Receding Horizon Differential Dynamic Programming Under Parametric Uncertainty

Yuichiro Aoyama\(^1,2\), Augustinos D. Saravanos\(^3\) and Evangelos A. Theodorou\(^1\)

Abstract—Generalized Polynomial Chaos (gPC) theory has been widely used for representing parametric uncertainty in a system, thanks to its ability to propagate uncertainty evolution. In an optimal control context, gPC can be combined with several optimization techniques to achieve a control policy that handles effectively this type of uncertainty. Such a suitable method is Differential Dynamic Programming (DDP), leading to an algorithm that inherits the scalability to high-dimensional systems and fast convergence nature of the latter. In this paper, we expand this combination aiming to acquire probabilistic guarantees on the satisfaction of nonlinear constraints. In particular, we exploit the ability of gPC to express higher order moments of the uncertainty distribution - without any Gaussianity assumption - and we incorporate chance constraints that lead to expressions involving the state covariance. Furthermore, we demonstrate that by implementing our algorithm in a receding horizon fashion, we are able to compute control policies that effectively reduce the accumulation of uncertainty on the trajectory. The applicability of our method is verified through simulation results on a differential wheeled robot and a quadrotor that perform obstacle avoidance tasks.

I. INTRODUCTION

One of the most challenging problems in the control field arises when a system operates under uncertainty. To address this issue, existing approaches can be classified into model-free and model-based ones. The former class has found several successful applications in the context of reinforcement learning (e.g., [1]–[3]), however main drawbacks of such methods include their requirement for numerous interactions with the physical system and slow convergence rates. On the other hand, methods that belong in the latter category (e.g., [4]–[6]) can be significantly faster, but their performance relies substantially on the accuracy of the model. Moreover, most of them, such as [7] and [8], utilize stochastic differential equations for representing uncertainty with Brownian motion which assumes Gaussianity. In addition, these methods are not taking into account any stochasticity arising from internal model parameters.

To address this problem, generalized Polynomial Chaos (gPC) theory [9] has been used for representing parametric uncertainty. This approach approximates stochastic processes using orthogonal polynomials which are chosen according to the type of distribution of the uncertain parameters. Therefore, a remarkable advantage of gPC is that it does not require any Gaussian assumptions on the dynamics. Moreover, by combining gPC with dynamical systems, it becomes possible to propagate uncertainty evolution and express higher order moments of its distribution.

From an optimal control perspective, gPC has found some applications in controlling systems under parametric uncertainty. In [10], a Model Predictive Control (MPC) with gPC approach is proposed, taking into account the expectations of constraints. Another gPC-MPC method is suggested in [11] for dealing with additive Gaussian noise, while satisfying linear chance-constraints. In [12], the authors developed an offset-free MPC based on gPC under linear constraints. By linearizing the constraints, [13] presents an effective way to address constrained MPC with gPC. Furthermore, an efficient method of solving the gPC-MPC problem was demonstrated in [14], by exploiting the second-order cone constraints that emerge from chance constraints. Finally, a chance-constrained stochastic nonlinear control approach with gPC was presented in [15], demonstrating a robot performing a reaching task while dodging an spherical obstacle.

Differential Dynamic Programming (DDP) is a trajectory optimization method that was first introduced in [4] and has found several successful applications such as [16]–[19] etc. Its main advantages include its quadratic convergence rate and its greater scalability to high-dimensional systems compared to other optimization techniques. For this reason, combining DDP with gPC [20] can lead to an algorithm that maintains these attributes, while being able to handle parametric uncertainty effectively.

In this paper, we extend the algorithmic framework suggested in [20], by proposing a receding horizon method that handles effectively parametric uncertainty and nonlinear constraints while maintaining proper convergence properties and scalability to high-dimensional systems. In particular, by taking advantage of our recent results on constrained DDP [21], we incorporate nonlinear chance constraints that through gPC theory yield deterministic constraints that involve the state covariance of the system. Moreover, in order to deal with the growth of uncertainty through the trajectory, we integrate feedback by executing our algorithm in a receding horizon scheme. As confirmed by the simulation results, our method can satisfy multiple nonlinear constraints under parametric uncertainty, while also being applicable to high-dimensional robotic tasks.

The remaining of the paper is organized as follows. A brief overview of gPC theory and DDP is provided in Section II. In Section III, we propose a receding horizon approach that combines DDP and gPC while satisfying nonlinear chance constraints. In Section IV, we verify the effectiveness of the suggested algorithm through simulation results on a

\(^1\)School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA
\(^2\)Komatsu Ltd., Tokyo, Japan
\{yaoyama3, asaravanos3, evangelos.theodorou\}@gatech.edu
differential wheeled robot and a quadrotor. The conclusions of our work along with future directions are provided in Section V.

II. PRELIMINARIES

In this section, we present some necessary preliminaries. We start with covering the theory of gPC and showing how it can represent dynamical systems with parametric uncertainty. Next, we give a concise overview of DDP and we describe how it can be applied to dynamical systems represented by gPC.

A. Generalized Polynomial Chaos

1) Polynomial Chaos Expansion: Let us define a probability space \((\Omega, \mathcal{F}, P)\) where \(\Omega\) is the sample space, \(\mathcal{F}\) is the \(\sigma\)-field of \(\Omega\), and \(P\) is the probability measure. Let \(\omega \in \Omega\) and \(\xi(\omega) = (\xi_1(\omega), \ldots, \xi_d(\omega)) \in \mathbb{R}^d\) be a continuous random variable vector with mutually independent components whose distributions are subject to the probability density function (PDF) \(\rho(\xi)\) with support \(I_\xi\). Furthermore, let \(F(\xi)\) be a function of the random variable \(\xi\) and \(L^2_\rho\) be the space of square integrable functions with norm \(\|F\|_{L^2_\rho} = \mathbb{E}[F^2]^{1/2}\), i.e.:

\[
L^2_\rho = \left\{ F : I_\xi \to \mathbb{R} \mid \int_{I_\xi} F^2(\rho(\xi)d\xi \to \infty \right\}.
\]

Then, the polynomial chaos expansion of \(F\) can be obtained as [22]:

\[
F(\xi) \approx \sum_{j=0}^{\infty} F_j \Phi_j(\xi).
\] (1)

The functions \(\Phi_j(\xi)\) in expression (1) are the generalized polynomial chaos basis functions, that satisfy orthogonality:

\[
\mathbb{E}[\Phi_m(\xi)\Phi_n(\xi)] = \int_{I_\xi} \Phi_m(\xi)\Phi_n(\xi)\rho(\xi)d\xi = \gamma\delta_{mn},
\]

where \(\gamma = \mathbb{E}[\Phi^2(\xi)] = \langle \Phi, \Phi \rangle\) are the normalization factors and \(\delta_{mn} = 0\) if \(m \neq n\), \(\delta_{mn} = 1\) if \(m = n\).

For several types of distributions, specific gPC basis functions exist for describing the underlying random variables of a given function. Some of them are listed in Table I.

In practical applications, a truncated version of the gPC expansion (1) can be used by taking into account polynomials of degree up to \(r\):

\[
F(\xi) \approx \sum_{j=0}^{K} F_j \Phi_j(\xi), \quad K = \frac{(r + d)!}{r!d!} - 1,
\]

where \(K\) is the total number of polynomials. The coefficients of this expansion can be obtained by exploiting the orthogonality of \(\Phi\). Taking the inner product of \(F\) and \(\Phi_j\) yields:

\[
F_j = \frac{\int_{I_\xi} F(\xi)\Phi_j(\xi)\rho(\xi)d\xi}{\int_{I_\xi} \Phi_j^2(\xi)\rho(\xi)d\xi},
\]

which is called Galerkin projection. Note that the gPC basis functions \(\Phi_j(\xi)\) are the products of the polynomials of mutually independent random variables with multi-index \(|i| = i_1 + \cdots + i_d\). More precisely, we have:

\[
\Phi_i(\xi) = \phi_{i_1}(\xi_1)\cdots\phi_{i_d}(\xi_d), \quad 0 \leq |i| \leq r.
\]

2) Dynamical systems with polynomial chaos: In order to apply DDP to a stochastic system represented by gPC, information about the evolution of the system including the dynamics and the Jacobian matrix is necessary. We consider the dynamical system whose state \(x(t) \in \mathbb{R}^n\) and initial state \(x(t_0) = x_0\) are influenced by the uncertain parameters \(\xi^p \in \mathbb{R}^d_p\) and \(\xi^0 \in \mathbb{R}^d_0\), respectively. It is also assumed that the distributions of these parameters are known. The control \(u(t) \in \mathbb{R}^m\) is set to be deterministic. The system dynamics \(f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n\) are provided by:

\[
\dot{x}(t) = f(x(t), u(t), t; \xi^p), \quad x(t_0) = x_0; \xi^0.
\] (2)

Applying the gPC expansion to random variables \(\xi^p\) and \(\xi^0\) with respect to the mutually independent standard random variables \(\xi^p\) and \(\xi^0\) yields:

\[
\xi^p_t \approx \sum_{j=0}^{Z} \xi^p_j \phi^p_j(\xi^p), \quad \xi^0_t \approx \sum_{j=0}^{Z_0} \xi^0_j \phi^0_j(\xi^0),
\]

where \(Z\) and \(Z_0\) are the orders of the polynomials. Likewise, the initial states are expanded as:

\[
x_i(t_0) \approx \sum_{j=0}^{K_0} x_{ij}(t_0) \Phi^0_j(\xi^0).
\]

In order to parameterize \(x\) with both \(\xi^p\) and \(\xi^0\), we define the concatenated vector of random parameters as \(\xi = (\xi^p, \xi^0) \in \mathbb{R}^d\), with \(d = d_p + d_0\). Since the evolution of \(x\) depends on \(\xi\), the gPC expansion of the state vector and its time derivatives are given by:

\[
x_i(t) \approx \sum_{j=0}^{K} x_{ij}(t) \Phi_j(\xi), \quad \dot{x}_i(t) \approx \sum_{j=0}^{K} \dot{x}_{ij}(t) \Phi_j(\xi).
\]

Applying Galerkin projection to these equations yields the coefficients:

\[
x_{ij} = \frac{\int_{I_\xi} x_i(t) \Phi_j(\xi)d\xi}{\langle \Phi_j, \Phi_j \rangle}, \quad \dot{x}_{ij} = \frac{\int_{I_\xi} \dot{x}_i(t) \Phi_j(\xi)d\xi}{\langle \Phi_j, \Phi_j \rangle}.
\]

Therefore, the evolution of the system can be obtained as an evolution of gPC coefficients, i.e.:

\[
\dot{X}(t) = f(X(t), u(t), t),
\]

where \(X = (x_{10}, \ldots, x_{1K}, \ldots, x_{nK}) \in \mathbb{R}^{n(K+1)}\).

| Distribution | PDF: \(\rho\) | gPC basis polynomials |
|--------------|-------------|---------------------|
| Gaussian     | \(\mathcal{N}(\mu, \sigma^2)\) | Hermite             |
| Beta         | \(\text{Beta}(\alpha, \beta)\) | Jacobi              |
| Uniform      | \(\text{Uniform}(a, b)\) | Legendre            |
| Gamma        | \(\text{Gamma}(\alpha, \beta)\) | Laguerre            |

TABLE I

CORRESPONDENCE BETWEEN THE TYPE OF UNDERLYING RANDOM VARIABLES AND THE TYPE OF GPC BASIS FUNCTIONS.
Finally, differentiating the time derivative of the coefficients with respect to the coefficients of $x$ and $u$ yields the elements of the Jacobian matrix:

$$\frac{\partial x_{ij}}{\partial x_{gh}} = \int_{t_e} \frac{\partial x_{ij}}{\partial x_{gh}} \Phi_j(\xi) d\xi = \int_{t_e} \frac{\partial x_{ij}}{\partial x_{gh}} \Phi_k(\xi) \rho(\xi) d\xi = \frac{\int_{t_e} \frac{\partial x_{ij}}{\partial x_{gh}} \Phi_h(\xi) \Phi_j(\xi) \rho(\xi) d\xi}{\langle \Phi_j, \Phi_j \rangle},$$

$$\frac{\partial x_{ij}}{\partial u_k} = \int_{t_e} \frac{\partial x_{ij}}{\partial u_k} \Phi_j(\xi) \rho(\xi) d\xi = \frac{\int_{t_e} \frac{\partial x_{ij}}{\partial u_k} \Phi_j(\xi) \rho(\xi) d\xi}{\langle \Phi_j, \Phi_j \rangle},$$

with $i, g = 1, \ldots, n$, $j, h = 0, \ldots, K$, $k = 1, \ldots, m$.

The integrals can be numerically evaluated online by Gaussian quadrature. The nodes and weights for the computation can be obtained by solving an eigenvalue problem [23].

Once the gPC expansion of $x_i(\xi)$ is obtained, the information can be computed analytically. The mean, variance, and covariance are given by:

$$\mathbb{E}[x_i(t)] = \int_{t_0}^{t_f} \sum_{k=1}^{K} x_{ij}(t) \Phi_j(\xi) \rho(\xi) d\xi = x_{i0}(t),$$

$$\text{VAR}[x_i(t)] = \sum_{j=1}^{K} x_{ij}^2(t) \langle \Phi_j, \Phi_j \rangle,$$

$$\text{COV}[x_i(t), x_j(t)] = \sum_{j=1}^{K} x_{ij}(t) x_{jj}(t) \langle \Phi_j, \Phi_j \rangle.$$

### B. Differential Dynamic Programming

1) Deterministic DDP: Next, we provide a brief overview of discrete-time DDP. For more details the reader can be referred to [4], [5]. Let us consider the discrete-time finite-horizon optimal control problem:

$$\min_{U} J(X_d, U) = \min_{U} \left[ \sum_{k=0}^{N-1} l(x_k, u_k) + \phi(x_N) \right]$$

subject to $x_{k+1} = f(x_k, u_k)$, $k = 0, \ldots, N-1$, where $x_k = x_t(k)$, $u_k = u_t(k)$ and $f$ is the transition dynamics function. The scalar functions $l, \phi$ and $J$ correspond to the running, terminal, and total cost, respectively. With $X_d := (x_0, \ldots, x_k)$ and $U := (u_0, \ldots, u_{N-1})$, we denote the state and control sequences, respectively, over the time horizon.

Next, the notion of the value function is introduced:

$$V_k(x_k) := \min_{u_k} J(X_d, U),$$

which provides the minimum cost-to-go at each state and time. The Bellman’s principle of optimality can be expressed through the following backward propagation rule:

$$V_k(x_k) = \min_{u_k} l(x_k, u_k) + V_{k+1}(x_{k+1}).$$

Let us also define the following function:

$$Q_k(x_k, u_k) := l(x_k, u_k) + V_{k+1}(x_{k+1}),$$

which is the quantity to be minimized in (4). During the backward pass of DDP, problem (3) is solved locally by expanding both sides of (4) about some given nominal trajectories $\bar{X}_d, \bar{U}$. By taking the quadratic expansions of $Q_k$ around the vicinity of $\bar{X}_d, \bar{U}$, we get:

$$Q_k(x_k, u_k) \approx Q_k + Q_k^T \delta x_k + \frac{1}{2} \delta x_k^T Q_{xx,k} \delta x_k + \delta u_k^T Q_{uu,k} \delta u_k,$$

where:

$$Q_{xx,k} = l_{xx} + f_x^T V_{xx,k+1} f_x, \quad Q_{uu,k} = l_{uu} + f_u^T V_{uu,k+1} f_u,$$

and $\delta x_k := x_k - \bar{x}_k$, $\delta u_k := u_k - \bar{u}_k$ being the deviations about the nominal sequences. Note that in (5), the $Q$ functions are evaluated on $\bar{X}_d, \bar{U}$. Given the expressions (5), we can now explicitly minimize (3) with respect to $\delta u$ in order to acquire the locally optimal control deviations as:

$$\delta u_k^* = K_k \delta x_k,$$

where $k := -Q^{-1}_{uu} Q_{u,k}$, $K := -Q^{-1}_{uu} Q_{u,x}$.

Moreover, the value function $V_k$ is also quadratically expanded and by plugging it into (4) along with $\delta u_k^*$, we get:

$$V_{x,k} = Q_{x,k} - Q_{x,u,k} Q^{-1}_{uu,k} Q_{u,k},$$

The equations (6) are propagated backwards in time using the terminal condition $V(x_N) = \phi(x_N)$.

Subsequently, in the forward pass, the new control sequence is applied to the system. The resulting trajectories will be used as the nominal ones at the next backward pass leading to an iterative process, which is terminated with the satisfaction of some predefined convergence criteria.

Regarding the cost functions, control effort and deviation from the desired state $x_d$ are typically penalized, i.e.:

$$l(x_k, u_k) = \frac{1}{2} (x_k - x_d^k)^T A (x_k - x_d^k) + \frac{1}{2} u_k^T R u_k,$$

$$\phi(x_N) = \frac{1}{2} (x_N - x_d^N)^T A_1 (x_N - x_d^N),$$

where $A_1, R$ are positive definite matrices and $A$ is a positive semidefinite matrix.

2) DDP with gPC: We will now demonstrate how unconstrained DDP can be combined with gPC. Since the dynamics are required to be discretized for applying discrete-time DDP, we can apply Euler discretization on them. Moreover, in order to obtain the optimal control of a system represented by gPC, the expectation of the cost can be used [24]. Taking the expectation of the running cost of the discretized system yields:

$$\mathbb{E}[l(X_k, u_k)] = \frac{1}{2} (X_k - X_d^k)^T A X (x_k - x_d^k) + \frac{1}{2} u_k^T R u_k,$$

with $X_d^k = (x_d^k, 0_{1 \times K}, x_d^{k+1}, 0_{1 \times K}, \ldots, x_d^{k+N-1}, 0_{1 \times K}) \in \mathbb{R}^{(K+1)},$

$$A X = A \otimes \text{diag}((\Phi_0, \Phi_0), (\Phi_1, \Phi_1), \ldots (\Phi_K, \Phi_K)),$$

where $X_d^k$ is the expanded desired state $x_d$ and $\otimes$ stands for the Kronecker product. Note that since $x_d$ is deterministic,
\( X_d \) only contains terms corresponding to the mean of \( x_d \) and zeros for higher order moments. The expectation of the terminal cost can be simplified using the same procedure. In order to penalize not only the mean but also the higher order moments of the state, \( A \) can be modified by taking into account elements corresponding to the higher order moments \((x_{ij}, i \neq 0)\) as:

\[
A^X = \begin{bmatrix}
A_1 & \ldots & 0_{K+1} \\
\vdots & \ddots & \vdots \\
0_{K+1} & \ldots & A_n
\end{bmatrix}
\]

with \( A_i = \text{diag}(a_{i0}, a_{i1}(\Phi_1, \Phi_1), \ldots, a_{iK}(\Phi_K, \Phi_K)) \). More details on the cost derivation can be found in [20].

### III. Receding Horizon Constrained DDP Under Parametric Uncertainty

#### A. Problem Formulation

Let us consider the following finite-horizon optimal control problem under parametric uncertainty and subject to control and state constraints:

\[
\begin{align*}
\min_U J(X_d, U) &= \min_U \left[ \sum_{k=0}^{N-1} l(x_k, u_k) + \phi(X_N) \right] \\
\text{subject to} & \quad x_{k+1} = f(x_k, u_k; \zeta), \\
& \quad P[g(x_k, u_k) \leq 0] > p_c, \\
& \quad u_{\min} \leq u_k \leq u_{\max}, \\
& \quad k = 0, \ldots, N-1.
\end{align*}
\]

The chance constraint \((7c)\) indicates that \( g(x_k, u_k) \leq 0 \) is satisfied with a probability greater than a specified value \( p_c \). By taking the gPC expansion of the dynamics and the expectation of the performance index, problem \((7)\) can be transformed into the following deterministic one:

\[
\begin{align*}
\min_U J(X, U) &= \min_U \mathbb{E}\left[ \sum_{k=0}^{N-1} l(x_k, u_k) + \phi(X_N) \right] \\
\text{subject to} & \quad X_{k+1} = f(x_k, u_k), \\
& \quad g_k(x_k, u_k) < \alpha, \\
& \quad u_{\min} \leq u_k \leq u_{\max}, \\
& \quad k = 0, \ldots, N-1,
\end{align*}
\]

where \( X := (X_0^T, \ldots, X_N^T) \). The chance constraint \((7c)\) has now been transformed into the deterministic \((8c)\) by using the appropriate function \( g_k \) and constant \( \alpha \). The choice of these two is further explained during the simulation examples in Section [IV].

#### B. Constrained DDP with Generalized Polynomial Chaos

After expressing the originally stochastic problem \((7)\) in a deterministic manner, we are now able to solve it using a deterministic constrained trajectory optimization method such as Augmented Lagrangian (AL) DDP [21]. From now on, we will refer to the resulting approach as gPC Constrained DDP (gPC CDDP). The constraints \((8c)\) are first, redefined as \( G = g_k - \alpha < 0 \), where \( G = (G_1, \ldots, G_w)^T \) is a vector of \( w \) constraints, and then, integrated into the original objective function by adding a function \( P \) which penalizes their violation. Therefore, problem \((8)\) can now be expressed as:

\[
\begin{align*}
\min_U \quad & \left[ J(X, U) + \sum_{k=0}^{N-1} \sum_{i=1}^{w} P(x_k, \mu_i^k, G_i(x_k, u_k)) \right] \\
\text{subject to} & \quad X_{k+1} = f(x_k, u_k), \\
& \quad u_{\min} \leq u_k \leq u_{\max}, \\
& \quad k = 0, \ldots, N-1,
\end{align*}
\]

where \( \lambda \) and \( \mu \) are the Lagrange multipliers and penalty parameters, respectively. It is known that \((9)\) can provide a solution to \((8)\) under mild assumptions [25].

The optimization process of AL DDP consists of two stages: an inner and an outer loop. In the inner loop, a locally optimal solution of \((9)\) is obtained by applying DDP. In the outer loop, the parameters \( \lambda \) and \( \mu \) are updated according to the value of the derivative of \( P \) and \( P \) itself, respectively. As for \( \mu \), they are increased monotonically, given the constraint improvement from the inner loop is not enough. More details regarding the choice of \( P \) can be found in [25], [26].

AL DDP is quite robust as a trajectory optimization method, reaching the desired state at an early stage of the optimization iterations. However, since the constraints \((8c)\) are treated as soft constraints - as they are part of the cost - they may be slightly violated. On the other hand, control constraints can be strictly satisfied by using control-limited DDP [27] in the inner loop instead of unconstrained DDP. Thus, we use a combination of AL and control-limited DDP similar to the one presented in [21]. The slight constraint violation that we mentioned can be observed especially when the horizon of the trajectory is quite long or the number of the constraints is large. In our receding horizon approach presented in [III-C] this drawback is alleviated by taking into account a shorter prediction horizon which leads to a smaller number of constraints.

#### C. Receding Horizon Constrained DDP

Applying gPC CDDP directly can be effective, but it may suffer from the accumulation of state variance on the trajectory. This variance growth can result in violating the chance constraints and being unable to reach the desired state target, as it is illustrated in Section [IV]. In order to deal with this problem, we implement our proposed method in a receding horizon fashion. An similar MPC variant of DDP has been presented in [28], but in a fully deterministic setting. In the parametric uncertainty case, the benefits of following an MPC approach are mainly the following. First of all, by taking into account the current state during the computation of each MPC control sequence, we integrate feedback into our method. Second, by considering the state evolution only during the prediction horizon \( H \) - which is shorter than the whole time horizon \( N \) - the growth of the variance in our computations can be maintained to lower levels. Finally, by addressing a shorter trajectory optimization problem than the original one, the constraints taken into account by gPC CDDP will be reduced.
polynomials are chosen - according to Table I - to describe on a differential wheeled robot while explaining in detail its effectiveness of our approach. Initially, we apply the method a desired target while avoiding obstacles. Its model contains three uncertain parameters: the tread \( d \) and the wheel radii \( r_R \) and \( r_L \), all of which are assumed to be normally distributed with means 0.2 and variances \( 1.5 \times 10^{-3} \). Hermite polynomials are chosen - according to Table I - to describe these parameters. The first few of them, their inner products, and the PDF are given by:

\[
\phi_0(z) = 1, \quad \phi_1(z) = z, \quad \phi_2(z) = z^2 - 1, \ldots,
\]

\[
\langle \phi_i, \phi_j \rangle = \delta_{ij}!^2, \quad p(z) = \frac{1}{\sqrt{2\pi}} \exp{-z^2/2}.
\]

Subsequently, the parameters, e.g. \( d \sim \mathcal{N}(\mu_d, \sigma_d^2) \), can be expanded by the standard random variable \( \xi_d \sim \mathcal{N}(0, 1) \), which will be an element of the random vector \( \xi \in \mathbb{R}^3 \), i.e.:

\[
d(\xi_d) = \mu_d \phi_0(\xi_d) + \sigma_d \phi_1(\xi_d) = \mu_d + \sigma_d \xi_d.
\]

The dynamics of the robot are provided by:

\[
\begin{bmatrix}
x_{k+1} \\
y_{k+1} \\
\theta_{k+1}
\end{bmatrix} =
\begin{bmatrix}
x_k \\
y_k \\
\theta_k
\end{bmatrix} +
\begin{bmatrix}
v \cos \theta_k \\
v \sin \theta_k \\
\omega
\end{bmatrix} dt,
\]

where \( x, y \) are its position coordinates, \( \theta \) is its orientation, \( v \) and \( \omega \) are its translational and rotational velocities, respectively, and \( dt \) is the time step. The control inputs of the robot are the rotational velocities of its two wheels: \( u_1 = \phi_R \) and \( u_2 = \phi_L \). The translational velocities of the wheels can be obtained from:

\[
v_R = r_R u_1, \quad v_L = r_L u_2,
\]

while \( v \) and \( \omega \) are given by:

\[
v = \frac{1}{2}(v_R + v_L), \quad \omega = \frac{v_R - v_L}{2d}.
\]

In this setting, chance constraints can be formulated by imposing the probability that the robot will avoid colliding with an obstacle to be greater than \( p \). The confidence region of the robot can be used to transform the chance constraints into deterministic ones. In particular, we approximate the area \( R \) where the robot might exist with probability \( p \) by using the ellipsoid:

\[
R = \{ z = (x, y)^T \mid (z - \bar{z})^T \Sigma^{-1}(z - \bar{z}) \leq s(p) \},
\]

where \( \bar{z} \) is mean of \( z \), \( \Sigma \) is covariance matrix given by:

\[
\Sigma = \begin{bmatrix}
\text{VAR}[x] & \text{COV}[x,y] \\
\text{COV}[y,x] & \text{VAR}[y]
\end{bmatrix},
\]

and \( s(p) \) is the scaling factor. If the position of the robot followed a Gaussian distribution, then \( s(p) \) could be obtained by computing the chi-square inverse cumulative distribution function with a specified \( p \), which is fixed over time. In our case, however, the distribution of the position is not Gaussian even though the uncertain parameters follow such a distribution. Thus, we evaluate \( s(p) \) using Monte Carlo simulation by sampling the random parameters at each time step. In order to facilitate the computation, we overestimate the confidence region with a circle whose radius is the major axis of the ellipsoid, which can be obtained as the largest eigenvalue of \( \Sigma \), i.e. \( \lambda_{\text{max}} \). Thus, a constraint for a circle obstacle with center \((x_c, y_c)\) and radius \( r_c \) can be formulated as follows:

\[
g(x, y) = (r_c + \sqrt{s(p)\lambda_{\text{max}}^2})^2 - [(x - x_c)^2 + (y - y_c)^2] \leq 0
\]

\[\text{IV. RESULTS}\]

In this section, we present simulation results that verify the effectiveness of our approach. Initially, we apply the method on a differential wheeled robot while explaining in detail its performance compared to other related approaches. Subsequently, we demonstrate the applicability of our method to more complex systems such as a quadrotor.

\[\text{A. Differential Wheeled Robot}\]

We will first test our algorithm on a differential wheeled robot as shown in Fig. I. The goal of the robot is to reach a desired target while avoiding obstacles. Its model contains three uncertain parameters: the tread \( d \) and the wheel radii \( r_R \) and \( r_L \), all of which are assumed to be normally distributed with means 0.2 and variances \( 1.5 \times 10^{-3} \). Hermite polynomials are chosen - according to Table I - to describe

\[\text{Essentially, instead of solving}\ (10), \text{we solve at each time step } k, \text{the following problem:}\]

\[
\min_U \left[ J(X, U) + \sum_{i=1}^{w} \sum_{k=k}^{k+H-1} \mathcal{P}(\lambda^i_k, \mu^i_k, G_{i,k}(X_k, u_k)) \right]
\]

subject to \( X_{k+1} = f(X_k, u_k) \),

\[
u_{\text{min}} \leq u_k \leq u_{\text{max}}, \quad \kappa = k, \ldots, k + H - 1.
\]

Our algorithm starts with computing the locally optimal control sequence for problem (10) using gPC CDDP. Next, the first segment of the control is applied to the real system, keeping it in the confidence region of gPC CDDP while satisfying the chance constraints. Note that the specific values of the parameters are not required for the system to remain inside of the bounds. After obtaining the updated state of the real system, the gPC coefficients are updated accordingly. We can update the mean part (0-th coefficient) of the gPC dynamics by using the real system information and set the remaining coefficients - which correspond to the variance - to be zero. If we assume noisy measurements, the latter coefficients can be given non-zero values. Finally, in order to exploit a warm start for the next MPC computation, the control is also applied to the gPC dynamics, whose variance is low due to the information from the real system. This gPC MPC CDDP method can successfully keep the variance low, addressing the problem stated earlier in this section. This is further explained and demonstrated in Section IV.
Note that the gradient and the Hessian of $g$ - which are required for constrained DDP - can be analytically computed with the derivative of $\lambda_{\text{max}}$ [29]. In this simulation, we have set $dt = 0.02s$, $N = 60$, $H = 10$, $r = 2$, and $p = 0.95$. The initial control is set to be zero and the control bounds to be $|u_i| \leq 100$, $i = 1, 2$.

In Fig. 2 we demonstrate a comparison between gPC CDDP, gPC MPC CDDP and MPC CDDP which is purely deterministic. In Fig. 2(a) the accumulation of the variance while using gPC CDDP is demonstrated, making the robot unable to pass through the two obstacles. The confidence region is validated by overlaying the realizations of the trajectories with sampled parameters. In Fig. 2(b) gPC MPC CDDP is applied to the dynamics obtained by the realizations of different sampled parameters. The results show that the method can successfully reduce the variance growth, so the robot reaches the desired target. Note that for this method, the exact values of the dynamics parameters are not required, but only information of the states at every time step. Fig. 2(c) shows one trajectory from the same algorithm and the confidence region from some MPC cycles. Note that the trajectory indeed lies inside the confidence region computed by the algorithm. Finally, the performance of MPC CDDP with the parameters being sampled, is presented in Fig. 2(d)

In this simulation, the parameters used in the model are the mean parameters. Unlike gPC CDDP, the uncertainty of the dynamics which arises from the parameters is not considered here. Therefore, the real dynamics trajectory may violate the constraints even though they appear to be satisfied by the model trajectories computed by MPC. The poor performance of this method validates that not only using MPC but also taking into consideration the uncertainty is necessary.

B. Quadrotor

Subsequently, we test our method on a quadrotor whose dynamics are provided in [30]. We assume that we are aware of the upper and lower limits of the drag and lift coefficients. Thus, they can be viewed as uniformly distributed random variables which can be described using Legendre polynomials - according to Table I. The first few polynomials, their inner products, and the PDF are as follows:

$$\phi_0(z) = 1, \quad \phi_1(z) = z, \quad \phi_2(z) = \frac{3}{2}z^2 - 1, \ldots$$

$$\langle \phi_i, \phi_j \rangle = \frac{1}{2i+1} \delta_{ij}, \quad \rho(z) = \frac{1}{2}.$$

with $z \in \mathbb{R}$ and $i, j = 0, 1, \ldots$. Note that the inner product is scaled by 1/2 compared to the orthogonality relation of the polynomial [22]. Each uncertain parameter, e.g. the drag coefficient $k_d$, is expanded by its mean $\mu_{k_d}$ and the distance between its mean and lower or upper limit of the distribution denoted by $\Delta_{k_d}$, i.e.:

$$k_d(\xi_1) = \mu_{k_d}\phi_0(\xi_1) + \Delta_{k_d}\phi_1(\xi_1) = \mu_{k_d} + \Delta_{k_d}\xi_1,$$

where $\xi_1$ is a uniformly distributed scalar random variable in $[-1, 1]$. Similarly, the lift coefficient $k_l$ can be expanded with the standard uniformly distributed random variable $\xi_2$.

Thus, we have the random vector $\xi = [\xi_1, \xi_2]^T$. We have used $\mu_{k_b} = 1.140 \times 10^{-7}$, $\Delta_{k} = \mu_{k_b}/3$, $\mu_{k_b} = 2.980 \times 10^{-6}$, $\Delta_{k_l} = \mu_{k_l}/3$. In this simulation, the control of the system is the force generated by four rotors with control limits $0 \leq u \leq 3$. The initial state of the quadrotor is hovering. Moreover, $N = 100$, $H = 25$, $dt = 0.02s$, $r = 2$, and $p = 0.95$. Although this problem is 3D, the concepts of confidence region and chance constraints presented in [IV-A] are readily extended and applied to this task.

Fig. 3 demonstrates the quadrotor simulation results. In Fig. 3(a) we observe that most of the realizations of the trajectories are captured within the blue ball which shows the confidence region of the quadrotor. We also observe again
that with gPC CDDP, the growth of the variance pushes the quadrotor away from the desired state in order to satisfy the constraints. The trajectories obtained from gPC MPC CDDP are presented in Fig. 3(b). Similarly to the case of the differential wheeled robot, our receding horizon approach successfully reduces the variance, driving the trajectories closer to the target.

V. CONCLUSION

In this work, we propose a novel chance-constrained receding horizon control method that is able to handle uncertainty arising from model parameters using gPC. In particular, by transforming a chance-constrained optimization problem into a deterministic one, we are able to solve it using a constrained DDP technique. Our method successfully handles high-dimensional dynamics while enjoying the convergence properties and scalability of DDP. Furthermore, in order to address the accumulation of uncertainty on the trajectory, we implement our algorithm in a receding horizon fashion. Simulation results demonstrate that our method is applicable to complex robotic tasks and able to deal effectively with parametric uncertainty and non-linear constraints.

Future work would examine the robustness of our method under other forms of uncertainty such as process noise. Moreover, we would like to explore potential combinations of our method with others that can handle non-parametric uncertainty. Finally, we are interested in incorporating learning techniques such as Gaussian processes that will lead to safe learning algorithms under parametric and non-parametric uncertainty.

ACKNOWLEDGMENTS

The work of the second and third authors was supported by awards NSF CMMI-1936079 and NSF CPS-1932288, respectively. Augustinos Saravanoz also acknowledges support by the A. Onassis Foundation Scholarship.

REFERENCES

[1] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.
[2] E. Theodorou, J. Buchli, and S. Schaal, “A generalized path integral control approach to reinforcement learning,” The Journal of Machine Learning Research, vol. 11, pp. 31–1781, 2010.
[3] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, “Learning variable impedance control,” The International Journal of Robotics Research, vol. 30, no. 7, pp. 820–932, 2011.
[4] D. H. Jacobson and D. Q. Mayne, Differential dynamic programming. Elsevier, 1970.
[5] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems.” in ICINCO (1), 2004, pp. 222–229.
[6] G. Lantoine and R. P. Russell, “A hybrid differential dynamic programming algorithm for constrained optimal control problems: part I: Theory,” Journal of Optimization Theory and Applications, vol. 154, no. 2, pp. 382–417, 2012.
[7] E. Todorov and W. Li, “A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems,” in Proceedings of the 2005 American Control Conference, 2005. IEEE, 2005, pp. 300–306.
[8] E. Theodorou, Y. Tassa, and E. Todorov, “Stochastic differential dynamic programming,” in Proceedings of the 2010 American Control Conference, 2010, pp. 1125–1132.
[9] D. Xiu and G. Karniadakis, “The wiener–askley polynomial chaos for stochastic differential equations,” SIAM J. Sci. Comput., vol. 24, pp. 619–644, 2002.
[10] L. Fogiano and M. Khammash, “Nonlinear stochastic model predictive control via regularized polynomial chaos expansions,” in 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), 2012, pp. 142–147.
[11] K.-K. Kim and R. Braatz, “Generalised polynomial chaos expansion approaches to approximate stochastic model predictive control,” International Journal of Control, vol. 8, 08 2013.
[12] M. von Andrian and R. D. Braatz, “Offset-free input-output formulation of stochastic model predictive control based on polynomial chaos theory,” in 2019 American Control Conference (ACC), 2019, pp. 360–365.
[13] S. Lucia, P. Zometa, M. Kögel, and R. Findeisen, “Efficient stochastic model predictive control based on polynomial chaos expansions for embedded applications,” in 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 3006–3012.
[14] A. Mesbah, S. Streif, R. Findeisen, and R. D. Braatz, “Stochastic nonlinear model predictive control with probabilistic constraints,” in 2014 American Control Conference, 2014, pp. 2413–2419.
[15] Y. K. Nakka and S. Chung, “Trajectory optimization for chance-constrained nonlinear stochastic systems,” in 2019 IEEE 58th Conference on Decision and Control (CDC), 2019, pp. 3813–3818.
[16] V. Kumar, E. Todorov, and S. Levine, “Optimal control with learned local models: Application to dexterous manipulation,” in 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016, pp. 378–383.
[17] J. Mortimoto, G. Zeglin, and C. G. Atkeson, “Minimax dynamic differential programming: Application to a biped walking robot,” in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37543), vol. 2. IEEE, 2003, pp. 1927–1932.
[18] G. I. Boutselis and G. Theodorou, “Discrete-time differential dynamic programming on lie groups: Derivation, convergence analysis, and numerical results,” IEEE Transactions on Automatic Control, vol. 66, no. 10, pp. 4636–4651, 2021.
[19] E. G. Birgin, R. A. Castillo, and J. M. Martínez, “Numerical comparison of augmented lagrangian algorithms for nonconvex problems,” Computational Optimization and Applications, vol. 31, no. 1, pp. 31–55, 2005.
[20] J. Fisher and R. Bhattacharya, “Optimal trajectory generation with probabilistic system uncertainty using polynomial chaos,” Journal of Dynamic Systems, Measurement, and Control, vol. 133, p. 014501, 01 2011.
[21] Y. Pan, G. I. Boutselis, and E. A. Theodorou, “Efficient reinforcement learning via probabilistic trajectory optimization,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 11, pp. 5459–5474, 2018.
[22] G. I. Boutselis, Y. Pan, and E. A. Theodorou, “Numerical trajectory optimization for stochastic mechanical systems,” SIAM Journal on Scientific Computing, vol. 41, no. 4, pp. A2065–A2087, 2019.
[23] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, “Constrained differential dynamic programming revisited,” arXiv preprint arXiv:2005.00985, 2020.
[24] L. Fagiano and M. Khammash, “Nonlinear stochastic model predictive control based on polynomial chaos expansions for embedded applications,” in 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 3006–3012.
[25] E. G. Birgin, R. A. Castillo, and J. M. Martínez, “Numerical comparison of augmented lagrangian algorithms for nonconvex problems,” Computational Optimization and Applications, vol. 31, no. 1, pp. 31–55, 2005.
[26] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.
[27] Y. Tassa, N. Mansard, and E. Todorov, “Control-limited differential dynamic programming,” in 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 1168–1175.
[28] Y. Tassa, T. Erez, and W. Smart, “Receding horizon differential dynamic programming,” Advances in Neural Information Processing Systems 20 – Proceedings of the 2007 Conference, 01 2007.
[29] R. B. Nelson, “Simplified calculation of eigenvector derivatives,” AIAA Journal, vol. 14, no. 9, pp. 1201–1205, 1976.
[30] T. Luukkonen, “Modelling and control of quadcopter,” Independent research project in applied mathematics, Espoo, 2011.