






# Automatic Nonlinear MPC Approximation With Closed-Loop Guarantees

Abdullah Tokmak , Christian Fiedler , Melanie N. Zeilinger , *Member, IEEE*,  
Sebastian Trimpe , *Member, IEEE*, and Johannes Köhler , *Member, IEEE*

**Abstract**—Safety guarantees are vital in many control applications, such as robotics. Model predictive control (MPC) provides a constructive framework for controlling safety-critical systems but is limited by its computational complexity. We address this problem by presenting a novel algorithm that automatically computes an explicit approximation to nonlinear MPC schemes while retaining closed-loop guarantees. Specifically, the problem can be reduced to a function approximation problem, which we tackle by proposing the Adaptive and Localized Kernel Interpolation Algorithm with eXtrapolated reproducing kernel Hilbert space norm, which we refer to as ALKIA-X. ALKIA-X is a noniterative algorithm that ensures well-conditioned computations, a fast-to-evaluate approximating function, and the guaranteed satisfaction of any desired bound on the approximation error. Hence, ALKIA-X automatically computes an explicit function that approximates the MPC, yielding a controller suitable for safety-critical systems and high sampling rates. We apply ALKIA-X to approximate two nonlinear MPC schemes, demonstrating reduced computational demand and applicability to realistic problems.

**Index Terms**—Constrained control, kernel-based function approximation, machine learning, nonlinear (NL) predictive control.

## I. INTRODUCTION

MODEL predictive control (MPC) [1] is an optimization-based control method for nonlinear constrained systems, in which the applied control input is implicitly defined by the solution of a nonlinear program (NLP). Hence, online appli-

Received 11 April 2024; revised 29 November 2024; accepted 23 March 2025. Date of publication 8 April 2025; date of current version 29 September 2025. This work was supported in part by the Swiss National Science Foundation through NCCR Automation under Grant 51NF40 180545, in part by the IDEA League, and in part by the German Academic Scholarship Foundation. Recommended by Associate Editor C. N. Jones. (Corresponding author: Abdullah Tokmak.)

Abdullah Tokmak is with the Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland, also with the Institute for Data Science in Mechanical Engineering, RWTH Aachen University, 52068 Aachen, Germany, and also with the Institute for Dynamic Systems and Control, ETH Zürich, CH-8092 Zürich, Switzerland (e-mail: abdullah.tokmak@aalto.fi).

Christian Fiedler and Sebastian Trimpe are with the Institute for Data Science in Mechanical Engineering, RWTH Aachen University, 52068 Aachen, Germany (e-mail: christian.fiedler@dsme.rwth-aachen.de; trimpe@dsme.rwth-aachen.de).

Melanie N. Zeilinger and Johannes Köhler are with the Institute for Dynamic Systems and Control, ETH Zürich, CH-8092 Zürich, Switzerland (e-mail: mzeilinger@ethz.ch; jkoehle@ethz.ch).

Digital Object Identifier 10.1109/TAC.2025.3558871

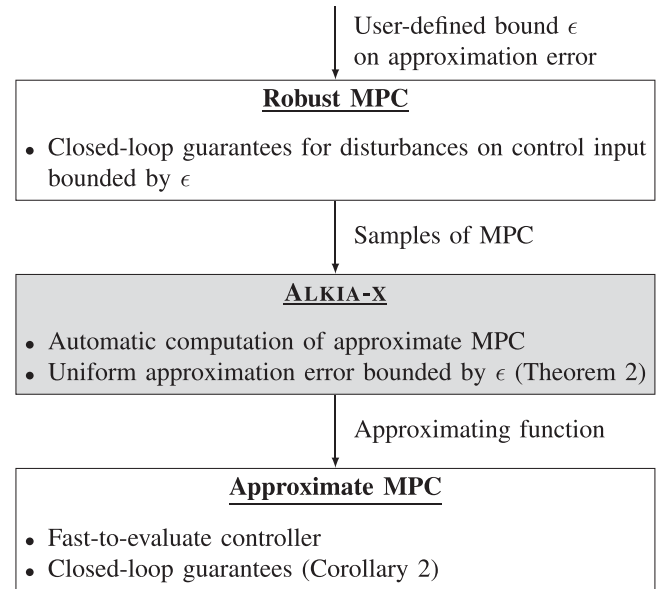


Fig. 1. High-level illustration of the proposed framework for approximate MPC with closed-loop guarantees. ALKIA-X automatically computes an explicit function that approximates a robust MPC scheme up to any specified accuracy  $\epsilon$ , which yields a fast-to-evaluate approximate MPC with closed-loop guarantees. Notably, the proposed process is *noniterative* due to the uniform error bounds inherently guaranteed by ALKIA-X.

cation of MPC under real-time requirements is challenging for fast applications, such as robotics. This has motivated extensive research on approximate MPC schemes [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] to obtain controllers suitable for high sampling rates. Specifically, these approaches approximate the MPC offline, i.e., prior to applying the controller. This offline approximation should ideally be automatic to simplify its application. Furthermore, the approximate MPC should retain closed-loop guarantees regarding stability and constraint satisfaction to allow for application on safety-critical systems.

Fig. 1 outlines the proposed approach. We consider a *robust* MPC formulation [20], [21], [22], [23], which is designed such that the desired closed-loop guarantees remain valid under input disturbances below a user-chosen error bound  $\epsilon > 0$ . By approximating the feedback law implicitly defined by this MPC up to a tolerance  $\epsilon$ , the approximate MPC preserves all control-theoretic guarantees induced by the MPC. Hence, approximating the MPC can be cast as a function approximation problem by sampling state and corresponding optimal inputs obtained by solving the

MPC offline. To address this function approximation problem, we propose the **Adaptive and Localized Kernel Interpolation Algorithm** with **eXtrapolated** reproducing kernel Hilbert space (RKHS) norm, which we refer to as **ALKIA-X**. ALKIA-X automatically computes an explicit function that approximates the MPC with a uniform approximation error  $\epsilon$ , resulting in a cheap-to-evaluate approximate MPC with guarantees on stability and constraint satisfaction.

*Related Work:* Linear MPC schemes can be exactly characterized as an explicit piecewise affine function over a polyhedral partition of the domain, which is known as explicit MPC [24], [25]. However, this exact solution tends to suffer from scalability issues, which has motivated research on approximate explicit MPC schemes. In [26], the domain is partitioned into hypercubes and the MPC is approximated using piecewise affine functions. Notably, [26, Lemma 1] ensures that the approximate MPC respects the constraints by using a constrained optimization problem that leverages convexity. A similar approximate MPC is suggested in [27], using a more general continuous interpolation on each hypercube. However, these approaches primarily rely on convexity, and hence cannot be applied to nonlinear MPC schemes.

An overview of explicit MPC for nonlinear systems is given in [28]. A very appealing approach is to treat the MPC as an unknown function and use supervised machine learning techniques to obtain an explicit approximation. Approximating MPC schemes with neural networks (NNs) is a particularly popular approach [2], [3], [4], [5], [6], [7], [8], [9], [10], which was first suggested in [11]. However, it is difficult to ensure that the resulting NN provides desirable closed-loop properties, see the recent review paper [12]. One solution to this problem is adding a safety filter to change the output of the NN [2], [3], [4], [5], [6]. For linear systems, this can be achieved with a projection on a robust control invariant set [2], [3] or by using additional active set iterations [4]. For general nonlinear systems, an implicit safety filter can be obtained by validating the safety of the NN policy online and using an MPC-intrinsic fallback in the case of predicted constraint violations [5], see also [6]. Other solutions include a posteriori safety validations of the NN using statistical validation [7].

A constructive approach to obtain closed-loop guarantees under approximation errors is to leverage *robust* MPC design methods [20], [21], [22], [23]. For instance, references [8], [9], and [10] first design an MPC scheme that is robust with respect to input disturbances bounded by  $\epsilon$  and then use NNs to approximate this MPC with an approximation error smaller than  $\epsilon$ . However, guaranteeing the desired approximation error bound for NNs is challenging. In [8] and [9], this issue is addressed by using statistical tools to validate the approximation accuracy with high probability. Nevertheless, this separate validation yields a nontrivial offline design consisting of multiple iterations between training the NN and validating its approximation.

A natural solution to this problem is nonparametric regression tools, such as kernel-based methods [18], [19], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43] or set-membership identification [13], [14], [15], [16], [44], [45], which both provide error bounds. Nonparametric set-membership estimation typically assumes Lipschitz continuity and builds a nonfalsified piecewise affine set that contains the ground truth, which is also called Kinky inference [44] or Lipschitz interpolation [45]. Applications of such tools to approximate nonlinear MPC schemes are explored in [13], [14], [15],

and [16]. Similarly, in [17], a known Lipschitz bound is utilized to approximate an MPC with guaranteed error bound  $\epsilon$ , using instead a quasi-interpolation approach. Kernel-based methods, such as kernel ridge regression (KRR) [29], [30], [31], [32], [33] or Gaussian process (GP) regression [18], [19], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], enable smooth approximations and error bounds, assuming that the ground truth lies in the corresponding RKHS. Applications of GPs to approximate MPC schemes are, e.g., explored in [18] and [19], where closed-loop guarantees on the approximate MPC are provided using a posteriori sampling, similar to [8] and [9]. These nonparametric estimation methods can in principle achieve any desired approximation accuracy  $\epsilon$ ; however, they are in general not computationally cheap-to-evaluate. In particular, the complexity of evaluating kernel-based approximations scales cubically in the number of samples [34, Sect. 2.3], making it challenging to use for control applications. For kernel-based methods, computationally cheap online evaluation can, e.g., be ensured by only considering nearest-neighbor samples for the online evaluation [35, Sect. 4.A] or by iteratively dividing the input space and using local GPs [36], [37].

Overall, most state-of-the-art methods lack closed-loop guarantees, are limited to linear systems, yield computationally expensive approximate controllers, or require an iterative design process using a posteriori validation of error bounds.

*Contribution:* In this work, we present ALKIA-X, a novel algorithm based on kernel interpolation [30], [31], [32], [33], a noise-free special case of KRR, to automatically approximate functions up to any desired accuracy. ALKIA-X automatically approximates nonlinear MPC schemes while inheriting closed-loop guarantees (see Fig. 1). ALKIA-X is tailored to approximate MPC due to the following properties:

- P1) fast-to-evaluate approximating function;
- P2) guaranteed satisfaction of desired approximation error;
- P3) bound on worst-case number of required samples;
- P4) automatic and noniterative algorithm with well-conditioned computations and ready-to-use Python implementation.<sup>1</sup>

Property (P1) enables the implementation of the approximate MPC for fast control applications with limited computational resources. Property (P2) ensures closed-loop guarantees on stability and constraint satisfaction on the approximate MPC and is the main theoretical contribution in combination with (P3). Properties (P3) and (P4) enable a reliable and automatic approximation. The aforementioned properties are obtained by developing, extending, and combining the following tools:

- T1) localized kernel interpolation;
- T2) adaptive subdomain partitioning and length scale adjustment;
- T3) RKHS norm extrapolation.

Specifically, we follow a localized kernel interpolation approach (T1) by only using a subset of data, resulting in a piecewise-defined approximating function. In addition, we adaptively partition the domain into subdomains (T2), adjust the length scale accordingly, and sample equidistantly in each subdomain. Finally, we derive a heuristic RKHS norm extrapolation (T3), eliminating the need for an oracle to provide the RKHS norm.

<sup>1</sup>The code can be found here. [Online.] Available: <https://github.com/tokmak1/ALKIA-X>

Overall, ALKIA-X successfully addresses the problem of approximating nonlinear MPC schemes with closed-loop guarantees by yielding a fast-to-evaluate approximating function (P1) and guaranteeing any desired approximation accuracy (P2) with a reliable and automatic algorithm (P3), (P4). We demonstrate the performance of ALKIA-X by approximating the MPC schemes for two nonlinear systems.

- 1) A simple continuous stirred tank reactor from prior work [8]. ALKIA-X computes the approximate MPC in 9 h, yielding a controller with closed-loop guarantees and an online evaluation time below 50  $\mu$ s, thus outperforming prior work [8] by over one order of magnitude.
- 2) A realistic application to control a cold atmospheric plasma device [46, Ch. 4.5]. ALKIA-X computes the approximate MPC in 66 h, yielding a controller with only 33 MB of memory and an online evaluation time of 100  $\mu$ s.

*Outline:* The rest of this article is organized as follows. We present the problem setting in Section II-A, where we formally reduce the MPC approximation problem into a function approximation problem. We use kernel interpolation to tackle the function approximation problem, for which we introduce preliminaries in Section II-B. Section III introduces the adaptive and localized kernel interpolation algorithm, including theoretical guarantees regarding the desired approximation error and worst-case sample complexity. Section IV extends the approach to unknown RKHS norms by using an RKHS norm extrapolation, yielding ALKIA-X. Furthermore, Section V investigates approximating MPC schemes via ALKIA-X and the resulting closed-loop guarantees on the approximate MPC. Section VI demonstrates the successful deployment of ALKIA-X for approximating two nonlinear MPC schemes. Finally, Section VII concludes this article.

*Notations:* The set of nonnegative real numbers is given by  $\mathbb{R}_{\geq 0}$ . The set of positive real numbers and the set of natural numbers greater or equal to  $N \in \mathbb{R}_{> 0}$  are denoted by  $\mathbb{R}_{> 0}$  and  $\mathbb{N}_{\geq N}$ , respectively. For a set  $X \subseteq \mathbb{R}^N$ , we denote the cardinality by  $\text{card}(X)$ . For a vector  $x \in \mathbb{R}^n$ , the Euclidean norm, the infinity norm, and the 1-norm are denoted by  $\|x\|_2$ ,  $\|x\|_\infty$ , and  $\|x\|_1$ , respectively. For a matrix  $A \in \mathbb{R}^{n \times n}$ , the induced infinity norm is denoted by  $\|A\|_\infty$ . We define  $\mathbf{1}_n := [1, \dots, 1]^\top \in \mathbb{R}^n$  and  $\mathbf{0}_n := [0, \dots, 0]^\top \in \mathbb{R}^n$ . For two functions  $f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  and  $g: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ , we write  $f(\epsilon) = \mathcal{O}(g(\epsilon))$  if  $\frac{f(\epsilon)}{g(\epsilon)} < \infty$  as  $\epsilon \rightarrow 0$  with  $\epsilon > 0$ . Moreover, a function  $f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  belongs to class  $\mathcal{K}_\infty$  if it is strictly increasing, continuous, and  $f(0) = 0$ .

## II. PROBLEM SETTING AND PRELIMINARIES

First, we describe the problem setting (see Section II-A) before giving preliminaries on kernel interpolation (see Section II-B). Then, we discuss the computation of an upper bound on the power function (see Section II-C).

### A. Problem Setting

We consider a general nonlinear dynamical system  $x_{t+1} = g(x_t, u_t)$  with system state  $x_t \in \mathbb{R}^n$ , control input  $u_t \in \mathbb{R}^{n_u}$ , and time index  $t \in \mathbb{N}$ . The control goal is to stabilize the steady-state  $x_s$  while ensuring the satisfaction of user-specified state and input constraints, i.e.,  $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ ,  $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ ,  $\forall t \in \mathbb{N}$ . MPC provides a constructive approach to define a feedback

law  $u = f(x)$  that ensures asymptotic stability and constraint satisfaction.

*Approximate MPC:* The feedback law  $f$  is based on the solution of a finite-horizon optimal control problem and solving such an NLP during online operation is computationally expensive. Hence, we aim to replace  $f$  with an explicit approximate feedback law  $u = h(x)$  that we construct offline. The main challenge is to ensure that the approximate MPC  $h$  inherits the closed-loop guarantees induced by the MPC  $f$ .

*Solution approach:* The proposed framework is outlined in Fig. 1. Analogous to the approach in [8] and [9], we consider an MPC feedback law  $f$  that is robust with respect to input disturbances bounded by  $\epsilon$ . Hence, the approximate MPC yields desirable closed-loop guarantees if  $\|f(x) - h(x)\|_\infty \leq \epsilon$  for all  $x \in \mathcal{X}$ .

*Function approximation:* Consequently, the problem of determining an approximate MPC with closed-loop guarantees reduces to computing an explicit function  $h$  that approximates the ground truth  $f$  with a uniform approximation error bounded by  $\epsilon$ , i.e.,  $\|f(x) - h(x)\|_\infty \leq \epsilon$  for all  $x \in \mathcal{X}$ . The ground truth  $f$  is implicitly defined through the solution of an NLP, and hence, we regard  $f$  as an unknown function that we can query to receive noise-free evaluations. In the following, we focus solely on this general function approximation problem. Specifically, in this article, we solve this function approximation problem by proposing a novel algorithm based on kernel interpolation. We revisit the resulting control properties of the approximate MPC scheme later in Section V.

*Simplifications:* We consider, w.l.o.g., a scalar ground truth  $f: \mathcal{X} \rightarrow \mathbb{R}$  and approximating function  $h: \mathcal{X} \rightarrow \mathbb{R}$ , and thus require

$$|f(x) - h(x)| \leq \epsilon \quad \forall x \in \mathcal{X}. \quad (1)$$

For vector-valued functions, each output dimension can be approximated individually, resulting in multiple scalar approximation problems ensuring  $\|f(x) - h(x)\|_\infty \leq \epsilon$  for all  $x \in \mathcal{X}$ . For simplicity of exposition, we assume that the domain is a unit cube, i.e.,  $\mathcal{X} = [0, 1]^n$ .

### B. Preliminaries on Kernel Interpolation

In this section, we introduce kernel interpolation, see [30], [31], [32], and [33] for more details. First, we collect the standing assumptions on the kernel  $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_{> 0}$ , which is the central object in kernel interpolation.

*Assumption 1:* The kernel  $k$  is:

- 1) continuous.
- 2) (strictly) positive definite, i.e.,  $\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j k(x_i, x_j) > 0$  for all  $\mathbb{R}^N \ni [\alpha_1, \dots, \alpha_N]^\top \neq \mathbf{0}_N$ , for all pairwise distinct  $x_1, \dots, x_N \in \mathcal{X}$ , and any  $N \in \mathbb{N}$ ;
- 3) only a function of the Euclidean distance, i.e.,  $\exists \tilde{k}: \mathbb{R}_{\geq 0} \rightarrow (0, 1]$  such that  $k(x, x') = \tilde{k}(\|x - x'\|_2)$  for all  $x, x' \in \mathcal{X}$ .

Moreover,  $\tilde{k}$  is strictly decreasing and  $\tilde{k}(0) = 1$ .

The first two conditions in Assumption 1 are standard assumptions for kernels [30]. Condition 3) implies a uniform length scale in all dimensions and that  $k$  is a radial kernel [31, Ch. 4.7]. We can assume that  $\tilde{k}(0) = 1$  w.l.o.g. by normalizing. Assumption 1 holds for many common kernels, such as the squared-exponential or the Matérn kernel [34].

Let  $X = \{x_1, \dots, x_N\} \subseteq \mathcal{X}$  be a set of samples and assume that  $x_1, \dots, x_N$  are pairwise distinct. We collect the noise-free



evaluations of  $f$  on the inputs  $X$  in a vector denoted by

$$f_X = [f(x_1), \dots, f(x_N)]^\top. \quad (2)$$

Given samples  $X$ , we denote by  $K_X \in \mathbb{R}_{>0}^{N \times N}$  the symmetric and positive definite covariance matrix, also known as the kernel matrix or the Gram matrix [31], [34], i.e., the entry at row  $i$  and column  $j$  is  $k(x_i, x_j)$  for any  $i, j \in \{1, \dots, N\}$ . Given samples  $X$ , the covariance vector  $k_X: \mathcal{X} \rightarrow \mathbb{R}_{>0}^N$  is defined as

$$k_X(x) := [k(x, x_1), k(x, x_2), \dots, k(x, x_N)]^\top. \quad (3)$$

With the introduced definitions, we can write the resulting approximating function  $h_X: \mathcal{X} \rightarrow \mathbb{R}$  as [33, Sect. 3.2]

$$h_X(x) = f_X^\top K_X^{-1} k_X(x). \quad (4)$$

The RKHS norm is a characteristic value of RKHS functions that quantifies their complexity, and which we later use to bound the approximation error. We denote the RKHS norm of a function with respect to the RKHS of kernel  $k$  by  $\|\cdot\|_k$  and the RKHS norm of the approximating function (4) satisfies

$$\|h_X\|_k = \sqrt{f_X^\top K_X^{-1} f_X}. \quad (5)$$

Notably, approximating function (4) is the unique solution of a variational problem [33, Thm. 3.5], interpolating the given samples, i.e.,  $h_X(x) = f(x)$  for all  $x \in X$ , with the minimal RKHS norm. Thus,  $\|f\|_k \geq \|h_X\|_k$  holds for all  $X \subseteq \mathcal{X}$ . Henceforth, we assume that the unknown ground truth  $f$  is a member of the RKHS of the chosen kernel  $k$ , which trivially implies that  $\|f\|_k < \infty$ . This is a standard assumption for kernel-based approximation (see, e.g., [29], [30], [38], [39], [40], [41], [42], and [43]), which we discuss in more detail later (see Assumptions 3 and 5).

Next, we define the power function  $P_X: \mathcal{X} \rightarrow [0, 1]$  [32, Sect. 9.3]

$$P_X(x) := \sqrt{1 - k_X(x)^\top K_X^{-1} k_X(x)} \quad (6)$$

which is used in the following proposition to characterize input-dependent error bounds for kernel interpolation.

**Proposition 1:** [32, Sect. 14.1] Let Assumption 1 hold. Then, we have

$$|f(x) - h_X(x)| \leq P_X(x) \sqrt{\|f\|_k^2 - \|h_X\|_k^2} \quad (7)$$

for all  $x \in \mathcal{X}$  and any samples  $X \subseteq \mathcal{X}$ .

Note that Proposition 1 is a modified version of the Golomb–Weinberger bound [32, Sect. 9.3]. Based on Proposition 1, we can obtain *uniform* bounds on the approximation error given a uniform upper bound on the power function  $P_X$  and the ground truth RKHS norm  $\|f\|_k$ . In addition, the following lemma ensures continuity of the RKHS function.

**Lemma 1:** Let Assumption 1 hold. Then, for all  $x, x' \in \mathcal{X}$ , we have

$$|f(x) - f(x')| \leq \|f\|_k \sqrt{2\mathfrak{K}(\|x - x'\|_2)} \quad (8)$$

with

$$\mathfrak{K} := 1 - \tilde{k} \in \mathcal{K}_\infty. \quad (9)$$

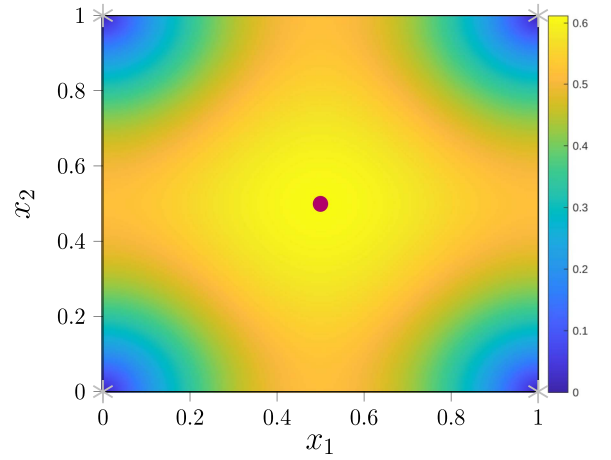


Fig. 2. Power function  $P_X$  on a cube with samples  $X$  on its vertices, which are illustrated by the gray asterisks. The maximum of the power function is at the center (magenta point). This plot is generated with the Matérn kernel with  $\nu = \frac{3}{2}$  to compute the power function on the cube  $[0, 1]^2$ .

Lemma 1 follows from standard<sup>2</sup> arguments, see, e.g., [31, Proof of Lemma 4.30]. It later assists in proving sample complexity bounds for arbitrarily accurate approximating functions.

### C. Upper Bound on the Power Function

Based on (7), we can ensure the desired uniform error bound (1) if a bound on the RKHS norm of the ground truth and a uniform upper bound on the power function  $P_X$  are available. However, finding the maximum of  $P_X$  over the space  $\mathcal{X}$  in general requires the solution of a nonconvex optimization problem. We circumvent this issue by using a local approximation scheme with samples  $X$  at the  $2^n$  vertices of a cube. In this symmetric setting, the center of the cube is the farthest away from the given samples.

**Assumption 2:** For any cube  $\tilde{\mathcal{X}} \subseteq \mathcal{X}$  with  $2^n$  samples  $X$  located at its vertices and with the center  $\bar{x} = \frac{1}{2^n} \sum_{x \in X} x$ , it holds that  $P_X(x) \leq P_X(\bar{x})$  for all  $x \in \tilde{\mathcal{X}}$ .

Since we consider *radial* kernels that are strictly decreasing functions of the Euclidean distance between two inputs (see Assumption 1), it is reasonable to assume that the maximum of the power function is at the center of the cube. Fig. 2 shows an exemplary power function of a kernel that satisfies Assumption 1, highlighting that the maximum of the power function is indeed at the center of the cube. Note that Assumption 2 is *not* an assumption on the ground truth  $f$  but follows from Assumption 1 and the utilized sampling scheme.

## III. ADAPTIVE AND LOCALIZED KERNEL INTERPOLATION ALGORITHM

In this section, we present the proposed adaptive and localized kernel interpolation algorithm to automatically compute a sufficiently accurate approximating function. In Section III-A,

<sup>2</sup>Due to Assumption 1, the function  $\mathfrak{K}: \mathbb{R}_{\geq 0} \rightarrow [0, 1]$  belongs to class  $\mathcal{K}_\infty$ , and thus, its inverse  $\mathfrak{K}^{-1}: [0, 1] \rightarrow \mathbb{R}_{\geq 0}$  is a continuous and strictly increasing function, i.e., it is again a class  $\mathcal{K}_\infty$  function [47, Lemma 4.2].

we describe the main idea before explaining the details in Section III-B. We provide a theoretical analysis in Section III-C, including the guaranteed error bound and a bound on the worst-case sample complexity.

### A. Main Idea

A naïve approach to determine a sufficiently accurate approximating function would be to equidistantly sample the ground truth until the kernel interpolation ensures the desired error bound (1). The proposed approach modifies classic kernel interpolation by using *local* approximations and an *adaptive* adjustment of the length scale.

We use a *localized* kernel interpolation approach (T1) by locally sampling equidistantly and computing a piecewise-defined approximating function based on local cubes that only use a subset of the available samples. This results in a fast online evaluation (P1) and enables the reliable computation of a guaranteed bound on the approximation error (P2), (P3).

In addition, we use an *adaptive* adjustment of the length scale by partitioning the domain into subdomains (T2) and perform localized kernel interpolation on each subdomain separately, where each subdomain has an individual length scale and RKHS norm. For every subdomain, we can compute the sufficient number of samples that ensure an approximation error that is uniformly bounded by  $\epsilon$ . Moreover, we impose an upper bound on the number of samples per subdomain using a hyperparameter  $\bar{p} \in \mathbb{N}$ . If more samples are required to ensure the desired error bound, we partition the subdomain and reduce the length scale. This leads to an *adaptive* subdomain partitioning with denser sampling in harder-to-approximate areas. Furthermore, upper bounding the number of samples for each subdomain enforces an upper bound on the condition number of the covariance matrices, leading to numerically reliable computations, even for millions of samples, without requiring any regularization (P4). Finally, independent execution on each subdomain enables a highly parallelized offline approximation, as is the case in other function approximation techniques [27], [36].

### B. Proposed Algorithm

We partition the domain  $\mathcal{X}$  into subdomains  $\mathcal{X}_a$ , i.e.,  $\mathcal{X} = \cup_{a \in \mathcal{A}} \mathcal{X}_a$ ,  $\mathcal{A} \subseteq \mathbb{N}$ , where each subdomain is a cube. Each subdomain  $\mathcal{X}_a$  has an individual length scale  $\ell_a$  and an individually scaled kernel  $\tilde{k}_a$  with

$$\tilde{k}_a(\cdot) := \tilde{k}\left(\frac{\cdot}{\ell_a}\right), \quad \ell_a = \max_{x_i, x_j \in \mathcal{X}_a} \|x_i - x_j\|_\infty. \quad (10)$$

Accordingly, the covariance matrix, the covariance vector, and the function  $\mathfrak{K}$  [see (9)] for each subdomain  $a \in \mathcal{A}$  are denoted by  $K_{a,X}$ ,  $k_{a,X}$ , and  $\mathfrak{K}_a$ , respectively.

We sample equidistantly in every subdomain  $\mathcal{X}_a$  with grid size  $\Delta x_a > 0$ , which is chosen such that  $\frac{\ell_a}{\Delta x_a} = 2^{p_a}$  with some later specified  $p_a \in \mathbb{N}_{\geq \bar{p}}$  and a user-chosen hyperparameter  $\bar{p} > 0$ . For any  $p \geq \bar{p}$ , we denote by  $X_{a,p}$  the set of  $(1 + 2^p)^n$  equidistant samples on subdomain  $\mathcal{X}_a$ . The total samples on the domain  $\mathcal{X}$  are thus given by  $X = \cup_{a \in \mathcal{A}} X_{a,p_a}$ .

In addition, we divide the subdomain  $\mathcal{X}_a$  into uniform local cubes  $\mathcal{X}_c$ , i.e.,  $\mathcal{X}_a = \cup_{c \in \mathcal{C}_{a,p_a}} \mathcal{X}_c$ ,  $\mathcal{C}_{a,p_a} \subseteq \mathbb{N}$ , of edge length  $\Delta x_a$ . Each local cube  $\mathcal{X}_c$  has  $2^n$  samples on its vertices, which we denote by  $X_c$ . The center  $\bar{x}_c$  of local cube  $\mathcal{X}_c$  is given by

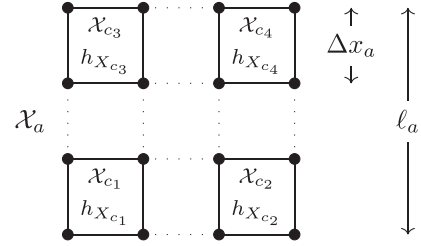


Fig. 3. Structure of the resulting uniform local cubes and the localized approximating functions on a subdomain  $\mathcal{X}_a$  with length scale  $\ell_a$  and grid size  $\Delta x_a$ . Illustrated are the local cubes  $\mathcal{X}_c$ ,  $c \in \{c_1, c_2, c_3, c_4\} \subseteq \mathcal{C}_{a,p_a}$ .

#### Algorithm 1: Shifted Localized Approximating Functions.

- 
- Require:**  $f, \tilde{k}_a, X_{a,p_a}$
- 1: Sample  $f_{X_{a,p_a}}$  ▷ (2)
  - 2: Compute mean  $\mu_a$  and shifted values  $\tilde{f}_{X_{a,p_a}}$  ▷ (11)
  - 3: Define cube partition  $\mathcal{C}_{a,p_a}$  from  $X_{a,p_a}$
  - 4: **for**  $c \in \mathcal{C}_{a,p_a}$  **do**
  - 5:   Build approximating function  $\tilde{h}_{X_c}$  ▷ (12)
- 

$\bar{x}_c = \frac{1}{2^n} \sum_{x \in X_c} x$ . The samples on subdomain  $\mathcal{X}_a \supseteq \mathcal{X}_c$  can be obtained with  $\tilde{X}_{a,p_a} = \cup_{c \in \mathcal{C}_{a,p_a}} X_c$ .

The resulting structure is shown in Fig. 3. For each local cube  $\mathcal{X}_c$ , we define a localized approximating function  $h_{X_c}$  that only depends on the  $2^n$  samples  $X_c$ , leading to a piecewise-defined approximating function and a localized kernel interpolation approach.

For the approximation procedure, we use function values that are shifted by an empirical mean value, i.e.,

$$\tilde{f}_{X_{a,p}} = f_{X_{a,p}} - \mu_a, \quad \mu_a := \frac{\sum_{x \in X_{a,p}} f(x)}{\text{card}(X_{a,p})}. \quad (11)$$

Due to the continuity of the ground truth (see Lemma 1), this shifting ensures arbitrarily small function values  $\tilde{f}_{X_{a,p}}$  for decreasing size of the subdomain  $\mathcal{X}_a$ , which later assists in guaranteeing an arbitrarily small approximation error  $\epsilon > 0$ . For any local cube  $\mathcal{X}_c \subseteq \mathcal{X}_a$  with samples  $X_c \subseteq X_{a,p_a}$ , the localized kernel interpolation with shifted function values (11) results in the shifted localized approximating function [see (4)]

$$\tilde{h}_{X_c}(x) = \tilde{f}_{X_c}^\top K_{a,X_c}^{-1} \tilde{k}_{a,X_c}(x). \quad (12)$$

To construct the shifted localized approximating functions on a subdomain  $\mathcal{X}_a$ , we first query the ground truth at the inputs  $X_{a,p_a}$  and shift the samples by the empirical mean  $\mu_a$ . With these samples, we compute the shifted localized approximating functions (12) for all local cubes  $\mathcal{X}_c \subseteq \mathcal{X}_a$ . Algorithm 1 summarizes this procedure.

In the following, we ensure that the shifted localized approximating functions computed in Algorithm 1 satisfy the desired error bound  $\epsilon$ . First, we define the shifted ground truth  $\tilde{f}_a$  on subdomain  $\mathcal{X}_a$  as

$$\tilde{f}_a(x) := f(x) - \mu_a. \quad (13)$$

Given the subdomain partitioning and the local approximation scheme, the desired approximation error bound (1) reduces to

$$|\tilde{f}_a(x) - \tilde{h}_{X_c}(x)| \leq \epsilon \quad \forall a \in \mathcal{A} \quad \forall c \in \mathcal{C}_{a,p_a} \quad \forall x \in \mathcal{X}_c. \quad (14)$$

Furthermore, analogous to standard kernel-based approximation methods (see, e.g., [29], [30], [38], [39], [40], [41], [42], and [43]), we require knowledge of an upper bound on the corresponding RKHS norm.

**Assumption 3:** For any  $a \in \mathcal{A}$ ,  $\tilde{f}_a$  is a member of the RHKS of kernel  $\tilde{k}_a$ . Moreover, we have access to an oracle  $\bar{\Gamma}_a \in (0, \infty)$  with  $\bar{\Gamma}_a \geq \|\tilde{f}_a\|_{\tilde{k}_a}$ .

In Section IV, we introduce a heuristic approximation that replaces the oracle assumption.

The following result provides a lower bound on the number of samples that ensure the desired error bound  $\epsilon$  for each subdomain  $\mathcal{X}_a$ .

**Proposition 2:** Let Assumptions 1–3 hold. For any  $a \in \mathcal{A}$  and any  $\epsilon > 0$ , there exists a  $p_a^* \in \mathbb{N}$  such that  $p_a \geq p_a^*$  implies  $P_{X_c}(\bar{x}_c)\bar{\Gamma}_a \leq \epsilon$  for all  $c \in \mathcal{C}_{a,p_a}$ . Furthermore, if  $p_a \geq p_a^*$  for all  $a \in \mathcal{A}$ , then the localized approximating functions according to Algorithm 1 satisfy (14).

*Proof:* The proof is divided into three parts. In Part I, we bound the power function  $P_{X_c}(\bar{x}_c)$  using an eigenvector of the covariance matrix. In Part II, we show that  $P_{X_c}(\bar{x}_c)\bar{\Gamma}_a \leq \epsilon$  holds for  $p_a \geq p_a^*$ , while in Part III, we prove that this implies (14).

**Part I:** Pick any  $a \in \mathcal{A}$ . For any  $c \in \mathcal{C}_{a,p_a}$ , given samples  $X_c$  on the vertices of local cube  $\mathcal{X}_c$ , we denote the scaled covariance matrix and the scaled covariance vector by  $K_{a,X_c}$  and  $k_{a,X_c}$ , respectively. In this cubic setting, the covariance matrix  $K_{a,X_c}$  has constant row sum. Moreover, the covariance vector satisfies  $k_{a,X_c}(\bar{x}_c) = \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)\mathbf{1}_{2^n}$ . Thus,  $k_{a,X_c}(\bar{x}_c)$  is an eigenvector of  $K_{a,X_c}$  with eigenvalue

$$\beta = \sum_{x \in X_c} \tilde{k}_a(\|x - x'\|_2) \stackrel{\text{Asm.1}}{\leq} 2^n \quad (15)$$

for an arbitrary  $x' \in X_c$ . Hence, we have

$$K_{a,X_c}^{-1} k_{a,X_c}(\bar{x}_c) = \frac{1}{\beta} k_{a,X_c}(\bar{x}_c). \quad (16)$$

This yields

$$\begin{aligned} P_{X_c}(\bar{x}_c) &\stackrel{(6),(16)}{=} \sqrt{1 - \frac{\tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)^2 2^n}{\beta}} \\ &\stackrel{(15)}{\leq} \sqrt{1 - \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)^2}. \end{aligned} \quad (17)$$

**Part II:** We introduce the notation  $\lceil \xi \rceil \in \mathbb{N}$  as rounding  $\xi \in \mathbb{R}$  up to the nearest integer. First, suppose  $\epsilon < \bar{\Gamma}_a$  and define

$$p_a^* := \left\lceil \log_2 \left( \frac{\ell_a \sqrt{n}}{2\mathfrak{K}_a^{-1}\left(\frac{1}{2}\left(\frac{\epsilon}{\bar{\Gamma}_a}\right)^2\right)} \right) \right\rceil. \quad (18)$$

For any  $p_a \geq p_a^*$ , any  $c \in \mathcal{C}_{a,p_a}$ , and any  $\epsilon > 0$ , it holds that

$$\begin{aligned} P_{X_c}(\bar{x}_c)\bar{\Gamma}_a &\stackrel{(17)}{\leq} \bar{\Gamma}_a \sqrt{1 - \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)^2} \\ &= \bar{\Gamma}_a \sqrt{\left(1 - \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)\right) \left(1 + \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)\right)} \\ &\stackrel{(9)}{=} \bar{\Gamma}_a \sqrt{\mathfrak{K}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right) \left(1 + \tilde{k}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)\right)} \\ &\leq \bar{\Gamma}_a \sqrt{2\mathfrak{K}_a\left(\frac{\sqrt{n}\Delta x_a}{2}\right)} = \bar{\Gamma}_a \sqrt{2\mathfrak{K}_a\left(\frac{\sqrt{n}\ell_a}{2 \cdot 2^{p_a}}\right)} \\ &\stackrel{(18)}{\leq} \bar{\Gamma}_a \sqrt{2\mathfrak{K}_a\left(\mathfrak{K}_a^{-1}\left(\frac{1}{2}\left(\frac{\epsilon}{\bar{\Gamma}_a}\right)^2\right)\right)} = \epsilon. \end{aligned} \quad (19)$$

Note that we used  $2^{p_a} = \frac{\ell_a}{\Delta x_a}$  and the fact that  $\mathfrak{K}_a \in \mathcal{K}_\infty$  and  $\tilde{k}_a: \mathbb{R}_{\geq 0} \rightarrow (0, 1]$ . In the case of  $\epsilon \geq \bar{\Gamma}_a$ ,  $P_{X_c}(\bar{x}_c)\bar{\Gamma}_a \leq \epsilon$  [see (19)] holds for any  $\Delta x_a > 0$  since  $P_{X_c}(\bar{x}_c) \leq 1$  for all  $p_a \geq p_a^* = 0$ .

**Part III:** For any  $c \in \mathcal{C}_{a,p_a}$  and for any  $x \in \mathcal{X}_c$ , given the subdomain partitioning and the local approximation scheme, error bound (7) of Proposition 1 yields

$$|\tilde{f}_a(x) - \tilde{h}_{X_c}(x)| \leq P_{X_c}(x)\|\tilde{f}_a\|_{\tilde{k}_a}. \quad (20)$$

Thus, for any  $p_a \geq p_a^*$ , any  $c \in \mathcal{C}_{a,p_a}$ , any  $x \in \mathcal{X}_c$ , and any  $\epsilon > 0$ , it holds that

$$\begin{aligned} |\tilde{f}_a(x) - \tilde{h}_{X_c}(x)| &\stackrel{(20)}{\leq} P_{X_c}(x)\|\tilde{f}_a\|_{\tilde{k}_a} \stackrel{\text{Asm.2}}{\leq} P_{X_c}(\bar{x}_c)\|\tilde{f}_a\|_{\tilde{k}_a} \\ &\stackrel{\text{Asm.3}}{\leq} P_{X_c}(\bar{x}_c)\bar{\Gamma}_a \leq \epsilon \end{aligned}$$

where the last inequality was shown in Part II.  $\square$

Proposition 2 ensures arbitrary approximation accuracy  $\epsilon$  using a local approximation scheme by sampling densely enough. Since we additionally enforce a lower bound  $p_a \geq \underline{p}$ , we choose

$$p_a = \max\{\underline{p}, p_a^*\}. \quad (21)$$

For small accuracies  $\epsilon$ , this may result in large values  $p_a$ , which deteriorates the conditioning of the covariance matrices  $K_{a,X_c}$ , resulting in numerical unreliability. Thus, we introduce  $\bar{p} > \underline{p}$ , an upper bound on the integer  $p_a$ . This yields a uniform upper bound  $\bar{\kappa}$  on the condition number

$$\text{cond}(K_{a,X_c}) \leq \bar{\kappa} \quad \forall a \in \mathcal{A} \quad \forall p \in \{\underline{p}, \dots, \bar{p}\} \quad \forall c \in \mathcal{C}_{a,p} \quad (22)$$

with  $\text{cond}(K_{a,X_c}) := \|K_{a,X_c}\| \|K_{a,X_c}^{-1}\|_\infty$ . For a given  $p_a \geq \underline{p}$ , we have a fixed ratio  $\frac{\ell_a}{\Delta x_a} = 2^{p_a}$ , and hence,  $K_{a,X_c}$  is identical for any  $c \in \mathcal{C}_{a,p_a}$  and any  $a \in \mathcal{A}$ . Thus, given a uniformly bounded  $p_a \in \{\underline{p}, \dots, \bar{p}\}$ , this yields a uniform bound on the condition number  $\bar{\kappa}$ . Furthermore,  $\bar{p}$  and  $\bar{\kappa}$  have one-to-one correspondence.<sup>3</sup> If (21) returns  $p_a > \bar{p}$ , we partition the subdomain

<sup>3</sup>In fact, (22) can be enforced a priori. First, we define  $\mathcal{C}_{\bar{p}}$  as the cube partition resulting from  $\text{card}(X) = (1 + 2^{\bar{p}})^n$  equidistant samples on the domain  $\mathcal{X} = [0, 1]^n$ . Then, we compute the covariance matrix of the unscaled kernel  $k$  (i.e., with  $\ell = 1$ ) for an arbitrary  $c \in \mathcal{C}_{\bar{p}}$  with  $2^n$  samples  $X_c$  and check whether  $\text{cond}(K_{X_c}) \leq \bar{\kappa}$ .

**Algorithm 2:** Adaptive and Localized Kernel Interpolation.

---

**Require:**  $f, \mathcal{X}, \tilde{k}, \epsilon, \underline{p}, \bar{p}$   
 1: Init:  $\mathcal{A} = \{0\}, \mathcal{X}_0 = \mathcal{X}$   
 2: **for**  $a \in \mathcal{A}$  **do** ▷ Dynamic for-loop  
 3:   Query oracle for  $\bar{\Gamma}_a$  ▷ Assumption 3  
 4:   Determine sufficiently large  $p_a$  ▷ Proposition 2, (21)  
 5:   **if**  $p_a \leq \bar{p}$  **then** ▷ (22)  
 6:     Get localized approximating functions ▷ Alg. 1  
 7:   **else**  
 8:     Define  $2^n$  new sub-domains  $\cup_{a' \in \mathcal{A}'} \mathcal{X}_{a'} = \mathcal{X}_a$   
 9:      $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}' \setminus a$  ▷ Update set partitions

---

**Algorithm 3:** Online Evaluation.

---

**Require:**  $\tilde{h}, x$   
 1: Tree search  $c \in \mathcal{C}_{a,p_a}, a \in \mathcal{A}$  such that  
     $x \in \mathcal{X}_c \subseteq \mathcal{X}_a \subseteq \mathcal{X}$   
 2: **return**  $h(x)$

---

$\mathcal{X}_a$  into  $2^n$  cubes of the same size, which halves the length scale  $\ell_a$ , allowing for a denser sampling.

The overall offline approximation is summarized in Algorithm 2. To compute a sufficiently accurate approximation of the ground truth  $f$  on the domain  $\mathcal{X}$ , we iterate through a dynamic for-loop to determine the localized approximating functions for all subdomains. After querying the oracle to receive an upper bound on the RKHS norm of the ground truth  $\bar{\Gamma}_a$ , we determine the number of samples for the current subdomain with (21). If  $p_a \leq \bar{p}$ , then the matrices are well-conditioned according to (22), and we define the localized approximating functions for the current subdomain. If  $p_a > \bar{p}$ , we partition  $\mathcal{X}_a$  further into  $2^n$  uniform subdomains. We continue with the described procedure until we have constructed localized approximating functions that cover the whole domain.

The resulting localized approximating functions are used for the online evaluation, which is described in Algorithm 3. First, we define the function  $h: \mathcal{X} \rightarrow \mathbb{R}$  as the piecewise-defined approximating function on the domain  $\mathcal{X}$ , i.e., for any  $a \in \mathcal{A}$ ,  $c \in \mathcal{C}_{a,p_a}$ , and  $x \in \mathcal{X}_c$ , we define

$$\tilde{h}(x) := \tilde{h}_{X_c}(x), \quad h(x) := \tilde{h}_{X_c}(x) + \mu_a. \quad (23)$$

For the online evaluation, we determine the relevant local cube  $\mathcal{X}_c$  such that  $x \in \mathcal{X}_c$  by performing a tree search, and then evaluate the localized approximating function, similar to the online evaluations in [27] and [36].

**C. Sample Complexity and Uniform Approximation Error**

To derive an upper bound on the maximum number of samples, we use the following assumption on the ground truth.

**Assumption 4:** There exists a function  $\alpha \in \mathcal{K}_\infty$  such that for any  $a \in \mathcal{A}$ , the oracle in Assumption 3 satisfies

$$\alpha(\bar{\Gamma}_a) \leq \ell_a. \quad (24)$$

Shifted function values in combination with a continuous ground truth (see Lemma 1) ensure  $\lim_{\ell_a \rightarrow 0} \tilde{f}_a(x) = 0$  for all  $x \in \mathcal{X}_a$ , which implies that  $\lim_{\ell_a \rightarrow 0} \|\tilde{f}_a\|_{\tilde{k}_a} = 0$ , and hence provides an intuition for Assumption 4. In fact, in the proof of Theorem 2, we show that the RKHS norm of the approximating

function is upper bounded by  $\alpha^{-1}(\ell_a)$  with some later specified  $\alpha \in \mathcal{K}_\infty$ .

The following theorem provides an upper bound on the sample complexity (P3) and ensures that the localized approximating functions satisfy error bound  $\epsilon$  (P2).

**Theorem 1:** Let Assumptions 1–4 hold. Then, Algorithm 2 terminates after querying at most  $\text{card}(X) \leq \mathcal{O}((\frac{1}{\alpha(\epsilon)})^n)$  samples. Furthermore, the resulting approximating function (see Algorithm 3) satisfies Inequality (1), i.e., the approximation error is uniformly bounded by  $\epsilon > 0$ .

**Proof:** The proof is divided into three parts. In Part I, we show that Algorithm 2 terminates once a threshold length scale is reached. In Part II, we derive the worst-case sample complexity in order to reach this threshold length scale. In Part III, we demonstrate that the resulting localized approximating functions satisfy error bound  $\epsilon$ .

**Part I:** Consider any  $a \in \mathcal{A}$ . First, we show that Algorithm 2 terminates whenever

$$\ell_a \leq \alpha(\epsilon). \quad (25)$$

For any  $c \in \mathcal{C}_{a,p_a}$ , it holds that

$$\bar{\Gamma}_a \stackrel{(24)}{\leq} \alpha^{-1}(\ell_a) \stackrel{(25)}{\leq} \alpha^{-1}(\alpha(\epsilon)) = \epsilon \quad (26)$$

which yields  $p_a^* = 0$  in Proposition 2. Therefore, (21) returns  $p_a = \underline{p} \leq \bar{p}$ , i.e., Algorithm 2 terminates if (25) is satisfied. Thus, since the length scale  $\ell_a$  is halved with each partitioning

$$\ell_a > \frac{\alpha(\epsilon)}{2} \quad (27)$$

always holds for any partition  $a \in \mathcal{A}$ .

**Part II:** The number of subdomains satisfies

$$\text{card}(\mathcal{A}) \leq \left( \min_{a \in \mathcal{A}} \ell_a \right)^{-n} \stackrel{(27)}{<} \left( \frac{2}{\alpha(\epsilon)} \right)^n. \quad (28)$$

Moreover, each subdomain  $\mathcal{X}_a$  has at most

$$\text{card}(X_a) \stackrel{(22)}{\leq} (1 + 2^{\bar{p}})^n \quad (29)$$

samples. Thus, Algorithm 2 terminates after at most

$$\begin{aligned} \text{card}(X) &= \sum_{a \in \mathcal{A}} \text{card}(X_a) \stackrel{(28),(29)}{<} \left( \frac{2}{\alpha(\epsilon)} \right)^n (1 + 2^{\bar{p}})^n \\ &= \mathcal{O} \left( \left( \frac{1}{\alpha(\epsilon)} \right)^n \right) \end{aligned}$$

samples.

**Part III:** Once Algorithm 2 has terminated, each subdomain  $\mathcal{X}_a$  satisfies  $p_a \geq p_a^*$ . Hence, Proposition 2 ensures the satisfaction of error bound (14). For any  $c \in \mathcal{C}_{a,p_a}$ , any  $a \in \mathcal{A}$ , and any  $x \in \mathcal{X}_c$ , we have that  $|f(x) - h(x)| \stackrel{(13),(23)}{=} |\tilde{f}_a(x) - \tilde{h}_{X_c}(x)| \stackrel{(14)}{\leq} \epsilon$ , i.e., error bound (1) holds.  $\square$

The approach proposed in this section leads to a one-shot algorithm with explicit worst-case sample complexity and a guaranteed error bound. However, assuming a known RKHS norm of the unknown ground truth (see Assumptions 3 and 4) can be restrictive in practice. In the next section, we alleviate this issue by introducing a heuristic.



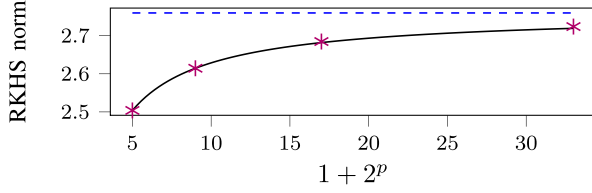


Fig. 4. Illustration of the RKHS norm extrapolation. The magenta asterisks show the RKHS norm of the approximating function  $\|\tilde{h}_{X_{a,p}}\|_{\tilde{k}_a}$  (30). The solid black line is the extrapolating function  $\gamma_a$  (33), while the dashed blue line depicts its limit value  $\bar{\Gamma}_a$  (35).

*Remark 1 (Adaptive subdomain partitioning):* The proposed adaptive subdomain partitioning is similar to [27], where also more samples are acquired in harder-to-approximate areas. The algorithm in [27] is based on wavelet multiresolution analysis, explicitly requires convexity for theoretical guarantees, and does not provide finite sample complexity bounds. In contrast, the algorithm we propose is based on kernel interpolation and guarantees the satisfaction of any uniform approximation error  $\epsilon$  without requiring linearity nor convexity. The approximation accuracy is guaranteed through kernel interpolation theory, which instead depends on the complexity of the ground truth and the sampling density.

#### IV. ALKIA-X

In this section, we extend the algorithm proposed in Section III to unknown RKHS norms by introducing a heuristic RKHS norm extrapolation. This extension yields the **Adaptive and Localized Kernel Interpolation Algorithm with eXtrapolated RKHS norm** (ALKIA-X). In Section IV-A, we introduce the main idea before explaining the details of the RKHS norm extrapolation in Section IV-B. We provide the theoretical analysis in Section IV-C, including the guaranteed error bound and the worst-case sample complexity. Then, we present simpler complexity bounds for a popular kernel choice in Section IV-D.

##### A. Main Idea

ALKIA-X has one major difference compared to the algorithm presented in Section III: instead of assuming access to an oracle that returns an upper bound on the RKHS norm of the unknown ground truth (see Assumption 3), we introduce a novel *RKHS norm extrapolation* (T3). Specifically, we follow an extrapolation from the RKHS norm of the approximating function for each subdomain  $\mathcal{X}_a$ . Fig. 4 shows that the trend of the RKHS norm of the approximating function empirically resembles exponential behavior.<sup>4</sup> Hence, we fit an *exponential* function to the first two values of the RKHS norm of the approximating function and assume that its limit provides an upper bound on the RKHS norm of the unknown ground truth for each subdomain.

<sup>4</sup>The results in Fig. 4 are generated using the Matérn kernel [34] with  $\nu = \frac{3}{2}$  and the function  $f: [0, 1]^2 \rightarrow \mathbb{R}$  with  $f(x) = \sin(2\pi x_1) + \cos(2\pi x_2)$ , where  $x = [x_1, x_2]^\top \in \mathbb{R}^2$  and  $\mathcal{X}_a = [0, \frac{1}{3}]^2$ . The remaining hyperparameters are  $\underline{p} = 2$ ,  $\bar{p} = 5$  ( $\bar{\kappa} = 1.14 \cdot 10^8$ ), and  $\epsilon = 10^{-6}$ .

##### B. RKHS Norm Extrapolation

In the following, we formalize the exponential extrapolating function. To this end, we first introduce the RKHS norm of the shifted approximating function of subdomain  $\mathcal{X}_a$  based on  $\text{card}(X_{a,\hat{p}}) = (1 + 2^{\hat{p}})^n$  samples as

$$\|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a} = \sqrt{\tilde{f}_{X_{a,\hat{p}}}^\top K_{a,X_{a,\hat{p}}}^{-1} \tilde{f}_{X_{a,\hat{p}}}} \quad \forall \hat{p} \in \{\underline{p}, \dots, \bar{p}\}. \quad (30)$$

Note that we only use  $\|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a}$  for  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$  to construct the extrapolating function. To guarantee well-conditioned computations, we require that

$$\text{cond}(K_{a,X_{a,\hat{p}}}) \leq \bar{\kappa} \quad \forall \hat{p} \in \{\underline{p}, \underline{p} + 1\} \quad \forall a \in \mathcal{A} \quad (31)$$

holds in addition to (22). Note that (31) is enforced through the choice of  $\underline{p}$  and that  $\underline{p}$  and  $\bar{\kappa}$  have one-to-one correspondence and are uniform bounds.<sup>5</sup>

We upper bound  $\|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a}$ ,  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$ , with

$$\hat{\gamma}_{a,\hat{p}} = \|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a} + \frac{\epsilon}{2^{\lambda+1}}, \quad \lambda := 2 + 2^{-\underline{p}}. \quad (32)$$

The form of this upper bound becomes clearer when developing the theoretical guarantees in Section IV-C. The extrapolating function  $\gamma_a: \mathbb{N}_{\geq \underline{p}} \rightarrow \mathbb{R}_{\geq 0}$  is given by

$$\gamma_a(p) := \bar{\Gamma}_a \exp\left(-\frac{\tau_a}{1 + 2^p}\right) \quad \forall p \in \{\underline{p}, \dots, \bar{p}\} \quad (33)$$

where  $\bar{\Gamma}_a$  is the limit of the extrapolating function and  $\tau_a$  determines the rate at which the extrapolating function approaches its limit value.

*Lemma 2:* For any  $a \in \mathcal{A}$  and any  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$

$$\gamma_a(\hat{p}) = \hat{\gamma}_{a,\hat{p}} \quad (34)$$

holds with

$$\tau_a = \ln\left(\frac{\hat{\gamma}_{a,\underline{p}+1}}{\hat{\gamma}_{a,\underline{p}}}\right) \lambda (1 + 2^{\underline{p}}), \quad \bar{\Gamma}_a = \hat{\gamma}_{a,\underline{p}} \left(\frac{\hat{\gamma}_{a,\underline{p}+1}}{\hat{\gamma}_{a,\underline{p}}}\right)^\lambda \quad (35)$$

and  $\bar{\Gamma}_a \in (0, \infty)$ .

The proof can be found in Appendix A. Fig. 4 depicts the RKHS norm extrapolation heuristic applied to a toy example. The resulting extrapolation approximates the trend of the RKHS norm of the approximating function well, although it has been constructed with the first two values  $\|\tilde{h}_{X_{a,p}}\|_{\tilde{k}_a}$ ,  $p \in \{\underline{p}, \underline{p} + 1\}$  only. We use the limit of the extrapolating function  $\bar{\Gamma}_a$  as an upper bound on the RKHS norm, which we formalize in Assumption 5.

*Assumption 5:* For any  $a \in \mathcal{A}$ ,  $\tilde{f}_a$  is a member of the RHKS of kernel  $\tilde{k}_a$ . Moreover, the limit of the extrapolating function  $\bar{\Gamma}_a \in (0, \infty)$  [see (33) and (35)] is an upper bound on the RHKS norm, i.e.,  $\bar{\Gamma}_a \geq \|\tilde{f}_a\|_{\tilde{k}_a}$ .

*Remark 2:* (RKHS norm approximation) In general, it is not possible to find an upper bound on the RKHS norm of the

<sup>5</sup>Akin to (22), for a fixed  $\underline{p} > 0$ , the condition number  $\text{cond}(K_{a,X_{a,\underline{p}+1}})$  is identical for each subdomain  $\mathcal{X}_a$ . Moreover, (31) can be enforced a priori by checking whether the covariance matrix of the unscaled kernel  $k$  (i.e., with  $\ell = 1$ ) with  $\text{card}(X) = (1 + 2^{\underline{p}+1})^n$  equidistant samples on domain  $\mathcal{X} = [0, 1]^n$  is well-conditioned, i.e., whether  $\text{cond}(K_X) \leq \bar{\kappa}$ .



**Algorithm 4:** ALKIA-X.

---

**Require:**  $f, \mathcal{X}, \tilde{k}, \epsilon, p, \bar{p}$   $\triangleright$  (31)  
1: Init:  $\mathcal{A} = \{0\}, \mathcal{X}_0 = \mathcal{X}$   
2: **for**  $a \in \mathcal{A}$  **do**  $\triangleright$  Dynamic for-loop  
3:   Get samples  $\tilde{f}_{X_{a,\hat{p}}}, \hat{p} \in \{\underline{p}, \underline{p} + 1\}$   $\triangleright$  (2), (11)  
4:   Compute  $\bar{\Gamma}_a$   $\triangleright$  (30), (32), (35)  
5:   Determine sufficiently large  $p_a$   $\triangleright$  Proposition 2, (21)  
6:   **if**  $p_a \leq \bar{p}$  **then**  $\triangleright$  (22)  
7:     Get localized approximating function  $\triangleright$  Algorithm 1  
8:   **else**  
9:     Define  $2^n$  new sub-domains  $\cup_{a' \in \mathcal{A}'} \mathcal{X}_{a'} = \mathcal{X}_a$   
10:     $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}' \setminus a$   $\triangleright$  Update set partitions

---

ground truth  $\|f\|_k$  (or any shifted or restricted version of it) from finite samples, since it is an underlying characteristic of an unknown function. Thus, there exists no perfect solution to tackle the challenging problem of working with the RKHS norm of an unknown ground truth. A basic approach, followed by many related works (see, e.g., [29], [30], [38], [39], [40], [41], [42], and [43]) and also in Section III of this article, is to assume that an oracle is available that returns a not overly conservative upper bound on the RKHS norm. In order to not rely on an oracle, we introduce the RKHS norm extrapolation (33) with its corresponding limit (35). The considered extrapolation is only a heuristic and Assumption 5 may fail on some occasions. Some early works in the field of information theory and signal detection compute the RKHS norm of the ground truth function. However, the authors assume that an explicit expression of the ground truth is known [48], [49], which is not the case in the setting of this article. Moreover, in [50], an upper bound on the RKHS norm of the ground truth based on the RKHS norm of the approximating function is derived. However, the authors assume that the ground truth is a member of a pre-RKHS spanned by the chosen kernel (see, e.g., [33, Sect. 2.3]), which is more restrictive. In the following, we briefly mention other approximations of  $\|f\|_k$  using  $\|h_X\|_k$ . As noted in [30, Remark 2], the RKHS norm of the approximating function  $\|h_X\|_k$  is an underestimation of the RKHS norm of the ground truth  $\|f\|_k$ . In [29, Appendix A], the authors use the intuition that  $\|h_X\|_k$  may converge to  $\|f\|_k$  for an increasing sample set  $X$ . Based hereon, [43, Appendix A] estimates an upper bound by *manually* checking the convergence of  $\|h_X\|_k$ . In contrast, we provide a simple formula to *automatically* extrapolate an estimate of the RKHS norm, which improves the ideas presented in [29] and [43], and empirically tends to closely match the underlying ground truth, as seen in Fig. 4.

The offline approximation using ALKIA-X is summarized in Algorithm 4. Compared to Algorithm 2 in Section III, instead of receiving an upper bound on the RKHS norm of the ground truth from an oracle (Assumption 3), we extrapolate  $\bar{\Gamma}_a$  (Assumption 5). Thus, the exact number of required samples is determined during the approximation procedure, as  $\bar{\Gamma}_a$  is not known a priori.

### C. Sample Complexity and Uniform Approximation Error

The following theorem provides a bound on the sample complexity and ensures the satisfaction of error bound  $\epsilon$ .

**Theorem 2:** Let Assumptions 1, 2, and 5 hold. Then, Algorithm 4 terminates after at most

$$\text{card}(X) \leq \mathcal{O} \left( \left( \frac{\sqrt{n}}{\mathfrak{K}^{-1} \left( C \left( \frac{\epsilon}{\|f\|_k} \right)^2 \right)} \right)^n \right) \quad (36)$$

samples, with some  $C > 0$  independent of  $\epsilon$  and  $f$ . Furthermore, the resulting approximating function (see Algorithm 3) satisfies Inequality (1), i.e., the approximation error is uniformly bounded by  $\epsilon > 0$ .

*Proof:* The proof is structured analogously to the proof of Theorem 1. In Part I, we show that Algorithm 4 terminates once a threshold length scale is reached. In Part II, we derive the worst-case sample complexity in order to reach that threshold length scale. In Part III, we demonstrate that the resulting localized approximating functions satisfy error bound  $\epsilon$ .

*Part I:* Consider any  $a \in \mathcal{A}$ . First, we prove that  $|\tilde{f}_a(x)| \leq \|f\|_k \sqrt{2\mathfrak{K}(\sqrt{n}\ell_a)}$  holds for all  $x \in \mathcal{X}_a$ . Consider an arbitrary  $x \in \mathcal{X}_a$ , where w.l.o.g.,  $\tilde{f}_a(x) \geq 0$ . Pick any other  $x' \in \mathcal{X}_a$  with  $\tilde{f}_a(x') \leq 0$ , which exists since  $\sum_{x \in X_{a,\hat{p}}} \tilde{f}_a(x) = 0$  [see (11)]. It holds that

$$\begin{aligned} |\tilde{f}_a(x)| &\leq |\tilde{f}_a(x) - \tilde{f}_a(x')| \stackrel{(13)}{=} |f(x) - f(x')| \\ &\stackrel{\text{Lem.1}}{\leq} \|f\|_k \sqrt{2\mathfrak{K}(\|x - x'\|_2)} \stackrel{(10)}{\leq} \|f\|_k \sqrt{2\mathfrak{K}(\sqrt{n}\ell_a)}. \end{aligned} \quad (37)$$

Next, we derive an upper bound on  $\|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a}$ . For all  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$ , it holds that

$$\begin{aligned} \|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a} &\stackrel{(5)}{=} \sqrt{\tilde{f}_{X_{a,\hat{p}}}^\top K_{a,X_{a,\hat{p}}}^{-1} \tilde{f}_{X_{a,\hat{p}}}} \\ &\leq \sqrt{\|\tilde{f}_{X_{a,\hat{p}}}\|_1 \|\tilde{f}_{X_{a,\hat{p}}}\|_\infty \|K_{a,X_{a,\hat{p}}}^{-1}\|_\infty} \\ &\stackrel{(37)}{\leq} \|f\|_k \sqrt{2\mathfrak{K}(\sqrt{n}\ell_a)} (1 + 2^{p+1})^n \|K_{a,X_{a,\hat{p}}}^{-1}\|_\infty \\ &\stackrel{(31)}{\leq} \|f\|_k \sqrt{2\mathfrak{K}(\sqrt{n}\ell_a)} (1 + 2^{p+1})^n \bar{\kappa}. \end{aligned} \quad (38)$$

The first inequality follows from Hölder's inequality, whereas the second inequality holds since  $\text{card}(X_{a,\hat{p}}) \leq (1 + 2^{p+1})^n$  for all  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$ . Moreover, the last inequality is satisfied

since  $\|K_{a,X_{a,\hat{p}}}^{-1}\|_\infty = \frac{\text{cond}(K_{a,X_{a,\hat{p}}})}{\|K_{a,X_{a,\hat{p}}}\|_\infty} \stackrel{(31)}{\leq} \bar{\kappa}$  with  $\|K_{a,X_{a,\hat{p}}}\|_\infty \geq \max_{i,j} K_{a,X_{a,\hat{p}}}(i,j) = \tilde{k}(0) = 1$ . In the following, we show that Algorithm 4 terminates whenever:

$$\ell_a \leq \frac{\mathfrak{K}^{-1} \left( \frac{\epsilon^2}{2\bar{\kappa}(1+2^{p+1})^n 2^{2\lambda+2} \|f\|_k^2} \right)}{\sqrt{n}}. \quad (39)$$

For any  $\hat{p} \in \{\underline{p}, \underline{p} + 1\}$ , (38) and (39) imply

$$\|\tilde{h}_{X_{a,\hat{p}}}\|_{\tilde{k}_a} \leq \frac{\epsilon}{2^{\lambda+1}}. \quad (40)$$

For any  $c \in \mathcal{C}_{a,p_a}$ , (40) implies

$$\bar{\Gamma}_a \stackrel{(35)}{=} \hat{\gamma}_{a,\underline{p}} \left( \frac{\hat{\gamma}_{a,\underline{p}+1}}{\hat{\gamma}_{a,\underline{p}}} \right)^\lambda$$

$$\begin{aligned}
&\stackrel{(32)}{=} \left( \|\tilde{h}_{X_{a,p}}\|_{\tilde{k}_a} + \frac{\epsilon}{2^{\lambda+1}} \right) \left( \frac{\|\tilde{h}_{X_{a,p+1}}\|_{\tilde{k}_a} + \frac{\epsilon}{2^{\lambda+1}}}{\|\tilde{h}_{X_{a,p}}\|_{\tilde{k}_a} + \frac{\epsilon}{2^{\lambda+1}}} \right)^\lambda \\
&\leq \left( \|\tilde{h}_{X_{a,p}}\|_{\tilde{k}_a} + \frac{\epsilon}{2^{\lambda+1}} \right) \left( \frac{\epsilon + \|\tilde{h}_{X_{a,p+1}}\|_{\tilde{k}_a} 2^{\lambda+1}}{\epsilon} \right)^\lambda \\
&\stackrel{(40)}{\leq} \left( \frac{\epsilon}{2^{\lambda+1}} + \frac{\epsilon}{2^{\lambda+1}} \right) \left( \frac{\epsilon + \epsilon}{\epsilon} \right)^\lambda = \left( \frac{2\epsilon}{2^{\lambda+1}} \right) 2^\lambda = \epsilon. \quad (41)
\end{aligned}$$

Thus, analogous to Part I of the proof of Theorem 1, this yields  $p_a^* = 0$  in Proposition 2 and  $p_a = p \leq \bar{p}$  in (21), wherefore, Algorithm 4 terminates if (39) is satisfied. Thus

$$\ell_a > \frac{\mathfrak{K}^{-1} \left( \frac{\epsilon^2}{2\kappa(1+2^{\bar{p}+1})^n 2^{2\lambda+2} \|f\|_k^2} \right)}{2\sqrt{n}} \quad (42)$$

always holds. Hence, we have that

$$\ell_a \leq \mathcal{O} \left( \frac{\mathfrak{K}^{-1} \left( C \left( \frac{\epsilon}{\|f\|_k} \right)^2 \right)}{\sqrt{n}} \right) \quad \forall a \in \mathcal{A} \quad (43)$$

where  $C > 0$  is a constant that is independent of  $\epsilon$  and  $f$ .

*Part II:* From Part II of the proof of Theorem 1, we know that  $\text{card}(X) \leq (\min_{a \in \mathcal{A}} \ell_a)^{-n} (1 + 2^{\bar{p}})^n$ . With (43), it follows that:

$$\begin{aligned}
\text{card}(X) &\leq \mathcal{O} \left( \left( \frac{\sqrt{n}(1 + 2^{\bar{p}})}{\mathfrak{K}^{-1} \left( C \left( \frac{\epsilon}{\|f\|_k} \right)^2 \right)} \right)^n \right) \\
&= \mathcal{O} \left( \left( \frac{\sqrt{n}}{\mathfrak{K}^{-1} \left( C \left( \frac{\epsilon}{\|f\|_k} \right)^2 \right)} \right)^n \right). \quad (44)
\end{aligned}$$

*Part III:* The proof is analogous to Part III of the proof of Theorem 1. We determine  $\bar{\Gamma}_a$  with Assumption 5 and receive  $p_a$  from Proposition 2 and (21). The satisfaction of Inequality (1) directly follows.  $\square$

Since  $\mathfrak{K}^{-1}(0) = 0$ , we have that  $\frac{\epsilon}{\|f\|_k} \rightarrow 0$  implies  $\text{card}(X) \rightarrow \infty$ . This result is intuitive, as more complex functions (corresponding to a larger RKHS norm  $\|f\|_k$ ) or a smaller allowed approximation error  $\epsilon$  require more samples. Furthermore, the theoretical results obtained in this section show that, in the worst case, the number of samples to terminate ALKIA-X may scale exponentially with the input dimension, which is expected and widely known as the curse of dimensionality.

#### D. Complexity Bounds for the squared-exponential (SE)-Kernel

In this section, we derive more intuitive bounds on the online and offline complexity for the special case of the SE kernel, which is one of the most common kernels. The following lemma provides a bound on the smallest length scale that can occur when executing Algorithm 4 (see Part II of the proof of Theorem 2).

*Lemma 3:* Suppose that the assumptions in Theorem 2 hold and that  $k$  is the SE-kernel. Then, for any  $a \in \mathcal{A}$ , it holds that

$$\ell_a \leq \mathcal{O} \left( \frac{\epsilon}{\sqrt{n}\|f\|_k} \right) \quad \text{as } \frac{\epsilon}{\|f\|_k} \rightarrow 0. \quad (45)$$

The proof can be found in Appendix B. Moreover, the following corollary provides an upper bound on the number of samples sufficient to terminate Algorithm 4.

*Corollary 1:* Suppose that the conditions in Lemma 3 hold. Then, it holds that

$$\text{card}(X) \leq \mathcal{O} \left( \left( \frac{\sqrt{n}\|f\|_k}{\epsilon} \right)^n \right) \quad \text{as } \frac{\epsilon}{\|f\|_k} \rightarrow 0. \quad (46)$$

*Proof:* From Part II of Theorem 1, we know that  $\text{card}(X) \leq (\min_{a \in \mathcal{A}} \ell_a)^{-n} (1 + 2^{\bar{p}})^n = \mathcal{O}((\min_{a \in \mathcal{A}} \ell_a)^{-n})$ . For  $\frac{\epsilon}{\|f\|_k} \rightarrow 0$ , (45) derived in Lemma 3 provides a bound on  $\ell_a$  for all  $a \in \mathcal{A}$ , from which (46) directly follows.  $\square$

*Offline computational complexity:* Inequality (46) shows that, for a given  $\frac{\epsilon}{\|f\|_k}$ , the number of maximum samples to terminate Algorithm 4 scales exponentially with the input dimension  $n$ . Since we store all samples  $X$  to generate the approximating functions (see Algorithm 1), the memory requirements also scale exponentially with  $n$ .

*Online computational complexity:* The online evaluation with Algorithm 3 for an input  $x \in \mathcal{X}$  can be divided into three steps. First, we execute a tree search to find  $\mathcal{X}_c$  such that  $x \in \mathcal{X}_c \subseteq \mathcal{X}_a \subseteq \mathcal{X}$ . Second, we construct the covariance vector (3). Third, we evaluate the approximating function (12). Next, we elaborate on the complexity of each step.

Each subdomain  $\mathcal{X}_a \subseteq \mathcal{X}$  corresponds to a node of the resulting tree of the offline approximation. The domain  $\mathcal{X}$  corresponds to the root node and the subdomains containing the local cubes with the localized approximating functions are located in the leaf nodes. The complexity of the tree search scales linearly with the depth of the tree. Since the length scale is halved from one depth level to the next one, the maximum depth  $\bar{d} \in \mathbb{N}$  is given by

$$\bar{d} = \mathcal{O} \left( \log_2 \left( \left( \min_{a \in \mathcal{A}} \ell_a \right)^{-1} \right) \right) \stackrel{(45)}{\leq} \mathcal{O} \left( \log_2 \left( \frac{\|f\|_k \sqrt{n}}{\epsilon} \right) \right). \quad (47)$$

Hence, the computational complexity of the tree search only scales *logarithmically*. This result is akin to the logarithmic complexity of localized regression obtained in [27] and [36].

Since we use local approximations based on local cubes with  $2^n$  samples, each covariance vector (3) is  $2^n$ -dimensional. Thus, to construct the covariance vector, the kernel needs to be evaluated at  $2^n$  points.

In (12), the vector-matrix multiplication  $f_{\mathcal{X}_c}^\top K_{a, \mathcal{X}_c}^{-1}$  can be computed offline, and hence evaluating the approximating function (12) online reduces to computing  $2^n \times 2^n$  vector-vector multiplications.

To conclude, although the offline sample complexity and the memory requirements scale exponentially with the input dimension  $n$  for a given  $\frac{\|f\|_k}{\epsilon}$ , the online evaluation is characterized by operations of order  $2^n$  that are *independent* of the desired accuracy  $\epsilon$  and the ground truth  $f$ , and the tree search that scales *logarithmically* with  $\frac{\|f\|_k \sqrt{n}}{\epsilon}$ .

## V. APPROXIMATE MPC VIA ALKIA-X

In this section, we investigate the characteristics of the approximate MPC obtained using ALKIA-X. As introduced in Section II-A, the ground truth  $f$  corresponds to the MPC feedback law  $u = f(x)$  and ALKIA-X determines an explicit feedback law  $u = h(x)$  that approximates the MPC feedback  $f$  on the domain  $\mathcal{X}$ .

*Infeasible states:* ALKIA-X requires a cubic domain  $\mathcal{X}$  and the evaluation of the ground truth  $f$  for any  $x \in \mathcal{X}$ . The set  $\mathcal{X}$  should represent the state constraints, and in the standard case of box constraints, they can be shifted and scaled to yield a cubic set  $\mathcal{X}$ . The evaluation of  $f$  requires the solution of the underlying NLP, which is only defined on some feasible set  $\mathcal{X}_{\text{feas}}^f \subseteq \mathcal{X}$ . We define function evaluations for infeasible states  $x \in \mathcal{X} \setminus \mathcal{X}_{\text{feas}}^f$  by relaxing hard state and terminal constraints in the MPC formulation using slack variables and penalties [51]. Then, under suitable regularity conditions, the optimization problem returns the same solution as the original MPC for all feasible states  $x \in \mathcal{X}_{\text{feas}}^f$  (see [52]), while also providing a well-defined solution for  $x \in \mathcal{X} \setminus \mathcal{X}_{\text{feas}}^f$ .

*Feasible domain of the approximate MPC:* Although we can obtain function evaluations for the whole domain  $\mathcal{X}$  during the offline approximation, there is no need for approximating the MPC on infeasible parts of the domain  $\mathcal{X}$ . If each state within a subdomain results in a nonzero slack variable, then ALKIA-X classifies that subdomain as infeasible and does not continue the approximation process on that subdomain. Hence, ALKIA-X defines its own feasible domain  $\mathcal{X}_{\text{feas}}^h$  on which it computes an approximating function with the desired approximation accuracy  $\epsilon$ . We assume that  $\mathcal{X}_{\text{feas}}^f \subseteq \mathcal{X}_{\text{feas}}^h$  holds to ensure that the error bound  $\epsilon$  is enforced for all feasible states  $x \in \mathcal{X}_{\text{feas}}^f$ .

*Nonscalar ground truths:* In Sections III and IV, we developed ALKIA-X with its corresponding guarantees for a scalar ground truth  $f: \mathcal{X} \rightarrow \mathbb{R}$ . However, as discussed in Section II-A, the MPC feedback law  $f: \mathcal{X} \rightarrow \mathbb{R}^{n_u}$  is in general nonscalar, i.e.,  $n_u > 1$ . We can approximate such nonscalar functions by treating each dimension separately. Specifically, in Algorithm 4, we determine the sufficiently large integer  $p_a$  corresponding to the sufficient number of samples as the maximum of the sufficient number of samples required to approximate each control input dimension separately. Moreover, the localized approximating functions for each dimension are constructed (see Algorithm 1) and evaluated (see Algorithm 3) independently. Hence, ALKIA-X ensures that  $\|f(x) - h(x)\|_\infty \leq \epsilon$  for all  $x \in \mathcal{X}_{\text{feas}}^f \subseteq \mathcal{X}_{\text{feas}}^h$ .

*Closed-loop guarantees:* We provide closed-loop guarantees on the approximate MPC by combining a robust MPC design with the guaranteed approximation error (see Fig. 1). To this end, consider a disturbance set  $\mathcal{W} \subseteq \mathbb{R}^n$  that upper bounds the effect of inexactly approximating the control input, i.e.,  $\mathcal{W} \supseteq \{g(x, \tilde{u}) - g(x, u) \mid x \in \mathcal{X}, u \in \mathcal{U}, \|u - \tilde{u}\|_\infty \leq \epsilon\}$ , see, e.g., constructions in [8] and [9].

*Corollary 2:* Let  $f$  be according to the robust MPC design in [21, Problem (30)] with disturbance bound  $\mathcal{W}$  and suppose that the conditions in [21, Thm. 2] regarding  $f, g, \mathcal{X}, \mathcal{U}$ , and  $\mathcal{W}$  hold.<sup>6</sup> Suppose further that the conditions in Theorem 2 hold

<sup>6</sup>As discussed above,  $f$  is the robust MPC design with constraints relaxed using penalties to ensure that the function is defined for all  $x \in \mathcal{X}$ . We assume that the conditions in [52] hold, ensuring that  $f(x)$  coincides with the minimizer of [21, Problem (30)] for all  $x \in \mathcal{X}_{\text{feas}}^f$ .

and that  $h$  is constructed according to Algorithm 4, where we approximate each dimension independently. Then, for any initial condition  $x_0 \in \mathcal{X}_{\text{feas}}^f$ , the closed-loop system  $x_{t+1} = g(x_t, u_t)$ ,  $u_t = h(x_t)$ , ensures the following:

- 1) recursive feasibility  $x_t \in \mathcal{X}_{\text{feas}}^f \subseteq \mathcal{X}_{\text{feas}}^h \forall t \in \mathbb{N}$ ;
- 2) constraint satisfaction  $x_t \in \mathcal{X}, u_t \in \mathcal{U} \forall t \in \mathbb{N}$ ;
- 3) and practical<sup>7</sup> asymptotic stability of  $x = x_s$ .

*Proof:* Theorem 2 ensures that each dimension of the approximating function  $h$ , which we receive independently from Algorithm 4, satisfies Inequality (1), i.e., the satisfaction of  $\|f(x) - h(x)\|_\infty \leq \epsilon$  is ensured for all  $x \in \mathcal{X}_{\text{feas}}^h$ . Hence, the closed-loop system satisfies  $x_{t+1} = g(x_t, h(x_t)) =: g(x_t, f(x_t)) + w_t$  with  $w_t \in \mathcal{W}$  for all  $x \in \mathcal{X}_{\text{feas}}^f \subseteq \mathcal{X}_{\text{feas}}^h$ . Thus, the closed-loop guarantees in [21, Thm. 2] with respect to disturbances  $w_t \in \mathcal{W}$  apply.  $\square$

These theoretical guarantees are equally applicable if any other nonlinear MPC scheme is approximated that is designed to be robust with respect to disturbances  $w_t \in \mathcal{W}$ , see, e.g., [20], [21], [22], and [23] for corresponding designs.

*Computational complexity:* In the complexity analysis in Section IV-D, we showed that the offline computational complexity of ALKIA-X scales exponentially with the number of states  $n$ . In contrast, the computational complexity of classical explicit MPC approaches scales with the number of states, the number of constraints, and the prediction horizon [25, Sect. 3]. Hence, a major benefit of approximating MPC schemes using ALKIA-X is the fact that the computational complexity does *not* directly increase with the prediction horizon nor the number of constraints used in the MPC formulation. Moreover, the online computational complexity of classical kernel-based methods scales cubically in the number of samples [34, Sect. 2.3]. In contrast, the online computational complexity of ALKIA-X only scales *logarithmically* with  $\frac{\|f\|_k \sqrt{n}}{\epsilon}$ . Thus, the resulting approximate MPC is fast-to-evaluate, and hence suitable to control nonlinear systems that require high sampling rates with relatively cheap hardware (P1).

*Practical considerations:* The theoretical analysis assumes that we obtain the optimal solution to the NLP and that this function is continuous. However, in practical application, the solver may return *local minima*, only solve to a specified tolerance, or the solution may even be discontinuous [53]. Discontinuities and local minima can be an issue as the finite termination guarantees (see Theorem 2) rely on continuity, i.e., it could theoretically happen that the algorithm partitions the domain endlessly. Notably, we have not observed this behavior in any numerical example since the states approach the same warmstarts in the solvers for smaller subdomains.

## VI. NUMERICAL EXPERIMENTS

In this section, we approximate two nonlinear MPC schemes using ALKIA-X. In Section VI-A, we apply ALKIA-X on an academic example and compare its performance with existing work. In Section VI-B, we demonstrate the practicability of ALKIA-X by approximating an MPC corresponding to a real-world application for which fast sampling rates are required and computational capacity is limited. ALKIA-X and both MPCs are implemented in Python. The MPC schemes are formulated

<sup>7</sup>Practical asymptotic stability implies convergence to a neighborhood of the steady-state  $x_s$ . In this case, the size of the neighborhood depends on the magnitude of the approximation error  $\max_{x \in \mathcal{X}_{\text{feas}}^f} \|f(x) - h(x)\|_\infty \leq \epsilon$ .



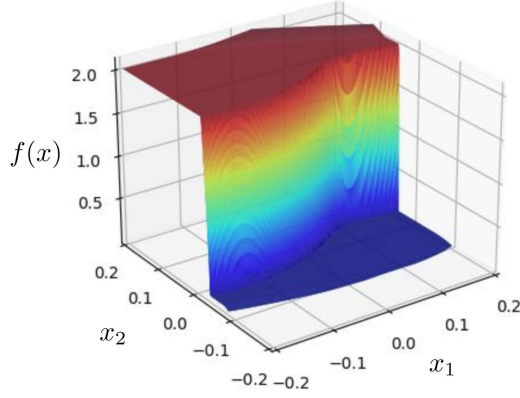


Fig. 5. Ground truth MPC feedback law  $f$  over the domain  $\mathcal{X} = [-0.2, 0.2]^2$ .

using CasADi [54] and the underlying NLPs are solved with IPOPT [55] using just-in-time compilation. We conducted the offline approximations on a Linux server with 32 GB random-access memory (RAM) parallelized on eight cores. The online evaluation was performed using an Ubuntu laptop with 32 GB RAM and an Intel Core i7-12700H processor. We used the Matérn kernel with  $\nu = \frac{3}{2}$  and a length scale of 0.8 for both experiments.

#### A. Continuous Stirred Tank Reactor

We consider a nonlinear continuous stirred tank reactor with

$$g(x, u) = \begin{pmatrix} x_1 + \delta \left( \frac{1-x_1}{\theta} - \hat{k}x_1e^{-\frac{M}{x_1}} \right) \\ x_2 + \delta \left( \frac{x_f-x_2}{\theta} + \hat{k}x_1e^{-\frac{M}{x_2}} - \hat{\alpha}u(x_2-x_c) \right) \end{pmatrix} \quad (48)$$

where  $x_1$  is the temperature,  $x_2$  is the concentration,  $u$  is the coolant flow, and  $\delta = 0.1$  s is the sampling time; see [8] for details. The nonlinear MPC is formulated with a horizon of  $N = 180$ , the input constraint is  $\mathcal{U} = [0, 2]$ , and we obtain a cubic domain  $\mathcal{X} = [-0.2, 0.2]^2$  after shifting, as done in [8].<sup>8</sup> Fig. 5 shows the corresponding MPC feedback law  $f$  over the domain  $\mathcal{X} = [-0.2, 0.2]^2$ .

We use ALKIA-X to determine an approximate MPC to control system (48) with high sampling rates and closed-loop guarantees. To this end, we consider the MPC scheme in [8], which is designed to be robust with respect to input disturbances bounded by  $\epsilon = 5.1 \cdot 10^{-3}$ . To compute the approximate MPC, ALKIA-X requires samples, i.e., solutions of the NLP of the ground truth MPC  $f$  for different states. In this experiment, the hyperparameters were  $\underline{p} = 2$  and  $\bar{p} = 5$ , yielding an upper bound on the condition number of  $\bar{\kappa} = 1.14 \cdot 10^8$  [see (22) and (31)].

**Result:** We approximate the ground truth MPC feedback law  $f$  using ALKIA-X with the error bound  $\epsilon = 5.1 \cdot 10^{-3}$ . We heuristically validate the approximation error by evaluating the error for  $90 \cdot 10^3$  equidistant states on the domain  $\mathcal{X}$ , yielding a maximum error of  $2.46 \cdot 10^{-3}$ , i.e., about 50% smaller than the guaranteed error bound  $\epsilon$ .

Fig. 6 depicts the subdomain partitioning of ALKIA-X, the feasible domain  $\mathcal{X}_{\text{feas}}^f$  of the MPC, and  $\mathcal{X}_{\text{feas}}^h$ , the feasible domain

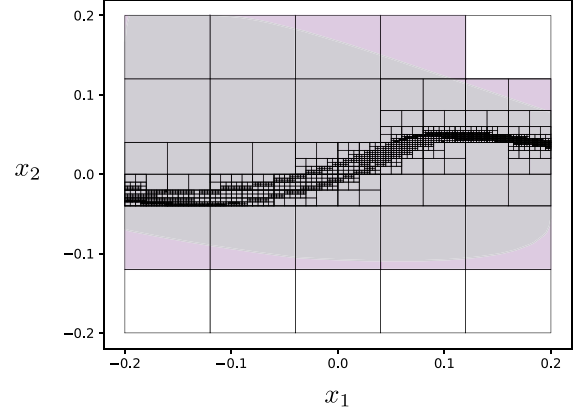


Fig. 6. Subdomain partitioning using ALKIA-X and feasible domains. The light gray shaded area corresponds to the feasible domain  $\mathcal{X}_{\text{feas}}^f$  of the ground truth MPC, whereas the area enclosed by the magenta shading corresponds to the feasible domain  $\mathcal{X}_{\text{feas}}^h$  of the obtained approximate MPC via ALKIA-X. The resulting partitioning is reminiscent of classical linear explicit MPC approaches [26], [27]. In total, we have 1996 subdomains.

TABLE I  
COMPARISON BETWEEN ALKIA-X AND THE NN APPROXIMATION IN [8] WITH THE CONTINUOUS STIRRED TANK REACTOR

	ALKIA-X	NN approach [8]
$t_{\text{online}}$	$44.02 \mu\text{s} \pm 0.14 \mu\text{s}$	$3000 \mu\text{s}$
$t_{\text{offline}}$	10.3 h	500 h
$\text{card}(X)$	$1.56 \cdot 10^6$	$1.6 \cdot 10^6$

computed by ALKIA-X. Recall that ALKIA-X partitions subdomains further if the maximum number of samples cannot ensure the desired approximation error  $\epsilon$ . This behavior can be observed in Fig. 6, where larger subdomains correspond to regions where the ground truth resembles a constant function (see Fig. 5), and hence fewer samples are required. Furthermore, the derived sample complexities in Theorem 2 and Corollary 1 also support this statement.<sup>9</sup> Overall, ALKIA-X successfully approximates the ground truth MPC  $f$ , yielding an approximate MPC  $h$  with closed-loop guarantees on stability and constraint satisfaction.

**Discussion:** In the following, we compare the MPC approximation obtained using ALKIA-X with the approach presented in [8]. Both works approximate the same ground truth MPC [see (48) and Fig. 5] with the identical error bound  $\epsilon$ . ALKIA-X is a kernel-based algorithm that inherently guarantees the satisfaction of any desired error bound  $\epsilon > 0$  [see Theorem 2, (P2), and (P3)]. In contrast, the approach in [8] uses NNs for the approximation and an additional statistical validation step to guarantee the desired approximation accuracy. Notably, the approach in [8] may lead to an *iterative* offline design between NN training and validation, e.g., requiring additional sampling or changes to the NN architecture whenever the statistical validation step fails.

Table I compares the number of samples  $\text{card}(X)$ , the offline approximation time  $t_{\text{offline}}$ , and the online evaluation time  $t_{\text{online}}$

<sup>9</sup>The sample complexity on subdomain  $\mathcal{X}_a$  increases with the RKHS norm  $\|\tilde{f}_a\|_{\tilde{\kappa}_a}$  (see Theorem 2, Corollary 1). For constant functions, we have  $\|\tilde{f}_a\|_{\tilde{\kappa}_a} = 0$ .

<sup>8</sup>We further shift and scale the domain  $\mathcal{X}$  to yield the unit cube  $[0, 1]^2$ .

of ALKIA-X and the NN approach proposed in [8].<sup>10</sup> The on-line evaluation time of ALKIA-X is obtained over three runs of evaluating  $90 \cdot 10^3$  equidistant states.

Both ALKIA-X and the NN approach require roughly  $\text{card}(X) \approx 10^6$  samples. However, the offline approximation time  $t_{\text{offline}}$  of [8] is significantly longer. The main reason for this discrepancy is that the NN approach requires further samples for the statistical validation step. In addition, ALKIA-X yields an almost 100-times faster<sup>11</sup> online evaluation time  $t_{\text{online}}$  compared to the NN approach. The fast online evaluation of ALKIA-X (P1) is due to the *localized* kernel interpolation approach (T1), whereas standard kernel-based methods tend to suffer from scalability issues [34, Sect. 2.3].

In addition to the presented numerical differences, the closed-loop guarantees on the approximate MPC using ALKIA-X are *deterministic*. Furthermore, both the sample acquisition and the offline approximation of ALKIA-X are entirely parallelized, thus accelerating the offline approximation. Besides the favorable offline approximation time, the offline approximation with ALKIA-X is *automatic* and *noniterative* (P4) in contrast to the iterative offline design in [8].

### B. Cold Atmospheric Plasma

We consider the MPC scheme to control a cold atmospheric plasma device from [46, Ch. 4.5]. The nonlinear systems dynamics  $g: \mathbb{R}^3 \rightarrow \mathbb{R}^2$  are described by

$$g(x, u) = \begin{pmatrix} 0.427x_1 + 0.68x_2 + 1.58u_1 - 1.02u_2 \\ -0.06x_1 + 0.26x_2 + 0.73u_1 + 0.03u_2 \\ x_3 + \mathbb{K}^{(\mathbb{T}-x_1)}\delta \end{pmatrix} \quad (49)$$

where  $x_1$  is the surface temperature,  $x_2$  is the gas temperature, and  $x_3$  is the thermal dose delivered to the target surface. The inputs  $u_1$  and  $u_2$  correspond to the applied power to the plasma and the flow rate of Argon, respectively, while  $\mathbb{K}$  and  $\mathbb{T}$  are constants and  $\delta = 0.5\text{ s}$  is the sampling time. Note that states and inputs are appropriately shifted in (49), see [46, Ch. 4.5]. The state constraints<sup>8</sup> are  $x_1 \in [25^\circ\text{C}, 42.5^\circ\text{C}]$ ,  $x_2 \in [20^\circ\text{C}, 80^\circ\text{C}]$ , and  $x_3 \in [0\text{ min}, 11\text{ min}]$ , whereas the input constraints are  $u_1 \in [1.5\text{ W}, 8\text{ W}]$  and  $u_2 \in [1\text{ slm}, 6\text{ slm}]$ . The control objective is to achieve a treatment time of the thermal dose of  $\bar{x}_3 = 10\text{ min}$  and satisfy the state and input constraints. The nonlinear MPC is formulated with a horizon of  $N = 120$ .

As mentioned in [46, Sects. 1.2.2 and 6.2.2], approximate MPC schemes are desired for this application due to the need for 1) subsecond sampling rates and 2) implementability on portable embedded devices. To this end, we use ALKIA-X to determine an approximate MPC to control system (49). In this application example, we focus on obtaining an implementable controller for relatively cheap hardware with limited memory capacities. Hence, instead of ensuring a desired error bound  $\epsilon$ , we here focus on achieving the best approximation subject to a memory

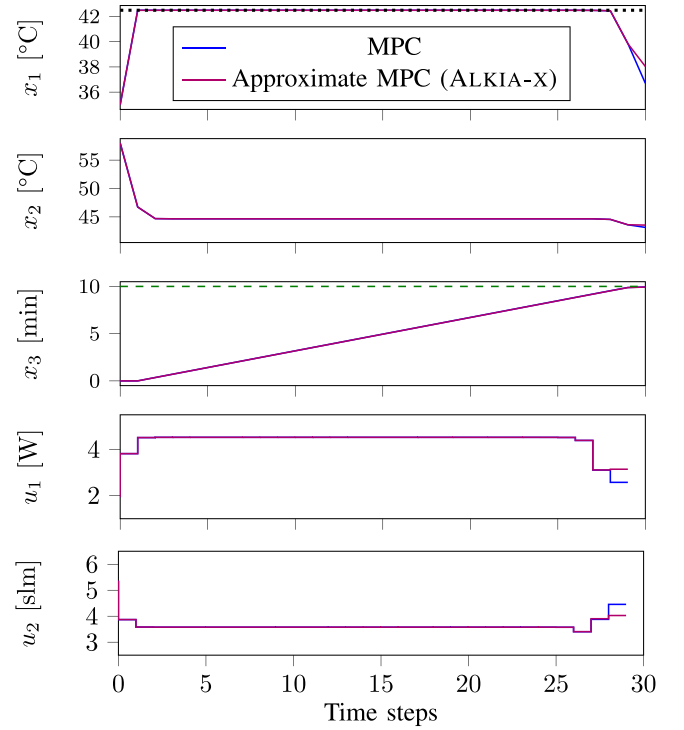


Fig. 7. Closed-loop state and input trajectories of the MPC (blue lines) and the approximate MPC (magenta lines) computed by ALKIA-X. The dotted black line shows the constraint on  $x_1$ , whereas the dashed green line depicts the desired treatment time of  $\bar{x}_3 = 10\text{ min}$ .

TABLE II  
COMPARISON BETWEEN THE APPROXIMATE MPC COMPUTED BY ALKIA-X AND THE MPC WITH THE COLD ATMOSPHERIC PLASMA DEVICE

	Approximate MPC (ALKIA-X)	MPC (IPOPT)
$t_{\text{online}}$	$96\ \mu\text{s} \pm 4\ \mu\text{s}$	$507\text{ ms} \pm 6\text{ ms}$
$t_{\text{offline}}$	69 h	[-]
$\text{card}(X)$	$8.22 \cdot 10^5$	[-]

requirement on the samples of 75 MB, which is compatible with the available memory of standard microcontrollers. We fulfill the memory requirement by upper bounding the maximum depth  $\bar{d}$  [see (47)] a priori. Specifically, for this application, we prepartitioned the domain into  $3^3$  subdomains and obtained  $\bar{d} = 1$ , i.e., each subdomain can at most be partitioned into  $2^3$  further subdomains. The hyperparameters for this experiment were  $\underline{p} = 2$  and  $\bar{p} = 4$ , yielding an upper bound on the condition number of  $\bar{\kappa} = 3.32 \cdot 10^7$  [see (22) and (31)].

**Result:** Table II compares the approximate MPC computed by ALKIA-X (Algorithm 4) and the MPC using online optimization (IPOPT) in terms of online evaluation time  $t_{\text{online}}$ , offline approximation time  $t_{\text{offline}}$ , and number of samples  $\text{card}(X)$ . The documented online evaluation times are obtained over three closed-loop runs of 30 time steps from the initial state  $x = [35^\circ\text{C}, 58^\circ\text{C}, 0\text{ min}]^\top$ , and the NLP of the MPC is solved using a warmstart. ALKIA-X requires a long time  $t_{\text{offline}}$  for the offline approximation,<sup>12</sup> whereas the MPC only runs the just-in-time

<sup>10</sup>Table I directly contains the computation times reported in [8], which are not obtained with the same hardware as ALKIA-X.

<sup>11</sup>More recently, [5] approximated the same ground truth MPC  $f$  with NNs, achieving an online evaluation time of  $t_{\text{online}} = 500\ \mu\text{s}$ , which is still ten-times slower than the proposed solution. Furthermore, additional parallelization with  $10^3$  cores and the usage of graphics processing units (GPUs) reduced the offline sampling time to  $t_{\text{offline}} = 0.5\text{ h}$ , resulting in  $t_{\text{offline}} = 4.5\text{ h}$  overall, including the NN training. However, in contrast to the proposed approximation method, [5] does not guarantee error bound  $\epsilon$  but utilizes a fallback policy to obtain closed-loop guarantees.

<sup>12</sup>Note that, due to parallelization, the offline approximation time can directly be reduced to  $t_{\text{offline}} \approx 2.5\text{ h}$  by increasing the number of cores from 8 to 216, provided sufficient RAM.

compilation prior to deployment. The benefit of the approximate MPC is the significant speedup of the online evaluation time  $t_{\text{online}}$  by a factor of over  $10^3$ .

Fig. 7 illustrates the closed-loop trajectories of the MPC, where the NLP is solved using IPOPT, and of the approximate MPC computed by ALKIA-X. As in [46, Ch. 4.5], closed-loop simulations are terminated as soon as  $x_3 \geq \bar{x}_3$ . The trajectories are very similar with  $x_3$  showing no noticeable visual difference and all constraints are satisfied. The difference in control input at the end of the simulation is due to the accumulated difference in the closed-loop state sequences of the two controllers, while the approximation error evaluated on the same state trajectory satisfies  $\max_t \|f(x_t) - h(x_t)\|_\infty \leq 3 \cdot 10^{-3}$ .

Overall, ALKIA-X (1) automatically computes an approximate MPC (P4) that closely resembles the MPC, (2) reduces the online evaluation time by a factor of over  $10^3$  (P1), and (3) only requires 33 MB of memory, thus being implementable on most microcontrollers.

## VII. CONCLUSION

We addressed the problem of automatically approximating nonlinear MPC schemes while ensuring that the resulting approximate MPC inherits desirable closed-loop properties on stability and constraint satisfaction. By considering a robust MPC scheme, we reduced this problem to a function approximation problem, which we tackled by proposing ALKIA-X. ALKIA-X is an automatic and reliable algorithm that yields a computationally efficient approximate MPC with closed-loop guarantees. We successfully applied ALKIA-X to approximate two nonlinear MPC schemes, (1) demonstrating reduced offline computation and online evaluation time by over one order of magnitude and (2) showcasing applicability to realistic problems with limited memory capacities.

Although ALKIA-X is tailored to approximate MPC schemes, the algorithm is capable of automatically approximating a wide range of unknown functions with guaranteed bounds on the approximation error. Future work includes improving the scalability of ALKIA-X to higher dimensions and further investigations of the RKHS norm extrapolation.

## ACKNOWLEDGMENT

The authors would like to thank D. Baumann for helpful comments and J. Sieber for setting up the server.

## APPENDIX A PROOF OF LEMMA 2

In Part I, we show that, for any  $a \in \mathcal{A}$ , (33) interpolates  $\hat{\gamma}_{a,p}$  for  $p \in \{p, p+1\}$ , i.e., (34) holds. In Part II, we prove that  $\bar{\Gamma}_a \in (0, \infty)$  holds for any  $a \in \mathcal{A}$ .

*Part I:* Consider any  $a \in \mathcal{A}$ . First, we show  $\gamma_a(p) = \hat{\gamma}_{a,p}$ . It holds that

$$\begin{aligned} \gamma_a(p) &\stackrel{(33)}{=} \bar{\Gamma}_a \exp\left(\frac{-\tau_a}{1+2^p}\right) \\ &\stackrel{(35)}{=} \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(\frac{-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) \lambda (1+2^p)}{1+2^p}\right) \end{aligned}$$

$$\begin{aligned} &= \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) \lambda\right) \\ &= \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \left(\frac{\hat{\gamma}_{a,p}}{\hat{\gamma}_{a,p+1}}\right)^\lambda = \hat{\gamma}_{a,p}. \end{aligned}$$

Second, we show  $\gamma_a(p+1) = \hat{\gamma}_{a,p+1}$ . It holds that

$$\begin{aligned} \gamma_a(p+1) &\stackrel{(33)}{=} \bar{\Gamma}_a \exp\left(\frac{-\tau_a}{1+2^{p+1}}\right) \\ &\stackrel{(35)}{=} \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(\frac{-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) \lambda (1+2^p)}{1+2^{p+1}}\right) \\ &\stackrel{(32)}{=} \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(\frac{-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) \lambda (1+2^p)}{2^p \lambda}\right) \\ &= \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) (1+2^{-p})\right) \\ &\stackrel{(32)}{=} \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \exp\left(-\ln\left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) (\lambda-1)\right) \\ &= \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right)^\lambda \left(\frac{\hat{\gamma}_{a,p}}{\hat{\gamma}_{a,p+1}}\right)^{\lambda-1} \\ &= \hat{\gamma}_{a,p} \left(\frac{\hat{\gamma}_{a,p+1}}{\hat{\gamma}_{a,p}}\right) = \hat{\gamma}_{a,p+1} \end{aligned}$$

i.e., (34) holds.

*Part II:* From (33), we have that  $\bar{\Gamma}_a \in (0, \infty)$  if  $\hat{\gamma}_{a,p} > 0$ ,  $p \in \{p, p+1\}$ , and  $\hat{\gamma}_{a,p+1} < \infty$ . The fact that  $\hat{\gamma}_{a,p} > 0$  and  $p \in \{p, p+1\}$  directly follows from (32) with  $\epsilon > 0$ . Moreover, from (33), it follows that  $\hat{\gamma}_{a,p+1} < \infty$  holds if  $\|\tilde{h}_{X_{a,p+1}}\|_{\tilde{k}_a} < \infty$  with  $\epsilon < \infty$ . Since the covariance matrix is always strictly positive definite (see Section II-B) and we have a continuous ground truth  $f$  on a bounded set  $\mathcal{X}$  (see Lemma 1),  $\|\tilde{h}_{X_{a,p+1}}\|_{\tilde{k}_a} < \infty$  follows from (30).  $\square$

## APPENDIX B PROOF OF LEMMA 3.

For the SE-kernel  $\tilde{k}_{\text{SE}}: \mathbb{R}_{\geq 0} \rightarrow [0, 1]$  with  $\tilde{k}_{\text{SE}}(x) = \exp(-x^2)$  [34], we have  $\mathfrak{K}_{\text{SE}}(x) = 1 - \exp(-x^2)$  [see (9)] and  $\mathfrak{K}_{\text{SE}}^{-1}: [0, 1] \rightarrow \mathbb{R}_{\geq 0}$  with

$$\mathfrak{K}_{\text{SE}}^{-1}(x) = \sqrt{\ln\left(\frac{1}{1-x}\right)}. \quad (50)$$

With L'Hôpital's rule, it follows that:

$$\frac{\ln\left(\frac{1}{1-x}\right)}{x} = \frac{1}{1-x} = 1, \quad \text{as } x \rightarrow 1. \quad (51)$$



Hence, (50) and (51) yield

$$\mathcal{R}_{SE}^{-1}(x) = \sqrt{x}, \quad \text{as } x \rightarrow 1. \quad (52)$$

Thus, with (43), it holds for any  $a \in \mathcal{X}_a$  that

$$\ell_a \leq \mathcal{O}\left(\frac{\sqrt{C}\epsilon}{\sqrt{n}\|f\|_k}\right) \quad \text{as } \frac{\epsilon}{\|f\|_k} \rightarrow 0$$

which yields (45).  $\square$

## REFERENCES

- [1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Madison, WI, USA: Nob Hill Publishing, 2017.
- [2] S. Chen et al., "Approximating explicit model predictive control using constrained neural networks," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 1520–1527.
- [3] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3866–3878, Sep. 2020.
- [4] S. W. Chen, T. Wang, N. Atanasov, V. Kumar, and M. Morari, "Large scale model predictive control with neural networks and primal active sets," *Automatica*, vol. 135, 2022, Art. no. 109947.
- [5] H. Hose, J. Köhler, M. N. Zeilinger, and S. Trimpe, "Approximate nonlinear model predictive control with safety-augmented neural networks," 2023, *arXiv:2304.09575*.
- [6] X. Zhang, M. Bujarbaruah, and F. Borrelli, "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 5, pp. 2102–2114, Sep. 2021.
- [7] B. Karg, T. Alamo, and S. Lucia, "Probabilistic performance validation of deep learning-based robust NMPC controllers," *Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 8855–8876, 2021.
- [8] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Contr. Syst. Lett.*, vol. 2, no. 3, pp. 543–548, Jul. 2018.
- [9] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3050–3057, Apr. 2020.
- [10] G. Pin, M. Filippio, F. Pellegrino, G. Fenu, and T. Parisini, "Approximate model predictive control laws for constrained nonlinear discrete-time systems: Analysis and offline design," *Int. J. Control*, vol. 86, no. 5, pp. 804–820, 2013.
- [11] T. Parisini, M. Sanguineti, and R. Zoppoli, "Nonlinear stabilization by receding-horizon neural regulators," *Int. J. Control*, vol. 70, no. 3, pp. 341–362, 1998.
- [12] C. Gonzalez, H. Asadi, L. Kooijman, and C. P. Lim, "Neural networks for fast optimisation in model predictive control: A review," 2023, *arXiv:2309.02668*.
- [13] M. Canale, L. Fagiano, and M. Milanese, *Fast Nonlinear Model Predictive Control via set Membership Approximation: An Overview*, vol. 12. Berlin, Germany: Springer, 2009, pp. 461–470.
- [14] M. Canale, L. Fagiano, and M. Milanese, "Efficient model predictive control for nonlinear systems via function approximation techniques," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1911–1916, Aug. 2010.
- [15] M. Canale, L. Fagiano, and M. Signorile, "Nonlinear model predictive control from data: A set membership approach," *Int. J. Robust Nonlinear Control*, vol. 24, no. 1, pp. 123–139, 2014.
- [16] M. Boggio, C. Novara, and M. Taragna, "Set membership based nonlinear model predictive control," 2022, *arXiv:2212.12414*.
- [17] S. Ganguly and D. Chatterjee, "Explicit feedback synthesis driven by quasi-interpolation for nonlinear robust model predictive control," *IEEE Trans. Autom. Control*, 2025.
- [18] M. Binder, G. Darivianakis, A. Eichler, and J. Lygeros, "Approximate explicit model predictive controller using Gaussian processes," in *Proc. 58th Conf. Decis. Control*, 2019, pp. 841–846.
- [19] A. Rose, M. Pfefferkorn, H. H. Nguyen, and R. Findeisen, "Learning a Gaussian process approximation of a model predictive controller with guarantees," in *Proc. 62nd Conf. Decis. Control*, 2023, pp. 4094–4099.
- [20] B. Houska and M. E. Villanueva, *Robust Optimization for MPC*. Cambridge, MA, USA: Birkhäuser, 2019, pp. 413–443.
- [21] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 794–801, Feb. 2021.
- [22] A. Sasfi, M. N. Zeilinger, and J. Köhler, "Robust adaptive MPC using control contraction metrics," *Automatica*, vol. 155, 2023, Art. no. 111169.
- [23] S. V. Raković, L. Dai, and Y. Xia, "Homothetic tube model predictive control for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 4554–4569, Aug. 2023.
- [24] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.
- [25] A. Alessio and A. Bemporad, *A Survey on Explicit Model Predictive Control*. Berlin, Germany: Springer, 2009, pp. 345–369.
- [26] T. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 810–815, May 2003.
- [27] S. Summers, C. N. Jones, J. Lygeros, and M. Morari, "A multiresolution approximation method for fast explicit model predictive control," *IEEE Trans. Autom. Control*, vol. 56, no. 11, pp. 2530–2541, Nov. 2011.
- [28] A. Grancharova and T. A. Johansen, *Explicit Nonlinear Model Predictive Control: Theory and Applications*. Berlin, Germany: Springer, 2012.
- [29] P. Scharnhorst, E. T. Maddalena, Y. Jiang, and C. N. Jones, "Robust uncertainty bounds in reproducing kernel Hilbert spaces: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2848–2861, May 2023.
- [30] E. T. Maddalena, P. Scharnhorst, and C. N. Jones, "Deterministic error bounds for kernel-based learning techniques under bounded noise," *Automatica*, vol. 134, 2021, Art. no. 109896.
- [31] I. Steinwart and A. Christmann, *Support Vector Machines*. Berlin, Germany: Springer, 2008.
- [32] G. M. Fasshauer and M. J. McCourt, *Kernel-Based Approximation Methods Using MATLAB*, vol. 19. Singapore: World Scientific, 2015.
- [33] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, "Gaussian processes and kernel methods: A review on connections and equivalences," 2018, *arXiv:1807.02582*.
- [34] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [35] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets Big Data: A review of scalable GPs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020.
- [36] A. Lederer, A. J. O. Conejo, K. A. Maier, W. Xiao, J. Umlauf, and S. Hirche, "Gaussian process-based real-time learning for safety critical applications," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 6055–6064.
- [37] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning and control," in *Proc. 21st Adv. Neural Inf. Process. Syst.*, 2008, vol. 21, pp. 1193–1200.
- [38] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *Mach. Learn.*, vol. 112, no. 10, pp. 3713–3747, 2023.
- [39] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.
- [40] S. R. Chowdhury and A. Gopalan, "On Kernelized multi-armed bandits," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 844–853.
- [41] C. Fiedler, C. W. Scherer, and S. Trimpe, "Practical and rigorous uncertainty bounds for Gaussian process regression," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 7439–7447.
- [42] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 997–1005.
- [43] K. Hashimoto, A. Saoud, M. Kishida, T. Ushio, and D. V. Dimarogonas, "Learning-based symbolic abstractions for nonlinear control systems," *Automatica*, vol. 146, 2022, Art. no. 110646.
- [44] J.-P. Calliess, "Conservative decision-making and inference in uncertain dynamical systems," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 2014.
- [45] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.
- [46] A. D. Bonzanini, "Safe and fast learning-based model predictive control of nonlinear systems with applications to cold atmospheric plasma," Ph.D. dissertation, Univ. California, Berkeley, Berkeley, CA, USA, 2022.
- [47] H. K. Khalil, *Nonlinear Systems*, vol. 115. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

- [48] T. Kailath, "RKHS approach to detection and estimation problems—Part I: Deterministic signals in Gaussian noise," *IEEE Trans. Inf. Theory*, vol. TIT-17, no. 5, pp. 530–549, Sep. 1971.
- [49] T. Kailath, R. Geesey, and H. Weinert, "Some relations among RKHS norms, Fredholm equations, and innovations representations," *IEEE Trans. Inf. Theory*, vol. TIT-18, no. 3, pp. 341–348, May 1972.
- [50] H. Pradhan, A. Koppel, and K. Rajawat, "On submodular set cover problems for near-optimal online kernel basis selection," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 4168–4172.
- [51] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *Proc. UKACC Int. Conf. Control*, 2000, pp. 2319–2327.
- [52] E. Rosenberg, "Exact penalty functions and stability in locally Lipschitz programming," *J. Optim. Theory Appl.*, vol. 30, pp. 340–356, 1984.
- [53] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel, "Examples when nonlinear model predictive control is nonrobust," *Automatica*, vol. 40, no. 10, pp. 1729–1738, 2004.
- [54] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi—A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.
- [55] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, 2006.



Finland. His research interests include machine learning and control theory.

**Abdullah Tokmak** received the B.Sc. and M.Sc. degrees in mechanical engineering from RWTH Aachen University, Aachen, Germany, in 2020 and 2022, respectively.

He conducted his master's thesis research with ETH Zürich in 2022. In 2023, he received the Springorium Commemorative Coin for the master's degree and the Friedrich-Wilhelm Award for his Master's thesis, both from RWTH Aachen University. He is currently a Doctoral Researcher with Aalto University, Espoo,



to RWTH Aachen University in 2021. During his studies, he was supported by the Max Weber program. His research interests include the intersection of control theory and machine learning, with a particular focus on kernel methods.

**Christian Fiedler** received the B.Sc. degree in mathematics from the University of Bayreuth, Bayreuth, Germany, in 2015, the M.Phil. degree in computational biology from the University of Cambridge, Cambridge, U.K., in 2017, and the M.Sc. degree in mathematics from the Technical University of Munich, Munich, Germany, in 2019.

In 2019, he joined the Max Planck Institute for Intelligent Systems and the University of Stuttgart as a Ph.D. student, before transferring



**Melanie N. Zeilinger** (Member, IEEE) received the Diploma degree in engineering cybernetics from the University of Stuttgart, Stuttgart, Germany, in 2006, and the Ph.D. (with Hons.) degree in electrical engineering from ETH Zurich, Zurich, Switzerland, in 2011.

She is currently an Associate Professor with ETH Zürich, Zurich, Switzerland. From 2011 to 2012, she was a Postdoctoral Fellow with the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. She was a

Marie Curie Fellow and Postdoctoral Researcher with the Max Planck Institute for Intelligent Systems, Tübingen, Germany, until 2015 and with the Department of Electrical Engineering and Computer Sciences at the University of California at Berkeley, CA, USA, from 2012 to 2014. From 2018 to 2019, she was a Professor with the University of Freiburg, Freiburg im Breisgau, Germany. Her research focuses on the learning-based control with applications to robotics and human-in-the-loop control.

Dr. Zeilinger was the recipient of the ETH medal for her PhD thesis, an SNF Professorship, the ETH Golden Owl for exceptional teaching in 2022, and the European Control Award in 2023.



**Sebastian Trimpe** (Member, IEEE) received the B.Sc. degree in general engineering science and the M.Sc. (Dipl.-Ing.) degree in electrical engineering from the Hamburg University of Technology, Hamburg, Germany, in 2005 and 2007, respectively, and the Ph.D. (Dr. sc.) degree in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2013.

Since 2020, he has been a Full Professor with RWTH Aachen University, Aachen, Germany, where he heads the Institute for Data Science

in Mechanical Engineering. Before, he was an independent Research Group Leader with the Max Planck Institute for Intelligent Systems in Stuttgart and Tübingen, Germany. His main research interests include systems and control theory, machine learning, networked systems, and robotics.

Dr. Trimpe was the recipient of several awards, including the triennial IFAC World Congress Interactive Paper Prize (2011), the Klaus Tschira Award for achievements in public understanding of science (2014), the Best Paper Award of the International Conference on Cyber-Physical Systems (2019), and the Future Prize by the Ewald Marquardt Stiftung for innovations in control engineering (2020).



**Johannes Köhler** (Member, IEEE) received the master's degree in engineering cybernetics and the Ph.D. degree in mechanical engineering from the University of Stuttgart, Stuttgart, Germany, in 2017 and 2021, respectively.

He is currently a Postdoctoral Researcher with ETH Zürich, Zürich, Switzerland. His research interests include model predictive control and control and estimation for nonlinear uncertain systems.

Dr. Köhler was the recipient of the 2021 European Systems & Control PhD award.