Asynchronous Distributed Gaussian Process Regression

Zewen Yang*, **Xiaobing Dai***, **Sandra Hirche**

Technical University of Munich
zewen.yang@tum.de, xiaobing.dai@tum.de, hirche@tum.de

Abstract

In this paper, we address a practical distributed Bayesian learning problem with asynchronous measurements and predictions due to diverse computational conditions. To this end, asynchronous distributed Gaussian process (AsyncDGP) regression is proposed, which is the first effective online distributed Gaussian processes (GPs) approach to improve the prediction accuracy in real-time learning tasks. By leveraging the devised evaluation criterion and established prediction error bounds, AsyncDGP enables the distinction of contributions of each model for prediction ensembling using aggregation strategy. Furthermore, we extend its utility to dynamic systems by introducing a learning-based control law, ensuring guaranteed control performance in safety-critical applications. Additionally, a networked online learning simulation platform for distributed GPs, namely online GP gym (GPgym), is introduced for testing the performance of learning and control of dynamical systems. Numerical simulations within GPgym across regression tasks with real-world data sets and dynamical control scenarios demonstrate the effectiveness and applicability of AsyncDGP.

Introduction

Distributed learning, employing parallelized training and cooperative learning within distributed systems, holds promise for enhancing the efficiency of machine learning in the training and prediction process. The application of distributed learning extends across diverse domains, including but not limited to multi-agent systems (Yan et al. 2020), edge computing (Chen and Ran 2019), and networked Internet of Things devices (Park and Saad 2019), etc. Specifically, for the safe operation of systems in complex and dynamic environments, real-time predictions and prompt model updates must be employed. However, the implementation of online learning within a distributed framework is confronted with challenges of the inherent heterogeneity in computational nodes, leading to asynchronous predictions (see Figure 1). In other words, variations in computation speeds among nodes result in distinct computational times, while discrepancies in data volume contribute to different processing times. Even in scenarios where computational nodes are homogeneous, non-simultaneous measurement and data reception exhibit asynchrony. Addressing this issue becomes imperative for ensuring the seamless usage of distributed learning algorithms in real-world applications, especially in contexts where timely and accurate predictions are essential for the safe real-time operations of dynamical systems[1].

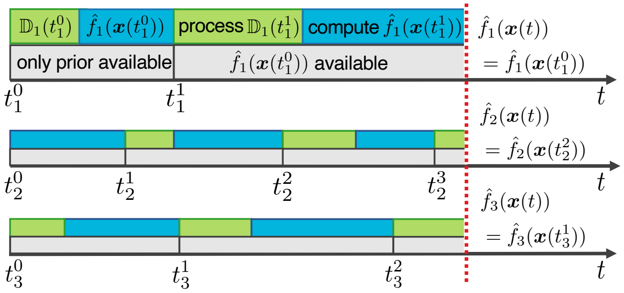
$\mathbb{D}_1(t_1^0)$
$\hat{f}_1(\boldsymbol{x}(t_1^0))$
process $\mathbb{D}_1(t_1^1)$
compute $\hat{f}_1(\boldsymbol{x}(t_1^1))$
only prior available
$\hat{f}_1(\boldsymbol{x}(t_1^0))$ available
$\hat{f}_1(\boldsymbol{x}(t))$
$= \hat{f}_1(\boldsymbol{x}(t_1^0))$
t_1^0
t_1^1
t
$\hat{f}_2(\boldsymbol{x}(t))$
$= \hat{f}_2(\boldsymbol{x}(t_2^2))$
t_2^0
t_2^1
t_2^2
t_2^3
t
$\hat{f}_3(\boldsymbol{x}(t))$
$= \hat{f}_3(\boldsymbol{x}(t_3^1))$
t_3^0
t_3^1
t_3^2
t

Figure 1: Asynchronous distributed predictions. The temporal sequence of the learning process of each computational node i is characterized by discrete phases: a green block representing the time frame for receiving and allocating data set $\mathbb{D}_i$ at its own time t_i^k with $k \in \mathbb{N}$, and a blue block denoting the computational duration of the prediction $\hat{f}_i$ of the true function f. During the intervening gray block, only previous predictions are available.

Related Work

To achieve online learning in distributed systems, we consider Gaussian process (GP) regression, a non-parametric supervised learning technique, increasingly utilized in safety-critical applications for its high expressive capabilities and probabilistic guaranteed bounded prediction errors (Hashimoto et al. 2022). As the computational complexity associated with updates and predictions in GP regression escalates with the expanding number of training points, an effective way to mitigate the issue is distributed computing inducing distributed Gaussian processes (DGPs) (Deisenroth and Ng 2015), where a center node aggregates the predictions from the distributed nodes (experts) with the divided sub-dataset. One such approach is

*These authors contributed equally.

[1] Extended related work, proofs, and results are available in the complementary document at arXiv (Yang, Dai, and Hirche 2024).

the mixture of experts (MOE) method (Tresp 2000b; Yuan and Neubauer 2008; Masoudnia and Ebrahimpour 2014), where each model's prediction is constant weight. Additionally, the product of experts (POE) family methods (Ng and Deisenroth 2014; Cao and Fleet 2015; Cohen et al. 2020), which determine aggregation weights using posterior variance, and Bayesian committee machine (BCM) family methods (Tresp 2000a; Deisenroth and Ng 2015; Liu et al. 2018), considering prior variance, have been proposed. Another avenue involves fusion methods utilizing dynamical average consensus (Lederer et al. 2023). However, this approach requires solving differential equations and interacting with the computational nodes, leading to additional computational delay. Despite these efforts, the prevailing issue of asynchronous DGP predictions remains unaddressed, posing inevitable challenges due to inherent variations and noise across diverse models. Peng et al. (2017) propose an offline asynchronous distributed variational GP method, however, the delay is introduced by the variational approximation meaning this method exacerbates the asynchronous problem rather than resolving it. The studies (Nguyen et al. 2022; Egelé et al. 2023) proposed an asynchronous optimization method for large-scale hyperparameter optimization, considering asynchronous communication between nodes. However, these approaches require pre-partitioning the entire dataset and limit the applicability to online learning scenarios where data arrives sequentially. While agent-based systems integrating DGPs demonstrate promise in cooperative learning (Yang et al. 2021; Dai et al. 2024a,b; Yang et al. 2024a,b), achieving simultaneous predictions remains a significant challenge without delay due to the communication and computational time (Dai et al. 2023). Therefore, there exists no algorithm designed explicitly to alleviate the problems associated with asynchronous predictions and delayed measurements in online learning with GPs.

Contribution

In addressing this practical problem posed by the nature of asynchronous online inference, we propose a multi-model learning methodology employing asynchronous distributed Gaussian processes (AsynDGPs), which answers two key questions. *Question 1: Are the previous temporal predictions useless? Question 2: How can the prior model of GP be applied in an asynchronous prediction scenario?* Moreover, we introduce a quantifiable criterion for assessing predictions from distinct GP models. In addition, a comprehensive analysis is presented establishing the prediction error bounds for aggregated predictions in asynchronous learning. Then, based on the introduced distributed learning framework, we formulate a learning-based protocol designed for the control task of dynamical systems. Through quantitative evaluations of the learning process, we ensure that control performance is guaranteed within a deterministic ultimate bound. Additionally, we introduce a MATLAB simulation platform called online GP gym (GPgym) to verify the online learning algorithms while considering the delays in both predictions and observations. Furthermore, we conduct real-world benchmarks achieving significant improvements in prediction accuracy over state-of-the-art DGP approaches in the regression task. Lastly, a dynamical system track control task simulation demonstrates the applicability of AsynDGP in safety-critical applications.

Problem Statement

This paper investigates a distributed system comprising $M \in \mathbb{N}_{>0}$ computational nodes. The primary objective is to cooperatively infer an unknown function $f : \mathbb{R}^n \to \mathbb{R}$ using its estimated function $\hat{f}$. To accomplish this goal, each node within the distributed system is equipped with a GP model for predictions with a data set $\mathbb{D}_i = \left\{\left(\boldsymbol{x}^{(\iota)}, y^{(\iota)}\right)\right\}_{\iota=1,\cdots,N_i}$ with $N_i = |\mathbb{D}_i|$, where $i = 1, \ldots, M$.

A Gaussian Process induces a Gaussian distribution of an unknown function f defined by a prior mean $m : \mathbb{R}^n \to \mathbb{R}$ and a kernel $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{>0}$, denoted as $f \in \mathcal{GP}(m, \kappa)$ (Rasmussen and Williams 2006). Given a dataset $\mathbb{D}$, the prediction of f at a query point $\boldsymbol{x}$ is derived using Bayesian theory, resulting in $f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}|\mathbb{D}), \sigma^2(\boldsymbol{x}|\mathbb{D}))$. The posterior mean $\mu(\boldsymbol{x}|\mathbb{D})$ and posterior variance $\sigma^2(\boldsymbol{x}|\mathbb{D})$ are expressed as

$$\mu(\boldsymbol{x}|\mathbb{D}) = m(\boldsymbol{x}) + \boldsymbol{k}_{\mathbb{D}}^T(\boldsymbol{x})(\boldsymbol{K}_{\mathbb{D}} + \sigma_n^2 \boldsymbol{I}_N)^{-1}(\boldsymbol{y}_{\mathbb{D}} - \boldsymbol{m}_{\mathbb{D}}), \quad (1)$$

$$\sigma^2(\boldsymbol{x}|\mathbb{D}) = \kappa(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_{\mathbb{D}}^T(\boldsymbol{x})(\boldsymbol{K}_{\mathbb{D}} + \sigma_n^2 \boldsymbol{I}_N)^{-1}\boldsymbol{k}_{\mathbb{D}}(\boldsymbol{x}), \quad (2)$$

where $\boldsymbol{k}_{\mathbb{D}}(\boldsymbol{x}) = [\kappa(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \cdots, \kappa(\boldsymbol{x}, \boldsymbol{x}^{(|\mathbb{D}|)})]^T$, $\boldsymbol{K}_{\mathbb{D}} = [\kappa(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})]_{i,j=1,\cdots,|\mathbb{D}|}$, $\boldsymbol{m}_{\mathbb{D}} = [m(\boldsymbol{x}^{(1)}), \cdots, m(\boldsymbol{x}^{(|\mathbb{D}|)})]^T$, and $\boldsymbol{y}_{\mathbb{D}} = [y^{(1)}, \cdots, y^{(|\mathbb{D}|)}]^T$. Without loss of generality, each data pair $\left(\boldsymbol{x}^{(\iota)}, y^{(\iota)}\right)$ satisfies the following assumption.

Assumption 1. For each data pair $\left(\boldsymbol{x}^{(\iota)}, y^{(\iota)}\right)$ in the data set $\mathbb{D} = \left\{\left(\boldsymbol{x}^{(\iota)}, y^{(\iota)}\right)\right\}_{\iota=1,\cdots,|\mathbb{D}|}$ is sampled from the unknown f with noise, where $y^{(\iota)} = f(\boldsymbol{x}^{(\iota)}) + w^{(\iota)}$. The noise $w^{(\iota)}$ is bounded by $|w^{(\iota)}| \leq \sigma_n$ for $\forall \iota = 1, \cdots, |\mathbb{D}|$ and $\sigma_n \in \mathbb{R}_{\geq 0}$.

In a practice setting, noisy data is commonly assumed, but it is necessary to establish that the noise is bound in order to derive the error bounds.

Considering real-world scenarios where delayed measurement data processing or computation time consumption must be factored in, the distributed system can only obtain asynchronous predictions. Consequently, the estimated function $\hat{f}$ at current time $t \in \mathbb{R}_{\geq 0}$ aggregates the individual node prediction functions $\hat{f}_i$, which is formulated as follows

$$\hat{f}(\boldsymbol{x}(t)) = \sum_{i=1}^{M} \hat{f}_i\left(\boldsymbol{x}(t_i^k), \mathbb{D}_i(t_i^k)\right), \quad (3)$$

$$k := \text{argmax}\{s \in \mathbb{N} : t_i^s \leq t\} \quad (4)$$

where t_i^k indicates the time that the computational node i computes the k-th prediction of its GP model. The time interval Δt_i^k for obtaining prediction $\hat{f}_i(\boldsymbol{x}(t), t_i^k)$ spans the interval from t_i^k to t_i^{k+1}, encompassing both data processing time and prediction computation time as shown in Figure 1. Therefore, the discrete-time t_i^{k+1} is defined by $t_i^{k+1} = t_i^k + \Delta t_i^k$ with the initial time $t_i^0 = 0$ for $i = 1, \ldots, M$. The data processing time involves various components not limited to data transmission and storage. Moreover, the computation of individual predictions involves calculating the posterior mean $\mu(\boldsymbol{x}(t_i^k)|\mathbb{D}_i(t_i^k))$ and posterior variance $\sigma^2(\boldsymbol{x}(t_i^k)|\mathbb{D}_i(t_i^k))$, which are only available for calculation after time t_i^{k+1}.

However, the existing distributed learning approaches often simplify this asynchronous prediction by aggregating the latest available results from each GP model. The individual prediction function is

$$\hat{f}_i\big(\boldsymbol{x}(t_i^k), \mathbb{D}_i(t_i^k)\big) = \omega_i(\boldsymbol{x}(t_i^k))\mu_i(\boldsymbol{x}(t_i^k)), \tag{5}$$

where $\mu_i(\cdot)$ is the simplified notation for $\mu(\cdot|\mathbb{D}_i(t_i^k))$, and the $\omega_i(\boldsymbol{x}(t_i^k))$ is the aggregation weight determining the contribution of the posterior mean from the i-th GP model. For instance, in the MOE approach, $\omega_i(\boldsymbol{x}(t_i^k)) = 1/M$ when considering all models have no information related to the weights. In addition, the specific formulation of the aggregation weights for both POE family and BCM family approaches is presented as follows

$$\omega_i(\boldsymbol{x}(t_i^k)) = \rho_i\sigma_i^{-2}(\boldsymbol{x}(t_i^k))\tilde{\sigma}_i^2(\boldsymbol{x}(t_i^k)), \tag{6}$$

where $\sigma_i(\cdot)$ is defined as $\sigma(\cdot|\mathbb{D}_i(t_i^k))$ for notation simplification. The weighting factor ρ_i is defined as the difference of information entropy between the posterior and prior (Cao and Fleet 2015), but open for other options as long as $\rho_i \geq 0$. In POE family approaches, $\tilde{\sigma}_i$ is formulated by

$$\tilde{\sigma}_i^{-2}(\boldsymbol{x}(t_i^k)) = \sum_{s=1}^{M}\rho_s\sigma_s^{-2}(\boldsymbol{x}(t_i^k)),$$

and in BCM family is defined by

$$\tilde{\sigma}_i^{-2}(\boldsymbol{x}(t_i^k)) = \sum_{s=1}^{M}\rho_s\sigma_s^{-2}(\boldsymbol{x}(t_i^k)) + (1-\sum_{s=1}^{M}\rho_s)\sigma_*^{-2},$$

where σ_* is the prior variance of the unknown function f. Note that the asynchronous aggregation problem (5), where $\boldsymbol{x}(t_i^k)$ and $\boldsymbol{x}(t_j^k)$ may not be identical for $i, j = 1, \ldots, M$ and $t \in \mathbb{R}_{\geq 0}$, cannot be trivially addressed by conventional aggregation strategies such as MOE, POE, and BCM, since these methods only deal with the aggregation with same input $\boldsymbol{x}$. Moreover, the lack of prediction error bounds for these methods with different input $\boldsymbol{x}$ poses a significant challenge, especially in the context of real-time learning for safety-critical applications. Therefore, in the following section, we propose the asyncDGP, which not only accounts for the significance of previous predictions but also balances the contributions of the prior mean. Additionally, we delve into an analysis of the prediction error bounds considering the asynchronous effects.

AsyncDGP and GPgym Platform

To tackle the inherent challenges posed by asynchrony, we initiate our approach by introducing a criterion for evaluating prediction performance. This metric serves as a critical foundation for the subsequent development of the asyncDGP approach. Furthermore, we provide a rigorous analysis of the prediction error bounds associated with our proposed approach. Subsequently, we present the GPgym that facilitates testing of distributed real-time learning performance and verifies safety-critical applications.

Performance Criterion for Predictions

Before designing the prediction performance criterion, we propose prediction criteria for facilitating the design of AsyncDGP in the following lemmas.

Lemma 1 (Hashimoto et al. (2022))**.** *Suppose the function $f(\cdot)$ belongs to a reproduced kernel Hilbert space (RKHS) corresponding to κ with the bounded RKHS norm denoted as $\|f\|_\kappa = \sqrt{\langle f, f\rangle_\kappa} \leq \Gamma, \Gamma \in \mathbb{R}_{0,+}$ in the compact input domain $\mathbb{X}$. Then, the prediction error by using data set $\mathbb{D}$ satisfying Assumption 1 is bounded by*

$$|f(\boldsymbol{x}) - \mu(\boldsymbol{x}|\mathbb{D})| \leq \beta\sigma(\boldsymbol{x}|\mathbb{D}), \qquad \forall \boldsymbol{x} \in \mathbb{X}, \tag{7}$$

where $\beta = \sqrt{\Gamma^2 - \boldsymbol{y}_{\mathbb{D}}^T(\boldsymbol{K}_{\mathbb{D}} + \sigma_n^2\boldsymbol{I}_N)^{-1}\boldsymbol{y}_{\mathbb{D}} + |\mathbb{D}|}$.

Based on the above lemma providing the prediction error criterion $\beta\sigma(\boldsymbol{x}|\mathbb{D})$, now we propose the performance criterion η for asynchronous predictions as follows.

Lemma 2. *Assume the kernel κ is chosen as Lipschitz continuous with the Lipschitz constant $L_\kappa \in \mathbb{R}_{\geq 0}$ w.r.t a defined distance $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, i.e.,*

$$|\kappa(\boldsymbol{x}, \boldsymbol{x}') - \kappa(\boldsymbol{x}, \boldsymbol{x}'')| \leq L_\kappa d(\boldsymbol{x}' - \boldsymbol{x}'') \tag{8}$$

for $\forall \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'' \in \mathbb{R}^n$. Suppose the function $f(\cdot)$ belongs to an RKHS corresponding to κ with the bounded RKHS norm denoted as $\Gamma \in \mathbb{R}_{\geq 0}$ in the compact input domain $\mathbb{X}$. Moreover, let the corresponding RKHS norm be bounded by Γ. Then, the asynchronous prediction error between $f(\boldsymbol{x}(t))$ and $\mu_i(\boldsymbol{x}(t_i^k))$ using the data set $\mathbb{D}_i$ satisfying Assumption 1 is bounded by

$$|f(\boldsymbol{x}(t)) - \mu_i(\boldsymbol{x}(t_i^k))| \leq \eta_i^k(t), \tag{9}$$

where $\eta_i^k(t) = L_f d(\boldsymbol{x}(t_i^k), \boldsymbol{x}(t)) + \beta\sigma_i(\boldsymbol{x}(t_i^k))$ with $L_f = \sqrt{2L_\kappa}\Gamma$.

This lemma shows that the prediction performance criterion $\eta_i^{k_i}(t)$ is comprised of two components: the previous posterior variance denoted as $\sigma_i(\cdot)$ signifying the confidence associated with the prediction, and $d(\cdot,\cdot)$ reflecting the bias against the input point. Since $\eta_i^{k_i}(t)$ exhibits monotonically increasing behavior w.r.t both $\sigma_i(\cdot)$ and $d(\cdot,\cdot)$, predictions characterized by higher accuracy and proximity to the query point result in a diminished prediction error. It's worth mentioning that the deterministic prediction error bounds presented in Lemma 1 and 2 are variants derived from prior works, such as (Maddalena, Scharnhorst, and Jones 2021; Hashimoto et al. 2022). Additionally, probabilistic bounds can also be derived, as demonstrated in (Srinivas et al. 2012; Whitehouse, Ramdas, and Wu 2024). This indicates the versatility and extendibility of our method, as it can readily incorporate probabilistic bounds.

Considering Equation (8), the selection of the distance function d is pivotal, reflecting the divergence in the kernel space κ corresponding to the state space. In the absence of prior knowledge, a commonly employed kernel is the squared exponential (SE) kernel defined as $\kappa_{\mathrm{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2\exp(-\frac{1}{2\sigma_l^2}(\boldsymbol{x}-\boldsymbol{x}')^T(\boldsymbol{x}-\boldsymbol{x}'))$, where $\sigma_f \in \mathbb{R}_{>0}$ and $\sigma_l \in \mathbb{R}_{>0}$ are hyper-parameters.

Lemma 3. *Consider the SE kernel and its distance function defined as* $d_{SE}(\boldsymbol{x},\boldsymbol{x}') = \|\boldsymbol{x}-\boldsymbol{x}'\|$, $\forall \boldsymbol{x},\boldsymbol{x}' \in \mathbb{R}^n$*, then the corresponding Lipschitz constant is* $L_{\kappa,SE} = \sigma_f^2 \exp(-0.5)/\sigma_l$*.*

In order to have an independent length scale for each dimension, automatic relevance determination (ARD) squared exponential (SE) kernel can be considered, which is defined

$$\kappa_{\text{ARD-SE}}(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2 \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T \boldsymbol{\Sigma}_L^{-2}(\boldsymbol{x}-\boldsymbol{x}')),$$

where $\boldsymbol{\Sigma}_L = \operatorname{diag}(l_1,\cdots,l_n)$ and $l_i \in \mathbb{R}+$ for $i = 1,\cdots,M$ associated with each dimension in the state space, facilitate dimension-wise distance. This choice of kernels and their parameterization aligns to capture the inherent relationships within the data, while the ARD-SE kernel provides additional flexibility in addressing varying length scales across dimensions.

Answers of *Question 1 and 2*: Design of AsyncDGP

By quantifying the prediction performance as the aggregating criteria in Lemma 1 and 2, we are able to utilize previous predictions from the GP models, which answers *Question 1*. That is to say, the earlier temporal predictions are valuable based on the state rather than the time. In order to leverage the previous prediction results, we define the collected prediction information set $\mathfrak{I}(t)$ defined as

$$\mathfrak{I}(t) = \bigcup_{i=1}^{M} \Big\{ \big(\boldsymbol{x}(t_i^k), \mu_i(\boldsymbol{x}(t_i^k)), \sigma_i(\boldsymbol{x}(t_i^k))\big) \Big| t_i^k \leq t, \forall k \in \mathbb{N} \Big\}. \tag{10}$$

With given information set $\mathfrak{I}(t)$, we present our asynchronous aggregation approach incorporated with posterior predictions and prior means of DGP models as

$$\hat{f}(\boldsymbol{x}(t)) = \sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \omega_k^i(t) \mu_i(\boldsymbol{x}(t_i^k)) + \omega_m(t) m(\boldsymbol{x}(t)), \tag{11}$$

with $\bar{k}_i(t) = \sup\{k \in \mathbb{N} | t_i^k \leq t\}$, which address the *Question 2*. The core of AsyncDGP is to filter out inferior predictions in the information set $\mathfrak{I}(t)$ according to the proposed prediction criteria. With the proposed performance criterion η, the aggregation weights $\omega_k^i(t)$ for the posterior mean are designed as

$$\omega_k^i(t) = \omega^2(t) \rho_k^i(t) (\eta_i^k(t))^{-2}, \tag{12}$$

where

$$\rho_k^i(t) = \max\{\log(\beta\sigma_f / \eta_i^k(t)), 0\}, \tag{13}$$

$$\omega^{-2}(t) = \sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \rho_k^i(t) (\eta_i^k(t))^{-2} + \Big(1 - \sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \rho_k^i(t)\Big) (\beta\sigma_f)^{-2}. \tag{14}$$

Here, the weighting factor $\rho_k^i(\cdot) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ functions as a selection criterion. Specifically, when the discrepancy in information entropy between the weighted prior variance $\beta\sigma_f$ and the proposed criterion $\eta_i^k(t)$ falls below a threshold of 1, the posterior mean is excluded from selection. Therefore, to take account of the prior mean, the weighting factor $\omega_m(t)$ is designed as

$$\omega_m(t) = \omega^2(t) \Big(1 - \sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \rho_k^i(t)\Big) (\beta\sigma_f)^{-2}. \tag{15}$$

Notably, the results of the prior mean at time t in (11) are unaffected by delays, because of its exclusion from learning and training processes. Particularly in instances of lacking prior knowledge of the unknown function, the prior mean is set to zero, indicating its result is readily available without latency. Moreover, the prior mean can be regarded as the posterior mean with an empty data set. Thus, with the results in Lemma 1 and the expression of $\sigma_i(\cdot)$ in (1), the weighted prior variance $\beta\sigma_f$ is used for evaluating the error bound between true function and prior mean function, i.e.,

$$|f(\boldsymbol{x}(t)) - m(\boldsymbol{x}(t))| \leq \beta\sigma_f. \tag{16}$$

With the accuracy evaluation of prior mean, the variable $\rho_k^i(t)$ is set such that only the predictions $\mu_i(\boldsymbol{x}(t_i^k))$ with better performance than $m(\boldsymbol{x}(t))$ will be aggregated with $\rho_k^i(t) > 0$.

Remark 1. Our aggregation method (11) reveals that the most useful prediction is one that is closest to the query state $\boldsymbol{x}(t)$, rather than the most temporally recent prediction at time t assuming both have the same prediction error bound from GP. Once the prediction for $\boldsymbol{x}(t_i^k)$ is completed, the result is promptly sent and stored based on the collected prediction information set. Moreover, there is no need to recalculate the posterior mean at the state point $\boldsymbol{x}(t_i^k)$, which can be reused for future calculation.

Remark 2. Notably, there are no upper bound for $\rho_k^i(t)$, especially considering the fact that $\rho_k^i(t) \to \infty$ when $\eta_i^k(t) \to 0$ in (13), the second term in (14) could be negative. However, considering only predictions with $\eta_i^k(t) \leq \beta\sigma_f$ have non-zero $\rho_k^i(t)$, it has

$$\sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \rho_k^i(t) (\eta_i^k(t))^{-2} \geq \sum_{i=1}^{M} \sum_{k=0}^{\bar{k}_i(t)} \rho_k^i(t) (\beta\sigma_f)^{-2}, \tag{17}$$

such that a valid $\omega(t)$ with $\omega^{-2}(t) \geq (\beta\sigma_f)^{-2}$ exists. Moreover, it is direct to see $\omega(t) \in (0, \beta\sigma_f]$.

Since the information set $\mathfrak{I}(t)$ expands over time, the associated data storage and computational requirements for the aggregation operation also grow proportionally. To circumvent these challenges and manage the finite information set practically, a well-defined constant $\bar{\mathfrak{I}} \in \mathbb{N}_{>0}$ is introduced, such that $|\mathfrak{I}(t)| \leq \bar{\mathfrak{I}}$ holds for all $t \in \mathbb{R}_{\geq 0}$ with memory management strategy. As the AsyncDGP only uses the prediction results, the aggregation algorithm only takes $\mathcal{O}(1)$ for sum and add operations. However, the computational complexity of the calculation of the results depends on the local GP model. For instance, utilizing the state-of-the-art LoG-GP for online learning approach, the update of a

```
Algorithm 1: AsynGP
while Prediction not finished do
    Get current state $\boldsymbol{x}(t)$ and send to all GP experts;
    Part I: Pre-processing
    Calculate $\eta_i^k(t)$ for each element in $\mathfrak{I}(t)$
    Delete the elements with $\eta_i^k(t) > \beta\sigma_f$ from $\mathfrak{I}(t)$
    Part II: New Predictions Reception
    for GP expert from $i = 1$ to $M$ do
        $\mathbb{P}_i^k \leftarrow$ receive prediction result from expert $i$
        if $\mathbb{P}_i^k \in \emptyset$ then
            Next $i$
        end if
        Calculate $\eta_i^k(t)$ for newly received prediction
        if $\eta_i^k(t) \leq \beta\sigma_f$ AND $|\mathfrak{I}(t)| < \bar{\mathfrak{I}}$ then
            $\mathfrak{I}(t) \leftarrow \{\mathfrak{I}(t), \mathbb{P}_i^k\}$
        else if $\eta_i^k(t) \leq \beta\sigma_f$ AND $|\mathfrak{I}(t)| = \bar{\mathfrak{I}}$ then
            $\mathfrak{I}(t) \leftarrow \bar{\mathfrak{I}}$ predictions with lowest $\eta_i^k(t)$
        end if
    end for
    Part III: Asynchronous Aggregation
    $\hat{f}(\boldsymbol{x}(t)) \leftarrow$ Aggregation using (11)
end while
```

LoG-GP model requires $\mathcal{O}_p(\log(N))$ and the mean and variance predictions require $\mathcal{O}_p(\log^2(N))$. Specifically, given the structure of $\rho_k^i(t)$, which permits only predictions with $\eta_i^k(t) \leq \beta\sigma_f$ to participate in the aggregation, a heuristic algorithm is designed as in Algorithm 1. This algorithm comprises three main parts: pre-processing, new prediction reception, and asynchronous aggregation.

- In the pre-processing part, the condition $\eta_i^k(t) \leq \beta\sigma_f$ filters each element in $\mathfrak{I}(t)$, i.e., the elements violate this condition will be removed from $\mathfrak{I}(t)$.
- In the prediction reception part, the centralized node will check whether any new prediction is available from expert i. According to the definition of $\mathfrak{I}(t)$ in (10), each GP expert will transmit $\mathbb{P}_i^k = \{\mu_i(\boldsymbol{x}(t_i^k)), \sigma_i(\boldsymbol{x}(t_i^k)), \boldsymbol{x}(t_i^k)\}$ to the centralized node. After receiving the new non-empty predictions, the condition $\eta_i^k(t) \leq \beta\sigma_f$ will be checked, and only maximal $\bar{\mathfrak{I}}$ predictions including the new one and old predictions will be left in $\mathfrak{I}(t)$.
- In the aggregation part, the prediction $\hat{f}(\boldsymbol{x}(t))$ is generated using (11). Therefore, if no element in $\mathfrak{I}(t)$, the prior mean $m(\boldsymbol{x}(t))$ is used, i.e., $\hat{f}(\boldsymbol{x}(t)) = m(\boldsymbol{x}(t))$.

Prediction Performance Guarantee

The proposed asynchronous aggregation strategy (11) combining with the information set management in Algorithm 1 inherits the prediction error bound from exact GP as in Lemma 1, which is shown as follows.

Theorem 1. *Consider M GP models generated predictions with different data sets $\mathbb{D}_i, \forall i = 1, \cdots, M$ satisfying Assumption 1. Choose Lipschitz kernel κ, and assume f has the bounded RKHS norm Γ corresponding to κ. The predictions are aggregated according to* (11) *with the information set management following Algorithm 1. Then, the prediction error is bounded by*

$$|f(\boldsymbol{x}(t)) - \hat{f}(\boldsymbol{x}(t))| \leq \omega(t), \ \forall t \in \mathbb{R}_{\geq 0}. \qquad (18)$$

This theorem shows the prediction error bound is monotonically increasing w.r.t $\eta_i^k(t)$ considering the definition of (14). This indicates aggregation of more accurate predictions, i.e., small prediction error $\sigma_i(\cdot)$ and close distance $d(\cdot,\cdot)$ benefit the aggregated prediction performance. To guarantee an error prediction bound, which is independent of time, we propose the following corollary.

Corollary 1. *Consider M GP models generated predictions with different data sets $\mathbb{D}_i, \forall i = 1, \cdots, M$ satisfying Assumption 1. Choose Lipschitz kernel κ, and assume f has the bounded RKHS norm Γ corresponding to κ. The predictions are aggregated according to* (11) *with the information set management following Algorithm 1. Then, the prediction error is bounded by*

$$|f(\boldsymbol{x}(t)) - \hat{f}(\boldsymbol{x}(t))| \leq \beta\sigma_f, \ \forall t \in \mathbb{R}_{\geq 0} \qquad (19)$$

The proposed AsyncDGP and the established prediction error bounds can be effectively leveraged in safety-critical applications, such as safe control tasks. To validate the algorithm and its implementation, it is essential to develop a real application, which is elucidated in the subsequent section.

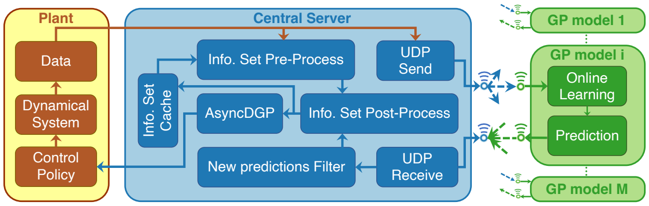

Figure 2: The framework of GPgym.

GPgym

For analyzing the learning performance and safety-critical applications of dynamical systems characterized by asynchronous phenomena, we develop an online learning platform, namely GPgym[2], whose framework is illustrated in Figure 2. This platform serves as a tool for validating the effectiveness of a designed algorithm and facilitating robust testing under asynchronous conditions. Furthermore, OL-GPgym provides a means to test the maximum capability of designed algorithms for distributed GPs and iteratively improve the algorithms, which ensures their effectiveness in real-world scenarios. The GP model module (green block) functions as the distributed component within the GPgym framework. This module is responsible for processing and learning from data received from the central server (blue block) via User Datagram Protocol (UDP) communication. After the prediction is calculated, the GP model transmits its

[2] The GPgym platform, including data set, code, and instructions, is provided in (Dai and Yang 2024).

results back to the central server. The central server employs the AsyncDGP algorithm to process data originating from the plant (yellow block), which constitutes the dynamic system. The control policy is designed to govern the system's behavior, while the system states are utilized as training data. This framework establishes a feedback loop that continuously updates the control input in response to the evolving dynamics of the system. Moreover, the data required for this process can be obtained through sensors integrated into the system. This structured flow of information within the GP-gym platform illustrates the seamless interaction between the GP model, the central server, and the plant.

Safe Online Learning-based Control

We illustrate the incorporation of AsyncDGP into a learning-based control framework, showcasing its effectiveness in employing to dynamical systems. Details regarding the controller design and implementation algorithm are provided, then a rigorous analysis of control performance using AsyncDGP is presented.

Control Law Design with AsyncDGP

We demonstrate the application of asyncDGP in learning-based control, focusing on a dynamical system depicted in the plant (refer to Figure 2), wherein data from the system is continuously collected during closed-loop control through sensors. Specifically, we consider a nonlinear control affine system governed by

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \cdots, \quad \dot{x}_n = f(\boldsymbol{x}) + u, \tag{20}$$

where the system state is denoted as $\boldsymbol{x} = [x_1, \cdots, x_n]^T \in \mathbb{R}^n$, and the single control input as $u \in \mathbb{R}$. While the system structure is known, the function f, encompassing factors such as environmental uncertainties and unmodeled components, is assumed to be unknown. Notably, the high-order form (20) encompasses a wide range of dynamical systems, such as robotic manipulators (Spong, Hutchinson, and Vidyasagar 2020), underwater vehicles (Fossen 2011), chemical processes (Subramanian 2021), etc.

The objective is to devise a control policy steering the system towards a predefined time-dependent reference $x_r(t)$, which follows an assumption that $x_r(t)$ is n times continuously differentiable with all its derivatives bounded. Formally, the tracking error, denoted as $\boldsymbol{e} = [e_1, \cdots, e_n]^T = \boldsymbol{x} - \boldsymbol{x}_d$, where $\boldsymbol{x}_d = [x_r, \dot{x}_r, \cdots, \frac{\mathrm{d}^{n-1}}{\mathrm{d}t^{n-1}} x_r]^T$, should be minimized and bounded to ensure the guaranteed control performance and stability. In other words, the control goal is that the tracking error converges to the neighborhood of zero. By employing the proposed AsyncDGP, we design the control law as follows

$$u(t) = \frac{\mathrm{d}^n}{\mathrm{d}t^n} x_r(t) - \hat{f}(\boldsymbol{x}(t)) - \sum_{i=1}^{n} \lambda_i \Big(x_i - \frac{\mathrm{d}^{i-1} x_r(t)}{\mathrm{d}t^{i-1}} \Big), \tag{21}$$

where the control gains $\lambda_1, \cdots, \lambda_n \in \mathbb{R}_{>0}$ are chosen, such that all the eigenvalues of matrix $\boldsymbol{A}$ defined as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{(n-1)\times 1} & \boldsymbol{I}_{n-1} \\ -\lambda_1 & -[\lambda_2, \cdots, \lambda_n] \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{22}$$

are negative real numbers. Notably, the controllability induced by the structure in (20) ensures the existence of $\lambda_1, \cdots, \lambda_n$ for any desired eigenvalues of $\boldsymbol{A}$.

Control Performance Analysis

In order to analyze the stability, we first establish the error dynamics, which is written as

$$\dot{\boldsymbol{e}}(t) = \boldsymbol{A}\boldsymbol{e}(t) + \boldsymbol{b}(f(\boldsymbol{x}(t)) - \hat{f}(\boldsymbol{x}(t))), \tag{23}$$

where $\boldsymbol{b} = [\boldsymbol{0}_{1\times(n-1)}, 1]^T$. Building upon the presented error dynamics (23), we rigorously demonstrate the existence of an ultimate upper bound (Khalil 2002) for the norm of tracking error $\|\boldsymbol{e}\|$. This upper bound is contingent upon both the control gains and the efficacy of the learning process, which is shown in the following theorem.

Theorem 2. *Consider a system* (20) *controlled by* (21) *for tracking tasks. The compensation $\hat{f}(\boldsymbol{x}(t))$ is obtained by asynchronous aggregation* (11) *with Algorithm 1 using the GP experts satisfying Assumption 1 and Lipschitz continuous kernel. The tracking error is ultimately bounded by*

$$\lim_{t\to\infty} \boldsymbol{e}(t) \le \|\boldsymbol{Q}\| \|\boldsymbol{Q}^{-1}\| |\bar{\Lambda}|^{-1} \bar{\omega}, \tag{24}$$

where $\bar{\Lambda}$ denotes the maximal value of the eigenvalue of $\boldsymbol{A}$, $\boldsymbol{Q} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n]$ with $\boldsymbol{v}_i, \forall i = 1, \cdots, n$ be the eigenvector of $\boldsymbol{A}$ and $\bar{\omega} = \max_{t\in\mathbb{R}_{\geq 0}} \omega(t)$.

It is evident that the bound on tracking error is directly proportional to the learning error bound $\bar{\omega}$, and inversely proportional to the control gains reflected by $|\bar{\Lambda}|$. Therefore, this theorem signifies the feasibility of determining adaptive control gains λ_i, that result in a diminished tracking error, which is crucial in safety critical applications.

Simulation Results

In the subsections, we demonstrate the effectiveness of the proposed AsyncDGP through its application to regression tasks on real-world datasets, alongside a control task for a common dynamical system. Note that each GP model receives streaming data after the prediction process is finished, leading to variations in their training data and predictions. However, the real-time streaming data comes from the same dataset. Comparative analyses are conducted, evaluating the performance of AsyncDGP against various state-of-the-art DGP methods.

Regression Benchmark

The regression performance on the three datasets, namely KIN40K (8-dimensional input, 10K data), SARCOS (21-dimensional input, 44484 data), and PUMADYN32NM (32-dimensional input, 7168 data) are evaluated. We consider a distributed system to have 4 GP models, where each model is built using the start-of-the-art online GP algorithm LoG-GP (Lederer et al. 2021) and connected through UDP via Wi-Fi. The standardized mean squared errors (SMSE) for regression, with the information set threshold $\bar{\mathfrak{I}}$ set to 4 and 10, are shown in Figure 3, respectively. In general, AsyncDGP exhibits superior performance compared to MOE, POE, and

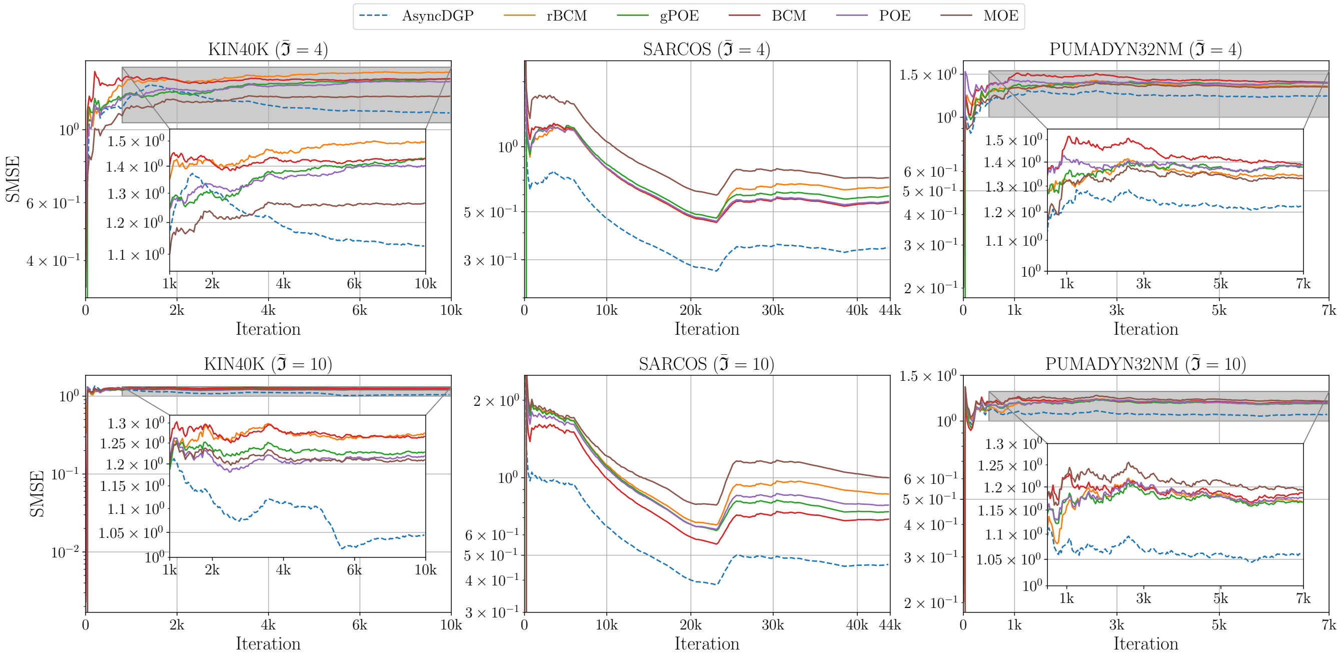

Figure 3: Regression performance on 3 datasets.

BCM, with the exception observed in the KIN40K dataset for $\bar{\Im} = 4$ before 3.5k iterations. However, AsyncDGP demonstrates enhanced performance throughout all iterations when $\bar{\Im} = 10$, suggesting that increasing the information set leads to improved prediction accuracy. Similarly, in the PUMADYN32NM dataset, all methods demonstrate improvement when $\bar{\Im} = 10$. In the SARCOS dataset, enlarging the information set does not yield a corresponding enhancement in prediction performance, thereby illustrating AsyncDGP with $\bar{\Im} = 4$ as a better option.

Control Task

By employing the proposed learning-based control defined in (21) using the DGP composed of 4 models with $\bar{\Im} = 20$, we evaluate the control performance of the dynamical system described as $\dot{x}_1 = x_2, \;\; \dot{x}_2 = f(\boldsymbol{x}) + u$, where

$$f(\boldsymbol{x}) = 1 + \frac{x_1 x_2}{10} + \frac{\cos(x_2)}{2} - 10\sin(5x_1) + \frac{1}{2(1+\exp(-\frac{x_2}{10}))}.$$

Furthermore, the desired reference is chosen with the form $x_r(t) = a_r \sin(w_r t)$, with coefficients $a_r, w_r \in \mathbb{R}$. The control gains are set to $\lambda_1 = 2$ and $\lambda_2 = 10$ and $\bar{\Im} = 20$. The minimum norm values of tracking errors and prediction errors are shown in Figure 4. Moreover, with $\Gamma = 1$ and the maximal data size is 100, the GP error bound is 35.5 and the tracking error bound is 4.9, which are both valid in the simulation results. Notably, both the mean and median values of AsyncDGP consistently outperform alternative approaches, which have similar and consistently inferior performance.

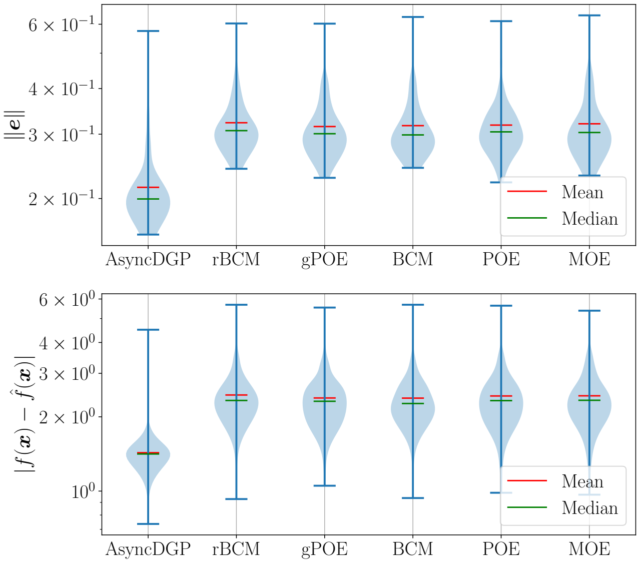

Figure 4: The violin plots of the minimum values over 100 Monte-Carlo simulations for tracking error (top) and prediction error (bottom).

Conclusion

In this paper, we introduce the AsyncDGP algorithm designed to enhance prediction accuracy in real-time distributed learning with GPs considering asynchronous scenarios. The superior performance of AsyncDGP in regression tasks reveals the inadequacy of aggregating only the latest available predictions. The incorporation of a guaranteed prediction error bound contributes to establishing the viability of employing AsyncDGP for addressing safety critical applications, as demonstrated through the control task of dynamical systems in the proposed GPgym platform.

Acknowledgments

This work has been financially supported by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." under joint project 6G-life with project 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.

References

Cao, Y.; and Fleet, D. J. 2015. Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions.

Chen, J.; and Ran, X. 2019. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, 107(8): 1655–1674.

Cohen, S.; Mbuvha, R.; Marwala, T.; and Deisenroth, M. 2020. Healing Products of Gaussian Process Experts. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 2068–2077. PMLR.

Dai, X.; Lederer, A.; Yang, Z.; and Hirche, S. 2023. Can Learning Deteriorate Control? Analyzing Computational Delays in Gaussian Process-Based Event-Triggered Online Learning. In *Learning for Dynamics and Control Conference*, 445–457. PMLR.

Dai, X.; and Yang, Z. 2024. GPgym: A Remote Service Platform with Gaussian Process Regression for Online Learning. arXiv:2412.13276.

Dai, X.; Yang, Z.; Xu, M.; Zhang, S.; Liu, F.; Hattab, G.; and Hirche, S. 2024a. Decentralized event-triggered online learning for safe consensus control of multi-agent systems with Gaussian process regression. *European Journal of Control*, 80: 101058.

Dai, X.; Yang, Z.; Zhang, S.; Zhai, D.-H.; Xia, Y.; and Hirche, S. 2024b. Cooperative Online Learning for Multiagent System Control via Gaussian Processes With Event-Triggered Mechanism. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Deisenroth, M.; and Ng, J. W. 2015. Distributed Gaussian Processes. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1481–1490. Lille, France: PMLR.

Egelé, R.; Guyon, I.; Vishwanath, V.; and Balaprakash, P. 2023. Asynchronous Decentralized Bayesian Optimization for Large Scale Hyperparameter Optimization. In *2023 IEEE 19th International Conference on e-Science (e-Science)*, 1–10.

Fossen, T. I. 2011. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons.

Hashimoto, K.; Saoud, A.; Kishida, M.; Ushio, T.; and Dimarogonas, D. V. 2022. Learning-based Symbolic Abstractions for Nonlinear Control Systems. *Automatica*, 146: 110646.

Khalil, H. K. 2002. *Control of nonlinear systems*. Prentice Hall, New York, NY.

Lederer, A.; Conejo, A. J. O.; Maier, K. A.; Xiao, W.; Umlauft, J.; and Hirche, S. 2021. Gaussian process-based real-time learning for safety critical applications. In *International Conference on Machine Learning*, 6055–6064. PMLR.

Lederer, A.; Yang, Z.; Jiao, J.; and Hirche, S. 2023. Cooperative Control of Uncertain Multiagent Systems via Distributed Gaussian Processes. *IEEE Transactions on Automatic Control*, 68(5): 3091–3098.

Liu, H.; Cai, J.; Wang, Y.; and Ong, Y. S. 2018. Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In *International Conference on Machine Learning*, 3131–3140. PMLR.

Maddalena, E. T.; Scharnhorst, P.; and Jones, C. N. 2021. Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*, 134: 109896.

Masoudnia, S.; and Ebrahimpour, R. 2014. Mixture of Experts: A Literature Survey. 42(2): 275–293.

Ng, J. W.; and Deisenroth, M. P. 2014. Hierarchical Mixture-of-Experts Model for Large-Scale Gaussian Process Regression. arXiv:1412.3078.

Nguyen, J.; Malik, K.; Zhan, H.; Yousefpour, A.; Rabbat, M.; Malek, M.; and Huba, D. 2022. Federated Learning with Buffered Asynchronous Aggregation . In Camps-Valls, G.; Ruiz, F. J. R.; and Valera, I., eds., *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 3581–3607. PMLR.

Park, T.; and Saad, W. 2019. Distributed learning for low latency machine type communication in a massive Internet of Things. *IEEE Internet of Things Journal*, 6(3): 5562–5576.

Peng, H.; Zhe, S.; Zhang, X.; and Qi, Y. 2017. Asynchronous distributed variational Gaussian process for regression. In *International Conference on Machine Learning*, 2788–2797. PMLR.

Rasmussen, C. E.; and Williams, C. K. I. 2006. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. ISBN 978-0-262-18253-9.

Spong, M. W.; Hutchinson, S.; and Vidyasagar, M. 2020. *Robot modeling and control*. John Wiley & Sons.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. W. 2012. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5): 3250–3265.

Subramanian, G. 2021. *Process control, intensification, and digitalisation in continuous biomanufacturing*. John Wiley & Sons.

Tresp, V. 2000a. A Bayesian committee machine. *Neural computation*, 12(11): 2719–2741.

Tresp, V. 2000b. Mixtures of Gaussian Processes. In Leen, T.; Dietterich, T.; and Tresp, V., eds., *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Whitehouse, J.; Ramdas, A.; and Wu, S. Z. 2024. On the sublinear regret of GP-UCB. *Advances in Neural Information Processing Systems*, 36.

Yan, Z.; Yang, Z.; Pan, X.; Zhou, J.; and Wu, D. 2020. Virtual leader based path tracking control for Multi-UUV considering sampled-data delays and packet losses. *Ocean Engineering*, 216: 108065.

Yang, Z.; Dai, X.; Dubey, A.; Hirche, S.; and Hattab, G. 2024a. Whom to Trust? Elective Learning for Distributed Gaussian Process Regression. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 2020–2028.

Yang, Z.; Dai, X.; and Hirche, S. 2024. Asynchronous Distributed Gaussian Process Regression for Online Learning and Dynamical Systems: Complementary Document. arXiv:2412.11950.

Yang, Z.; Dong, S.; Lederer, A.; Dai, X.; Chen, S.; Sosnowski, S.; Hattab, G.; and Hirche, S. 2024b. Cooperative Learning with Gaussian Processes for Euler-Lagrange Systems Tracking Control Under Switching Topologies. In *2024 American Control Conference (ACC)*, 560–567.

Yang, Z.; Sosnowski, S.; Liu, Q.; Jiao, J.; Lederer, A.; and Hirche, S. 2021. Distributed Learning Consensus Control for Unknown Nonlinear Multi-Agent Systems Based on Gaussian Processes. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 4406–4411. IEEE.

Yuan, C.; and Neubauer, C. 2008. Variational mixture of Gaussian process experts. *Advances in neural information processing systems*, 21.