2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Recursive learning of feedforward parameters in high-tech motion systems: An experimental case study

Thijs van Keulen[1,2], Bram Kleefstra[1], Ruud Beerens[2]

***Abstract*— This article provides an experimental case study of a state-of-the-art recursive feedforward parameter learning framework on a high-tech industrial metrology and inspection machine. The aim of the learning framework is to recursively adapt the feedforward parameters to compensate for time-varying and position-dependent system behavior, e.g., caused by wear, position and temperature dependent actuator characteristics, changes in shape and stiffness due to thermal expansion, and sample time jitter. The strength of the approach is demonstrated through experiments on a high-tech motion system which show a peak error reduction of circa 45% compared to the industrial controller with offline calibrated feedforward parameters.**

I. Introduction

High-tech motion systems such as industrial printers, electron microscopes, lithography machines, and metrology inspection tools experience an ever-increasing demand in positioning accuracy and throughput, which translates to stringent requirements on motion control performance. Whereas feedback control is essential for achieving external disturbance suppression, feedforward control aims at compensation for reference-induced disturbances, which is crucial for achieving accurate reference tracking.

Feedforward control for high-tech motion systems is often designed based on a limited-order parametric model resembling the reciprocal plant dynamics. Such a compensation structure utilizes higher-order derivatives of the reference position, e.g., velocity, acceleration, jerk, and snap to generate the force trajectory that leads to optimal reference tracking. In this feedforward structure, acceleration induced forces relate to the inertial forces of the system, along with velocity-related forces caused by, e.g., viscous friction or motor damping forces. A jerk related feedforward term is used to compensate digital-to-analog converter effects and (sub)sample time mismatch between feedback and acceleration feedforward [13], and snap feedforward can be used to compensate for compliance in the system [5], [12].

Given the complexity of the controlled plant, feedforward accuracy is hindered by, e.g., the limited structure of the feedforward model, time or position dependency of the controlled plant dynamics, or calibration errors. In order to improve feedforward accuracy, and, thereby, reference tracking performance, several learning strategies exist that aim at correcting for position dependency or plant variations by online learning or adaptation of the feedforward signal. A well known example is iterative learning control, see, e.g., [8], where a feedforward signal is updated between iterations of a repetitive positioning task. Further advancements has led to increased robustness to setpoint variations, see, e.g., [18], [7]. Iterative feedforward tuning [4], [1] exploits a data-driven optimization of feedforward parameters in either a physics-guided structure or a FIR filter. These techniques, however, are only suitable for motion systems with repetitive servo tasks, and may, therefore, be restrictive.

Adaptive control, see, e.g., [15], [9], provides a continuous feedforward parameter adaptation by solving an online parameter estimation problem recursively. However, these methods are sensitive to bias due to, e.g., measurement noise, which is addressed in, e.g., [16] and [17], by compiling an approach for recursive feedforward estimation based on instrumental variables. In particular, the approach in [17] is a continuous-time approach to online parameter adaptation, which continuously updates feedforward parameters during machine operation and is robust to any bounded setpoint variation. Specifically, the method uses a low complexity model of the controlled plant in combination with basis functions and selected sub-optimal feedforward parameters as instrumental variables to recursively -and bias free- estimate the optimal feedforward parameters. The plant model is hereby used to extract individual contributions of the feedforward parameters on the (measured) positioning error, based on which the optimal feedforward parameters are estimated in order to minimize the error. The framework exploits the time-scale separation principle well known in the context of extremum-seeking control, see, e.g., [19], by employing fast parameter estimation and relatively slow parameter updates.

This paper presents the implementation and an experimental performance assessment of the feedforward parameter learning framework presented in [17] on the wafer stage of a state-of-the-art industrial metrology inspection machine, used in the production process of microchips. This system has inherently position-dependent dynamics due to its geometry, motor constant variations and cable slabs. Frictional effects, that are hard to measure and to compensate for by offline calibrated feedforward parameters, play a significant role.

The remainder of this article is organized as follows. In Section II, we provide a closed-loop system description and a parameterized feedforward control design. The learning feedforward framework of [17] and its implementation aspects are discussed in Section III. The experimental performance assessment is presented in Section IV, and conclusions are provided in Section V.

[1]Eindhoven Univ. of Tech., Dep. of Mechanical Engineering, Control Systems Technology group, The Netherlands.
[2]ASML, De Run 6501, 5505DR Veldhoven, The Netherlands.
ruud.beerens-rbkg@asml.com, t.a.c.v.keulen@tue.nl

II. Feedback and feedforward control design for high-tech motion systems

In this section, we present a model of the dynamics of a high-tech motion system and the design of the reference trajectory. Furthermore, the design of the feedback and feedforward control is elaborated.

A. System dynamics

High-tech motion systems can generally be described by [5], [6]

$$P(s) = \frac{1}{ms^2 + \zeta s} + \sum_{k=1}^{N_{\text{flex}}} \frac{u_k^\top v_k}{(s^2 + 2\zeta_k \omega_k s + \omega_k^2)}, \quad (1)$$

in which s is the Laplace variable, m is the mass of the system, ζ is the viscous damping parameter, N_{flex} are the number of flexible modes, u_k and v_k are the associated eigenmode vectors, ω_k is the kth resonance frequency, ζ_k is the damping of the kth flexible mode. Note that, at $\lim_{s\to 0} P(s)$, the flexible modes have a static contribution, namely, a constant defined as $P_{\text{NRB0}} \coloneqq P(0) \in \mathbb{R}$, i.e., the compliance of the system.

B. The reference trajectory

The reference trajectory for motion systems can be shaped to minimise the duration of a move to a desired position, to comply with the force and power limits of the actuator by limiting the velocity and acceleration values, and to mitigate the excitation of resonances in the plant by selecting appropriate jerk and snap values, see [2], [14].

To ensure convergence of the optimization process, the following assumptions are required.

Assumption 1. *The reference trajectory of the system is assumed to be a non-zero trajectory.*

Assumption 2. *The first four derivatives of the position set-point are bounded, i.e., the velocity, acceleration, jerk and snap. Furthermore, zero velocity is assumed at the start and the end of the reference trajectory.*

Assumption 1 provides a persistence-of-excitation condition to the optimization problem while Assumption 2 ensures a bounded solution.

C. Control design

The control design includes a combination of feedforward and feedback, see Fig. 1.

The feedback control entails Proportional-Integral-Derivative and second-order Low-Pass (PID+LP) filtering, see, e.g., [10]

$$C_{\text{pid}}(s) = k_{\text{p}} \left(1 + \frac{s}{\omega_{\text{d}}} + \frac{\omega_{\text{i}}}{s}\right) \cdot \frac{\omega_{\text{lp}}^2}{s^2 + 2\zeta_{\text{lp}}\omega_{\text{lp}} s + \omega_{\text{lp}}^2}, \quad (2)$$

where k_{p} is the proportional gain, ω_{d} the derivative frequency, ω_{i} the integrator frequency, ω_{lp} the cut-off frequency of the low-pass filter and ζ_{lp} the damping of the low-pass filter.

In addition to (2), notch filtering can be used to dampen excited flexible plant modes

$$C_{\text{notch}}(s) = \prod_{i=1}^{q} \left(\frac{s^2 + 2\beta_{z,i}\omega_{z,i} s + \omega_{z,i}^2}{s^2 + 2\beta_{p,i}\omega_{p,i} s + \omega_{p,i}^2} \right). \quad (3)$$

Here, $\beta_{z,i}$ and $\beta_{p,i}$ are the damping parameters, and $\omega_{z,i}$ and $\omega_{p,i}$ are the notch pole and zero frequency of the qth notch filter, respectively. The feedback control is then defined by:

$$C_{\text{fb}}(s) = C_{\text{pid}}(s) \cdot C_{\text{notch}}(s). \quad (4)$$

The feedforward control uses the reference trajectory to calculate the force input required to move the system in a desired way. The optimal feedforward resembles the inverse of the plant, $C_{\text{ff}}(s) = P^{-1}(s)$, such that, $y = P(s)P^{-1}(s)r$. However, in the case of high-tech motion systems (1), inverting $P(s)$ yields an improper transfer function. Nevertheless, in [6] it is shown that, although inverting the plant is not possible, the feedforward control can be modelled as a linear combination of feedforward parameters and derivatives of the reference signal. Conventionally, these feedforward parameters are calibrated offline. However, as indicated before, the plant behavior can vary in time.

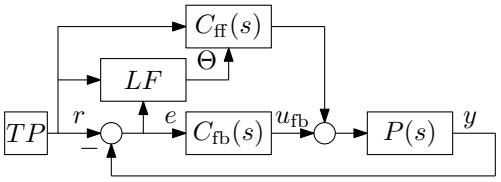

Fig. 1. Interconnection of a system working in a closed loop with plant P, Trajectory Planner (TP), feedback control C_{fb} and feedforward control C_{ff}, and Learning Framework (LF).

Agains this backdrop, we define the following feedforward control structure

$$C_{\text{ff}}(s, \Theta) = \hat{\zeta} s + \hat{m} s^2 + \Theta(s)\Psi(s), \quad (5)$$

where $\hat{\zeta}$ and $\hat{m}$ are the offline calibrated damping and mass estimates, respectively, $\Theta(s) \in \mathbb{R}^{1\times 4}$ is a vector of feedforward parameters that can compensate time-varying plant behavior

$$\Theta(s) = \begin{bmatrix} \theta_{\text{v}}(s) & \theta_{\text{a}}(s) & \theta_{\text{j}}(s) & \theta_{\text{d}}(s) \end{bmatrix}, \quad (6)$$

in which, θ_{v}, θ_{a}, θ_{j}, and θ_{d}, are the velocity, acceleration, jerk and snap parameters, respectively. Furthermore, $\Psi \in \mathbb{R}^{4\times 1}$ denotes an array of basis functions given by

$$\Psi(s) = \begin{bmatrix} s & s^2 & s^3 & s^4 \end{bmatrix}^\top. \quad (7)$$

The relation between reference signal r and error signal e, is described by the following transfer function,

$$H(s, \Theta) = \frac{e}{r} = \frac{1 - P(s)C_{\text{ff}}(s, \Theta)}{1 + P(s)C_{\text{fb}}(s)}. \quad (8)$$

Under the assumptions [17] that $\theta_{\text{a}} \ll m$ and that the rigid-body approximation $\hat{P}(s) = \frac{1}{\hat{m}s^2 + \hat{\zeta}s}$ of (1) is accurate at

frequencies below the first resonance, we can approximate (8) by

$$\hat{H}(s,\Theta) = -\hat{S}_{\mathrm{p}}(s)\Theta(s)\Psi(s), \tag{9}$$

where

$$\hat{S}_{\mathrm{p}}(s) = \frac{\hat{P}(s)}{1+\hat{P}(s)C_{\mathrm{pid}}(s)}, \tag{10}$$

is the rigid-body process sensitivity function estimate. Crucially for the learning framework, relation (9) is linear in the parameters Θ.

III. Feedforward parameter learning framework

A. Problem description

Consider now the learning objective to adapt the feedforward parameters Θ in (5) such that the squared servo error gets minimized:

$$\Theta^{\star}(t) = \arg\min_{\Theta(t)} \int_{-\infty}^{t} e^2(\tau,\Theta(\tau))\mathrm{d}\tau. \tag{11}$$

To limit the solution space, a constant parameter estimate is considered in a receding horizon of past measurement data. In this context, we can construct the parameter estimation problem with solution:

$$\overline{\Theta}^{\star} = \arg\min_{\overline{\Theta}} \int_{t-\mathrm{T}}^{t} e^2(\tau,\overline{\Theta})\mathrm{d}\tau, \tag{12}$$

in which T is the learning horizon and $\overline{\Theta}^{\star}$ is a vector of constant parameters that provides the best fit in the considered horizon of past data.

B. Parameter estimation

To adapt the feedforward control, we thus seek a mapping of measured error e to the parameter set Θ. To this end, let us choose the following receding-horizon objective function:

$$V(t,\overline{\Theta}^{\delta}) = \frac{1}{\mathrm{T}}\int_{t-\mathrm{T}}^{t}\left(e(\tau,\Theta(\tau)) - \Phi(\tau,\overline{\Theta}^{\mathrm{IV}})\overline{\Theta}^{\delta}\right)^2\mathrm{d}\tau, \tag{13}$$

where $\Theta(.)$ are the actual applied feedforward parameters within the learning horizon, $\overline{\Theta}^{\delta} \in \mathbb{R}^{4\times 1}$ is the scaling of the columns of $\Phi(t,\overline{\Theta}^{\mathrm{IV}}) \in \mathbb{R}^{T\times 4}$, i.e., the regressors defined as

$$\Phi(t,\overline{\Theta}^{\mathrm{IV}}) \coloneqq \hat{H}(t,\overline{\Theta}^{\mathrm{IV}})r, \tag{14}$$

in which $\hat{H}$ is defined in (9) and $\overline{\Theta}^{\mathrm{IV}} \in \mathbb{R}^{4\times 1}$ are selected constant sub-optimal compensation parameters used as instrumental variables (IV) [3].

Since (13) is quadratic in the parameters a least-squares solution of the objective function is given by

$$\overline{\Theta}^{\delta}(t) = \frac{1}{\int_{t-\mathrm{T}}^{t}\Phi^{\top}(\tau)\Phi(\tau)\mathrm{d}\tau}\int_{t-\mathrm{T}}^{t}\Phi^{\top}(\tau,\overline{\Theta}^{\mathrm{IV}})e(\tau,\Theta)\mathrm{d}\tau. \tag{15}$$

The recursive least squares solution is computationally expensive. To reduce the memory allocation, the moving-average filter from (15) is replaced by a first-order low-pass filter and the term $\frac{1}{\int_{t-\mathrm{T}}^{t}\Phi^{\top}(\tau)\Phi(\tau)\mathrm{d}\tau}$ is replaced with a constant $\mathbf{C}$, leading to the approximate least-squares solution

$$\tilde{\Theta}^{\delta} = \frac{\omega_{\mathrm{T}}}{s+\omega_{\mathrm{T}}}\mathbf{C}\Phi^{\top}(k,\overline{\Theta}^{\mathrm{IV}})e(k,\Theta). \tag{16}$$

Here, $\omega_{\mathrm{T}} = \frac{1}{\mathrm{T}}$ is the cut-off frequency of the low-pass filter that replaces the moving average.

C. Parameter adaptation

Immediate adaption of the estimated feedforward parameters $\tilde{\Theta}^{\delta}$ in (5) would affect the solution (16) in future time. Therefore, a relatively slow adaption of the feedforward parameters is needed [17] compared to the learning horizon required to estimate $\tilde{\Theta}^{\delta}$.

Against this backdrop, consider a slow linear dynamic system, where the input is the estimated scaling parameter $\tilde{\Theta}^{\delta}$ and the output is the adapted feedforward parameter set, Θ used in (5):

$$F(s) = \begin{bmatrix} \frac{\omega_{\mathrm{v}}}{s} & \frac{\omega_{\mathrm{a}}}{s} & \frac{\omega_{\mathrm{j}}}{s} & \frac{\omega_{\mathrm{s}}}{s} \end{bmatrix}, \tag{17}$$

where ω_{v}, ω_{a}, ω_{j} and ω_{s} are the adaption frequencies of the feedforward parameters. Note that, the following condition must hold $\left\{ \omega_{\mathrm{v}}, \quad \omega_{\mathrm{a}}, \quad \omega_{\mathrm{j}}, \quad \omega_{\mathrm{s}} \right\} \ll \omega_{\mathrm{T}}$, for the adaption not to affect the estimation this is based on the time-scale separation principle [17], as shown later in this section.

Crucial for the stability of the learning framework is time-scale separation [11]. Using the time-scale separation principle, we demand the adaption of the feedforward parameters to be sufficiently slow. This way, the feedforward parameter over the finite time window is considered to be steady-state and will not influence the parameter estimate in the next step of the recursive scheme. In order to demonstrate this, we show that the change in time of $\Theta(t)$ (6), $\frac{\mathrm{d}\Theta}{\mathrm{d}t}(t)$, is small compared to the change of $\tilde{\Theta}^{\delta}(t)$ (16) in time, $\frac{\mathrm{d}\tilde{\Theta}^{\delta}}{\mathrm{d}t}(t)$. The update rate is defined with $\omega_{\mathrm{T}} = \frac{1}{\mathrm{T}}$. In this context, we obtain the update-rate of (16) as a low-pass filter with the learning horizon as cut-off frequency:

$$\frac{1}{\omega_{\mathrm{T}}}\frac{\mathrm{d}\tilde{\Theta}^{\delta}}{\mathrm{d}t} = \mathrm{C}\Phi(t,\overline{\Theta}^{\mathrm{IV}})e(t,\Theta(t)) - \tilde{\Theta}^{\delta}(t). \tag{18}$$

Note that, $e(t,\Theta(t))$ is dependent on $\Theta(t)$. To demonstrate that the dependency of $e(t,\Theta(t))$ on $\Theta(t)$ does not introduce instability, we need to show that, on the time frame of the parameter estimation, the rate of change of $\Theta(t)$ in time, $\frac{\mathrm{d}\Theta}{\mathrm{d}t}(t)$, is negligible. We define the dynamics of $\frac{\mathrm{d}\Theta}{\mathrm{d}t}(t)$ by

$$\frac{\mathrm{d}\Theta}{\mathrm{d}t} = \omega_{\mathrm{F}}\tilde{\Theta}^{\delta}(t), \tag{19}$$

see (17). We normalise the time variable with the frequency ω_{T}, $\sigma = \omega_{\mathrm{T}}t$ and rewrite (19) as

$$\frac{\mathrm{d}\Theta}{\mathrm{d}\sigma} = \frac{\omega_{\mathrm{F}}}{\omega_{\mathrm{T}}}\tilde{\Theta}^{\delta}(\sigma). \tag{20}$$

Using the new time variable σ, (18) rewrites as

$$\frac{\mathrm{d}\tilde{\Theta}^{\delta}}{\mathrm{d}\sigma} = \mathrm{C}\Phi(\sigma, \overline{\Theta}^{\mathrm{IV}})e(\sigma, \Theta(\sigma)) - \tilde{\Theta}^{\delta}(\sigma). \tag{21}$$

Knowing that the initial condition of $\Theta(t)$ is $\Theta(0) = \Theta^{\mathrm{IV}}$, we can solve (20) as

$$\Theta(\sigma) = \Theta^{\mathrm{IV}} + \varepsilon \int_0^{\sigma} \tilde{\Theta}^{\delta}(\tau)\mathrm{d}\tau, \tag{22}$$

where $\varepsilon = \frac{\omega_{\mathrm{F}}}{\omega_{\mathrm{T}}}$ is a small, dimensionless parameter. Since we can select $\omega_{\mathrm{T}} \gg \omega_{\mathrm{F}}$, we can consider ε negligibly small. Using the singular perturbation method [11], we can set $\varepsilon = 0$ in (22), which results in $\Theta(\sigma) = \Theta^{\mathrm{IV}}$. Following this, $\Theta(\sigma)$ can be considered "quasi-steady-state" [11] and substituted in (21) to show the parameter estimation is not dependent on the parameter update:

$$\frac{\mathrm{d}\tilde{\Theta}^{\delta}}{\mathrm{d}\sigma} = \mathrm{C}\Phi(\sigma, \overline{\Theta}^{\mathrm{IV}})e(\sigma, \Theta^{\mathrm{IV}}) - \tilde{\Theta}^{\delta}(\sigma). \tag{23}$$

This yields a non-varying $\Theta(t)$ on the time frame normalised to the parameter estimation time-scale, $\sigma = \omega_{\mathrm{T}}t$. Considering the relatively slow adaption of the feedforward parameters, the following situations are possible:

- The plant characteristics do not change over time, this framework will yield the same solution as the infinite horizon case, only as fast as the adaption function in (17);
- The plant characteristics change faster than the adaption function in (17), the solution of the feedforward parameter will be averaged out, depending on the rate of change of the feedforward parameter and the adaption frequency;
- The plant characteristics change slower than the adaption function in (17) the change in the feedforward parameter will be fully captured by the learning framework.

Thus, the feedforward parameters will be estimated in a stable manner.

IV. Experimental case study

In this section we present an experimental case study on an industrial metrology inspection machine. We provide a system description, after which we present and discuss the experimental results.

A. System description

The case study is performed on the wafer stage of an industrial metrology inspection machine used in the production process of microchips. Modern microchips contain several layers that all need to align on top of each other as accurately as possible. Such an overlay property is measured by a metrology inspection machine, which positions a wafer under on optical sensor using a wafer stage. The sensor measures overlay by emitting light onto specific markers at several locations on the wafer and measures back the diffraction pattern. The intensity difference of the scattered light pattern is translated to a metric for overlay accuracy. To inspect all markers on the wafer, the wafer stage performs a point-to-point motion profile from one marker to the next, with stringent requirements on settling time and peak-to-peak positioning errors during overlay measurements.

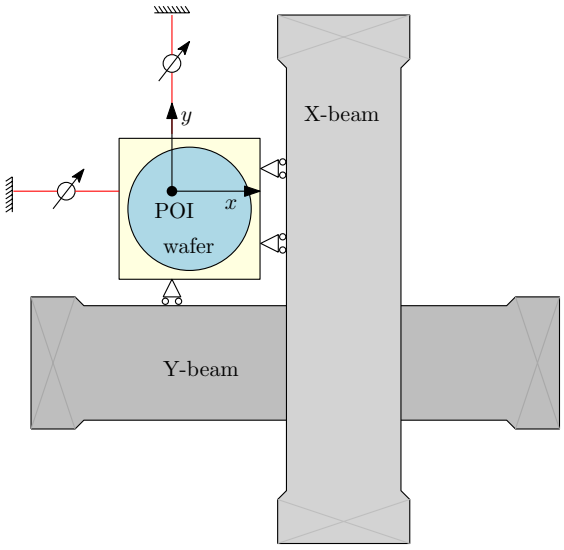

Fig. 2. Schematic representation of the wafer stage of the optical metrology inspection machine. The point-of-interest (POI) is measured using interferometers.

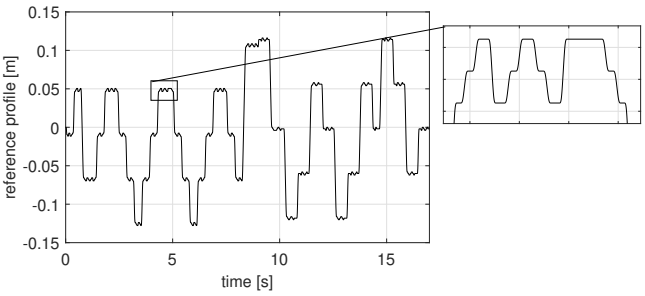

Fig. 3. Wafer stage reference profile in x direction, comprising of large and small point-to-point moves.

Consider Fig. 2, which depicts a schematic represenation of the wafer stage. The stage consists of two perpendicular beams, actuated at each side, dragging along a wafer carrier. The logical degrees of freedom are the translations in x and y direction, and individual rotations of the beams. A geometric decoupling method maps the actuator forces to forces and moments acting on the logical degrees of freedom. In this study, we consider only motion in x direction subject to a reference profile that is typical for normal operation of the machine, see Fig. 3. The reference profile consists of several large and small point-to-point moves, covering the full wafer in x direction.

The *default* feedforward strategy (i.e., the current state-of-practice) consists of offline calibrated mass feedforward, delay compensation, and compliance compensation, see, e.g., [12]. The latter corrects the measured wafer stage position for contributions related to compliancy (see Sec. II-A), which prevents the feedback controller to react on compliancy

related errors. For the learning framework, we employ a feedforward structure with mass, jerk, and velocity dependent parameters, i.e., θ_a, θ_j, and θ_v, respectively. Since compliance compensation is already applied, snap feedforward is not used.

B. Experimental results

We will now compare the performance of the wafer stage for 1) the case without the feedforward learning framework applied (i.e., using the default feedforward setting), and for 2) the case with the learning feedforward algorithm enabled. Recall that compliance compensation remains intact for both cases. Performance is qualified by settling time and positioning errors in the time intervals where overlay measurements of the markers take place, typically between 5ms and 15ms after each point-to-point move ends (the allowable settling time is 5ms).

Consider Fig. 4, which shows the convergence of the feedforward parameters during execution of the reference profile. The figure shows (slow) convergence towards nominal feedforward parameter values, but also the effect of continuous learning of the parameters is visible (see the insets). Namely, small variations in the feedforward parameters over time arise as a result of position-dependent effects, such as varying friction, cable deformation, and motor constant variation.

We will now illustrate the performance benefits of the learning feedforward algorithm compared to the default feedforward. To this end, consider Fig. 5, which depicts the positioning error profile for three point-to-point moves in the reference with different move sizes. For each individual move, we define the time instant where the move ends as $t = 0$. After the move ends, we allow 5ms of settling time after which an overlay measurement (i.e., acquirement) takes place, which, in this case, takes 10ms. These relevant time instances are indicated in Fig. 5 by the vertical blue lines. The error contribution related to the mass and jerk feedforward parameter manifests itself during acceleration and deceleration phases, and therefore plays a significant role in settling performance. The default mass feedforward (and delay compensation) is clearly not optimal, since the application of the learning feedforward results in an overshoot reduction up to 50% in the time interval where an overlay measurement takes place (i.e., at $t \in [5, 15]$ms). The peak-to-peak error for each move in this time interval is given by

$$e_{p2p} := |\max(e(t)) - \min(e(t))|, \tag{24}$$

with $e(t)$, $t \in [5, 15]$ms, the position error. The peak-to-peak error is shown to reduce up to 45%, which results in a significantly more accurate overlay measurement.

Let us now elaborate on the performance benefits of the learning feedforward on the complete reference, after the feedforward parameters have nominally converged. Specifically, we take the last 100 moves of the total reference (which contains 252 moves in total) to omit the initial transient effects of the learning algorithm. We inspect both the peak-to-peak error for each move, as defined in (24), and the

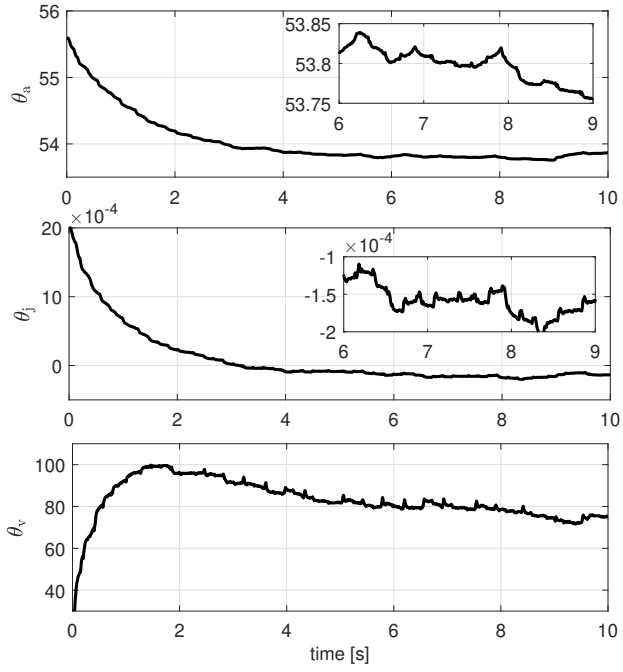

Fig. 4. Convergence of the learning feedforward parameters.

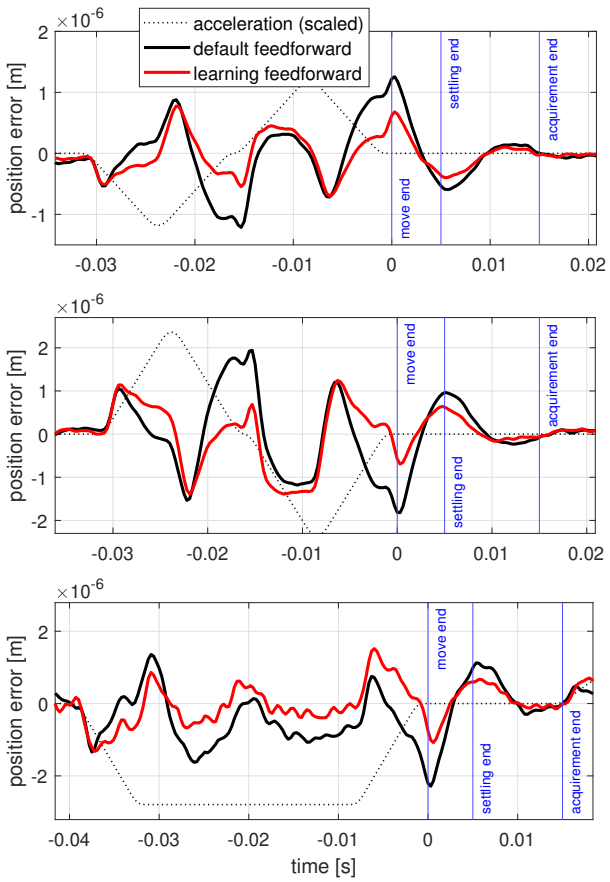

Fig. 5. Position errors for three point-to-point moves with different move sizes.

moving average (MA) at $t = 5$ms for each move, i.e,

$$e_{MA}(t) := \frac{1}{T}\int_{t}^{t+T} e(\tau)d\tau, \tag{25}$$

with $T = 10$ms the acquirement time. We use the MA value at $t = 5$ms (i.e., at the start of an overlay measurement) as a metric for transient performance. Consider Fig. 6, where the MA and peak-to-peak error distribution for 100 subsequent wafer stage moves are depicted. A consistent improvement on both metrics can be observed for the learning feedforward algorithm compared to the default feedforward strategy.

Finally, we observe that the velocity component (which is not present in the default feedforward strategy) of the learning feedforward results in improved reference tracking by compensating for, e.g., viscous friction (which is generally uncertain and time-varying). The significance of its contribution is clearly visible in the Cumulative Power Spectral Density (cPSD) of the position error of the complete reference, presented in Fig. 7, which provides a metric for the total energy of the servo signals.

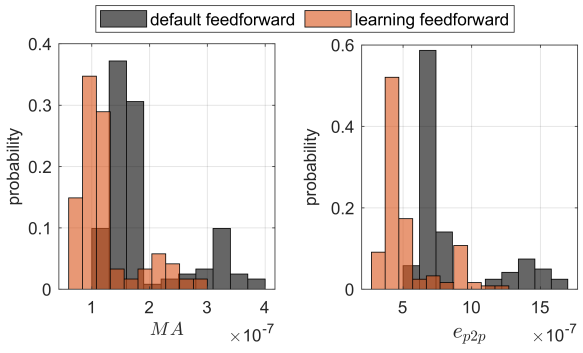

Fig. 6. Distribution of the Moving Average (MA) at $t = 5$ms (left) and peak-to-peak error in the interval $t \in [5, 15]$ms for 100 subsequent wafer stage moves.

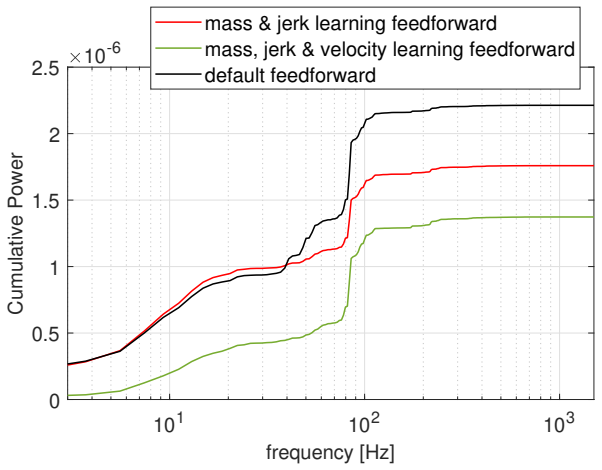

Fig. 7. Cumulative Power Spectral Density of the measured positioning errors.

V. Conclusion and Future Work

This paper studied a feedforward parameter learning framework. The learning framework decomposes the measured error in to the effect of individual feedforward parameters. The proposed method is evaluated with an experimental case study on the wafer-stage of a metrology and inspection machine. In this case study the performance increase for the learning framework becomes apparent. The experimental case study shows a reduction in peak-to-peak error of 45% compared to the default production controller. Future work will focus on extending the learning framework with nonlinear feedback control, and a multi-input multi-output implementation of the learning framework.

References

[1] M. Baggen, M. Heertjes, and R. Kamidi. Data-based feed-forward control in mimo motion systems. *Proc. 2008 American Control Conference (2008)*, pages 3011–3016.

[2] L. Biagiotti and C. Melchiorri. FIR filters for online trajectory planning with time- and frequency-domain specifications. *IEEE Control Systems Magazine*, 20(12):1385–1399, 2012.

[3] F. Boeren, L. Blanken, D. Bruijnen, and T. Oomen. Optimal estimation of rational feedforward controllers: An instrumental variable approach. *Proc. 2015 IEEE Conference on Decision and Control (CDC)*, pages 6058–6063, 2015.

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

[5] M. Boerlage. Mimo jerk derivative feedforward for motion systems. *Proc. 2006 American Control Conference (ACC)*, pages 3892–3897, 2006.

[6] M. Boerlage, R. Tousain, and M. Steinbuch. Jerk derivative feedforward control for motion systems. *Proc. 2004 American Control Conference (ACC)*, pages 4843–4848, 2004.

[7] J. Bolder and T. Oomen. Rational basis functions in iterative learning control with experimental verification on a motion system. *IEEE Trans. Contr. Syst.*, 23:722–729, 2015.

[8] D. Bristow, M. Tharayil, and A Alleyne. A survey of iterative learning control: a learning-based method for high-performance tracking control. *IEEE Control Systems Magazine*, 26:96–114, 2006.

[9] H. Butler. Adaptive feedforward for a wafer stage in a lithography tool. *IEEE Trans. Contr. Syst. Techn.*, 21:875–881, 2012.

[10] M.F. Heertjes, H. Butler, N.J. Dirkx, S.H. van der Meulen, R. Ahlawat, K. O'Brien, J. Simonelli, K-T. Teng, and Y. Zhao. Control of wafer scanners: Methods and developments. *Proc. 2020 American Control Conference (ACC)*, pages 3686–3703, 2020.

[11] H. Khalil. *Nonlinear Systems*. Prentice-Hall, 1996.

[12] N. Kontaras, M. Heertjes, H. Zwart, and M. Steinbuch. A compliance feedforward scheme for a class of ltv motion systems. *Proc. 2017 American Control Conference (ACC)*, pages 4504–4509, 2017.

[13] T.I. Laakso, V. Valimaki, M. Karjalainen, and U.K. Laine. Splitting the unit delay. *IEEE Signal Processing Magazine*, 13(1):30–60, 1996.

[14] P. Lambrechts, M. Boerlage, and M. Steinbuch. Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, 13(2):145–157, 2005.

[15] N. Mooren, G. Witvoet, and T. Oomen. From batch-to-batch to online learning control: experimental motion control case study. *IFAC PapersOnline*, 52:406–411, 2019.

[16] N. Mooren, G. Witvoet, and T. Oomen. On-line instrumental variable-based feedforward tuning for non-resetting motion tasks. *International Journal of Robust and Nonlinear Control*, 2023.

[17] T. van Keulen, T. Oomen, and M. Heemels. Online feedforward parameter learning with robustness to set-point variations. *Proc. 2023 IFAC World Congress*.

[18] J. van Zundert, J. Bolder, and T. Oomen. Optimality and flexibility in iterative learning control for varying tasks. *Automatica*, 67:295–302, 2016.

[19] H.H. Wang and M.' Krstić. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.