Parameter-varying feedforward control
A kernel-based learning approach

van Haren, Max; Blanken, Lennart; Oomen, Tom

DOI
10.1016/j.mechatronics.2025.103337

Publication date
2025

Document Version
Final published version

Published in
Mechatronics

Mechatronics 109 (2025) 103337

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Parameter-varying feedforward control: A kernel-based learning approach☆,☆☆,★

Max van Haren [a],*, Lennart Blanken [b,a], Tom Oomen [a,c]

[a] Department of Mechanical Engineering, Control Systems Technology Section, Eindhoven University of Technology, Groene Loper 5, Eindhoven, 5612 AE, The Netherlands
[b] Sioux Technologies, Esp 130, Eindhoven, 5633 AA, The Netherlands
[c] Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, Delft, 2628 CN, The Netherlands

ARTICLE INFO

Keywords:
Feedforward control
Iterative learning control
Kernel regularization
Mechatronic systems
Motion control
Linear parameter-varying

ABSTRACT

The increasing demands for high accuracy in mechatronic systems necessitate the incorporation of parameter variations in feedforward control. The aim of this paper is to develop a data-driven approach for direct learning of parameter-varying feedforward control to increase tracking performance. The developed approach is based on kernel-regularized function estimation in conjunction with iterative learning to directly learn parameter-varying feedforward control from data. This approach enables high tracking performance for feedforward control of linear parameter-varying dynamics, providing flexibility to varying reference tasks. The developed framework is validated on a benchmark industrial experimental setup featuring a belt-driven carriage.

1. Introduction

Feedforward control is capable of suppressing known disturbances in motion control, specifically a reference trajectory. Feedforward control is widely applied in numerous applications, including nanopositioning [1] and robotics [2]. The reference tracking performance of feedforward control is directly determined by the accuracy of estimating the system's inverse dynamics [3]. As systems are designed progressively more complex, accurately describing their inverse dynamics becomes increasingly challenging.

Increasing complex dynamics in mechatronic systems can effectively be modeled using Linear Parameter-Varying (LPV) descriptions [4,5]. Forward LPV models can be identified through various methods [6–9]. The tracking performance of inversion-based LPV feedforward control is then directly determined by the quality of the identified forward LPV model [10–12]. The two-step approach of forward LPV identification and inversion often degrades performance due to the intricate link between tracking performance and inverse quality, and properties such as stability are not guaranteed since inversion is typically done through optimization methods.

The limitations of inversion-based feedforward control for LPV systems have led to several approaches that directly optimize the feedforward controller based on the tracking performance. In Butcher and Karimi [13], LPV feedforward controllers are directly determined based on input–output data, while de Rozario et al. [14] optimizes an LPV feedforward controller using Iterative Learning Control (ILC) [15]. Both Butcher and Karimi [13] and de Rozario et al. [14] restrict the dependence on the scheduling sequence to a generally unknown predefined structure, resulting in limited tracking performance. Furthermore, Kon et al. [16] employs a neural network to directly identify an LPV feedforward controller. However, its practical applicability remains limited due to the added complexity of estimating the zero dynamics of LPV systems and since the neural network is not directly capable of utilizing physical insights, which can improve estimation quality. Finally, direct data-driven control approaches for LPV or linear time-varying systems, such as [17–19], generally do not allow for the incorporation of physical insights, and their computational complexity limits their practical applicability. Overall, the applicability of current approaches for direct optimization of LPV feedforward controllers based on tracking performance is limited, as the structure of the

☆ 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).

☆☆ This research has received funding from the ECSEL Joint Undertaking under grant agreement 101007311 (IMOCO4.E). The Joint Undertaking receives support from the European Union Horizon 2020 research and innovation programme.

★ This paper was recommended for publication by Associate Editor Peter Hehenberger.

* Corresponding author.
E-mail address: m.j.v.haren@tue.nl (M. van Haren).

https://doi.org/10.1016/j.mechatronics.2025.103337
Received 8 November 2024; Received in revised form 10 April 2025; Accepted 23 April 2025
Available online 9 May 2025

dependency on the scheduling sequence must be known in advance, and no physical insights can be utilized.

Although several techniques have been developed for LPV feedforward control, there is currently no method that directly optimizes the tracking performance without constraining the dependence on the scheduling sequence, while also allowing for the incorporation of physical insights. In this paper, feedforward parameter functions are identified through kernel-regularized function estimation [20], and the estimates are refined by iteratively minimizing the tracking error. Kernel methods have been successfully applied in control, including system identification [20–22], learning control [23,24], and feedforward control [25,26]. The main advantage of kernel-regularized function estimation is that it does not restrict the estimated function to a specific structure, but is searched over a possibly infinite-dimensional functional space that admits a finite-dimensional representation with a closed-form solution [20]. Furthermore, kernel-regularized estimation makes it particularly easy to incorporate physical insights of the system by enforcing high-level properties of the estimated function, such as periodicity. Additionally, the iterative nature enhances estimation quality, improving tracking performance. The following contributions are distinguished.

(C1) Iteratively learning the feedforward parameter functions, which enhances estimation quality and improves tracking performance.
(C2) Identifying feedforward parameter functions through kernel-regularized estimation, allowing any dependence on the scheduling sequence to be modeled.
(C3) Experimental characterization and validation of the developed framework for a low-cost belt-driven carriage, which exhibits position-dependent behavior.

This work extends earlier research presented in [27,28] by generalizing these studies to iteratively learn continuous parameter variations and apply a generic parameterization in the feedforward controller, including experimental validation.

Notation. The discrete-time index is denoted by $k \in \mathbb{Z}_{[0,N-1]}$. The amount of samples in a measurement period is equal to N. Scalars and row and column vectors are denoted by lowercase letters, e.g., x. Matrices are denoted by capitals, e.g., X. Functions and time-dependent signals are denoted explicitly, e.g., $x(k)$. Time-dependent signals are vectorized as

$$x = \begin{bmatrix} x^\top(0) & x^\top(1) & \dots & x^\top(N-1) \end{bmatrix}^\top . \tag{1}$$

Systems are denoted calligraphically, e.g., $\mathcal{H}$. LPV systems are described using the discrete-time state-space

$$\mathcal{H}(\rho): \begin{cases} x(k+1) &= A(\rho(k))x(k) + B(\rho(k))u(k), \\ y(k) &= C(\rho(k))x(k) + D(\rho(k))u(k), \end{cases} \tag{2}$$

with scheduling sequence $\rho(k)$. The response of system $\mathcal{H}(\rho)$ to input $u(k)$ is denoted with $y(k) = \mathcal{H}(\rho)u(k)$. Let $x(k) = 0$ for $k = 0$, and $u(k) = 0$ for $k < 0$ and $k \geq N$, to obtain the finite-time LPV convolution

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N\text{-}1) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} D(\rho(0)) & 0 & \cdots & 0 \\ C(\rho(1))\,B(\rho(0)) & D(\rho(1)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C(\rho(N\text{-}1))\prod_{k=1}^{N\text{-}2} A(\rho(k))B(\rho(0)) & \cdots & \cdots & D(\rho(N\text{-}1)) \end{bmatrix}}_{H} \times \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N\text{-}1) \end{bmatrix}}_{u}, \tag{3}$$

with LPV convolution matrix H. Linear Time-Invariant (LTI) systems are described using the forward shift operator $qu(k) = u(k+1)$.

2. Problem formulation

In this section, a motivating application and the problem setting are shown for LPV feedforward control. Finally, the problem addressed in this paper is defined.

2.1. Motivating application

The problem addressed in this paper is directly motivated by the belt-driven carriage in Fig. 1, which represents a transmission used regularly for mechatronic systems. Specifically, the belt-driven carriage in Fig. 1 exhibits position-dependent dynamics, which are commonly observed in mechatronic systems and can be accurately modeled using LPV system descriptions. The objective of the setup is to accurately position the carriage in the y-direction by using the BrushLess AC (BLAC) motor. Belt-driven motion systems are typically used in applications such as printing, where the accuracy during constant velocity motion is important. The BLAC motor is connected to the carriage via a toothed pulley and a timing belt. The belt-driven carriage suffers from position-dependent dynamics due to several effects, including pulley out-of-roundness, pulley-teeth interactions, motor cogging, misalignment of the linear guide, and position-dependent drivetrain stiffness [29, Section 4.2], which is shown in Example 1.

Example 1. By modeling the timing belt in Fig. 1 as an elastic element under pretension with Young's modulus E and cross-sectional area A, the perceived stiffness at the driven pulley is

$$k(\rho) = \frac{EA}{\frac{1}{2}L + \rho R} + \frac{EA}{1\frac{1}{2}L - \rho R}, \tag{4}$$

for scheduling ρ being the angular rotation of the driven pulley, resulting in quasi-LPV behavior. The system is modeled as a mass–spring system with continuous-time inverse dynamics [28]

$$\begin{aligned} u(t) &= \left(mR + \frac{J}{R}\right)\frac{d^2y(t)}{dt^2} + \frac{mJ}{R}\frac{d^2}{dt^2}\left(\frac{1}{k(\rho(t))}\frac{d^2y(t)}{dt^2}\right) \\ &= \sum_{i\in\{2,3,4\}} \frac{d^i y(t)}{dt^i}\theta_i(\rho(t)), \end{aligned} \tag{5}$$

consisting of derivatives of the desired output $\frac{d^i y(t)}{dt^i}$ and LPV parameter functions $\theta_i(\rho(t))$.

LPV descriptions of complex mechatronic systems, including belt-driven carriages such as the system in Fig. 1, directly motivate the development of parameter-varying feedforward control.

2.2. Problem setting

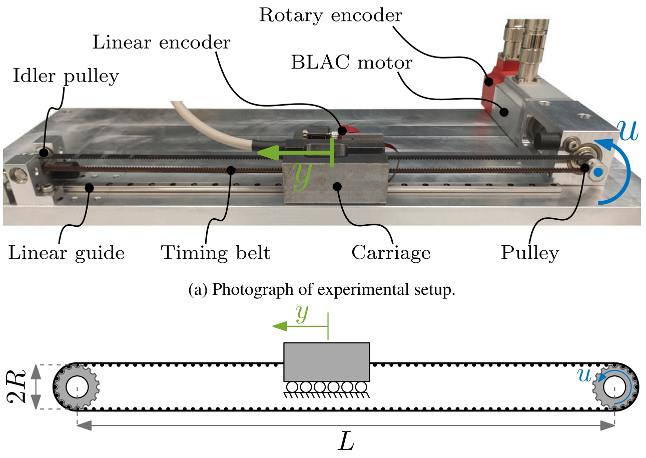
The control structure is seen in Fig. 2. The considered class of LPV systems $\mathcal{G}$ is described using the convolution in (3). The system is operating in feedback with LTI controller $\mathcal{C}$, as seen in Fig. 2. The tracking error, assuming zero initial state and error, is given by the interconnection of the system $\mathcal{G}$ and controller $\mathcal{C}$ by

$$e(k) = \mathcal{S}(\rho)(r(k) - v(k)) - \mathcal{J}(\rho)f(k), \tag{6}$$

with LPV sensitivity $\mathcal{S}(\rho)$ and process sensitivity $\mathcal{J}(\rho)$. LTI feedforward is not capable of compensating for the LPV dynamics in (6), resulting in residual error. Motivated by the inverse dynamics of LPV systems, for example (5) in Example 1, the goal in this paper is to decrease the tracking error using the parameter-varying feedforward signal

$$f(k) = \psi(r(k))\theta(\rho(k)), \tag{7}$$

with basis functions $\psi(r(k)) \in \mathbb{R}^{1\times n_\theta}$ and feedforward parameter functions $\theta(\rho(k)) \in \mathbb{R}^{n_\theta \times 1}$. For motion systems, the basis functions $\psi(r(k))$ are often chosen as derivatives of the reference signal, such as the reference velocity and acceleration [30,31]. The feedforward parameterization (7) is flexible to task variations due to the dependency on $r(k)$ in the basis function $\psi(r(k))$.



(a) Photograph of experimental setup.

(b) Sketch of experimental setup.

Fig. 1. Experimental setup considered, where the position of the belt-driven carriage y is to be controlled using the input to the motor u.

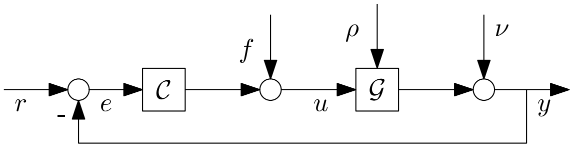


Fig. 2. Control structure considered.

2.3. Problem definition

The problem considered in this paper is as follows. Given a reference signal $r(k)$, corresponding choice of basis functions $\psi\left(r(k)\right) \in \mathbb{R}^{1\times n_\theta}$, a model of the system $\hat{\mathcal{G}}$, and the measured scheduling sequence $\rho(k)$, determine the n_θ feedforward parameter functions $\theta(\rho(k)) \in \mathbb{R}^{n_\theta \times 1}$, such that the tracking error (6) is minimized using the feedforward parameterization (7).

3. Learning feedforward parameter functions

In this section, feedforward parameter functions are iteratively learned through kernel-regularized estimation. The non-parametric nature of kernel-regularized estimation enables modeling any dependence on the scheduling sequence, and iterative learning improves estimation quality, both contributing to C1 and C2. Furthermore, the design of kernels and the developed procedure are presented.

3.1. Iterative learning of feedforward parameter functions

The key idea is to iteratively learn the feedforward parameter functions $\theta(\rho(k))$ in (7) such that the tracking error is minimized. The error (6) is approximated in the next trial $j+1$ for feedforward signal (7) based on a model of the process sensitivity $\hat{\mathcal{J}}$ as

$$\hat{e}_{j+1}(k) = e_j(k) - \hat{\mathcal{J}}\left(\rho_j\right)\psi\left(r_j(k)\right)\left(\theta_{j+1}\left(\rho_j(k)\right) - \theta_j\left(\rho_j(k)\right)\right). \tag{8}$$

The parameters are then iteratively learned by minimizing a certain cost function with respect to the approximated error, that is

$$\min_{\theta_{j+1}} V\left(\hat{e}_{j+1}\right). \tag{9}$$

As a result, the feedforward parameters are iteratively updated over iterations or trials j as

$$\theta_{j+1}(\rho^*) = \theta_j(\rho^*) + \theta_j^\Delta(\rho^*), \tag{10}$$

for arbitrary scheduling value ρ^*.

Remark 2. By choosing $V\left(\hat{e}_{j+1}\right) = \sum_{k=0}^{N-1}\hat{e}_{j+1}^2(k)$ and optimizing for LTI feedforward parameters $\theta_j(\rho(k)) \equiv \bar{\theta}_j \;\; \forall\rho(k)$, LTI ILC with basis functions [32] is recovered.

The iterative procedure is effective since it directly minimizes the least-squares tracking error, and uses both a model of the process sensitivity and the measured data, thus reducing the requirements on the modeling quality [33]. A key challenge is to learn $\theta_j^\Delta(\rho^*)$ in (10) without constraining the estimated function to a certain structure, which is presented in the next section.

3.2. Kernel-regularized learning of feedforward parameter functions

Unlike traditional estimation methods that restrict the estimated function $\theta_j^\Delta(\rho^*)$ in (10) to a specific structure, the function $\theta_j^\Delta(\rho^*)$ is estimated in a possibly infinite-dimensional function space $\mathscr{H}$ and can be evaluated at any arbitrary $\rho^* \in \mathbb{R}$. Specifically, the feedforward parameter functions are estimated by iteratively minimizing the predicted least-squares tracking error (8) with a regularization term J to prevent overfitting and ill-posedness, i.e.,

$$\min_{\theta_j^\Delta \in \mathscr{H}} \sum_{k=0}^{N-1} \hat{e}_{j+1}^2(k) + \gamma J\left(\theta_j^\Delta\right). \tag{11}$$

The regularizer J can be chosen to penalize unwanted behavior of the estimated functions θ_j^Δ. For example, the energy of the estimated functions θ_j^Δ can be reduced by penalizing with $J(\theta_j^\Delta) = \int\left(\theta_j^\Delta(\rho)\right)^\top \theta_j^\Delta(\rho)\, d\rho$.

Properties of the estimated function are effectively enforced through designing $\mathscr{H}$ as a Hilbert space, and choosing the regularizer as the squared norm in this space,

$$\min_{\theta_j^\Delta \in \mathscr{H}} \sum_{k=0}^{N-1} \hat{e}_{j+1}^2(k) + \gamma\|\theta_j^\Delta\|_{\mathscr{H}}^2, \tag{12}$$

with Reproducing Kernel Hilbert Space (RKHS) norm $\|\theta_j^\Delta\|_{\mathscr{H}}^2 = \left\langle \theta_j^\Delta, \theta_j^\Delta \right\rangle_{\mathscr{H}}$ [20]. The RKHS $\mathscr{H}$ is associated with a kernel function that is capable of reproducing every function in the space, in this case the $n_\theta \times n_\theta$ kernel function matrix

$$K\left(\rho^*, \rho\right) = \begin{bmatrix} k_{11}\left(\rho^*, \rho\right) & k_{12}\left(\rho^*, \rho\right) & \cdots & k_{1n_\theta}\left(\rho^*, \rho\right) \\ k_{21}\left(\rho^*, \rho\right) & k_{22}\left(\rho^*, \rho\right) & \cdots & k_{2n_\theta}\left(\rho^*, \rho\right) \\ \vdots & \vdots & \ddots & \vdots \\ k_{n_\theta 1}\left(\rho^*, \rho\right) & k_{n_\theta 2}\left(\rho^*, \rho\right) & \cdots & k_{n_\theta n_\theta}\left(\rho^*, \rho\right) \end{bmatrix}, \tag{13}$$

where each kernel function k_{ij} describes the correlation between feedforward parameters i and j. As a result, the kernel functions enable to enforce desired properties of the estimated function $\theta_j^\Delta(\rho^*)$, such as smoothness or periodicity.

Remark 3. Cross-correlation between different feedforward parameters, which is commonly observed for motion systems [28], is directly enabled by setting $k_{ij} \neq 0 \; \forall i \neq j$.

Since the kernel-regularized estimates of the feedforward parameter functions are estimated in the possibly infinite dimensional space $\mathscr{H}$, the function estimates are not restricted to a specific class of functions and are thus capable of modeling any feedforward parameter function.

Although the feedforward parameter functions are modeled in a possibly infinite dimensional space $\mathscr{H}$, it admits a finite-dimensional solution through the representer theorem [20, (63)]

$$\theta_j^\Delta\left(\rho^*\right) = \sum_{k=0}^{N-1} K\left(\rho^*, \rho_j(k)\right)\psi^\top\left(r_j(k)\right)\hat{\mathcal{J}}\left(\rho_j\right)\hat{c}_j(k), \tag{14}$$

where kernel function matrix $K(\rho^*, \rho_j(k)) \in \mathbb{R}^{n_\theta \times n_\theta}$ is determined with (13). The (modified) representers $\hat{c}_j = \begin{bmatrix}\hat{c}_j(0) & \hat{c}_j(1) & \cdots & \hat{c}_j(N-1)\end{bmatrix}^\top \in \mathbb{R}^N$ are given by [20, (64) and (70b)]

$$\hat{c}_j = \left(\hat{J}_j \Psi_j K_j \Psi_j^\top \hat{J}_j^\top + \gamma I_N\right)^{-1} e_j, \tag{15}$$

with convolution matrix $\hat{\mathcal{J}}_j \in \mathbb{R}^{N\times N}$ of LPV system $\hat{\mathcal{J}}$ evaluated at the measured scheduling sequence ρ_j using (3), and the kernel matrix $K_j \in \mathbb{R}^{Nn_\theta \times Nn_\theta}$ is evaluated by

$$K_j = \begin{bmatrix} K\left(\rho_j(0), \rho_j(0)\right) & K\left(\rho_j(0), \rho_j(1)\right) & \cdots & K\left(\rho_j(0), \rho_j(N\text{-}1)\right) \\ K\left(\rho_j(1), \rho_j(0)\right) & K\left(\rho_j(1), \rho_j(1)\right) & \cdots & K\left(\rho_j(1), \rho_j(N\text{-}1)\right) \\ \vdots & \vdots & \ddots & \vdots \\ K\left(\rho_j(N\text{-}1), \rho_j(0)\right) & K\left(\rho_j(N\text{-}1), \rho_j(1)\right) & \cdots & K\left(\rho_j(N\text{-}1), \rho_j(N\text{-}1)\right) \end{bmatrix}. \tag{16}$$

Note that the feedforward parameters for the next trial in (14) are calculated based on the scheduling sequence measured in the current trial. The basis function matrix $\Psi_j \in \mathbb{R}^{N\times Nn_\theta}$ is constructed such that $f_j = \Psi_j \theta_j$ as

$$\Psi_j = \begin{bmatrix} \psi\left(r_j(0)\right) & 0 & \cdots & 0 \\ 0 & \psi\left(r_j(1)\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \psi\left(r_j(N-1)\right) \end{bmatrix}. \tag{17}$$

The feedforward parameters for the next trial (10) are estimated by propagating the feedforward update using the representer theorem (14) over the trials, resulting in

$$\theta_{j+1}\left(\rho^*\right) = \sum_{i=0}^{j} \sum_{k=0}^{N-1} K\left(\rho^*, \rho_i(k)\right) \psi^\top\left(r_i(k)\right) \hat{\mathcal{J}}\left(\rho_i\right) \hat{c}_i(k). \tag{18}$$

Kernel-regularized learning of LPV feedforward parameter functions in (18) estimates the functions without restricting the dependence on the scheduling sequence, while allowing for the incorporation of prior knowledge.

Remark 4. For the special case where the scheduling sequence during iterative learning is constant, i.e., $\rho_j(k) = \rho(k)\ \forall j$, the parameter update (18) simplifies to

$$\theta_{j+1}\left(\rho^*\right) = \theta_j\left(\rho^*\right) + \sum_{k=0}^{N-1} K\left(\rho^*, \rho(k)\right) \psi^\top\left(r_j(k)\right) \hat{\mathcal{J}}(\rho)\, \hat{c}_j(k). \tag{19}$$

Remark 5. The convergence of learning feedforward parameter functions (18) is primarily influenced by

1. the quality of model $\hat{\mathcal{J}}$;
2. the choice of kernel functions in (13); and
3. the regularization coefficient γ in (12) and (15).

Generally, the regularization parameter γ can be increased to ensure convergence of the framework. For a trial-invariant basis function $\Psi_j = \Psi\ \forall j$ and scheduling sequence $\rho_j = \rho\ \forall j$, which leads to $J_j = J$, $K_j = K\ \forall j$, and no modeling uncertainty $\hat{\mathcal{J}} = J$, the propagation of feedforward parameters (18) and (19) is written in vector notation $\theta = \begin{bmatrix} \theta(\rho(0)) & \theta(\rho(1)) & \cdots & \theta(\rho(N-1)) \end{bmatrix}^\top$ as

$$\begin{aligned} \theta_{j+1} &= \theta_j + K\Psi^\top J^\top \left(J\Psi K\Psi^\top J^\top + \gamma I_N\right)^{-1} e_j \\ &= \gamma\left(K\Psi^\top J^\top J\Psi + \gamma I_{Nn_\theta}\right)^{-1} \theta_j + (\ldots) S\left(r_j - v_j\right), \end{aligned} \tag{20}$$

with e_j from (6). Monotonic convergence of the feedforward parameters θ_{j+1} at the trial-invariant scheduling ρ (20) is guaranteed if $\bar{\sigma}\left(\gamma\left(K\Psi^\top J^\top J\Psi + \gamma I_{Nn_\theta}\right)^{-1}\right) < 1$ [32], with $\bar{\sigma}$ the largest singular value. Furthermore, ILC with basis functions and its associated convergence properties [32] are recovered by $K(\rho^*, \rho) = I$ and $\gamma = 0$.

Remark 6. The robustness against modeling uncertainties of ILC [32] allows the use of LTI model $\hat{\mathcal{J}}(q)$ instead of the LPV model $\hat{\mathcal{J}}\left(\rho_j\right)$.

3.3. Design of Kernel functions

The developed update law requires the design of kernel functions $k_{ij}(\rho^*, \rho)$ in matrix $K(\rho^*, \rho)$ in (13). Many different kernel functions are possible, and the suitable choice depends on the problem at hand. Three examples of high-level properties that can be enforced on the feedforward parameters through the use of kernel functions are the following.

1. Constant feedforward parameter functions are realized by the constant kernel

$$k_{ij}^{c}\left(\rho^*, \rho\right) = \sigma^2, \tag{21}$$

with hyperparameter σ^2 determining the average distance of the function to its mean.

Remark 7. The constant kernel recovers LTI ILC with basis functions [32], with parameters estimated using Tikhonov regularization for $\gamma > 0$ and without regularization for $\gamma = 0$.

2. Smooth feedforward parameter functions are estimated using the squared-exponential kernel

$$k_{ij}^{SE}\left(\rho^*, \rho\right) = \sigma^2 \exp\left(-\frac{\left(\rho^* - \rho\right)^2}{2\ell^2}\right), \tag{22}$$

where σ^2 has the same function as for the kernel (21), and the hyperparameter ℓ determines the level of smoothness of the estimated function.

3. Periodic feedforward parameter functions are realized through the periodic kernel

$$k_{ij}^{per}\left(\rho^*, \rho\right) = \sigma^2 \exp\left(-\frac{2\sin^2\left(\pi\left|\rho^* - \rho\right|/p\right)}{\ell^2}\right), \tag{23}$$

where σ^2 and ℓ have the same role as in the squared-exponential kernel (22), and the hyperparameter p forces the feedforward parameter functions to be periodic with period p.

In addition, multiple kernels can be combined such that they have the properties of various kernels, and can be trivially extended for multidimensional inputs [34]. The choice of kernel functions is determined by the situation at hand, where several guidelines are given in Pillonetto et al. [20] and Rasmussen [34, Chapter 4]. Considering the timing-belt system in Fig. 1, periodicity along the pulley circumference can be embedded in the feedforward parameter functions through the use of a periodic kernel.

Remark 8. The LPV feedforward signal should in some cases be dynamically dependent, meaning that it should be dependent on derivatives or time-shifted values of ρ [16,28]. The kernel functions can be straightforwardly extended by using these values as additional input to the kernel.

3.4. Procedure

The developed procedure that iteratively learns feedforward parameter functions through kernel-regularized estimation is summarized in Procedure 9.

Procedure 9 (*Iterative kernel-regularized learning of LPV feedforward parameters*).

Inputs: *Model $\hat{\mathcal{J}}$, reference signal r_j, choice of basis functions in ψ, which (measured) signals are the scheduling sequence ρ_j, kernel function matrix $K(\rho_j^*, \rho_j)$ from (13) and corresponding hyperparameters.*

1. *Initialize $\theta_0\left(\rho^*\right)$, e.g., $\theta_0\left(\rho^*\right) = 0\ \forall \rho^*$.*
2. *For $j \in \mathbb{Z}_{[0, N_{trial}-1]}$*

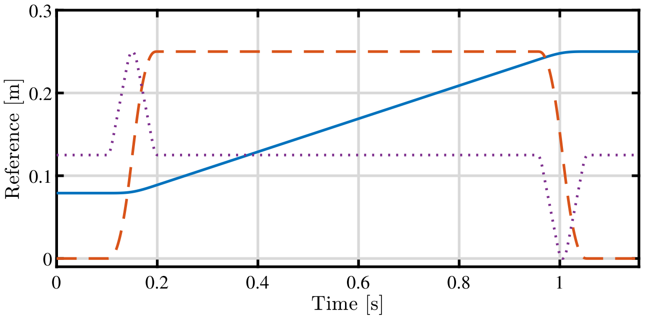


Fig. 3. Reference r (—), scaled reference velocity $\dot{r}$ (- -), and scaled reference acceleration (···) applied to the experimental setup.

(a) *Compute basis functions $\psi(r_j(k))$ in (7) using the reference $r_j(k)$, and construct basis function matrix Ψ_j using (17).*
(b) *Compute $f_j(k)$ with (7).*
(c) *Apply $f_j(k)$ and $r_j(k)$ to the system in Fig. 2, and measure error $e_j(k)$ and scheduling sequence $\rho_j(k)$.*
(d) *Construct convolution matrix $\hat{\mathcal{J}}_j$ using (3) with model $\hat{\mathcal{J}}$ and measured scheduling sequence $\rho_j(k)$.*

- *If no LPV model is available, set $\hat{\mathcal{J}}(\rho_j)$ to an LTI approximate, i.e., $\hat{\mathcal{J}}(q)$.*

(e) *Determine kernel matrix K_j based on measured scheduling sequence ρ_j using (16).*
(f) *Calculate the representers $\hat{c}_j$ for trial j using (15).*
(g) *Compute the feedforward parameters for the next trial $j+1$ based on the current measured scheduling sequence ρ_j, meaning $\theta_{j+1}(\rho_j(k))$, using (18).*

4. Experimental setup characterization

In this section, the experimental setup is introduced and characterized, leading to an appropriate parameter-varying feedforward parameterization, hence contributing to C3. Specifically, several preliminary experiments are performed to determine which feedforward structure and kernels will be used.

4.1. Experimental setup

The experimental setup considered is a 1 degree-of-freedom belt-driven carriage, which is mounted on a linear guide, as shown in Section 2.1 and Fig. 1. The BLAC motor is equipped with a rotary encoder having a resolution of 8192 counts per revolution, in addition to a linear encoder on the linear guide with a resolution of 100 nm. The timing belt has a tooth pitch of 2 mm, which is made of rubber with an added carbon fiber core for additional stiffness. The pulleys have 15 teeth per revolution, and hence, a circumference of 0.03 m. The distance between the pulleys is $L = 0.3$ m. The goal is to track a third-order scanning motion with the carriage using the input to the motor u, consisting of a large constant velocity part, which is seen in Fig. 3. The performance is evaluated during constant velocity, since this type of drivetrain is typically used in scanning applications such as printing systems.

The control scheme in Fig. 2 is used, where the LTI feedback controller is a discrete-time lead filter with an additional low-pass filter described by the transfer function

$$C(q) = \frac{2169q^{-1} - 381q^{-2} - 1747q^{-3}}{1 - 2.421q^{-1} + 1.961q^{-2} - 0.5337q^{-3}}. \tag{24}$$

The settings during experimentation are shown in Table 1.

Table 1
Experimental settings.

Variable	Abbreviation	Value	Unit
Sampling time	T_s	$2.5 \cdot 10^{-4}$	s
Number of samples	N	4630	–
Reference stroke	–	0.171	m
Maximum velocity	–	0.2	m/s
Maximum acceleration	–	4	m/s^2

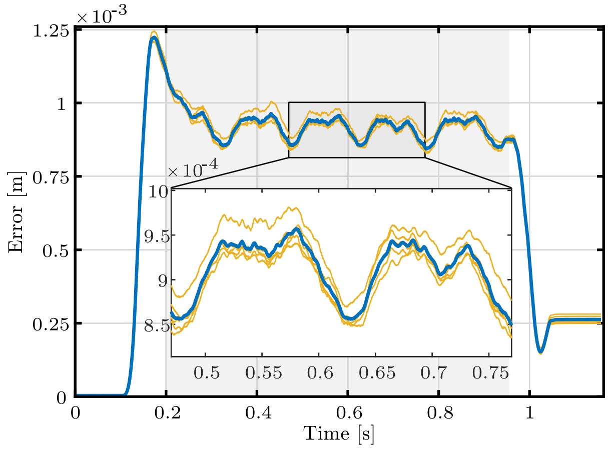


Fig. 4. During constant velocity (▨), the experimental tracking error for 5 repetitions of the reference in Fig. 3 with zero feedforward $f_j = 0$ (—) and their sample mean (—) show highly repeatable position-dependent effects.

4.2. Characterization of position-dependent dynamics

The tracking error with zero feedforward $f_j = 0$ for 5 repetitions of the reference in Fig. 3 is shown in Fig. 4. The following is observed from the tracking error in Fig. 4.

- The tracking error is highly repeatable, and hence, an iterative approach is suitable.
- The large offset in the tracking error indicates that both Coulomb and viscous friction feedforward might be necessary.
- During acceleration, the error reaches its maximum, motivating the need for acceleration feedforward.
- The tracking error has a period of 0.15 s during a constant velocity of 0.2 m/s, resulting in a spatial period of 0.03 m, which is the circumference of the pulley.

The spatial periodic effect is further analyzed with a power spectrum of the tracking error during constant velocity as a function of the number of pulley revolutions in Fig. 5. The spatial power spectrum shows that during constant velocity, the error is dominated by the zero frequency, the fundamental frequency [1/rev], its second harmonic [2/rev] and marginally by its fourth harmonic [4/rev]. The zero frequency contribution is mainly caused by the lack of Coulomb friction feedforward, which is also seen from the constant offset of the tracking error in Fig. 4. The fundamental frequency, and second and fourth harmonic are most likely caused due position-dependent effects introduced by pulley out-of-roundness or motor cogging.

5. Experimental application

In this section, both LTI feedforward control and the developed LPV feedforward control methods are compared on the experimental setup, further contributing to C3. Both the learning procedure and the tracking results are shown.

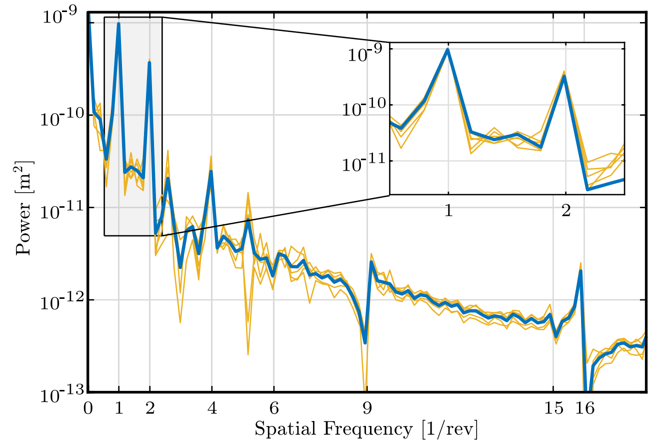


Fig. 5. Power spectrum of the tracking error during constant velocity for 5 times tracking the reference in Fig. 3 with zero feedforward $f_j = 0$ (—) and their sample mean (—) shows highly repetitive behavior in spatial domain.

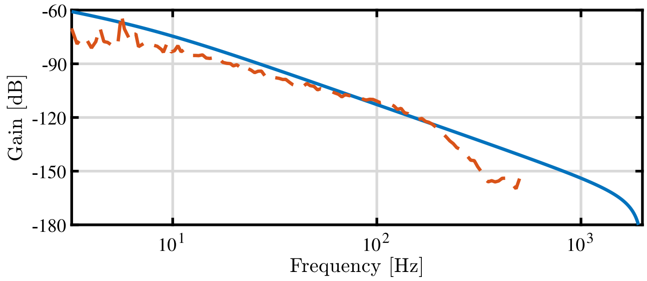


Fig. 6. The LTI model $\hat{\mathcal{G}}(q)$ (—) is constructed to approximate the measured frequency response function (- -) of the experimental belt-driven carriage.

5.1. Experimental learning settings

The model of the process sensitivity $\hat{\mathcal{J}}$ is determined by using a simplified LTI model of the system $\mathcal{G}$ in feedback with the controller C in (24). The simplified LTI model $\hat{\mathcal{G}}$ is determined by using the measured frequency response function in Fig. 6. As a result, the simplified LTI model $\hat{\mathcal{G}}$ consists of a mass with a damper attached to the fixed world, and is discretized with a zero order hold, resulting in

$$\hat{\mathcal{G}}(q) = \frac{2.82q^{-1} + 2.798q^{-2}}{1 - 1.978q^{-1} + 0.9775q^{-2}} \cdot 10^{-8}. \tag{25}$$

After interconnecting $\hat{\mathcal{G}}$ with the controller C from (24), the LTI model of the process sensitivity is

$$\hat{\mathcal{J}}(q) = \frac{2.82q^{-1} - 4.029q^{-2} - 1.244q^{-3} + 3.982q^{-4} - 1.493q^{-5}}{1 - 4.399q^{-1} + 7.727q^{-2} - 6.779q^{-3} + 2.972q^{-4} - 0.522q^{-5}} \cdot 10^{-8}. \tag{26}$$

The reference tracking task is constant during learning $r_j = r\ \forall j$ and the same as in Section 4.1, shown in Fig. 3. The reference signal is used as scheduling sequence $\rho_j = r\ \forall j$, since this is relatively close to the position of the carriage y, and it is known in advance of the tracking experiment.

The observations in Section 4.2 motivate using viscous friction, Coulomb friction, and acceleration feedforward in (7), which is widely applied in feedforward control [35], i.e.,

$$f_j(k) = \theta_j^a(\rho(k))\,\ddot{r}(k) + \theta_j^v(\rho(k))\,\dot{r}(k) + \theta_j^c(\rho(k))\,\mathrm{sign}\,(\dot{r}(k)), \tag{27}$$

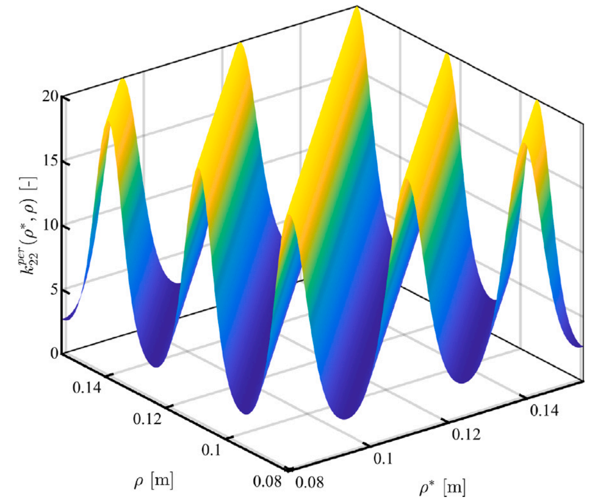


Fig. 7. Surface plot of kernel function $k_{22}^{per}(\rho^*, \rho)$ evaluated at a scheduling sequence $\rho^*, \rho \in \mathbb{R}_{[0.08, 0.16]}$ (▨), which enforces a smooth and periodic function having a period of $p = 0.03$ m.

with discrete-time derivatives $\ddot{r}(k)$ and $\dot{r}(k)$. The corresponding basis functions and feedforward parameters from (7) are

$$\begin{aligned} \psi(r(k)) &= \left[\left(\frac{q - q^{-1}}{2T_s}\right)^2 r(k) \quad \frac{q - q^{-1}}{2T_s} r(k) \quad \mathrm{sign}\left(\frac{q - q^{-1}}{2T_s} r(k)\right) \right], \\ \theta_j(\rho(k)) &= \left[\theta_j^a(\rho(k)) \quad \theta_j^v(\rho(k)) \quad \theta_j^c(\rho(k))\right]^\top, \end{aligned} \tag{28}$$

where discrete-time derivatives $\ddot{r}(k)$ and $\dot{r}(k)$ from (27) are computed using the central difference. The 3×3 kernel function matrix (13) is defined such that the estimated feedforward parameters $\theta_j(\rho(k))$ do not correlate, and the acceleration and Coulomb friction feedforward parameter functions are enforced to be constant using the kernel (21), i.e.,

$$K(\rho^*, \rho) = \begin{bmatrix} k^c(\rho^*, \rho) & 0 & 0 \\ 0 & k_{22}(\rho^*, \rho) & 0 \\ 0 & 0 & k^c(\rho^*, \rho) \end{bmatrix}. \tag{29}$$

The hyperparameters of the constant kernels for the acceleration and coulomb friction feedforward parameter functions are respectively $\sigma^2 = 1$ and $\sigma^2 = 3$. The amount of trials is set to 10 and the regularization coefficient γ from (12) to $\gamma = 5 \cdot 10^{-5}$.

LTI feedforward control and the developed LPV feedforward control are compared by using a constant or varying viscous friction feedforward parameter function. Specifically, the compared methods use the following kernel functions for the viscous friction feedforward parameter.

- **LTI feedforward**: constant kernel from (21), $k_{22}(\rho^*, \rho) = k^c(\rho^*, \rho)$, with hyperparameter $\sigma^2 = 20$.
- **LPV feedforward**: periodic kernel from (23), $k_{22}(\rho^*, \rho) = k_{22}^{per}(\rho^*, \rho)$, with hyperparameters $\sigma^2 = 20$, $\ell = 1$ m and $p = 3 \cdot 10^{-2}$ m.

Note that the LTI feedforward approach recovers ILC with basis functions, where the feedforward parameters are estimated using Tikhonov regularization, see Remark 7. The spatial periodic effect of the tracking error in Section 4.2 motivates to select k_{22} being periodic in the pulley circumference. A surface plot of the kernel function k_{22}^{per} is shown in Fig. 7. LTI feedforward control recovers LTI ILC with basis functions [33], where Tikhonov regularization is used to estimate the feedforward parameters.

5.2. Experimental results

The learned viscous, Coulomb, and acceleration feedforward parameter functions are shown in Figs. 8 and 9, which demonstrate that

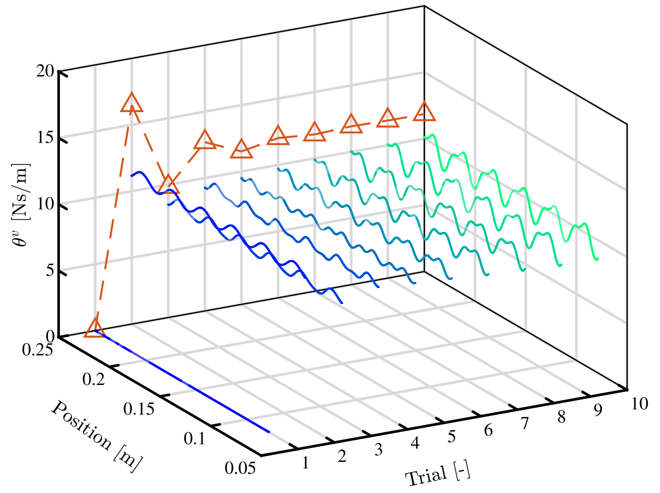


Fig. 8. Feedforward parameter function $\theta_j^v(\rho(k))$ over the trials estimated by the developed LPV feedforward method (▮) and LTI feedforward method θ_j^v (△).

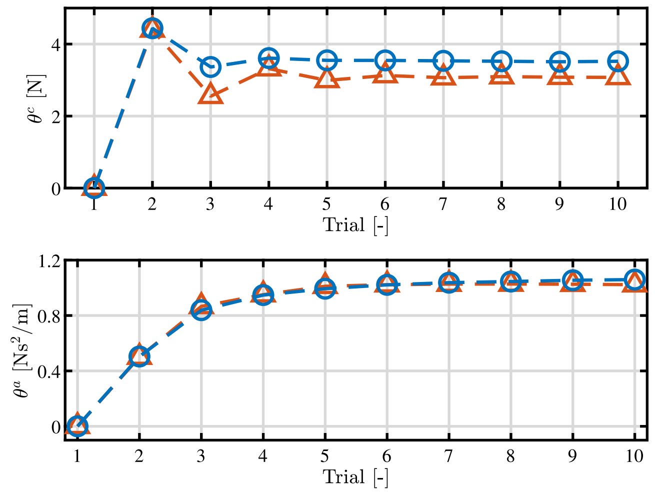


Fig. 9. Constant feedforward parameters θ_j^c and θ_j^a estimated by LTI feedforward control (△) and LPV feedforward control (○).

the learning has converged. The viscous friction feedforward parameter function clearly shows the periodic behavior enforced by the periodic kernel. The Coulomb friction and acceleration feedforward parameters do not show any significant difference between the LTI and LPV feedforward methods.

The converged feedforward signal in the final trial from (27) using the feedforward parameters from Figs. 8 and 9 for the reference in Fig. 3 is shown in Fig. 10.

The converged reference tracking performance is illustrated through the Root-Mean-Squared (RMS) tracking errors in Fig. 11, the tracking error in the final trial in Fig. 12, and the power spectrum of the final tracking error in Fig. 13. In addition, several error metrics that are particularly relevant for systems performing scanning motions, including the RMS and maximum value of the error during constant velocity e_j^v, are presented in Table 2. The following observations are made concerning the tracking performance.

- The RMS error during constant velocity in the final trial is 43% lower for LPV feedforward than for LTI feedforward as shown in Fig. 11 and Table 2.
- The maximum absolute error in the final trial $\|e_{10}\|_\infty$ during constant velocity improved by 25% by using LPV feedforward compared to LTI feedforward, as illustrated in Fig. 12 and Table 2.

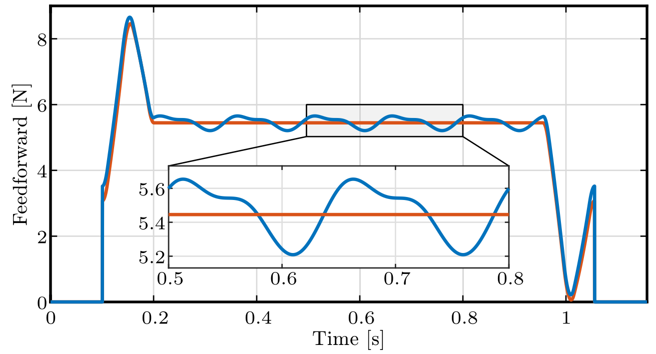


Fig. 10. The feedforward signal during the final trial f_{10} shows the periodic effect obtained by the LPV feedforward signal (—) compared to the LTI feedforward signal (—).

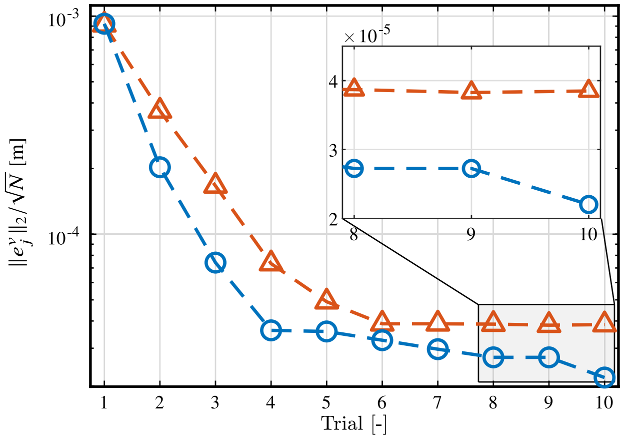


Fig. 11. The RMS error during constant velocity e_j^v converges faster and to a lower value for the developed iterative kernel-regularized estimates of LPV parameters (○) compared to the LTI feedforward parameters (△).

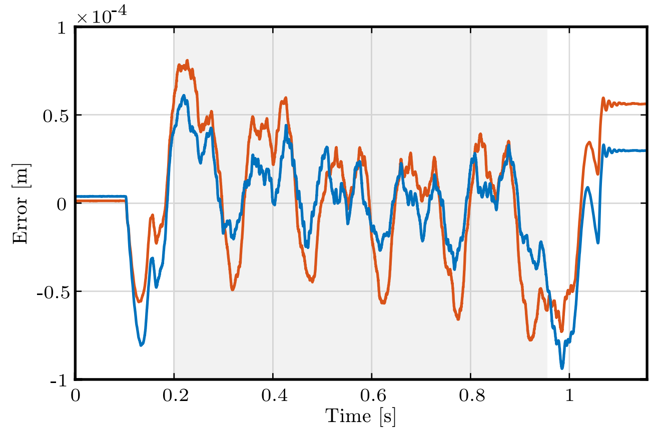


Fig. 12. During constant velocity (▮), the tracking error in the final trial e_{10} is reduced by the developed iterative kernel-regularized estimates of LPV feedforward parameters (—) compared to the LTI feedforward parameters (—).

- The amplitude of the first and second harmonic are decreased by respectively a factor 6 and 2.3 in terms of their power, as shown in Fig. 13 and Table 2.

The observations show that learning feedforward parameter functions using the developed iterative kernel-regularized estimator increases performance significantly for motion systems.

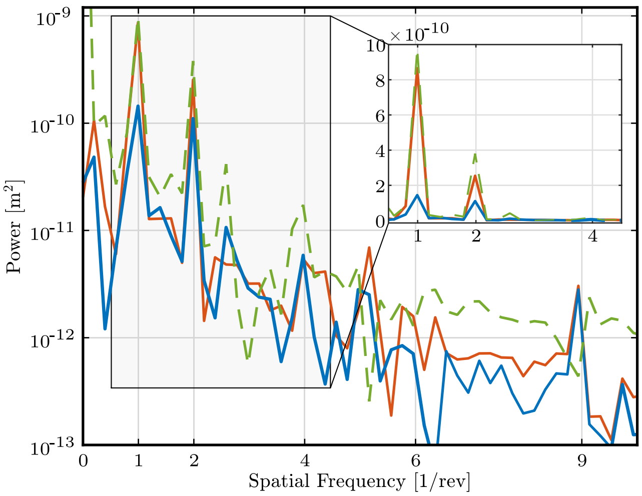


Fig. 13. The power of the tracking error for the final trial e_{10} is significantly lower for the developed LPV feedforward control (—) compared to LTI feedforward control (—) and zero feedforward $f_j = 0$ (- -).

Table 2
Experimental error metrics during constant velocity for LTI and LPV feedforward.

Metric	LTI	LPV	Unit
$\|e^v_{10}\|_2/\sqrt{N}$	$3.85 \cdot 10^{-5}$	$2.2 \cdot 10^{-5}$	[m]
$\|e^v_{10}\|_\infty$	$8.1 \cdot 10^{-5}$	$6.1 \cdot 10^{-5}$	[m]
Power e^v_{10} (1/rev)	$8.65 \cdot 10^{-10}$	$1.44 \cdot 10^{-10}$	[m^2]
Power e^v_{10} (2/rev)	$2.54 \cdot 10^{-10}$	$1.10 \cdot 10^{-10}$	[m^2]

6. Conclusions

The results in this paper enable data-driven learning of parameter-variations in feedforward control, which can significantly improve the tracking performance of LPV systems. The key idea is to iteratively learn LPV feedforward parameter functions using kernel-regularized estimation. Kernel-regularized function estimation is advantageous since it is non-parametric, meaning no specific structure needs to be enforced on the dependency on the scheduling sequence. The iterative learning approach directly optimizes the tracking error, which enhances the estimation quality without the need for accurate system models. The developed framework is experimentally validated on a belt-driven motion system, demonstrating effective compensation of position-dependent drivetrain dynamics. The developed method demonstrates significant potential for industrial applications, particularly in mechatronic systems, by enabling improved tracking performance.

Future work includes experimental comparison of the developed framework to other LPV or non-linear feedforward control approaches.

CRediT authorship contribution statement

Max van Haren: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lennart Blanken:** Writing – review & editing, Software, Investigation, Data curation. **Tom Oomen:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tom Oomen reports financial support was provided by the Dutch Research Council. Max van Haren reports financial support was provided by Horizon Europe. The co-author Tom Oomen is both Senior Editor of IEEE Control Systems Letters (L-CSS) and Co-Editor-in-Chief of IFAC Mechatronics. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Clayton GM, Tien S, Leang KK, Zou Q, Devasia S. A review of feedforward control approaches in nanopositioning for high-speed SPM. J Dyn Syst Meas Control 2009;131(6). http://dx.doi.org/10.1115/1.4000158.

[2] Grotjahn M, Heimann B. Model-based feedforward control in industrial robotics. Int J Robot Res 2002;21(1):45–60. http://dx.doi.org/10.1177/027836402320556476.

[3] Devasia S. Should model-based inverse inputs be used as feedforward under plant uncertainty? IEEE Trans Autom Control 2002;47(11). http://dx.doi.org/10.1109/TAC.2002.804478.

[4] Groot Wassink M, van de Wal M, Scherer C, Bosgra O. LPV control for a wafer stage: beyond the theoretical solution. Control Eng Pract 2005;13(2). http://dx.doi.org/10.1016/j.conengprac.2004.03.008.

[5] Hoffmann C, Werner H. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. IEEE Trans Control Syst Technol 2015;23(2):416–33. http://dx.doi.org/10.1109/TCST.2014.2327584.

[6] Bamieh B, Giarré L. Identification of linear parameter varying models. Internat J Robust Nonlinear Control 2002;12(9):841–53. http://dx.doi.org/10.1002/rnc.706.

[7] Previdi F, Lovera M. Identification of non-linear parametrically varying models using separable least squares. Internat J Control 2004;77(16):1382–92. http://dx.doi.org/10.1080/0020717041233318863.

[8] Tóth R. Modeling and identification of linear parameter-varying systems. Springer; 2010, http://dx.doi.org/10.1007/978-3-642-13812-6.

[9] Zhao Y, Huang B, Su H, Chu J. Prediction error method for identification of LPV models. J Process Control 2012;22(1):180–93. http://dx.doi.org/10.1016/j.jprocont.2011.09.004.

[10] Theis J, Pfifer H, Knoblach A, Saupe F, Werner H. Linear parameter-varying feedforward control: A missile autopilot design. In: Proceedings of the AIAA guidance, navigation, and control conference. 2015, http://dx.doi.org/10.2514/6.2015-2001.

[11] de Rozario R, Oomen T, Steinbuch M. Iterative learning control and feedforward for LPV systems: Applied to a position-dependent motion system. In: Proceedings of the American control conference. 2017, p. 3518–23. http://dx.doi.org/10.23919/ACC.2017.7963491.

[12] Bloemers T, Proimadis I, Kasemsinsup Y, Toth R. Parameter-dependent feedforward strategies for motion systems. In: Proceedings of the American control conference. 2018, p. 2017–22. http://dx.doi.org/10.23919/ACC.2018.8431116.

[13] Butcher M, Karimi A. Data-driven tuning of linear parameter-varying precompensators. Internat J Adapt Control Signal Process 2010;24(7):592–609. http://dx.doi.org/10.1002/acs.1151.

[14] de Rozario R, Pelzer R, Koekebakker S, Oomen T. Accommodating trial-varying tasks in iterative learning control for LPV systems, applied to printer sheet positioning. In: Proceedings of the American control conference. 2018, p. 5213–8. http://dx.doi.org/10.23919/ACC.2018.8430906.

[15] Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. IEEE Control Syst 2006;26(3):96–114. http://dx.doi.org/10.1109/MCS.2006.1636313.

[16] Kon J, van de Wijdeven J, Bruijnen D, Tóth R, Heertjes M, Oomen T. Direct learning for parameter-varying feedforward control: A neural-network approach. In: Proceedings of the 62nd IEEE conference on decision and control. 2023, p. 3720–5. http://dx.doi.org/10.1109/CDC49753.2023.10383877.

[17] Formentin S, Piga D, Tóth R, Savaresi SM. Direct learning of LPV controllers from data. Automatica 2016;65:98–110. http://dx.doi.org/10.1016/j.automatica.2015.11.031.

[18] Verhoek C, Abbas HS, Tóth R, Haesaert S. Data-driven predictive control for linear parameter-varying systems. In: 4th IFAC workshop on linear parameter varying systems. LPVS, 2021, p. 101–8. http://dx.doi.org/10.1016/j.ifacol.2021.08.588.

[19] Nortmann B, Mylvaganam T. Direct data-driven control of linear time-varying systems. IEEE Trans Autom Control 2023;68(8):4888–95. http://dx.doi.org/10.1109/TAC.2023.3276909.

[20] Pillonetto G, Dinuzzo F, Chen T, De Nicolao G, Ljung L. Kernel methods in system identification, machine learning and function estimation: A survey. Automatica 2014;50(3):657–82. http://dx.doi.org/10.1016/j.automatica.2014.01.001.

[21] Chen T, Ohlsson H, Ljung L. On the estimation of transfer functions, regularizations and Gaussian processes-Revisited. Automatica 2012;48(8). http://dx.doi.org/10.1016/j.automatica.2012.05.026.

[22] Pillonetto G. A new kernel-based approach to hybrid system identification. Automatica 2016;70. http://dx.doi.org/10.1016/j.automatica.2016.03.011.

[23] Liu M, Chowdhary G, Castra da Silva B, Liu S-Y, How JP. Gaussian processes for learning and control: A tutorial with examples. IEEE Control Syst 2018;38(5):53–86. http://dx.doi.org/10.1109/MCS.2018.2851010.

[24] Koller T, Berkenkamp F, Turchetta M, Krause A. Learning-based model predictive control for safe exploration. In: IEEE conference on decision and control. CDC, 2018, p. 6059–66. http://dx.doi.org/10.1109/CDC.2018.8619572.

[25] de Kruif B, de Vries T. On using a support vector machine in learning feedforward control. In: IEEE/aSME international conference on advanced intelligent mechatronics. 2001, p. 272–7. http://dx.doi.org/10.1109/AIM.2001.936466.

[26] Blanken L, Oomen T. Kernel-based identification of non-causal systems with application to inverse model control. Automatica 2020;114:108830. http://dx.doi.org/10.1016/j.automatica.2020.108830.

[27] van Haren M, Poot M, Portegies J, Oomen T. Position-dependent snap feedforward: A Gaussian process framework. In: 2022 American control conference. ACC, 2022, p. 4778–83. http://dx.doi.org/10.23919/ACC53348.2022.9867449.

[28] van Haren M, Blanken L, Oomen T. A kernel-based identification approach to LPV feedforward: With application to motion systems. In: Proceedings of the 22nd world congress of the IFAC. 2023, p. 6063–8. http://dx.doi.org/10.1016/j.ifacol.2023.10.662.

[29] Perneder R, Osborne I. Springer; 2012, http://dx.doi.org/10.1007/978-3-642-17755-2.

[30] Lambrechts P, Boerlage M, Steinbuch M. Trajectory planning and feedforward design for electromechanical motion systems. Control Eng Pract 2005;13(2):145–57. http://dx.doi.org/10.1016/j.conengprac.2004.02.010.

[31] Boeren F, Oomen T, Steinbuch M. Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification. Control Eng Pract 2015;37:11–9. http://dx.doi.org/10.1016/j.conengprac.2014.12.015.

[32] van de Wijdeven J, Bosgra O. Using basis functions in iterative learning control: analysis and design theory. Internat J Control 2010;83(4). http://dx.doi.org/10.1080/00207170903334805.

[33] van der Meulen SH, Tousain RL, Bosgra OH. Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer. J Dyn Syst Meas Control 2008;130(5):051006/1–051006/16. http://dx.doi.org/10.1115/1.2957626.

[34] Rasmussen CE. Gaussian processes for machine learning. MIT Press; 2006.

[35] Steinbuch M, Merry R, Boerlage M, Ronde M, van de Molengraft M. Advanced motion control design. In: The control handbook, control system applications. 2nd ed.. CRC Press; 2010, p. 27/1–5. http://dx.doi.org/10.1201/b10382-35.

Max van Haren is currently working toward the Ph.D. degree in mechanical engineering with the Control Systems Technology section, Eindhoven University of Technology. He received the M.Sc. degree (cum laude) in mechanical engineering in 2021 from the Eindhoven University of Technology, Eindhoven, The Netherlands. His research interests include control and identification of mechatronic systems, including sampled-data, multirate, and linear parameter-varying systems.

Lennart Blanken is currently a System Designer Mechatronics with Sioux Technologies, Eindhoven, The Netherlands. Additionally, he is an assistant professor with the Department of Mechanical Engineering at the Eindhoven University of Technology. He received the M.Sc. degree (cum laude) and Ph.D. degree in mechanical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2015 and 2019, respectively. His research interests include advanced feedforward control, learning control, repetitive control, and their applications to mechatronic systems.

Tom Oomen is full professor with the Department of Mechanical Engineering at the Eindhoven University of Technology. He is also a part-time full professor with the Delft University of Technology. He received the M.Sc. degree (cum laude) and Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands. He held visiting positions at KTH, Stockholm, Sweden, and at The University of Newcastle, Australia. He is a recipient of the 7th Grand Nagamori Award, the Corus Young Talent Graduation Award, the IFAC 2019 TC 4.2 Mechatronics Young Research Award, the 2015 IEEE Transactions on Control Systems Technology Outstanding Paper Award, the 2017 IFAC Mechatronics Best Paper Award, the 2019 IEEJ Journal of Industry Applications Best Paper Award, and recipient of a Veni and Vidi personal grant. He is currently a Senior Editor of IEEE Control Systems Letters (L-CSS) and Co-Editor-in-Chief of IFAC Mechatronics, and he has served on the editorial board of IEEE Transactions on Control Systems Technology. He has also been vice-chair for IFAC TC 4.2 and a member of the Eindhoven Young Academy of Engineering. His research interests are in the field of data-driven modeling, learning, and control, with applications in precision mechatronics.