2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Learning a Gaussian Process Approximation of a Model Predictive Controller with Guarantees

Alexander Rose[1], Maik Pfefferkorn[1,2], Hoang Hai Nguyen[1], Rolf Findeisen[1]

Abstract— **Model predictive control effectively handles complex dynamical systems with constraints, but its high computational demand often makes real-time application infeasible. We propose using Gaussian process regression to learn an approximation of the controller offline for online use. Our approach incorporates a robust predictive control scheme and provides bounds on approximation errors to ensure recursive feasibility and input-to-state stability. Exploiting a sampling-based scenario approach, we develop an efficient sampling strategy and guarantee that, with high probability, the approximation error remains within acceptable bounds. Our method demonstrates enhanced efficiency and reduced computational demand in an example application.**

I. Introduction

Nowadays, autonomous systems are used in a wide variety of applications, where they need to reliably satisfy safety-critical constraints and achieve performance requirements. One way of dealing with these challenging tasks is to deploy model predictive control (MPC), a nowadays widely used, advanced control scheme [1]. MPC is based on the repeated solution of a constrained finite-horizon optimal control problem to optimally control a dynamical system [2], [3], [4]. By dedicated – particularly robust – problem formulations, rigorous guarantees on stability and constraint satisfaction of the closed-loop system can be obtained. However, MPC requires to repeatedly solve an optimal control problem online. Often, this renders MPC unsuitable for applications where fast decisions are needed. Therefore, reducing the computational demand of MPC is an active research area.

One possible approach is to shift the computational demand for solving the optimization problem to an offline phase. This approach is known as explicit MPC [5]. Contrary to traditional MPC, where the control law is evaluated implicitly for the current system state by solving the optimization problem online, explicit MPC relies on pre-computing the control law as a function of the states. However, this is often only possible for linear MPC formulations, where the control law is a piecewise affine function of the state. In this case, the solution can be obtained via multiparametric programming. The online effort then reduces to assigning the current state to a corresponding domain and evaluating the associated affine control law. Approximate explicit approaches exist for nonlinear MPC formulations [6]. However, the number of regions that form the explicit solution drastically increases with the problem size, especially with the state dimension and the number of constraints [7], [8]. This limits the applicability of explicit MPC to small-scale problems and renders robust MPC problems out of reach.

One way to tackle the outlined problem is to use universal function approximators. Unsurprisingly, neural networks have been widely used to approximate MPC, especially after the rise of deep learning techniques, see for example [9], [10]. There are also numerous works that provide guarantees for learned MPC controllers using neural networks. For example, in [11], the authors use neural networks to approximate MPC that is robust to inaccurate inputs within given bounds. Ensuring with high probability that the neural network's approximation error is within those bounds, the guarantees obtained from MPC design transfer to the neural-network-based approximation. Applicability of this approach is shown in [12]. Besides works that aim at guaranteeing stability of the neural-network-controlled system by design, research effort has been dedicated towards deriving stability certificates for existing neural-network-based controllers. For example in [13], a simple-to-evaluate stability criterion has been developed for neural network approximations of MPC. In [14], stability and safety are ensured by restricting the output of the NN controller to a suitable set of control inputs that enables to establish the desired guarantees.

In this work, we focus on the use of Gaussian process (GP) regression (cf., [15]). In [16], GP regression is used to approximate MPC to control the flow around a circular cylinder, where the model is governed by a partial differential equation. Although the GP-based controller is validated by means of simulation, no rigorous closed-loop guarantees are provided. A GP-based approximation of MPC for nonlinear systems governed by ordinary differential equations is considered in [17], where it is combined with an active and control-oriented learning strategy. In [18], the GP-based approximation of MPC for linear systems is considered and probabilistic bounds on the approximation error of the GP are derived using a scenario-based approach. Feasibility guarantees for the closed-loop system are provided via an additional projection step.

We employ Gaussian process regression to approximate a robust nonlinear MPC. Specifically, we design a robust MPC exploiting Lipschitz continuity for constraint tightening as proposed in [19]. In this way, we ensure that the closed-loop system satisfies constraints robustly and is input-to-state stable for disturbances induced by a given maximum approximation error of the GP controller. We propose a GP design method for learning the MPC law while guaranteeing

[1]Control and Cyber-Physical Systems Laboratory, Technical University of Darmstadt, Germany, {alexander.rose, maik.pfefferkorn, hoang.nguyen, rolf.findeisen}@iat.tu-darmstadt.de

[2]Laboratory for Systems Theory and Automatic Control, Otto-von-Guericke University Magdeburg, Magdeburg, Germany

satisfaction of a desired maximum approximation error by the GP with high probability. To this end, we incorporate a scenario program in the training procedure and adapt the GP according to its outcome until the approximation error bounds are satisfied. A key advantage of GPs over neural networks for learning the controller is due to their proposed structured design. While GPs explicitly rely on the training data, which enables to identify particular data points that are beneficial for model quality and to incorporate them systematically, this is not the case for neural networks. The latter do not rely on the training data anymore once trained. Their capability of satisfying the desired approximation error bounds can only be validated after training has completed and adaptations of the training data set cannot be made in a systematic way as proposed for GPs. In summary, the proposed approach enables the systematic design of GP-based approximate MPC laws that guarantee constraint satisfaction and input-to-state stability of the closed-loop system with high-probability. We underline the effectiveness of our proposed approach in simulation.

II. Robust Model Predictive Control

We start by introducing the control problem, followed by an overview of the fundamentals of the robust MPC scheme employed in this work, including its adaption to control input errors in view of the application of approximate GP controllers. We conclude by reviewing recursive feasibility and input-to-state stability for the presented robust MPC scheme and elaborate on maximum approximation errors bounds that are still tolerable by the MPC without loss of guarantees.

A. Problem Formulation

We consider a discrete-time nonlinear system given by

$$x(k+1) = f(x(k), u(k)). \tag{1}$$

Here $x(k) \in \mathbb{R}^{n_x}$ is the state, $u(k) \in \mathbb{R}^{n_u}$ is the control and $k \in \mathbb{N}$ denotes the discrete time index. Moreover, system (1) is subject to state and input constraints $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$, where $\mathcal{X} \subset \mathbb{R}^{n_x}$ is closed and $\mathcal{U} \subset \mathbb{R}^{n_u}$ is compact. Furthermore, we assume local Lipschitz continuity of the nominal model f as we specify in Assumption 1.

Assumption 1 (Lipschitz continuity of nominal model): The origin is a steady state for system (1) and $f(x,u)$ is locally Lipschitz in x in the domain $\mathcal{X} \times \mathcal{U}$, i. e. there is a constant $0 < L_f < \infty$ such that for all x_1, $x_2 \in \mathcal{X}$ and for all $u \in \mathcal{U}$ it holds that

$$\|f(x_1,u) - f(x_2,u)\| \leq L_f \|x_1 - x_2\|.$$

We aim at controlling system (1) by a low-complexity, GP-based approximation of a model predictive controller $u = \kappa_{\text{MPC}}(x)$, denoted by $\bar{u} = \kappa_{\text{GP}}(x)$. This results in an approximation error $e = \kappa_{\text{MPC}}(x) - \kappa_{\text{GP}}(x)$, which induces an additive disturbance on the system, denoted as

$$\begin{aligned} x(k+1) &= f(x(k),u(k)) + w(k), \qquad \text{with} \\ w(k) &= f(x(k),u(k)+e) - f(x(k),u(k)). \end{aligned} \tag{2}$$

The disturbance $w(k)$ is bounded according to the following assumption.

Assumption 2 (Effect of the approximation error): There is a constant $0 < L_u < \infty$ such that for all $x \in \mathcal{X}$ and for all $u \in \mathcal{U}$ it holds that

$$\|w\| = \|f(x,u+e) - f(x,u)\| \leq L_u \|e\|. \tag{3}$$

The key idea is to design a robust MPC κ_{MPC} for the system with uncertainty (2), first, and to replace it by a GP-based approximate controller afterwards. This way, the closed-loop system is still guaranteed to be stable and to satisfy all constraints when the GP controller is applied. To this end, we rely on a robust MPC scheme with appropriate constraint tightening proposed in [19]. For the constraint tightening (to be discussed shortly), it is necessary to determine several Lipschitz constants. As discussed in [20], finding Lipschitz constants is in general challenging. However, for a particular application, it is often possible find an estimate of the required Lipschitz constants.

B. Control Scheme

In [19], the authors have presented a robust MPC scheme for disturbed systems that is based on constraint tightening and nominal predictions. If the nominal system exploited in the controller fulfills the tightened constraints, then the disturbed real-world system satisfies the original constraints. The optimal control problem underlying the MPC is formulated as

$$\min_{U(k)} \sum_{i=0}^{N-1} \ell(\hat{x}(i|k), u(i|k)) + V(\hat{x}(N|k)) \tag{4a}$$

s.t.

$$\hat{x}(i+1|k) = f(\hat{x}(i|k), u(i|k)),\ \hat{x}(0|k) = x(k), \tag{4b}$$

$$B_i = \{x \in \mathbb{R}^{n_x} | \, \|x\| \leq \frac{L_f^i - 1}{L_f - 1} L_u e_{\max}\}, \tag{4c}$$

$$\hat{x}(i|k) \in \mathcal{X}_i = \mathcal{X} \ominus B_i, \tag{4d}$$

$$\hat{x}(N|k) \in \mathcal{X}_{\text{term}} = \{x \in \mathbb{R}^{n_x} | V(x) \leq \alpha_v\}, \tag{4e}$$

$$u(i|k) \in \tilde{\mathcal{U}} = \mathcal{U} \ominus \{u \in \mathbb{R}^{n_u} | \, \|u\| \leq e_{\max}\}, \tag{4f}$$

$$U(k) = \{u(0|k), u(1|k), \ldots, u(N-1|k)\}, \tag{4g}$$

$$X(k) = \{x(k), \hat{x}(1|k), \ldots, \hat{x}(N|k)\}, \tag{4h}$$

where $N \in \mathbb{N}$ is the prediction horizon, $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}$ is the stage cost and $V : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ is the terminal cost. The time index $i \,|\, k$ indicates i steps ahead of k, where k is the current real-world time instance. The MPC exploits tightened state and input constraints (4d), (4f) to prepare for the use of an approximate controller with bounded approximation error $\|e\| \leq e_{\max}$. Therein, $\ominus$ denotes the Pontryagin difference defined by $A \ominus B = \{a \in A | a + b \in A, \forall b \in B\}$. The terminal region $\mathcal{X}_{\text{term}}$ in (4e) is defined as a sublevel set of V whose size is controlled by $\alpha_v \in \mathbb{R}_+$.

We denote the set of initial conditions $\hat{x}(0|k)$ for which (4) is feasible as $\mathcal{X}_{\text{feas}}$. In closed-loop, we repeatedly solve (4) and apply the first part of the optimal input sequence,

$u^*(0|k)$, to the plant. We denote this implicitly defined control law by $u = \kappa_{\text{MPC}}(x)$.

Assumption 3 (Stage cost): Let $\ell(x,u)$ be such that $\ell(0,0) = 0$, let there be positive constants $a > 0, b > 0$ such that $\ell(x,u) \geq a \left\| [x^T \, u^T]^T \right\|^b$ and let $\ell(x,u)$ be Lipschitz continuous in $\mathcal{X} \times \tilde{\mathcal{U}}$, i. e. $\|\ell(x_1,u) - \ell(x_2,u)\| \leq L_c \|x_1 - x_2\|$.

Assumption 4 (Locally stabilizing controller): There is a local controller $u = h(x)$ and a region $\Phi = \{x | V(x) \leq \alpha, h(x) \in \tilde{\mathcal{U}}, x \in \mathcal{X}\}$ for some $\alpha > 0$ such that

1) $V(f(x,h(x))) - V(x) \leq -\ell(x,h(x)), \forall x \in \Phi$.
2) there is a Lipschitz constant such that $\|V(x_1) - V(x_2)\| \leq L_v \|x_1 - x_2\|, \forall x_1, x_2 \in \Phi$.

Assumption 5 (Terminal region, [19]): The set $\mathcal{X}_{\text{term}} = \{x | V(x) \leq \alpha_v\}$ is such that $\forall x \in \Phi,\ f(x,h(x)) \in \mathcal{X}_{\text{term}}$.

C. Bound on the Approximation Error

In the following, we review feasibility properties of OCP (4) and provide upper bounds on the tolerable approximation error.

Theorem 1 (Recursive feasibility, [19]): Let Assumptions 1, 2, 4 and 5 hold and optimal control problem (4) be initially feasible, i. e. $x(0) \in \mathcal{X}_{\text{feas}}$. Then, the MPC scheme is repeatedly feasible if the approximation error is bounded by

$$\|e\| \leq \frac{\alpha - \alpha_v}{L_u L_v L_f^{N-1}} = e_{\max}. \tag{5}$$

Proof: The proof follows that of Theorem 1 in [19] by substituting $\gamma = L_u e_{\max}$. We omit details for brevity. ∎

D. Input-to-state stability

We first introduce necessary preliminaries before reviewing the stability properties of the MPC scheme based on (4).

Definition 1 (Class $\mathcal{K}$ and $\mathcal{K}_\infty$ function): A function η is of class $\mathcal{K}$ if it is a strictly increasing function $\eta : [0,\omega) \to [0,\infty)$ with $\eta(0) = 0$. If $\omega = \infty$ and $\eta(r) \to \infty$ when $r \to \infty$, then the function $\eta(\cdot)$ belongs to class $\mathcal{K}_\infty$

Definition 2 (Class $\mathcal{KL}$ function): A function $\beta : [0,a) \times [0,\infty) \to [0,\infty)$ belongs to class $\mathcal{KL}$ if for fixed s, $\beta(r,s) \in \mathcal{K}$ and for fixed r, $\beta(r,s)$ decreases with respect to s, and $\beta(r,s) \to 0$ as $s \to 0$.

Definition 3 (Input-to-state-stability, [21]): Consider the system $x(k+1) = F(x(k),w(k))$ with a bounded disturbance $\|w(k)\| \leq \bar{w}$. This system is input-to-state stable (ISS) if there exist functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that $\forall k$

$$\|x(k)\| \leq \beta(\|x_0\|, k) + \gamma(\bar{w}). \tag{6}$$

One approach to prove ISS for a closed-loop system is to show the existence of an ISS-Lyapunov function by using the following proposition.

Proposition 1 (ISS Lyapunov function, [19]): Consider the system $x(k+1) = F(x(k),w(k))$ with a bounded disturbance $\|w(k)\| \leq \bar{w}$. If there exist a continuous function $V(\cdot) : \mathbb{R}^n \to \mathbb{R}_+$ and functions $\alpha_1(\cdot), \alpha_2(\cdot), \alpha_3(\cdot) \in \mathcal{K}_\infty$, $\rho(\cdot) \in \mathcal{K}$ such that

$$\begin{aligned} \alpha_1(||x||) &\leq V(x) \leq \alpha_2(||x||), \\ V(F(x,w)) - V(x) &\leq -\alpha_3(||x||) + \rho(||w||), \end{aligned} \tag{7}$$

then the system is ISS and the function $V(\cdot)$ is called an ISS-Lyapunov function for the system.

Theorem 2 (Input-to-state stability, [19]): If the Assumptions 1, 2, 3, 4 and 5 hold, then the closed-loop system is input-to-state stable for all feasible initial conditions, i. e. $\forall x(0) \in \mathcal{X}_{\text{feas}}$.

Proof: The proof follows that of Theorem 2 in [19] by substituting $\gamma = L_u \|e\|$ and shows that the optimal cost function is an ISS-Lyapunov function. We omit details for brevity. ∎

III. Approximate Gaussian-Process-Based Controller

We introduce Gaussian process regression for learning a data-driven approximation of the MPC law including hyperparameter optimization. Thereafter, we describe a tailored GP design approach to learn an approximate controller with high-probability approximation error bounds. Finally, we illustrate how the tailored GP design enables to guarantee constraint satisfaction and stability of the closed-loop system with high probability.

A. Gaussian-Process-Based Approximation of MPC Laws

The objective is to obtain a data-driven approximation of the MPC law $\kappa_{\text{MPC}}(\cdot)$ in explicit form with lower computational complexity to avoid solving the optimal control problem (4) online. To this end, we rely on a training data set $\mathcal{D} = \{(x_j, u_j = \kappa_{\text{MPC}}(x_j) \mid j = 1, \ldots, n_D\}$ of state-input pairs, where the input u_j is obtained from the solution of the optimal control problem (4) with initial condition x_j. For simplicity of presentation, we assume $n_u = 1$ [1]. We define the matrix $Z \in \mathbb{R}^{n_D \times n_x}$ with rows $[Z]_j = x_j, j = 1, \ldots, n_D$ and, with a slight abuse of notation, we denote by $\kappa_{\text{MPC}}(Z) \in \mathbb{R}^{n_D \times 1}$ the vector of corresponding control inputs.

In this article, we employ Gaussian process regression for learning a data-driven approximation of $\kappa_{\text{MPC}}(\cdot)$ and start by defining a Gaussian process prior, denoted by

$$\hat{\kappa}_{\text{MPC}}(x) \sim \mathcal{GP}(m(x), c(x,x')). \tag{8}$$

A Gaussian process is defined as *a collection of random variables, any finite number of which have a joint Gaussian distribution* [15] and is often interpreted as a Gaussian probability distribution over functions. A GP is fully defined by its prior mean function $m : \mathbb{R}^{n_x} \to \mathbb{R}, x \mapsto \mathrm{E}[\hat{\kappa}_{\text{MPC}}(x)]$ and prior covariance function $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}, (x,x') \mapsto \mathrm{Cov}[\hat{\kappa}_{\text{MPC}}(\mathrm{x}), \hat{\kappa}_{\text{MPC}}(\mathrm{x}')]$.

We use the Gaussian process model of the MPC law to infer control inputs $u_* = \kappa_{\text{MPC}}(x_*)$ at so-far unobserved states x_* exploiting the available observations $\{Z, \kappa_{\text{MPC}}(Z)\}$. To this end, we consider the joint prior distribution of training observations $\kappa_{\text{MPC}}(Z)$ and test observations $\hat{\kappa}_{\text{MPC}}(x_*)$ given

[1] If $n_u > 1$, a simple approach is to use independent GP models for each control input dimension.

by [2]

$$\begin{bmatrix}\kappa_{\text{MPC}}(Z)\\ \hat{\kappa}_{\text{MPC}}(x_*)\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}m(Z)\\ m(x_*)\end{bmatrix},\begin{bmatrix}c(Z,Z)+\epsilon I & c(Z,x_*)\\ c(x_*,Z) & c(x_*,x_*)\end{bmatrix}\right), \quad (9)$$

which is defined through the GP prior model and where I denotes the unit matrix[3]. To obtain a meaningful model of the MPC law, we now incorporate the knowledge provided by the training data through conditioning the joint prior distribution on the training observations. Assuming a zero mean prior $m(x) = 0$ (w.l.o.g.), this yields the posterior distribution $\hat{\kappa}_{\text{MPC}}(x_*)|x_*, Z, \kappa_{\text{MPC}}(Z) \sim \mathcal{N}(\kappa_{\text{GP}}(x), \Sigma(x))$ with

$$\kappa_{\text{GP}}(x_*) = c(x_*, Z)(c(Z,Z)+\epsilon I)^{-1}\kappa_{\text{MPC}}(Z),$$
$$\Sigma(x_*) = c(x_*,x_*) - c(x_*,Z)^T(c(Z,Z)+\epsilon I)^{-1}c(x_*,Z).$$

The posterior mean $\kappa_{\text{GP}}(x_*)$ is the best available estimate for the unobserved control input $u_* = \kappa_{\text{MPC}}(x_*)$; the posterior variance $\Sigma(x_*)$ quantifies the model uncertainty [15]. As we can only apply deterministic control inputs to the system, we use the posterior mean to approximate the MPC law, i.e., $\kappa_{\text{MPC}}(x_*) = \kappa_{\text{GP}}(x_*) + e$ with a-priori unknown approximation error e.

Typically, the covariance function c depends on a set of free parameters θ, so-called hyperparameters, i. e. $c = c(x, x'; \theta)$[4]. In order to obtain a meaningful GP approximation, the hyperparameters need to be adapted to the underlying problem. To this end, we maximize the expressiveness of the GP model and obtain suitable hyperparamters as [15]

$$\theta^* = \arg\max_\theta \{\log(p(\kappa_{\text{MPC}}(Z) \mid Z, \theta))\}, \quad (10)$$

$$\log(p(\kappa_{\text{MPC}}(Z) \mid Z, \theta) = -\frac{1}{2}\kappa_{\text{MPC}}(Z)^T A^{-1}\kappa_{\text{MPC}}(Z) - \frac{1}{2}\log|A| - \frac{n_D}{2}\log 2\pi, \quad (11)$$

where $A = c(Z, Z; \theta) + \epsilon I$ and $|\cdot|$ denotes the determinant.

B. GP Controller Design

The MPC based on the optimal control problem (4) robustly guarantees repeated feasibility and ISS of the closed-loop system despite control approximation errors that satisfy $\|e\| \le e_{\max}$ (see Section II). In order to preserve these guarantees when deploying the approximate GP-based controller, we need to ensure that

$$\|\kappa_{\text{MPC}}(x_*) - \kappa_{\text{GP}}(x_*)\| \le e_{\max}, \ \forall x_* \in \mathcal{X}_{\text{feas}}. \quad (12)$$

We rely on a scenario-based approach [22] similar to [18] and obtain probabilistic guarantees for (12). To this end, we observe N_s *scenarios* $(x_{*,j}, \kappa_{\text{MPC}}(x_{*,j}), \kappa_{\text{GP}}(x_{*,j}))$, $x_{*,j} \in \mathcal{X}_{\text{feas}}, j = 1, \dots, N_s$ and compute the tightest error bound e^*_{GP} that holds true for all scenarios according to

$$e^*_{\text{GP}} = \arg\min_{e_{\text{GP}}}\{e_{\text{GP}}\}$$
$$\text{s. t. } \forall j = 1, \dots, N_s : \|\kappa_{\text{MPC}}(x_*) - \kappa_{\text{GP}}(x_*)\| \le e_{\text{GP}}. \quad (13)$$

The error bound e^*_{GP} generalizes to so-far unobserverd scenarios according to the following proposition.

Proposition 2 (Theorem 1.3, [22]): If the number of scenarios (sampled iid. according to a probability measure $\mathbb{P}$ on $\mathcal{X}_{\text{feas}}$) satisfies

$$N_s \ge \frac{2}{\varepsilon}\log\frac{1}{\omega} \quad (14)$$

for any $\varepsilon \in (0,1)$ (risk parameter) and $\omega \in (0,1)$ (confidence parameter), then it holds with confidence $1-\omega$ that

$$P\{\forall x \in \mathcal{X}_{\text{feas}} : \|\kappa_{\text{MPC}}(x) - \kappa_{\text{GP}}(x)\| \le e^*_{\text{GP}}\} \ge 1-\varepsilon. \quad (15)$$

We exploit Proposition 2 to iteratively build the training data set of a GP that satisfies a desired error bound e_{des}, such that $e^*_{\text{GP}} \le e_{\text{des}} \le e_{\max}$, with a desired probability and confidence in the sense of Proposition (2), considering e_{des} a design choice. That is, in each iteration, we solve (13) for N_s test locations according to (14). If $e^*_{\text{GP}} \le e_{\text{des}}$ is not satisfied, we add the scenario with the largest approximation error as an active training sample to the GP, retrain it and repeat this procedure. This error-based selection has shown to yield smaller training data sets when compared to other approaches such as random or uncertainty-based selection of the new training sample to be included in each iteration. The proposed design approach is summarized in Algorithm 1.

Algorithm 1 Algorithm to design the GP

Require: ε, ω, $e_{\max}$
1: choose $N_s \ge \frac{2}{\varepsilon}\log\frac{1}{\omega}$, $e_{\text{des}} \le e_{\max}$
2: initialize $\mathcal{D}$, $Z \leftarrow \{\}$, $\mu(Z) \leftarrow \{\}$
3: **repeat**
4: sample N_s iid. points $x_* \in \mathcal{X}_{\text{feas}}$
5: $e(x_*) \leftarrow \|\kappa_{\text{MPC}}(x_*) - \kappa_{\text{GP}}(x_*)\|$ for all N_s samples
6: $x_i \leftarrow \arg\max e(x_*)$
7: $Z \leftarrow \{Z, x_i\}$
8: $\kappa_{\text{MPC}}(Z) \leftarrow \{\kappa_{\text{MPC}}(Z), \kappa_{\text{MPC}}(x_i)\}$
9: $\theta \leftarrow \arg\max_\theta\{\log(p(\kappa_{\text{MPC}}(Z) \mid Z, \theta))\}$
10: **until** $e(x_*) \le e_{\text{des}}$ $\forall$ N_s samples

We summarize the results in Theorem 3.

Theorem 3 (Guarantees for the GP-based controller): System (1) under the GP-based controller designed according to Algorithm 1 is repeatedly feasible and input-to-state stable with high probability in the sense of (15).

Proof: If Algorithm 1 terminates, the GP satisfies the error bound $e_{\text{des}} \le e_{\max}$ with high probability according to Proposition 2. As the original MPC is designed to be robust w.r.t. the error bound $e_{\max}$, the guarantees presented in Section II transfer to the GP-based controller, using that (4f) ensures that $\kappa_{\text{GP}}(x) \in \mathcal{U}, \forall x \in \mathcal{X}_{\text{feas}}$ despite the approximation error. ■

[2]With a slight abuse of notation, we overload functions κ_{MPC}, m and c and mean by $\kappa_{\text{MPC}}(Z)$, $m(Z)$ and $c(Z, x_*)$ column vectors with $[\kappa_{\text{MPC}}(Z)]_i = \kappa_{\text{MPC}}(x_i)$, $[m(Z)]_i = m(x_i)$ and $[c(Z, x_*)]_i = c(x_i, x_*)$ and by $c(Z, Z)$ a matrix with $[c(Z, Z)]_{ij} = c(x_i, x_j)$, where x_i, x_j are the i^{th} and j^{th} row of Z.

[3]For computational reasons, we introduce a regularization constant $\epsilon > 0$ to ensure positive definiteness of the training covariance matrix up to machine precision.

[4]The following can be extended to include hyperparameters of arbitrary mean functions.

Remark 1: The presented approach can be extended to systems with bounded additive disturbance, $x(k+1) = f(x(k), u(k)) + \delta_k, \|\delta_k\| \leq \bar{\delta}$, by substituting (4d) with

$$B_i = \{x \in \mathbb{R}^{n_x} | \, \|x\| \leq \frac{L_f^i - 1}{L_f - 1}(L_u e_{\max} + \bar{\delta})\}.$$

IV. Case Study

We illustrate our approach in a case study. First, we introduce the problem set-up. Afterwards, the proposed method for designing GP-based approximations of the MPC law is applied. We conclude this section by comparing and evaluating the performance of the approximate GP controller against the that of the MPC. Note that in the following, all norms refer to the infinity norm if not stated otherwise.

A. System and MPC

We consider a discretized version of a simplified inverted pendulum model

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) + T_s(\sin(x_1(k)) - u), \end{aligned} \tag{16}$$

with sampling time $T_s = 0.1$ and constraints $\|u\| \leq 5$, $-\frac{\pi}{2} \leq x_1 \leq \pi + \frac{\pi}{8}$ and $\|x_2\| \leq 3$. The prediction horizon is set to $N = 20$. The stage cost function is $\ell(x, u) = \|(x, u)\|_{Q,R}$ with $Q = \mathrm{diag}([10, 1])$ and $R = 1$. We employ a linear quadratic regulator based on the linearization of system (16) around the upright position $x = [0 \quad 0]^T$ as terminal controller and obtain $h(x) = [3.8527 \quad 2.9613]\, x$ with the corresponding solution of the Ricatti equation $\tilde{P} = \begin{pmatrix} 114.19 & 48.56 \\ 48.56 & 35.17 \end{pmatrix}$. We now choose $P = 17.63\tilde{P}$, $\alpha = 41.94$ and $\alpha_v = 36.92$ such that Assumptions 4, 5 are satisfied. The terminal cost is $V(x) = \sqrt{x^T P x}$. Note that system (16) is of the form $f(x) = h(x) + g(x)u$. Therefore, we find $\|f(x(k), u(k) + e) - f(x(k), u(k))\| = \|g(x)\| \, \|e\|$ and thus $L_u = \max_{x \in \mathcal{X}} \|g(x)\| = T_s$. The remaining parameters are $L_f = 1 + T_s$ and $L_v = \sqrt{2} \left\| P^{\frac{1}{2}} \right\|_2$, resulting in $e_{\max} = 0.118$ according to (5).

B. GP-based Approximation of MPC

To learn the MPC law, we employ a GP with zero-mean prior, i. e. $m(x) = 0$, and the neural network covariance function [15]

$$c(x, x') = s_f \arcsin \frac{\tilde{x}^T \Lambda \tilde{x}'}{\sqrt{(1 + 2\tilde{x}^T \Lambda \tilde{x})(1 + 2\tilde{x}'^T \Lambda \tilde{x}')}},$$

where $\tilde{x} = [1 \, x^T]^T$. The hyperparameters are $\Lambda = \lambda^{-2} I$ with $\lambda \in \mathbb{R}$ and $s_f \in \mathbb{R}$, which are obtained via hyperparameter optimization according to (10). Using the neural network covariance function, the GP is capable of approximating even discontinuous MPC laws reasonably well. We train a GP on the MPC law according to Algorithm 1 with $e_{\text{des}} = e_{\max} = 0.118$, $\varepsilon = \omega = 0.001$ and therefore $N_s \geq 14816$. Algorithm (1) terminates after 178 are included in the GP model (shown as black dots in Figure 2). The validation results of the GP for 18000 samples are shown in Figure 1. As none of those samples shows an approximation error greater than $e_{\max}$, the GP satisfies the desired error bound with high probability according to Proposition (2).

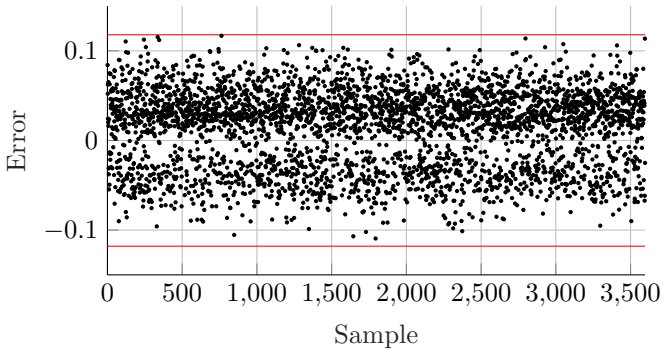

Fig. 1. Approximation Error of the GP controller with 178 active training data points for 18000 samples. Red lines indicate the desired norm-bound on the approximation error. For clarity, only every fifth data point is shown.

C. Analysis of the Approximated Controller

We now compare the performance of the approximate GP controller against that of the original MPC by means of computation time and closed-loop performance. To this end, we compute the state trajectories of system (16) starting from several initial conditions (Figure 2, black diamonds) when applying the MPC (Figure 2, solid lines) and the GP controller (Figure 2, dashed lines). Although the GP controller guarantees input-to-state stability of and robust constraint satisfaction by the closed-loop system, there is a visible difference between the trajectories resulting from the MPC and the GP controller. This difference is explained by the approximation error of the GP controller. However, note that we can achieve lower approximation errors $e_{\text{des}} < e_{\max}$ via Algorithm 1 by incorporating more data in the GP model as e_{des} is a design choice. In such case of lower approximation errors, the performance of the GP controller will be increased such that the resulting trajectories are closer to those obtained from employing the MPC. The mean closed-loop costs under the MPC and the GP controller are summarized in Table (I), indicating that the GP controller is an accurate approximation of the MPC. Furthermore, the mean and worst-case computation times for evaluating both the MPC and the GP controller are shown in Table I. The mean computation time of the GP controller is reduced by a factor of approximately 269 compared to that of the MPC, which involves solving the optimal control problem online. Hence, the GP controller shows a significant reduction in the computational complexity while maintaining a high and provably safe performance. Note that all computations are performed on an Intel Core i7-1165G7 with 4 cores and 32GB RAM. We solve (4) using direct multiple shooting [23] with warm starting using CasADi [24] and IPOPT.

V. Conclusion

In this article, we presented a method to approximate model predictive controllers using Gaussian processes with

TABLE I

COMPUTATION TIME AND CLOSED-LOOP PERFORMANCE.

	Computation time		Cost
	Mean	Worst	Mean
MPC	7.29 ms	18.14 ms	986
GP controller	0.027 ms	0.79 ms	989

high-probability guarantees on the closed-loop system. We started by illustrating the design of model predictive controllers that are robust against control input disturbances and reviewed guarantees on recursive feasibility and input-to-state stability. Furthermore, we proposed a design method for Gaussian process models that is dedicated towards deriving Gaussian-process-based approximations of MPC laws. The key advantage of the presented GP design approach is its capability of enforcing desired approximation error bounds on the GP with high probability, which are still tolerable by the MPC without loss of guarantees. Therefore, the guarantees obtained via MPC design are transferred to the approximate GP-based controller. The resulting GP-based approximation of the MPC control was shown to significantly alleviate the computational burden associated with traditional MPC, in which an optimal control problem is repeatedly solved online. We illustrated our approach in a case study, where the approximated controller was about 269 times faster with only a minor performance decrease.

Future research will focus on approximating controllers for the output feedback case. Furthermore, we will investigate the effect of measurement noise and estimation errors on the proposed approach and how we can take them explicitly into account. Finally, we plan on applying the proposed approach to a real-world system.

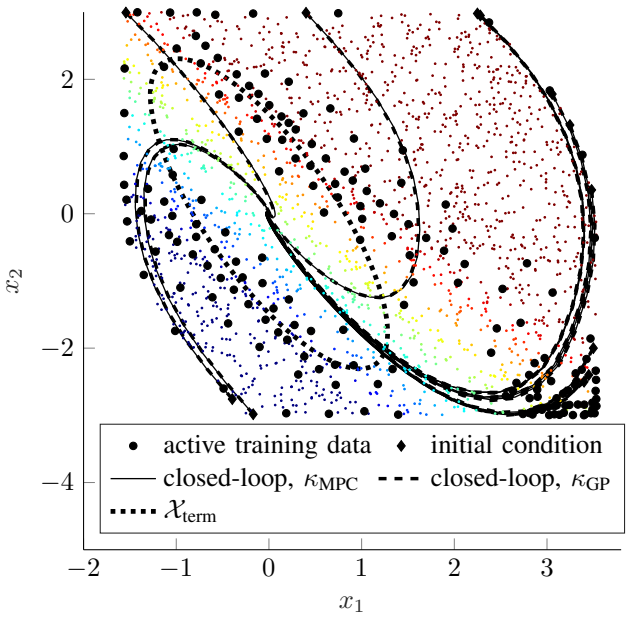

Fig. 2. Closed-loop trajectories of the system under the MPC and the GP controller. Colored dots represent a subset of the 18000 samples used for training and validating the GP, whereupon dark blue indicates a low control input and dark red indicates a high control input according to the true MPC law.

REFERENCES

[1] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: an engineering perspective," *The International Journal of Advanced Manufacturing Technology*, 2021.

[2] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting on Systems and Control*, vol. 11, 2002, pp. 119–141.

[3] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Nob Hill Publishing, LLC, 2019.

[4] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer International Publishing, 2017.

[5] A. Alessio and A. Bemporad, *A Survey on Explicit Model Predictive Control*. Springer Berlin Heidelberg, 2009, pp. 345–369.

[6] T. A. Johansen, "Approximate explicit receding horizon control of constrained nonlinear systems," *Automatica*, vol. 40, no. 2, pp. 293–300, 2004.

[7] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming - the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[8] C. Jones and M. Morari, "Polytopic approximation of explicit model predictive controllers," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2542–2553, 12 2010.

[9] T. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[10] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.

[11] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[12] 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 Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

[13] H. H. Nguyen, T. Zieger, R. D. Braatz, and R. Findeisen, "Robust control theory based stability certificates for neural network approximated nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 347–352, 2021.

[14] T. Zieger, A. Savchenko, T. Oehlschlaegel, and R. Findeisen, "One-step safe neural network supported control," in *American Control Conference*, 2021, pp. 514–519.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[16] Y. Sasaki and D. Tsubakino, "Explicit model predictive control with gaussian process regression for flows around a cylinder," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 38–43, 2018.

[17] D. Gángó, T. Péni, and R. Tóth, "Learning based approximate model predictive control for nonlinear systems," *IFAC-PapersOnLine*, vol. 52, no. 28, pp. 152–157, 2019.

[18] M. Binder, G. Darivianakis, A. Eichler, and J. Lygeros, "Approximate explicit model predictive controller using gaussian processes," in *58th IEEE Conference on Decision and Control (CDC)*, 2019, pp. 841–846.

[19] D. Limón Marruedo, T. Álamo, and E. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *41st IEEE Conference on Decision and Control*, vol. 4, 2002, pp. 4619–4624.

[20] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[21] E. Sontag, *Input to State Stability: Basic Concepts and Results*. Springer Berlin Heidelberg, 2008, pp. 163–220.

[22] M. C. Campi and S. Garatti, *Introduction to the Scenario Approach*. Society for Industrial and Applied Mathematics, 2018.

[23] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[24] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.