

# Learning Random Feature Dynamics for Uncertainty Quantification

Diego Agudelo-España, Yassine Nemmour, Bernhard Schölkopf, Jia-Jie Zhu

**Abstract**—An inherent challenge of learning-based control tasks is posed by uncertainty due to finite training datasets. Even though there are principled tools to obtain confidence bounds for pointwise evaluation of learned dynamics models, it remains a challenging task to quantify the induced uncertainty in downstream quantities of interest due to the intrinsic recursive structure of dynamical systems. In this paper, we view the unknown one-step dynamics as a smooth function in a reproducing kernel Hilbert space and leverage random features for an approximate but highly structured parameterization of pointwise confidence bounds. As a result, we obtain downstream confidence bounds through an optimal control formulation under an uncertainty-aware random feature dynamics model. Our model is effectively a shallow neural network, which enables us to view the whole dynamical system as a deep neural network. Exploiting this perspective, we show that a Pontryagin’s minimum principle solution is equivalent to using the Frank-Wolfe algorithm on the induced neural network. Various numerical experiments on dynamics learning showcase the capacity of our methodology.

## I. INTRODUCTION

Data-driven dynamics models have become commonplace in modern control tasks, given the availability of machine learning tools and potentially large training datasets. These models are, however, inherently subject to uncertainty since they are derived from finite datasets. The assessment of this so-called epistemic uncertainty, although often neglected, is of utmost importance in certain control settings such as safe operation [1] or active exploration [2]. Intuitively, uncertainty quantification entails establishing confidence bounds for a variable of interest in the light of a specific training dataset.

Principled uncertainty quantification is in general a very difficult task without further regularity assumptions on the true data-generating mechanism. For instance, in the context of unknown dynamical systems, we can assume that the function class of the true dynamics function  $f$  is a Reproducing Kernel Hilbert Space (RKHS) similar to [1], [3]. This non-restrictive assumption on  $f$  enables us to use a Gaussian Process (GP) as an statistical tool for *robustly* bounding plausible evaluations of  $f$  given access to potentially noisy observations thereof [4]. We remark that these bounds are defined in an *pointwise* manner such that for a set of input variables  $z_0, \dots, z_{H-1}$ , we deal with confidence sets of the form  $f_h := f(z_h) \in \mathcal{F}_h, h = 0, \dots, H - 1$ , which typically hold with high probability. We emphasize that the bounded confidence sets  $\mathcal{F}_0, \dots, \mathcal{F}_{H-1}$  depend on a particular set of

observations  $\mathcal{D}$  and are determined by a GP under an RKHS hypothesis space.

We are particularly interested in computing confidence bounds for *downstream* quantities that depend in a complex manner on an unknown dynamics model  $f$ . Consider, for instance, any typical cost function  $J$  in an optimal control problem (OCP), under an unknown dynamics function  $f$ , that penalizes the full state trajectory in a certain way for a given fixed controller. Ultimately, we would like to determine a set  $\mathcal{J}$  of plausible values such that  $J \in \mathcal{J}$  with high probability under  $\mathcal{D}$ . This set can be characterized harnessing the confidence bounds for evaluations of  $f$  as follows

$$\mathcal{J} = \{J(f_0, \dots, f_{H-1}) \in \mathbb{R} \mid f_h \in \mathcal{F}_h, h = 0, \dots, H - 1\}, \quad (1)$$

where we make explicit the dependence of the cost function on the (unknown) dynamics through the evaluations  $f_0, \dots, f_{H-1}$ . Note that the total number of evaluations  $H$  corresponds in this case with the horizon of the considered OCP.

A natural approach to estimate (1) consists in *sampling* plausible realizations of the unknown dynamics  $f_h$  from their corresponding confidence sets  $\mathcal{F}_h$  for  $h = 0, \dots, H - 1$ , which requires exhaustive simulations of the data-driven dynamics and therefore might be impractical in certain settings. Another perspective to estimate (1) is based on uncertainty propagation, which implies the difficult task of characterizing how the uncertainty is transformed through the unknown and potentially nonlinear dynamics  $f$  during the evaluation of  $J$ . Given this difficulty, some methods rely on approximate propagation strategies such as those based on linearization of the dynamics model [5], [6]. A notable exception is [1], where the uncertainty propagation is overapproximated leveraging a Lipschitz assumption on  $f$ , which enables rigorous guarantees but might lead to overconservatism [7].

In this work, we depart from the previous approaches by exploiting the specific form of both the quantity of interest and the structure of the data-driven confidence sets. More formally, we adopt a *constrained optimization* perspective and define an upper confidence bound on  $J$  as

$$U_J := \max_{f_h \in \mathcal{F}_h, h=0, \dots, H-1} J(f_0, \dots, f_{H-1}), \quad (2)$$

and we likewise define the corresponding lower confidence bound  $L_J$  by taking  $\min$  in (2) instead. The resulting confidence interval can then be defined as  $\mathcal{J} = [L_J, U_J]$ . Note that the worst-case cost  $U_J$  can be used to stress test a system or certify certain controller design, whereas the best-case cost  $L_J$  can drive active exploration under the principle

The authors are with the Empirical Inference Department, Max Planck Institute for Intelligent Systems, Tübingen, Germany {first.last}@tuebingen.mpg.de

Jia-Jie Zhu is now with the Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany.

of optimism in the face of uncertainty [3].

From a practical perspective, computing confidence bounds using *exact* GP inference is computationally demanding due to its cubic scaling in the number of training observations and the difficulty to efficiently incorporate new observations. There are, however, a handful of well-established alternatives for *approximate* GP regression that exhibit a better scaling in terms of computational tractability for which we refer to [8] for an overview. We focus on *Random Features* (RF), an approximation tool originated from large-scale kernel machines [9], not only for its practicality but also for yielding a structured parameterization of pointwise confidence bounds (See Section II-C), which we set out to exploit for the estimation of *downstream* confidence bounds. Due to its simple *linear-in-parameter* structure, RF have been extensively used to speed up kernel methods such as GPs [10], see, e.g., [11] for a recent survey.

Remarkably, the convenient uncertainty representation induced by RF enables an efficient gradient-based solution of (2) under *structured* constraints. We summarize our contributions as: 1) We revisit existing GP-based pointwise confidence bounds under the lenses of RF and show that the resulting bounds are highly structured. Based on this, we propose the *set-valued* uncertainty-aware random feature (URF) dynamics, which we can efficiently infer from available observations. 2) Exploiting the structure of URF, we propose an algorithm to solve (2) based on Pontryagin’s minimum principle (PMP). 3) We point out an equivalence of our PMP-based method to the conditional gradient (a.k.a., Frank-Wolfe) by adopting a neural network view of URF. 4) Finally, we show that the eigenvalue-decay structure of common RKHSs allows us to learn a convenient lower dimensional representation of the URF dynamics model.

## II. PRELIMINARIES

In this section, we will formally introduce our problem setting and briefly review GP methods, the RF framework and GP-based confidence bounds.

### A. Problem Formulation

We assume a partially-known discrete-time dynamics according to

$$x_{h+1} = h(x_h, u_h) + f(x_h, u_h), \quad (3)$$

where  $x_h \in \mathbb{R}^p$ ,  $u_h \in \mathbb{R}^q$  denote the system state and input action respectively,  $h: \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^p$  represents a (known) nominal model and  $f: \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^p$  accounts for unknown deviations from the nominal component. We also assume that both  $h$  and  $f$  are differentiable functions. We are interested in learning  $f$  from collected transitions

$$\mathcal{D} := \{(z_i, x_{i+1})\}_{i=1}^t, \quad (4)$$

where  $z_i := (x_i, u_i)$  denotes the concatenated inputs of (3) and we assume full observability of the system state and

actions. Furthermore, we consider a particular task specified as a finite-horizon cost function

$$J(f) = \sum_{h=0}^H c_h(x_h, u_h), \quad (5)$$

where  $c_h(\cdot, \cdot)$  denotes a differentiable stage-wise cost and we emphasize the dependence on the unknown dynamics  $f$  through the dynamics constraint (see (3)) assuming a *constant* sequence of control inputs  $U = [u_0, u_1, \dots, u_H]$  and a fixed initial condition  $x_0$ . In particular, we want to determine the worst- and best-case *plausible* performance we could possibly have given a limited training dataset  $\mathcal{D}$ , as measured by (5). Establishing such confidence bounds is, in general, an extremely difficult task without further regularity assumptions on the unknown  $f$ . Following previous work [4], we assume that  $f$  belongs to an RKHS  $\mathcal{H}_k$ , which can be alternatively stated as follows.

**Assumption 1.** *The unknown function  $f$  in (3) has bounded norm in an RKHS associated with a known kernel  $k$ ; i.e.,  $\|f\|_{\mathcal{H}_k} \leq B_f < \infty$ .*

Intuitively,  $\mathcal{H}_k$  contains well behaved functions whose *smoothness* w.r.t. the kernel  $k$  is measured through  $\|g\|_{\mathcal{H}_k} = \sqrt{\langle g, g \rangle_{\mathcal{H}_k}}$  for any  $g \in \mathcal{H}_k$ . Note that every positive definite kernel function  $k$  is associated with a Hilbert space  $\mathcal{H}_k$  and a feature map  $\varphi: \mathcal{Z} \rightarrow \mathcal{H}_k$ , for which  $k(z, z') = \langle \varphi(z), \varphi(z') \rangle_{\mathcal{H}_k}$  defines an inner product on  $\mathcal{H}_k$ . The space  $\mathcal{H}_k$  contains real-valued functions on  $\mathcal{Z}$  and is called an RKHS, equipped with the *reproducing property*:  $g(z) = \langle g, \varphi(z) \rangle_{\mathcal{H}_k}$  for any  $g \in \mathcal{H}_k, z \in \mathcal{Z}$ .

Assumption 1 is not only motivated by the expressiveness of RKHSs to represent the true data-generating functions in nature such as  $f$  based on empirical data, but also for the existence of statistical tools to obtain principled uncertainty estimates. In fact, it enables us to use GPs as statistical models to derive confidence bounds for the evaluation of  $f$ , which we briefly introduce next.

### B. Gaussian Process Regression

A GP can be intuitively understood as a Gaussian distribution defined over a space of functions. We write  $g(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$  to denote that the scalar-valued<sup>1</sup> function  $g$  is a sample from a GP with mean function  $\mu(\cdot)$  and covariance function  $k(\cdot, \cdot)$ . Formally, this means that any finite collection of function evaluations  $[g(z_1), g(z_2), \dots, g(z_m)]$  is *a priori* jointly Gaussian distributed with mean  $\mathbb{E}[g(z_i)] = \mu(z_i)$  and covariance  $\mathbb{E}[(g(z_i) - \mu(z_i))(g(z_j) - \mu(z_j))] = k(z_i, z_j), 1 \leq i, j \leq m$ . GP regression therefore amounts to computing the resulting posterior distribution after conditioning on potentially noisy observations  $y_{1:t} = [y_1, \dots, y_t]^\top$  at input locations  $A_t = \{z_1, \dots, z_t\}$ , where the noise realizations  $\epsilon_i = y_i - g(z_i)$  are assumed i.i.d. samples from  $\mathcal{N}(0, \lambda^2)$  if present. The resulting posterior distribution over  $g$  can

<sup>1</sup>we focus on the scalar-valued case for ease of exposition. In the vector-valued setting we assume *independent* GP priors across state dimensions.

be shown to be again a GP [8],  $g(\cdot) \sim \mathcal{GP}(\mu_t(\cdot), k_t(\cdot, \cdot))$ . By using standard properties of Gaussian distributions and assuming  $\mu(\cdot) = 0$  from now on w.l.o.g. we obtain

$$\begin{aligned}\mu_t(z) &= k_t(z)^\top (K_t + \lambda^2 I)^{-1} y_{1:t}, \\ k_t(z, z') &= k(z, z') - k_t(z)^\top (K_t + \lambda^2 I)^{-1} k_t(z'), \\ \sigma_t^2(z) &= k_t(z, z),\end{aligned}\quad (6)$$

where  $K_t := [k(z, z')]_{z, z' \in A_t}$  is the so-called Gram matrix,  $k_t(z) := [k(z_1, z), \dots, k(z_t, z)]^\top$  and  $\sigma_t^2(z)$  denotes the marginal variance at  $z$ .

There is an important and well-known practical consideration relevant for the implementation of GP-based methods, namely the computational burden of *exact* GP regression. As dictated by (6), computing  $\mu_t(\cdot)$  and  $\sigma_t(\cdot)$  is computationally taxing given the  $\mathcal{O}(t^3)$  time complexity of matrix inversion and the  $\mathcal{O}(t^2)$  memory requirement to store the Gram matrix. We now introduce an alternative approach to approximate GP regression that offers a favourable trade-off in terms of computational complexity.

### C. Random Features for Gaussian Process Regression

Instead of adopting a function-space view of GP regression (Section II-B), we can alternatively leverage a weight-space perspective, in which GP samples are seen as a linear combinations of basis functions. The RF framework seeks to approximate the kernel  $k(\cdot, \cdot)$  (and the corresponding RKHS functions) through a randomized finite feature map  $\phi : \mathcal{Z} \rightarrow \mathbb{R}^l$  such that for a sufficiently large number of basis functions  $l$ ,

$$k(z, z') \approx \phi(z)^\top \phi(z'), \quad \forall z \in \mathcal{Z}, \quad (7)$$

where  $\phi(z) := [\phi_1(z), \dots, \phi_l(z)]^\top \in \mathbb{R}^l$  are the random features, e.g., Random Fourier Features<sup>2</sup>(RFF) [9]. To illustrate how the kernel approximation in (7) can be exploited for GP regression consider the Bayesian linear-in-parameter model

$$g(\cdot) = w^\top \phi(\cdot), \quad w \sim \mathcal{N}(\hat{w}, \Sigma), \quad (8)$$

which can be shown to be a  $l$ -dimensional approximation of the prior  $\mathcal{GP}(0, k(\cdot, \cdot))$  for  $\hat{w} = 0$  and  $\Sigma = I_d$  by (7). We emphasize that RF effectively yields a *parametric* approximation of  $g$ , which implies that the uncertainty over  $g$  is expressed (approximately) as the uncertainty over the weights  $w$  in (8). Crucially, this means that the corresponding posterior GP can be computed by means of Bayesian linear regression (BLR) as first proposed by [10] under the name of Sparse Spectrum GP regression. In fact, under i.i.d. measurement errors drawn from  $\mathcal{N}(0, \lambda^2)$ , as in Section II-B, conditioning on the observed inputs  $A_t$  and measurements  $y_{1:t}$ , the posterior over weights is also Gaussian with moments

$$\begin{aligned}\hat{w}_t &= (\Phi(A_t)^\top \Phi(A_t) + \lambda^2 I_l)^{-1} \Phi(A_t)^\top y_{1:t}, \\ \Sigma_t &= (\Phi(A_t)^\top \Phi(A_t) + \lambda^2 I_l)^{-1} \lambda^2,\end{aligned}\quad (9)$$

<sup>2</sup>RFF approximate stationary (i.e., shift-invariant) kernels, where  $\phi_i(z) = \sqrt{2/L} \cos(a_i^\top z + b_i)$  and the vector  $a_i$  is sampled proportional to the kernel's spectral density and the offset as  $b_i \sim \mathcal{U}(0, 2\pi)$ . We can therefore approximate the popular Gaussian kernel  $k(z, z') = e^{-\|z - z'\|_2^2 / 2l^2}$  by sampling  $a_i \sim \mathcal{N}(0, l^{-2} I_d)$ .

where  $\Phi(A_t) := [\phi(z_1), \phi(z_2), \dots, \phi(z_t)]^\top \in \mathbb{R}^{t \times l}$  denotes the RF evaluated at the training inputs  $A_t$ . Analogous to (6), we can compute the Gaussian posterior over a *function value* at a test input  $z \in \mathcal{Z}$ , whose moments can be easily derived thanks to the linear structure of (8) and are given by

$$\begin{aligned}\tilde{\mu}_t(z) &= \hat{w}_t^\top \phi(z), \\ \tilde{\sigma}_t^2(z) &= \phi(z)^\top \Sigma_t \phi(z) = \|\phi(z)\|_{\Sigma_t}^2.\end{aligned}\quad (10)$$

The RF-based representation of GPs has proven to be a practical alternative to the function-space approach, particularly in the context of large-scale regression problems. Note that computing the posterior distribution in (9) scales proportional to  $\mathcal{O}(l^3 + tl^2)$ , which compares favourably with respect to the  $\mathcal{O}(t^3)$  scaling of exact GP regression when  $l \ll t$ . Moreover, it is also possible to perform efficient online incremental posterior updates as new data becomes available.

These are all useful properties in the context of GP-based dynamics models, where we might want to update the dynamics model with freshly-acquired data, as in a receding-horizon MPC scheme. We focus in this work on the RF approximation of GP regression and exploit it for the estimation of downstream confidence bounds.

### D. Gaussian Process-based Pointwise Confidence Bounds

Given our problem of interest and assumptions, we now state a key result that enables us to use a GP to obtain rigorous confidence bounds for pointwise evaluation of an unknown function, such as  $f$  in (3).

**Proposition 1** ([1], Lemma 5). *Let Assumption 1 hold and assume access to  $t$  measurements of  $f$  subject to  $\lambda$ -sub-Gaussian noise, then with probability at least  $1 - \delta$ ,  $\delta \in (0, 1)$ , there exists  $\beta_t > 0$  such that it holds for all  $z \in \mathcal{X} \times \mathcal{U}, j = 1, \dots, p$  and  $t \geq 0$  that<sup>3</sup>*

$$|f_j(z) - \mu_{t,j}(z)| \leq \beta_t \sigma_{t,j}(z), \quad (11)$$

where we use  $f_j(\cdot)$  to denote the  $j$ -th output of the vector-valued function  $f(\cdot)$  and  $\mu_{t,j}(\cdot), \sigma_{t,j}(\cdot)$  to denote the predictive mean and variance functions of its corresponding GP model. We emphasize that this holds jointly for all  $z \in \mathcal{X} \times \mathcal{U}$ . This means that we have access to principled model uncertainty quantification uniformly across the input domain through the GP-posterior statistics  $\mu_t(\cdot), \sigma_t(\cdot)$  and scaling constant  $\beta_t$ .

## III. RANDOM FEATURES CONFIDENCE BOUNDS FOR UNCERTAINTY-AWARE DYNAMICS MODELS

In this Section, we set out to first revisit Proposition (1) under the finite-dimensional GP approximation introduced in Section II-C. Based on this, we then proceed to define our RF-based dynamics model, which involves multiple evaluations of the unknown dynamics term  $f$ .

<sup>3</sup> $\beta_t = B_f + 4\lambda \sqrt{\gamma_{t+1} + 1 + \ln(1/\delta)}$ , where  $\gamma_{t+1}$  is the information capacity associated with the kernel  $k$  and depends on the number of observations  $t$ , is a theoretically motivated choice [1]. However, our methodology is agnostic to its specific choice.

### A. Pointwise Confidence Bounds with Random Features

Instead of using  $\mu_t(\cdot)$  and  $\sigma_t(\cdot)$  in (6), we can use the finite-dimensional approximations  $\tilde{\mu}_t(\cdot)$  and  $\tilde{\sigma}_t(\cdot)$  in (10) yielding confidence intervals of the form

$$\begin{aligned} |f_j(z) - \tilde{\mu}_{t,j}(z)| &\leq \beta_t \tilde{\sigma}_{t,j}(z), \\ |f_j(z) - \hat{w}_{t,j}^\top \phi_j(z)| &\leq \beta_t \|\phi_j(z)\|_{\Sigma_{t,j}}, \end{aligned} \quad (12)$$

for appropriately defined  $\beta_t$ . Equivalently, we can write (12) as

$$\min_{w \in \mathcal{W}_{t,j}} w^\top \phi_j(z) \leq f_j(z) \leq \max_{w \in \mathcal{W}_{t,j}} w^\top \phi_j(z), \quad (13)$$

where we define for  $j = 1, \dots, p$  the ellipsoid

$$\mathcal{W}_{t,j} := \{w \in \mathbb{R}^l \mid \|w - \hat{w}_{t,j}\|_{\Sigma_{t,j}^{-1}} \leq \beta_t\}, \quad (14)$$

with radius  $\beta_t$ , center  $\hat{w}_{t,j}$  and orientation defined by  $\Sigma_{t,j}$ , cf., (9). To verify (13) note that  $w \in \mathcal{W}_{t,j}$  implies that  $\forall a \in \mathbb{R}^l$ ,  $|a^\top (w - \hat{w}_{t,j})| \leq \beta_t \|a\|_{\Sigma_{t,j}}$ . Remarkably, (13) enables us to take a set-valued perspective, where under the RF-based approximation of GPs and assumptions in Proposition 1 (11) reads as

$$f_j(z) \in \{w^\top \phi_j(z) \mid w \in \mathcal{W}_{t,j}\}, \quad (15)$$

and holds for  $j = 1, \dots, p$ ,  $\forall z \in \mathcal{Z}$  and  $t \geq 0$ . This is a highly structured parametrization of confidence bounds for pointwise evaluation of  $f_j$ , namely (15) exhibits linear-in-parameter structure and ellipsoidal parametric uncertainty.

### B. Uncertainty-aware Random Feature Dynamics Model

We leverage the RF framework and the set-valued view in (15) to approximately bound the (unknown) vector-valued dynamics in (3) with high probability as

$$f(z) \in \{W\phi(z) \mid W \in \mathcal{W}_t\}, \quad (16)$$

where  $W = [w_1, \dots, w_p]^\top \in \mathbb{R}^{p \times l}$  denotes the uncertain parameters of the multiple-output mapping, and we assume for ease of exposition that the RF vector  $\phi$  is shared across state dimensions and therefore drop the state index  $j$ . We write  $W \in \mathcal{W}_t$  in (16) to denote that  $w_j \in \mathcal{W}_{t,j}$  holds across state dimensions  $j = 1, \dots, p$ ; i.e., we have an ellipsoidal constraint for each row of  $W$ , cf., (14). Using the terminology of *robust optimization*, we refer to  $\mathcal{W}_t$  as an uncertainty set and consider the following equivalent but more concise parameterization

$$\mathcal{W}_t := \{W : \text{diag}((W - \hat{W}_t)\Sigma_t^{-1}(W - \hat{W}_t)^\top) \leq \beta_t^2\}, \quad (17)$$

where  $\hat{W}_t = [\hat{w}_{t,1}, \dots, \hat{w}_{t,p}]^\top \in \mathbb{R}^{p \times l}$ , the positive-definitive matrix  $\Sigma_t \in \mathbb{R}^{l \times l}$  fully characterize the uncertainty set and the inequality is interpreted componentwise<sup>4</sup>. We highlight that  $\mathcal{W}_t$  is a *data-driven* uncertainty set given its dependence on the posterior distribution over the weights determined by (9), which in turn depends on the observed data. We refer to the special case  $\mathcal{W}_t = \{\hat{W}_t\}$  as *certainty-equivalent random feature* (CERF) dynamics model, where

<sup>4</sup> $\text{diag}(A)$  is the column vector containing the diagonal elements of matrix  $A$ .

the uncertainty encoded by  $\Sigma_t$  is neglected yielding a point estimate for the unknown dynamics.

For ease of exposition we introduce the notation

$$\hat{f}(x, u, W) := W\phi(x, u), \quad (18)$$

where we unpack  $z = (x, u)$ , and propose our main set-valued dynamics model, the *uncertainty-aware random feature* (URF) dynamics model

$$\mathcal{X}_{h+1} := \left\{ x_{h+1} = h(x_h, u_h) + \hat{f}(x_h, u_h, W_h) \mid \begin{array}{l} W_h \in \mathcal{W}_t, \\ x_h \in \mathcal{X}_h \end{array} \right\}, \quad (19)$$

for  $h = 0, 1, \dots, H-1$ , and  $\mathcal{X}_0 = \{x_0\}$ . As in Section II, we assume that both the initial condition  $x_0$  and controls  $u_0, \dots, u_{H-1}$  are given and fixed. In fact, the sets  $\mathcal{X}_0, \mathcal{X}_1, \dots, \mathcal{X}_H$  are reachable sets induced by the *parametric uncertainty*  $\mathcal{W}_t$ , and we would expect them to contain the true state trajectory under (3) with high probability in the light of Proposition (1). We note however that we rely on an approximate GP representation in this work, which implies that the obtained reachable sets should be likewise interpreted in an approximate sense.

**Remark.** We note that (19) considers potentially different realizations of the parameters  $W_0, W_1, \dots, W_{H-1}$ . This is because the URF dynamics model fundamentally stems from a robust view where confidence bounds are defined pointwise and independently across the input domain, as in (11).

## IV. WORST- AND BEST-CASE DYNAMICS VIA PONTYAGIN'S MINIMUM PRINCIPLE (PMP)

In this section we focus on finding the worst-case  $U_J$  (or analogously the best-case  $L_J$ ) with respect to a given cost function  $J$  as an optimization problem — see (2). Formally, given some overall cost objective  $J$  and a horizon  $H$  as in (5) subject to the URF dynamics constraint in (19), we wish to find the worst pointwise realizations of the unknown dynamics across the horizon by solving

$$(W_0^*, \dots, W_{H-1}^*) = \arg \max_{W_h \in \mathcal{W}_t, h=0, \dots, H-1} J(W_0, \dots, W_{H-1}), \quad (20)$$

where we emphasize the dependence on the uncertain RF parameters  $W_h, h = 0, \dots, H-1$ , and assume everything else (e.g., controls) fixed.

Due to the nonlinear dynamics, solving for the worst-case dynamics under URF model (19) is a non-convex optimization problem, which prohibits finding globally optimal solutions of (20). However, the shallow structure of URF, and the reproducing property in general, views the dynamics as linear in a lifted space, which together with an optimal control perspective allows us to characterize a tailored *necessary* optimality condition for the worst-case dynamics via PMP.

Let us denote, with a slight abuse of notation, the total *negative* cost as  $\hat{J}(W_0, \dots, W_{H-1}) = \sum_{h=1}^H \hat{c}_h(x_h, u_h)$ , where  $\hat{c}_h(x_h, u_h) = -c_h(x_h, u_h)$  denotes the negative of the stage cost  $c_h$  in (5). Then the maximization in (20) can equivalently

be formulated as the minimization of  $\hat{J}$ . Following the PMP formalism, we define the control Hamiltonian as

$$H_h(z, p, W) := \hat{c}_h(z) + p^\top \hat{f}(z, W),$$

where  $p \in \mathbb{R}^p$  denotes the so-called co-state variable and  $z := (x, u)$  as before. We omit for simplicity the known part  $h(z)$  of the model since it does not depend on  $W$ .

**Proposition 2** (PMP for worst-case dynamics). *Suppose  $(W_0^*, \dots, W_{H-1}^*)$  are the worst-case parameter realizations in the uncertainty set  $\mathcal{W}_t$  and  $(x_0^*, x_1^*, \dots, x_H^*)$  is the corresponding state trajectory under a fixed sequence of controls  $(u_0, u_1, \dots, u_H)$ . Then there exists a sequence of (co-state) variables  $(p_0^*, p_1^*, \dots, p_H^*)$  that satisfy the adjoint equations*

$$p_H^* = \nabla_x \hat{c}_H(z_H^*), \quad p_h^* = \nabla_x H_h(z_h^*, p_{h+1}^*, W_h^*),$$

for  $h = 0, 1, \dots, H-1$  and  $z_h^* := (x_h^*, u_h)$ , under which the worst-case dynamics parameters  $(W_0^*, \dots, W_{H-1}^*)$  minimize the corresponding Hamiltonian<sup>5</sup>

$$W_h^* = \arg \min_{W \in \mathcal{W}_t} H_h(z_h^*, p_{h+1}^*, W), \quad \text{for } h = 0, \dots, H-1. \quad (21)$$

The PMP motivates us to find the worst-case dynamics via an *indirect method* of optimal control in which we find a candidate solution of (20) by solving for the necessary condition stated in Proposition 2. Specifically, we rely on the method of successive approximations (MSA) [13] to do so, which alternates between forward-backward passes and Hamiltonian minimization, as illustrated in Algorithm 1. We

**Input:** Controls  $u_0, \dots, u_H$ , set  $\mathcal{W}_t$ , initial  $x_0$   
**Output:** Worst-case parameters  $W_0, W_1, \dots, W_{H-1}$   
1 Initialize  $W_0, W_1, \dots, W_{H-1}$  (e.g.,  $W_i = \hat{W}_t$ , cf. (17))  
2 **for**  $e = 0, 1, \dots, E$  **do**  
3     // Forward pass/Shooting dynamics  
4     Initialize  $z_0 = (x_0, u_0)$   
5     **for**  $h = 0$  **to**  $H-1$  **do**  
6          $x_{h+1} = \hat{f}(z_h, W_h)$   
7     // Backward pass/Adjoint equation  
8     Initialize  $p_H = \nabla_x \hat{c}_H(z_H)$   
9     **for**  $h = H-1$  **to**  $0$  **do**  
10          $p_h = \nabla_x H_h(z_h, p_{h+1}, W_h)$   
11     // Update worst-case dynamics  
12     **for**  $h = 0$  **to**  $H-1$  **do**  
13         Set  $W_h = \arg \min_{W \in \mathcal{W}_t} H_h(z_h, p_{h+1}, W)$

**Algorithm 1:** Worst-case dynamics MSA

note that PMP-based optimality conditions are stronger than typical first-order necessary conditions, in particular because of the *global* Hamiltonian minimization in (21), which under our URF dynamics reads as

$$\min_{W \in \mathcal{W}_t} \left\{ H_h(z_h, p_{h+1}, W) = \hat{c}_h(z_h) + p_{h+1}^\top W \phi(z_h) \right\}, \quad (22)$$

<sup>5</sup>Once we interpret the worst-case dynamics parameters  $W_h \in \mathcal{W}_t$ ,  $h = 0, \dots, H-1$  as control inputs to the RF dynamics  $\hat{f}(z_h, W_h)$ , the proposition follows from the standard discrete-time PMP; see, e.g., [12, Volume I, 4th Edition].

where  $\mathcal{W}_t$  is an uncertainty set defined by some  $\hat{W}_t$  and  $\Sigma_t$ , as in (17). Thanks to the linear objective and the ellipsoidal constraints on the rows of  $W$ , (22) can be *globally* minimized in closed-form as follows.

**Proposition 3.** *Under URF dynamics, the minimizer of the Hamiltonian in (22) for  $h = 0, \dots, H-1$ , is given by*<sup>6</sup>

$$W_h^* = \hat{W} - \frac{\lambda_{h+1} \phi(z_h)^\top \Sigma_t}{\|\phi(z_h)\|_{\Sigma_t}}, \quad \lambda_{h+1} := \text{sgn}(p_{h+1}) \in \mathbb{R}^p. \quad (23)$$

**Remark.** *The URF dynamics is effectively a shallow neural network; i.e., its computation can be interpreted in terms of an input layer, a hidden layer (defined by  $\phi$ ) and an output layer (defined by  $W$ ). Consequently, the whole dynamical system can be seen as a deep neural network (DNN) and the optimization in (20) as (constrained) DNN training.*

*A. Incremental update of Hamiltonian dynamics: equivalence to deep learning optimization*

We note however that the convergence of MSA can only be established for particular choices of dynamics and cost functions, and it might diverge in the case of nonlinear dynamics such as URF. [14] recently analyzed this issue in the context of PMP-based training of DNN and advocated for an incremental update of the Hamiltonian to avoid divergence. This means that the parameters  $W_h$  in Step 10 of Algorithm 1 can be updated in the direction of the minimizer  $W_h^*$  of (23) rather than set equal to it. Formally, we update each  $W_h$  to  $W_h^+$  according to

$$W_h^* = \arg \min_{W \in \mathcal{W}_t} H_h(z_h, p_{h+1}, W), \quad (24)$$

$$W_h^+ = W_h + \gamma_e (W_h^* - W_h),$$

for some step size schedule  $\gamma_e$  and  $h = 0, \dots, H-1$ . We will refer to Algorithm 1 with the incremental update (24) as the *inexact PMP*. We now show that, the incremental minimization of Hamiltonian under URF dynamics is equivalent to performing the conditional gradient method, also known as the Frank-Wolfe algorithm [15], on the weights of a DNN.

**Proposition 4** (Equivalence of inexact PMP and Frank-Wolfe for deep learning). *Inexact PMP, i.e., Algorithm 1 with update step replaced by (24), is equivalent to performing a Frank-Wolfe update on the total negative cost  $\hat{J}$ :*

$$\bar{W}_h = \arg \min_{W \in \mathcal{W}_t} \langle \nabla_{W_h} \hat{J}(W_0, \dots, W_{H-1}), W \rangle_F, \quad (25)$$

$$W_h^+ = W_h + \gamma_e (\bar{W}_h - W_h).$$

*Furthermore, this Frank-Wolfe algorithm with  $\gamma_e = 1$  recovers the exact PMP update*<sup>7</sup>.

<sup>6</sup>This follows from the simpler fact that  $w^* = \arg \min_{w \in \mathcal{W}} a^\top w$  (where  $\mathcal{W}$  is an ellipsoid defined by  $\hat{w}$  and  $\Sigma$ ) has the closed-form solution  $w^* = \hat{w} - \frac{\Sigma a}{\|a\|_\Sigma}$ . We use  $\text{sgn}(\cdot)$  to denote the sign function.

<sup>7</sup>It holds that  $\nabla_{W_h} \hat{J}(W_0, \dots, W_{H-1}) = \nabla_{W_h} H_h(z_h, p_{h+1}, W_h) = p_{h+1} \phi(z_h)^\top$  [14, Proposition 5], which in turn implies that  $W_h^*$  in (24) and  $\bar{W}_h$  in (25) are equivalent. If  $\gamma_e = 1$ , then the exact minimization of the Hamiltonian is obtained given the linear-in-parameter URF dynamics. Hence, we recover exact PMP.

**Remark.** Note that, if the dynamics  $\hat{f}$  is not linear-in-parameter, the last statement of Proposition 4 does not hold, which can also be seen as an implication of the reproducing property in Hamiltonian dynamics. That is the motivation of using PMP in the context of URF dynamics.

To the best of our knowledge, Proposition 4 also constitutes a contribution to the current deep learning literature, aside from this paper’s context of URF dynamics. It characterizes the equivalence of constrained minimization of the Hamiltonian via PMP and optimizing deep models via conditional gradient method, which generalizes the results in [14], [16] in the context of DNNs. In summary, it tells us that, to compute a candidate worst-case dynamics, it suffices to apply the conditional gradient method on the DNN induced by the URF dynamics across the time horizon  $H$ .

We note that we can take advantage of algorithmic differentiation (AD) tools, such as PyTorch, to avoid the explicit implementation of the backward pass in Algorithm 1. Moreover, AD can compute the backward pass with a cost proportional to the cost of forward simulation. Given that both the forward and backward passes can be efficiently implemented, we should also consider the cost of solving the Hamiltonian optimization in Step (10) of Alg. 1. The computational complexity in this step is dominated by matrix-vector products w.r.t.  $\Sigma_t$  according to (23), which are very efficient to compute in modern computer architectures and often implemented as an optimized primitive operation.

## V. LOWER-DIMENSIONAL REPRESENTATION USING RANDOM FEATURE NONLINEAR COMPONENT ANALYSIS

The more features we sample to form  $\phi$ , as detailed in Section II-C, the better we can approximate exact GP regression. From the computational viewpoint, however, we want to keep the number of features reasonably low such that it is still cheaper to compute the weights posterior in (9) than the exact GP posterior in (6). In this section, we propose to use a tailored dimensionality reduction procedure that enables us to leverage the expressivity of a large number of features while retaining computational tractability.

It is well-known that, for some common kernels, the Gram matrix  $K_t$ , which in the context of RF approximations we write as  $K_t = \Phi(A_t)\Phi(A_t)^\top$  following the notation in (9), has special eigenspectrum structure [17], [8]. For example, the eigenvalues of Gaussian kernel Gram matrices decay at an exponential rate. As the RF approximate the feature maps associated with the RKHS of the Gaussian kernel, we can expect  $K_t$  to have rapidly decaying eigenvalues. Intuitively, this gives us the power to reduce the dimensionality of the feature representation, such as by using kernel PCA. We now show how to exploit this structure to learn a *lower-dimensional representation* of the URF dynamics.

Specifically, we perform PCA on the Gram matrix  $K_t$  to obtain the lower-dimensional representation, which we denote as  $\psi(z) := P\phi(z)$ , where  $P \in \mathbb{R}^{\hat{l} \times l}$  is a PCA projection matrix obtained, e.g., by performing singular value decomposition of  $K_t$ . Note that  $\hat{l} \ll l$  denotes the

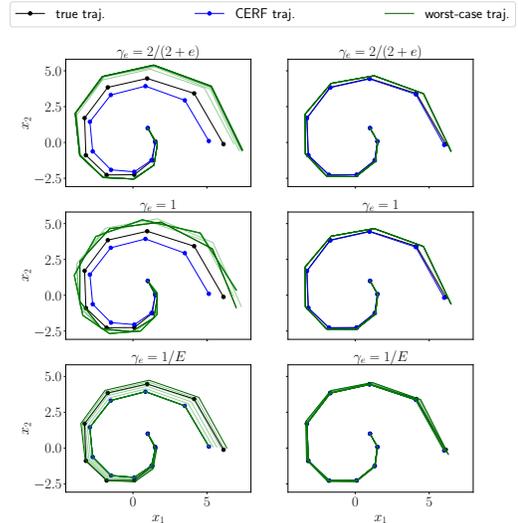


Fig. 1. Worst-case dynamics MSA over Source Spiral (Section VI). The worst-case trajectory is shown in dark green for scarce (left column) and large (right column) data regimes under different learning rates (rows). We denote intermediate trajectories with lighter green.

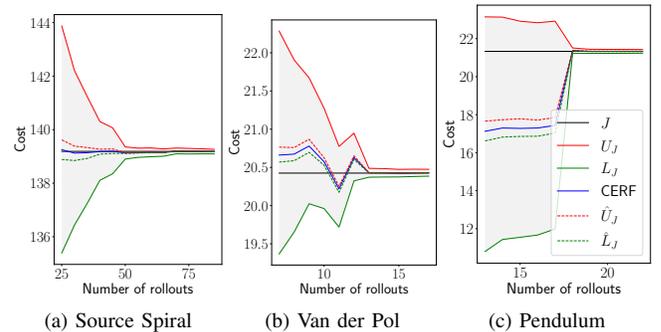


Fig. 2. Computed worst- ( $U_J$ ) and best-case ( $L_J$ ) trajectory cost under a fixed initial condition as a function of the number of training rollouts. In the small-data regime the mean cost (CERF) might be biased, however the true cost ( $J$ ) is always within the computed bounds, i.e.,  $J \in [L_J, U_J]$ . The sampling-based bounds  $[\hat{L}_J, \hat{U}_J]$  severely underestimate  $[L_J, U_J]$ . Note that both the uncertainty set  $\mathcal{W}_t$  and the induced bounds shrink with observed rollouts.

new dimension of the feature-based representation of the dynamics. As a result, our URF dynamics in (18) can be approximated using the *lower dimensional representation* as

$$\hat{f}_{\text{PCA}}(z, W) := W\psi(z) = WP\phi(z),$$

where now  $W \in \mathbb{R}^{p \times \hat{l}}$ . We highlight that the BLR algorithm described in section II-C can seamlessly incorporate the lower dimensional representation by using  $\Psi(A_t) := [\psi(z_0), \psi(z_1), \dots, \psi(z_t)]^\top$  instead of  $\Phi(A_t)$  in (9). A rigorous statistical analysis of the approximation error is beyond our current scope, for which we refer to [18], [11]

## VI. NUMERICAL EXPERIMENT

We use three nonlinear autonomous systems as testbeds since we do not focus on the control problem. We assume limited prior information about the true systems and our

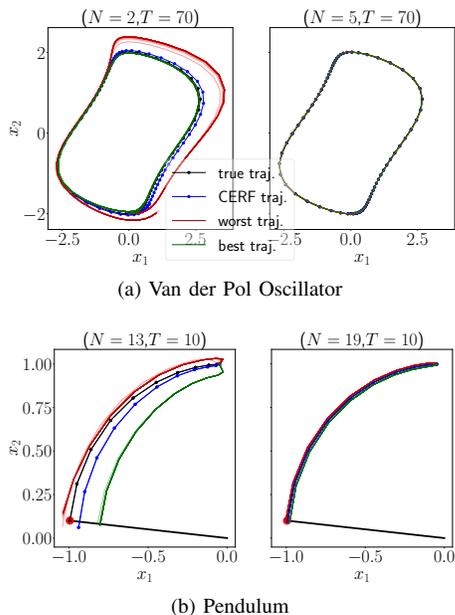


Fig. 3. Worst- (red) and Best-case (green) MSA trajectories (Sec. VI) under high (left column) and low uncertainty (right column). In uncertain settings, the URF model characterizes a set of plausible trajectories in contrast with a potentially biased point estimate given by CERF (blue). Intermediate MSA trajectories are denoted with lighter tones.

model consists therefore of an identity nominal component  $h(x_t) = x_t$ . We learn the residuals  $x_{t+1} - h(x_t)$  with our URF model using as training data a varying number trajectories ( $N$ ) of fixed length ( $T$ ) with randomly-sampled initial conditions. We use  $l = 1000$  RFFs throughout and apply RF nonlinear component analysis (Section V) to obtain a lower-dimensional representation ( $\hat{l} = 100$ ). Following common practice, we assume  $\beta_t = 2$  in Proposition 1, since the theoretical choice is known to be conservative [1].

*a) Source Spiral:* First we consider the plant  $x_{t+1} = Ax_t + \cos(Bx_t + c)$ , where  $A$  is fixed such that the resulting plot exhibits a source-spiral pattern as shown in Fig. 1, and the affine mapping  $(B, c)$  is randomly sampled. We assume as cost function the quadratic form  $c(x_t) = x_t^\top x_t$  and sample initial conditions according to  $x_0 \sim \mathcal{N}(0, I_2)$ . The left column of Fig. 1 depicts the scarce training data setting ( $N = 3, T = 50$ ) and the right column accounts for the large training data regime ( $N = 200, T = 50$ ). We also show the certainty-equivalent prediction (i.e., CERF) alongside the true trajectory (black).

Moreover, we explore the role of the learning rate in the (incremental) Hamiltonian minimization (24) to find the worst plausible dynamics and the corresponding state trajectory in the aforementioned system. In each row panel of Fig. 1 we consider a variation in the scheduling of said learning rate in the Frank-Wolfe algorithm, namely a standard Frank-Wolfe scheduling  $\gamma_e = 2/(e + 2)$ ,  $\gamma_e = 1$  (i.e., exact PMP, Prep. 4) which implies that the optimization successively moves along the uncertainty set boundary and  $\gamma_e = 1/E$ , with  $E$  being the total number of optimization steps. Lighter green denotes earlier iterations of the opti-

mization with the final result being shown in dark green ( $e = E$ ). The optimization behavior is strongly dependent on the used learning rate schedule, with the standard Frank-Wolfe scheduling yielding the highest cost (i.e., the worst) dynamics for the performed experiments. Furthermore, we note that the setting  $\gamma_e = 1$  exhibits a divergent behavior in Fig. 1, which is due to large errors in the Hamiltonian dynamics between consecutive iterations according to [14].

*b) Van der Pol oscillator:* This system is governed by the second-order differential equation  $\dot{x}_1 = (1 - x_2^2)x_1 - x_2, \dot{x}_2 = x_1$ . We use an explicit Runge-Kutta integrator for its discrete-time simulation. We sample initial conditions from  $\mathcal{U}(-1, 1)$  for training data generation. The URF model's prediction and obtained worst- and best-case trajectories are shown in Fig. 3a for the quadratic cost  $c(x_t) = x_t^\top x_t$  for both the scarce- and big-data regimes (left and right, respectively) following a similar convention to Fig. 1.

*c) Pendulum:* we consider the continuous-time pendulum dynamics  $\dot{x}_1 = x_2, \dot{x}_2 = -(g/l) \sin x_1 - (\beta/(ml^2))x_2$ , where the gravity is set to  $g = 9.81$ ; and the mass  $m$ , the rod length  $l$  and the friction parameter  $\beta$  are all set to 1. We sample the initial pendulum's angle from  $\mathcal{U}(-\pi, \pi)$  and the initial angular velocity from  $\mathcal{U}(-1, 1)$  for rollout generation. We discretize the system using the semi-implicit Euler integrator and use a cost function that encourages the pendulum to stay upright<sup>8</sup>. As previously, we show the resulting worst- and best-case trajectories in Fig. 3b.

We highlight that in the absence of large amounts of training data, the computed uncertainty set contains dynamics that could drive the system to high cost regions, as shown in the left columns of Figures 1 and 3. However, once the inferred uncertainty set is reduced as a result of a larger training dataset, the obtained worst-case state trajectory is closer to the CERF trajectory.

A finer-grained analysis of the previous point is presented in Fig. 2, where we see that both the computed worst- and best-case cost ( $U_J$  and  $L_J$ , respectively) of a test trajectory define a range containing the true cost across different training data sizes and systems; i.e.,  $J \in [L_J, U_J]$ . Note that the gaps among the worst, best, mean and true cost vanish after large enough training datasets. However, we emphasize that although we might get poor cost (mean) predictions under scarce training data (e.g., rightmost column of Fig. 2), the true cost is always within the region defined by the best- and worst-case curves. We remark that our cost bounds are computed using  $E = 50$  forward/backward passes in Alg. 1 initialized at the CERF trajectory.

We also consider sampling-based bounds as a baseline where we set the empirical worst-case cost as  $\hat{U}_J = \max_{i=1, \dots, S} J(\hat{W}_0^i, \hat{W}_1^i, \dots, \hat{W}_{H-1}^i)$  by drawing each  $\hat{W}_h^i$  uniformly at random from  $\mathcal{W}_t$  and set  $\hat{L}_J$  analogously by taking the minimum instead. The empirical bounds  $[\hat{L}_J, \hat{U}_J]$  with  $S = 1000$  samples consistently underestimate our optimization-based bounds  $[L_J, U_J]$ , as shown in Fig. 2.

<sup>8</sup>To ease implementation we use as state representation  $\hat{x} = (a, b, c) = (l \cos x_1, l \sin x_1, x_2)$  and define the cost as  $c(\hat{x}) = b^2 - a + 0.1c^2$ .

This is particularly evident in the scarce data regime, which hints that our method can be a more reliable estimator of downstream uncertainty induced by limited training data.

## VII. OTHER RELATED WORKS

Our use of PMP for finding the worst-case URF dynamics is similar to the use of adjoint sensitivity in robust nonlinear optimization [19]. [16] used PMP for deep adversarial learning, but with no equivalence theorems for the constrained optimization. Moreover, they used gradient projection algorithms which typically perform poorly in nonlinear programs. In [20], a PMP-based analysis is presented for Model Predictive Control (MPC) with GP dynamics, however they rely on *approximate* uncertainty propagation.

Our perspective, which views the whole dynamical system as a DNN instead of the one-step dynamics model, is in contrast with a large body of deep learning literature that employ DNN as dynamics model [21].

In the GP context, [10] proposed the Sparse Spectrum GP based on RFF, which has been previously considered to learn dynamics for filtering and control problems in [22] and to sample from GP dynamics in [23].

[24] and [3] make similar assumptions regarding the function class of the unknown dynamics (i.e., an RKHS) but focus on theoretical guarantees for system identification and online control, respectively.

## VIII. DISCUSSION AND FUTURE WORK

By leveraging a RF perspective on GP confidence bounds, we propose in this paper the uncertainty-aware random feature (URF) model for learning unknown dynamical systems from data. We exploit the structure present in our model to devise a PMP-based numerical algorithm to estimate worst- and best-case bounds on quantities that are complex functions of the unknown dynamical system (e.g., cost of an OCP). We also show that this algorithm is equivalent to performing the conditional gradient (i.e., Franke-Wolfe) method on a DNN by viewing our (one-step) URF model as a shallow neural net, which we believe to be of independent interest. We also validate the usefulness of the URF model through various numerical experiments under different amounts of training data. In future work, we envision applications of our dynamics model in data-driven robust control, such as learning-based MPC, or in active learning strategies based on optimism, where we offer an alternative characterization of confidence bounds.

## REFERENCES

- [1] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6059–6066.
- [2] M. Buisson-Fenet, F. Solowjow, and S. Trimpe, "Actively learning Gaussian process dynamics," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, vol. 120. PMLR, Jun. 2020, pp. 5–15.
- [3] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 15 312–15 325.
- [4] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of the 27th International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010, pp. 1015–1022.
- [5] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2011, pp. 465–472.
- [6] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.
- [7] T. Lew and M. Pavone, "Sampling-based reachability analysis: A random set theory approach with adversarial sampling," in *Proceedings of the 2020 Conference on Robot Learning*, vol. 155. PMLR, Nov. 2020, pp. 2055–2070.
- [8] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. The MIT Press, 2005.
- [9] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, vol. 20. Curran Associates, Inc., 2008.
- [10] M. Lázaro-Gredilla, J. Quiñero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *Journal of Machine Learning Research*, vol. 11, no. 63, pp. 1865–1881, 2010.
- [11] F. Liu, X. Huang, Y. Chen, and J. A. K. Suykens, "Random features for kernel approximation: A survey on algorithms, theory, and beyond," *arXiv:2004.11154 [cs, stat]*, Mar. 2021.
- [12] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1.
- [13] F. L. Chernousko and A. A. Lyubushin, "Method of successive approximations for solution of optimal control problems," *Optimal Control Applications and Methods*, vol. 3, no. 2, pp. 101–114, 1982.
- [14] Q. Li, L. Chen, C. Tai, and W. E, "Maximum principle based algorithms for deep learning," *Journal of Machine Learning Research*, vol. 18, no. 165, pp. 1–29, 2018.
- [15] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.
- [16] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [17] H. Zhu, C. K. I. Williams, R. Rohwer, and M. Morciniec, "Gaussian regression and optimal finite dimensional linear models," in *Neural Networks and Machine Learning*. Springer-Verlag, 1997, pp. 167–184.
- [18] D. Lopez-Paz, S. Sra, A. Smola, Z. Ghahramani, and B. Schoelkopf, "Randomized nonlinear component analysis," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32. Beijing, China: PMLR, Jun. 2014, pp. 1359–1367.
- [19] M. Diehl, H. Bock, and E. Kostina, "An approximation technique for robust nonlinear optimization," *Mathematical Programming*, vol. 107, pp. 213–230, Jun. 2006.
- [20] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84. PMLR, Apr. 2018, pp. 1701–1710.
- [21] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.
- [22] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum Gaussian processes with applications to filtering and control," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, Aug. 2017, pp. 2760–2768.
- [23] L. Hewing, E. Arcari, L. P. Fröhlich, and M. N. Zeilinger, "On simulation and trajectory prediction with Gaussian process dynamics," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, vol. 120. The Cloud: PMLR, Jun. 2020, pp. 424–434.
- [24] H. Mania, M. I. Jordan, and B. Recht, "Active Learning for Nonlinear System Identification with Guarantees," *arXiv:2006.10277 [cs, math, stat]*, Jun. 2020.