Gaussian Process Position-Dependent Feedforward: With Application to a Wire Bonder

Max van Haren*, Maurice Poot*, Dragan Kostić†, Robin van Es†, Jim Portegies‡ and Tom Oomen*§
*Control Systems Technology, Dept. of Mechanical Engineering, Eindhoven University of Technology,
Eindhoven, The Netherlands, email: m.j.v.haren@tue.nl
†ASM Pacific Technology, Centre of Competency, Beuningen, The Netherlands
‡CASA, Dept. of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
§Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands

***Abstract*—Mechatronic systems have increasingly stringent performance requirements for motion control, leading to a situation where many factors, such as position-dependency, cannot be neglected in feedforward control. The aim of this paper is to compensate for position-dependent effects by modeling feedforward parameters as a function of position. A framework to model and identify feedforward parameters as a continuous function of position is developed by combining Gaussian processes and feedforward parameter learning techniques. The framework results in a fully data-driven approach, which can be readily implemented for industrial control applications. The framework is experimentally validated and shows a significant performance increase on a commercial wire bonder.**

I. Introduction

Increasing performance requirements for motion control leads to situations where position-dependency of mechatronic systems cannot be neglected anymore. An example is the wafer stage, where a flexible mode of a wafer is observed differently for each position [1]. Furthermore, H-drive machines, such as large-format printing systems, suffer from position-dependent dynamics due to a changing configuration [2]. Traditionally, position-dependency is neglected and the feedforward controller is kept constant over the machine operating range, resulting in suboptimal performance. Recently, due to the developments in computational power, data-driven techniques, such as Gaussian Processes (GPs), are getting more relevant for motion control.

Traditional Linear Time-Invariant (LTI) feedforward design attempts to compensate for a known reference signal of a system. Typically, feedforward controllers are based on models. For low frequencies, motion systems can generally be modeled as a rigid-body, resulting in the well-known acceleration feedforward. Acceleration feedforward controllers can directly be extended to compensate for higher-order or non-linear dynamics such as flexible mechanics [3]. The parameters used in traditional LTI controllers can be tuned manually in a straightforward manner [4]. Nevertheless, traditional LTI feedforward does not compensate for position-dependent effects.

Position-dependent feedforward design can compensate for position-dependent behavior of systems. For this purpose, feedforward parameters can be determined in a grid and estimated with parametric regression techniques such as linear interpolation. However, interpolations have approximation errors since the dependency between position and feedforward parameters is generally unknown.

High motion control performance for systems with position-dependent dynamics can furthermore be achieved through the use of Linear Parameter Varying (LPV) or non-linear inversion-based control of the system. First, LTI dynamics can be scheduled according to the current configuration of the LPV system, resulting in high control performance for e.g. wafer stages [5] or xy-positioning tables [6]. Second, data-driven learning techniques, such as Iterative Learning Control (ILC), can be extended to LPV systems, which results in high performance through learning [7]. Third, a non-linear inversion-based output tracking solution can be applied, resulting in asymptotically exact output tracking [8]. LPV model-free approaches are investigated in [9], directly learning LPV controllers from data, but are at present not competitive with model-based designs. The high performance achieved by using LPV or non-linear control typically requires accurate and extensive modeling, that is often very challenging and the high cost and complexity are usually not justified for industrial control applications.

Although feedforward design has improved significantly compared with traditional acceleration feedforward, a feedforward with systematic tuning for position-dependent effects, capable of estimation at any arbitrary position, is currently lacking. This paper models feedforward parameters as a continuous function of position through a GP [10], [11], which allows for the compensation of position-dependent effects without a full LPV or non-linear model. In addition, a GP is non-parametric and therefore does not require an assumption on the parametric form between the position and the feedforward parameters. In this paper, the feedforward parameters of a system are learned in a trial-to-trial fashion using ILC with Basis Functions (ILCBF) [12]. The contributions include:

This work was carried out when the first author was an MSc student, in collaboration with ASM PT. Currently, the first author is a PhD researcher with the IMOCO4.E consortium, which has received funding from the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E). The Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme. In addition, this work is part of the research programme VIDI with project number 15698, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

C1 a generic framework to model feedforward parameters as a function of position using GPs, which can be readily implemented for industrial applications,

C2 ILCBF to automatically learn feedforward parameters for multiple fixed positions, suitable for industrial machines, that are directly used in the GPs,

C3 application and validation of the framework to a state-of-the-art industrial experimental setup, showing the capabilities of the framework.

The outline of this paper is as follows. In Section II, the problem that is considered in this paper is defined. In Section III, the method for modeling feedforward parameters as a GP and an approach to automatically learn feedforward parameters with ILCBF is described, leading to contributions C1 and C2. In Section IV, a case study on an experimental setup is performed, constituting contribution C3. Finally, in Section V, concluding remarks are given.

Notation: Systems can be single-input single-output or multiple-input multiple-output with n_i inputs and n_o outputs. All systems are discrete-time, unless stated otherwise, with discrete-time $k \in \{0, 1, \ldots, N-1\}$. Continuous time systems are transformed in their discrete-time counterpart using finite difference approximation. The trial number is indicated with the index j. Signals are assumed to be of length N. The weighted 2-norm of a vector $x \in \mathbb{R}^N$ is denoted as $\|x\|_W := \sqrt{(x^\top W x)}$, where $W \in \mathbb{R}^{N\times N}$ is a weighting matrix. Matrix $A \in \mathbb{R}^{N\times N}$ is positive (semi-)definite if and only if $x^\top A x \geq 0$, $\forall x \neq 0 \in \mathbb{R}^N$ and is denoted as $A \succeq 0$.

II. Problem Definition

In this section, the problem for determining a position-dependent feedforward controller is formulated. First, a problem setup is given, including the systems considered and the parametrization of the feedforward signal. Second, the three categories of position-dependent effects in mechatronic systems considered are elaborated upon. Finally, a hypothesis of the largest contribution to the position-dependency is made and the problem addressed in this paper is defined.

A. Problem Setup

The considered class of position-dependent systems are spatially distributed LTI systems [13]

$$y(k) = G(\rho, q^{-1})u(k), \tag{1}$$

with output $y(k) \in \mathbb{R}^{N\times n_o}$, input $u(k) \in \mathbb{R}^{N\times n_i}$, system $G(\rho, q^{-1}) \in \mathbb{R}^{n_o\times n_i}$, the initial position $\rho \in \mathbb{R}^{n_o}$ and q denotes the forward-shift operator, i.e. $q^{-\tau}a(k) = a(k-\tau)$. Spatially distributed LTI systems are generally applicable to systems with slowly varying position-dependency, i.e., the position-dependency is mostly exerted due to the initial position and not due to the reference signal. The wire bonder in Fig. 1 is a benchmark example, since references are relatively short compared with the machine operating range.

The applied control structure can be seen in Fig. 2. Typically, the design of feedforward controller F tries to reduce the reference-induced error to zero, i.e.,

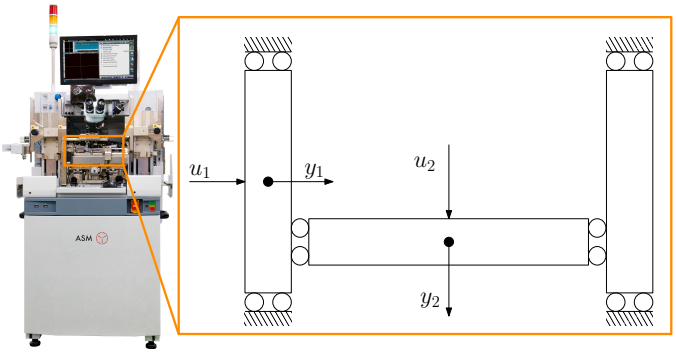

Fig. 1. Commercial wire bonder which is used for the experimental case study. The system might have position-dependent dynamics, due to the changing mass distribution.

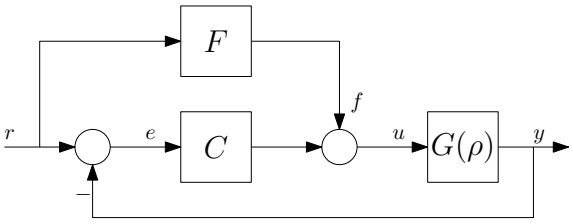

Fig. 2. Control structure considered in this paper, consisting of feedforward controller F, feedback controller C and the spatially distributed LTI system $G(\rho)$.

$$e(k) = 0 = r(k) - G(\rho, q^{-1})u(k),$$

resulting in the ideal feedforward controller $F = G^{-1}(\rho, q^{-1})$. Often, the feedforward design is based on the physical model of a system. Consider, e.g., a finite-difference approximation of a (position-independent) mass-damper system

$$G_{md}(q^{-1}) = \frac{1}{m\frac{(1-q^{-1})^2}{T_s^2} + d\frac{1-q^{-1}}{T_s}},$$

with T_s the sampling time of the system, m the mass of the system and a damping coefficient d. Zero reference-induced error is achieved by the ideal feedforward controller $F(q^{-1}) = m\frac{(1-q^{-1})^2}{T_s^2} + d\frac{1-q^{-1}}{T_s}$. However, due to model uncertainties, the exact system parameters are unknown and therefore the controller takes the form

$$F(q^{-1}) = \begin{bmatrix} \frac{(1-q^{-1})^2}{T_s^2} & \frac{1-q^{-1}}{T_s} \end{bmatrix} \begin{bmatrix} \hat{m} \\ \hat{d} \end{bmatrix} = \Psi(q^{-1})\theta,$$

with $\hat{\cdot}$ an estimate or modeled value in the feedforward parameters $\theta \in \mathbb{R}^{n_\theta\times n_i}$ and the basis function matrix $\Psi \in \mathbb{R}^{n_o\times n_\theta}$. Due to the position-dependency of system $G(\rho, q^{-1})$, best performance is achieved by modeling the feedforward parameters θ as a function of position.

B. Considered Position-Dependent Effects

Position-dependency of the system $G(\rho, q^{-1})$ can be caused by several factors, here separated in actuation, mechanical

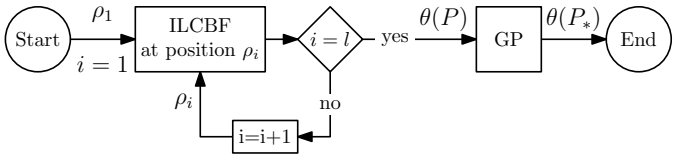

Fig. 3. Schematic illustration of the developed framework, where feedforward parameters are being modeled as a function of position using a combination of ILCBF and GPs.

or sensing. First, actuation can cause position-dependency due to for instance cogging [14] or varying magnetic flux density [15]. Second, mechanical position-dependency can directly affect the feedforward parameters that achieve optimal performance. Examples of such are raster scanning of atomic force microscopy [16] or position-dependent gantry systems, such as the large-format printer [2]. Lastly, since sensors are typically attached to the fixed world and motion systems are moving by definition, deformations or movements of the system are observed differently for particular positions. This is for example seen in lithography, where a flexible mode of a wafer is observed differently [1]. Due to the high costs associated with modeling each position-dependent effect and since manual tuning for all operating positions is infeasible, a data-driven approach for position-dependent feedforward is necessitated.

C. Hypotheses and Problem Definition

Often, flexible mechanics and the observation thereof are the largest contribution of position-dependency of mechatronic systems, see for instance [1], [17]. This is anticipated due to the recent developments in lightweight design of mechatronic systems, typically resulting in more flexible structures. Furthermore, actuation can have significant contribution on the position-dependency as well, but is generally compensated for in an earlier step, e.g., by using calibration [18].

Position-dependent dynamics of system $G(\rho, q^{-1})$ is not compensated for by using constant feedforward parameters. Hence, the aim of this paper is to model feedforward parameters θ as a continuous function of position, such that it can compensate for any unknown position-dependent effect, without an LPV model or specification of a parametric form, that might lead to estimation errors.

III. Approach

In this section, feedforward parameters are modeled as a function of position, such that it can compensate for position-dependent effects, hence constituting contribution C1. Furthermore, a data-driven feedforward parameter learning technique is presented, constituting contribution C2. First, GPs are investigated, followed by the application of GPs for feedforward parameters. Third, ILCBF is presented to learn feedforward parameters for a single position. Finally, an example and overview of the framework is presented, which can schematically be seen in Fig. 3.

A. Gaussian Processes

A GP is defined as a collection of random variables $f(\rho)$, indexed by ρ, such that the joint distribution of any finite subset of random variables is multivariate Gaussian. A GP is written as

$$f(\rho) \sim \mathcal{GP}\big(m(\rho), k(\rho, \rho')\big), \tag{2}$$

which shows a GP is fully defined by the covariance function $k(\rho, \rho')$ and the mean function $m(\rho)$,

$$\begin{aligned} k\left(\rho, \rho'\right) &= \mathbb{E}\left[\left(f(\rho) - m(\rho)\right)\left(f\left(\rho'\right) - m\left(\rho'\right)\right)\right], \\ m(\rho) &= \mathbb{E}[f(\rho)]. \end{aligned} \tag{3}$$

The mean function m can be interpreted as the mean at any input point and the covariance function k as the similarity between values of $f(\rho)$ on different inputs ρ and ρ'.

B. Gaussian Processes for Position-Dependent Feedforward Parameters

In this section, feedforward parameters are estimated on test positions $\rho \in P_*$, given feedforward parameters learned on training positions $\rho \in P$ using GPs, leading to contribution C1. The test and training positions are defined as

$$\begin{aligned} P_* &= \begin{bmatrix} \rho_{1_*} & \rho_{2_*} & \dots & \rho_{l_*} \end{bmatrix}^\top \in \mathbb{R}^{l_* \times n_o}, \\ P &= \begin{bmatrix} \rho_1 & \rho_2 & \dots & \rho_l \end{bmatrix}^\top \in \mathbb{R}^{l \times n_o}, \end{aligned} \tag{4}$$

with $l_*, l \in \mathbb{N}$ the amount of test and training positions, respectively. Observations of the feedforward parameters, i.e. $\theta(P)$, are contaminated with noise,

$$\theta(P) = f(P) + \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 I). \tag{5}$$

The continuous function of position $\theta(P_*)$ and the function observations $\theta(P)$ are assumed to be random variables and have a joint multivariate Gaussian distribution

$$\begin{bmatrix} \theta(P) \\ \theta(P_*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(P,P) + \sigma_n^2 I & K(P,P_*) \\ K(P_*,P) & K(P_*,P_*) \end{bmatrix}\right), \tag{6}$$

with covariance matrices $K_y := K(P,P) + \sigma_n^2 I \in \mathbb{R}^{l \times l}$, $K_* := K(P,P_*) = K(P_*,P)^\top \in \mathbb{R}^{l \times l_*}$ and $K_{**} := K(P_*,P_*) \in \mathbb{R}^{l_* \times l_*}$, which specify similarity between outputs on different positions. σ_n^2 is an approximation of the noise variance on the observations in (5). Note that the mean function is assumed to be zero, which is not strictly necessary, see e.g. [10, Section 2.7]. The joint distribution in (6) can be conditioned on function observations using Bayes' rule, resulting in the posterior distribution

$$\theta(P_*)\Big|\left[P, P_*, \theta(P)\right] \sim \mathcal{N}\big(\bar{\theta}(P_*), \operatorname{cov}(\theta(P_*))\big), \tag{7}$$

with posterior mean and posterior covariance

$$\begin{aligned} \bar{\theta}(P_*) := \mathbb{E}(\theta(P_*)) &= K_*^\top K_y^{-1} \theta(P), \\ \operatorname{cov}(\theta(P_*)) &= K_{**} - K_*^\top K_y^{-1} K_*. \end{aligned} \tag{8}$$

In combination with a suitable covariance function, (8) is used to estimate feedforward parameters, given observations of the feedforward parameters.

C. Learning Feedforward Parameters using Basis Functions for a Fixed Position

In this section, feedforward parameters are learned for multiple fixed positions to serve as training data for the GP, leading to contribution C2. Here, ILCBF is used to learn the parameters, but the framework can directly be extended to other feedforward parameter tuning approaches.

The optimization criterion in ILCBF is specified as [19]

$$V(\theta_{j+1}) = \|e_{j+1}(k)\|^2_{W_e} + \|f_{j+1}(k)\|^2_{W_f} + \|f_{j+1}(k) - f_j(k)\|^2_{W_{\Delta f}}, \quad (9)$$

with weighting matrices $W_e \succ 0$ and W_f, $W_{\Delta f} \succeq 0$ and θ_j the feedforward parameters in trial j. The error e_{j+1} is

$$\begin{aligned} e_{j+1}(k) &= S(q^{-1})r(k) - S(q^{-1})G_0(q^{-1})f_{j+1}(k) \\ &= e_j(k) - S(q^{-1})G_0(q^{-1})\big(f_{j+1}(k) - f_j(k)\big), \end{aligned} \quad (10)$$

where now, G_0 is for instance a nominal model of a position-dependent system. The feedforward signal is parameterized in terms of the feedforward parameters θ_j, i.e., $f_j(k) = \Psi(q^{-1})r(k)\theta_j$. The feedforward parameters are updated as

$$\theta^*_{j+1} = \arg\min_{\theta_{j+1}} V\left(\theta_{j+1}\right). \quad (11)$$

Since the feedforward force is chosen linearly in the feedforward parameters, the optimization criterion in (9) becomes quadratic in θ_{j+1}. Hence, an analytic solution to (12) is [20]

$$\begin{aligned} \theta_{j+1} &= Le_j + Q\theta_j, \\ L &= R^{-1}\left(\Psi^\top G_0^\top S^\top W_e\right), \\ Q &= R^{-1}\Psi^\top\left(G_0^\top S^\top W_e G_0 S + W_{\Delta f}\right)\Psi, \\ R &= \left(\Psi^\top\left(G_0^\top S^\top W_e G_0 S + W_f + W_{\Delta f}\right)\Psi\right), \end{aligned} \quad (12)$$

where (q^{-1}) and (k) have been left out for brevity. The parameter update in (12) leads to monotonic convergence of $\|f_j(k)\|$, provided matrices W_e, W_f and $W_{\Delta f}$ are selected properly [19]. Robustness, with respect to model mismatch due to the position-dependent dynamics, can be enforced by increasing W_f. Now, (12) can be used in combination with the error in trial j to compute a new set of feedforward parameters θ_{j+1} for a fixed training position.

D. Developed Framework and Example

The combination of GPs and ILCBF is used to model feedforward parameters as a function of position. The framework is schematically shown in Fig. 3. First, ILCBF is performed l times on the positions $\rho \in P$, where P is chosen by the user. Second, the feedforward parameters learned in ILCBF, i.e. the training data $\theta(P)$, can be used in a GP regression. Finally, the GP results in the feedforward parameters modeled as a function of the test inputs P_*, i.e. $\theta(P_*)$.

Example 1. Consider the system with a spatially-distributed mass

$$G_m(\rho, q^{-1}) = \frac{1}{\bar{m}\left(1 - 2\left(\frac{1}{2} - \rho\right)^2\right)\frac{(1-q^{-1})^2}{T_s^2}} \quad \forall \rho \in [0, 1],$$

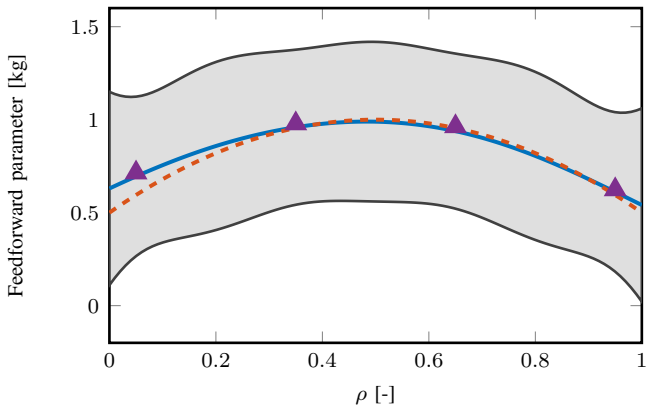

Fig. 4. Example GP regression created using the framework developed of a system with a spatially-distributed mass. The training data (▲) is used to create the posterior mean (—) from (8) and the uncertainty bound $\bar{\theta}(P_*) \pm 2\sigma$ (■). An accurate representation of the true mass (- -) is achieved with the GP regression.

with $\bar{m} = 1$ kg a nominal mass. The feedforward controller is designed using acceleration feedforward, i.e.,

$$f_j(k) = \Psi(q^{-1})r(k)\theta_j = \ddot{r}(k)\theta_j.$$

The feedforward parameters are learned on 4 positions $\rho \in P$, with $P = \begin{bmatrix}0.05 & 0.35 & 0.65 & 0.95\end{bmatrix}$, using ILCBF. The feedforward parameters have bias with respect to the true system values due to measurement noise, which is typically observed in ILCBF [21]. The framework is applied using a squared exponential covariance function, see e.g. [10, Section 4.2]. Fig. 4 shows how the framework models a position-dependent feedforward parameter, resulting in an accurate model. ■

IV. Experimental Case Study

In this section, position-dependent feedforward with GPs is applied to a benchmark experimental setup, hence constituting contribution C3. First, the example setup is presented, followed by the application and results of the framework. Finally, a discussion is presented regarding the outcome of the experiments.

A. Experimental Setup

Position-dependent effects hamper the performance of semiconductor back end equipment, such as the commercial wire bonder seen in Fig. 1. Position-dependent dynamics can be caused due to several reasons, e.g. position-dependent actuators, flexible dynamics and changing configuration. The wire bonder consists of two inputs u_1 and u_2 and two outputs y_1 and y_2. The feedforward parameters are modeled as a function of the initial position of the machine for y_1 and y_2. The feedforward force f is parameterized using

$$\Psi(q^{-1})r(k) = \begin{bmatrix} \dot{r}_1(k) & \ddot{r}_1(k) & 0 & 0 & 0 \\ 0 & 0 & \dot{r}_2(k) & \ddot{r}_2(k) & \psi_2(k) \end{bmatrix}, \quad (13)$$

with $\psi_2(k)$ a non-linear basis function. The references r_1 and r_2 are polynomial trajectories, see [22]. The feedforward parameters are defined as

$$\theta = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5\end{bmatrix}^\top, \quad (14)$$

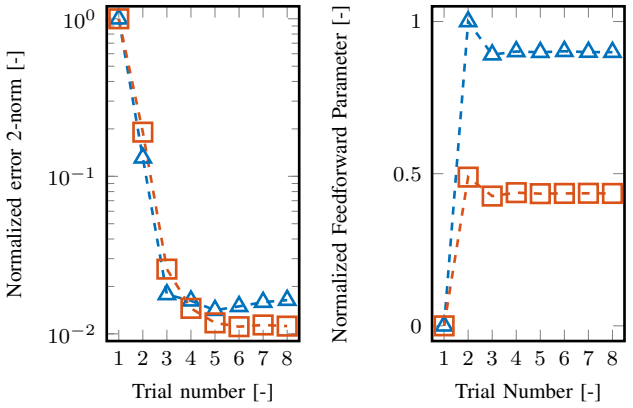

Fig. 5. Left: Normalized error 2-norm for the first 8 trials of ILCBF in the center position for y_1 (-△-) and y_2 (-□-). Right: Acceleration feedforward parameters θ_2 (-△-) and θ_4 (-□-) for the first 8 trials of ILCBF in the center position, normalized with respect to parameter θ_2.

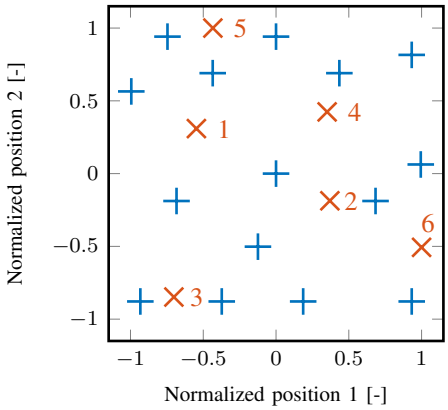

Fig. 6. ILCBF will be performed on the arbitrarily chosen training positions (+) and the learned feedforward parameters on these positions will be used as training data for the GPs. The framework will be tested and compared on the test positions (×), consisting of interpolation and extrapolation on the training data.

which are used, in combination with the basis function matrix in (13), to calculate the feedforward force f.

B. Experimental Results

The developed framework is applied to the wire bonder seen in Fig. 1. First, feedforward parameters are learned on several positions with ILCBF using 20 trials, as described in Section III-C. The error 2-norm and feedforward parameters of the first 8 trials in the center position can be seen in Fig. 5, which shows that ILCBF has converged. Second, each feedforward parameter in (14) is modeled as a separate GP, as seen in Section III-B, by using the feedforward parameters learned in trials 13 up to 20 on the training positions in Fig. 6. A squared exponential covariance function is used. For visualization purposes, the positions P_* are chosen as a fine grid covering the operating range of the machine. In Fig. 7 and Fig. 8 the GP regressions of the acceleration feedforward parameters can be seen and show a significant difference in the parameters for the operating range of the machine.

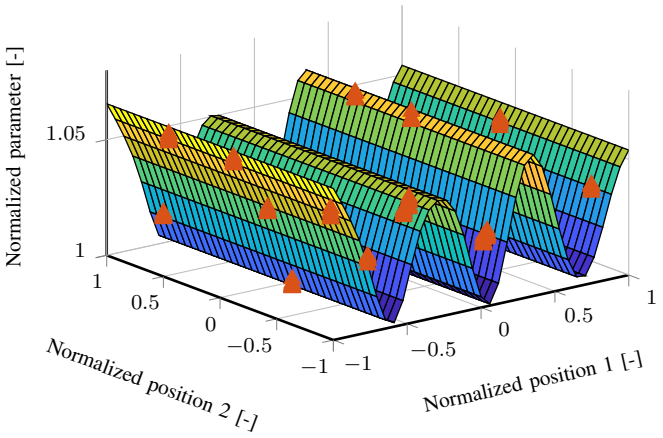

Fig. 7. Normalized GP regression of the acceleration feedforward parameter θ_2, representing the estimated mass of the axis, made using the training data (▲). The feedforward parameters are normalized with respect to the center value, i.e., $\theta_n(P_*) = \theta(P_*)/\theta(0)$. The feedforward parameter has a clear dependency on position 1.

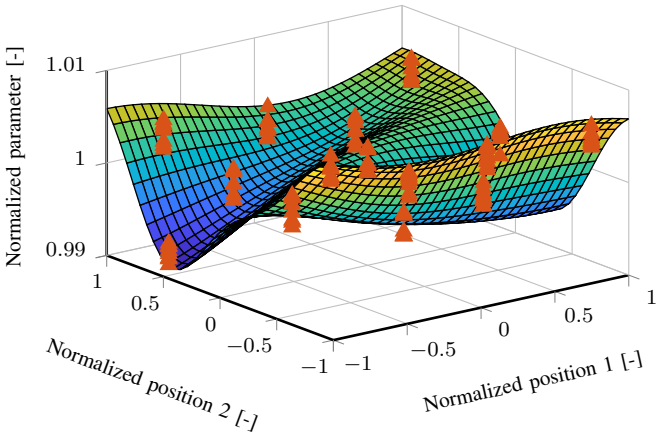

Fig. 8. Normalized GP regression of the acceleration feedforward parameter θ_4 as a function of position, representing the estimated mass of the axis, made using the training data (▲). The feedforward parameters are normalized with respect to the center value, i.e., $\theta_n(P_*) = \theta(P_*)/\theta(0)$. Position-dependency is both seen for the y_1 and the y_2 direction.

To evaluate the performance of the framework, several test positions other than the training positions have been chosen, seen in Fig. 6. The GP regressions made with the feedforward parameters learned on the training positions are used to estimate feedforward parameters on the test positions using (8). On the test positions, three methods of choosing feedforward parameters have been performed:

- **M1** center, where the feedforward parameters determined in the center position of the machine are used,
- **M2** GP, where the feedforward parameters are estimated using the GP at the given test position,
- **M3** local ILCBF, where the feedforward parameters are determined at the test position using the method described in Section III-C.

Here, local ILCBF is performed to serve as a reference frame to compare the first two methods with. The error 2-norm

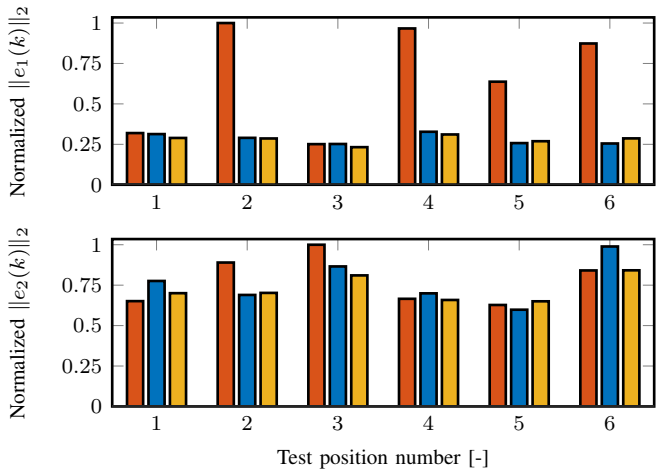

Fig. 9. Top: Error 2-norm $\|e_1(k)\|_2$ for center (■), GP (■) and local ILCBF (■) feedforward. Both GP and ILCBF achieve constant performance, while center has significantly higher error 2-norm for several positions. Bottom: Error 2-norm $\|e_2(k)\|_2$ for center (■), GP (■) and local ILCBF (■) feedforward. The performance for y_2 is similar for center, GP and ILCBF feedforward.

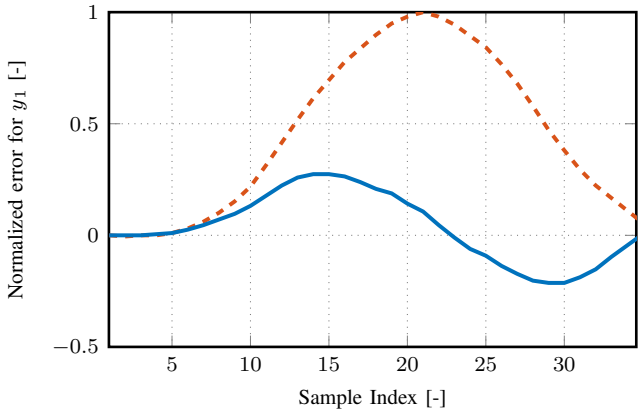

Fig. 10. Time-domain error $e_1(k)$ for test position 2, for both center (- -) and GP (—) feedforward. The maximum error for GP feedforward is roughly 3 times smaller than center feedforward.

for the test positions for all methods can be seen in Fig. 9. The error 2-norm of y_1 in Fig. 9 shows that a significant performance increase can be achieved when using feedforward parameters modeled as a function of position by a GP. This is additionally supported by looking at the time domain error of y_1 for test position 2 in Fig. 10. Furthermore, the GP feedforward achieves an error 2-norm similar to that of ILCBF, indicating optimal performance in terms of (9) with the specified basis function. The error 2-norm of y_2 shows the performance difference for the two methods and ILCBF is only marginal, showing that the y_2 direction does not have much position-dependency.

C. Discussion of Experimental Case Study

The experimental setup considered has position-dependent effects, which can be concluded by looking at the feedforward parameters in Fig. 7 and Fig. 8. For the acceleration feedforward parameter θ_2 in Fig. 7, significant position-dependent effect is seen. The unusual behavior of the parameter might be caused due to the periodic magnetic flux density in linear actuators [15]. Indeed, when measuring the distance between the peaks in Fig. 7, a distance roughly equal to the magnet pitch of the linear actuator is observed. The position-dependent effect is directly seen in the performance difference for y_1 in Fig. 9.

For test positions 1 and 3, the performance for center and GP feedforward are equal for the y_1 direction. This can directly be explained by looking at Fig. 6 and Fig. 7, that shows test positions 1 and 3 are located roughly one magnet pitch away from the center position, resulting in optimal feedforward parameters equal to the center position.

Compared with the acceleration feedforward parameter θ_2, the acceleration feedforward parameter θ_4 in Fig. 8 has considerably less position-dependency. The unusual behavior is also not observed for θ_4, indicating that the magnetic flux density is not affecting the feedforward parameter as much. When taking a look at the performance for y_2, the difference is only marginal between the feedforward methods. Therefore, it can be concluded that for the considered experimental setup, position-dependent effects in the actuator heavily outweigh position-dependent effects in the mechanics or sensing.

V. Conclusion

The developed framework models feedforward parameters as a function of position with GPs, resulting in constant performance for any arbitrary position. The training data for the GPs are determined using ILCBF, but can be extended to any feedforward parameter tuning approach. Since the position-dependent effects of motion systems are typically unknown or hard to model, non-parametric regression techniques, such as GPs, work especially well. GPs, in combination with a data-driven feedforward parameter tuning approach, results in a framework where data outweighs assumptions on the position-dependency of a motion system. Lastly, experiments on a complex motion system show that the framework can improve control performance significantly by compensating for, e.g., an unknown position-dependent magnetic flux density. Future research on this topic is directed at optimal and automatic computation of the training positions using sensor placement techniques.

Acknowledgment

The authors wish to thank Kelvin Kai Wa Yan for his support during the experimental study.

References

[1] R. Voorhoeve, R. de Rozario, W. Aangenent, and T. Oomen, "Identifying position-dependent mechanical systems: A modal approach applied to a flexible wafer stage," *IEEE Trans. Control Syst. Technol.*, vol. 29, pp. 194–206, 1 2021.

[2] J. van Zundert, J. Bolder, S. Koekebakker, and T. Oomen, "Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer," *Mechatronics*, vol. 38, pp. 76–90, 9 2016.

[3] M. Boerlage, R. Tousain, and M. Steinbuch, "Jerk derivative feedforward control for motion systems," in *Proc. American Control Conference (ACC'04)*, vol. 5. IEEE, 2004, pp. 4843–4848 vol.5.

[4] T. Oomen, "Learning for advanced motion control," in *2020 IEEE 16th International Workshop on Advanced Motion Control (AMC)*. IEEE, 9 2020, pp. 65–72.

[5] M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "LPV control for a wafer stage: beyond the theoretical solution," *Control Engineering Practice*, vol. 13, pp. 231–245, 2 2005.

[6] R. Toth, M. van de Wal, P. S. C. Heuberger, and P. M. J. V. den Hof, "LPV identification of high performance positioning devices," in *Proc. American Control Conference (ACC'11)*. IEEE, 6 2011, pp. 151–158.

[7] R. de Rozario, T. Oomen, and M. Steinbuch, "Iterative learning control and feedforward for LPV systems: Applied to a position-dependent motion system," in *Proc. American Control Conference (ACC'17)*. IEEE, 5 2017, pp. 3518–3523.

[8] S. Devasia, Degang Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. on Automatic Control*, vol. 41, no. 7, pp. 930–942, jul 1996.

[9] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi, "Direct learning of LPV controllers from data," *Automatica*, vol. 65, pp. 98–110, 2016.

[10] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT Press, 2004, vol. 3176.

[11] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, pp. 657–682, 3 2014.

[12] J. van de Wijdeven and O. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," *International Journal of Control*, vol. 83, pp. 661–675, 4 2010.

[13] S. O. R. Moheimani, D. Halim, and A. J. Fleming, *Spatial Control of Vibration - Theory and Experiments*. World Scientific Publishing Co. Pte. Ltd., 2003, vol. 10.

[14] K. van Berkel, I. Rotariu, and M. Steinbuch, "Cogging compensating piecewise iterative learning control with application to a motion system," in *Proc. American Control Conference (ACC'07)*. IEEE, 7 2007, pp. 1275–1280.

[15] J. Gieras, Z. Piech, and B. Tomczuk, *Linear Synchronous Motors*. Taylor & Francis CRC Press, 1 2012, vol. 2.

[16] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Dual-adaptive feedforward control for raster tracking with applications to AFMs," in *Proc. International Conference on Control Applications (CCA'11)*. IEEE, 9 2011, pp. 1081–1087.

[17] 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, 12th IFAC Workshop on Adaptation and Learning in Control and Signal Processing ALCOSP 2016.

[18] H.-H. Mu, Y.-F. Zhou, X. Wen, and Y.-H. Zhou, "Calibration and compensation of cogging effect in a permanent magnet linear motor," *Mechatronics*, vol. 19, pp. 577–585, 6 2009.

[19] 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," *IFAC Mechatronics*, vol. 24, pp. 944–953, 12 2014.

[20] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control—with experimental verification on a motion system," *IEEE Trans. Control Syst. Technol.*, vol. 23, pp. 722–729, 3 2015.

[21] F. Boeren, T. Oomen, and M. Steinbuch, "Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification," *Control Engineering Practice*, vol. 37, pp. 11–19, 2015.

[22] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, pp. 145–157, 2 2005.