Backstepping Tracking Control Using Gaussian Processes with Event-Triggered Online Learning

Junjie Jiao*, *Member, IEEE*, Alexandre Capone*, *Student Member, IEEE*, and Sandra Hirche, *Fellow, IEEE*

***Abstract*—In this paper, we present a trajectory tracking control law for a class of partially unknown nonlinear systems that combines backstepping and event-triggered online learning. We employ Gaussian processes to learn the unknown system model using measurement data collected online, while the proposed control law is active. Our approach uses an efficient event-triggered online learning scheme that exclusively collects informative data to update the estimated model used for control. The resulting control law guarantees that the tracking error is globally uniformly ultimately bounded. The inter-event time is shown to be lower-bounded by a positive constant. Moreover, we also discuss how to obtain a trade-off between the cardinality of the collected training data and the size of the ultimate tracking error bound. In a simulation example, our approach is shown to outperform a state-of-the-art offline learning-based approach both in terms of tracking performance and data efficiency.**

***Index Terms*—Backstepping, Gaussian processes, nonlinear systems, event-triggered learning, online learning.**

I. Introduction

In recent years, learning-based methods using Gaussian processes (GPs) have been successfully applied to a variety of control problems, including feedback linearization [1], backstepping [2], and model predictive control [3], to name a few. As in most GP-based methods, system models are learned *offline* with *fixed* training data sets collected exclusively before the control design. In such settings, if the collected data lies far away from the desired trajectory, the model error will typically be large around the desired trajectory, leading to potentially poor tracking performance. To remedy this, one needs to resort either to complex exploration strategies [4] or multiple iterations of training and control, until the data required for the desired performance is obtained. However, both approaches can be time-consuming and might produce data that is unnecessary for control.

By contrast, *online* learning-based GP methods collect data and update the GP model while the proposed control law is active. As a result, the collected data points are potentially more beneficial for performance than data collected offline, since they correspond to states visited during control. However, due to the adaptive nature of online learning-based control, formal guarantees are much more difficult to obtain. Moreover, since GPs scale poorly with the number of training data points, computationally cheaper approximations might still be required, e.g., sparse GPs, which typically favor overall prediction quality over control performance and seldom yield theoretical guarantees [5]. In spite of these challenges, a handful of online learning-based control strategies using GPs are available. In [6], a safe exploration algorithm is proposed that simultaneously estimates and explores the region of attraction of a control policy using GPs. An optimal experimental design approach is presented in [7] where information theoretical quantities are maximized, yielding a faster learning rate than uniform random sampling. In [8], an uncertainty-triggered online learning scheme is proposed for obtaining tracking control laws, and a stability analysis is provided using stochastic stability theory for switching systems. Event-triggered methods that use models other than GPs are provided in [9] and [10], where event-triggered learning is used to reduce communication for linear stochastic systems. In [11], an event-triggered online learning approach is proposed where data points are collected only when a Lyapunov decrease condition is violated. However, most of these approaches either provide no guarantees or collect data that is unnecessary for control, while [11] only considers single input controllable canonical systems and collects an undetermined amount of data, narrowing its applicability.

In this work, we present an event-triggered online learning-based backstepping control method using GPs for a broad class of partially unknown nonlinear systems. During control, whenever the GP model is too inaccurate to satisfy a Lyapunov decrease condition, a new data point is collected and used to update the GP model. This way, only strictly necessary data is collected, avoiding the high computational complexity of GPs with large data sets.

The main contributions of this paper are as follows: (i) We provide an efficient event-triggered online learning scheme that updates a GP model for the control law. (ii) We show that the resulting tracking error is globally uniformly ultimately bounded and that the inter-event time is lower-bounded by a positive constant. (iii) We show how to obtain a trade-off between the cardinality of data collected online and the size

*Equal contribution.

This work was supported in part by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 899987, and in part by the European Research Council Consolidator Grant Safe data-driven control for human-centric systems (CO-MAN) under grant agreement number 864686.

The authors are with the Chair of Information-oriented Control, TUM School of Computation, Information and Technology, Technical University of Munich, Munich, Germany. Email: junjie.jiao@tum.de; alexandre.capone@tum.de; hirche@tum.de

of the ultimate tracking error bound. (iv) Using a simulation example, we show that our approach outperforms a state-of-the-art offline learning-based control method, with the added benefit of requiring less training data to do so. The present paper extends the result in [11] on single input controllable canonical systems to a broader class of nonlinear systems, and also generalizes the results from [2] to accommodate the event-triggered online learning case.

The outline of this paper is as follows. In Sec. II, we formulate the backstepping control problem with online data collection. In Sec. III we review Gaussian process regression. We then present our event-triggered learning backstepping approach and discuss the trade-off between the data cardinality and tracking performance, in Sec. IV. In Sec. V we compare our approach to a state-of-the-art method with offline learning. In Sec. VI, we provide a brief conclusion.

II. Problem formulation

We consider a nonlinear system in strict feedback form[1]

$$\begin{aligned}\dot{x}_i &= f_i(\mathbf{w}_i) + g_i(\mathbf{w}_i)x_{i+1} + h_i(\mathbf{w}_i), i = 1, \dots, n-1,\\ \dot{x}_n &= f_n(\mathbf{w}_n) + g_n(\mathbf{w}_n)u + h_n(\mathbf{w}_n),\end{aligned} \tag{1}$$

where $x_i \in \mathbb{X}_i \subseteq \mathbb{R}$, $i = 1, 2, \dots, n$ are the states, $\mathbb{X}_i$ are compact sets, $u \in \mathbb{U} \subseteq \mathbb{R}$ is the control input, an the vectors $\mathbf{w}_i = (x_1, x_2, \dots, x_i) \in \mathbb{X}_1 \times \mathbb{X}_2 \times \dots \times \mathbb{X}_i := \mathbb{W}_i$ denote the concatenation of the states. Many practical systems can be expressed in the form of (1), e.g., rigid robots and motors [12], and jet engines [13]. We consider the case where the nonlinear functions $f_i : \mathbb{W}_i \mapsto \mathbb{R}$ and $g_i : \mathbb{W}_i \mapsto \mathbb{R}$ are *known*, whereas the nonlinear functions $h_i : \mathbb{W}_i \mapsto \mathbb{R}$ are *unknown*, representing unmodeled nonlinearities. Furthermore, for all $i = 1, 2, \dots, n$, we assume that the functions $f_i(\cdot)$ vanish at the origin, which is a common assumption for systems in strict feedback form [13]. The unknown unmodeled nonlinearities $h_i(\cdot)$ can be due to, e.g., friction [12]. The system (1) is to be controlled using a backstepping approach. We then make the following standard assumptions for backstepping control [13].

Assumption 1: For all $i = 1, 2, \dots, n$, the functions $f_i(\cdot)$ and $g_i(\cdot)$ are bounded and $n - i$ continuously differentiable.

Assumption 2: For all $i = 1, 2, \dots, n$, the functions $g_i(\cdot)$ are invertible within $\mathbb{W}_i$.

Note that, although Assumption 2 might seem restrictive, it can in some settings be circumvented by exploiting physical properties of the system [12].

Although the functions $h_i(\cdot)$ are unknown, we assume that we can collect measurements of the system (1) at arbitrary times to learn models of $h_i(\cdot)$, while the control law is active.

Assumption 3: At arbitrary countable time instances t_m, $m \in \mathbb{N}$, we can collect noiseless measurements of the system states $\mathbf{w}_i^{(m)} := \mathbf{w}_i(t_m)$, $i = 1, \dots, n$, and noisy measurements $y_i^{(m)} = h_i(\mathbf{w}_i^{(m)}) + \varepsilon$, where $\varepsilon \sim \mathsf{N}(0, \sigma_\varepsilon^2)$ and $\sigma_\varepsilon > 0$.

[1] We denote by $\mathbb{R}$ the field of real numbers, $\mathbb{R}_+$ the field of real positive numbers, and $\mathbb{N}$ the set of positive integers. The field of n dimensional vectors over $\mathbb{R}$ is denoted by $\mathbb{R}^n$. We use $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ to denote the matrix A with the ij-th component being a_{ij}. We denote by $\mathrm{diag}(d_1, d_2, \cdots, d_n)$ the $n \times n$ diagonal matrix with $d_1, d_2, \cdots, d_n$ on the diagonal. We use $|\cdot|$ to denote the absolute value and $\|\cdot\|$ to denote the Euclidean norm. We use $\mathsf{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2.

The exact measurement of the system state is a common assumption for backstepping control [14]. Essentially, the noisy measurements $y_i^{(m)}$ require noisy observations of the time derivative of the states $\dot{x}_i$, since $f_i(\cdot)$ and $g_i(\cdot)$ are known. In practice, $\dot{x}_i$ can be approximated through finite differences. One can also consider the approximation error as part of the measurement noise [2].

Using data collected according to Assumption 3, we define, for every $\kappa \in \mathbb{N}$, the *time-varying* data sets consisting of the first κ collected measurements as

$$\mathbb{D}_{i,\kappa} = \left\{\mathbf{w}_i^{(m)}, y_i^{(m)}\right\}_{m=1,\dots,\kappa}, \quad i = 1, 2, \dots, n. \tag{2}$$

Note that the data sets (2) are updated at time t_κ and remain constant until time $t_{\kappa+1}$.

In this paper, we train Gaussian process (GP) models based on the data (2) to estimate the unknown models h_i, $i = 1, 2, \dots, n$. To this end, in the following we make an assumption regarding the reproducing kernel Hilbert space (RKHS) norm of the unknown functions $h_i(\cdot)$, denoted by $\|h_i\|_{\mathsf{k}_i}$. An RKHS is a Hilbert space induced by a positive definite function $\mathsf{k}_i : \mathbb{W}_i \times \mathbb{W}_i \mapsto \mathbb{R}$, which is called *kernel*. In rough terms, the RKHS norm of a function $h_i(\cdot)$ measures its smoothness with respect to $\mathsf{k}_i(\cdot, \cdot)$. In the following, we consider the commonly used squared-exponential kernel

$$\mathsf{k}_i(\mathbf{w}_i, \mathbf{w}_i') = \sigma_{h_i}^2 \exp\Big(-\frac{1}{2}(\mathbf{w}_i - \mathbf{w}_i')^\top L_i^{-2}(\mathbf{w}_i - \mathbf{w}_i')\Big), \tag{3}$$

where the matrix of lengthscales $L_i = \mathrm{diag}(l_{i,1}, l_{i,2}, \dots, l_{i,i})$, $l_{i,1}, l_{i,2}, \dots, l_{i,i} \in \mathbb{R}_+$, and the signal variance $\sigma_{h_i}^2 \in \mathbb{R}_+$, capture the rate of change and magnitude of the underlying function, respectively. The corresponding RKHS is extremely rich, as its elements can approximate uniformly any continuous function [15]. As such, the following assumption is considerably less restrictive than assuming a parametric model for $h_i(\cdot)$.

Assumption 4: For all $i = 1, 2, \dots, n$, the function $h_i(\cdot)$ in (1) has a bounded RKHS norm with respect to a known squared-exponential kernel k_i, and a corresponding upper bound $0 < B_{h_i} < \infty$ with $\|h_i\|_{\mathsf{k}_i} \leq B_{h_i}$ is known.

Since the kernel k_i encodes information about h_i, Assumption 4, together with the choice of kernel (3), implies that the functions h_i are smooth and bounded. Although typically we are not able to compute the exact RKHS norm of the unknown functions $h_i(\cdot)$ a priori, in practice we can still choose B_{h_i} very large, or obtain an estimate for B_{h_i} by using a guess-and-doubling approach [16].

Our objective is to design a data-efficient event-triggered online learning-based control scheme, such that the state x_1 accurately tracks a desired trajectory $x_d(t) := x_d$. We make the following assumption on the desired trajectory x_d, which is a standard assumption for backstepping control [13].

Assumption 5: The desired trajectory $x_d \in \mathbb{R}$ and its derivatives with respect to time are bounded.

III. Gaussian processes

In this section, we briefly review Gaussian process regression and provide some preliminary theoretical results.

Formally, a GP is a collection of random variables of which any finite subset obeys a joint normal distribution. It is fully specified by a prior mean, which we set to zero in the following without loss of generality, and a positive-definite kernel $\mathsf{k}_i : \mathbb{W}_i \times \mathbb{W}_i \mapsto \mathbb{R}$. Given an unknown function $h_i : \mathbb{W}_i \subseteq \mathbb{R}^i \mapsto \mathbb{R}$ and a corresponding set of noisy measurements $\mathbb{D}_{i,\kappa}$, we are able to compute the posterior mean and variance of the GP at an arbitrary test point $\mathbf{w}_i^*$ conditioned on the data $\mathbb{D}_{i,\kappa}$ as

$$\mu_{i,\kappa}(\mathbf{w}_i^*) = \mathbf{k}_i^\top (\mathsf{K}_i + \sigma_\varepsilon^2 I_\kappa)^{-1} \mathbf{y}_i, \tag{4a}$$

$$\sigma_{i,\kappa}^2(\mathbf{w}_i^*) = \mathsf{k}_i^* - \mathbf{k}_i^\top (\mathsf{K}_i + \sigma_\varepsilon^2 I_\kappa)^{-1} \mathbf{k}_i, \tag{4b}$$

where $\mathbf{y}_i = (y_i^{(1)}, \dots, y_i^{(\kappa)})^\top$, $\mathsf{k}_i^* = \mathsf{k}_i(\mathbf{w}_i^*, \mathbf{w}_i^*)$, $\mathbf{k}_i = (\mathsf{k}_i(\mathbf{w}_i^{(1)}, \mathbf{w}_i^*), \dots, \mathsf{k}_i(\mathbf{w}_i^{(\kappa)}, \mathbf{w}_i^*))^\top$, and $\mathsf{K}_i = [\mathsf{k}_i^{ab}] \in \mathbb{R}^{\kappa \times \kappa}$, $\mathsf{k}_i^{ab} = \mathsf{k}_i(\mathbf{w}_i^{(a)}, \mathbf{w}_i^{(b)})$.

A GP model for the unknown functions $h_i(\cdot)$ is then given by $\hat{h}_{i,\kappa}(\cdot) \coloneqq \mu_{i,\kappa}(\cdot)$. The following result provides a bound for the corresponding model error.

Lemma 1: Let Assumption 4 hold. Let $\hat{h}_{i,\kappa}(\cdot)$ denote the learned GP models of the unknown functions $h_i(\cdot)$ using the data sets $\mathbb{D}_{i,\kappa}$. Choose $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$, the following holds

$$|h_i(\mathbf{w}_i) - \hat{h}_{i,\kappa}(\mathbf{w}_i)| \le \beta_{i,\kappa} \sigma_{i,\kappa}(\mathbf{w}_i), \ \forall\, \mathbf{w}_i \in \mathbb{W}_i, \ \kappa \in \mathbb{N}, \tag{5}$$

where $\sigma_{i,\kappa}(\mathbf{w}_i)$ is the posterior standard deviation of the GP in (4b), $\beta_{i,\kappa} = B_{h_i} + \sigma_\varepsilon \sqrt{2\left(\gamma_{i,\kappa} + 1 + \ln\left(n\delta^{-1}\right)\right)}$, and $\gamma_{i,\kappa} = \max_{\bar{\mathbf{w}}_i^{(1)}, \dots, \bar{\mathbf{w}}_i^{(\kappa)} \in \mathbb{W}_i} \frac{1}{2} \log |I_\kappa + \sigma_\varepsilon^{-2} \bar{\mathsf{K}}_i|$, where $\bar{\mathsf{K}}_i = [\mathsf{k}_i(\bar{\mathbf{w}}_i^{(a)}, \bar{\mathbf{w}}_i^{(b)})]$.

A proof of Lemma 1 can be given by following the lines of the proof in [2, Lemma 1].

Remark 1: In the case of unknown lengthscales and signal variance for the kernel (3), a result similar to Lemma 1 holds if the lengthscales are underestimated and $\beta_{i,\kappa}$ is scaled appropriately [17]. Furthermore, in practice, in the case of a piecewise continuous function, it is reasonable to expect that Lemma 1 holds for portions of the state space where the function is continuous.

IV. Event-triggered online learning-based backstepping using GPs

In this section, we develop an efficient event-triggered online learning scheme that collects data (2) to update the estimated GP models and yields a backstepping control law such that the system state x_1 accurately tracks the desired trajectory x_d. We then discuss how to obtain a trade-off between tracking performance and maximal data set cardinality.

A. Event-triggered online learning

In the sequel, unless stated otherwise, we omit the arguments of the system dynamics as $f_i \coloneqq f_i(\mathbf{w}_i)$, $g_i \coloneqq g_i(\mathbf{w}_i)$ and $h_i \coloneqq h_i(\mathbf{w}_i)$. We then introduce the virtual control signals and control input

$$\begin{aligned} \alpha_1 &= g_1^{-1}\left(-k_1 e_1 + \dot{x}_d - f_1 - \hat{h}_{1,\kappa}\right), \ i = 2, \dots, n, \\ \alpha_i &= g_i^{-1}\left(-k_i e_i + \dot{\alpha}_{i-1} - g_{i-1} e_{i-1} - f_i - \hat{h}_{i,\kappa}\right), \end{aligned} \tag{6}$$

$$u = \alpha_n, \qquad \text{for } t \in [t_\kappa, t_{\kappa+1}), \tag{7}$$

Algorithm 1 Backstepping with event-triggered learning

initialize $\kappa = 0$, $\mathbb{D}_{i,\kappa} = \{\ \}$, $\hat{h}_{i,\kappa} = 0$, for $i = 1, 2, \dots, n$
while simulation time not exceeded **do**
  while $t < t_{\kappa+1}$ **do**
    run controller u in (7)
  end while
  set $\kappa \leftarrow \kappa + 1$
  measure $\mathbf{w}_i^{(\kappa)} = \mathbf{w}_i(t_\kappa)$ and $y_i^{(\kappa)} = h_i(\mathbf{w}_i^{(\kappa)}) + \varepsilon_i^{(\kappa)}$
  update data sets $\mathbb{D}_{i,\kappa} = \mathbb{D}_{i,\kappa-1} \cup \{(\mathbf{w}_i^{(\kappa)}, y_i^{(\kappa)})\}$
  update GP models $\hat{h}_{i,\kappa}$ in (6)
end while

where $e_1 = x_1 - x_d$ and $e_i = x_i - \alpha_{i-1}$ for $i = 2, \dots, n$ are the tracking errors, and $\hat{h}_{i,\kappa}$ is the estimate of h_i, learned using GPs based on the data (2) collected up to time t_κ, $\kappa \in \mathbb{N}$.

By substituting the control law (7) and the virtual signals (6) into (1), the tracking error dynamics become

$$\begin{aligned} \dot{e}_1 &= -k_1 e_1 + g_1 e_2 + (h_1 - \hat{h}_{1,\kappa}), \quad i = 2, \dots, n-1, \\ \dot{e}_i &= -k_i e_i + g_i e_{i+1} - g_{i-1} e_{i-1} + (h_i - \hat{h}_{i,\kappa}), \\ \dot{e}_n &= -k_n e_n - g_{n-1} e_{n-1} + (h_n - \hat{h}_{n,\kappa}), t \in [t_\kappa, t_{\kappa+1}). \end{aligned} \tag{8}$$

Denote $\mathbf{e} = (e_1, e_2, \dots, e_n)^\top \in \mathbb{R}^n$. The error dynamics (8) can then be written in compact form as

$$\dot{\mathbf{e}} = (-K + G)\mathbf{e} + \boldsymbol{\Delta}_{\mathbf{h}_\kappa}, \quad t \in [t_\kappa, t_{\kappa+1}), \tag{9}$$

where $K = \operatorname{diag}(k_1, k_2, \dots, k_n)$, $\boldsymbol{\Delta}_{\mathbf{h}_\kappa} = (\Delta_{h_{1,\kappa}} \ \Delta_{h_{2,\kappa}} \ \dots \ \Delta_{h_{n,\kappa}})^\top$ with $\Delta_{h_{i,\kappa}} = h_i - \hat{h}_{i,\kappa}$, and

$$G = \begin{pmatrix} 0 & g_1 & 0 & \dots & 0 \\ -g_1 & 0 & g_2 & \dots & 0 \\ 0 & -g_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & g_{n-1} \\ 0 & 0 & \dots & -g_{n-1} & 0 \end{pmatrix}. \tag{10}$$

According to Lemma 1, the variance function (4b) of a GP bounds the maximum error with high probability. It is therefore a suitable indicator for choosing when to add new data points. Based on this observation, we propose the following event for collecting a new data point and updating the GP models:

$$t_{\kappa+1} = \inf\left\{ t > t_\kappa \,\middle|\, \|\boldsymbol{\beta}_\kappa\| \, \|\boldsymbol{\sigma}_\kappa\| \ge \min_{i=1,2,\dots,n} k_i ||\mathbf{e}||, \ ||\mathbf{e}|| \notin \mathbb{B}_\kappa \right\}, \tag{11}$$

where $\boldsymbol{\beta}_\kappa = (\beta_{1,\kappa}, \beta_{2,\kappa}, \dots, \beta_{n,\kappa})^\top$ with $\beta_{i,\kappa}$ defined as in Lemma 1, $\boldsymbol{\sigma}_\kappa = (\sigma_{1,\kappa}, \sigma_{2,\kappa}, \dots, \sigma_{n,\kappa})^\top$ with $\sigma_{i,\kappa}$ the standard deviation posterior function using data up to time t_κ, and

$$\mathbb{B}_\kappa = \left\{ \mathbf{e} \in \mathbb{R}^n \,\middle|\, ||\mathbf{e}|| \le \frac{\|\boldsymbol{\beta}_\kappa\| \sqrt{n} \sigma_\varepsilon}{\min\limits_{i=1,2,\dots,n} k_i} \right\}. \tag{12}$$

The reasoning behind (11) and (12) is as follows. The inequality $\|\boldsymbol{\beta}_\kappa\| \, \|\boldsymbol{\sigma}_\kappa\| \ge \min_{i=1,2,\dots,n} k_i ||\mathbf{e}||$ indicates that a Lyapunov decrease condition is violated, which will become clear later on from the proof of Theorem 2. Furthermore, noise corrupts data to the point that a significant improvement in predictive performance cannot be guaranteed if we collect data within the ball (12). Hence, if the tracking error is outside the ball (12) and $\|\boldsymbol{\beta}_\kappa\| \, \|\boldsymbol{\sigma}_\kappa\| \ge \min_{i=1,2,\dots,n} k_i ||\mathbf{e}||$ holds, an event is triggered, and a new data point is added to update the GP models. Whenever the tracking error is inside the ball (12), there is no trigger.

The steps for updating the control law (7) are given in Algorithm 1. Using (11) as a trigger to update the GP models $\hat{h}_{i,\kappa}$ in the control law (7), we have the following result.

Theorem 2: Consider the system (1) and the desired trajectory x_d. Let Assumptions 1–5 hold. For $i = 1, 2, \ldots, n$, let $\hat{h}_{i,\kappa}$ be the learned GP models of the unknown functions h_i, based on the time-varying data sets (2). Choose $\delta \in (0, 1)$. Let the control gains be $k_i > 0$ and let $\hat{h}_{i,\kappa}$ be updated at time t_κ, where t_κ is chosen according to (11). Then, with probability at least $1 - \delta$, the resulting control law (7) guarantees that the tracking error is globally uniformly ultimately bounded to the ball (12). Furthermore, the inter-event time is lower-bounded by a positive constant.

Proof: Consider the Lyapunov function $V = \frac{1}{2}\sum_{i=1}^{n} e_i^2$. Since we aim to show that the error is globally ultimately bounded to the ball (12), we only need to show that the time derivative of V is negative outside (12). First, provided that the error is outside (12) and no trigger takes place, the time derivative of V with respect to time along (9) satisfies

$$\dot{V} = \frac{1}{2}(\dot{\mathbf{e}}^\top \mathbf{e} + \mathbf{e}^\top \dot{\mathbf{e}}) < \|\mathbf{e}\| \Big(- \min_{i=1,\ldots,n} k_i \|\mathbf{e}\| + \|\boldsymbol{\beta}_\kappa\| \|\boldsymbol{\sigma}_\kappa\| \Big) < 0,$$

for all $t \in (t_\kappa, t_{\kappa+1})$, where we obtain the first inequality by applying the Cauchy-Schwarz inequality to Lemma 1. Hence, the Lyapunov function is strictly decreasing between two consecutive events.

Next, note that the posterior variance at an arbitrary test point decreases every time a new training data point is added [11]. In addition, according to [1, Lemma 4], the posterior variance immediately after collecting the measurement at $\mathbf{w}_i(t_\kappa)$ is upper bounded by the posterior variance conditioned only on the measurement at $\mathbf{w}_i(t_\kappa)$. Using (4b), we then have

$$\sigma_{i,\kappa}^2 \leq \sigma_{h_i}^2 - \frac{\sigma_{h_i}^4}{\sigma_{h_i}^2 + \sigma_\varepsilon^2} = \frac{\sigma_\varepsilon^2}{1 + \sigma_\varepsilon^2/\sigma_{h_i}^2} < \sigma_\varepsilon^2, \tag{13}$$

where we employ $\mathsf{k}_i(\mathbf{w}_i(t_\kappa), \mathbf{w}_i(t_\kappa)) = \sigma_{h_i}^2$. It then follows from (13), Lemma 1 and the event (11) that $\|\boldsymbol{\Delta}_{\mathbf{h}_\kappa}\| < \|\boldsymbol{\beta}_\kappa\| \sqrt{n}\sigma_\varepsilon$. Using (12), it can be shown that, immediately after the data collection at the event time t_κ, since the error is still outside the ball (12), we have

$$\dot{V} < \|\mathbf{e}\| \Big(- \min_{i=1,2,\ldots,n} k_i \|\mathbf{e}\| + \|\boldsymbol{\beta}_\kappa\| \sqrt{n}\sigma_\varepsilon \Big) < 0.$$

Hence, the time derivative of V is strictly negative immediately after each data collection. Since the event guarantees that the time derivative of V remains negative, the tracking error $\mathbf{e}$ converges to $\mathbb{B}_\kappa$.

Next, we show that the inter-event time is lower-bounded by a positive constant. Since for $\mathbf{e} \in \mathbb{B}_\kappa$, no event is triggered, only the case $\mathbf{e} \notin \mathbb{B}_\kappa$ needs to be analyzed. To show that the inter-event time is lower-bounded, we first define the Lipschitz constant $L_{\boldsymbol{\sigma}_\kappa} > 0$ such that $\dot{\boldsymbol{\sigma}}_\kappa \leq L_{\boldsymbol{\sigma}_\kappa} \dot{\mathbf{e}}$, which exists due to the differentiability of $\boldsymbol{\sigma}_\kappa$ with respect to $\mathbf{e}$. Then, following the lines of [11], [18], we have

$$\begin{aligned}
\frac{d}{dt}\frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|} &= \frac{d}{dt}\frac{(\boldsymbol{\sigma}_\kappa^\top \boldsymbol{\sigma}_\kappa)^{\frac{1}{2}}}{(\mathbf{e}^\top \mathbf{e})^{\frac{1}{2}}} \\
&= \frac{(\boldsymbol{\sigma}_\kappa^\top \boldsymbol{\sigma}_\kappa)^{-\frac{1}{2}}(\boldsymbol{\sigma}_\kappa^\top \dot{\boldsymbol{\sigma}}_\kappa)(\mathbf{e}^\top \mathbf{e})^{\frac{1}{2}} - (\mathbf{e}^\top \mathbf{e})^{-\frac{1}{2}}(\mathbf{e}^\top \dot{\mathbf{e}})(\boldsymbol{\sigma}_\kappa^\top \boldsymbol{\sigma}_\kappa)^{\frac{1}{2}}}{\mathbf{e}^\top \mathbf{e}} \\
&\leq \frac{L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\sigma}_\kappa\| \|{-K\mathbf{e}} + \boldsymbol{\Delta}_{\mathbf{h}_\kappa}\|}{\|\boldsymbol{\sigma}_\kappa\| \|\mathbf{e}\|} + \frac{\|\mathbf{e}\| \|\boldsymbol{\sigma}_\kappa\| \|{-K\mathbf{e}} + \boldsymbol{\Delta}_{\mathbf{h}_\kappa}\|}{\|\mathbf{e}\|^3} \\
&\leq L_{\boldsymbol{\sigma}_\kappa} \|K\| + \|K\| \frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|} + L_{\boldsymbol{\sigma}_\kappa} \frac{\|\boldsymbol{\Delta}_{\mathbf{h}_\kappa}\|}{\|\mathbf{e}\|} + \frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|}\frac{\|\boldsymbol{\Delta}_{\mathbf{h}_\kappa}\|}{\|\mathbf{e}\|}.
\end{aligned}$$

Now, using (5) in Lemma 1 for $i = 1, 2, \ldots, n$ and Cauchy-Schwarz inequality, we have

$$\mathsf{P}\Big\{ \|\boldsymbol{\Delta}_{\mathbf{h}_\kappa}\| \leq \|\boldsymbol{\beta}_\kappa\| \|\boldsymbol{\sigma}_\kappa\|, \ \forall\, \mathbf{e} \in \mathbb{R}^n \Big\} \geq 1 - \delta.$$

Subsequently, we have

$$\begin{aligned}
\mathsf{P}\Big\{ \frac{d}{dt}\frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|} \leq{}& L_{\boldsymbol{\sigma}_\kappa} \|K\| + \|K\| \frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|} + L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\beta}_\kappa\| \frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|} \\
&+ \|\boldsymbol{\beta}_\kappa\| \left(\frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|}\right)^2, \ \forall\, \mathbf{e} \in \mathbb{R}^n \Big\} \geq 1 - \delta.
\end{aligned}$$

Denote $y = \frac{\|\boldsymbol{\sigma}_\kappa\|}{\|\mathbf{e}\|}$, with probability $1 - \delta$, we have

$$\dot{y} \leq \|\boldsymbol{\beta}_\kappa\| y^2 + (L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\beta}_\kappa\| + \|K\|)y + L_{\boldsymbol{\sigma}_\kappa} \|K\|,$$

and it holds that $y(t) \leq \phi(t, \phi(t_\kappa))$, where $\phi(t, \phi(t_\kappa))$ is the solution of the differential equation

$$\dot{\phi} = \|\boldsymbol{\beta}_\kappa\| \phi^2 + (L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\beta}_\kappa\| + \|K\|)\phi + L_{\boldsymbol{\sigma}_\kappa} \|K\| \tag{14}$$

with the initial condition $\phi(t_\kappa)$. Using (13), $\phi(t_\kappa)$ is upper bounded by $\phi(t_\kappa) < \frac{\sqrt{n}\sigma_\epsilon}{\|\mathbf{e}\|} := \phi_0$. By design, the event is triggered at $\phi(t_{\kappa+1}) = \frac{\min_{i=1,2,\ldots,n} k_i}{\|\boldsymbol{\beta}_\kappa\|}$. Now, since $\mathbf{e} \notin \mathbb{B}_\kappa$, it follows from (12) that $\phi(t_\kappa) < \phi_0 < \phi(t_{\kappa+1})$. Then the inter-event time is lower-bounded by the time it takes for ϕ from ϕ_0 to $\phi(t_{\kappa+1})$. According to WolframAlpha, the solution of (14) is given by

$$\phi(t) = \frac{c_2 \tan\left(\tan^{-1}\left(\frac{2c_1\phi_0 + c_3}{c_2}\right) + \frac{1}{2}(t - t_\kappa)c_2\right) - c_3}{2c_1},$$

where $c_1 = \|\boldsymbol{\beta}_\kappa\|$, $c_2 = \sqrt{4\|\boldsymbol{\beta}_\kappa\| \|K\| L_{\boldsymbol{\sigma}_\kappa} - c_3^2}$, and $c_3 = L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\beta}_\kappa\| + \|K\|$. The lower bound on the inter-event time is $\Delta_{\kappa+1} := t_{\kappa+1} - t_\kappa = \frac{1}{c_4}\ln\left(\frac{c_4 + c_5}{c_4 - c_5} \cdot \frac{c_4 - c_6}{c_4 + c_6}\right) > 0$, where $c_4 = \big|L_{\boldsymbol{\sigma}_\kappa} \|\boldsymbol{\beta}_\kappa\| - \|K\|\big|$, $c_5 = 2\|\boldsymbol{\beta}_\kappa\| \phi_0 + \|\boldsymbol{\beta}_\kappa\| L_{\boldsymbol{\sigma}_\kappa} + \|K\|$ and $c_6 = 2\min_{i=1,2,\ldots,n} k_i + \|\boldsymbol{\beta}_\kappa\| L_{\boldsymbol{\sigma}_\kappa} + \|K\|$. Therefore, the inter-event time is lower-bounded by a positive constant. ■

Remark 2: Note that, in (12), the tracking error bound can be made arbitrarily small by choosing high control gains $k_i > 0$, albeit at the cost of potentially very high control inputs. Note also that, given $k_i > 0$, the radius of (12) for event-triggered online learning is proportional to σ_ϵ. As a result, if σ_ϵ is small, then the proposed online learning scheme always guarantees a small tracking error bound. By contrast, given $k_i > 0$, the radius of the ultimate error bound for offline learning is proportional to the maximal posterior standard deviation, which is large when the state space is not sufficiently covered by the training data [2]. This is a major advantage of event-triggered online learning over offline learning.

B. Obtaining a trade-off between data cardinality and tracking performance

Although the results from Subsection IV-A guarantee ultimate boundedness of the tracking error with efficient data collection, the number of data points required to achieve the corresponding performance might still be prohibitive in some settings due to the poor scalability of GPs. To address this issue, in this section we discuss the trade-off between the cardinality of online collected training data and the size of the ultimate tracking error bound. To this end, we consider a modified version of the event (11) that requires only *an arbitrarily small* amount of online learning events and, subsequently, a small number of training data points, while potentially sacrificing tracking performance. More specifically, we consider again the event (11) where the corresponding ultimate ball is now given by

$$\mathbb{B}'_\kappa = \left\{ \mathbf{e} \in \mathbb{R}^n \;\middle|\; ||\mathbf{e}|| \leq \eta \frac{\|\boldsymbol{\beta}_\kappa\| \sqrt{n}\sigma_\varepsilon}{\min\limits_{i=1,2,\ldots,n} k_i} \right\}, \tag{15}$$

where $\eta \geq 1$ is a design parameter that provides a trade-off between tracking accuracy and amount of required data. Note that the ball (15) is potentially larger than (12), due to the design parameter η. The associated algorithm for updating the control law (7) is given in Algorithm 1.

Theorem 3: Consider the system (1) and the desired trajectory x_d. Let Assumptions 1–5 hold. For $i = 1, 2, \ldots, n$, let $\hat{h}_{i,\kappa}$ be the GP models of the unknown functions h_i, based on the time-varying data sets (2). Choose $\delta \in (0, 1)$. Let the control gains be $k_i > 0$ and let $\hat{h}_{i,\kappa}$ be updated at time t_κ, where t_κ is chosen according to (11). Then, with probability at least $1-\delta$, the resulting control law (7) guarantees that the tracking error is globally uniformly ultimately bounded to the ball (15) and that the inter-event time is lower-bounded by a positive constant. Moreover, for every $T \in \mathbb{N}$, there exists an $\eta \geq 1$, such that Algorithm 1 stops collecting data after T events and the tracking error is globally ultimately bounded to the ball (15) with $\kappa = T$ and η as a scaling factor.

Proof: It follows directly from Theorem 2 that, with probability at least $1-\delta$, the resulting control law (7) guarantees that the tracking error is globally uniformly ultimately bounded to the ball (15), and the inter-event time is lower-bounded by a positive constant.

Next, we show that, for every $T \in \mathbb{N}$, there exists an $\eta \geq 1$, such that Algorithm 1 stops collecting data after T events. We prove this by contradiction. Assume that the contrary is true, i.e., the number of events T goes to infinity for all $\eta \geq 1$. It then follows from the proofs of [16, Lemma 5.4] and [16, Theorem 5] that, for $T \geq 1$, there exists a positive constant $C > 0$, such that $\sum_{\kappa=1}^{T} \|\boldsymbol{\sigma}_T\|^2 \leq C(\log(T))^{n+1}$ holds. Hence, since the event is only triggered when the inequality in (11) holds with equality and $||\mathbf{e}|| \notin \mathbb{B}_\kappa$, and the number of events T goes to infinity, we have that, for any fixed $T \geq 1$ and the corresponding constant $C > 0$,

$$\|\boldsymbol{\beta}_T\| \, C(\log(T))^{n+1} \geq \sum_{\kappa=1}^{T} \|\boldsymbol{\beta}_\kappa\| \, \|\boldsymbol{\sigma}_\kappa\| = \sum_{\kappa=1}^{T} \left(\min_{i=1,\ldots,n} k_i ||\mathbf{e}(t_\kappa)|| \right)^2$$
$$\geq \sum_{\kappa=1}^{T} \left(\min_{i=1,\ldots,n} k_i \eta \frac{\|\boldsymbol{\beta}_\kappa\| \sqrt{n}\sigma_\varepsilon}{\min_{i=1,\ldots,n} k_i} \right)^2 \geq nT\eta^2\sigma_\varepsilon^2 \|\boldsymbol{\beta}_1\|^2 .$$

However, this does not hold for all $\eta \geq 1$, which is a contradiction. ■

Here, the role of $\eta \geq 1$ is primarily to limit the amount of collected data, as opposed to improving control performance. As a result, we are able to limit the total number of collected data points by increasing the radius of the ball $\mathbb{B}'_\kappa$. This is intuitive, since it implies that the size of the region where data has to be collected decreases, reducing the amount of information needed for model-learning.

Remark 3: Although specific formulas for choosing η as a function of T can be obtained, here we refrain from computing them, as they will typically be very conservative. In the simulation section, we present convincing evidence that a good trade-off between performance and data set size T is already obtained for small η.

TABLE I

LOWER DECILE/MEDIAN/UPPER DECILE OF L_2 ERROR OF ROBOTIC MANIPULATOR SIMULATION USING OUR APPROACH AND THE OFFLINE LEARNING-BASED APPROACH FROM [2].

	Lower decile	Median	Upper decile
Our approach	13.0	71.2	167.3
Offline learning [2]	86.8	155.8	370.1

TABLE II

AMOUNT OF COLLECTED DATA THROUGH EVENT-TRIGGERED LEARNING AT THE END OF SIMULATION.

	Lower decile	Median	Upper decile
Collected data	250	296	347

V. SIMULATION EXAMPLE

We now use a simulation example to illustrate our approach and compare it to the state-of-the art offline learning-based backstepping approach from [2]. The code for the simulation can be obtained from https://github.com/aCapone1/event_trig_backstepping. The simulated environment consists of a one-link planar manipulator with motor dynamics, as presented in [12]. The known and unknown components of the dynamics are the same as in [2], and the desired trajectory x_d consists of a sinusoidal signal. The GP hyperparameters were chosen by taking prior knowledge of the system into account. In practice, the values for $\boldsymbol{\beta}_\kappa$ in Lemma 1 can be conservative. Hence, we follow [16] by choosing $\|\boldsymbol{\beta}_\kappa\| = 2$, aiming also to show that a non-conservative estimate for β_κ is not of disadvantage in practice. When employing our approach, the GPs used for the control law start off with $N = 0$ data points and collect data according to the proposed event trigger (11) using (15) and $\eta = 1$. The offline learning-based approach employs $N = 400$ data points that were collected using a backstepping control law without training data. We run the

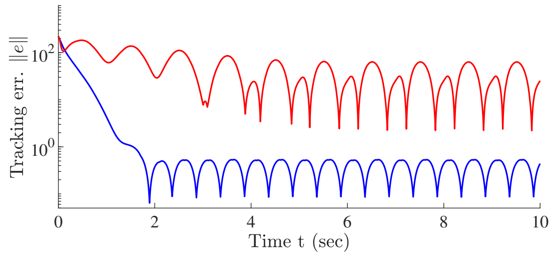

Fig. 1. Tracking error with our approach, in blue, and control law from [2], in red. Our control law collects a total of **319** data points within the first **3.5** seconds, which are informative enough to obtain an ultimate error bound of less than **1**, whereas the approach in [2] yields a very high error due to the lack of informative data.

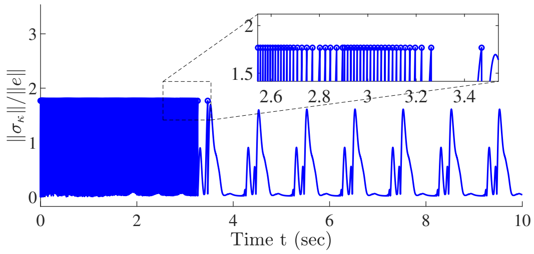

Fig. 2. GP variance divided by tracking error and event times. Events during which online learning takes place (blue circles). In total, **319** data points were gathered, all within the first four seconds of the simulation.

simulation 100 times using initial conditions drawn from a Gaussian distribution with mean zero and standard deviation of 10.

The median, lower and upper deciles of the L_2-norm of the tracking errors can be seen in Table I. For the number of training data required by our approach, the median, lower and upper deciles are listed in Table II. In more than 90% of the cases our approach required less data than the offline learning-based approach while yielding better performance, as seen in Table I. The plots of the tracking errors with the offline learning-based method from [2] and with event-triggered learning for a sample simulation can be seen in Fig. 1. The corresponding events in the event-triggered case are depicted in Fig. 2. As can be seen, our approach leads to a drastic improvement in control performance compared to the method from [2] while simultaneously requiring less data. This is because the data collected online using our approach always contributes to a significant reduction of the maximum model error, which in turn reduces the time-derivative of the Lyapunov function and improves control performance. By contrast, the data used to train the GPs for the offline learning-based approach was collected using a pre-defined control law, which does not guarantee that the collected data is useful for control.

In Fig. 2, it can be seen that model uncertainty in the event-triggered case is highest during the first 3 seconds of the simulation, which sets off the event trigger multiple times, leading to online model updates. Afterwards model uncertainty is fairly low and no additional learning takes place. This is to be expected due to Theorem 3. In total, 319 data points were collected during the simulation, i.e., our approach yields better performance than [2] while requiring less data.

VI. Conclusion

In this paper, we have studied the backstepping tracking control problem with event-triggered online learning for a class of partially unknown nonlinear systems. The unknown system models are learned using GPs based on data that is collected online, while the proposed control law is active. The estimated GP models and the control law are updated in an event-triggered fashion, where the event is triggered by a condition depending on the model uncertainty. We have shown that the proposed event-triggered learning scheme guarantees that the tracking error is globally uniformly ultimately bounded and the inter-event time is lower-bounded by a positive constant. In addition, we have discussed the trade-off between the cardinality of online collected data and the tracking performance.

References

[1] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using Gaussian processes," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5249–5255.

[2] A. Capone and S. Hirche, "Backstepping for partially unknown nonlinear systems using Gaussian processes," *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 416–421, 2019.

[3] 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.

[4] A. Capone, G. Noske, J. Umlauft, T. Beckers, A. Lederer, and S. Hirche, "Localized active learning of Gaussian process state space models," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 490–499.

[5] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18. MIT Press, 2006.

[6] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4661–4666.

[7] A. Jain, T. Nghiem, M. Morari, and R. Mangharam, "Learning and control using Gaussian processes," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pp. 140–149.

[8] 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, 2015.

[9] F. Solowjow, D. Baumann, J. Garcke, and S. Trimpe, "Event-triggered learning for resource-efficient networked control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 6506–6512.

[10] F. Solowjow and S. Trimpe, "Event-triggered learning," *Automatica*, vol. 117, p. 109009, 2020.

[11] 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, 2020.

[12] C. Kwan and F. Lewis, "Robust backstepping control of nonlinear systems using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 6, pp. 753–766, 2000.

[13] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, 1995.

[14] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong, "Command filtered backstepping," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1391–1395, 2009.

[15] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels." *Journal of Machine Learning Research*, vol. 7, no. 12, 2006.

[16] 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.

[17] F. Berkenkamp, A. P. Schoellig, and A. Krause, "No-regret Bayesian optimization with unknown hyperparameters," *Journal of Machine Learning Research*, vol. 20, pp. 1–24, 2019.

[18] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.