Position-Dependent Motion Feedforward via Gaussian Processes Applied to Snap and Force Ripple in Semiconductor Equipment

Poot, Maurice; Van Haren, Max; Kostic, Dragan; Portegies, Jim; Oomen, Tom

DOI
10.1109/TCST.2024.3385632

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Control Systems Technology

Position-Dependent Motion Feedforward via Gaussian Processes: Applied to Snap and Force Ripple in Semiconductor Equipment

Maurice Poot, Max van Haren, Dragan Kostić, Jim Portegies, and Tom Oomen, *Senior Member, IEEE*

***Abstract*— The requirements for high accuracy and throughput in next-generation data-intensive motion systems lead to situations where position-dependent feedforward is essential. This article aims to develop a framework for interpretable and task-flexible position-dependent feedforward through systematic learning with automated experimental design. A data-driven and interpretable framework is developed by employing Gaussian process (GP) regression, enabling accurate modeling of feedforward parameters as a continuous function of position. The data is efficiently collected and illustrated through an iterative learning control (ILC) algorithm. Moreover, a framework for experimental design in the sense of automatically determining the training positions is presented by exploiting the uncertainty estimates of the GP and the specified first-principles knowledge. Two relevant case studies show the importance and significant performance improvement of the approach for position-dependent snap feedforward for a simplified 1-D wafer stage simulation and experimental application to position-dependent motor force constant compensation in an industrial wirebonder.**

***Index Terms*— Gaussian processes (GPs), iterative learning control (ILC), mutual information.**

I. Introduction

FEEDFORWARD control is essential for high accuracy and throughput in next-generation data-intensive motion systems. These motion systems, including wirebonders [1], [2], lithography machines [3], and printer systems [4], consist of actuators, mechanics, and sensors. Typically, the actuators generate forces or torques and are considered inputs of the system, and the sensors measure the position or rotation and are considered outputs of the system. As a key development, precision mechatronics is becoming more lightweight and hence exhibits flexible dynamics [2]. To enable high throughput and usability of these precision mechatronic systems, accurate control consisting of both feedback and feedforward is necessary. In particular, systematically tunable feedforward that enables flexibility in the motion task is desired.

Feedforward control has been substantially developed in recent years to automatically tune the feedforward parameters to achieve accurate feedforward control. In [5], data-based tuning of manual feedforward is presented, including approaches based on iterative learning control (ILC) [6] through polynomial basis functions. The ILC-based approaches are further developed toward an instrumental variable (IV) approach in [7], which eliminates the bias on the parameters due to noise. Further extensions to the structure of the model, such as rational structures in [8], allow for greater accuracy in the presence of flexible dynamics.

Increasing performance requirements lead to a situation where the dynamics become position-dependent, which has consequences for both feedback and feedforward control. Linear parameter varying (LPV) approaches have been developed in [9], including related system identification methods [10], but their performance increase is limited as these rely only on feedback control [9]. A key component of state-of-the-art feedforward controllers that enable performance beyond that of feedback control is snap feedforward [11]. Snap feedforward compensates for the compliance of a system [12], that is, the low-frequency contribution of the flexible dynamics. Since compliance is typically a function of position due to position-dependent flexible dynamics, a position-dependent feedforward design is justified to enable high accuracy.

In addition to position-dependent dynamics, some actuators used in precision mechatronics also exhibit position dependence, for example, linear synchronous motors that are subjected to force ripple [13]. Often, through identification and modeling [14], that can also be performed in closed-loop based on data [15], the position dependence of the force ripple can be compensated, such that only position-dependent flexible dynamics need compensation using feedforward control.

Although position-dependent dynamics are well recognized in control applications and many relevant control approaches have been developed, at present a systematic position-dependent motion control framework that is specifically tailored toward typical position-dependent dynamics occurring in motion systems is lacking. In particular, many motion systems exhibit position-dependent dynamics but

Manuscript received 30 January 2023; revised 6 February 2023, 19 September 2023, and 8 February 2024; accepted 8 March 2024. Date of publication 16 April 2024; date of current version 23 October 2024. This work was supported by ASMPT. Recommended by Associate Editor T. Namerikawa. *(Corresponding author: Maurice Poot.)*

Maurice Poot and Max van Haren are with the Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: m.m.poot@tue.nl; m.j.v.haren@tue.nl).

Dragan Kostić is with ASMPT, Center of Competency, 6641 TL Beuningen, The Netherlands (e-mail: dragan.kostic@asmpt.com).

Jim Portegies is with the Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: j.w.portegies@tue.nl).

Tom Oomen is with the Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands, and also with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: t.a.e.oomen@tue.nl).

Digital Object Identifier 10.1109/TCST.2024.3385632

perform varying motions that stay close to a fixed position. This requires an interpretable and task-flexible position-dependent feedforward [8]. Moreover, systematic learning with automated experiment design is essential for industry adoption.

The central idea of this article is to develop a framework for position-dependent feedforward control using Gaussian processes (GPs). GPs allow for nonparametric modeling and interpolation using data and specification of prior knowledge about the modeled function (see [16]). At the same time, GP-based approaches allow calculation of the uncertainty of the estimation, which is exploited in experiment design to automatically place training positions through mutual information optimization [17].

Recent relevant developments in GPs for application in feedforward control (see [18]) have already led to substantial performance improvements, but do not directly address the class of motion control problems outlined above. For example, GPs are used in feedforward control to extend the model structure beyond that of traditional acceleration and snap feedforward (see [19]) and allow for a more general structure with a different perspective on model order selection. This approach is further extended to include prescribed nonlinearities [20] and is further developed toward nonlinear systems [21], enabling accurate control despite unknown nonlinearities. Moreover, feedforward controllers are extended with GPs in [22], where a GP is employed as an augmentation of a rigid-body feedforward in a fully data-based manner, which lacks interpretability.

The main contribution of this article is an interpretable and task-flexible position-dependent feedforward framework for motion systems. More specifically, current LTI feedforward models are extended toward position-dependent feedforward through modeling of the feedforward parameters as a function of position using GPs. Additionally, this framework enables systematic learning with automated experiment design, which is essential for industry adoption. The subcontributions are the following.

1) Position-dependent feedforward via GPs.
2) Systematic learning and automated experiment design using ILC and mutual information optimization.
3) Application to position-dependent snap feedforward in a simulation of a simplified 1-D wafer stage.
4) Experimental application to position-dependent motor force constant compensation in an industrial wirebonder.

Preliminary results related to the main contribution and subcontribution 3) appeared in [23] and [24]. This article provides a full framework and subcontributions 2) and 4). This framework could lead to rapid commissioning and industry adoption, enabled by efficient initialization through prior knowledge in the GP, near-optimal distribution of training positions, and the continuous learning and systematic learning nature of ILC.

The presented work relates to and differs from other work in the following way. In this work, *frozen* position dependency is considered, that is, linear dynamics are assumed at frozen positions, which is inherently different from dynamic approaches, such as linear parameter-varying feedforward [25] and linear periodically time-varying feedforward [26]. In [27], a position-dependent compliance function is modeled from PDEs, which requires more specific prior knowledge than the developed GP approach. Furthermore, other machine-learning techniques besides GPs are introduced in feedforward control, such as neural networks that could also compensate for position-dependent effects (see [28] and [29]), but these approaches require more training data and are difficult to interpret.

The outline is as follows. In Section II, the problem is formulated. In Section III, the developed approach is described, including interpolation, optimization, and learning. In Sections IV and V, two case studies are described. The first case study elaborates on position-dependent snap feedforward using GPs in a simulation of a simplified 1-D wafer stage and showcases the automated experiment design approach through the near-optimal distribution of the training positions. The second case study discusses the experimental validation of the approach on an industrial wirebonder and showcases the compensation of the force ripple. In Section VI, conclusions and ongoing work are stated.

Notation: The considered systems are assumed to be linear and time-invariant (LTI), n_u inputs and n_y outputs, unless otherwise stated. In most cases, unless explicitly stated otherwise, discrete-time systems are considered. For continuous-time systems, s denotes a complex indeterminate. A continuous-time LTI system for a fixed parameter p is denoted by $G(p, s)$. A single-input, single-output discrete-time transfer function is generally rational in the complex indeterminate z and is denoted as $\underline{G}(z)$. Signals are often assumed tacitly of length $N \in \mathbb{Z}_+$. Let $h(l), l \in \mathbb{Z}$ be the infinite-time impulse response vector of the system $\underline{G}(z)$. For a given input u, the output $y(k) = \sum_{l=-\infty}^{\infty} h(l)u(k-l)$. Assuming $u(k) = 0$ for $k < 0$ and $k > N-1$

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} h(0) & h(-1) & \cdots & h(1-N) \\ h(1) & h(0) & \cdots & h(2-N) \\ \vdots & \vdots & \ddots & \vdots \\ h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix}}_{u} \tag{1}$$

with $u, y \in \mathbb{R}^N$ the input and output vectors, respectively. Note that G is not restricted to be a causal system. Representations for MIMO systems with n_u inputs and n_y outputs follow directly from stacking SISO signals and convolution matrices. An LTI system for a fixed parameter p is denoted by the convolution matrix $G(p)$, where the elements $h(l)$ are a function of p. The lth element of a vector θ is expressed as $\theta[l]$. The weighted two-norm of a vector x is denoted as $\|x\|_W := (x^\top W x)^{1/2}$, where W is a weighting matrix. W is positive definite ($W \succ 0$) if and only if $x^\top W x > 0, \forall x \neq 0$. The cardinality of a set A is denoted by $|A|$. $\mathcal{A} \subseteq \mathcal{B}$ is used to represent that $\mathcal{A}$ is a subset of $\mathcal{B}$, meaning that all elements of

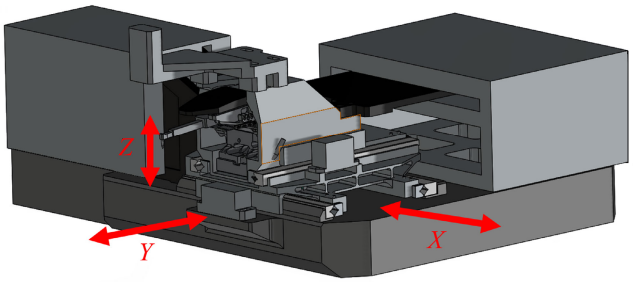

Fig. 1. Three-dimensional model of a wirebonder by ASMPT with the stacked *xyz*-stage used to bond wires on an integrated circuit.

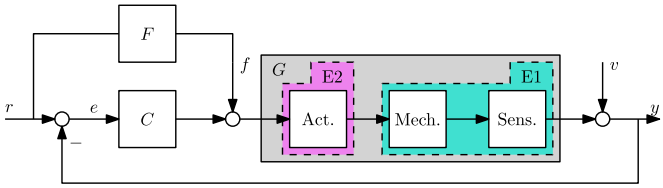

Fig. 2. Control scheme with feedback and feedforward on the system which consists of actuators, mechanics, and sensors. Here, E1 and E2 highlight the subcomponents of the system where two position-dependent effects originate from that are considered in the case studies CS1 and CS2.

$\mathcal{A}$ are inside $\mathcal{B}$. Furthermore, $\mathcal{A} \setminus \mathcal{B}$ denotes the set difference between $\mathbb{A}$ and $\mathcal{B}$, which means elements in $\mathcal{A}$ but not in $\mathcal{B}$. $\mathcal{A} \cup \mathcal{B}$ expresses the set union, which means all elements in $\mathcal{A}$ or $\mathcal{B}$.

II. Problem Formulation

A. Motion Control for Small-Amplitude Motions

Typical motion systems consist of actuators, mechanics, and sensors to enable systems to perform accurate motion tasks, for example, the wirebonder depicted in Fig. 1. The closed-loop feedback control scheme with feedforward for this system is shown in Fig. 2. The main goal in motion control is a servo task, that is, obtaining high tracking performance of the output y for a reference trajectory r. In particular, the tracking error $e = r - y$ is to be minimized. In Fig. 2, a feedforward controller F is added in conjunction to the traditional feedback controller [2]. From Fig. 2, it follows that

$$e = Sr - SGf - Sv \tag{2}$$

where $S = (I + GC)^{-1}$ the sensitivity function, $f = Fr$ the feedforward control signal, and v represents measurement noise that is assumed Gaussian with zero mean and variance σI_N. Recall that shaping of S by means of feedback control design is limited in reducing the tracking error due to Bode's sensitivity integral [30, Sec. 3.1]. Hence, the goal of minimizing the tracking error, that is, obtaining zero reference-induced error $e = -Sv$, is achieved for $f = Fr$ with $F = G^{-1}$. The feedback controller is assumed to be stabilizing and robust with respect to any position-dependent effect. The focus is on the design of the feedforward controller F to achieve the goal of high tracking performance. The other requirements on the feedforward in addition to the high tracking performance are generalizations to different motion tasks, systematic and automated tunable feedforward, and interpretability of the feedforward control signal.

The motion systems under consideration perform relatively short motions on different positions to create a product, that is, small-amplitude motions and possibly different reference trajectories, with starting points spread over the broad operating range. Two examples of such systems and their processes are shown schematically in Fig. 3 and are examples used in the case studies CS1 and CS2.

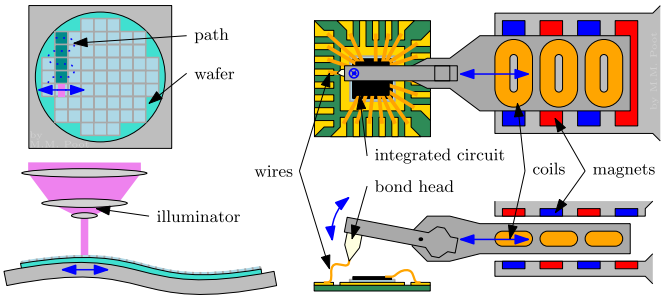

Fig. 3. Schematic representation (not to scale) of semiconductor manufacturing processes with their small-amplitude motions. Example of CS1: wafer illumination process in lithography where the behavior of the flexible dynamics is clearly dependent on position (left). Example of CS2: wire bonding process of an integrated circuit where the placement of the magnets with respect to the coils causes position dependency in the actuators (right).

CS1: In Fig. 3 (left), a lithography wafer stage is depicted where a mask of an integrated circuit is illuminated in short motions spread-out over a large silicon wafer [31].

CS2: In Fig. 3 (right), a wirebonder is shown where wires are bonded on an integrated circuit, wire by wire, often starting at different positions on the integrated circuit [1].

These systems are approximately linear for short motion tasks, that is, motions that stay close to a fixed position, yet these linearized systems are different for different fixed positions. In other words, nonlinear systems are considered for which the nonlinearity can be interpreted as a form of *position dependence*. This leads to the following assumption.

Assumption 1: For every position p, the system can be approximated accurately by a linear time-invariant system for motion tasks that stay close to the fixed position p. □

In this article, two position-dependent effects, originating from different subcomponents of the system as highlighted in Fig. 2, are analyzed and include the following.

E1: Position-dependent flexible dynamics, for example, compliance.

E2: Position-dependent actuators, for example, force ripple.

Note that for both effects, the system can satisfy Assumption 1.

In addition, these underlying effects are present in the relevant case studies considered in this article, that is, CS1 and CS2 directly connect to E1 and E2, respectively. Indeed, in both case studies, other effects could also play a role but these are not considered, for example, E2 in CS1, E1 in CS2, and other effects that can be modeled through the developed approach.

Next, the position-dependent flexible dynamics effect E1 is further elaborated upon.

B. Position-Dependent Flexible Dynamics Effect E1 Present in CS1

Consider the mechanics and sensors denoted by E1 in Fig. 2. For typical motion control systems as in Fig. 3, Assumption 1 translates to the following. For every position p, the system

is approximated accurately by an LTI system for which the transfer function (depending on p) can be written as a modal decomposition as [32], [33, Sec. 3.2]

$$G(p,s) = \underbrace{\sum_{i=1}^{n_{\mathrm{RB}}} \frac{c_i(p)b_i^\top(p)}{s^2}}_{\text{rigid-body modes}} + \underbrace{\sum_{i=n_{\mathrm{RB}}+1}^{n_{\mathrm{RB}}+n_f} \frac{c_i(p)b_i^\top(p)}{s^2+2\zeta_i\omega_i s+\omega_i^2}}_{\text{flexible modes}} \tag{3}$$

where n_{RB} is the number of rigid-body modes that are not suspended, n_f is the number of flexible modes, the vectors $c_i(p) \in \mathbb{R}^{n_y}$, $b_i(p) \in \mathbb{R}^{n_u}$ are a function of position and associated with the inertia or mode shapes, and $\zeta_i, \omega_i \in \mathbb{R} \geq 0$. Often, inertia is not a function of position, for example, for translational systems, in which case $b_i(p), c_i(p)$ for $i = 1, \ldots, n_{\mathrm{RB}}$ are position-independent.

Now, considering the linear system as a function of position described by (3), a position-dependent feedforward is essential for systems that change as a function of position. Although dynamical behavior at low frequencies is often dominated by rigid-body dynamics, the contribution of position-dependent flexible modes of (3) at low frequencies, called compliance, can often not be neglected [11], [34]. Snap feedforward systematically compensates for compliance (see [23]). Since the system's compliance is position-dependent (see (3)), the snap feedforward parameter needs to be made position-dependent. To summarize, E1 follows from the system in (3) where the compliance changes as a function of position, which necessitates the need to accurately model the snap parameter as a function of position.

Next, the position-dependent actuators effect E2 is further elaborated.

C. Position-Dependent Actuators Effect E2 Present in CS2

Consider the actuators denoted by E2 in Fig. 2. In many motion systems, linear permanent magnet motors are used as actuators (see [13]), where a force is generated by applying a current to a coil that is subjected to a magnetic field generated by magnets. This is illustrated in Fig. 3 (top right), where the actuator translates horizontally and the magnets alternate in polarity over the operating range. These actuators have a high force density and allow high position accuracy, but are subjected to position-dependent force ripple [35] due to fluctuations in the magnetic field.

The nonlinear position-dependent force ripple effect is modeled by the motor force constant and can be estimated using traditional but troublesome calibration methods. The motor force constant $K_{\mathrm{mfc}}(p) : [N] \rightarrow [A]$ describes the relation between the actuator output force and the coil input current and is a function of position due to the placement of the magnets. To estimate this relation, traditional force measurements can be performed at different positions, but this is a time-intensive and invasive operation, as it requires the physical insertion of a force sensor. Alternatively, traditional noninvasive measurements of the motor force constant can be performed (see [13] and [15]). However, these noninvasive measurements are time-intensive and require numerous slow and dedicated experiments that cannot be performed during normal machine operation.

A motor force constant that is not calibrated accurately results in a systematic error in the feedforward signal as a function of position and may significantly affect the tracking accuracy. Typically, uncompensated variations of the motor force constant will lead to feedforward parameters that need to vary as a function of position (see (4)).

In summary, E2 follows from the position dependency in linear motors caused by the force ripple modeled in the motor force constant $K_{\mathrm{mfc}}(p)$. If the motor force constant is not calibrated accurately, this position-dependent effect may cause possibly additional position dependence in the feedforward parameters, necessitating the need for accurate modeling of the feedforward parameters as a function of position.

D. Problem Formulation

In view of the position-dependent effects E1 and E2, as highlighted in Fig. 2, which can lead to parasitic effects, this article aims to develop a framework for interpretable position-dependent feedforward for accurate tracking of varying motion tasks.

To achieve an interpretable position-dependent feedforward for these two cases, the feedforward is chosen to be a linear combination of functionals of the reference r. Moreover, the feedforward parameters, related to the coefficients in the linear combination, may depend on the position of the system. An example of such a feedforward signal is

$$f(p,r,t) = K_{\mathrm{mfc}}(p)\Big(k_{fa}(p)\ddot{r}(t) + k_{fs}(p)\ddddot{r}(t) + k_{fc}(p)\,\mathrm{sign}(\dot{r}(t))\Big). \tag{4}$$

More generally, position-dependent feedforward of the following form is considered:

$$f(p,r,t) = K_{\mathrm{mfc}}(p)\sum_{i=1}^{n_\theta}\theta_i(p)\Psi_i(r,t) \tag{5}$$

where $K_{\mathrm{mfc}}(p)$ is a position-dependent constant, which in this article often has the interpretation of an estimate of the motor force constant, the functions $\theta_i(p)$ are the position-dependent feedforward parameters, and the functionals $\Psi_i(r)$ map the reference signal to a new signal, and $\Psi_i(r,t)$ stands for the evaluation of the signal $\Psi_i(r)$ at time t. To enhance interpretability, the function $\Psi_i(r,t)$ is often taken of the form

$$\Psi_i(r,t) = \psi_i\left(\frac{\mathrm{d}^{n_i}}{\mathrm{d}t^{n_i}}r(t)\right) \tag{6}$$

for some $n_i \in \mathbb{N}$ and for $i = 1, \ldots, n_\theta$ (see also [36]). The functions ψ_i may be nonlinear, for example, $\psi_i(x) = \mathrm{sign}(x)$. After discretizing signals, the derivatives get replaced by discrete derivatives, for example, $((1-z^{-1})/T_s)$, which closely relates the feedforward parameterization to FIR parameterizations (see [37, p. 53]). Note that the traditional motion feedforward of [36] is recovered in the case the parameters are position independent, that is, $\theta_i(p) = k_i \ \forall \ p$, then the example (5) would become

$$f(r,t) = K_{\mathrm{mfc}}(p)\Big(k_{fa}\ddot{r}(t) + k_{fs}\ddddot{r}(t) + k_{fc}\,\mathrm{sign}(\dot{r}(t))\Big) \tag{7}$$

where k_{fa} is the acceleration parameter, k_{fs} is the snap parameter, and k_{fc} is the Coulomb friction parameter. The

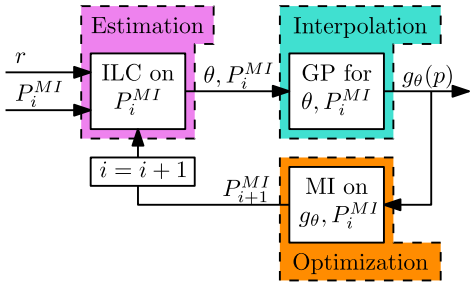

Fig. 4. Iterative scheme for learning position-dependent feedforward parameters $\theta(p)$ with estimation using ILC, modeling using a GP $g_\theta(p)$ for interpolation, and mutual information optimization for optimizing the next near-optimal training position P^{MI}.

advantage of such a feedforward control is its flexibility in the motion task and the physical interpretability of the parameters.

To conclude, the problem of systematically learning an interpretable position-dependent feedforward for accurate tracking of a variety of motion tasks is reduced to estimating and modeling the position-dependent parameters $\theta_i(p)$ of the discrete-time feedforward signal f given by (5).

Next, the approach to estimate and model the position-dependent feedforward parameters is presented, and afterward, the two case studies CS1 and CS2 are discussed.

III. Approach

In this section, the approach to achieve position-dependent feedforward is proposed. First, the concept is motivated. In the remainder of this section, the estimation, interpolation, and optimization subproblems are covered. Finally, an interconnection and summary are given.

A. Motivation of the Approach

The key idea to determine the position-dependent parameters of the feedforward signal (5) is to *estimate* the feedforward parameters for a fixed position p, to *interpolate* with GP regression, and to *optimize* the fixed positions near optimally with mutual information optimization to achieve experiment design, motivated in the following subproblems and visually shown in Fig. 4.

1) Estimation: To accurately determine the parameters θ for a fixed position p, learning approaches, for example, ILC or IVs, can be used. These data-driven methods iteratively determine feedforward parameters that minimize a cost function or the error for a reference trajectory. Hence, the reference trajectory must not excite the position dependence of the system, which requires Assumption 1 to hold. Using a data-driven learning method, θ for a variety of fixed positions p can be obtained, that is, the dataset $\{p_i, \theta_i\}_{i=1}^n$, where n is the number of data points.

2) Interpolation: For (5) to be an accurate position-dependent feedforward signal, accurate estimates of θ for every unseen position p must be inferred using the dataset. This aspect is rather challenging since the position dependence of a parameter is often not known beforehand, making parametric modeling for interpolation difficult. In contrast to parametric modeling, nonparametric modeling via GP regression enables accurate interpolation where prior knowledge about the position dependence can directly be specified in a kernel.

3) Optimization: To achieve a systematic learning approach that ensures an accurate estimate of θ for the entire operating range of the machine, the fixed positions p for which θ will be learned must be determined. The GP provides an estimate of the uncertainty of the conditioned parameter θ, allowing optimization of the distribution of the data points over the operating range using, for example, mutual information optimization, resulting in automated experiment design.

In summary, the estimation, interpolation, and optimization challenges motivate a data-driven learning approach of the frozen feedforward parameters, for example, using ILC, a GP regression approach for accurate nonparametric interpolation of the parameter, and a mutual information optimization approach for near-optimal distribution of the training positions.

In Section III-B, GP regression for interpolation is discussed.

B. Interpolation of the Data Using GP Regression

To model the feedforward parameters as a continuous function of position, GP regression is employed, constituting Contribution 1). GP regression methods are a convenient and versatile mathematical tool for function estimation and interpolation. Given an unknown function g_0, for example, one of the feedforward parameters as a function of position, the goal is to obtain an accurate model g of g_0 from measurements. The process of function estimation is performed as follows. An initial guess of the unknown function is encoded in the *prior* in the form of a probability distribution over functions. Afterward, a few measurements of the unknown function are observed. The prior distribution is then conditioned on these measurements by Bayesian inference to obtain the *posterior* distribution. The posterior distribution gives an estimate of the unknown function and the variance of the estimation. This posterior distribution can be easily computed if the prior is a GP since then the posterior is also a GP.

1) GP Definition: A GP is a collection of real-valued random variables $g(p)$, of which every finite subcollection follows a multivariate Gaussian distribution. Here, $p \in \mathcal{X}$ are the values that index $g(p)$, for example, the fixed position of the system. A GP $g(p)$ is fully specified by its mean and covariance function [16, Sec. 2] and is denoted by

$$g(p) \sim \mathcal{GP}\left(m(p), k(p, p')\right) \tag{8}$$

where $m : \mathcal{X} \to \mathbb{R}$ is the mean function

$$m(p) := \mathbb{E}\left[g(p)\right], \quad \text{for } p \in \mathcal{X} \tag{9}$$

and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the covariance function, or kernel

$$k(p, p') := \text{Cov}\left(g(p), g(p')\right), \quad \text{for } p, p' \in \mathcal{X}. \tag{10}$$

The covariance function and selection thereof are elaborated on in Section III-B4.

2) Noisy Measurements: To model the unknown function $g_0 : \mathcal{X} \to \mathbb{R}$, noisy measurements of the unknown function are obtained, which are estimates of the feedforward parameters. The noisy measurements consist of training inputs, or measurement positions, in the input domain $\mathcal{X}$

$$P = [p_1, \ldots, p_n] \tag{11}$$

and corresponding noisy measurements in $\mathbb{R}$ at those inputs

$$Y = [y_1, \ldots, y_n] \tag{12}$$

that is, n noisy measurements estimates of the GP g. Hence,

$$y_i = g(p_i) + \epsilon_i, \quad \text{for } i = 1, \ldots n \tag{13}$$

where ϵ_i are i.i.d. normally distributed with mean 0 and standard deviation σ.

As a result, there are n training inputs in P that correspond to fixed positions p_i and noisy measurements Y that correspond to noisy estimations of the feedforward parameters θ. The variance of the noisy measurements is σ^2 and acts on the estimation of the parameters that are assumed to be Gaussian. This results in the dataset $\mathcal{D} : [Y, P]$. Note that multiple parameter estimations per p are possible in the dataset, in which case these estimates should be independent of each other according to (13).

3) Conditioning the GP on the Data: After obtaining the dataset, function estimation and interpolation, that is, predictions of unseen inputs, can be performed by conditioning the GP $g(p)$ on the data. As such, the GP $g(p)$ for $p \in \mathcal{X}$ after conditioning, denoted by $g(p) \mid Y$, becomes a model of the unknown function $g_0(p)$.

To describe the conditioned process, the Gaussian distribution of the conditioned variables $g(p)$ for $p \in P_*$ must be described for a finite subcollection of test inputs

$$P_* := [p_{*,1}, \ldots, p_{*,n_*}] \tag{14}$$

for n_* test inputs in $\mathcal{X}$. The conditioned distribution of $g(p)$ for $p \in P_*$, denoted by $g(P_*) \mid Y$, can be computed using Bayes' rule and is a multivariate Gaussian distribution with computationally tractable mean

$$\mu_* + K(P_*, P)\left(K(P, P) + \sigma^2 I_n\right)^{-1}(y - \mu) \tag{15}$$

and covariance

$$K(P_*, P_*) - K(P_*, P)\left(K(P, P) + \sigma^2 I_n\right)^{-1} K(P, P_*) \tag{16}$$

where

$$\begin{aligned} \mu_* &= \mathbb{E}\left[g(p) : p \in P_*\right] \\ \mu &= \mathbb{E}\left[g(p) : p \in P\right] \end{aligned} \tag{17}$$

and the matrix $K(A, B)$, for finite sequences A and B in $\mathcal{X}$ [16], is defined by

$$K(A, B)_{ij} := k(a_i, b_j). \tag{18}$$

Conveniently, the variance of the conditioned GP yields an uncertainty estimate that can be exploited later.

4) Kernel and Hyperparameter Optimization: The choice of the kernel in the prior distribution is relevant as it can encode prior knowledge, for example, smoothness, symmetry, or periodicity, of the modeled function g. In our particular case, the prior describes how the feedforward parameters are expected to behave as a function of position, which is in mechanical systems often smooth, as will be evident from the two case studies in Sections III and IV.

The quality of GP regression, that is, interpolation of the position-dependent parameter, depends on the data P, Y and on the selected kernel and its hyperparameters. The latter can be conveniently optimized by marginal likelihood optimization based on the data (see [16, Ch. 5]). Hence, the training inputs P should be carefully selected with respect to the operating range of the machine to achieve good interpolation of the data.

In the next section, a solution is presented for systematically distributing the data near-optimally over the input domain.

C. Experimental Design Using Mutual Information Optimization

To achieve automated experiment design, measurement positions must be placed over a broad operating range, ideally using as few measurements as possible. The learning data is automatically and near-optimally distributed by employing mutual information (MI) optimization, constituting Contribution 2) and leading to rapid commissioning. Mutual information optimization (see [17] for a general introduction) aims to find measurements or training positions that are most informative about the unmeasured positions. These training positions are derived by exploiting the variance of the GP, in particular, the criterion takes into account the effect of training positions on the posterior uncertainty, enabling the exploitation of the incorporated prior knowledge to distribute the training data P over the input domain $\mathcal{X}$. In comparison to alternatives, such as the equidistant-spaced grid of the training data, mutual information optimization results in an automated and systematic learning approach to achieve an accurate GP with less user intervention and few training data, thus less experimentation time. Additionally, mutual information is expected to lead to better results than entropy optimization, that is, maximizing the uncertainty for the distribution of training positions, since it is likely to *waste* information by placement at the boundaries of the operating range [38]. This section presents mutual information optimization, a greedy approximation that makes computations feasible, and an a priori and sequential approach.

1) Mutual Information Definition and Objective: Mutual information optimization searches for the training positions that most significantly reduce the uncertainty about the estimates in the remaining space. These training positions are selected from a predetermined and finite subset of all possible training positions covering the broad operating range of the machine, denoted by $\bar{P}$.

To determine where to add the next measurement position, in the context of mutual information optimization, it is necessary to derive the probability distribution of the noisy measurements. If measurements will take place at different positions, the distribution of the noisy measurements y_i of (13) are described by the GP $h(p)$, given by

$$h(p) \sim \mathcal{GP}\left(m(p), k(p, p') + \sigma^2\right) \tag{19}$$

where σ is the standard deviation of the measurement noise. The GP $h(p)$ can be loosely regarded as the prior of the measurements.

Now, using the distribution of the noisy measurements, the idea is to select training positions P that will give good predictions at all unmeasured positions $\bar{P} \setminus P$ according to the mutual information criterion. Mutual information between the

measurements at the positions P and the measurements at the remaining positions $\bar{P} \setminus P$ [17], [39] is defined as

$$\mathrm{MI}\,[P] = H\left[h(\bar{P} \setminus P)\right] - H\left[h(\bar{P} \setminus P) \mid h(P)\right] \tag{20}$$

where $H[A \mid B]$ is the conditional entropy function of the GP A conditioned on B and is computed as

$$H[A \mid B] = \frac{1}{2}\ln\left(\det \Sigma^2_{A|B}\right) + \frac{1}{2}\left(\ln(2\pi) + I\right). \tag{21}$$

In other words, the mutual information criterion describes the difference in the entropy of the training position P with the remaining training positions $\bar{P} \setminus P$, where the entropy is expressed in terms of the variance of the GP.

The set of MI-optimal training positions $P^{\mathrm{MI,opt}}$ is found by maximizing the mutual information over an arbitrary set of $n = |P^{\mathrm{MI,opt}}|$ training positions, that is,

$$P^{\mathrm{MI,opt}} = \underset{P \subseteq \bar{P},\, |P|=n}{\operatorname{argmax}}\ \mathrm{MI}\,[P] \tag{22}$$

where MI[P] is given in (20). This optimization problem is NP-complete (see [17, Th. 2]), hence a greedy one-step algorithm is employed in the next section to find a computationally tractable approximation to the optimal solution.

2) Greedy Mutual Information Approximation: The greedy mutual information approximation algorithm repeatedly adds a single training position $p \in \mathcal{X}$ to the near-optimal greedy MI set P^{MI} until n training positions have been chosen. The next near-optimal greedy MI set P^{MI}_{i+1}, given the current set P^{MI}_i [17], is given by

$$P^{\mathrm{MI}}_{i+1} = P^{\mathrm{MI}}_i \cup \left\{ \underset{p \subseteq \left(\bar{P} \setminus P^{\mathrm{MI}}_i\right)}{\operatorname{argmax}} \left(\mathrm{MI}\left[h(P^{\mathrm{MI}}_i \cup p)\right] - \mathrm{MI}\left[h(P^{\mathrm{MI}}_i)\right]\right)\right\} \tag{23}$$

where

$$\mathrm{MI}\left[h(P^{\mathrm{MI}}_i \cup p)\right] - \mathrm{MI}\left[h(P^{\mathrm{MI}}_i)\right] = H\left[h(p) \mid h(P^{\mathrm{MI}}_i)\right] - H\left[h(p) \mid h(\hat{P})\right] \tag{24}$$

and $\hat{P}(p) := \bar{P} \setminus (P^{\mathrm{MI}}_i \cup p)$ the remaining set of possible training positions in $\mathcal{X}$ after selecting p, which depends on p (see [17] for the derivation of (24)). The greedy MI algorithm of (23) selects the position p that is uncertain given the previous positions P^{MI}_i, that is, $H[h(p) \mid h(P^{\mathrm{MI}}_i)]$ is large, and informative for the uncertain unobserved positions, that is, $H[h(p) \mid h(\hat{P})]$ is small. The entropy function H as a function of the variance of these conditioned GPs is given in (21). For a GP as in (19) with zero mean function, the posterior variances of $h(p)$ for p conditioned on $h(P^{\mathrm{MI}}_i)$ and $h(\hat{P})$, respectively, are given by

$$\sigma^2_{h(p)|h(P^{\mathrm{MI}}_i)} = K(p,p) + \sigma^2 - K\left(p, P^{\mathrm{MI}}_i\right)\left(K(P^{\mathrm{MI}}_i, P^{\mathrm{MI}}_i) + \sigma^2 I\right)^{-1} K(P^{\mathrm{MI}}_i, p) \tag{25a}$$

$$\sigma^2_{h(p)|h(\hat{P})} = K(p,p) + \sigma^2 - K(p,\hat{P})\left(K(\hat{P},\hat{P}) + \sigma^2 I\right)^{-1} K(\hat{P}, p) \tag{25b}$$

with the matrix K as defined in (18) and σ the standard deviation of the measurement noise. Note that these posterior variances only hold when p and $\hat{P}$ and p and P^{MI}_i are disjoint, that is, $p \cap \hat{P} = \varnothing$ and $p \cap P^{\mathrm{MI}}_i = \varnothing$ due to consecutive measurements at different positions. By substitution of (21) and (24) into (23), the near-optimal greedy MI set can be computed for each $p \subseteq (\bar{P} \setminus P^{\mathrm{MI}}_i)$ with

$$P^{\mathrm{MI}}_{i+1} = P^{\mathrm{MI}}_i \cup \underset{p \subseteq \left(\bar{P} \setminus P^{\mathrm{MI}}_i\right)}{\operatorname{argmax}} \frac{1}{2}\ln\left(\delta(p)\right) \tag{26}$$

where $\delta(p)$ is the ratio of the posterior variances of (25), that is,

$$\delta(p) := \frac{\sigma^2_{h(p)|h(P^{\mathrm{MI}}_i)}}{\sigma^2_{h(p)|h(\hat{P})}}. \tag{27}$$

The following should be noted. First, the computation is quite heavy for fine grids $\bar{P}$ since determining n near-optimal greedy training position requires $O(n\bar{n}^4)$, where $\bar{n} = |\bar{P}|$ (see [17] for approaches to reduce the computational complexity to $O(n\bar{n})$). Second, a sufficiently fine grid for $\bar{P}$ is achieved only for $|\bar{n}| > 2n$ since the mutual information is approximately monotonic [17, Sec. 4.2]. Third, $\delta(p)$ in (27) is a function of the kernel, hence the optimization of mutual information exploits prior knowledge in the kernel and the result is dependent on the selected hyperparameters of the GP.

3) A Priori and Sequential Mutual Information Optimization: The optimization of the MI in (23) does not require training output Y, thus the algorithm can be executed a priori for some selected hyperparameters. Since it is difficult to select the hyperparameters and no data Y is available to tune the hyperparameters using marginal likelihood optimization as described in Section III-B4, this a priori MI optimization is typically not optimal.

In contrast to the a priori approach, the MI optimization can be performed *sequentially*, that is, the hyperparameters of the GP are optimized based on the training outputs Y that are measured in the training input P^{MI}_i before computing (23). Hence, for every iteration i in (23), the near-optimal training position is based on the best possible hyperparameters (see [40]). This sequential MI optimization approach in relation to the GP is presented in Section III-E and illustrated in Fig. 4.

D. Estimation of the Feedforward Parameters Using ILC

Accurate estimates of the feedforward parameters have to be learned to obtain an accurate representation of the position-dependent feedforward parameters. For reference trajectories that do not *excite* the position dependence of the system, the parameters can be automatically learned using a data-driven method, such as ILC with basis functions (ILCBF).

1) ILC Definition and Objective: Considering the closed-loop scheme of Fig. 2, the error is derived as a function of the feedforward signal per iteration. To facilitate the exposition, the system is assumed SISO, for MIMO extensions (see [41]). An experiment, or iteration or trial, has length $N \in \mathbb{N}$ with reference $r_j \in \mathbb{R}^N$ and measured output $y_j \in \mathbb{R}^N$, where the trial index is denoted as $j \in \mathbb{N}$. The measurement noise v_j is assumed to be Gaussian with zero

mean and variance $\sigma_j^2 I_N$. From Fig. 2, it follows that the tracking error of trial j is given by

$$e_j = S r_j - SGf_j - Sv_j. \tag{28}$$

The objective in ILC is to minimize the tracking error of the next trial

$$e_{j+1} = S r_{j+1} - SGf_{j+1} - Sv_{j+1}. \tag{29}$$

By subtracting (28) from (29), and under the assumption of $r_{j+1} = r_j = r$, that is, r is constant with respect to the trial, the error propagation from trial j to $j+1$ is given by

$$e_{j+1} = e_j - SG\left(f_{j+1} - f_j\right) - S(v_{j+1} - v_j). \tag{30}$$

The objective in ILC is to minimize the predicted tracking error for trial $j+1$, $\hat{e}_{j+1}$, based on measurements e_j and f_j and a model of the real system SG.

2) Feedforward Parameterization: Recall that the postulated feedforward of (5) for a given fixed position p reduces to (7). Moreover, this time-stacked feedforward signal can be rewritten as a polynomial basis functions feedforward parameterization, given by

$$f_j = \Psi(r_j)\theta_j \tag{31}$$

where $\Psi(r_j) \in \mathbb{R}^{N \times n_\theta}$ is the basis function matrix, where often (r) is omitted for brevity, and $\theta \in \mathbb{R}^{n_\theta}$ are the feedforward parameters (see [5] and [42]). Regarding the selection of these basis functions, zero reference-induced errors, that is, $e_j = -Sv_j$ in (28), can be achieved for $f_j = G^{-1} r_j$, hence the basis functions should be chosen such that $F(\theta_j) \approx G^{-1}$, where $F(\theta_j)$ is the convolution matrix of $\underline{\Psi}(z)$ with θ_j (see [42]). Indeed, (7) is recovered in case $\Psi(r) = \left[(\mathrm{d}^2/\mathrm{d}t^2)r \;\; (\mathrm{d}^4/\mathrm{d}t^4)r \;\; \mathrm{sign}((\mathrm{d}/\mathrm{d}t)r)\right]$, $K_{\mathrm{mfc}} = 1$, and $\theta = \begin{bmatrix} k_{fa} & k_{fs} & k_{fc} \end{bmatrix}^\top$.

3) Solution of ILC: To minimize the tracking error of the next trial, the predicted error of trial $j+1$, denoted by $\hat{e}_{j+1}$, is achieved by minimization of the cost function

$$\mathcal{J}_j(\theta_{j+1}) := \|e_j - \widehat{SG}\left(\Psi(r_{j+1})\theta_{j+1} - f_j\right)\|_{W_e}^2 \tag{32}$$

where $W_e \succ 0$, $\hat{e}_{j+1} = e_j - \widehat{SG}(\Psi(r_{j+1})\theta_{j+1} - f_j)$ the predicted error for trial $j+1$ based on measurements e_j and f_j, and $\widehat{SG}$ a model of the real system SG. The optimal feedforward parameters θ, thus the feedforward parameters at position p corresponding to (5), are given by

$$\theta_{j+1}^{\mathrm{opt}} := \arg\min_{\theta_{j+1}} \mathcal{J}_j(\theta_{j+1}). \tag{33}$$

The solution is computed analytically and is given by

$$\theta_{j+1}^{\mathrm{opt}} = \left[\Psi^\top \widehat{SG}^\top W_e \widehat{SG}\Psi\right]^{-1} \Psi^\top \widehat{SG}^\top W_e e_j. \tag{34}$$

Model mismatch may sometimes lead to instability of the parameter update rule. An input penalty term $\|\Psi(r_{j+1})\theta_{j+1}\|_{W_f}^2$ with weight $W_f \succeq 0$ can be included in (32) to guarantee monotonic convergence of the parameters $\|\theta_j - \theta_\infty\|_2$, see [42]. Note that, in practice, a limited set of basis functions is often sufficient for convergence of the parameter update rule. Moreover, model mismatch usually results in slower learning, as such, sufficient consecutive trials should be performed till the parameters seem to converge.

4) Aspects Regarding Estimation of Feedforward Parameters: Three important aspects of ILC in regard to estimating feedforward parameters should be noted. First, for certain systems the feedforward parameters found by ILC are indeed accurate approximations of physical parameters, for other systems these approximations can be very inaccurate since the algorithm optimizes the cost function (32) rather than the accuracy of the parameter estimation. For example, the acceleration parameter should really be viewed as an *effective mass*, that is, a mass parameter corrected for motor force constants, and only in some cases corresponds to the actual total mass of the system under consideration. In general, the approximations of physical parameters can be made more accurate with careful selection of the basis functions and the reference trajectories, such that the basis functions describe the system dynamics accurately in the frequency range of the energy of the reference. In particular, for estimating acceleration, friction, or snap parameters, references with energy in the low-frequency range should be used, as these avoid excitation of high-frequency dynamics that cannot be captured by the polynomial parameterization of (31). Furthermore, these reference trajectories should not excite position dependence, so that Assumption 1 can be exploited.

Second, the GP interpolates the results of the ILC algorithm, that is, the feedforward parameters that minimize the error. Thus, the GP can be viewed as estimating the outcome of ILC when it would be applied to a previously unseen position.

Third, besides the data-driven method of ILC, other parameter optimization approaches such as IVs can be used to estimate the feedforward parameters at a position (see [7] for considerations).

To conclude, ILC with basis functions allows for data-driven and systematic learning of the feedforward parameters at different fixed positions.

E. Integration of the Estimation, Interpolation, and Optimization Methods

To achieve a systematic approach for position-dependent feedforward, the estimation, interpolation, and optimization appraoch of Sections III-B–III-D are combined, as depicted in the scheme of Fig. 4, and summarized below.

1) Estimation: Starting with a reference r for which the system satisfies Assumption 1 and one or more initial positions P_i^{MI}, the feedforward parameters θ can be estimated using (34) given a set of basis functions as in (31). Note that multiple consecutive ILC trials can be performed to gain insights into the variance of the parameters due to, for example, trial-varying disturbances, however, due to measurement noise care must be taken to ensure independence of consecutive parameter estimations of ILC.

2) Interpolation: The GP can be constructed using the obtained data $\theta, P_i^{\mathrm{MI}}$. Note that for multiple feedforward parameters, one can construct independent GPs and optimize the hyperparameters of each GP separately. Possible extensions include the combination of all parameters in one GP. Considering independent GPs, the GP of the lth feedforward parameter, $\theta[l]$, is denoted by $g_{\theta[l]}$, and for each GP, a kernel and hyperparameters need to be selected, possibly

incorporating prior knowledge and optimizing the hyperparameters based on the data (see Section III-B4).

3) Optimization: The near-optimal training positions P_{i+1}^{MI} can be determined using (26) based on the kernel and selected hyperparameters and the set P_i^{MI}. Note that if multiple GPs are constructed, only one of the GPs can be selected to base the mutual information optimization scheme on, for example, the GP of the parameter that is expected to be the most position-dependent. Again, possible extensions include the use of the combined GP for mutual information optimization. At the next near-optimal training position, the ILC estimation procedure can be repeated to extend the dataset, starting the loop over again by setting $i = i + 1$. This iterative procedure can be stopped when there is sufficient data for the GP to make accurate estimates. Note that the framework also allows for a continuous learning approach, where the feedforward parameters for an unseen position are regressed from the GP, and subsequent consecutive estimates of ILCBF during normal operation are added to the data of the GP.

IV. Case Study CS1: Snap

In this section, a case study of CS1 of the proposed GP method is presented that includes mutual information optimization in a simulation of a simplified 1-D wafer stage, constituting contributions 2) and 3). In particular, the snap feedforward parameter is modeled using a GP that enables high control accuracy for the broad operating range, and the benefits and ease of use of optimizing the training positions to achieve automated experiment design are highlighted. The main motivations for a simulation of a simplified 1-D wafer stage are to enhance reproducibility and to isolate the effect of the flexible dynamics, therefore, an unsupported flexible beam is chosen as the simulation model. First, the flexible beam setup is described, followed by the setup to estimate the snap parameter. Second, the GP is introduced that models the snap parameter. Third, the influence of the training positions obtained by mutual information optimization and an equidistant-spaced grid is compared. Finally, the resulting performance of the proposed GP method with and without mutual information optimization is compared to position-independent feedforward for different test positions.

A. Flexible Beam Simulation Setup

An unsupported flexible beam has dominant flexible dynamics that are a function of position due to varying output locations. A schematic of the beam is shown in Fig. 5 and represents a simplified 1-D example of a wafer stage depicted in Fig. 3. The beam has a length of 500 mm, two actuators on both sides of the beam that have the same input, that is, $a_1 = a_2 = (u/2)$, and a sensor with varying position. The output location is emulated by moving the sensor position $p \in \mathcal{X} = [-250, 250]$ mm with respect to the center of the beam (see Fig. 3). The rigid-body mode, that is, translation mode, and two flexible modes are considered, thus $n_{\text{RB}} = 1$ and $n_f = 3$ in (3). A Bode magnitude plot of the linear dynamics at different positions is shown in Fig. 6 and clearly highlights the position-dependent flexible dynamics. A stabilizing feedback controller is used and consists of a lead filter and gain to achieve a bandwidth of 4 Hz. The beam follows a small fourth-order reference profile of 0.5 mm designed according to [36]. Moreover, the dynamics of the beam satisfies Assumption 1 since the position dependency is not aligned with the direction of the motion.

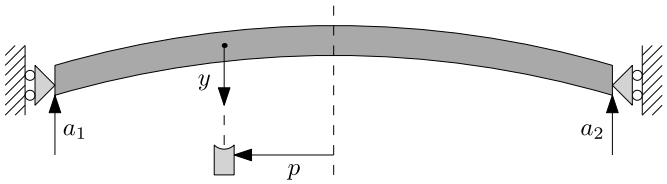

Fig. 5. Schematic of the unsupported flexible beam setup with dominant flexible dynamics as a function of position due to varying performance location emulated by changing sensor position $p \in [-250, 250]$ mm. Actuators on both sides generate the same input force, that is, $a_1 = a_2 = (u/2)$.

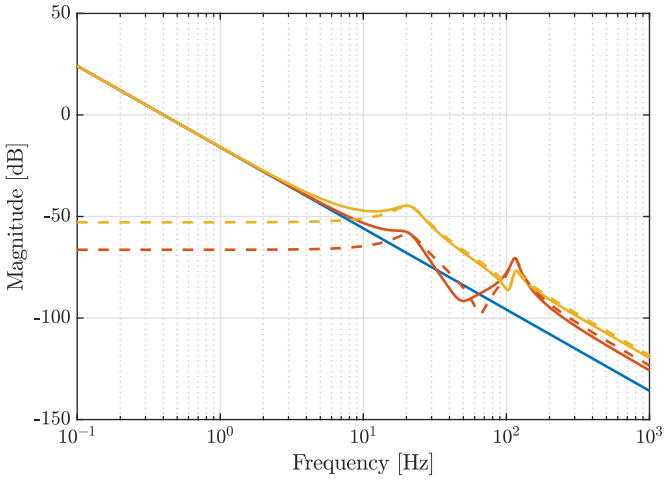

Fig. 6. Bode magnitude plot of the flexible beam $G(p)$ for $p = -250$ mm (—), $p = -215$ mm (—), and $p = 0$ mm (—). The flexible dynamics of $G(p)$ at $p = -215$ mm (- -) and $p = 0$ mm (- -), obtained after subtraction of the rigid-body dynamics (—) according to (3), clearly highlight the difference in compliance as function of position at low frequencies.

B. Estimation of the Snap Parameter

To estimate the snap parameter at various positions, the estimation method of Section III-D is used. The basis functions in (31) are selected as $\Psi(r_j) = \left[(\mathrm{d}^2/\mathrm{d}t^2)r_j \;\; (\mathrm{d}^4/\mathrm{d}t^4)r_j\right]$ to obtain the acceleration and snap feedforward parameters $\theta[1]$ and $\theta[2]$, respectively. An accurate model $\hat{G}$ of the beam at $p = 0$ mm, as shown in Fig. 6, is constructed using the Euler–Bernoulli beam theory and is used to construct the model $\widehat{SG}$ for the ILC solution of (34). This model is position-independent, hence there is substantial model mismatch depending on the actual position of the output. In Fig. 7, it is shown that ILC significantly reduces the two-norm of the error compared to only feedback of trial 1 and obtains accurate estimates of the feedforward parameters.

C. GP of the Snap Parameter

A GP with prior knowledge of the snap parameter is constructed to model the snap parameter as a continuous function of position, as described in Section III-B. Since the flexible beam is symmetric around $p = 0$ mm, that is, the middle of the beam, a zero-mean prior with an axis-aligned reflective symmetric kernel is chosen to reflect this first-principles

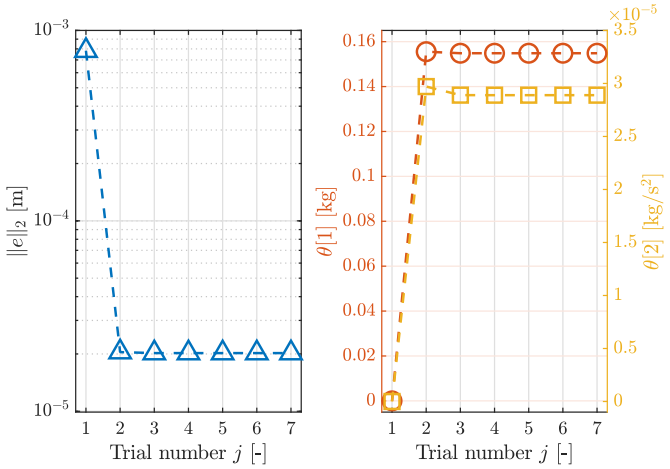

Fig. 7. Error two-norm (-△-) and acceleration parameter $\theta[1]$ (-○-) and snap parameter $\theta[2]$ (-□-) per ILC trial number at $p = 0$ mm.

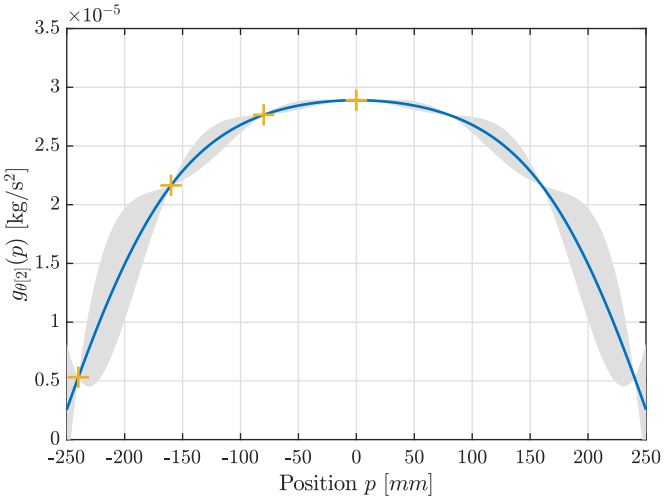

Fig. 8. Snap parameter modeled by GP using the axis-aligned reflective symmetric squared-exponential kernel and equidistant-spaced grid P^G with mean (—), training data P^G (+), and $\pm 2\sigma$ bounds ( ). Note the symmetry in the mean and variance due to the kernel choice. The equidistantly spaced grid does not fully utilize the prior knowledge incorporated in the kernel, hence, user intervention is required to construct an equidistantly spaced grid that, for example, avoids the center position.

knowledge. The axis-aligned reflective symmetric kernel [43, Ch. 2] is the sum of two kernels [16, Sec. 4.2.4] and is defined as

$$k_{\text{aars}}(p, p') := k_{\text{se}}(p, p') + k_{\text{se}}(p, -p') \tag{35}$$

with

$$k_{\text{se}}(p, p') := \sigma_f e^{-\frac{(p-p')^2}{2\ell^2}} \tag{36}$$

where k_{se} is a squared-exponential kernel, ℓ is the characteristic length-scale defining the smoothness, and σ_f the output variance that defines the average distance of the function from its mean, that is, a scaling factor. The hyperparameters ℓ, σ_f, and σ are optimized based on marginal likelihood optimization, see Section III-B4.

D. Comparison of Mutual Information Optimization and Equidistantly Spaced Grid

To highlight the automated experiment design of mutual information optimization, a comparison between MI optimization and an equidistantly spaced grid is performed. In total, $n = 4$ training positions are selected for both methods. The equidistantly spaced grid is defined as $P^G = [-240, -160, -80, 0]$, that is, the positions are distributed equidistantly over half the operating range due to knowledge about symmetry. For each position p_i, the ILC estimate of the snap parameter at trial $j = 7$, $\theta_7[2]$, is considered as training data y_i. The resulting GP regression of the snap feedforward parameter $g_{\theta[2]}$ using the grid is shown in Fig. 8. Clearly, using an equidistantly spaced grid does not fully utilize the prior knowledge incorporated in the kernel, as measured in the center position, that is, the symmetry point $p = 0$ mm, can be regarded as measuring twice on the boundary due to the symmetry.

The a priori MI approach is followed (see Section III-C3), that is, the hyperparameters are fixed beforehand and optimized based on marginal likelihood optimization (see Section III-C). The set of all possible training positions $\bar{P} = [-250, -249, \ldots, 250]$ and the initial point is $P_1^{\text{MI}} = -240$ mm, and repeatedly the next near-optimal greedy MI set P_i^{MI} for $i = 1, 2, 3, 4$ is derived using (26), resulting in $P_4^{\text{MI}} = [-240, -142, -52, 196]$. The evolution of $\delta(p)$ of (27) for $i = 1, 2, 3, 4$ in (23) is shown in Fig. 9. Indeed, the a priori MI optimization scheme clearly utilizes prior knowledge of symmetry and selects the position with the highest information gain to distribute the training data near optimally without any additional user effort. The snap parameter modeled by the GP with MI-optimized training positions is shown in Fig. 10. Note that the variance is lower for positions far from the center position compared to the grid approach in Fig. 8 due to optimization of the training positions.

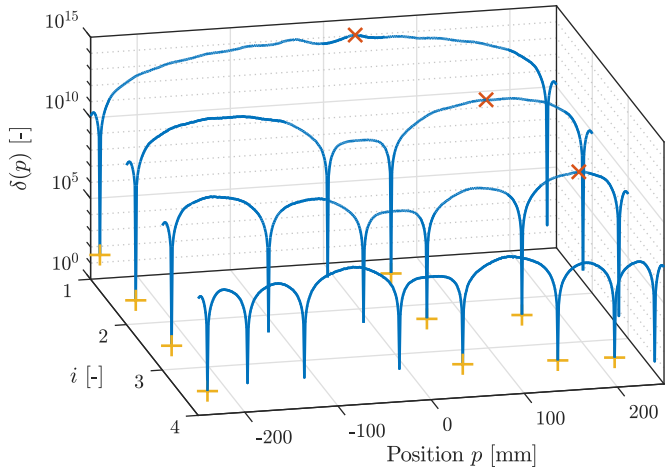

Fig. 9. Evolution of mutual information metric $\delta(p)$ of (27) per number of training positions $i = 1, 2, 3, 4$. The training position are denoted with (+) and start with $P_1^{\text{MI}} = -240$ mm. The training position that maximizes $\delta(p)$ is indicated by (×) and is added to training data P_{i+1}^{MI} according to (23). Clearly, $\delta(p)$ is symmetric around $p = 0$ mm, and prior knowledge is exploited in mutual information optimization to distribute the training positions.

E. Results

The performance with respect to the two-norm of the error is evaluated at every position in the operating range, comparing acceleration feedforward, position-independent snap feedforward, and the proposed method using MI optimization and an equidistantly spaced grid. For the GP-based methods,

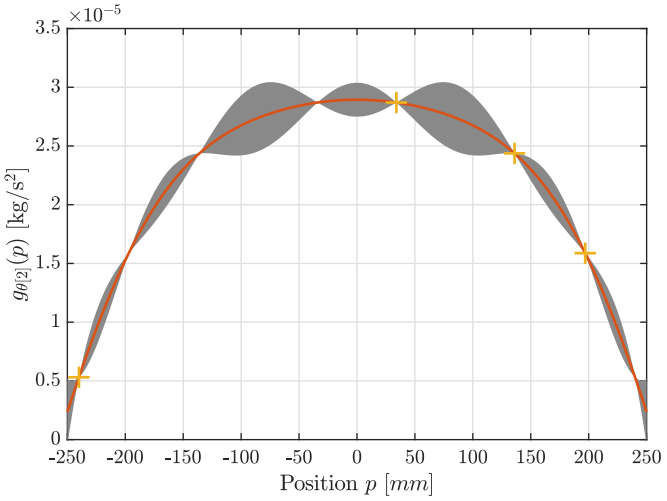

Fig. 10. Snap parameter modeled by GP using the axis-aligned reflective symmetric squared-exponential kernel and MI-optimized training data P^{MI} with mean (—), training data P^{MI} (+), and $\pm 2\sigma$ bounds ( ). Note that the training positions are distributed optimally without any user intervention compared to using an equidistantly spaced grid as in Fig. 8.

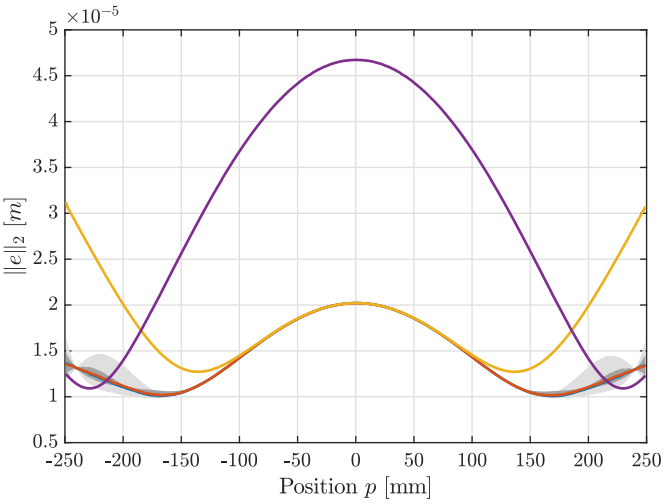

Fig. 11. Error two-norm comparison of position-dependent snap feedforward with the proposed method with an equidistantly spaced grid (—) with $\pm 2\sigma$ bounds ( ), the proposed method using MI optimization (—) with $\pm 2\sigma$ bounds ( ), position-independent snap feedforward (—), and acceleration feedforward (—) per position p. Clearly, the performance of acceleration feedforward is significantly lower than the other approaches, and using the equidistantly spaced grid leads to a higher variance, hence, the MI approach is significantly better.

the snap parameter is regressed using (15) for the entire operating range. For the acceleration feedforward and the position-independent snap feedforward, the parameters learned with ILC at $p = 0$ mm are used for all positions. The error two-norm of the different approaches are shown in Fig. 11. Clearly, the position-independent snap feedforward approach highlights the benefits of snap feedforward compared to acceleration feedforward. The grid and MI-based approaches both outperform the position-independent snap feedforward approach, especially toward the edges of the beam where a factor 2 reduction in the two-norm of the error is observed. In the center position, the least performance gain is obtained, as is expected where the center feedforward parameters are estimated. Note that the grid- and MI-based approaches differ slightly in performance depending on the

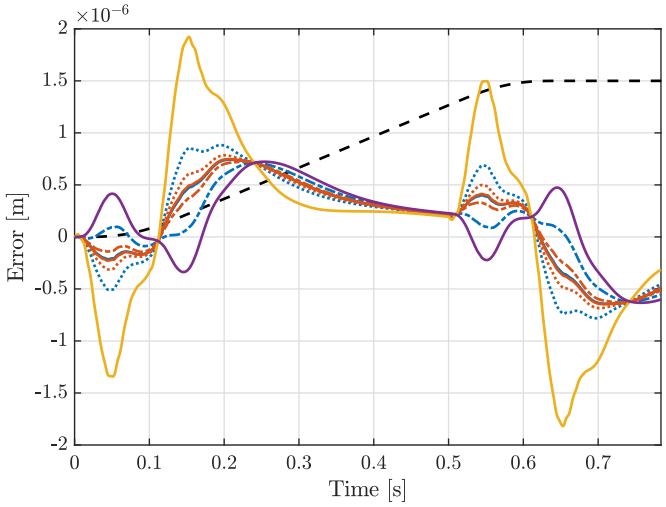

Fig. 12. Error time-domain signals of using position-dependent snap feedforward with the proposed method with an equidistantly spaced grid (—) with -2σ and 2σ bounds, respectively (-·-and ·····), the proposed method using MI optimization (—) with -2σ and 2σ bounds, respectively (-·-and ·····), and position-independent snap feedforward (—) and acceleration feedforward (—) at $p = -220$ mm. The maximum error of the equidistantly spaced grid is slightly worse than the MI-optimized method due to the difference in variance.

position and can be attributed to slight variations in the GP due to the distance between test positions and the training data. Moreover, in Fig. 12, the time-domain error is shown for $p = -220$ mm, highlighting that the peak error is significantly reduced using the GP-based approaches. The time-domain error for the parameter values of $\mathbb{E}[g(p) : p = -220] \pm \sigma$ is also shown in Fig. 12, reflecting the uncertainty of the GP estimates on the time-domain error. The MI approach has a lower variance and thus a lower time-domain error for $p = -220$ mm compared to the equidistantly spaced grid. To conclude, the position-dependent snap parameter can be modeled using a GP with prior knowledge and is applicable in CS1. The MI approach enables automated experiment design and reduces user intervention by using the prior knowledge specified.

V. Case Study CS2: Motor Force Constant

In this section, the case study CS2 is analyzed for the proposed method on an industrial wirebonder machine subjected to the force ripple, constituting Contribution 4). First, the wirebonder setup, design aspects of ILC, and the GPs are described. Fourth, the relation between the GP of the position-dependent acceleration parameter and the motor force constant is explained. Fifth, the results of the proposed method are presented in comparison to position-independent feedforward.

A. Wirebonder Experimental Setup

The experimental setup is an industrial wirebonder by ASMPT, shown in Fig. 13, and consists of a stacked xyz-stage design as depicted in Fig. 1. Here, the input currents for the linear synchronous motors of the x- and y-axis and the voice-coil motor of the z-axis are considered as inputs, and the encoder measurements of the positions of the x- and y-stage and rotation of the z-stage are considered as outputs. A schematic view of the actuator for the x-axis only is shown

Fig. 13. Industrial wirebonder from ASMPT consisting of a stacked xyz-stage, as shown in Fig. 1, and with position-dependent actuators, as schematically shown in Fig. 3 (right).

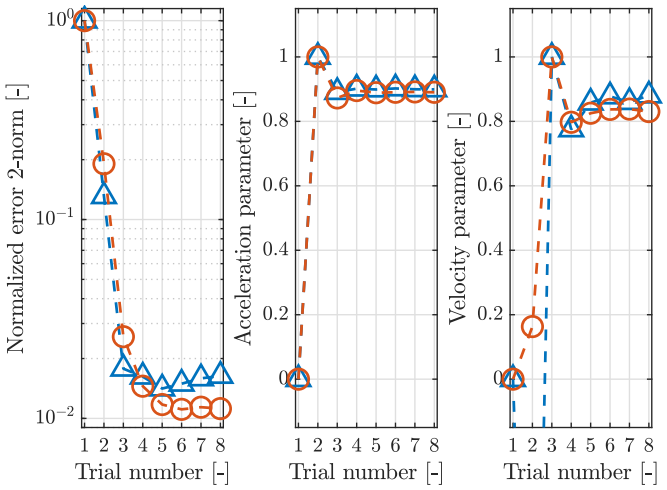

Fig. 14. Normalized error two-norm per ILC trial (left) in the center location with corresponding acceleration parameters (middle) and velocity parameters (right), where the x-axis is (-△-) and the y-axis is (-○-). The error two-norm and the feedforward parameters converge within eight trials.

in Fig. 3 (right), where the force is expected to be a function of position and current due to the movement of the coils with respect to the magnets. Regarding the stacked-stage design of the machine, only the actuator force of the x-stage is expected to be a function of x-position, and the actuator force of the y-stage is expected to be a function of both x- and y-position, hence, the influence of z on position dependence is neglected.

This experimental validation aims to achieve an accurate position-dependent feedforward for any given position $p = (x, y)$, where $(x, y) \in [-1, 1]^2 \subseteq \mathbb{R}^2$ covers the entire normalized operating range of the machine. The reference trajectories for the x- and y-axes, denoted by r_x and r_y, respectively, are fifth-order motion profiles with normalized distances of 0.0075 and 0.0213 with respect to the total operating range for x and y, respectively, and are designed so that the machine stays close to a fixed position p. The z-stage is regulated to zero by feedback control, possibly influencing the performance of the y-stage due to rigid-body coupling between the y- and z-stages. Due to confidentiality reasons, all results related to the industrial wirebonder are normalized with respect to the maximum value of the respective figure.

B. Estimation of the Feedforward Parameters

The feedforward parameters are accurately estimated as described in Section III-D. The basis functions of (31) are derived as follows. The stacked xy-stage has a high stiffness between actuators and sensors, that is, the rigid-body dynamics are expected to be dominant, and friction effects in the bearings are known to significantly affect the performance. Furthermore, a certain level of rigid-body coupling is expected between the z-stage in relation to the y-stage and no coupling between the x- and y-stages is expected. Taking into account these effects for the x- and y-axes, the basis function matrix in (31) is defined as

$$\Psi(r_j) = \begin{bmatrix} \frac{\mathrm{d}^2}{\mathrm{d}t^2} r_{x,j} & \frac{\mathrm{d}}{\mathrm{d}t} r_{x,j} & 0 & 0 & 0 \\ 0 & 0 & \frac{\mathrm{d}^2}{\mathrm{d}t^2} r_{y,j} & \frac{\mathrm{d}}{\mathrm{d}t} r_{y,j} & \psi(r_{y,j}) \end{bmatrix} \tag{37}$$

to obtain acceleration and velocity feedforward for both stages in addition to rigid-body feedforward of the coupling between y- and z-stages, described by the nonlinear basis function $\psi(r_{y,j})$. This is a decentralized feedforward design (see [41]), due to the lack of coupling between the x- and y-stages. The model $\widehat{SG}$ used in (34) is constructed using a parametric model $\hat{G}$ obtained from frequency response measurements and is position independent, this model mismatch may lead to slow convergence of the ILC algorithm in Section III-D.

In Fig. 14 (left), the two-norm of the error for both axes in the center location is shown, highlighting the convergence of the error in eight trials. Moreover, in the middle and right plots, respectively, the acceleration and velocity parameters are shown for both axes and variance in the parameters is clearly present.

To construct an accurate training dataset for each parameter in each position, 20 ILC trials are performed in each position. The corresponding training position for each parameter is $p_i = [x, y]$, where x, y denotes the initial position of the respective axis for the training position $i = 1, \ldots, n$. At every training position p_i, 8 estimates of the feedforward parameter $(\theta_j)_{j=13,\ldots,20}$ are taken to construct the data y_i corresponding to p_i. Importantly, note that due to the iterative nature of ILC, the parameters $(\theta_j)_{j=13,\ldots,20}$ might not be i.i.d. as assumed in the noisy measurements of (13). It is expected that for these iteration numbers, ILC is close to the converged value and the trial-varying disturbances lead to approximately independent contributions, so that in practice the results are not influenced much.

Next, optimization of training positions p_i and construction of the GPs are investigated.

C. GPs of the Feedforward Parameters

To describe the position dependence of the feedforward parameters, for each parameter, a separate GP is constructed with a zero mean prior and a squared-exponential covariance function (36). Thus, no prior knowledge of symmetry is incorporated, as in CS1 of Section IV, nor is any periodicity

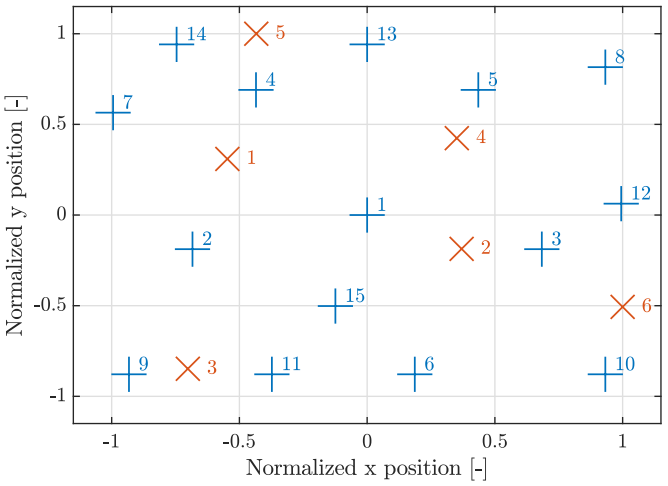

Fig. 15. Normalized training positions P_{15}^{MI} (+) in the xy plane determined using MI optimization and six arbitrary test positions (×) for evaluating the performance of the experiments.

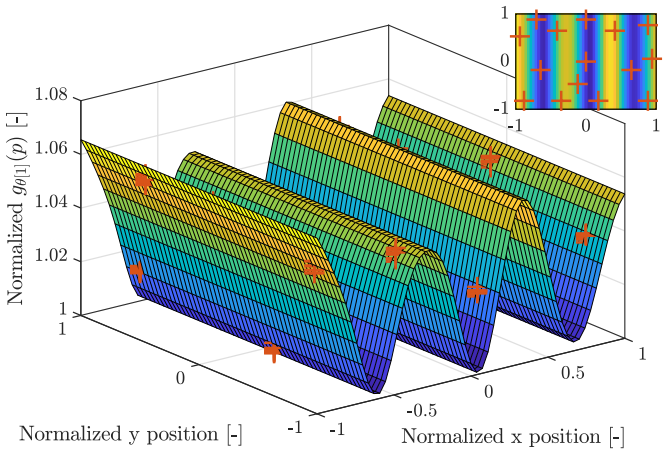

Fig. 16. Normalized GP of acceleration parameter of the x-axis, $g_{\theta_1}(p)$, with training positions (×). Top view with training positions P_{15}^{MI} (top right).

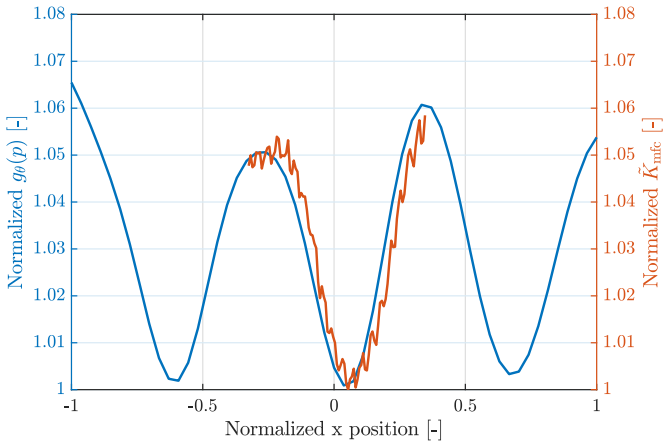

Fig. 17. Normalized GP of the acceleration parameter of the x-axis, $g_\theta(p)$, for the y-axis in center location, that is, $y = 0$ and variation of motor force constant $\bar{K}_{\text{mfc}}(p)$ measured using traditional motor force constant calibration procedure. Note that both methods describe the same position-dependent behavior and this confirms that the position-dependent acceleration parameter modeled by a GP compensates the variation of the motor force constant.

assumed. Moreover, a sequential MI optimization scheme with $n = 15$ total training positions starting at the center position is used. In each consecutive training position, ILCBF is performed to obtain eight estimates of the feedforward parameter, after which the hyperparameters are optimized based on the marginal likelihood optimization. Since MI optimization requires only one GP and multiple GPs are constructed, only the GP of the acceleration parameter of the y-axis is used to determine the next near-optimal training position. The resulting training positions P_{15}^{MI} are depicted in Fig. 15 and show a near-optimal distribution of the training data.

The constructed GP for the acceleration parameter, $\theta[1]$, of the x-axis, $g_{\theta[1]}(p)$, is illustrated in Fig. 16, clearly highlighting position dependence of the parameter in the x-direction and no position dependence in the y-direction, as expected. Furthermore, the GP accurately describes the fluctuation of the acceleration parameter as a continuous function of position using only $n = 15$ training positions, each with eight estimates of $\theta[1]$. The GPs of the other feedforward parameters are omitted for brevity, as the position dependence of these parameters does not exhibit interesting position-dependent behavior.

D. Relation of Position-Dependent Acceleration Parameter and Motor Force Constant

The GP of the acceleration parameter of the x-axis in Fig. 16 exhibits a clear position-dependent behavior that can be related to the variation in the motor force constant caused by the force ripple in linear synchronous motors. In this section, the position-dependent acceleration parameter GP is related to the motor force constant, and a validation of the GP using a traditional motor force constant calibration procedure is performed.

The relation between the position-dependent acceleration parameter and the motor force constant can be described as follows. Recall (4), if $K_{\text{mfc}}(p)$ differs from its actual value, the feedforward parameters k_f must compensate for this variation. Under the assumption of dominant rigid-body dynamics, the acceleration feedforward is the main contributor to the feedforward signal, hence, the variation in motor force constant is directly captured by a varying acceleration parameter as a function of position.

A traditional motor force constant calibration procedure is performed on the wirebonder to validate the GP. The motor force constant is calibrated by first segmenting the operating range and performing small-amplitude motions in each segment. When comparing the control output necessary for performing that motion, variations of the control output are related to variations in the motor force constant for each individual segment. The motor force constant obtained from this calibration method is shown in Fig. 17 and is compared to the GP $g_{\theta[1]}(p)$ of Fig. 16 for a fixed position of $y = 0$ mm. Clearly, both methods describe the same position-dependent behavior of the actuator, and this confirms that the position-dependent acceleration parameter modeled by a GP compensates for the variation of the motor force constant. A slight offset in Fig. 17 (right) is observed for the calibration of the motor force constant with respect to $g_{\theta[1]}(p)$. This could be explained by the directionality of the reference, as all motions are only in the positive x-direction, or by the definition of the initial

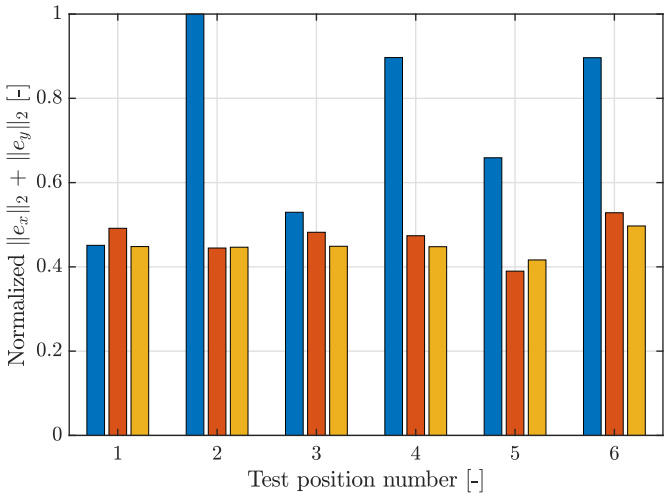

Fig. 18. Normalized error two-norm comparison of position-independent feedforward from center position ( ), the position-dependent GP feedforward approach ( ), and best achievable performance of performing ILCBF locally ( ) that indicates the interpolation error of the GP at the 6 test positions defined in Fig. 15. Clearly, the proposed method achieves an accuracy similar to ILCBF, that is, a very low interpolation error is achieved and it significantly outperforms the position-independent feedforward method.

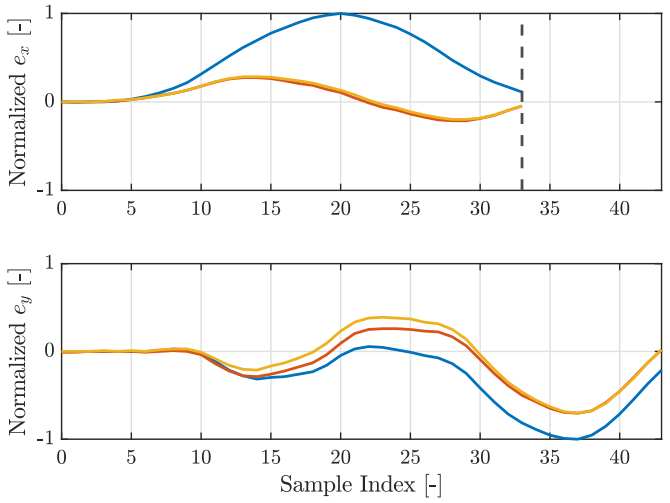

Fig. 19. Normalized time-domain error signals at test position 2 of position-independent feedforward from center position (—), position-dependent GP feedforward approach (—), and best achievable performance of performing ILCBF locally (—) at test position 2. The maximum error of the proposed approach significantly outperforms the position-independent feedforward and almost achieves the performance of performing ILCBF locally.

position of the machine, as the start point of the reference is considered instead of the end or midpoint of the motion, or could be a consequence of position-dependent friction.

E. Results

In Fig. 15, six arbitrary test positions are shown in which the proposed method is compared to position-independent feedforward, and the results are depicted in Fig. 18. The position-independent feedforward method employs the feedforward parameters estimated at the center position throughout all six test positions. At each test position, ILCBF is also performed, showing the best possible performance given the feedforward parameterization, thus indicating the interpolation error of the GP. Clearly, the proposed method achieves accuracy similar to ILCBF, that is, a very low interpolation error is achieved using the GP approach, and it significantly outperforms the position-independent feedforward method. In particular, the maximum error in the time domain for the second test position is more than a factor two lower using the proposed method, as shown in Fig. 19. Interestingly, the performance of position-independent feedforward is similar to the GP-based method for test positions 1 and 3, which can be explained by the period of the position dependence of Fig. 16 being equal to the magnet pitch, that is, the distance between magnets, as shown schematically in Fig. 3 (right). To conclude, the GP approach enables accurate feedforward control for the entire operating range of the wire bonder subjected to force ripple, showing the applicability of the framework in CS2.

VI. Conclusion

The presented framework for interpretable and task-flexible position-dependent feedforward via GPs enables accurate control for position-dependent systems over the entire operating range of machines. The data-driven framework combines estimation, interpolation, and optimization methods. ILC with basis functions enables data-driven estimation of feedforward parameters for a given position, GPs enable accurate modeling of feedforward parameters using prior knowledge that is incorporated in the kernel, and mutual information optimization exploits the variance estimates of the GP to near-optimally distribute the training positions for automated experiment design. The benefits of the developed framework are illustrated in a case study of a simplified 1-D wafer stage where improved tracking accuracy is achieved by position-dependent snap feedforward and automated experiment design is enabled by utilizing prior knowledge, for example, symmetry, about the system. In the case study of a wirebonder, significant performance improvement of the developed approach is achieved compared to position-independent feedforward. In addition, position-dependent variations in the motor force constant, which are caused by the force ripple in linear synchronous motors, are clearly compensated for by the acceleration feedforward parameter as it recovers results obtained using a traditional motor force constant calibration method.

Ongoing research focuses on modeling unknown nonlinear effects using GPs that cannot be easily specified in the basis functions [21], extending the basis functions with a different approach to model-order selection [20], estimation and modeling of magnetic saturation in the actuators [44], and extensions to evaluate the GP as function of position, similar to [45].

Acknowledgment

The authors would like to thank Kelvin Kai Wa Yan and Robin van Es for their valuable contributions to this research and the experimental case study.

References

[1] F. Boeren, A. Bareja, T. Kok, and T. Oomen, "Frequency-domain ILC approach for repeating and varying tasks: With application to semiconductor bonding equipment," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 6, pp. 2716–2727, Dec. 2016.

[2] T. Oomen, "Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems," *IEEJ J. Ind. Appl.*, vol. 7, no. 2, pp. 127–140, 2018.

[3] H. Butler, "Position control in lithographic equipment: An enabler for current-day chip manufacturing," *IEEE Control Syst.*, vol. 31, no. 5, pp. 28–47, Jan. 2011.

[4] J. Bolder, J. van Zundert, S. Koekebakker, and T. Oomen, "Enhancing flatbed printer accuracy and throughput: Optimal rational feedforward controller tuning via iterative learning control," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4207–4216, May 2017.

[5] S. Van Der Meulen, R. Tousain, and O. Bosgra, "Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer," *J. Dyn. Syst., Meas. Control, Trans. ASME*, vol. 130, no. 5, 2008, Art. no. 05100616.

[6] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning," *EEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, May 2006.

[7] F. Boeren, L. Blanken, D. Bruijnen, and T. Oomen, "Optimal estimation of rational feedforward control via instrumental variables: With application to a wafer stage," *Asian J. Control*, vol. 20, no. 3, pp. 975–992, May 2018.

[8] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen, "Batch-to-batch rational feedforward control: From iterative learning to identification approaches, with application to a wafer stage," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 826–837, Apr. 2017.

[9] M. Groot Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "LPV control for a wafer stage: Beyond the theoretical solution," *Control Eng. Pract.*, vol. 13, no. 2, pp. 231–245, Feb. 2005.

[10] R. van der Maas, A. van der Maas, R. Voorhoeve, and T. Oomen, "Accurate FRF identification of LPV systems: ND-LPM with application to a medical X-ray system," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 5, pp. 1724–1735, Sep. 2017.

[11] M. Boerlage, R. Tousain, and M. Steinbuch, "Jerk derivative feedforward control for motion systems," in *Proc. Amer. Control Conf.*, Jun. 2004, pp. 4843–4848.

[12] T. Oomen, "Control for precision mechatronics," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-44184-5_100044.

[13] C. Rohrig and A. Jochheim, "Identification and compensation of force ripple in linear permanent magnet motors," in *Proc. Amer. Control Conf.*, Jun. 2001, pp. 2161–2166.

[14] S. Zhen, P. Chen, X. Chen, F. Qin, and H. Zhou, "Force ripple modeling and minimizing of an ironless permanent-magnet linear synchronous motor," *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 6, pp. 927–935, Jun. 2019.

[15] L. Bascetta, P. Rocco, and G. Magnani, "Force ripple compensation in linear motors based on closed-loop position-dependent identification," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 3, pp. 349–359, Jun. 2010.

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

[17] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Jun. 2008.

[18] M. Poot, J. Portegies, N. Mooren, M. van Haren, M. van Meer, and T. Oomen, "Gaussian processes for advanced motion control," *IEEJ J. Ind. Appl.*, vol. 11, no. 3, pp. 396–407, 2022.

[19] L. Blanken and T. Oomen, "Kernel-based identification of non-causal systems with application to inverse model control," *Automatica*, vol. 114, Apr. 2020, Art. no. 108830.

[20] M. Poot, J. Portegies, and T. Oomen, "Kernel-based learning control for iteration-varying tasks applied to a printer with friction," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2021, pp. 1052–1057.

[21] M. van Meer, M. M. Poot, J. W. Portegies, and T. A. E. Oomen, "Learning nonlinear feedforward: A Gaussian process approach applied to a printer with friction," *IFAC-PapersOnLine*, vol. 55, no. 37, pp. 241–246, 2022, doi: 10.1016/j.ifacol.2022.11.191.

[22] I. Proimadis, Y. Broens, R. Tóth, and H. Butler, "Learning-based feedforward augmentation for steady state rejection of residual dynamics on a nanometer-accurate planar actuator system," *Mach. Learn. Res.*, vol. 144, pp. 1–12, May 2021.

[23] M. van Haren, M. Poot, J. Portegies, and T. Oomen, "Position-dependent snap feedforward: A Gaussian process framework," in *Proc. Amer. Control Conf. (ACC)*, Atlanta, Georgia, Jun. 2022, pp. 4778–4783.

[24] M. van Haren, M. Poot, D. Kostic, R. van Es, J. Portegies, and T. Oomen, "Gaussian process position-dependent feedforward: With application to a wire bonder," in *Proc. IEEE 17th Int. Conf. Adv. Motion Control (AMC)*, Padova, Italy, Feb. 2022, pp. 268–273.

[25] G. J. Balas, I. Fialho, A. Packard, J. Renfrow, and C. Mullaney, "On the design of LPV controllers for the F-14 aircraft lateral-directional axis during powered approach," in *Proc. Amer. Control Conf.*, Jun. 1997, pp. 123–127.

[26] J. van Zundert and T. Oomen, "Stable inversion of LPTV systems with application in position-dependent and non-equidistantly sampled systems," *Int. J. Control*, vol. 92, no. 5, pp. 1022–1032, May 2019.

[27] N. Kontaras, M. Heertjes, and H. Zwart, "Continuous compliance compensation of position-dependent flexible structures," *IFAC-PapersOnLine*, vol. 49, no. 13, pp. 76–81, 2016.

[28] J. V. Amerongen, "A MRAS-based learning feed-forward controller," *IFAC Proc. Volumes*, vol. 39, no. 16, pp. 758–763, 2006.

[29] L. Aarnoudse et al., "Control-relevant neural networks for feedforward control with preview: Applied to an industrial flatbed printer," *IFAC J. Syst. Control*, vol. 27, Mar. 2024, Art. no. 100241.

[30] M. M. Seron, J. H. Braslavsky, and G. C. Goodwin, *Fundamental Limitations Filtering Control* (Communications and Control Engineering). London, U.K.: Springer, 1997.

[31] H. Butler, "Adaptive feedforward for a wafer stage in a lithographic tool," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 875–881, May 2013.

[32] W. K. Gawronski, Ed., *Advanced Structural Dynamics and Active Control of Structures* (Mechanical Engineering Series). New York, NY, USA: Springer, 2004.

[33] S. O. R. Moheimani, D. Halim, and A. J. Fleming, *Spatial Control of Vibration* (Series on Stability, Vibration and Control of Systems, Series A), vol. 10. Singapore: World Scientific, Dec. 2003.

[34] N. J. Dirkx, W. H. Aangenent, and M. M. Van De Wal, "Improving positioning performance via advanced feedforward design with position-dependent compliance identification," in *Proc. ASPE Spring Topical Meeting: Precis. Mech. Syst. Design Control*, 2016, pp. 68–73.

[35] H. Polinder, "Modelling a linear PM motor including magnetic saturation," in *Proc. Int. Conf. Power Electron. Mach. Drives*, 2002, pp. 632–637.

[36] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Eng. Pract.*, vol. 13, no. 2, pp. 145–157, Feb. 2005.

[37] S. Elliot, *Signal Processing for Active Control*. New York, NY, USA: Academic, Dec. 2000.

[38] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. N. Pandey, "Gaussian processes for active data mining of spatial aggregates," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 427–438.

[39] W. F. Caselton and J. V. Zidek, "Optimal monitoring network designs," *Stat. Probab. Lett.*, vol. 2, no. 4, pp. 223–227, Aug. 1984.

[40] A. Krause and C. Guestrin, "Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 449–456.

[41] L. Blanken and T. Oomen, "Multivariable iterative learning control design procedures: From decentralized to centralized, illustrated on an industrial printer," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 4, pp. 1534–1541, Jul. 2020.

[42] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch, "Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer," *Mechatronics*, vol. 24, no. 8, pp. 944–953, Dec. 2014.

[43] D. K. Duvenaud, "Automatic model construction with Gaussian processes," Ph.D. dissertation, Pembroke College, Univ. Cambridge, Cambridge, U.K., 2014. [Online]. Available: https://www.cs.toronto.edu/~duvenaud/thesis.pdf, doi: 10.17863/CAM.14087.

[44] J. van Hulst, M. Poot, D. Kostić, K. W. Yan, J. Portegies, and T. Oomen, "Feedforward control in the presence of input nonlinearities: A learning-based approach," in *Proc. Modeling, Estimation, Control Conf.*, 2022. Ref. [44]: M. M. Poot, J. van Hulst, K. W. Yan, D. Kostic, J. W. Portegies, and T. A. E. Oomen, "Feedforward control in the presence of input nonlinearities: With application to a wirebonder," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1895–1900, 2023, doi: 10.1016/j.ifacol.2023.10.1078.

[45] N. Mooren, G. Witvoet, and T. Oomen, "Gaussian process repetitive control with application to an industrial substrate carrier system with spatial disturbances," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 1, pp. 344–358, Jan. 2023.