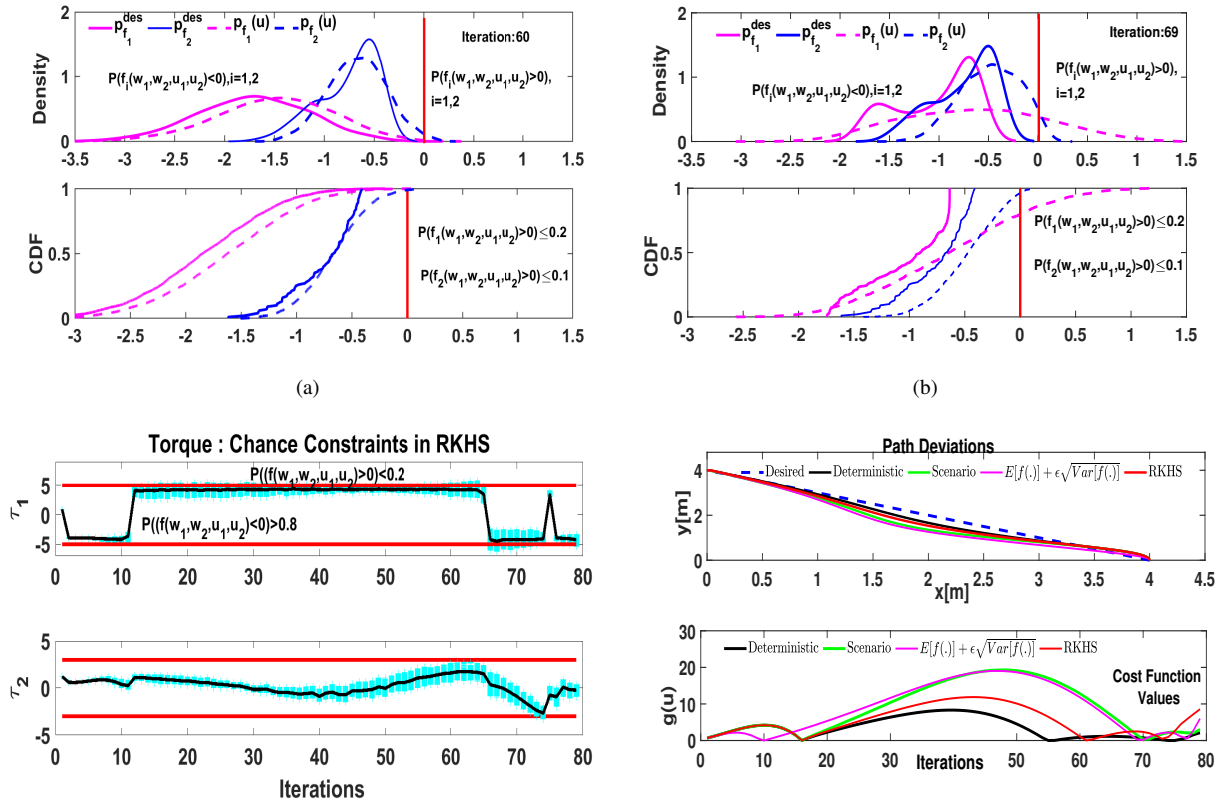
Approach	$\tau_{max} = \pm 8$	$\tau_{max} = \pm 5$	$\tau_{max} = \pm 3$
Scenario	0.062	0.08	0.1401
$E[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)]} \leq 0$	0.6	0.6	0.6
Proposed RKHS formulation	0.02	0.02	0.02

TABLE VIII
COMPUTATION TIME (S) FOR PATH TRACKING APPLICATION CORRESPONDING TO $P(f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2) < 0) \approx 0.9$.

Approach	$\tau_{max} = \pm 8$	$\tau_{max} = \pm 5$	$\tau_{max} = \pm 3$
Scenario	0.07	0.15	0.2
$E[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)] + \epsilon\sqrt{Var[f(\mathbf{w}_1, \mathbf{w}_2, u_1, u_2)]} \leq 0$	0.6	0.6	0.6
Proposed RKHS formulation	0.02	0.03	0.05

bounds. Our benchmarking clearly establishes the improvement that our formulation provides over existing approaches in terms of sample complexity and optimal cost.

At the moment, our formulation has some limitations, which we would be looking to rectify in our future works. Firstly, the cost function in our formulation is assumed to be deterministic, i.e., it does not contain the uncertain parameters. One simple way of rectifying this would be to formulate stochastic cost as constraints using some slack variables. We are currently evaluating the scalability of this idea. Secondly, we are working on benchmarking our formulation with approaches, which first fits some distribution to the non-parametric uncertainty and then performs the subsequent analysis. Examples of such fitting techniques include Gaussian Mixture Model, Kernel Density Estimator, Gaussian Process, etc. Finally, we are also looking at more complex applications like multi-agent navigation, reinforcement learning, etc.



(a) (b) (c) (d)

Fig. 9. The simulation results for inverse dynamics based path tracking under non-Gaussian uncertainty (Fig.2(d)). In this example, we repeatedly solve the optimization (39a)-(39b), formulated with polynomial kernel with $d = 2$. At each iteration, we need to construct a desired distribution corresponding to each chance constraint. Figures (a) and (b) show the desired distribution constructed at iteration 60 and 69. The figures also show the distribution of $f_i(.)$ for u_1, u_2 obtained as a solution to (39a)-(39b). Figure (c) shows the torque plots. The solid black lines represent the mean torque values, while the cyan lines show the uncertainty around it. Figure (d) shows the tracking performance in terms of path deviation and cost plot. Refer to the text for further insight.

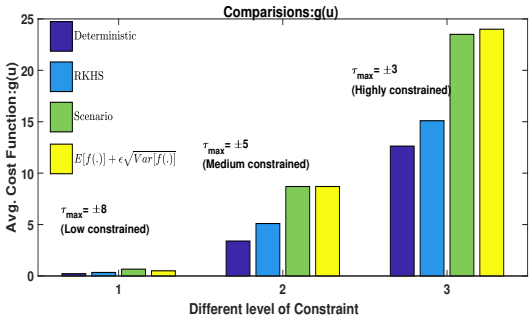


Fig. 10. Comparisons of optimal cost obtained for path tracking application with different approaches. A lower cost directly translates to better tracking while minimizing the control effort.

REFERENCES

[1] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.

[2] D. Lenz, T. Kessler, and A. Knoll, "Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles." in *Intelligent Vehicles Symposium*, 2015, pp. 292–297.

[3] M. P. Vitus and C. J. Tomlin, "A probabilistic approach to planning and control in autonomous urban driving," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 2459–2464.

[4] G. Schildbach and F. Borrelli, "Scenario model predictive control for lane change assistance on highways." in *Intelligent Vehicles Symposium*, 2015, pp. 611–616.

[5] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel *et al.*, "Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 532–539.

[6] S. Sanan, M. H. Ornstein, and C. G. Atkeson, "Physical human interaction for an inflatable manipulator," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 7401–7404.

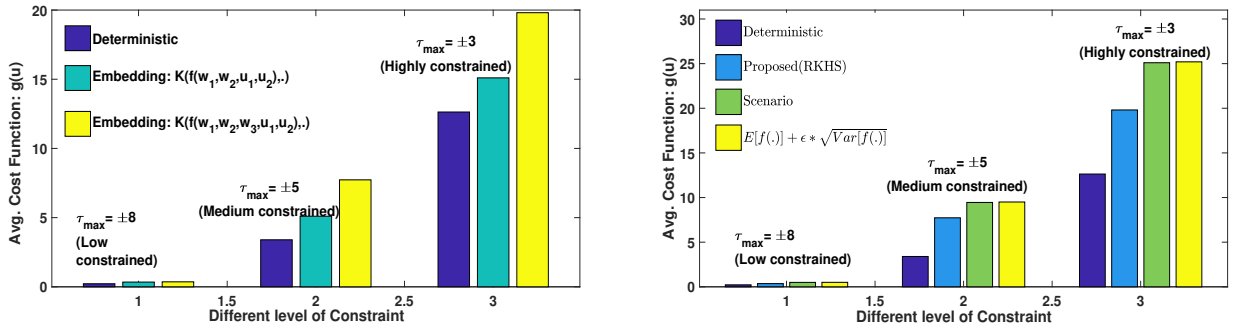
[7] S. Boyd, "Stochastic programming," *Lecture Notes, Stanford University*, 2008.

[8] A. Nemirovski and A. Shapiro, "Scenario approximations of chance constraints," *Probabilistic and randomized methods for design under uncertainty*, pp. 3–47, 2006.

[9] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

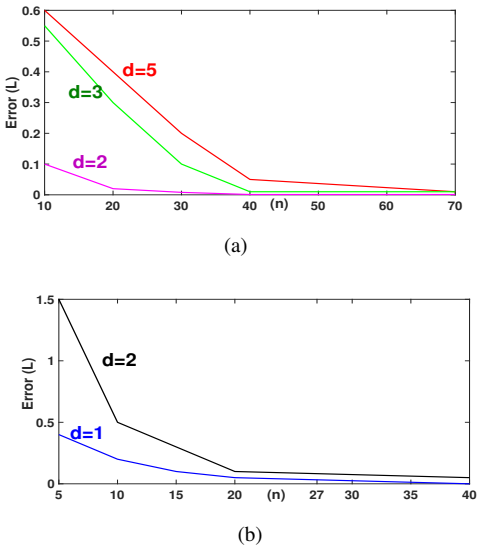
[10] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 219–224, 2013.

[11] M. C. Campi and S. Garatti, "A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality," *Journal of*



(a) (b)

Fig. 11. Figure (a) shows the effect of incorporating $\mathbf{w}_3$ into the proposed RKHS formulation. It can be clearly observed that the average cost incurred is high when $\mathbf{w}_3$ is incorporated into the formulation. Figure (b) shows that in spite of the increased uncertainty, our RKHS formulation still performs better than the other methods like the scenario approach and that based on surrogate constraints (3).



(a)

(b)

Fig. 12. Plots showing consistency of our RKHS embedding. As the number of samples increase, the error between the embedding constructed with the finite sample and the ground truth embedding reduces. Fig. (a) shows the consistency plots for collision avoidance application while (b) shows the same for manipulator path tracking application.

Optimization Theory and Applications, vol. 148, no. 2, pp. 257–280, 2011.

[12] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[13] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, "Stochastic nonlinear model predictive control with probabilistic constraints," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 2413–2419.

[14] B. Gopalakrishnan, A. K. Singh, and K. M. Krishna, "Closed form characterization of collision free velocities and confidence bounds for non-holonomic robots in uncertain dynamic environments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4961–4968.

[15] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2012.

[16] D. Lyons, J.-P. Calliess, and U. D. Hanebeck, "Chance-constrained model predictive control for multi-agent systems," *arXiv preprint arXiv:1104.5384*, 2011.

[17] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[18] B. Pagnoncelli, S. Ahmed, and A. Shapiro, "Sample average approximation method for chance constrained programming: theory and applications," *Journal of optimization theory and applications*, vol. 142, no. 2, pp. 399–416, 2009.

[19] B. Schölkopf, K. Muandet, K. Fukumizu, S. Harmeling, and J. Peters, "Computing functions of random variables via reproducing kernel hilbert space representations," *Statistics and Computing*, vol. 25, no. 4, pp. 755–766, 2015.

[20] A. Scibior, C.-J. Simon-Gabriel, I. O. Tolstikhin, and B. Schölkopf, "Consistent kernel mean estimation for functions of random variables," in *Advances in Neural Information Processing Systems*, 2016, pp. 1732–1740.

[21] B. G. Lindsay and P. Basak, "Moments determine the tail of a distribution (but not much else)," *The American Statistician*, vol. 54, no. 4, pp. 248–251, 2000.

[22] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1517–1561, 2010.

[23] B. Gopalakrishnan, A. K. Singh, and K. M. Krishna, "Time scaled collision cone based trajectory optimization approach for reactive planning in dynamic environments," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 4169–4176.

[24] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 1089–1096.

[25] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1928–1935.

[26] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[27] B. Schölkopf and S. AJ, "Learning with kernels: Support vector machines,regularization,optimization,and beyond." in *MIT press,Cambridge,MA,USA*. IEEE, 2002, pp. chapter 18,553–554.