2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Robust Model Reference Gaussian Process Regression: Enhancing Adaptability through Domain Randomization

Hyuntae Kim

Abstract— **Nonlinear data-driven control strategies, particularly Model Reference Gaussian Process Regression (MR-GPR), have been effective in designing controllers directly from system input/output data, bypassing the need for explicit system modeling. This approach is advantageous for complex nonlinear systems where traditional modeling methods may be inadequate. MR-GPR employs Gaussian Process Regression to provide a non-parametric control method, enhancing adaptability and performance. However, real-world applications present challenges due to variability in system parameters, such as ensuring robustness and consistent performance. To address these challenges, this paper proposes a robust MR-GPR controller incorporating domain randomization to improve adaptability to varying operational conditions. This extension aims to maintain stable performance across diverse settings, mitigating the impact of parameter changes on control efficacy.**

I. Introduction

In the field of control theory, the exploration and implementation of nonlinear data-driven control strategies have become increasingly important. These approaches offer advanced mechanisms for handling complex systems, particularly when conventional modeling techniques are inadequate. Specifically, the Model Reference Gaussian Process Regression (MR-GPR) controller [1], [2] is recognized for its capability to handle nonlinear systems effectively. It enables the direct use of system input/output data for controller design, eliminating the need for explicit system modeling. This method utilizes Gaussian Process Regression (GPR) [3], [4] to provide a dynamic, non-parametric representation of system behavior. It facilitates the incorporation of extensive datasets and existing knowledge into the control scheme, thereby improving the system's operational reliability and performance [4]–[7].

However, real-world applications of these control systems reveal significant challenges, particularly when transitioning from controlled or simulated environments. A core issue lies in the variability of system parameters—such as changes in mass or length—which can introduce discrepancies between the modeled behaviors and actual system dynamics. These discrepancies can severely impact the controller's performance, leading to reduced accuracy or even instability. Traditional system identification-based control design methodologies [8], [9] have encountered similar challenges, underscoring their commonality across different approaches. As a result, these models may face limitations when applied beyond their initial training domain, risking performance degradation or failure in the face of novel or unanticipated parameter changes. This situation highlights the imperative for control systems that are robust to parameter variations not explicitly encountered during training.

Addressing this issue requires innovative approaches that enhance a model's ability to generalize from simulated training scenarios to real-world applications. Domain randomization emerges as a key strategy in this context, involving training the model across a wide range of simulated conditions. This approach aims to increase the model's robustness and flexibility, making it more adept at handling unforeseen scenarios in real-world deployments. By systematically varying simulation parameters, including environmental conditions and system dynamics, and observing a wider range of the system's past inputs and outputs, domain randomization exposes the model to a comprehensive spectrum of potential operational scenarios. This method has shown promising results in several fields, including robotics and autonomous systems, where real-world data collection may be challenging or infeasible [10]–[12].

This paper presents a robust MR-GPR controller that leverages domain randomization to effectively handle discrepancies between the modeled system parameters used during controller design and the actual parameters encountered in real-world experiments. By enriching the training dataset with simulated instances of parameter variations, the controller is crafted to ensure consistent performance and stability under conditions where the system's actual parameters diverge from those anticipated during the design phase. This approach introduces a method in nonlinear data-driven control that is flexible and robust, enabling the control system to perform effectively even when there is a discrepancy between the system parameters used in design and those encountered in actual operation.

The paper is structured as follows: Section II introduces the problem setup, discussing a specific category of nonlinear systems and establishing foundational assumptions. Section III details the proposed methodology for enhancing the robustness of the MR-GPR controller against variations in system parameters through GPR. An illustrative example demonstrating the efficacy of the robust MR-GPR controller is presented in Section IV. The paper concludes with a summary in Section V.

Notation: For column vectors a and b, $[a; b]$ denotes $[a^T, b^T]^T$. For discrete-time vector sequences $y(t)$ and $z(t)$,

This work was supported by the grant from Hyundai Motor Company's R&D Division.

The author is with ASRI, Department of Electrical and Computer Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea. Corresponding author: htkim@cdsl.kr

we define a vector

$$z_{[k,k+T]} := [z(k); z(k+1); \cdots ; z(k+T)],$$

and a set

$$\begin{aligned}\{(y(t), z(t))\}_{t=k}^{k+T} \\ := \{(y(k), z(k)), \cdots , (y(k+T), z(k+T))\}.\end{aligned}$$

II. Problem Formulation

Consider a single-input single-output (SISO) nonlinear discrete-time control-affine system characterized by a relative degree one in the Byrnes-Isidori normal form [13]:

$$y(t+1) = f(\lambda, z(t), y(t)) + g(\lambda, z(t), y(t))u(t), \quad \text{(1a)}$$

$$z(t+1) = h(\lambda, z(t), y(t)). \quad \text{(1b)}$$

In this model, $u(t) \in \mathbb{R}$ represents the control input, $z(t) \in \mathbb{R}^{n-1}$ denotes the internal state associated with the system's zero dynamics, $\lambda \in \mathbb{R}^j$ is a fixed but unknown parameter of the system, and $y(t) \in \mathbb{R}$ is the observed output. It is assumed that the functions $f(\cdot,\cdot,\cdot)$, $g(\cdot,\cdot,\cdot)$, and $h(\cdot,\cdot,\cdot)$ are unknown but smooth, and similarly, the parameter λ is unknown; the system is understood through its input/output data.

To proceed with the controller design, we introduce the following assumption regarding the system properties:

Assumption 1: The system (1) satisfies the following:

(a) There exists $\bar{g} > 0$ such that $|g(\lambda, z, y)| \leq \bar{g}$ and $g(\lambda, z, y) \neq 0$ for all $[\lambda; z; y] \in \mathbb{R}^{n+j}$. Also, the system dimension n and the global relative degree one are known.

(b) The internal dynamics (1b) are input-to-state stable with the input being y. □

Discretizing a continuous-time physical system generally results in a discrete-time model with a relative degree of one [14], suggesting the adaptability of the structure in (1). We assume that the state $z(t)$ is observable from past input/output data, as formalized below.

Assumption 2: There exists a smooth mapping $\mathcal{O}_z : \mathbb{R}^j \times \mathbb{R}^{2n-1} \to \mathbb{R}^{n-1}$ such that

$$z(t) = \mathcal{O}_z(\lambda, [y_{[t,t+n-1]}; u_{[t,t+n-2]}])$$

for any input sequence $u_{[t,t+n-2]}$ and output sequence $y_{[t,t+n-1]}$ of the system (1). □

Assumption 2 establishes the observability of the system's z dynamics, meaning that the state $z(t)$ can be consistently deduced from past input/output data, given the parameter λ.

Additionally, we consider a stable target model expressed as:

$$y_r(t+1) = f_r(y_r(t)) \quad \in \mathbb{R}, \quad (2)$$

assuming that

$$y_r(t+1) = f_r(y_r(t)) + \eta(t)$$

maintains stability in response to inputs, with $\eta(t)$ treated as an external input.

To synchronize the controlled system (1) with the target model (2), the control input needed is:

$$u(t) = \frac{f_r(y(t)) - f(\lambda, z(t), y(t))}{g(\lambda, z(t), y(t))}. \quad (3)$$

However, applying the control law (3) requires not only knowledge of the functions $f(\cdot,\cdot,\cdot)$ and $g(\cdot,\cdot,\cdot)$, but also the state $z(t)$ and, importantly, the parameter λ.

Previous work [1] has shown that it's possible to design a controller akin to (3) by relying solely on the input/output data of system (1) for a fixed λ. This suggests that the controller can function correctly for a predetermined λ using the acquired input/output data. However, in practical applications, the system parameters may vary, and the system from which data was obtained may differ from the actual system in operation due to changes in physical parameters.

This highlights the need for a control strategy that is robust against variations in system parameters not accounted for during the initial data collection phase. Therefore, our objective is to establish a method for designing a controller that maintains its effectiveness even when λ differs from the values used to obtain the initial dataset. This underscores the importance of adaptability and robustness in control systems to handle real-world uncertainties and variations in system dynamics.

To facilitate the estimation of the system parameter, we introduce the following assumption:

Assumption 3: It is assumed that there exists a smooth mapping $\mathcal{O}_\lambda : \mathbb{R}^{2m-1} \to \mathbb{R}^j$ that determines the parameter λ as

$$\lambda = \mathcal{O}_\lambda([y_{[t,t+m-1]}; u_{[t,t+m-2]}])$$

for any pair of input $u_{[t,t+m-2]}$ and output $y_{[t,t+m-1]}$ of the system (1). An upper limit for m, denoted as $\bar{m}$, is known. □

Unlike Assumption 2, which relates to state observability, Assumption 3 allows us to infer that λ is time-invariant and can be determined from past input/output data. While this assumption facilitates the estimation of λ, it may impose stringent conditions, especially when input/output data lead to undefined values for λ.

Remark 1: Consider a simple first-order physical model where the next output y^+ is defined by the equation $y^+ = \lambda u$. In this scenario, λ can be explicitly derived as $\lambda = y^+/u$. However, this direct method encounters a fundamental challenge if u becomes zero, leading to an indeterminate form for $\mathcal{O}_\lambda$. A practical approach involves ensuring that u does not equal zero, possibly by adding a small perturbation to the input. This maintains the definability and smoothness of $\mathcal{O}_\lambda$, even when direct computation would fail. □

It is important to note that the exact form of $\mathcal{O}_\lambda$ need not be known. This paper aims to demonstrate that it's feasible to design a controller operating as in (3) using only the system's input/output data from (1), even for λ values not previously encountered.

III. Main Result

This section introduces a data-driven controller designed to emulate the control input suggested by (3) through the application of GPR, trained exclusively on the input/output data from the system (1).

First, we show that the system's state and parameter, $z(t)$ and λ, can be inferred from its past input/output data. Let $\bar{n} := \max(n, \bar{m})$, and define:

$$\begin{aligned}
\eta_0(t) &:= [y_{[t-n+1,t-1]};\, u_{[t-n+1,t-1]}] \quad \in \mathbb{R}^{2(n-1)},\\
\zeta_0(t) &:= [y_{[t-\bar{n}+1,t-1]};\, u_{[t-\bar{n}+1,t-1]}] \quad \in \mathbb{R}^{2(\bar{n}-1)},\\
\zeta_1(t) &:= [\zeta_0(t);\, y(t)] \quad \in \mathbb{R}^{2\bar{n}-1},\\
\xi(t) &:= [\zeta_1(t);\, y(t+1)] \quad \in \mathbb{R}^{2\bar{n}},
\end{aligned}$$

where these vectors represent segments of an arbitrary input/output trajectory of the system (1), and $\mathcal{E}$ denotes the set containing all possible instances of $\xi(t)$.

Lemma 1: Under Assumptions 2 and 3, there exist smooth functions $\theta_\lambda : \mathbb{R}^{2\bar{n}-1} \times \mathbb{R} \to \mathbb{R}^j$ and $\theta_z : \mathbb{R}^{2\bar{n}-1} \times \mathbb{R} \to \mathbb{R}^{n-1}$ such that the parameter λ and the state $z(t)$ of the system described by (1) can be determined as follows:

$$\begin{aligned}
\lambda &= \theta_\lambda(\zeta_0(t), y(t)),\\
z(t) &= \theta_z(\zeta_0(t), y(t)),
\end{aligned}$$

applicable at every time step t.

Proof: We start by applying Assumption 2, which provides a smooth mapping $\mathcal{O}_z$:

$$z(t-n+1) = \mathcal{O}_z(\lambda, [y_{[t-n+1,t]}; u_{[t-n+1,t-1]}]),$$

linking λ to the input/output data and deducing the system's previous state, $z(t-n+1)$.

Subsequently, the system dynamics, represented by the function h, are recursively applied to determine the current state $z(t)$. This process unveils a dependency on λ and the system's historical input/output:

$$\begin{aligned}
z(t) &= h(\lambda, z(t-1), y(t-1))\\
&\quad\vdots\\
&= h(\cdots(h(\lambda, \mathcal{O}_\lambda([y_{[t-n+1,t]}; u_{[t-n+1,t-1]}]),\\
&\qquad\qquad y(t-n+1)), \cdots), y(t-1))\\
&=: \theta(\lambda, \eta_0(t), y(t)).
\end{aligned}$$

Incorporating Assumption 3, $z(t)$ is expressed as:

$$\begin{aligned}
z(t) &= \theta(\lambda, \eta_0(t), y(t))\\
&= \theta(\mathcal{O}_\lambda([y_{[t,t+m-1]}; u_{[t,t+m-2]}]), \eta_0(t), y(t))\\
&= \theta(\mathcal{O}_\lambda([y_{[t-m+1,t]}; u_{[t-m+1,t-1]}]), \eta_0(t), y(t))\\
&=: \theta_z(\zeta_0(t), y(t)).
\end{aligned}$$

Furthermore, leveraging Assumption 3, λ is deduced from the input/output sequence:

$$\begin{aligned}
\lambda &= \mathcal{O}_\lambda([y_{[t,t+m-1]}; u_{[t,t+m-2]}])\\
&= \mathcal{O}_\lambda([y_{[t-m+1,t]}; u_{[t-m+1,t-1]}])\\
&=: \theta_\lambda(\zeta_0(t), y(t)),
\end{aligned}$$

thereby aligning λ and $z(t)$ with their respective mappings, θ_λ and θ_z, derived directly from the system's input/output data. ■

Let us define the function $c : \mathbb{R}^{2\bar{n}} \to \mathbb{R}$ as follows:

$$c([\zeta_1(t); s]) := \frac{s - f(\theta_\lambda(\zeta_0(t), y(t)), \theta_z(\zeta_0(t), y(t)), y(t))}{g(\theta_\lambda(\zeta_0(t), y(t)), \theta_z(\zeta_0(t), y(t)), y(t))}.$$

This function facilitates the generation of the ideal control input as outlined in (3), specifically:

$$u(t) = c([\zeta_1(t); f_r(y(t))]), \tag{4}$$

where $\mathcal{C}$ denotes the set encompassing all feasible instances of $[\zeta_1(t); f_r(y(t))]$, clearly a subset of $\mathcal{E}$, i.e., $\mathcal{C} \subset \mathcal{E}$.

Implementing the ideal control strategy (3) through (4) follows the principle of output feedback control, relying entirely on the system's input/output data. A challenge arises from the need to determine the functions $f(\cdot)$, $g(\cdot)$, $\theta_\lambda(\cdot)$, and $\theta_z(\cdot)$ to construct (4). To overcome this, GPR is used to directly identify the function $c(\cdot)$. This approach treats $[\zeta_1(t); y(t+1)]$ as the input and $u(t)$ as the output for the function c, utilizing the relationship in (1a) where:

$$u(t) = c([\zeta_1(t); y(t+1)]).$$

To implement the proposed strategy, input/output data from experiments conducted under varying conditions of the system parameter λ are first collected as follows:

$$\{(u_{\mathsf{d}}^i(t), y_{\mathsf{d}}^i(t))\}_{t=1}^{N_i} \tag{5}$$

for each experiment i, where $N_i > \bar{n}$ represents the number of input/output data points for the i-th experiment. The subscript d denotes the sample data collected during the experiment, and i indexes the experiments, each possibly associated with a different λ.

The data from each experiment i are then rearranged to form the training input and output as:

$$\begin{aligned}
&\xi_{\mathsf{d}}^i(t+\bar{n}-1)\\
&= [\zeta_{1\mathsf{d}}^i(t+\bar{n}-1); y_{\mathsf{d}}^i(t+\bar{n})]\\
&= [y_{\mathsf{d}}^i[t, t+\bar{n}-2]; u_{\mathsf{d}}^i[t, t+\bar{n}-2]; y_{\mathsf{d}}^i[t+\bar{n}-1, t+\bar{n}]],
\end{aligned}$$

with the corresponding training output being

$$u_{\mathsf{d}}^i(t+\bar{n}-1),$$

thereby constituting the training dataset for the i-th experiment:

$$\mathcal{D}_i := \left\{(\xi_{\mathsf{d}}^i(t+\bar{n}-1), u_{\mathsf{d}}^i(t+\bar{n}-1))\right\}_{t=1}^{N_i-\bar{n}}.$$

To synthesize the insights gained across different λ settings, the datasets from all experiments are combined into a single comprehensive dataset:

$$\mathcal{D} := \bigcup_{i=1}^{S} \mathcal{D}_i, \tag{6}$$

where S is the total number of experiments conducted under varying parameters λ. Also, $N := \sum_{i=1}^{S}(N_i - \bar{n})$ is the total number of input/output data.

A Gaussian Process (GP) model is defined by its mean function $m : \mathcal{E} \to \mathbb{R}$ and a covariance function $k : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$. For identifying the control function $c(\cdot)$, our GP model utilizes a zero mean function and adopts the squared exponential (SE) kernel as its covariance function:

$$k(\xi, \xi') = \sigma_f^2 \exp\left(-\frac{1}{2}(\xi - \xi')^T L^{-1}(\xi - \xi')\right), \quad (7)$$

where σ_f and $L = \text{diag}(l_1^2, \ldots, l_{2\bar{n}}^2)$ serve as hyperparameters. These hyperparameters are optimized using marginal likelihood optimization, a method grounded in Bayesian principles as detailed in [3, Chapter 5].

In order to apply GPR to the problem of control function $c(\cdot)$, we introduce the following assumption regarding the control mappings:

Assumption 4: It is assumed that the control function $c(\cdot)$ resides within a reproducing kernel Hilbert space $\mathcal{H}_k$, induced by a corresponding kernel function k [15].

Upon preparing the training dataset $\mathcal{D}$ as specified in (6), the GP model provides the posterior mean and variance for any test input ξ within $\mathcal{E}$:

$$\mu_{\mathcal{D}}(\xi) = \mathbf{k}^T(\xi)\mathbf{K}^{-1}\mathbf{u}, \quad (8)$$

$$\sigma_{\mathcal{D}}(\xi) = k(\xi, \xi) - \mathbf{k}^T(\xi)\mathbf{K}^{-1}\mathbf{k}(\xi), \quad (9)$$

respectively, where

$$\mathbf{u} := [u_{\mathsf{d}}^1(\bar{n}); \cdots; u_{\mathsf{d}}^S(N_S - 1)],$$

$$\mathbf{k}(\xi) := [k(\xi_{\mathsf{d}}^1(\bar{n}), \xi); \cdots; k(\xi_{\mathsf{d}}^S(N_S - 1), \xi)],$$

$$\mathbf{K} := \begin{bmatrix} k(\xi_{\mathsf{d}}^1(\bar{n}), \xi_{\mathsf{d}}^1(\bar{n})) & \cdots & k(\xi_{\mathsf{d}}^1(\bar{n}), \xi_{\mathsf{d}}^S(N_S - 1)) \\ \vdots & \ddots & \vdots \\ k(\xi_{\mathsf{d}}^S(N_S - 1), \xi_{\mathsf{d}}^1(\bar{n})) & \cdots & k(\xi_{\mathsf{d}}^S(N_S - 1), \xi_{\mathsf{d}}^S(N_S - 1)) \end{bmatrix}.$$

It's important to note that these posterior functions are precisely defined unless the training inputs include identical elements, a situation that is rare in practice [16, Remark 3.3].

The posterior mean $\mu_{\mathcal{D}}(\cdot)$ effectively estimates the target control function $c(\cdot)$ using solely the system's input/output data, whereas the posterior variance $\sigma_{\mathcal{D}}(\cdot)$ reflects the estimation's confidence level.

The MR-GPR controller is formulated using the posterior mean $\mu_{\mathcal{D}}$ as follows:

$$\begin{aligned} u(t) &= \mu_{\mathcal{D}}([\zeta_1(t); f_r(y(t))]) \\ &= \mu_{\mathcal{D}}([y_{[t-\bar{n}+1, t-1]}; u_{[t-\bar{n}+1, t-1]}; y(t); f_r(y(t))]) \end{aligned} \quad (10)$$

effectuating an output feedback control mechanism.

In anticipation of our principal theorem, we establish the groundwork that elucidates the MR-GPR controller's approach to approximating the desired state feedback control. This approximation critically relies on the differential between the GPR-derived function $\mu_{\mathcal{D}}(\cdot)$ and the target control function $c(\cdot)$. Utilizing the SE kernel, GPR provides a mechanism to assess this discrepancy through the posterior variance $\sigma_{\mathcal{D}}(\cdot)$. The effectiveness of the MR-GPR controller is directly proportional to the proximity of $\mu_{\mathcal{D}}(\cdot)$ to $c(\cdot)$, with the potential for a bounded difference:

$$|\mu_{\mathcal{D}}(\xi) - c(\xi)| \leq \beta\sqrt{|\sigma_{\mathcal{D}}(\xi)|}, \quad \forall \xi \in \mathcal{E},$$

where β represents a constant multiplier. Studies [2], [16, Corollary 3.11], and [17, Corollary 3.2] suggest that a strategically curated dataset $\mathcal{D}$ can minimize $\sigma_{\mathcal{D}}(\cdot)$, thereby enhancing the alignment of $\mu_{\mathcal{D}}(\cdot)$ with the theoretical control function $c(\cdot)$, underscoring the significance of dataset selection in achieving optimal controller performance.

The forthcoming theorem provides a detailed analysis of the system's stability and performance when governed by the MR-GPR controller (10).

Theorem 1: Given Assumptions 1, 2, 3, and 4, a class-$\mathcal{K}$ function γ exists, ensuring that for a dataset $\mathcal{D}$ meeting the condition

$$|\mu_{\mathcal{D}}([\zeta_1; s]) - c([\zeta_1; s])| < \delta, \quad \forall [\zeta_1; s] \in \mathcal{C}, \quad (11)$$

with a specified $\delta > 0$, the ensuing closed-loop system (1), under the influence of the MR-GPR controller (10), secures

$$\limsup_{t \to \infty} \|[y(t); z(t)]\| < \gamma(\delta).$$

The proof follows a methodology similar to that employed in [1], and is thus not included here for brevity.

IV. Illustrative Example

In this section, we present an example to illustrate the effectiveness of the data-driven controller we propose. For ease of expression, the notation y_t might occasionally represent $y(t)$ throughout this discussion.

Consider a SISO system as described in prior work [1]:

$$y(t+1) = \lambda y^2(t) + z(t) + u(t), \quad (12a)$$

$$z(t+1) = 0.5\sin(y(t))z(t), \quad (12b)$$

where λ, u, y, z belong to $\mathbb{R}$. The system, with dimension $n = 2$ and a known global relative degree of one, satisfies Assumption 1(a). Given that the internal dynamics from y to z are input-to-state stable, the system also meets the conditions of Assumption 1(b). Moreover, the unique determination of $z(t)$ as

$$\begin{aligned} z(t) &= y(t+1) - \lambda y^2(t) - u(t) \\ &= \mathcal{O}_z(\lambda, [y_{[t,t+1]}; u(t)]). \end{aligned}$$

ensures compliance with Assumption 2. Additionally, the condition specified in Assumption 3 is met through the expression

$$\lambda = \frac{y_{t+2} - 0.5\sin(y_t)(y_{t+1} - u_t) - u_{t+1}}{y_{t+1}^2 - 0.5\sin(y_t)y_t^2},$$

assuming the input noise remains negligibly small. This leads to the determination that $\bar{m} = 3$, thereby setting $\bar{n} = 3$.

Following this, the definitions for θ_λ and θ_z are derived as

$$\begin{aligned}\lambda &= \frac{y_{t+2} - 0.5\sin(y_t)(y_{t+1} - u_t) - u_{t+1}}{y_{t+1}^2 - 0.5\sin(y_t)y_t^2} \\ &= \frac{y_t - 0.5\sin(y_{t-2})(y_{t-1} - u_{t-2}) - u_{t-1}}{y_{t-1}^2 - 0.5\sin(y_{t-2})y_{t-2}^2} \\ &= \theta_\lambda\left(\zeta_0(t), y(t)\right), \\ z(t) &= 0.5\sin(y(t-1))z(t-1) \\ &= 0.5\sin(y(t-1))\big(y(t) - \lambda y^2(t-1) - u(t-1)\big) \\ &= \theta_z\left(\zeta_0(t), y(t)\right)\end{aligned}$$

as indicated in Lemma 1.

Noting that

$$\begin{aligned}&c([\zeta_1(t); y(t+1)]) \\ &\quad = y(t+1) - \theta_\lambda\left(\zeta_0(t), y(t)\right) y^2(t) - \theta_z\left(\zeta_0(t), y(t)\right),\end{aligned}$$

we prepare the training dataset $\mathcal{D} = \bigcup_{i=1}^{S} \mathcal{D}_i$, where $\mathcal{D}_i$ includes observations from tests with random parameter $\lambda \in [0.1, 1]$ across $N_i = 5$ periods for every i. For each trial, the dataset

$$\xi_{\mathsf{d}}(t+2) = \left[y_{\mathsf{d}[t,t+1]}; u_{\mathsf{d}[t,t+1]}; y_{\mathsf{d}[t+2,t+3]}\right]$$

is collected as the training input, and

$$u_{\mathsf{d}}(t+2)$$

is used as the training output for $t = 1, 2$. Furthermore, in alignment with methodologies from the paper [1], we collect data under random initial conditions $y_{\mathsf{d}}(0), z_{\mathsf{d}}(0) \in [-1.2, 1.2]$ and random inputs $u_{\mathsf{d}}(t) \in [-1.2, 1.2]$. With $\mathcal{D}$, we adjust the hyperparameters in (7) by optimizing the marginal likelihood using the GPML toolbox [18]. A stable reference model is chosen as

$$y_r(t+1) = f_r(y_r(t)) = -0.4y_r(t),$$

ensuring input-to-state stability with

$$y_r(t+1) = -0.4y_r(t) + \eta(t).$$

The proposed output feedback controller is thus defined as

$$u(t) = \mu_{\mathcal{D}}([\zeta_1(t); -0.4y(t)]). \qquad (13)$$

For all experiments, the initial conditions are set as:

$$\begin{aligned}&(y(0), z(0)) \\ &\quad \in \{(0.4, 1.1), (0.4, -1.1), (-0.4, 1.1), (-0.4, -1.1)\},\end{aligned}$$

with zero input $u(0) = u(1) = 0$ applied initially. The real value of λ is set as $\lambda = 1$.

Figures 1 and 2 illustrate the performance of the proposed controller, designed with training data for $S = 8$ as

$$\lambda = \{0.2, 0.19, 0.18, 0.17, 0.16, 0.15, 0.14, 0.13\}$$

and derived from 250 and 500 initial conditions, respectively. The controllers' performance is compared to an ideal controller. Both figures show that the MR-GPR controller ensures asymptotic stabilization across a variety of initial conditions. Notably, Figure 2 reveals improved control performance due to a more extensive dataset compared to Figure 1. It is crucial to note that designing the MR-GPR controller with training data based on a single λ value from the set results in unstable system behavior, emphasizing the importance of including a diverse set of λ values in the training phase to enhance stability.

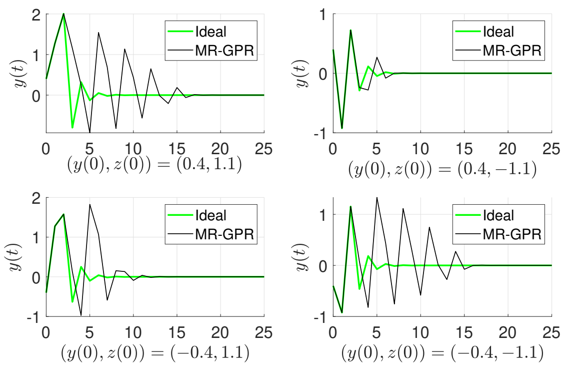

Fig. 1. Performance comparison via output trajectories of the system (12) between the MR-GPR controller $\mu_{\mathcal{D}}$ (black line) and the ideal controller c (green line), based on a dataset generated from 250 initial conditions with $S = 8$ distinct λ values.

Figure 3 expands the experiment by using only four λ values:

$$\lambda = \{0.2, 0.19, 0.18, 0.17\},$$

while maintaining a dataset size consistent with that of Figure 2, derived from 1000 initial conditions. The comparison of the error observed in Figure 2 with that in Figure 3 highlights the advantages of employing a wider range of λ values, in accordance with the principles of domain randomization.

Furthermore, Figure 4 explores the effects of a broader range of λ values:

$$\lambda = \{0.2, 0.18, 0.16, 0.14, 0.12, 0.1, 0.08, 0.06\},$$

demonstrating that expanding the variety of λ values within the dataset can further improve performance. This indicates that incorporating a broader spectrum of system parameters during training significantly enhances the MR-GPR controller's robustness and effectiveness.

V. Conclusion

This paper introduced a robust MR-GPR controller designed to use domain randomization for improving control performance across varied operational conditions. Unlike traditional approaches that rely on explicit system modeling or system identification, our method utilizes input/output data through GPR to design the controller. This non-parametric approach provides adaptability and improved performance by incorporating a wide range of system behaviors simulated under different conditions, thereby increasing the applicability of the controller to real-world scenarios.

A key aspect of this work is the integration of domain randomization into the MR-GPR controller's training

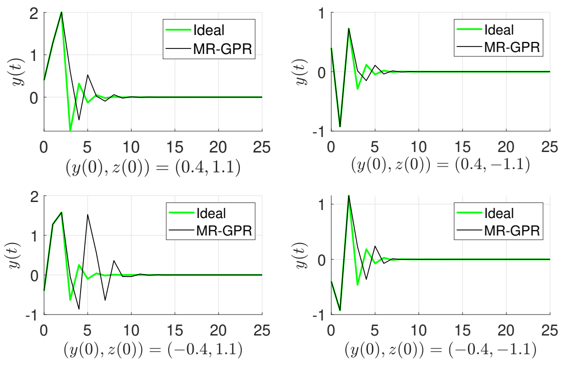

Fig. 2. Performance comparison via output trajectories of the system (12) between the MR-GPR controller $\mu_{\mathcal{D}}$ (black line) and the ideal controller c (green line), based on a dataset generated from 500 initial conditions with $S = 8$ distinct λ values.

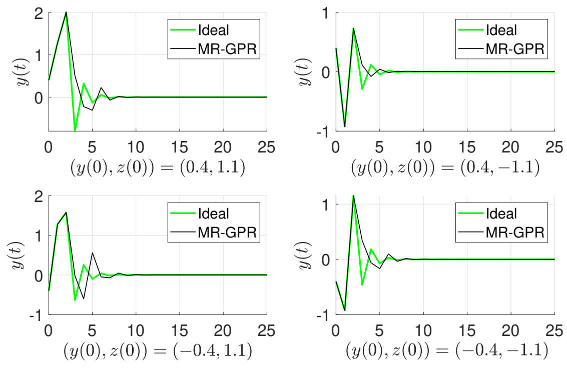

Fig. 4. Performance comparison via output trajectories of the system (12) between the MR-GPR controller $\mu_{\mathcal{D}}$ (black line) and the ideal controller c (green line), employing an expanded λ range compared to that in Figure 2.

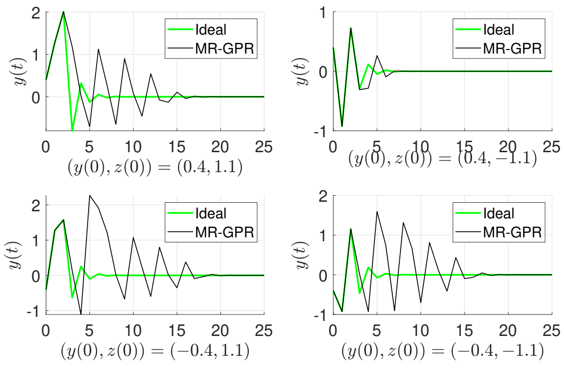

Fig. 3. Performance comparison via output trajectories of the system (12) between the MR-GPR controller $\mu_{\mathcal{D}}$ (black line) and the ideal controller c (green line), using a reduced set of λ values while maintaining the dataset size equivalent to that of Figure 2.

process, addressing the challenge of parameter variability in nonlinear systems. By systematically varying simulation parameters and expanding the dataset to include a broad spectrum of system inputs and outputs, we demonstrated that the controller maintains stability and performance even in operational conditions not explicitly included during training. The illustrative example confirmed that the proposed MR-GPR controller achieves asymptotic stabilization and shows improved control performance as the training dataset increases, highlighting the importance of a comprehensive dataset in managing real-world uncertainties.

The results suggest that the MR-GPR controller's ability to adapt to varying conditions makes it suitable for applications in fields where robustness to parameter variations is critical. This approach contributes to advancing data-driven control strategies by offering a flexible and effective method for managing complex nonlinear systems across diverse operational environments.

References

[1] H. Kim, H. Chang, and H. Shim, "Model Reference Gaussian Process Regression: Data-Driven Output Feedback Controller," in *Proceedings of the American Control Conference*, pp. 955-960, 2023.

[2] H. Kim and H. Chang, "Model Reference Gaussian Process Regression: Data-Driven State Feedback Controller," *IEEE Access*, vol. 11, pp. 134374–134381, 2023.

[3] C. K. I. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Cambridge, MA: MIT Press, 2006.

[4] J. Kocijan, *Modelling and Control of Dynamic Systems Using Gaussian Process Models*, Cham: Springer International Publishing, 2016.

[5] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen, and A. Girard, "Adaptive, Cautious, Predictive Control with Gaussian Process Priors," *IFAC Proceedings*, vol. 36, no. 16, pp. 1155-1160, 2003.

[6] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian Process Model Based Predictive Control," in *Proceedings of the American Control Conference*, pp. 2214-2219, 2004.

[7] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian Processes for Data-Efficient Learning in Robotics and Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408-423, 2013.

[8] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman Operator in Systems and Control*, Springer, 2020.

[9] L. Ljung, *System Identification: Theory for the User*, Upper Saddle River, NJ: PTR Prentice Hall, 1999.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23-30, 2017.

[11] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803-3810, 2018.

[12] F. Muratore, F. Treede, M. Gienger, and J. Peters, "Domain Randomization for Simulation-Based Policy Optimization with Transferability Assessment," in *Conference on Robot Learning*, pp. 700-713, 2018.

[13] A. Isidori, *Nonlinear Control Systems*, Springer Berlin Heidelberg, 1995.

[14] J. I. Yuz and G. C. Goodwin, *Sampled-Data Models for Linear and Nonlinear Systems*, London: Springer, 2014.

[15] N. Aronszajn, "Theory of Reproducing Kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337-404, 1950.

[16] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, "Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences," 2018. `http://arxiv.org/abs/1807.02582`

[17] A. Lederer, J. Umlauft, and S. Hirche, "Posterior Variance Analysis of Gaussian Processes with Application to Average Learning Curves," 2019. `http://arxiv.org/abs/1906.01404`

[18] C. E. Rasmussen and H. Nickisch, "The GPML Toolbox Version 4.0," *Technical Documentation*, 2016.