Learning-driven Nonlinear Optimal Control via Gaussian Process Regression

This is the post peer-review accepted manuscript of:

L. Sforni, I. Notarnicola, and G. Notarstefano "Learning-driven Nonlinear Optimal Control via Gaussian Process Regression," in 2021 60th IEEE Conference on Decision and Control (CDC), Austin, 2021, pp. 4412-4417,

The published version is available online at:

10.1109/CDC45484.2021.9683153

Learning-driven Nonlinear Optimal Control via Gaussian Process Regression

Lorenzo Sforni, Ivano Notarnicola, Giuseppe Notarstefano

***Abstract*— In this paper we propose a novel numerical strategy to solve nonlinear optimal control problems for dynamical systems with partially unknown dynamics. The goal is to explore feasible system trajectories minimizing a given finite-time performance criterion. We suppose to be able to actuate an input sequence on the real system, but only an inaccurate description of the dynamics is available for the control design. The proposed learning-driven optimal control strategy combines a trajectory optimization procedure with a Gaussian process regression to iteratively enrich the model and perform the optimization steps. Thanks to this combined scheme, the strategy is able to explore the trajectory manifold while minimizing the cost function. To corroborate the theoretical results, numerical simulations on the optimal control of a pendulum are shown.**

I. Introduction

Optimal control techniques rely on system models which, if inaccurate, can lead to the design of suboptimal trajectories for the true system. In this paper we aim at achieving the twofold goal of finding optimal trajectories while improving the knowledge on the systems using data.

Literature Review: Classic system identification methods adopt parametric models and exploit observation to tune their parameters to achieve model accuracy [1], [2]. In view of the recent success in the field of machine learning, data-driven control techniques have gained growing interest in the control system community [3], [4]. Different data-driven approaches have been considered, e.g., reinforcement learning [5], model fitting [6], [7] and stochastic nonparametric estimation [8]. In this paper we leverage Gaussian Processes (GP) and their associated nonparametric regression techniques. recognized in the field of controls for its flexibility in modelling complex unknown dynamics [9]. GPs have also been recently exploited in the quantification of model uncertainty [10]. In the field of optimal control, GPs have been exploited as a valid alternative to the prominent approach of modeling uncertainty as a stochastic disturbance. In [11] GPs are used in a dynamic programming framework. In [12], a scenario-based optimal control strategy is proposed based on a GP approximation of the dynamics. In [13] a combined Bayesian approach and GPs is proposed to select the most informative data for optimally updating a nominal model. Many interesting applications can be found also in the field of adaptive control [14]–[16]. GPs have been also successfully applied in robotics [17], [18], aircraft control [19] and power demand management [20].

The authors are with the Department of Electrical, Electronic and Information Engineering, Alma Mater Studiorum - Università di Bologna, Bologna, Italy, name.lastname@unibo.it.

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

Finally, also Model Predictive Control (MPC) schemes based on GP have been investigated [21]–[23]. Among the other works, we refer to [24] where a MPC approach that integrates a nominal system with an additive unknown dynamics modeled as a GP is exploited. The strategy is then extended to autonomous racing driving in [25]. While GPs have been successfully applied, most approaches lack formal guarantees for GP models. Recently, some control approaches with formal guarantees have been developed in [26], [27]. Bounds on the estimation error, in particular, are deeply analyzed in the field of Bayesian optimization [28], [29]. A theoretical analysis of the GP-based MPC controllers is presented in [30].

Contributions: In this paper we propose a learning-based optimization strategy to solve nonlinear finite-horizon optimal control problems with partially unknown dynamics. We propose a two-step iterative procedure in which the optimization process and the learning phase are concurrently performed. Specifically, the unknown term in the dynamics is approximated through an iteratively refined Gaussian process. At each iteration, the current optimal input estimate is improved by a gradient-like update step performed by taking derivatives of the nominal dynamics enhanced with the GP and actuated on the real system. During this experiment, novel measurements from the system evolution are collected and used in the learning phase. Under suitable technical conditions, the proposed strategy is proved to converge to a neighborhood of a stationary point of the optimal control problem. In order to prove the result, the algorithmic updated is recast into a suitable gradient-with-error update with error uniformly bounded across iterations.

The paper unfolds as follows. In Section II we present the problem set-up and some preliminaries. In Section III the learning-based optimal control strategy is presented and then tested in Section IV on a simulation example.

II. Set-up and Preliminaries

In this section we first present the learning-driven optimal control problem set-up. Then, a gradient method approach for optimal control and a regression technique based on Gaussian process are reviewed.

A. Learning-driven Nonlinear Optimal Control

In this paper we focus on discrete-time nonlinear systems described by

$$x_{t+1} = f(x_t, u_t), \qquad t \in \mathbb{T}_{[0,T-1]}, \tag{1}$$

with $\mathbb{T}_{[0,T-1]} := \{0, 1, \dots, T-1\}$, where $x_t \in \mathbb{R}^{n_x}$ is the state and $u_t \in \mathbb{R}^{n_u}$ is the input at time t. The initial condition

$x_0 = x_{\text{init}}$ with $x_{\text{init}} \in \mathbb{R}^{n_x}$ given. The vector field modeling the dynamics $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is assumed to be twice differentiable.

The key challenge addressed in the paper is that the dynamics $f(\cdot)$ is composed by a nominal, known model (e.g., derived from first principles), and by an unknown part as

$$x_{t+1} = f_0(x_t, u_t) + g(x_t, u_t), \qquad t \in \mathbb{T}_{[0,T-1]}, \quad (2)$$

where $f_0 : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ models the nominal dynamics while $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is unknown. We will refer both to (2), when we want to highlight the peculiar structure of the dynamics, and to (1) when we mean the real, though unknown, system.

We investigate nonlinear optimal control problems in which we look for trajectories of the unknown system (1) that minimize a performance criterion defined over a fixed, time horizon $\mathbb{T}_{[0,T]}$. Formally, we aim to solve the problem

$$\min_{\substack{x_1,\dots,x_T \\ u_0,\dots,u_{T-1}}} \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \quad (3a)$$

$$\text{subj. to } x_{t+1} = f(x_t, u_t), \quad t \in \mathbb{T}_{[0,T-1]}, \quad x_0 = x_{\text{init}}, \quad (3b)$$

with stage costs $\ell_t : \mathbb{R}^{n_u T} \to \mathbb{R}$, for all t, and terminal cost $\ell_T : \mathbb{R}^{n_u T} \to \mathbb{R}$. These functions are assumed to be twice differentiable.

The main challenge of the optimal control problem (3) is that the dynamics is only partially known and the presence of the unknown term calls for novel learning techniques to be combined into an optimal control scheme.

For notational convenience, we let $\mathbf{x} := (x_1, \dots, x_T)$ and $\mathbf{u} := (u_0, \dots, u_{T-1})$. A pair $(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^{n_x T} \times \mathbb{R}^{n_u T}$ is called a *trajectory* of the real system if its components satisfy the dynamics (1) for all $t \in \mathbb{T}_{[0,T-1]}$.

Let us also introduce the following shorthand notation

$$a_t^k := \nabla_{x_t} \ell_t(x_t^k, u_t^k), \qquad b_t^k := \nabla_{u_t} \ell_t(x_t^k, u_t^k), \quad (4a)$$

$$A_t^k := \nabla_{x_t} f(x_t^k, u_t^k)^\top, \qquad B_t^k := \nabla_{u_t} f(x_t^k, u_t^k)^\top. \quad (4b)$$

B. Gradient Method for Optimal Control

In this subsection we briefly recall a strategy proposed in [31, Section 1.9] to solve discrete-time optimal control problems (3) based on a gradient method. We point out that here the dynamics (1) is assumed to be known.

The leading idea is to express the state x_t, for all $t \in \mathbb{T}_{[0,T-1]}$, as a function of the input sequence $\mathbf{u}$ only. Indeed, for all t we can formally introduce a map $\phi_t : \mathbb{R}^{n_u T} \to \mathbb{R}^{n_x}$ such that $x_t := \phi_t(\mathbf{u})$. In this way problem (3) can be recast into the so-called reduced version

$$\min_{\mathbf{u} \in \mathbb{R}^{n_u T}} \sum_{t=0}^{T-1} \ell_t(\phi_t(\mathbf{u}), u_t) + \ell_T(\phi_T(\mathbf{u})) = \min_{\mathbf{u} \in \mathbb{R}^{n_u T}} J(\mathbf{u}) \quad (5)$$

where $\mathbf{u} := (u_0, \dots, u_{T-1})$ is the only optimization variable.

Problem (5) is an unconstrained optimization problem in $\mathbf{u}$ with a sufficiently regular cost function. Hence, it can be addressed via a gradient descent method. Let $k \in \mathbb{N}$, then the components of the tentative solution $\mathbf{u}^k \in \mathbb{R}^{n_u T}$ are iteratively updated according to

$$u_t^{k+1} = u_t^k - \beta^k \underbrace{\nabla_{u_t} J(\mathbf{u}^k)}_{-\bar{v}_t^k} \quad (6)$$

for all $t \in \mathbb{T}_{[0,T-1]}$, where the parameter $\beta^k > 0$ is the so-called step-size.

The overall procedure is summarized by Algorithm 1 where we assume that the state trajectory is initialized as $x_0^k = x_{\text{init}}$ for all k and $\bar{\lambda}_T^k = \nabla \ell_T(x_T^k)$.

Algorithm 1 Gradient Method for Optimal Control

for $k = 0, 1, 2 \dots$ **do**
  for $t = T-1, \dots, 0$ **do**
    compute a descent direction

$$\bar{\lambda}_t^k = A_t^{k\top} \bar{\lambda}_{t+1}^k + a_t^k \quad (7a)$$

$$\bar{v}_t^k = -B_t^{k\top} \bar{\lambda}_{t+1}^k - b_t^k \quad (7b)$$

  for $t = 0, \dots, T-1$ **do**
    compute the perturbed input

$$u_t^{k+1} = u_t^k + \beta^k \bar{v}_t^k$$

    run real system

$$x_{t+1}^{k+1} = f(x_t^{k+1}, u_t^{k+1})$$

The constructive derivation underlying Algorithm 1 shows that it generates a sequence of inputs $\{\mathbf{u}^k\}_{k \geq 0}$ that can be associated to a gradient method applied to problem (5). Thus, it inherits its convergence results.

We point out that Algorithm 1 cannot be implemented if the dynamics are partially unknown. In Section III we show how to enhance the method with a learning procedure.

C. Gaussian Process Regression

In this subsection we recall the popular nonparametric regression technique in machine learning based on Gaussian processes as presented, e.g., in [32]. It represents a powerful tool to infer from data a nonlinear vector-valued function $\varphi : \mathbb{R}^{n_z} \to \mathbb{R}^{n_y}$ describing a nonlinear map between the input z and its corresponding output $y = \varphi(z)$. We suppose to have access to a data-set $D := ((z^1, y^1), \dots, (z^H, y^H))$ with each pair $(z^h, y^h) \in \mathbb{R}^{n_z} \times \in \mathbb{R}^{n_y}$ obtained as $y^h = \varphi(z^h) + \epsilon^h$ where $\epsilon^h \in \mathbb{R}^{n_y}$ is a white Gaussian noise with covariance matrix $\sigma_\epsilon^2 I_{n_y}$.

In Gaussian process regression, we assume that values of the components φ_a, $a = 1, \dots, n_y$, of φ are drawn from independent Gaussian distributions. A GP is fully specified by a mean m function $m : \mathbb{R}^{n_z} \to \mathbb{R}^{n_y}$ and a kernel covariance function $\kappa : R^{n_z} \times R^{n_z} \to \mathbb{R}$. To maintain computational feasibility, it is customary to train independent GPs for each component $\varphi_a(\cdot)$ of the vector field $\varphi(\cdot)$.

We choose the commonly adopted squared exponential kernel defined for any $z, z' \in \mathbb{R}^{n_z}$ as

$$\kappa(z, z') = \sigma_\varphi^2 \exp\left(-\frac{\|z - z'\|^2}{2L_\varphi}\right) \quad (8)$$

where $L_\varphi > 0$ denotes the signal length scale while σ_φ^2 is its variance.

Given the data-set D and the kernel covariance function, we introduce the Gram matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{H\times H}$ whose (h,i)-th entry is $\boldsymbol{\mathcal{K}}_{hi} = \kappa(z^h, z^i)$ and the kernel vector $\boldsymbol{\kappa}(z) \in \mathbb{R}^H$ at a generic $z \in \mathbb{R}^{n_z}$ with h-th component $\boldsymbol{\kappa}_h(z) = \kappa(z^h, z)$.

We specify a zero mean prior on $\varphi(\cdot)$ which means that no prior knowledge is available. Based on the previous definitions, the posterior predictive distribution of each component $\varphi_a(\cdot)$ conditioned on the data-set D at a given point z is Gaussian with mean and covariance

$$m_a(z) = \boldsymbol{\kappa}(z)^\top (\boldsymbol{\mathcal{K}} + \sigma_\epsilon^2 I_H)^{-1} Y_a \tag{9a}$$

$$\sigma^2(z) = \kappa(z,z) - \boldsymbol{\kappa}(z)^\top (\boldsymbol{\mathcal{K}} + \sigma_\epsilon^2 I_H)^{-1} \boldsymbol{\kappa}(z), \tag{9b}$$

for all $a = 1, \dots, n_y$, where $Y_a \in \mathbb{R}^H$ collects only the a-th component of measurements $y^h \in Y$. The resulting posterior of the vector field $\varphi(\cdot)$ is $\varphi(z) \sim \mathcal{N}(m(z), \Sigma(z))$ with $m(z) := \text{col}(m_1(z), \dots, m_{n_y}(z))$ and $\Sigma(z) := I_{n_y}\sigma^2(z)$.

In the forthcoming strategy, we will also use the derivative of a GP, which since differentiation is a linear operator, is another Gaussian process [32]. Specifically, each component $m_a(\cdot)$ of $m(\cdot)$ has a posterior multivariate Gaussian $\nabla m_a(z) \sim \mathcal{N}(m'_a(z), \Sigma'_a(z))$ with mean and covariance

$$m'_a(z) := \nabla\boldsymbol{\kappa}(z)^\top (\boldsymbol{\mathcal{K}} + \sigma_\epsilon^2 I_H)^{-1} Y_a \tag{10a}$$

$$\Sigma'_a(z) := \nabla^2\kappa(z,z) - \nabla\boldsymbol{\kappa}(z)^\top (\boldsymbol{\mathcal{K}} + \sigma_\epsilon^2 I_H)^{-1} \nabla\boldsymbol{\kappa}(z) \tag{10b}$$

where $\nabla^2\kappa(z,z) = \frac{\sigma_\varphi^2}{L_\varphi^2} I_H$ while the h-th row of the matrix $\nabla\boldsymbol{\kappa}(z) \in \mathbb{R}^{H\times n_z}$ is $\nabla\boldsymbol{\kappa}_h(z) = \frac{1}{L_\varphi^2}\kappa(z^h, z)\,(z^h - z)^\top$.

III. Learning-driven Optimal Control via Gaussian Process Regression

In this section we present the main contribution of this paper which is the optimization-based control strategy for nonlinear systems with partially unknown dynamics. We start by presenting how to implement the iterative learning phase specifically tailored for dynamics learning and then we embed it in the novel optimal control strategy.

A. Dynamics Learning via Gaussian Process Regression

In this subsection we tailor the GP regression presented in Section II-C to the dynamics learning process. To this end, we start by modeling the unknown dynamics $g(\cdot)$ in (2) using a Gaussian process. To ease the presentation and without loss of generality, we consider in this part a scalar system, i.e., $x \in \mathbb{R}$ and $u \in \mathbb{R}$. Therefore, the unknown function is $g(\cdot) : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$. As a consequence, also the mean and the variance of the GP are scalar functions. The results can be extended to vector-valued functions by considering one scalar GP for each component.

We consider observations of $g(\cdot)$ taken in the following form. For each trajectory $(\mathbf{x}, \mathbf{u})$ of the real system (1), we define T observation pairs $((z^1, y^1), \dots, (z^T, y^T))$ as

$$z^t := (x_t, u_t)$$

$$y^t := \underbrace{x_{t+1} - f_0(x_t, u_t)}_{\varphi(z^t)} + \epsilon^t \tag{11}$$

with $t \in \mathbb{T}_{[0,T-1]}$, ϵ^t white Gaussian noise with variance σ_ϵ^2.

The leading idea is to iteratively refine the GP approximation as the optimization process proceeds. Therefore, we suppose to arrive at a given iteration k with data-set $D^k = (Z^k, Y^k)$ collecting state-input trajectories explored up to k. As described in Section II-C, we assume that $g(\cdot)$ is drawn from a GP prior and we compute its posterior distribution which is Gaussian

$$g(x,u) \sim \mathcal{N}\Big(m^k(x,u), \Sigma^k(x,u)\Big)$$

where the mean and the variance are computed as in (9).

Once the GP regression on $g(\cdot)$ has been posed, we choose to approximate the partially unknown dynamics (1) using a deterministic approach where $g(\cdot)$ is approximated by the posterior mean of the GP, i.e.,

$$x_{t+1} = f_0(x_t, u_t) + m^k(x_t, u_t), \quad t \in \mathbb{T}_{[0,T-1]}. \tag{12}$$

Notice that the use of the squared exponential kernel (8) induces differentiability and boundedness properties to all functions represented by the GP [32]. Thus also the posterior mean function $m^k(x,u)$ is smooth and, as recalled in Section II-C, its derivative is a GP itself

$$\nabla m^k(x_t, u_t) \sim \mathcal{N}(m'^k(x_t, u_t), \Sigma'^k(x_t, u_t)) \tag{13}$$

with mean and covariance computed as in (10).

Clearly, we are also interested in providing a quantitative measure on the quality of the approximation made using the approximation described so far. We define the model estimation error $\Delta g^k(x_t, u_t) \in \mathbb{R}$, for all k as

$$\Delta g^k(x_t, u_t) := |g(x_t, u_t) - m^k(x_t, u_t)|.$$

Due to the stochastic nature of regression framework, the approximation error $\Delta g^k(x_t, u_t)$ can be characterized only probabilistically. According to [28], the maximum estimation error $\Delta g^k(\cdot)$ can be bounded with high-probability only for a restricted class of functions as stated in the next assumption.

Assumption 3.1: The function $g(\cdot)$ belongs to the reproducing kernel Hilbert space (RKHS) associated to the kernel function $\kappa(\cdot,\cdot)$ in (8). Moreover, it has bounded RKHS norm with respect to $\kappa(\cdot,\cdot)$, i.e., $\|g(\cdot)\|_\kappa^2 \le B_g$. □

We now recall a result based on [28, Thm. 6].

Lemma 3.2: Let Assumption 3.1 hold and define

$$\rho_\delta^k := \sqrt{2B_g + 300\cdot\gamma^k \log^3((|D^k|+1)/\delta)}$$

for all $\delta \in (0,1)$, where $|D^k|$ is the cardinality of the D^k, γ^k is the maximum mutual information[1] that can be obtained about $g(\cdot)$ from the data-set D^k. Then, for all $\delta \in (0,1)$, it holds

$$\Pr\Big\{ |m^k(x,u) - g(x,u)| \le \rho_\delta^k \Sigma^k(x,u), \quad \forall\,(x,u),\ k \in \mathbb{N} \Big\} \ge 1-\delta,$$

[1] See [28, Sec. IV] for a definition of γ^k. Informally, it quantifies the quality of the data for the learning purposes.

with $m^k(\cdot)$ and $\Sigma^k(\cdot)$ being the posterior mean and variance given data-set D^k. □

If, moreover, we assume that the data points belong to a compact set, i.e., $(x, u) \in \mathbb{X}\times\mathbb{U}$ with $\mathbb{X}$ and $\mathbb{U}$ being compact sets, then a uniform bound on γ^k can be established [28]. Therefore, under the compactness requirement, Lemma 3.2 also provides a uniform quantitative bound on the error made by the approximation.

B. GP-Enhanced Gradient for Optimal Control

Let us now consider the optimal control problem (3) in which the dynamics is only partially known. We exploit the GP regression presented in Section III-A to include an iteratively refined approximation of the unknown term $g(\cdot)$ in the optimal control strategy. Specifically, we insert the learning procedure in the optimal control strategy and let both optimization and learning be concurrently performed as shown in Figure 1.

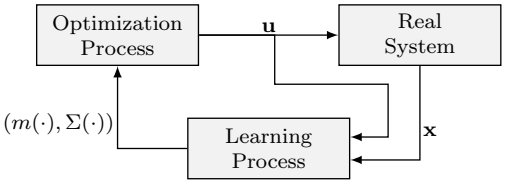

Fig. 1. Scheme representing the information flow of the proposed strategy.

Given the data-set D^k and the corresponding approximation at iteration k, the optimal control problem to be solved can be written as

$$\begin{aligned} \min_{\substack{x_1,\dots,x_T \\ u_0,\dots,u_{T-1}}} \; & \sum_{t=0}^{T-1} \ell_t(x_t, u_t) + \ell_T(x_T) \\ \text{subj. to } & x_{t+1} = f_0(x_t, u_t) + m^k(x_t, u_t), \\ & x_0 = x_{\text{init}}, \;\; t \in \mathbb{T}_{[0,T-1]}, \end{aligned} \tag{14}$$

in which the dynamics constraint includes the posterior mean $m^k(\cdot)$ in place of the unknown term $g(\cdot)$.

With problem formulation (14) in place, we can resort to the approach described in Section II-B. Notice that the adjoint system (7) cannot be computed as done earlier in Section II-B since $g(\cdot)$ and, hence, A_t^k, B_t^k are not known. Therefore, we consistently adapt the adjoint system (7) to compute the descent direction based on the approximated dynamics. The GP approximation of the unknown vector field allows us to easily evaluate the linearization matrices about any point (x_t^k, u_t^k) being the derivative of the mean function a GP itself. Thus, let the shorthands in (4b) be adapted as

$$\begin{aligned} A_t^k &\longmapsto \hat{A}_t^k + m_x'^k(x_t^k, u_t^k) \\ B_t^k &\longmapsto \hat{B}_t^k + m_u'^k(x_t^k, u_t^k), \end{aligned}$$

where $m_x'^k(x_t^k, u_t^k)$ and $m_u'^k(x_t^k, u_t^k)$ are the components of the mean function $m'^k(x_t^k, u_t^k)$ in (13) corresponding to the state and to the input, respectively. As for the cost linearization a_t^k and b_t^k they are defined as before, i.e., as in (4a). The descent direction based can be therefore computed as shown in (15).

Next, the input sequence $\mathbf{u}^k$, is applied to the real system through an experiment and, then, T novel measurements are taken from the resulting state trajectory $\mathbf{x}^k$ as in (11). These measurements are then included in the new data-set $D^{k+1} = (Z^{k+1}, Y^{k+1})$. The proposed method described so far is summarized in Algorithm 2.

Algorithm 2 GP-Enhanced Gradient for Optimal Control

for $k = 0, 1, 2\dots$ with $\lambda_T^k = \nabla\ell_T(x_T^k)$ **do**
  for $t = T-1, \dots, 0$ **do**
    compute descent direction

$$\lambda_t^k = \left(\hat{A}_t^k + m_x'^k(x_t^k, u_t^k)\right)^\top \lambda_{t+1}^k + a_t^k \tag{15a}$$

$$v_t^k = -\left(\hat{B}_t^k + m_u'^k(x_t^k, u_t^k)\right)^\top \lambda_{t+1}^k - b_t^k \tag{15b}$$

  for $t = 0, \dots, T-1$ **do**
    compute the perturbed input

$$u_t^{k+1} = u_t^k + \beta^k v_t^k \tag{16}$$

    run the real system

$$x_{t+1}^{k+1} = f(x_t^{k+1}, u_t^{k+1}),$$

    collect a measurement

$$y_t^{k+1} = x_{t+1}^{k+1} - f_0(x_t^{k+1}, u_t^{k+1}) + \epsilon_t^{k+1}$$

    update data-set (Z^{k+1}, Y^{k+1}).

C. Algorithm Analysis

In this subsection we analyze Algorithm 2 showing that Algorithm 2 realizes a gradient descent method with error.

Let us rewrite the approximate dynamics (12) into an equivalent form as

$$\begin{aligned} x_{t+1} &= f_0(x_t, u_t) + m^k(x_t, u_t) \pm g(x_t, u_t) \\ &= f(x_t, u_t) + \Delta^k(x_t, u_t) \end{aligned} \tag{17}$$

where $\Delta^k(x_t, u_t) := m^k(x_t, u_t) - g(x_t, u_t)$.

Let $\Delta_{x,t}'^k$ and $\Delta_{u,t}'^k$ be the matrices obtained, respectively, by differentiating $\Delta^k(x, u)$ with respect to x and u about any state-input pair (x_t^k, u_t^k).

Assumption 3.3: The time-varying matrices $\{\Delta_{x,t}'^k\}_{t\in\mathbb{T}_{[0,T-1]}}$ and $\{\Delta_{u,t}'^k\}_{t\in\mathbb{T}_{[0,T-1]}}$ are uniformly bounded in t and k. □

Assumption 3.4: The set $\mathbb{U} \subset \mathbb{R}^{n_u T}$ is compact and the vector $\mathbf{u}^k \in \mathbb{U}$ for all k. □

Theorem 3.5: Let Assumption 3.3 and 3.4 hold. Hence, any limit point $\mathbf{u}^*$ of $\{\mathbf{u}^k\}_{k\geq 0}$ generated by Algorithm 2 belongs to a neighborhood of a stationary point of problem (5).

Proof: The proof relies on showing that the strategy implemented in Algorithm 2 is equivalent to a gradient method with error, which is then proved to be bounded. In light of (17), the adjoint system in (15a) can be rearranged

equivalently as

$$\lambda_t^k = \underline{A_t^{k\top}\lambda_{t+1}^k + a_t^k} + \Delta_{x,t}'^{k\top}\lambda_{t+1}^k, \tag{18}$$

where the underlined quantity is the term associated to the true dynamics $f(\cdot)$.

We can write $\lambda_t^k = \bar{\lambda}_t^k + \Delta\lambda_t^k$, where $\Delta\lambda_t^k$ evolves according to

$$\Delta\lambda_t^k = (A_t^k + \Delta_{x,t}'^k)^\top \Delta\lambda_{t+1}^k + \Delta_{x,t}'^{k\top}\bar{\lambda}_{t+1}^k \tag{19}$$

with terminal condition $\Delta\lambda_T^k = 0$.

The descent direction in (15b) is an algebraic time-varying map depending on λ_{t+1}^k and b_t^k and can be expressed as

$$\begin{aligned} v_t^k &\overset{(a)}{=} -B_t^{k\top}\bar{\lambda}_{t+1}^k - b_t^k - B_t^{k\top}\Delta\lambda_{t+1}^k - \Delta_{u,t}'^{k\top}\lambda_{t+1}^k \\ &\overset{(b)}{=} \bar{v}_t^k + \Delta v_t^k \end{aligned}$$

where in (a) we have used (III-C), and in (b) the term $\bar{v}_t^k$ is the same computed by the full-knowledge Algorithm 1 and we have defined

$$\Delta v_t^k := -(B_t^k - \Delta_{u,t}'^k)^\top \Delta\lambda_{t+1}^k - \Delta_{u,t}'^{k\top}\bar{\lambda}_{t+1}^k. \tag{20}$$

Therefore, the components of the updated input $\mathbf{u}^{k+1}$ are

$$u_t^{k+1} = \underline{u_t^k + \beta^k\,\bar{v}_t^k} + \beta^k\,\Delta v_t^k \tag{21}$$

where $\bar{v}_t^k$ is a t-th component of the gradient $-\nabla J(\mathbf{u}^k)$ of the cost function (cf. problem (5)) using the true, though unknown, dynamics as described in Section II-B.

By forward simulation of the dynamics (1) we can write $\mathbf{x}^k := \phi(\mathbf{u}^k)$, for all k, with $\phi(\cdot)$ being a continuous map since f is smooth (cf. Section II-B). Therefore, $\mathbf{x}^k \in \mathbb{X}$ for all k where $\mathbb{X} = \phi(\mathbb{U})$ is a compact set. Consider now the linearization of the dynamics as in (4b). From $\mathbf{u}^k \in \mathbb{U}$, we have that $A_t^k = \nabla_{x_t} f(\phi_t(\mathbf{u}^k), u_t^k)^\top$ is uniformly bounded in t and in k. That is, for all t and k it holds $\|A_t^k\| \le A_0$ and $\|B_t^k\| \le B_0$ for some $A_0 > 0$ and $B_0 > 0$. By exploiting the linearity of (7a) and defining $\bar{\boldsymbol{\lambda}}^k := (\bar{\lambda}_1^k, \dots, \bar{\lambda}_T^k)$, we can write $\bar{\boldsymbol{\lambda}}^k = \hat{\Phi}_1^k\bar{\lambda}_T^k + \hat{R}_1^k\mathbf{a}^k$ for suitably defined matrices $\hat{\Phi}_1^k$ (collecting the state transition matrices for each t, made by bounded state matrices) and $\hat{R}_1^k$ (involving the convolution between the state and the input matrices, both bounded) where $\mathbf{a}^k := (a_0^k, \dots, a_{T-1}^k)$. Since both $\bar{\lambda}_T^k$ and $\mathbf{a}^k$ are bounded, we can write $\|\bar{\boldsymbol{\lambda}}^k\| \le c_1$ for all k and for some $c_1 > 0$. Then, using similar arguments for the linear system (19), we introduce $\Delta\boldsymbol{\lambda}^k := (\Delta\lambda_1^k, \dots, \Delta\lambda_T^k)$ and write

$$\Delta\boldsymbol{\lambda}^k = \hat{\Phi}_2^k \underbrace{\Delta\lambda_T^k}_{=0} + \hat{R}_2^k\bar{\boldsymbol{\lambda}}^k$$

for suitable matrices $\hat{\Phi}_2^k$ and $\hat{R}_2^k$, which are defined similarly to $\hat{\Phi}_1^k$ and $\hat{R}_1^k$. The norm of $\Delta\boldsymbol{\lambda}^k$ can be bounded as $\|\Delta\boldsymbol{\lambda}^k\| \le c_2\,\|\bar{\boldsymbol{\lambda}}^k\| \le c_2c_1$ for all k and for some $c_2 > 0$. Finally, stacking the components Δv_t^k in a single vector $\Delta\boldsymbol{v}^k$ we can use (20) to compactly write $\Delta\boldsymbol{v}^k = (\hat{C}^k R_2^k + \hat{D}^k)\bar{\boldsymbol{\lambda}}^k$ for suitably defined matrices $\hat{C}^k$ and $\hat{D}^k$. Taking the norms we can write

$$\|\Delta\boldsymbol{v}^k\| \le \|\hat{C}^k R_2^k + \hat{D}^k\|\|\bar{\boldsymbol{\lambda}}^k\| \le c_3c_2c_1 \tag{22}$$

for all k and for some $c_3 > 0$.

Then, Algorithm 2 generates a sequence $\{\mathbf{u}^k\}_{k\ge0}$ that can be associated to a gradient method with error given by (21). Being the error uniformly bounded by (22), we can invoke convergence results for gradient method with error (see, e.g., [31, Chap. 1]) to conclude the proof. ∎

It is worth noting that Assumption 3.3 is reasonable since one expects $m^k(\cdot)$ get close to $g(\cdot)$ when data are informative and Assumption 3.1 is satisfied. Assuming that $(x, u) \in \mathbb{X} \times \mathbb{U}$, with $\mathbb{X}$ and $\mathbb{U}$ compact sets, Lemma 3.2 gives a uniform quantitative bound on the distance of the posterior mean from $g(\cdot)$. Moreover, if also $\nabla g(\cdot)$ lives in a RKHS then a uniform bound on the derivatives can be established as well.

Theorem 3.5 states that the scheme generates a sequence $\mathbf{u}^k$ converging to a neighborhood of a stationary point. However, if, as just discussed, $m(\cdot)$ approaches the unknown value of $g(\cdot)$, then in the limit $\Delta\boldsymbol{\lambda}^k$ vanishes, thus giving a vanishing error in the gradient scheme.

IV. Numerical Simulations

We consider, as testbed, a pendulum governed by the dynamics $Ml^2\ddot{\theta} = -Mgl\sin(\theta) - \mathrm{f}_\ell l\dot{\theta} - \mathrm{f}_c l\dot{\theta}^3 + u$. where M is the mass, l is the pendulum length, g is the gravity acceleration, $\mathrm{f}_\ell = \mathrm{f}_{\ell,0} + \Delta\mathrm{f}_\ell$ is the linear friction coefficient and $\mathrm{f}_c = \mathrm{f}_{c,0} + \Delta\mathrm{f}_c$ is the cubic friction coefficient. We suppose to partially know the linear coefficient, i.e. $\mathrm{f}_{\ell,0}$ is known, while the cubic term is unmodeled, i.e., $\mathrm{f}_c = \Delta\mathrm{f}_c$.

Let us consider a discrete-time state-space representation of the pendulum dynamics, obtained for simplicity via forward Euler. By making explicit the uncertain terms, we can obtain a system in the form (1), i.e., as the sum of a known and unknown term given respectively by

$$f_0(x_t, u_t) = \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} + \delta \begin{bmatrix} x_{2,t} \\ a_0\sin(x_{1,t}) + b_0 x_{2,t} \end{bmatrix} + \delta \begin{bmatrix} 0 \\ d_0 \end{bmatrix} u_t$$

with $a_0 = -\frac{g}{l}$, $b_0 = -\frac{\mathrm{f}_{\ell,0}}{M_0 l}$, $d_0 = \frac{1}{M_0 l^2}$, and

$$g(x_t, u_t) = \delta \begin{bmatrix} x_{2,t} \\ \Delta b x_{2,t} + \Delta c x_{2,t}^3 \end{bmatrix}$$

with $\Delta b = -\frac{\Delta\mathrm{f}}{M_0 l}$, $\Delta c = -\frac{\Delta\mathrm{f}_c}{M_0 l}$. and where x_1 corresponds to θ, x_2 corresponds to $\dot{\theta}$, δ is the sampling period. We consider as nominal parameters $l = 1$ m, $M_0 = 1$ Kg, $\mathrm{f}_{\ell,0} = 0.5$ Nm$\frac{\mathrm{s}}{\mathrm{rad}}$ and $\mathrm{f}_{c,0} = 0$ Nm$\frac{\mathrm{s}}{\mathrm{rad}}$. The uncertain terms are $\Delta\mathrm{f}_\ell = 37.5$ Nm$\frac{\mathrm{s}}{\mathrm{rad}}$ and $\Delta\mathrm{f}_c = 30$ Nm$\frac{\mathrm{s}}{\mathrm{rad}}$.

We consider a tracking problem, which translates in a quadratic cost function with $\ell_t(x_t, u_t) = \|x_t - x_{\mathrm{ref},t}\|_Q^2 + \|u_t - u_{\mathrm{ref},t}\|_R^2$ and $\ell_T(x_T) = \|x_T - x_{\mathrm{ref},T}\|_{Q_f}^2$ where $Q = \mathrm{diag}(10, 1)$, $Q_f = \mathrm{diag}(10, 100)$ and $R = 10^{-3}$. We set the sampling period to $\delta = 10^{-3}s^{-1}$. As reference trajectory we used a step signal between two equilibrium configurations.

The step-size is diminishing with $\beta^k = 0.1/k^{0.009}$. The parameters of the squared exponential kernel (8) are $\sigma_\varphi^2 = 1$ and $L_\varphi = \mathrm{diag}(10^{-4}, 10, 10^{-4})$.

The cost error is represented in Figure 2 and shows the difference between $J(\mathbf{u}^k)$, the cost evaluated at the k-th iteration of the GP-enhanced algorithm, and $J(\mathbf{u}^*)$, the optimal cost computed with a full knowledge of $f(\cdot)$.

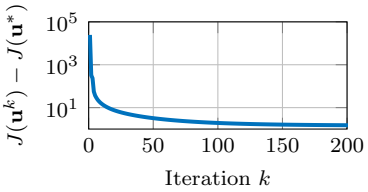

Fig. 2. Evolution of the cost error across iterations.

In Figure 3 a comparison between the state-input trajectories of Algorithms 1 and 2 is proposed. It can be appreciated that the discrepancies between the real system and its nominal model are well captured by the GP regression. Indeed the red and the blue trajectories overlap showing the effectiveness of the proposed approach.

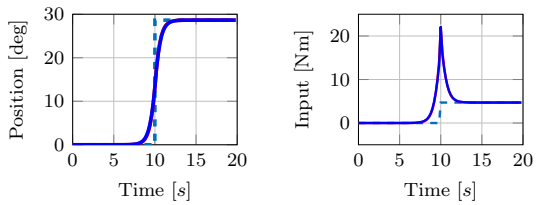

Fig. 3. Comparison among the reference curve (dashed blue) and the results of the algorithms based on full-knowledge (red) and on GP regression (blue).

V. Conclusions

In this paper we presented an optimal control strategy that combines a gradient method with a learning procedure based on Gaussian process regression to deal with nonlinear systems with partially unknown dynamics. We proposed a novel optimization-based algorithm in which the unknown term in the dynamics is approximated with the mean of a Gaussian process which is iteratively refined concurrently to the optimization procedure.

References

[1] L. Ljung, "System identification," *Wiley encyclopedia of electrical and electronics engineering*, pp. 1–19, 1999.

[2] A. Chiuso and G. Pillonetto, "System identification: A machine learning perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 281–304, 2019.

[3] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 259–286, 2018.

[4] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[5] S. Gros and M. Zanon, "Data-driven economic nmpc using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2019.

[6] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[7] S. Dean, S. Tu, N. Matni, and B. Recht, "Safely learning to control the constrained linear quadratic regulator," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 5582–5588.

[8] M. Liu, G. Chowdhary, B. C. Da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Systems Magazine*, vol. 38, no. 5, pp. 53–86, 2018.

[9] J. Kocijan, *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[10] L. Yingzhao and C. Jones, "On Gaussian process based Koopman operator," in *IFAC World Congress*, 2020.

[11] M. P. Deisenroth, J. Peters, and C. E. Rasmussen, "Approximate dynamic programming with Gaussian processes," in *American Control Conference (ACC)*, 2008, pp. 4480–4485.

[12] J. Umlauft, T. Beckers, and S. Hirche, "Scenario-based optimal control for Gaussian process state space models," in *European Control Conference (ECC)*, 2018, pp. 1386–1392.

[13] A. Jain, T. Nghiem, M. Morari, and R. Mangharam, "Learning and control using Gaussian processes," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pp. 140–149.

[14] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler–Lagrange systems," *Automatica*, vol. 103, pp. 390–397, 2019.

[15] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2019.

[16] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *International Conference on Robotics and Automation (ICRA)*, 2016, pp. 491–496.

[17] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[18] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[19] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using Gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 537–550, 2014.

[20] T. X. Nghiem and C. N. Jones, "Data-driven demand response modeling and control of buildings with gaussian processes," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 2919–2924.

[21] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[22] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2017.

[23] M. Binder, G. Darivianakis, A. Eichler, and J. Lygeros, "Approximate explicit model predictive controller using Gaussian processes," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 841–846.

[24] 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, 2019.

[25] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[26] Y. Fanger, J. Umlauft, and S. Hirche, "Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3913–3919.

[27] T. Beckers, J. Umlauft, and S. Hirche, "Stable model-based control with Gaussian process regression for robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3877–3884, 2017.

[28] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[29] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6059–6066.

[30] M. Maiworm, D. Limon, J. M. Manzano, and R. Findeisen, "Stability of Gaussian process learning based output feedback model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 455–461, 2018.

[31] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[32] C. E. Rasmussen, C. K. Williams, and F. Bach, *Gaussian Processes for Machine Learning*. MIT Press, 2006.