Smart Forgetting for Safe Online Learning with Gaussian Processes

Jonas Umlauft JONAS.UMLAUFT@TUM.DE
Thomas Beckers T.BECKERS@TUM.DE
Alexandre Capone ALEXANDRE.CAPONE@TUM.DE
Armin Lederer ARMIN.LEDERER@TUM.DE
Sandra Hirche HIRCHE@TUM.DE
Chair of Information-oriented Control,
Department of Electrical and Computer Engineering,
Technical University of Munich, D-80333 Munich, Germany

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger[]

Abstract

The identification of unknown dynamical systems using supervised learning enables model-based control of systems that cannot be modeled based on first principles. While most control literature focuses on the analysis of a static dataset, online learning control, where data points are added while the controller is running, has rarely been studied in depth. In this paper, we present a data-efficient approach for online learning control based on Gaussian process models. To enable real-time capability despite high computational loads with growing datasets, we propose a safe forgetting mechanism. Using an entropy criterion, data points are selected based on their utility for the future trajectory under consideration of the stability of the closed-loop system. The approach is evaluated in a simulation and in a robotic experiment to demonstrate its computational efficiency.

Keywords: data-driven control, Gaussian processes, data-efficient learning, safe exploration, online learning, safe forgetting

1. Introduction

Data-driven techniques are disrupting various fields in engineering as the cost for processing and storing data is drastically decreasing. In control, supervised learning allows to identify unknown dynamical systems which are difficult to model based on first principles. Safety concerns of self-learning autonomous systems triggered the research on the stability of such systems (Koller et al., 2018). While many supervised learning techniques deliver accurate identification results, only a few are suitable for a rigorous analysis of safety. Gaussian processes (GPs) are preferred here due to their flexible nonparametric nature, and the inherent uncertainty measure (Rasmussen and Williams, 2006). The latter allows to bound the model error (Lederer et al., 2019) and enables uncertainty-aware control (Umlauft et al., 2018; Fanger et al., 2016). Most GP-based approaches separate the learning stage (model is trained on fixed data set), from the control stage, where the model-based control law is applied to the system (Beckers et al., 2019; Umlauft et al., 2017). However, to reduce the dependency of the control performance on the data set, adapting the model during the operation is promising (Chowdhary et al., 2015).

As nonparametric models store their information in the data points themselves, as opposed to a fixed number of parameters, the computational load increases with the size of the data set. This is not just challenging for all applications under real-time requirements (Nguyen-Tuong et al.,

2009) but makes data selection inevitable for learning control systems (Gijsberts and Metta, 2013). Event-triggered learning shows to be beneficial in such resource constraint learning control settings (Solowjow et al., 2018). Alternatively, variants of GP models have been proposed (Meier et al., 2016; Meier and Schaal, 2016; Snelson and Ghahramani, 2005) including sparse GPs with pseudo inputs (Quinonero-Candela and Rasmussen, 2005). However, existing model error bounds (Srinivas et al., 2012; Lederer et al., 2019) are not applicable to any of these variants.

This paper proposes an entropy-based forgetting rule for safe tracking control with an unknown dynamical system. By building on an online learning control law proposed by Umlauft and Hirche (2020), we improve the data set management by keeping only the most valuable data points considering the future desired trajectory. The approach guarantees asymptotic convergence of the tracking error based on an event-triggered model update. A robotic experiment demonstrates that the novel data set management reduces the computational complexity sufficiently and satisfies real-time constraints. The paper is organized as follows. Section 2 defines the problem setting and Sec. 3 reviews the online learning controller. The safe data selection is proposed in Sec. 4 followed by the experiments in Sec. 5.

2. Problem Formulation

We consider a nonlinear control-affine state feedback linearizable system[1]

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \cdots \quad \dot{x}_n = f(\boldsymbol{x}) + g(\boldsymbol{x})u, \quad \text{with} \quad \boldsymbol{x}_0 = \boldsymbol{x}(0), \tag{1}$$

with state $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\intercal \in \mathbb{X} \subseteq \mathbb{R}^n$ and input $u \in \mathbb{R}$ under the following assumptions.

Assumption 1 *The function $f\colon \mathbb{X} \to \mathbb{R}$ is unknown but globally bounded and differentiable, whereas the function $g\colon \mathbb{X} \to \mathbb{R}$ is known, differentiable and strictly positive.*

Differentiability and boundedness are assumptions which many physical systems satisfy naturally because, e.g., in real-world settings most systems saturate. The positiveness of $g(\cdot)$ ensures a globally well defined relative degree of the system (Khalil and Grizzle, 1996). Knowing $g(\cdot)$ is often assumed, e.g., for Lagrangian systems, where it corresponds to perfect knowledge of the generalized inertia matrix (which must therefore be modeled analytically in practice).

Assumption 2 *The state vector $\boldsymbol{x}^{(\kappa)} = \boldsymbol{x}(t_\kappa)$ and a noisy version of the highest derivative*

$$y^{(\kappa)} = \dot{x}_n(t_\kappa) - g(\boldsymbol{x}(t_\kappa))u + \epsilon^{(\kappa)} = f(\boldsymbol{x}(t_\kappa)) + \epsilon^{(\kappa)}, \qquad \epsilon^{(\kappa)} \sim \mathcal{N}(0, \sigma_{on}^2) \text{ i.i.d.} \tag{2}$$

can be measured at arbitrary time instances t_κ with $\kappa \in \mathbb{N}_0$. The dataset $\mathbb{D}_\kappa = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}_{i=1}^{N_\kappa}$ is updated at time t_κ, where $N_\kappa \in \mathbb{N}$ denotes the current number of data points.

Noise-free measurements are commonly assumed for state feedback control laws. The employed GP framework allows noise in the output (here: $\dot{x}_n$) which might result from numerical differentiation. The task is to track with state x_1 a desired trajectory x_d with the following property.

Assumption 3 *The trajectory x_d is bounded and at least $n-1$ times differentiable, thus $\boldsymbol{x}_d = \begin{bmatrix} x_d & \dot{x}_d & \cdots & \frac{d^{n-1}x_d}{dt^{n-1}} \end{bmatrix}^\intercal$ is continuous and $\frac{d^n x_d}{dt^n}$ is bounded.*

1. Lower/upper case bold symbols denote vectors/matrices, $\mathbb{R}_{+,0}$,$\mathbb{R}_+$/$\mathbb{N}_0$,$\mathbb{N}$ real positive/natural numbers with/without zero, $\boldsymbol{I}_n$ denotes a $n \times n$ identity matrix, $\mathcal{N}(\mu, \sigma^2)$ a Gaussian distribution and $\|\cdot\|$ the Euclidean norm.

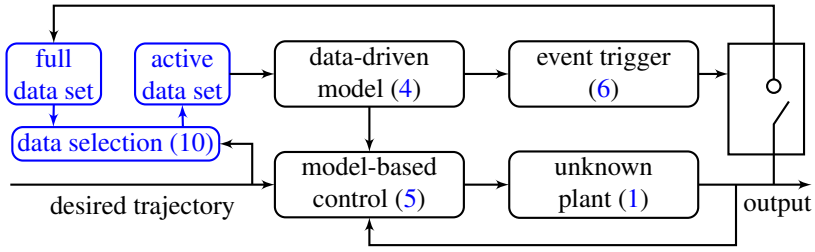

Figure 1: Online learning control framework with the proposed data selection in blue.

Using an online learned model $\hat{f}_\kappa \colon \mathbb{X} \to \mathbb{R}$ for $f(\cdot)$, we utilize a feedback linearizing controller

$$u_\kappa(\boldsymbol{x}) = \frac{1}{g(\boldsymbol{x})} \left(-\hat{f}_\kappa(\boldsymbol{x}) + \nu \right), \quad \kappa \in \mathbb{N}_0, \tag{3}$$

where $\nu \in \mathbb{R}$ is the input to the resulting approximately linearized system. Since the regression model $\hat{f}(\cdot)$ can be computationally expensive, real-time constraints might not be satisfied for arbitrarily large datasets $\mathbb{D}_\kappa$. Hence, we restrict the number of utilized training data as follows.

Assumption 4 *Considering computational constraints of the regression model $\hat{f}(\cdot)$ for the unknown function $f(\cdot)$ only $\bar{N}^a \in \mathbb{N}$ data points are allowed to be used in prediction, resulting in an active data set $\mathbb{D}_\kappa^a \subseteq \mathbb{D}_\kappa$, such that $|\mathbb{D}_\kappa^a| \leq \bar{N}^a$.*

The challenge is to find a selection of data which leads to a precise model (and thereby high tracking performance) and guarantees stability of the closed-loop system. An overview is provided in Fig. 1.

3. Event-triggered Online Learning

3.1. Identification with Gaussian processes

Consider measurements $y^{(i)} = f\left(\boldsymbol{x}^{(i)}\right) + \epsilon^{(i)}$, $i = 1, \ldots, N$ of $f(\cdot)$ with $\epsilon \sim \mathcal{N}(0, \sigma_{\text{on}}^2)$ and a GP model $f_{\mathcal{GP}}(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$ which is fully specified by a mean $m \colon \mathbb{X} \to \mathbb{R}$ and covariance $k \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ function. For notational convenience, we concatenate the measurements $\boldsymbol{y} = \begin{bmatrix} y^{(1)} & \cdots & y^{(N)} \end{bmatrix}^\intercal$, denote $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$, define the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ by $K_{ij} = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$ and set the prior mean function to zero $m(\boldsymbol{x}) = 0$. By conditioning on the data $\mathcal{X}$, $\boldsymbol{y}$, we obtain a posterior Gaussian distribution at each input $\boldsymbol{x}^*$ given by

$$\mu_{\mathbb{D}_N}(\boldsymbol{x}^*) \coloneqq \boldsymbol{k}^\intercal (\boldsymbol{K} + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} \boldsymbol{y}, \qquad \sigma_{\mathbb{D}_N}^2(\boldsymbol{x}^*) \coloneqq k^*(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^\intercal (\boldsymbol{K} + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} \boldsymbol{k}, \tag{4}$$

where the elements of the covariance vector $\boldsymbol{k}$ are given by $k_i = k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^*)$. The kernel $k(\cdot, \cdot)$ is specified with hyperparameters usually determined through a likelihood optimization (Rasmussen and Williams, 2006). However, for the sake of focus, this paper considers hyperparameters to be given and refers to the literature for a discussion of misspecified kernels (Beckers et al., 2018).

3.2. Feedback Linearizing Control law with Event-triggered Model Update

The input ν to the approximately linearized system uses the filtered state $r = \begin{bmatrix} \boldsymbol{\lambda}^\intercal & 1 \end{bmatrix} \boldsymbol{e}$, where the Hurwitz coefficient vector $\boldsymbol{\lambda} \in \mathbb{R}^{n-1}$ implies exponential convergence of $\boldsymbol{e} \coloneqq \boldsymbol{x} - \boldsymbol{x}_d \to \boldsymbol{0}$

as $r \to 0$ (Yesildirak and Lewis, 1995). In the filtered state dynamics $\dot{r} = f(\boldsymbol{x}) + g(\boldsymbol{x})u(\boldsymbol{x}) + \rho$, where $\rho = [e_2 \cdots e_n]\,\boldsymbol{\lambda} - \frac{d^n x_d}{dt^n}$, the nonlinearity is canceled via state feedback linearization (3) and the input $\nu = -k^c r - \rho$ with $k^c \in \mathbb{R}_+$ is applied to the approximately linearized system. Hence,

$$u_\kappa(\boldsymbol{x}) = \frac{1}{g(\boldsymbol{x})}\left(-\hat{f}_\kappa(\boldsymbol{x}) - k^c r - \rho\right), \quad \forall t \in [t_\kappa; t_{\kappa+1}). \tag{5}$$

The model $\hat{f}_\kappa(\cdot) = \mu_{\mathbb{D}_\kappa}(\cdot)$ is only updated when the uncertainty, given by $\sigma^2_{\mathbb{D}_\kappa}(\cdot)$, becomes too large to ensure convergence of the tracking error. Thus, new training data is added in an event-triggered fashion at time $t_{\kappa+1}$ to ensure global uniform ultimate boundedness of the tracking error (Umlauft and Hirche, 2020, Corollary 4).

Lemma 1 *Under Assumptions 1-3, the system (1) controlled by (5) and model $\hat{f}_\kappa(\cdot) = \mu_{\mathbb{D}_\kappa}(\cdot)$, which is updated according to the event-triggering law*

$$t_{\kappa+1} \coloneqq \{t > t_\kappa | \beta_{\mathbb{D}_\kappa}\sigma_{\mathbb{D}_\kappa}(\boldsymbol{x}) \geq k^c|r| \cap \boldsymbol{e} \notin \mathbb{B}_{\sigma_{on}}\}, \tag{6}$$

is, with probability $1-\delta$, $\delta \in (0;1)$, globally uniformly ultimately bounded. The ultimate bound is

$$\mathbb{B}_{\sigma_{on}} = \left\{ \boldsymbol{e} \in \tilde{\mathbb{X}} \middle| \, \|\boldsymbol{e}\| \leq \frac{\sigma_{on}\beta_{\mathbb{D}_\kappa}}{k^c \|\begin{bmatrix}\boldsymbol{\lambda}^\intercal & 1\end{bmatrix}\|} \right\}, \tag{7}$$

where $\beta_{\mathbb{D}_\kappa} \in \mathbb{R}_+$ is a constant such that (Srinivas et al., 2012, Theorem 6) is satisfied, i.e.

$$P(|f(\boldsymbol{x}) - \mu_{\mathbb{D}_\kappa}(\boldsymbol{x})| \geq \beta_{\mathbb{D}_\kappa}\sigma_{\mathbb{D}_\kappa}(\boldsymbol{x}), \forall \boldsymbol{x} \in \tilde{\mathbb{X}}, \kappa \in \mathbb{N}_0) \leq \delta. \tag{8}$$

According to boundedness arguments in Umlauft and Hirche (2020, Corollary 1), the state converges to a compact set $\tilde{\mathbb{X}} \subset \mathbb{X}$ as required for (8). So far, the model uses the full data set $\mathbb{D}_\kappa$, which grows unboundedly and therefore violates Assumption 4. The next section resolves this issue.

4. Efficient Data Selection

To ensure that at most $\bar{N}^a$ data points are employed by the model $\hat{f}(\cdot)$, we choose an active data set $\mathbb{D}^a_\kappa$ to ensure high tracking performance while guaranteeing convergence. Given the inputs of the full data set $\mathcal{X}_{\mathbb{D}_\kappa} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N_\kappa)}\} \subset \mathbb{X}$, we select a subset $\mathcal{X}_{\mathbb{D}^a_\kappa} \subset \mathcal{X}_{\mathbb{D}_\kappa}$ of size $\bar{N}^a$ to maximize prediction accuracy of our model $\hat{f}(\cdot)$ at states of interest given by a finite set $\mathcal{X}_v \subset \mathbb{X}$.

4.1. Information value of data points

The output of our GP model $f_{\mathcal{GP}}(\boldsymbol{x})$ at the states $\boldsymbol{x} \in \mathcal{X}_v$ can be considered as random variables with joint Gaussian distribution and the model quality can be measured by the associated entropy. Let $\boldsymbol{\Sigma}_{\mathcal{X}_v}$ denote the covariance matrix obtained from (4), the entropy is given by $H(\mathcal{X}_v) = 0.5\log\left(2\pi\det(\boldsymbol{\Sigma}_{\mathcal{X}_v})\right)$, where $H(\mathcal{X}_v) \coloneqq H\left(f_{\mathcal{GP}}\left(\boldsymbol{x}^{(1)}\right), \ldots, f_{\mathcal{GP}}\left(\boldsymbol{x}^{(N_\kappa)}\right)\right)$ to simplify notation. The conditional entropy $H(\mathcal{X}_v|\mathcal{X}_{\mathbb{D}^a_\kappa}) \coloneqq H(\mathcal{X}_v, \mathcal{X}_{\mathbb{D}^a_\kappa}) - H(\mathcal{X}_{\mathbb{D}^a_\kappa})$ quantifies the information about $f(\cdot)$ at $\mathcal{X}_v$, when using $\mathbb{D}^a_\kappa$. Using this criterion, the most informative data set $\mathbb{D}^a_\kappa \subset \mathbb{D}_\kappa$ with respect to the function values at $\mathcal{X}_v$ is the solution of the optimization problem

$$\mathbb{D}^{a^*}_\kappa = \underset{\mathbb{D}^a_\kappa \subset \mathbb{D}_\kappa,\ |\mathbb{D}^a_\kappa| = \bar{N}^a}{\arg\min} H(\mathcal{X}_v|\mathcal{X}_{\mathbb{D}^a_\kappa}). \tag{9}$$

4.2. Safe and optimal data selection

The conditional entropy criterion selects the optimal training set with respect to a set of states of interest $\mathcal{X}_v$. However, reducing the amount of data from N_κ to $\bar{N}^a$ leads to higher uncertainties in the model, which possibly violates the stability stated in Lemma 1. To ensure that the convergence guarantee is maintained, we consider an additional constraint in the optimization in (9).

Theorem 2 *Consider the system (1), the control law (3) and the event trigger (6). Furthermore, consider Assumptions 1-4 and a data point selection*

$$\mathbb{D}_\kappa^{a^*} = \underset{\mathbb{D}_\kappa^a \subset \mathbb{D}_\kappa, |\mathbb{D}_\kappa^a| = \bar{N}^a}{\arg\min} H(\mathcal{X}_v | \mathcal{X}_{\mathbb{D}_\kappa^a}). \qquad \text{s.t.} \quad \beta_{\mathbb{D}_\kappa^a} \sigma_{\mathbb{D}_\kappa^a}(\boldsymbol{x}) < k^c |r|, \tag{10}$$

then, for any $\mathcal{X}_v$, the tracking error is globally uniformly ultimately bounded to the set $\mathbb{B}_{\sigma_{on}}$ with probability $1-\delta$, $\delta \in (0;1)$, following the definitions in (7) and (8).

Proof Similar to the proof of Lemma 1, the triggering condition is defined such that a new measurement is taken whenever the uncertainty becomes too large to possibly violate the decreasing Lyapunov condition for $V(\boldsymbol{x}) = r^2$. The constraint $\beta_{\mathbb{D}_\kappa^a} \sigma_{\mathbb{D}_\kappa^a}(\boldsymbol{x}) < k^c |r|$ ensures that this condition still holds for the current state $\boldsymbol{x}$ after reducing the training data set. The analysis of the posterior variance shows that $\sigma_{\mathbb{D}}(\boldsymbol{x}) \leq \sigma_{\text{on}}^2$ as long as $\boldsymbol{x} \subset \mathcal{X}_{\mathbb{D}}$. This ensures feasibility of the optimization problem (10), since for all $\boldsymbol{x} \in \mathbb{X} \setminus \mathbb{B}_{\sigma_{\text{on}}}$ there exists at least one $\mathbb{D}_\kappa^a$ (which contains the current state $\boldsymbol{x}$) such that the constrained is fulfilled. ■

Remark 3 *To avoid the combinatorial problem (10), which is computationally very complex, a greedy approximation can be employed as investigated by Krause et al. (2008). However, special care must be taken to ensure the constraint in (10) is fulfilled to ensure the stability given by Theorem 2. Algorithm 2 guarantees this by always including the current state in the training data set in the initialization.*

As shown by Theorem 2, the choice of the set $\mathcal{X}_v$ is not critical for the convergence but can affect the number of triggered events or the modeling error $|\hat{f}(\cdot) - f(\cdot)|$ which has crucial impact on the tracking precision. The optimization (10) minimizes this error by means of the following result.

Proposition 1 *If $\mathcal{X}_v = \{\boldsymbol{x}\}$, there exists no set $\tilde{\mathbb{D}} \subset \mathbb{D}_\kappa$, $\left|\tilde{\mathbb{D}}\right| = \bar{N}^a$, for which the bound of the model error at state $\boldsymbol{x}$ is smaller than for $\mathbb{D}_\kappa^{a^*}$ obtained by (10), hence $\beta_{\mathbb{D}_\kappa^{a^*}} \sigma_{\mathbb{D}_\kappa^{a^*}}(\boldsymbol{x}) \leq \beta_{\tilde{\mathbb{D}}} \sigma_{\tilde{\mathbb{D}}}(\boldsymbol{x})$.*

Proof As β only depends on the number of training data $\bar{N}^a$ but not on their distribution, $\beta_{\tilde{\mathbb{D}}} = \beta_{\mathbb{D}_\kappa^{a^*}}$ holds. Minimizing the conditional entropy $H(\{\boldsymbol{x}\} | \mathcal{X}_{\mathbb{D}_\kappa^a}) = 0.5 \log(\sigma^2_{\mathbb{D}_N}(\boldsymbol{x})) + 0.5(\log(2\pi) + 1)$ results in a minimal $\sigma^2_{\mathbb{D}_N}(\boldsymbol{x})$ due to the monotony, which yields the presented result. ■

In general, any choices for $\mathcal{X}_v$, are possible as, e.g. uniform grids over the state space (Krause et al., 2008), but, since our control law only requires an accurate model near the trajectory, we propose to use a finite subset of the future desired trajectory $\boldsymbol{x}_d(t)$, $t \in [t_\kappa; \infty)$

$$\mathcal{X}_v = \{\boldsymbol{x}_d(t_\kappa), \boldsymbol{x}_d(t_\kappa + \Delta t_d), \ldots, \boldsymbol{x}_d(t_\kappa + N_d \Delta t_d))\}, \tag{11}$$

where $\Delta t_d \in \mathbb{R}_+$ and $N_d \in \mathbb{N}$. This choice attempts to reduce the number of triggered events as the uncertainty along the future trajectory is reduced, but a more depth analysis is outside the scope of this paper. As $\mathcal{X}_v$ must be finite, it requires a discretization, but only a long a single dimension (time) and not across the entire state space.

Algorithm 1: Data-efficient online learning

initialize $\kappa = 0$, $\mathbb{D}_0 = \{\}$, $\hat{f}_0 = 0$
while *simulation time not exceeded* **do**
  while $\beta_{\mathbb{D}_\kappa}\sigma_{\mathbb{D}_\kappa}(\boldsymbol{x}) \geq k^c|r|$ **do**
    run controller u_κ in (5)
  end
  set $\kappa \leftarrow \kappa + 1$ and measure $\boldsymbol{x}^{(\kappa)} = \boldsymbol{x}(t_\kappa)$ and $y^{(\kappa)} = \dot{x}_n(t_\kappa) + \epsilon^{(\kappa)}$
  add training point $\mathbb{D}_\kappa = \mathbb{D}_{\kappa-1} \cup \left\{\left(\boldsymbol{x}^{(\kappa)}, y^{(\kappa)}\right)\right\}$
  set $\mathcal{X}_v$ based on (11) and select $\mathbb{D}^a_\kappa$ from $\mathbb{D}_\kappa$ by solving (10) (or set $\mathbb{D}^a_\kappa = \mathbb{D}_\kappa$ if $|\mathbb{D}_\kappa| \leq \bar{N}^a$)
  update $\hat{f}_\kappa(\cdot)$ and $\sigma_{\mathbb{D}_\kappa}(\cdot)$ based on $\mathbb{D}^a_\kappa$
end

Algorithm 2: Greedy approximation to minimize the conditional entropy.

input: $\mathcal{X}_{\mathbb{D}_\kappa}, \mathcal{X}_v, \bar{N}^a, \boldsymbol{x}^{(t_\kappa)}$, **output**: $\mathcal{X}^*_{\mathbb{D}^a_\kappa}$
initialize $\mathcal{X}^*_{\mathbb{D}^a_\kappa} = \left\{\boldsymbol{x}^{(t_\kappa)}\right\}$;
for $\bar{n} = 1, \ldots, \bar{N}^a - 1$ **do**
  $\mathcal{X}^*_{\mathbb{D}^a_\kappa} \leftarrow \mathcal{X}^*_{\mathbb{D}^a_\kappa} \cup \underset{\boldsymbol{x}^{(i)} \in \mathcal{X}_{\mathbb{D}_\kappa}}{\arg\min}\, H(\mathcal{X}_v | \mathcal{X}^*_{\mathbb{D}^a_\kappa} \cup \boldsymbol{x}^{(i)})$;
end

5. Experiments

An overview of the overall control design is given in Algorithm 1. A simulation and a robotic experiment are presented to illustrate and demonstrate the benefits of the proposed approach.

5.1. Simulation

Setup: The system to control is given by $\dot{x}_1 = x_2$, $\dot{x}_2 = 1 - \sin(x_1) + \frac{0.5}{1+\exp(-x_2/10)} + u$. An upper limit of $\bar{N}^a = 10$ is imposed on the number of data points employed in the GP model. We set $N_d = 100$ and $\Delta t_d = \pi/100$ in (11) and take measurements with a noise level of $\sigma^2_{\text{on}} = 10^{-16}$. The hyperparameters are fixed, i.e. no likelihood optimization during the simulation and $\beta_{\mathbb{D}_\kappa} = 7$ is constant for all κ. For the sake of focus, we do not compute an exact value for $\beta_{\mathbb{D}_\kappa}$ but point to related work, which provide methods to efficiently upper bound it (Srinivas et al., 2012). To avoid any numeric difficulties, a lower bound on the filtered state is implemented $|r| > 10^{-2}$.

To highlight the benefits of the proposed approach, we compare the data selection in Algorithm 2 with a selection based on a uniform grid, similarly to Krause et al. (2008). We choose a sinusoidal reference trajectory with increasing amplitude $x_d(t) = 0.2t\sin(t)$. Figure 2 shows the resulting trajectories at $t = 12$ and the active set of training data. For the proposed entropy-based selection, it is observed that data points near $\mathcal{X}_v$ are chosen, resulting in small values of $\sigma_{\mathbb{D}_\kappa}(\cdot)$ along the future desired trajectory. This results in less events (36) compared to the grid-based data selection (63).

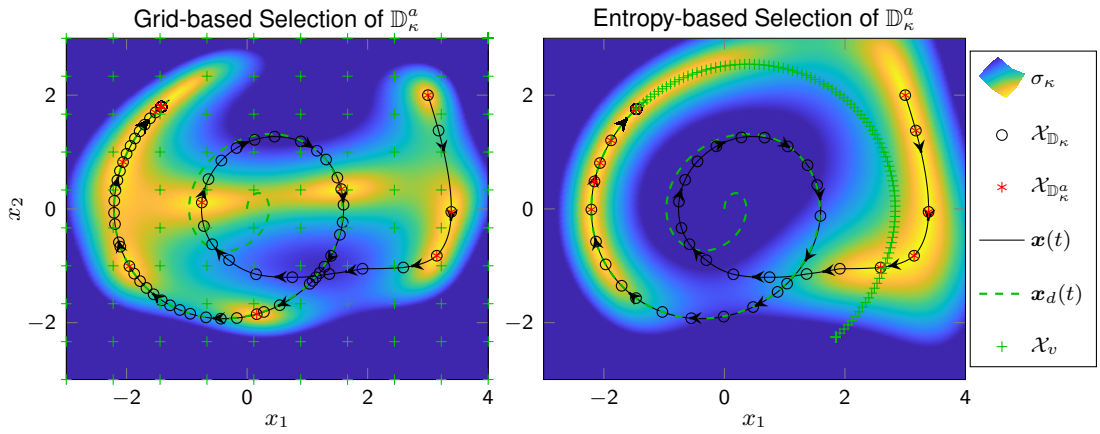

Figure 2: Comparison between grid-based selection (left) and proposed selection method (right). Black circles indicate all points $\mathbb{D}_\kappa$, red asterisks the selected points $\mathbb{D}_\kappa^a$, black solid line illustrates the actual, the green dashed the desired trajectory. The colormap shows the standard deviation $\sigma_{\mathbb{D}_\kappa^a}(\cdot)$ after the last update (yellow indicates low, blue high values).

5.2. Robotic Experiment

Setup: For the evaluation in a robotic experiment, the two degrees of freedom (DoF) manipulator CARBO, shown in Fig. 3, is used. With two rotational joints $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^\intercal \in \mathbb{R}^2$, its dynamics are

$$\ddot{\boldsymbol{q}} = \underbrace{-\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1}\left(\boldsymbol{C}_{\text{rob}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_{\text{rob}}(\boldsymbol{q})\right)}_{\boldsymbol{f}(\boldsymbol{x})} + \underbrace{\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1}}_{\boldsymbol{G}(\boldsymbol{x})}\boldsymbol{u}, \tag{12}$$

where $\boldsymbol{M}_{\text{rob}}\colon \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$ is the inertia matrix, $\boldsymbol{C}_{\text{rob}}\colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^{2\times 2}$ is the Coriolis matrix and $\boldsymbol{g}_{\text{rob}}\colon \mathbb{R}^2 \to \mathbb{R}^2$ is the gravity vector. The input $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\intercal \in \mathbb{R}^2$ is a vector of the applied torques in each joint, the state is $\boldsymbol{x} = \begin{bmatrix} q_1 & q_2 & \dot{q}_1 & \dot{q}_2 \end{bmatrix}^\intercal$. Since $\boldsymbol{G}(\cdot)$ is assumed to be known, a parametric model for the inertia matrix $\boldsymbol{M}_{\text{rob}}(\cdot)$ is employed (Murray, 2017) where the parameters values for the moments of inertia $\mathfrak{I}_1$, $\mathfrak{I}_2$, the centers of mass $\tilde{\mathfrak{l}}_1$, $\tilde{\mathfrak{l}}_2$, the lengths of the links $\mathfrak{l}_1$, $\mathfrak{l}_2$ and the masses $\mathfrak{M}_1$, $\mathfrak{M}_2$ are provided in Table 1. The expressions for $\boldsymbol{C}_{\text{rob}}(\cdot,\cdot)$ and $\boldsymbol{g}_{\text{rob}}(\cdot)$ are considered unknown according to Assumption 1. The proposed online learning approach identifies $\boldsymbol{f}(\cdot)$, which includes all unknown components in the dynamics (e,g. external forces, friction) which are challenging to model analytically. Since $\boldsymbol{f}(\cdot)$ is two-dimensional, two independent GPs are employed. The input data are for both the same measurements of the state $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}^\intercal & \dot{\boldsymbol{q}}^\intercal \end{bmatrix}^\intercal$, the output data are the rows of $-\boldsymbol{M}_{\text{rob}}(\boldsymbol{q})^{-1}\left(\boldsymbol{C}_{\text{rob}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_{\text{rob}}(\boldsymbol{q})\right)$, respectively. The even-triggering law (6) is extended to multiple dimensions using a logical OR-operation to trigger whenever one of the GPs has reached the threshold, thus $t_{\kappa+1} \coloneqq \{t > t_\kappa \,|\beta_{1,\mathbb{D}_\kappa}\sigma_{1,\mathbb{D}_\kappa}(\boldsymbol{x}) \geq k_1^c|r_1| \vee \beta_{2,\mathbb{D}_\kappa}\sigma_{2,\mathbb{D}_\kappa}(\boldsymbol{x}) \geq k_2^c|r_2|\}$. The hyperparameters of the GP models are set to fixed values (see Table 1) obtained from an offline learning procedure along the same reference trajectories $x_{1,d} = \frac{\pi}{6}\cos(0.5t)$ and $x_{2,d} = \frac{\pi}{6}\cos(t)$ for joint 1 and 2, respectively. The controller runs on an Ubuntu 14.04 real-time kernel in Matlab/Simulink 2017 with a sample rate of 1 kHz. To keep the real-time constraint, the dataset is limited $N_\kappa \leq 80$, from which $\bar{N}^a = 10$ are selected based on Algorithm 2. The limit N_κ is not considered in the theoretic analysis, thus an additional mechanism (e.g. deleting the oldest data point)

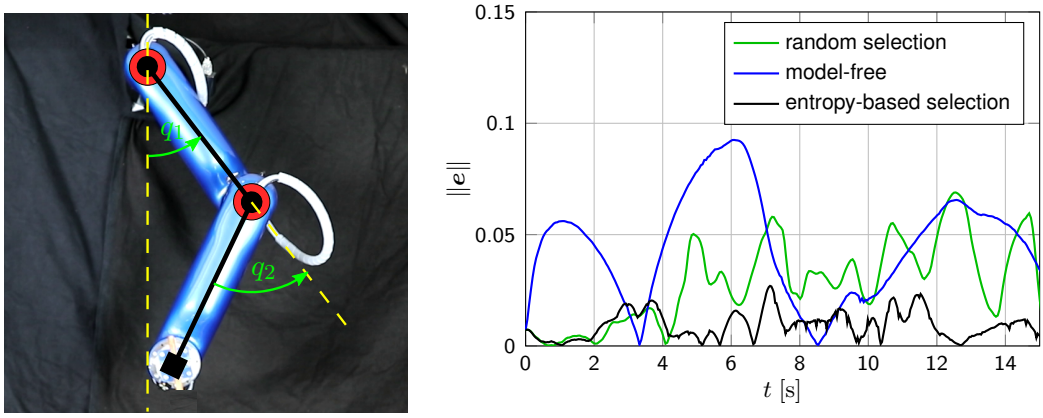

Figure 3: Left: The two DoF robotic manipulator CARBO used for evaluation of the proposed event-triggered online learning approach. Right: Norm of the tracking error $\|\boldsymbol{e}\|$ for the proposed entropy-based selection (black, RMSE: 0.010), a random selection (green, RMSE: 0.028) and a model-free PD-controller (green, RMSE: 0.045).

$\zeta_{f_1}^2$	$\zeta_{f_2}^2$	$\ell_{1\dots n,f_1}^2$	$\ell_{1\dots n,f_2}^2$	$\beta_{1,\mathbb{D}_\kappa}$	$\beta_{2,\mathbb{D}_\kappa}$	k_1^c	k_2^c	λ_1,λ_2	$\mathfrak{l}_1,\mathfrak{l}_2$	$\tilde{\mathfrak{l}}_1,\tilde{\mathfrak{l}}_2$	$\mathfrak{I}_1,\mathfrak{I}_2$	$\mathfrak{M}_1,\mathfrak{M}_2$
39	80	[0.8 4.9 13 11]	[13 3913 26 2.5]	29	21	4	3	10	0.3 m	0.15 m	1 kg m^2	1.5 kg

Table 1: Parameters of the online learning controller in the robotic experiment.

is required to guarantee the stability. Furthermore, we chose $\mathcal{X}_v = \boldsymbol{x}$ to minimize the computational complexity for the optimization (9). Figure 3 shows the resulting tracking error for a model-free PD-controller, randomly selected data and the proposed selection method. The significantly lower RMSE of the proposed methods indicates the superiority to a random data selection.

6. Conclusion

This paper proposes an entropy-based forgetting strategy for data-efficient online learning. Based on Gaussian processes, an optimal selection of the training set is performed with respect to the future desired trajectory. Rather than striving for a global approximation, we aim for high precision of the model in the region of interest. As a result, accurate inference on a reduced data set is achieved solving the challenge of a high computational complexity of Gaussian process models. This enabled us to apply the proposed control law under real-time constraint in a robotic experiment. The selection itself requires also computational resources, but only if an event occurs, which happens rarely compared to the prediction (in our experiment approximately 10^2 times less often). Furthermore, we have shown that the provided stability guarantees of the event-triggered online learning hold despite the reduced data set making the approach applicable also in safety critical domains.

For future work, an online adaptation of the hyperparameters of the GP should be considered since an initialization run as performed in the robotic experiment might not always be possible in practice. New theoretical difficulties arise if the likelihood optimization cannot be completed within the real-time constraint, but, e.g., runs at a lower update rate than the main control loop.

References

Thomas Beckers, Jonas Umlauft, and Sandra Hirche. Mean square prediction error of misspecified Gaussian process models. In *Conference on Decision and Control (CDC)*, pages 1162–1167, December 2018.

Thomas Beckers, Dana Kulić, and Sandra Hirche. Stable Gaussian process based tracking control of Euler-Lagrange systems. *Automatica*, 23(103):390–397, 2019.

Girish Chowdhary, Hassan A. Kingravi, Jonathan. P. How, and Patricio A. Vela. Bayesian nonparametric adaptive control using Gaussian processes. *IEEE Transactions on Neural Networks and Learning Systems*, 26(3):537–550, March 2015.

Yunis Fanger, Jonas Umlauft, and Sandra Hirche. Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3913–3919. IEEE, October 2016.

Arjan Gijsberts and Giorgio Metta. Real-time model learning using incremental sparse spectrum Gaussian process regression. *Neural Networks*, 41:59–69, 2013. Special Issue on Autonomous Learning.

Hassan K. Khalil and Jessy. W. Grizzle. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *Conference on Decision and Control (CDC)*, pages 6059–6066, December 2018.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for Gaussian process regression with application to safe control. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2019.

Franziska. Meier and Stefan Schaal. Drifting Gaussian processes with varying neighborhood sizes for online model learning. In *International Conference on Robotics and Automation (ICRA)*, pages 264–269, May 2016.

Franziska Meier, Daniel Kappler, Nathan Ratliff, and S. Stefan Schaal. Towards robust online inverse dynamics learning. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

Richard M. Murray. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

Duy Nguyen-Tuong, Jan R. Peters, and Matthias Seeger. Local Gaussian process regression for real time online model learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1193–1200. Curran Associates, Inc., 2009.

Joaquin Quinonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, January 2006.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1257–1264, 2005.

Friedrich Solowjow, Dominik Baumann, Jochen Garcke, and Sebastian Trimpe. Event-triggered learning for resource-efficient networked control. In *American Control Conference (ACC)*, pages 6506–6512. IEEE, 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, May 2012.

Jonas Umlauft and Sandra Hirche. Feedback linearization based on Gaussian processes with event-triggered online learning. *IEEE Transactions on Automatic Control (TAC)*, 2020.

Jonas Umlauft, Thomas Beckers, Melanie Kimmel, and Sandra Hirche. Feedback linearization using Gaussian processes. In *Conference on Decision and Control (CDC)*, pages 5249–5255. IEEE, December 2017.

Jonas Umlauft, Lukas Pöhler, and Sandra Hirche. An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process. *IEEE Control Systems Letters*, 2(3): 483–488, July 2018.

Aydin Yesildirak and Frank L. Lewis. Feedback linearization using neural networks. *Automatica*, 31(11):1659–1664, 1995.