

Can Learning Deteriorate Control? Analyzing Computational Delays in Gaussian Process-Based Event-Triggered Online Learning

Xiaobing Dai*

Armin Lederer*

Zewen Yang

Sandra Hirche

XIAOBING.DAI@TUM.COM

ARMIN.LEDERER@TUM.COM

ZEWEN.YANG@TUM.COM

HIRCHE@TUM.COM

Chair of Information-oriented Control, Technical University of Munich, D-80333 Munich, Germany

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

When the dynamics of systems are unknown, supervised machine learning techniques are commonly employed to infer models from data. Gaussian process (GP) regression is a particularly popular learning method for this purpose due to the existence of prediction error bounds. Moreover, GP models can be efficiently updated online, such that event-triggered online learning strategies can be pursued to ensure specified tracking accuracies. However, existing trigger conditions must be able to be evaluated at arbitrary times, which cannot be achieved in practice due to non-negligible computation times. Therefore, we first derive a delay-aware tracking error bound, which reveals an accuracy-delay trade-off. Based on this result, we propose a novel event trigger for GP-based online learning with computational delays, which we show to offer advantages over offline trained GP models for sufficiently small computation times. Finally, we demonstrate the effectiveness of the proposed event trigger for online learning in simulations.

Keywords: Gaussian process regression, learning-based control, computational delay, event-triggered learning, online learning

1. Introduction

The control of systems with unknown or uncertain dynamics is a challenging problem arising in many applications such as underwater vehicles (Yan et al., 2019), electric machines (Yin and Zhao, 2022) and soft robotics (Gao et al., 2023). In order to overcome this challenge, supervised machine learning methods are frequently applied to learn models of the unknown components. In particular when safety or performance guarantees are required for control, Gaussian process (GP) regression (Williams and Rasmussen, 2006) is popular for model inference.

GP models have been employed together with a wide range of control techniques such as feedback linearization (Greeff and Schoellig, 2021), sliding mode control (Lima et al., 2020), control barrier function approaches (Jagtap et al., 2020), and model predictive control (Maiworm et al., 2021). By combining Lyapunov stability theory (Khalil, 2002) with GP prediction error bounds (Srinivas et al., 2012; Lederer et al., 2019), data-dependent safety and performance guarantees can often be derived. When no data is available before system operation, GP models of the dynamics can be efficiently learned online via iterative model updates (Nguyen-Tuong et al., 2009). While this can be done using time-triggered model updates (Meier et al., 2016), event-triggered learning (Solowjow et al., 2018) offers the beneficial property that it can ensure the satisfaction of prescribed accuracy

* Equal contribution.

guarantees, such that desired tracking and safety guarantees of the controller are ensured by design (Umlauf and Hirche, 2019; Castañeda et al., 2022). However, these guarantees crucially rely on the capability to evaluate the triggering conditions at arbitrary times, which is generally not possible in practice, in particular since GP model updates and predictions take non-negligible computation time (Lederer et al., 2021). Therefore, GP models introduce a computational delay in control loops.

While the control of delayed systems is a well-understood problem in classical control theory (Fridman, 2014), it has received little attention in the context of learning-based control. When linear regression is employed with nonlinear features, different forms of delays such as state (Ho et al., 2005) and input delays (Li et al., 2018) have been analyzed. However, these results do not extend beyond linear regression, such that they are not applicable to GP models, even though computational delays can be modeled as time-varying input delay. Delays in GP models have, to the best of our knowledge, only been considered in the context of sampled-data control, e.g., realized through a self-triggered update of control inputs (Dhiman et al., 2021). However, these approaches do not consider the problem of designing event-triggers for model updates. Moreover, they only take into account the effect of zero-order hold GP predictions on their safety guarantees, but ignore their delayed availability due to the required computation time. Therefore, modeling learning-based control as sampled-data system does not sufficiently address the challenges caused by computational delays.

In this work, we consider the problem of designing an event trigger for GP-based online learning with computational delays. For this purpose, we quantitatively analyze the effect of computational delays on tracking error bounds, which reveals a trade-off between prediction accuracy and computational delay. Based on this result, a trigger condition for online model updates is designed, such that a specified tracking error is guaranteed. Since online model updates generally increase the computation time, we show that event-triggered learning offers advantages when no sufficiently accurate GP model can be trained offline. Finally, numerical simulations demonstrate the practical importance of the derived theories and the effectiveness of the proposed online learning method.

The remainder of this paper is structured as follows. Section 2 defines the problem setting. The control performance with computational delays is analyzed in Section 3. In Section 4, the event-triggered model update strategy is proposed. Numerical simulations are provided to illustrate the proposed method in Section 5. Finally, Section 6 concludes this paper.

2. Problem Setting

In this paper, we consider plants described by m -th order dynamical systems in the controllable canonical form with unknown dynamics, i.e.,

$$\dot{\mathbf{q}}_1 = \mathbf{q}_2, \quad \dot{\mathbf{q}}_2 = \mathbf{q}_3, \quad \dots \quad \dot{\mathbf{q}}_m = \mathbf{f}(\mathbf{x}) + \mathbf{u}, \quad (1)$$

where $\mathbf{x} = [\mathbf{q}_1^T, \dots, \mathbf{q}_m^T]^T \in \mathbb{X} \subset \mathbb{R}^{mn}$ denotes the state composed of vectors $\mathbf{q}_i \in \mathbb{R}^n$, $i = 1, \dots, m$ with dimension $n \in \mathbb{N}_+$ and $\mathbf{u} \in \mathbb{R}^n$ is the control input. This structure is found in many practical systems, e.g., in Euler-Lagrange dynamics such as robotic manipulators, drones, and underwater vehicles. While the structure of these systems is often known, accurate knowledge of the function $\mathbf{f}(\cdot) = [f_1(\cdot), \dots, f_n(\cdot)]^T$, $f_i : \mathbb{X} \rightarrow \mathbb{R}$, $i = 1, \dots, n$ is usually not available, e.g., due to a high system complexity or unknown embedded environmental uncertainties. Therefore, we assume $\mathbf{f}(\cdot)$ to be unknown, but pose the following assumption on it.

Assumption 1 *The unknown function $\mathbf{f} : \mathbb{X} \rightarrow \mathbb{R}^n$ is Lipschitz continuous on the compact domain \mathbb{X} with Lipschitz constant L_f induced by the Euclidean norm, i.e. $\|\nabla \mathbf{f}(\mathbf{x})\| \leq L_f$.*

This assumption or even stronger versions requiring global differentiability is commonly found since they ensure the existence of unique solutions of the differential equation (1) (Khalil, 2002). The required local Lipschitz continuity is a very natural property of many physical systems which usually evolve smoothly over time. Therefore, Assumption 1 is not restrictive in practice.

The goal is to track a continuously differentiable and bounded reference trajectory $\mathbf{x}_d: \mathbb{R}_{0,+} \rightarrow \mathbb{X}$ with the state \mathbf{x} . In order to ensure the possibility of exact tracking, we assume reference trajectories of the form $\mathbf{x}_d = [\mathbf{q}_{d,1}^T, \dots, \mathbf{q}_{d,m}^T]^T$, $\mathbf{q}_{d,i} \in \mathbb{R}^n$, $i = 1, \dots, m$ defined by $\dot{\mathbf{q}}_{d,1} = \mathbf{q}_{d,2}, \dots, \dot{\mathbf{q}}_{d,m} = \mathbf{q}_{d,m+1}$. For achieving a high tracking accuracy, we employ a control law of the form

$$\mathbf{u}(t) = \dot{\mathbf{q}}_{d,m}(t) - \hat{\mathbf{f}}(t) + \sum_{i=1}^m \mathbf{\Lambda}_i \mathbf{e}_i(t), \quad (2)$$

where $\mathbf{e}_i(t) = \mathbf{q}_i(t) - \mathbf{q}_{d,i}(t) \in \mathbb{R}^n$, $i = 1, 2, \dots, m$ are tracking errors and $\mathbf{\Lambda}_i \in \mathbb{R}^{n \times n}$, $i = 1, \dots, m$ are control gain matrices. The function $\hat{\mathbf{f}}(\cdot): \mathbb{R}_{0,+} \rightarrow \mathbb{R}^n$ is a compensation of the unknown function $\mathbf{f}(\mathbf{x}(\cdot))$. This compensation can be realized by inferring a model $\boldsymbol{\mu}(\cdot): \mathbb{X} \rightarrow \mathbb{R}^n$ of $\mathbf{f}(\cdot)$, which requires the capability to generate training data as formalized in the following.

Assumption 2 *Noiseless measurements of the state $\mathbf{x}^{(\iota)} = \mathbf{x}(t_\iota)$ and noisy measurements of the highest derivative $\mathbf{y}^{(\iota)} = \dot{\mathbf{q}}_m(t_\iota) + \mathbf{w}^{(\iota)}$ can be taken at arbitrary time instances t_ι with $\iota \in \mathbb{N}_0$. The observation noise $\mathbf{w}^{(\iota)} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\sigma_{o,1}^2, \dots, \sigma_{o,n}^2))$ with $\sigma_{o,i} \in \mathbb{R}_+$, $i = 1, \dots, n$ is assumed Gaussian, independent and identically distributed.*

This assumption requires the exact measurement of the system state, which is a common requirement for the design of nonlinear control laws (Khalil, 2002). Moreover, it admits Gaussian perturbed observations of the derivative $\dot{\mathbf{q}}_m$, which for practical reasons often has to be approximated, e.g., using finite differences. Therefore, this assumption is commonly found when dealing with unknown dynamics (Koller et al., 2018; Greeff and Schoellig, 2021). Note that relaxations to other distributions exist (Chowdhury and Gopalan, 2017), but are out of the scope of this paper.

Moreover, Assumption 2 allows the generation of data during active control, such that a model $\boldsymbol{\mu}(\cdot)$ can be learned on-line. When real-world algorithms are used for this purpose, they need a considerable amount of time to update the model $\boldsymbol{\mu}_\iota(\cdot)$ using new data $(\mathbf{x}^{(\iota+1)}, \mathbf{y}^{(\iota+1)})$ ¹. Moreover, the evaluation of such models at a state \mathbf{x} often takes a non-negligible amount of time. Therefore, we cannot employ $\boldsymbol{\mu}(\mathbf{x}(t))$ as compensation $\hat{\mathbf{f}}(t)$ in practice, but need to work with the delayed value $\hat{\mathbf{f}}(t) = \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))$, where

$$t_{k+1} = t_k + \Delta(t_k), \quad \kappa(t) = \arg \max_{k \in \mathbb{N}} t_{k+1} < t \quad (3)$$

with the discretized time $t_k, \forall k \in \mathbb{N}, t_0 = 0$, and $\Delta: \mathbb{R}_{0,+} \rightarrow \mathbb{R}_+$ denotes the overall computation time which includes the time for determining a prediction in the off-line learning scenario in Section 3 and the time it takes to compute a model update and a prediction in the online learning problem in Section 4. Note that the number of training samples ι at time t does generally not correspond to the number of previous model evaluations $\kappa(t)$. In order to be able to determine the worst-case impact of the computation time $\Delta(\cdot)$ on the tracking accuracy, we require the following assumption.

Assumption 3 *The overall computation time $\Delta(t)$ is bounded by a finite constant $\bar{\Delta} \in \mathbb{R}_+$ for all $t \in \mathbb{R}_{0,+}$ and its induced delay only affects the compensation $\hat{\mathbf{f}}(t)$.*

1. We use an index ι whenever necessary to emphasize the number of previously generated training samples.

Since a bounded computation time merely requires the termination of a learning algorithm after a finite number of computations, this assumption is satisfied by most practically employed methods. Moreover, the limitation of the induced delay to the compensation $\hat{f}(t)$ can be easily realized in practice by implementing the learned model in a parallel, non-blocking, process. Therefore, Assumption 3 does not pose a severe restriction.

Based on these assumptions, the controller (2) yields the error dynamics

$$\dot{e}(t) = \mathbf{A}e(t) + \mathbf{B}(\mathbf{f}(\mathbf{x}(t)) - \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))), \quad (4)$$

where $e = \mathbf{x} - \mathbf{x}_d$, the matrices $\mathbf{A} \in \mathbb{R}^{mn \times mn}$ and $\mathbf{B} \in \mathbb{R}^{mn \times n}$ are defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{(m-1)n \times n} & \mathbf{I}_{(m-1)n} \\ \boldsymbol{\Lambda}_1 & [\boldsymbol{\Lambda}_2, \dots, \boldsymbol{\Lambda}_m] \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{(m-1)n \times n} \\ \mathbf{I}_n \end{bmatrix}. \quad (5)$$

In this work, we consider the problem of ensuring an upper bound $\bar{e} \in \mathbb{R}_+$ for the tracking error $e = \mathbf{x} - \mathbf{x}_d$ with high probability using a control law of the form (2) by designing an event-triggered data selection strategy of the form

$$t_{\iota+1} = \min_{t_k > t_\iota} t_k, \quad \text{such that } \|\boldsymbol{\mu}(\mathbf{x}(t_k)) - \mathbf{f}(\mathbf{x}(t_k))\| \geq v^*(t_k, \bar{e}, \bar{\Delta}) \quad (6)$$

where $v^*(\cdot) : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ denotes the trigger threshold function.

3. Tracking Error Bounds with Delayed Gaussian Process Predictions

Since the derivation of tracking error guarantees requires model error bounds, we consider Gaussian process regression as a machine learning method in this paper. The foundations of GP regression are outlined together with a prediction error bound in Section 3.1. Based on this bound, tracking accuracy guarantees for control with delayed model predictions are derived in Section 3.2.

3.1. Gaussian Process Regression

Gaussian process regression is a modern machine learning technique, which can be used for the supervised inference of unknown functions. For simplicity, we assume for the moment $n = 1$, such that the unknown function $f(\cdot)$ in (1) is scalar. Then, a Gaussian process induces a distribution over $f(\cdot)$, denoted as $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$, which is completely specified by the prior mean $m : \mathbb{X} \rightarrow \mathbb{R}$ and the kernel $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_{0,+}$. The prior mean $m(\cdot)$ can be used to incorporate parametric models into the regression. Since they are often not available, a zero prior $m(\mathbf{x}) = 0$, $\forall \mathbf{x} \in \mathbb{X}$ is frequently used, which we also assume in the following without loss of generality. The kernel $k(\cdot, \cdot)$ reflects the prior covariance between evaluations of $f(\cdot)$ at different states \mathbf{x} and can encode information such as periodicity or symmetry of $f(\cdot)$.

When $N \in \mathbb{N}$ measurements $\mathbf{x}^{(\iota)}, y^{(\iota)}$, $\iota = 1, \dots, N$ satisfying Assumption 2 are available, this GP prior can be employed for regression using Bayesian principles. This merely requires the computation of the posterior distribution by conditioning the prior on the training data, which yields a Gaussian distribution with mean and variance

$$\boldsymbol{\mu}(\mathbf{x}) = \mathbf{k}_X^T (\mathbf{K} + \sigma_o^2 \mathbf{I})^{-1} \mathbf{y}, \quad \sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_X^T (\mathbf{K} + \sigma_o^2 \mathbf{I})^{-1} \mathbf{k}_X, \quad (7)$$

at every test point $\mathbf{x} \in \mathbb{X}$. For notational simplicity, the training targets $y^{(\iota)}$ are concatenated into a vector $\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]^T$ and the kernel gram matrix and vector are defined as $\mathbf{K} = [k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})]_{i,j=1,\dots,N}$ and $\mathbf{k}_X = [k(\mathbf{x}^{(1)}, \mathbf{x}), \dots, k(\mathbf{x}^{(N)}, \mathbf{x})]^T$, respectively. The posterior mean $\boldsymbol{\mu}(\cdot)$ can be used as a model of the unknown function $f(\cdot)$, while the variance $\sigma^2(\cdot)$ serves as a

measure of epistemic uncertainty. In order to ensure that the epistemic uncertainty properly reflects the potential of model errors, we require the following assumption.

Assumption 4 *The unknown function $f(\cdot)$ is a sample of a zero-mean prior GP with stationary, Lipschitz continuous and monotonically decreasing kernel $k(\cdot, \cdot)$.*

A prior GP satisfying Assumption 4 is flexible enough to represent the unknown function with suitable hyperparameters. Ensuring appropriate kernel structures can often be achieved in practice, e.g., by choosing universal kernels such as the squared exponential covariance function, which allows us to approximate continuous functions arbitrarily well (Michelli et al., 2006). The hyperparameters of such kernels can be obtained in practice via suitable hyperparameter tuning methods, e.g., (Capone et al., 2022). Therefore, Assumption 4 is not restrictive in practice.

Based on this assumption, the following prediction error bound for GP regression can be shown.

Lemma 1 (Lederer et al. (2019)) *Consider an unknown function $f(\cdot)$ satisfying Assumptions 1 and 4. Moreover, assume that $N \in \mathbb{N}$ measurements $\mathbf{x}^{(i)}, y^{(i)}$ are available and satisfy Assumption 2. Then, for every $\delta \in (0, 1) \subset \mathbb{R}$ and grid factor $\tau \in \mathbb{R}_+$, the prediction error of GP regression satisfies*

$$\mathbb{P}(|f(\mathbf{x}) - \mu(\mathbf{x})| \leq \eta_\delta(\mathbf{x}), \forall \mathbf{x} \in \mathbb{X}) \geq 1 - \delta, \quad \eta_\delta(\mathbf{x}) = \sqrt{\beta_\delta} \sigma(\mathbf{x}) + \gamma_\delta, \quad (8)$$

on a compact domain $\mathbb{X} \subset \mathbb{R}^{mn}$, where

$$\beta_\delta = 2 \sum_{j=1}^{nm} \log \left(\frac{\sqrt{nm}}{2\tau} (\bar{x}_j - \underline{x}_j) + 1 \right) - 2 \log \delta, \quad \gamma_\delta = \left(\sqrt{\beta_\delta} L_\sigma + L_f + L_\mu \right) \tau, \quad (9)$$

with $\bar{x}_j = \max_{\mathbf{x} \in \mathbb{X}} x_j$, $\underline{x}_j = \min_{\mathbf{x} \in \mathbb{X}} x_j$, x_j is the j -th dimension of \mathbf{x} , and L_σ and L_μ denote the Lipschitz constants of standard deviation $\sigma(\cdot)$ and the mean $\mu(\cdot)$.

This lemma allows us to bound the error of a learned GP model uniformly on a compact domain \mathbb{X} . The Lipschitz constants L_σ and L_μ required for the error bound $\eta(\cdot)$ can be straightforwardly determined as shown in (Lederer et al., 2021). Therefore, Lemma 1 provides a practically usable, state and data dependent error bound for models learned using GP regression.

3.2. Performance Guarantees for Learning-Based Control with Delayed Predictions

In order to apply GP regression for learning a model of the unknown function $\mathbf{f}(\cdot) = [f_1(\cdot) \cdots f_n(\cdot)]^T$ in (1) for the general case $n \geq 1$, we perform GP regression for each component $f_i(\cdot), i = 1, \dots, n$, independently. Afterwards, we concatenate the result in vectors $\boldsymbol{\mu}(\cdot) = [\mu_1(\cdot) \cdots \mu_n(\cdot)]^T$ and $\boldsymbol{\sigma}(\cdot) = [\sigma_1(\cdot) \cdots \sigma_n(\cdot)]^T$, where $\mu_i(\cdot)$ and $\sigma_i(\cdot)$ denote the mean and standard deviation of the GP trained using measurements of $f_i(\cdot)$. This allows us to directly employ the concatenated mean $\boldsymbol{\mu}(\cdot)$ in the control law (2) by setting $\hat{\mathbf{f}}(t) = \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))$, where $t_{\kappa(t)}$ denotes the last model evaluation time defined through (3). Since this allows us to bound the model error using Lemma 1, we employ Lyapunov stability theory to analyze the tracking performance. For this purpose, we use a common quadratic Lyapunov function $V(\mathbf{e}(t)) = \mathbf{e}^T(t) \mathbf{P} \mathbf{e}(t)$ with a symmetric, positive definite matrix $\mathbf{P} \in \mathbb{R}^{mn \times mn}$, which is the solution of the continuous algebraic Riccati equation $\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}$ for a given positive definite matrix $\mathbf{Q} \in \mathbb{R}^{mn \times mn}$. This common Lyapunov function allows us to derive a tracking error bound as shown in the following theorem².

2. Proofs for all theoretical results can be found at <https://mediatum.ub.tum.de/doc/1693032>.

Theorem 2 Consider a system (1), where $\mathbf{f}(\cdot)$ is an unknown nonlinearity consisting of scalar functions $f_i(\cdot)$, $i = 1, \dots, n$, which satisfy Assumptions 1 and 4. Assume that n GPs are trained with $N \in \mathbb{N}$ measurements $\mathbf{x}^{(t)}$, $\mathbf{y}^{(t)}$ satisfying Assumption 2, which results in the concatenated mean $\boldsymbol{\mu}(\cdot)$. Assume that the computation time $\Delta(\cdot)$ of GP predictions satisfies Assumption 3, such that we can employ the control law (2) with $\hat{\mathbf{f}}(t) = \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))$ for $t_{\kappa(t)}$ defined in (3) and control gains $\boldsymbol{\Lambda}_i$ inducing a Hurwitz matrix \mathbf{A} in (5). If the computation time bound $\bar{\Delta}$ satisfies $\bar{\Delta} < 1/(2L_f)$ and $\|\mathbf{e}(0)\| = 0$, then, the tracking error is bounded by

$$\|\mathbf{x}(t) - \mathbf{x}_d(t)\| \leq \bar{e} = \chi \xi (2L_f F \bar{\Delta} + \bar{\eta}_\delta), \quad \forall t \in \mathbb{R}_{0,+}, \quad (10)$$

with probability of at least $2(1 - \delta)^n - 1$, where $\xi = 2\|\mathbf{P}\|\|\mathbf{Q}^{-1}\|$, $\chi^2 = \|\mathbf{P}^{-1}\|\|\mathbf{P}\|$, $\chi > 0$, $\bar{\eta}_\delta = \sup_{\mathbf{x} \in \mathbb{X}} \|\boldsymbol{\eta}_\delta\|$, $\boldsymbol{\eta}_\delta(\mathbf{x}) = [\eta_{\delta,1}(\mathbf{x}) \cdots \eta_{\delta,n}(\mathbf{x})]^T$ and

$$F = \frac{1}{1 - 2L_f \bar{\Delta}} \left(\|\mathbf{A}\| \sup_{\mathbf{x} \in \mathbb{X}} \|\mathbf{x}\| + \left\| \begin{bmatrix} \boldsymbol{\Lambda}_1 & \cdots & \boldsymbol{\Lambda}_m \end{bmatrix} \right\| \sup_{t \in \mathbb{R}_{0,+}} \|\mathbf{x}_d(t)\| + \sup_{t \in \mathbb{R}_{0,+}} \|\dot{\mathbf{q}}_{d,m}(t)\| + \bar{\eta}_\delta \right). \quad (11)$$

Theorem 2 provides an intuitive insight into the sources of the tracking error. On the one hand, the accuracy of the GP model directly influences the tracking error as indicated by the model error bound $\bar{\eta}_\delta$, which can be reduced using more training data. On the other hand, lower computation time measured by $\bar{\Delta}$ can also decrease the tracking error bound \bar{e} . Since the computation time of GP predictions for $\boldsymbol{\mu}(\cdot)$ depends linearly ($\mathcal{O}(N)$) on the number of training samples N , these two values are inherently coupled, such that a suitable trade-off for the training set size N must be found.

4. Event-Triggered Online Learning under Computation Delays

Since the tracking error bound (10) increases with the computational delay and thus with the data set size in practice, data-efficiency is of crucial importance to ensure a high accuracy. Therefore, training samples should be carefully selected, which can be efficiently performed online by updating the model at the times when the model uncertainty is too high to guarantee the desired tracking error bound. This approach leads to the event-triggered online learning strategy derived in Section 4.1. Since such an online data generation process leads to potentially infinitely growing data sets, we analyze the effect of data deletion strategies on the derived guarantees in Section 4.2. In Section 4.3, we finally investigate the effect of the practically increased computation time due to online model updates to derive conditions when offline learning provides better tracking accuracy guarantees.

4.1. Event-triggered Model Update Strategy

While the training data cannot directly be determined using (6) due to the unknown dynamics $\mathbf{f}(\cdot)$, we can use the GP prediction error bound (8) as a proxy. This leads to the trigger condition

$$\|\boldsymbol{\eta}_\delta(\mathbf{x}(t_k))\| \geq v(t_k, \bar{e}, \bar{\Delta}), \quad (12)$$

which can be interpreted as a conservative, but implementable version of the right side of (6). Note that this trigger condition cannot be evaluated at arbitrary times $t \in \mathbb{R}_{0,+}$ as in previous approaches (Umlauf and Hirche, 2019; Jiao et al., 2022) due to the considered non-negligible time required for the computation of the GP variance $\sigma^2(\mathbf{x})$ and thus $\|\boldsymbol{\eta}_\delta(\mathbf{x}(t_k))\|$. Instead, all computations are started at discrete times t_k , such that the trigger condition has only access to $\mathbf{x}(t_k)$, and the new evaluation of the GP mean $\boldsymbol{\mu}(\cdot)$ is available earliest at $t_k + \Delta(t_k)$. These restrictions need to be anticipated in the design of the event trigger as shown in the following theorem.

Theorem 3 Consider a system (1), where $\mathbf{f}(\cdot)$ is an unknown nonlinearity consisting functions $f_i(\cdot), i = 1, \dots, n$ which satisfies Assumptions 1 and 4. Assume that measurements $\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$ satisfying Assumption 2 are used in each of the n GPs, which results the concatenated mean $\boldsymbol{\mu}(\cdot)$ and standard deviation $\boldsymbol{\sigma}(\cdot)$. Assume that the computation time $\Delta(\cdot)$ required for a prediction and model update satisfies Assumption 3 and is bounded by $\bar{\Delta} < 1/(2L_f)$, such that the control law (2) is employed with $\hat{\mathbf{f}}(t) = \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))$ for $t_{\kappa(t)}$ defined in (3) and control gains $\boldsymbol{\Lambda}_i$ inducing a Hurwitz matrix \mathbf{A} in (5). Pick $\bar{e} \geq 2\chi(F + F_d + \xi L_f F)\bar{\Delta} + \chi\xi\eta_\delta$ with $\eta_\delta = \|\boldsymbol{\eta}_{\delta,1}, \dots, \boldsymbol{\eta}_{\delta,n}\|^T$, $\eta_{\delta,i} = \sqrt{\beta_\delta}\sigma_{o,i} + \gamma_{\delta,i}$, and execute model updates at times t_i defined by the event-trigger (12) with threshold

$$v(t_k, \bar{e}, \bar{\Delta}) = \xi^{-1} \max(\|e(t_k)\|, \chi^{-1}\bar{e}) - 2(\xi^{-1}(F + F_d) + L_f F)\bar{\Delta}, \quad (13)$$

where $F_d \geq \|\dot{\mathbf{x}}_d(t)\|, \forall t \in \mathbb{R}_{0,+}$. Then, the tracking error is bounded by \bar{e} with probability at least $2(1 - \delta)^n - 1$, if the system initializes with $\|e(0)\| \leq \bar{e}$.

Due to the delayed impact of model updates, the trigger threshold (13) cannot guarantee arbitrarily small tracking error bounds. Moreover, the time $\Delta(t_k)$ between two potential training samples prevents arbitrarily small prediction error bounds $\|\boldsymbol{\eta}_\delta(\mathbf{x})\|$, which also influences the achievable tracking error bound. Despite these restrictions, the trigger threshold (13) exhibits the intuitive behavior that smaller desired error bounds yield more frequent model updates as indicated by a smaller value $v(t_k, \bar{e}, \bar{\Delta})$. Moreover, a higher computation time bound $\bar{\Delta}$ has a similar effect as the trigger anticipates a potential future increase in the model and tracking error. Therefore, Theorem 3 provides an effective method for ensuring a desired tracking error bound \bar{e} in a practically relevant setting.

4.2. Tracking Error Guarantees with Data Deletion

When data is generated online using the event-trigger (13), the data set size N can possibly grow infinitely, which would imply an unbounded computation time $\Delta(\cdot)$. Since this must be avoided in practice, a common strategy for online learning with GPs is the deletion of previous samples, such that the training set size N and the computation time $\Delta(\cdot)$ remain bounded. This can be realized, e.g., by deleting the oldest data points (Meier and Schaal, 2016) or through information-theoretic criteria (Han and Yi, 2021). Similarly as in (Umlauft and Hirche, 2019), our guarantees for event-triggered online learning are not affected by such deletion strategies.

Corollary 4 Under the assumptions of Theorem 3, if model updates are executed at times t_i defined through the event trigger (12) with threshold (13) after an arbitrary deletion strategy has been used for removing a data sample from the training set, then, the tracking error is bounded by \bar{e} for all $\|e(0)\| \leq \bar{e}$ with probability at least $2(1 - \delta)^n - 1$.

This corollary ensures that we can ensure Assumption 3 in practice by limiting the admissible data set size. While it allows the application of arbitrary deletion strategies without changes to the tracking error bound, the specific choice can still have a significant impact on the behavior of the control system. For example, poor deletion strategies, such as the deletion of all existing data, can significantly increase the number of triggered model updates. If the induced switching is excessive and the discrete steps in the control law become too large, real-world actuator can fail to follow the control signal with negative effects on the tracking accuracy. Therefore, a suitable method such as the established approach of deleting the oldest data point (Meier and Schaal, 2016) should be employed. Note that the time for data deletion is directly considered in the overall computation time $\Delta(\cdot)$.

4.3. Comparison between Offline Training and Online Model Updates

While we abstractly assume the existence of computation time bounds $\bar{\Delta}$ both for offline training in Theorem 2 and for online updates in Theorem 3, the actual values of this bound significantly differ between these two scenarios in practice. The reason for this is the requirement of the additional evaluation of the GP variance $\sigma^2(\cdot)$ and the model update for event-triggered learning, which cause a quadratic complexity $\mathcal{O}(N^2)$. This leads to generally higher computation times $\Delta(\cdot)$ for online learning in comparison to offline training, for which only mean evaluations ($\mathcal{O}(N)$) have to be computed online. Therefore, online learning cannot always be ensured to provide a benefit.

Corollary 5 *Consider a system (1), where $\mathbf{f}(\cdot)$ is an unknown nonlinearity consisting of scalar functions $f_i(\cdot)$, $i = 1, \dots, n$, which satisfy Assumptions 1 and 4. Assume that the following two GP models can be employed in the control law (2) with $\hat{\mathbf{f}}(t) = \boldsymbol{\mu}(\mathbf{x}(t_{\kappa(t)}))$ for $t_{\kappa(t)}$ defined in (3) and control gains Λ_i inducing a Hurwitz matrix \mathbf{A} in (5):*

1. *an offline GP model trained with $N \in \mathbb{N}$ arbitrary measurements $\mathbf{x}^{(l)}$, $\mathbf{y}^{(l)}$ satisfying Assumption 2, such that its computation time $\Delta(\cdot)$ for predictions is bounded by $\bar{\Delta}_1$ and its prediction error satisfies $\|\boldsymbol{\eta}_\delta(\mathbf{x})\| \leq \bar{\eta}_\delta, \forall \mathbf{x} \in \mathbb{X}$, which ensures a tracking error bound \bar{e}_1 ;*
2. *an online learning GP model updated with measurements $\mathbf{x}^{(l)}$, $\mathbf{y}^{(l)}$ satisfying Assumption 2 and defined through the event trigger (12) with threshold (13) and $\bar{e}_2 = 2\chi(F + F_d + \xi L_f F)\bar{\Delta} + \chi\xi\bar{\eta}_\delta$ in combination with an arbitrary data deletion strategy, such that the overall computation time $\Delta(\cdot)$ is bounded by $\bar{\Delta}_2$.*

Assume that it holds that $\bar{\Delta}_1 \leq \bar{\Delta}_2 < 1/(2L_f)$ and

$$\bar{\Delta}_2 \geq \frac{\xi\bar{\eta}_\delta}{2(F + F_d)} \quad \vee \quad \tilde{\Delta} \geq \frac{\xi\bar{\eta}_\delta - 2(F + F_d)\bar{\Delta}_1}{2(\xi L_f F + F + F_d)}, \quad (14)$$

where $\tilde{\Delta} = \bar{\Delta}_2 - \bar{\Delta}_1$ and $\tilde{\eta}_\delta = \bar{\eta}_\delta - \underline{\eta}_\delta$. Then, the guaranteed tracking error bound using the offline learning GP model 1 is lower than for the online learning GP model 2, i.e., $\bar{e}_1 \leq \bar{e}_2$.

Since \bar{e}_2 in Theorem 5 is the best guaranteeable tracking error bound using Theorem 3, this theorem implies that online learning does not offer a certifiable advantage over offline trained models if condition (14) holds. This condition crucially relies on the variable $\tilde{\eta}_\delta$, which measures the improvement in terms of the GP prediction error bound $\|\boldsymbol{\eta}_\delta\|$ due to event-triggered model updates. When the offline trained GP model is already highly accurate, event-triggered online learning can only provide a marginal improvement $\tilde{\eta}_\delta$. If the negative impact due to a computation time increase $\tilde{\Delta}$ exceeds this improvement as formalized on the right side of (14), online learning can potentially yield worse tracking errors than offline trained models. Therefore, Theorem 5 reflects the intuitive result that online learning is only meaningful if we do not have access to accurate offline models.

5. Numerical Simulation

To illustrate the effect of the computation delay on the control performance, we evaluate the proposed event-triggered model update strategy in a numerical simulation using the system dynamics $\dot{x}_1 = x_2$, $\dot{x}_2 = f(\mathbf{x}) + u = \sin(x_1) + 0.5(1 + \exp(x_2/10))^{-1} + u$, where $\mathbf{x} = [x_1, x_2]^T \in \mathbb{X} = [-1.5, 1.5] \times [-1.5, 1.5] \subset \mathbb{R}^2$. The desired trajectory is set to $x_{d,1}(t) = \sin(t)$, $x_{d,2}(t) = \cos(t)$ satisfying Assumption 3. The control gains Λ_i in (2) is chosen as $\Lambda_1 = \Lambda_2 = -2$, and the symmetric positive definite matrix is selected as $\mathbf{Q} = \mathbf{I}_2$. For learning, we use Gaussian processes

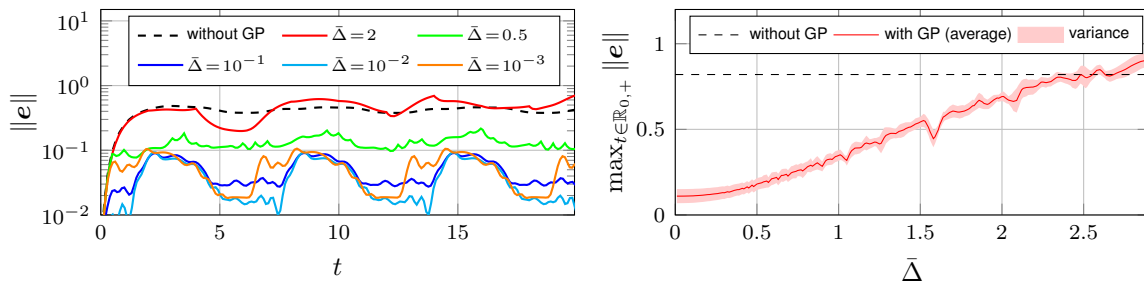


Figure 1: Tracking errors without online learning under different computational delays (left). Dependency of the maximum tracking error on the computational delay (right).

with squared exponential kernel $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-0.5l^{-2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}'))$, where $\sigma_f = 1$ and $l = 0.2$. All training samples $y^{(i)}$ are perturbed by Gaussian noise with standard deviation $\sigma_o = 0.01$. The simulation time is set to 20 and every simulation is repeated 100 times to account for the randomness in training data. Unless stated otherwise, all simulations are started with an initial data set containing $N_0 = 100$ samples evenly distributed in \mathbb{X} .

Based on these data sets, we first illustrate the effect of computation delays on the tracking error and investigate the necessary trade-off between model accuracy and computation time in Section 5.1. In Section 5.2, the proposed event-triggered model update strategy is demonstrated.

5.1. Influence of Delay on Tracking Errors

In order to investigate the effect of the computation delay $\Delta(t)$ on the control performance, we first consider GP models for fixed data sets, i.e., with only N_0 initial samples and without model updates. For simplicity, we assume constant computation times $\Delta(t) = \bar{\Delta}, \forall t \in \mathbb{R}_{0,+}$ which are independent of N_0 , such that the effect of available computational resources can be investigated. The resulting tracking errors for different delays $\bar{\Delta}$ are depicted on the left of Fig. 1. For large values such as $\bar{\Delta} = 2$ and $\bar{\Delta} = 0.5$, the performance is strongly affected by the computation delay and can even exceed that of a controller without any GP model, i.e., $\hat{\mathbf{f}}(t) = \mathbf{0}, \forall t \in \mathbb{R}_{0,+}$. In contrast, a decrease of the delay below $\bar{\Delta} = 10^{-1}$ has only a marginal effect on the tracking error. This effect is in line with Theorem 2, which guarantees a linear dependency of the tracking error bound \bar{e} on both the prediction error bound $\bar{\eta}_\delta$ and the delay $\bar{\Delta}$. When $\bar{\eta}_\delta$ is kept constant, this implies the observed vanishing impact of a delay reduction. As illustrated on the right side of Fig. 1, we can even see the linear growth of the maximum error for sufficiently large delays, which is guaranteed for the error bound \bar{e} in (10).

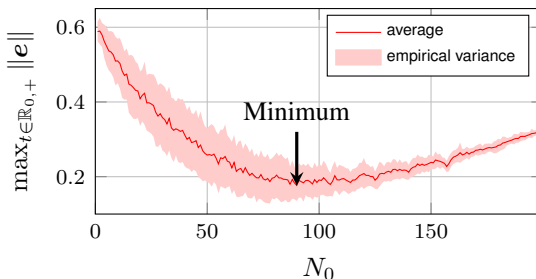


Figure 2: Accuracy-delay trade-off due to a joint dependence on the training set size N_0 .

In addition to this linear dependency of the error on the delay bound $\bar{\Delta}$, we can also directly observe the accuracy-delay trade-off discussed in Section 3. For this purpose, we employ the knowledge of the linear GP prediction complexity to model the computational delay as $\bar{\Delta} = cN_0$, where we exemplarily choose $c = 0.05$ and N_0 in the range between 0 and 200. The resulting dependency of the maximum tracking error on the number of training samples is depicted in Fig. 2. It

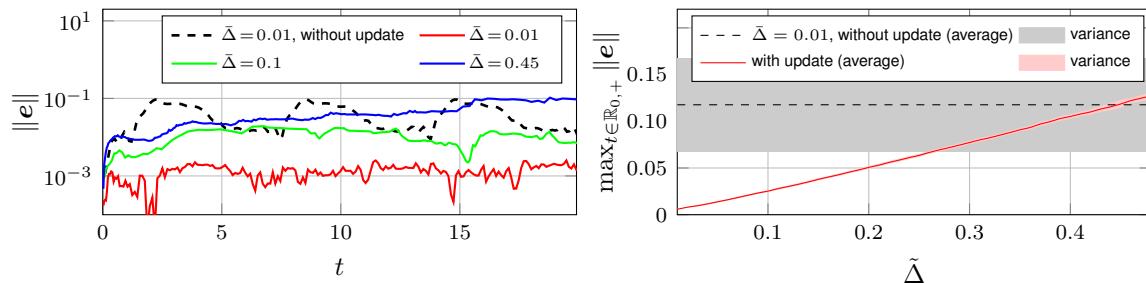


Figure 3: (Left) Tracking errors with event-triggered online learning under different computational delays. Overall, 595 of 2077, 197 of 900 and 115 of 118 new samples are online selected for model update with $\bar{\Delta} = 0.01$, 0.1 and 0.45, respectively. (Right) Dependency of the maximum tracking error on the additional delay $\bar{\Delta}$ caused by model updates in comparison to offline trained GPs with fixed computation time.

can be clearly seen that the error reduces at first due to the improved accuracy. After $N_0 \approx 90$ data points have been reached, the negative effect of the delay becomes dominant, such that the error starts to increase. This leads to a clearly visible minimum, at which the GP accuracy and the caused computational delay are optimally balanced.

5.2. Control Performance with Event-triggered Update

We demonstrate the effectiveness of the proposed online learning strategy by choosing the minimal, admissible error bound $\bar{e} = 2\chi(F + F_d + \xi L_f F)\bar{\Delta} + \chi\xi\eta_\delta$ in the trigger threshold (13). Due to the quadratic complexity of GP updates, we model the computation time as $\Delta(t) = cN^2(t)$, where $N(t)$ denotes the number of training samples at time t . In order to ensure a bounded computation time, we delete the oldest data point once the data set reaches the threshold $\bar{N} = 200$, yielding $\bar{\Delta} = c\bar{N}^2$. Examples of the tracking errors resulting from different values of c are depicted on the left side of Fig. 3. It can be clearly seen that larger delay bounds $\bar{\Delta}$ result in higher errors. When comparing to an offline trained GP model with computational delay $\bar{\Delta} = 0.01$, we can even observe that the growth in computation time due to online learning can increase the error, which is guaranteed to happen for the tracking error bound due to Theorem 5. In fact, this effect consistently occurs with increasing difference $\bar{\Delta}$ between computation times for the considered random initial data sets, as shown on the right side of Fig. 3. This is due to the linear growth of the error with event-triggered learning, which corresponds to the behavior of the tracking error bound guaranteed by Theorem 3. Therefore, all effects derived for the tracking error bounds can be observed in the actual tracking errors, which demonstrates the practical importance of the derived theoretical guarantees.

6. Conclusions

In this paper, the effect of computational delay caused by GP models on the control performance is analyzed, which reveals a trade-off between prediction accuracy and computational delay. An event-triggered online model update strategy under computational delays is proposed, and we show that it can offer advantages over offline trained GP models. Finally, simulations demonstrate the effectiveness of the derived theories and the event-triggered online learning strategy.

Acknowledgments

This work is supported by the Federal Ministry of Education and Research of Germany in the programme of “Souverän. Digital. Vernetzt.” under the joint project 6G-life with identification number: 16KISK002, and by the European Research Council (ERC) Consolidator Grant “Safe data-driven control for human-centric systems (CO-MAN)” under grant agreement number 864686, and by the European Union’s Horizon 2020 research and innovation programme under grant agreement number 871295.

References

- Alexandre Capone, Armin Lederer, and Sandra Hirche. Gaussian Process Uniform Error Bounds with Unknown Hyperparameters for Safety-Critical Applications. In *Proceedings of the International Conference on Machine Learning*, pages 2609–2624, 2022.
- Fernando Castañeda, Jason J Choi, Wonsuhk Jung, Bike Zhang, Claire J Tomlin, and Koushil Sreenath. Probabilistic Safe Online Learning with Control Barrier Functions. *arXiv preprint arXiv:2208.10733*, 2022.
- Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.
- Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 2021.
- Emilia Fridman. *Introduction to Time-delay Systems: Analysis and Control*. Springer, 2014.
- Lingchong Gao, Xiaobing Dai, Michael Kleeberger, and Johannes Fottner. Quasi-static Optimal Control Strategy of Lattice Boom Crane Based on Large-Scale Flexible Non-linear Dynamics. In *Simulation and Modeling Methodologies, Technologies and Applications: International Online Conference (SIMULTECH 2021)*, pages 153–177. Springer, 2023.
- Melissa Greeff and Angela P. Schoellig. Exploiting Differential Flatness for Robust Learning-based Tracking Control using Gaussian Processes. *IEEE Control Systems Letters*, 5(4):1121–1126, 2021.
- Feng Han and Jingang Yi. Stable Learning-based Tracking Control of Underactuated Balance Robots. *IEEE Robotics and Automation Letters*, 6(2):1543–1550, 2021.
- Daniel WC Ho, Junmin Li, and Yugang Niu. Adaptive Neural Nontrol for a Class of Nonlinearly Parametric Time-delay Systems. *IEEE Transactions on Neural Networks*, 16(3):625–635, 2005.
- Pushpak Jagtap, George J. Pappas, and Majid Zamani. Control Barrier Functions for Unknown Non-linear Systems using Gaussian Processes. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3699–3704, 2020.
- Junjie Jiao, Alexandre Capone, and Sandra Hirche. Backstepping Tracking Control using Gaussian Processes with Event-triggered Online Learning. *IEEE Control Systems Letters*, 6:3176–3181, 2022.

- Hassan K Khalil. *Nonlinear Systems*. Prentice-Hall, 2002.
- Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based Model Predictive Control for Safe Exploration. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6059–6066, 2018.
- Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control. *Advances in Neural Information Processing Systems*, 32, 2019.
- Armin Lederer, Alejandro J Ordóñez Conejo, Korbinian A Maier, Wenxin Xiao, Jonas Umlauft, and Sandra Hirche. Gaussian Process-based Real-time Learning for Safety Critical Applications. In *International Conference on Machine Learning*, pages 6055–6064. PMLR, 2021.
- Da-Peng Li, Yan-Jun Liu, Shaocheng Tong, CL Philip Chen, and Dong-Juan Li. Neural Networks-based Adaptive Control for Nonlinear State Constrained Systems with Input Delay. *IEEE Transactions on Cybernetics*, 49(4):1249–1258, 2018.
- Gabriel S Lima, Sebastian Trimpe, and Wallace M Bessa. Sliding Mode Control with Gaussian Process Regression for Underwater Robots. *Journal of Intelligent & Robotic Systems*, 99(3): 487–498, 2020.
- Michael Maiworm, Daniel Limon, and Rolf Findeisen. Online Learning-based Model Predictive Control with Gaussian Process Models and Stability Guarantees. *International Journal of Robust and Nonlinear Control*, pages 1–28, 2021.
- Franziska Meier and Stefan Schaal. Drifting Gaussian Processes with Varying Neighborhood Sizes for Online Model Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 264–269, 2016.
- Franziska Meier, Daniel Kappler, Nathan Ratliff, and Stefan Schaal. Towards Robust Online Inverse Dynamics Learning. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4034–4039, 2016.
- Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal Kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.
- Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. Model Learning with Local Gaussian Process Regression. *Advanced Robotics*, 23(15):2015–2034, 2009.
- Friedrich Solowjow, Dominik Baumann, Jochen Garcke, and Sebastian Trimpe. Event-Triggered Learning for Resource-Efficient Networked Control. In *Proceedings of the American Control Conference*, pages 6506–6512, 2018.
- Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.
- Jonas Umlauft and Sandra Hirche. Feedback Linearization based on Gaussian Processes with Event-triggered Online Learning. *IEEE Transactions on Automatic Control*, 65(10):4154–4169, 2019.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

Zheping Yan, Zewen Yang, Jinzhong Zhang, Jiajia Zhou, Anzuo Jiang, and Xue Du. Trajectory Tracking Control of UUV Based on Backstepping Sliding Mode With Fuzzy Switching Gain in Diving Plane. *IEEE Access*, 7:166788–166795, 2019.

Zhenxiao Yin and Hang Zhao. Implementation of Various Neural-Network-Based Adaptive Speed PI Controllers for Dual-Three-Phase PMSM. In *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6. IEEE, 2022.