2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Feedforward Motion Control: From Batch-to-Batch Learning to Online Parameter Estimation

Noud Mooren[†], Gert Witvoet[†,‡] and Tom Oomen[†]

Abstract— **Feedforward control is essential in high-performance motion control. The aim of this paper is to develop a unified framework for automatic feedforward optimization from both batch-wise data sets as well as real-time data. A statistical analysis is employed to analyze the effect of noise, i.e., an iteration varying disturbance, on feedforward controller performance. This provides new insights, both potential advantages as well as possible hazards of real-time estimation are considered. Finally, a case study confirms and illustrates the results.**

I. INTRODUCTION

Feedforward control, ranging from manual tuning to learning control, is essential for enhancing positioning performance of motion systems. Manual feedforward tuning enables performance improvement by anticipating for known exogenous disturbances, where typically the to be applied reference is used [1]. For systems with repeating motion tasks, learning feedforward algorithms such as iterative learning control (ILC) are able to automatically learn from previous tasks, to compensate for the repeating contributions in the error [2], [3]. These methods benefit from preview, allowing non-causal feedforward controllers to compensate for disturbances before these affect the system, resulting in a major performance improvement compared to causal controllers [4].

Despite potential performance improvement of learning control, standard approaches may actually lead to a performance degradation in typical motion systems, where varying tasks are commonly performed. Standard learning algorithms generate a feedforward signal that exactly compensates for trial-invariant disturbances during a specific task [5], [6]. However, many motion systems perform non-repeating motion tasks, e.g., semiconductor wire-bonding [7], lithography [8] and printing [9]. Hence, flexibility towards trial-varying references is essential, whereas current learning feedforward algorithms are generally highly sensitive to trial-varying exogenous signals [10].

ILC algorithms that are flexible against varying tasks have been developed [9], [6], [10]. These methods combine model-based feedforward and ILC, resulting in flexible learning feedforward. The central idea in this method is to postulate a controller parameterization and learn the parameters using common principles from ILC [2]. From the perspective of system identification, flexible feedforward tuning techniques have been developed which essentially replaces the learning filter design in ILC by an estimation step. In [11], extended parameterizations are explored, encompassing joint input shaping and feedforward control. For both learning and identification methods batch-wise approaches are exploited, i.e., controller parameters are updated after each experiment, where a key aspect is to obtain unbiased estimates. In [8], these learning and identification approaches are compared, directly connection iterative feedforward tuning to inverse system identification.

Although important progress has been made in learning and identification for feedforward control for a class of tasks, these approaches typically consider a batch-wise operation. Consequently, performance improvement takes place after each task is completed. The aim of this paper is to develop a framework for current-iteration feedforward tuning, i.e., updating the feedforward controller during a motion task for direct performance improvement, for general controller structures as in [12]. The main contribution in the present paper is to propose a unified framework for learning feedforward control, covering both batch-wise approaches as the development of current-iteration learning. The following sub-contributions are addressed in this work;

1) a detailed statistical analysis of the proposed framework is provided to show that biased estimates are obtained in earlier approaches, and,
2) a simulation study with a benchmark system is performed to confirm the theoretical conclusions, including the immediate benefit of direct learning.

Indeed, theoretical conclusions in this paper may provide relevant new perspectives on earlier related approaches, including [13], which are further exploited.

This paper is organized as follows. In Section II, the general framework for feedforward control is covered and a parameter optimization problem is proposed. In Section III, a theoretical analysis of the proposed parameter optimization in the presence of noise is provided. In Section IV, a simulation study is conducted in which the proposed framework is applied to a benchmark system. Both the cases with and without noise are compared. Finally, conclusions and ongoing research are provided in Section V.

II. TOWARDS PARAMETER OPTIMIZATION FOR FEEDFORWARD CONTROL

In this section, a general framework is developed for both batch-to-batch learning as well as for current-iteration

* The research leading to these results has received funding from the European Union H2020 program under grant agreement n. 637095 (FourByThree) and ECSEL-2016-1 under grant agreement n. 737453 (I-MECH). Also, 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).

[†] Faculty of Mechanical Engineering, Control Systems Technology Group, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. [‡] TNO Technical Sciences, Optomechatronics department, 2628 CK Delft, The Netherlands. E-mail of corresponding author `n.f.m.mooren@tue.nl`

978-1-5386-7928-9/$31.00 ©2019 AACC

feedforward optimization. First, a feedforward controller parameterization is considered that allows an analytic solution, i.e., through linear least squares optimization. Second, the connection is established between feedforward controller tuning and parameter estimation from a system identification point of view. For the latter, solutions are provided both for a batch-wise setting as in a current-iteration setting.

A. Problem definition in a unified framework

To define the feedforward control problem, consider the general control configuration in Fig. 1 consisting of an input shaper C_r, feedforward controller C_{ff}, feedback controller C_{fb} and true plant P_0. The true plant, mapping inputs u to outputs y_0, is given by

$$P_0(q^{-1}, \theta_0) = \frac{B(q^{-1}, \theta_b^0)}{A(q^{-1}, \theta_a^0)} \tag{1}$$

which is LTI, with $A(q^{-1})$ and $B(q^{-1})$ polynomials in the backward shift operator q^{-1} and, θ_a^0 and θ_b^0 are the true plant coefficients. Furthermore, it is assumed that $A(q^{-1})$ and $B(q^{-1})$ are given by

$$\begin{aligned} A(q^{-1}, \theta_a) &= \sum_{i=1}^{n_a} \psi_i(q^{-1})\theta_{a,i} \\ B(q^{-1}, \theta_b) &= \sum_{i=n_a+1}^{n_b+n_a} \psi_i(q^{-1})\theta_{b,i} \end{aligned} \tag{2}$$

in which $\psi(q) \in \mathbb{R}[q^{-1}]$ are referred to as basis functions.

The scheme in Fig. 1 includes both feedback and feedforward control. The goal of the feedforward controller is to anticipate for known or repeating exogenous disturbances that act on the system, whereas the feedback controller attenuates unknown disturbances and model errors. In the remainder of this work, the focus is on feedforward control, i.e., optimization of $C_{ff}(\theta)$, since this is the main contribution to the tracking performance. The goal of the feedforward controller is defined as follows.

Definition 1 (Feedforward control goal) Determine a feedforward controller such that the output y_0 tracks the reference r, i.e., eliminate reference induced tracking errors $e = r - y_0$, defined as

$$e = S(C_r - PC_{ff})r \tag{3}$$

with $S = (I + PC_{fb})^{-1}$.

A well known result from classical feedforward control is that the plant inverse must be reflected in the feedforward controller [1], which is also the case for the proposed combined input shaper and feedforward controller. By satisfying

$$C_{ff}(q, \theta_a)C_r^{-1}(q, \theta_b) = P^{-1}(q, \theta_0) \implies e(t) = 0 \forall t \tag{4}$$

it follows that the reference induced positioning error is eliminated. This can also be obtained by setting $C_r = I$ and directly parameterizing C_{ff} as P^{-1}. However, for direct inversion non-minimum phase zeros and properness of the plant might lead to internal stability issues and non-causal feedforward control, see e.g., [4]. Direct inversion issues can be avoided by using an input shaper, however this is only applicable to point-to-point motion since C_r modifies the reference, see [12] for a detailed motivation.

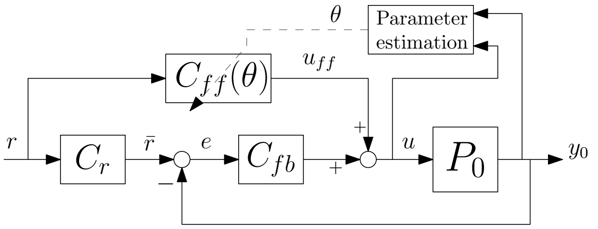

Fig. 1. General control scheme with feedback controller C_{fb}, feedforward controller C_{ff}, input shaper C_r and plant P_0.

Next, consider the following feedforward controller parameterization, directly establishes the connection between plant parameters and controller tuning knobs θ.

Definition 2 (Feedforward controller parameterization) The input shaper and feedforward controller are parameterized as follows.

$$\mathcal{C} = \left\{ (C_r, C_{ff}(\theta)) \middle| \begin{array}{l} C_{ff}(\theta) = A(q^{-1}, \theta) \\ C_r = B(q^{-1}) \end{array}, \theta \in \mathbb{R}^{n_a} \right\} \tag{5}$$

Where $C_{ff}(\theta)$ is parameterized as function of the to be optimized parameters θ. Next,

$$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \dots & \psi_{n_a} \end{bmatrix}^\top, \tag{6}$$

and to be estimated parameters

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_{n_a} \end{bmatrix}^\top \in \mathbb{R}^{n_a} \tag{7}$$

such that $C_{ff}(q^{-1}, \theta) = \Psi^\top \theta$.

Remark 1 In the present paper, the focus is on optimization of $C_{ff}(\theta)$ and C_r is assumed to be fixed as in Fig. 1, to avoid cumbersome notation. Conceptually, the following approach and analyses is similar for the general case, i.e., with $C_r(\theta)$. In the remainder of this work the notation θ refers to θ_a, similar for θ_0 referring to θ_a^0 and, $B(q^{-1}, \theta_b)$ is referred to as $B(q^{-1})$ assuming that θ_b is known.

Using the parameterization (5) the reference induced error becomes

$$e = S(q^{-1}) \left(B(q^{-1}) - \frac{B(q^{-1})}{A(q^{-1}, \theta^0)} A(q^{-1}, \theta) \right) r, \tag{8}$$

hence, if $\theta \to \theta^0$, then the positioning error $e \to 0$, such that (4) is satisfied.

B. Parameter optimization for feedforward control

In the remainder of this section, an optimization problem is formulated to optimize $C_{ff}(\theta)$. The optimization is based on linear least squares, for which analytic solutions are provided for both the batch-wise case as for real-time approaches.

Definition 3 (Parameter optimization problem) Given measurement data sequences $\{u\}$ and $\{y_0\}$, determine θ^* such that the feedforward controller (5) minimizes the reference induced tracking error.

Consider the following optimization problem for computation of θ^*.

$$\min_{\theta} V(C_{ff}(\theta)) \tag{9}$$

In which the objective function is of the form

$$V(\theta, k) = \frac{1}{2}\sum_{i=1}^{k} \epsilon^2(i) \tag{10}$$

with residual function $\epsilon(k)$ linear in the parameter θ and given by,

$$\epsilon(k) := \bar{u}(k) - \phi(k)\theta \tag{11}$$

where $\phi(k)$ is a filtered version of the measured output data using the basis functions,

$$\phi(k) = \Psi^{\top}(q)y(k) \tag{12}$$

and $\bar{u}(k)$ is a filtered version of the input data, i.e.

$$\bar{u}(k) = B(q)u(k) \tag{13}$$

as depicted in Fig. 2.

The objective function (10) is minimized if $\theta^* \to \theta_0$, as is shown by substituting $\phi(k)$ and $\bar{u}(k)$ in (11),

$$\begin{aligned} \epsilon(k) &= B(q)u(k) - \Psi^{\top}\theta y(k) && (14)\\ &= B(q)u(k) - A(q,\theta)y(k) && (15) \end{aligned}$$

combining this with (1) results in,

$$\epsilon(k,\theta)\big|_{\theta=\theta_0} = 0. \tag{16}$$

Hence the gradient of the cost function

$$\nabla V(\theta,k)\big|_{\theta=\theta_0} = \sum_{i=1}^{k} \epsilon(k,\theta)\frac{\partial\epsilon(k,\theta)}{\partial\theta}\bigg|_{\theta=\theta_0} = 0 \tag{17}$$

also becomes zero. This implies that the minimum of the cost function is obtained for $\theta^* \to \theta_0$.

In the following two subsection, analytic solutions to the optimization problem are derived for both a batch-wise setting as for an online optimization setting.

B.1) Batch-to-batch parameter optimization

In a batch-wise optimization setting, an experiment or task is performed from which a batch of data is collected [8]. The data collected during task nr j, denoted with $\{u^j\}$ and $\{y^j\}$ is used to optimize $C_{ff}(\theta)$ for the next task, i.e., determine θ^{j+1} and update the controller.

Next, it is shown that for batch-wise optimization, the optimization problem (9) is equivalent to a linear least squares problem. To this extend, the proposed residual function (11) is rewritten in matrix form using data from task j resulting in the following notation,

$$\mathcal{E} = \bar{U} - \Phi\theta \tag{18}$$

in which

$$\mathcal{E} = \begin{bmatrix}\epsilon(1)\\ \epsilon(2)\\ \vdots \\ \epsilon(k)\end{bmatrix}, \Phi = \begin{bmatrix}\phi(1)\\ \phi(2)\\ \vdots \\ \phi(k)\end{bmatrix}, \bar{U} = \begin{bmatrix}\bar{u}(1)\\ \bar{u}(2)\\ \vdots \\ \bar{u}(k)\end{bmatrix}$$

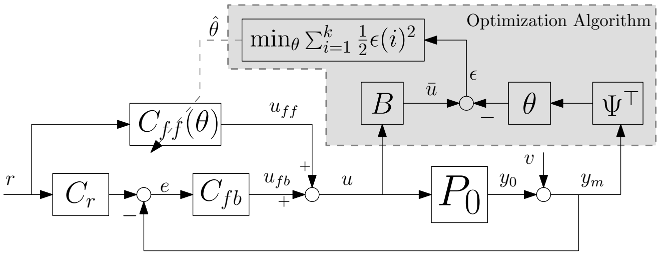

Fig. 2. Combined framework for feedforward control and parameter optimization.

such that the objective function is alternatively written as $V(\theta) = \frac{1}{2}\mathcal{E}^{\top}\mathcal{E}$, then the optimization problem (9) is equivalent to the least squares problem

$$\bar{U} = \Phi\theta \tag{19}$$

with analytic solution

$$\theta^* = (\Phi^{\top}\Phi)^{-1}\Phi^{\top}\bar{U}. \tag{20}$$

Assumption 1 Assume that $u(k)$ in open-loop or $r(k)$ in closed-loop is persistently exciting such that the matrix $\Phi^{\top}\Phi$ is non-singular.

To solve this problem in a batch-wise fashion, consider the following procedure.

Procedure 1 (Batch-to-batch parameter optimization)

1) Collect input-output data $\{u^j\}$ and $\{y^j\}$ from the current task
2) Determine $\bar{U}^j$ and Φ^j
3) Compute $\theta^{j+1} = \left((\Phi^j)^{\top}\Phi^j\right)^{-1}(\Phi^j)^{\top}\bar{U}^j$
4) Update the controller parameters, i.e., determine $C_{ff}^{j+1}(\theta^{j+1})$, and continue with step 1 for task $j+1$

B.2) Current-iteration parameter optimization

In the previous section, a batch-wise solution for the optimization problem (9) is outlined. A disadvantage of this method is that the controller can only be updated in-between tasks, i.e., performance improvement after each task. In this section, an online optimization is proposed based on recursive least squares (RLS), which allows to optimize the parameters efficiently during each iteration. Hence, enabling performance improvement during a task.

The recursive equivalent of the solution to (9) is given by

$$\theta(k) = \theta(k-1) + K(k)\left(\bar{u}(k) - \phi^{\top}\theta(k-1)\right) \tag{21}$$

in which the time dependent learning gain K is given by

$$K(k) = P(k)\phi(k) \tag{22}$$

and $P(k)$ is recursively computed as follows

$$P(k) = P(k-1)\left[I - \phi(k)\Sigma\phi(k)^{\top}P(k-1)\right] \tag{23}$$

where

$$\Sigma = \left(I + \phi(k)^{\top}P(k-1)\phi(k)\right)^{-1}. \tag{24}$$

with initial conditions $P(t_0) = \left(\Phi(t_0)^{\top}\Phi(t_0)\right)^{-1}$ and $\theta(t_0)$. A detailed derivation of the RLS algorithm can be found in [14, Chapter 2], [15]. Next, the following procedure is proposed for online optimization of the parameter using the RLS algorithm in (21) - (24)

Procedure 2 (Online parameter optimization)

1) Define an initial parameter estimate $\theta(t_0)$ and initial condition $P(t_0)$
2) At time k compute the learning gain $K(k)$ using (22) - (24)
3) Compute the parameters $\theta(k)$ using (21)
4) Update the controller $C_{ff}(\theta(k))$ using $\theta(k)$ and start at step 2 for the next iteration

To conclude this section, an optimization problem is proposed for feedforward controller tuning and solutions for both batch-to-batch and online settings have been provided. The main advantage of online optimization is that performance can directly be improved instead of improvement after each iteration as in a batch-wise setting. However, both methods rely on obtaining an unbiased estimate of the optimal tuning parameters in order to minimize the positioning error [16], as given in Definition 2. Therefore, in the next section a statistical analysis of the proposed optimization problem is provided in the presence of measurement noise.

III. Bias analysis in presence of noise

In this section, the estimator proposed in section II is further analyzed for a real life situation, i.e., where measurement noise is present in the signals that are used for parameter optimization. To further specify how noise effects these signals consider the following assumption.

Assumption 2 The input signal u is assumed to be noise free. The output y_0 is polluted with measurement noise v such that the measured output becomes

$$y_m(t) = y_0(t) + v(t), \tag{25}$$

Note that this is highly related to the errors-in-variables setting [17], where noise is present in both the measured signals as in the regressor signal.

Assumption 3 Measurement noise $v(t) \sim \mathcal{N}(0, \sigma_v^2)$ is assumed to be zero-mean white noise with independent and identically distributed samples (i.i.d.) and variance σ_v^2.

Next, the optimization problem (9), that has been shown to be equivalent to the least squares problem (19), is further analyzed in the presence of noise. Therefore, the optimal estimate θ^* that minimizes the cost function, i.e.,

$$\left.\frac{\partial V(\theta)}{\partial \theta}\right|_{\theta=\theta^*} = 0 \tag{26}$$

is further analyzed to provide insight in which cases a biased estimate is obtained. To compute the gradient of the objective function, consider the residual signal ϵ where y_0 is now replaced by y_m leading to (18) where the matrix $\Phi(k)$ is now composed from

$$\phi(k) = \Psi^\top(q) y_m(k). \tag{27}$$

Computing the gradient of $V(\theta)$ and equating this to zero

$$\frac{\partial V(\theta)}{\partial \theta} = -\Phi^\top \left(\bar{U} - \Phi\theta\right) = 0 \tag{28}$$

gives the optimal parameter estimate (20).

Before stating the main result in this work consider the following parameterization of the basis functions $\psi(q)$. The n^{th} basis function is of the form

$$\psi_n(q) = \sum_{j=0}^{n_\psi} a_n^j q^{-j} \tag{29}$$

where n_ψ is the highest order of all n_a basis functions and $a_n^j \in \mathbb{R}$ is the j^{th} coefficient corresponding to the n^{th} basis function.

Theorem 1 (Bias of the estimator) *The expected value of the optimal estimate θ^* as function of y_0 and v is given by*

$$\begin{aligned}\mathbb{E}\,\theta^* = \left[\left(\mathbf{R}_{yy}(k) + 2\mathbf{R}_{yv}(k) + \mathbf{R}_{vv}(k)\right)\Gamma_a\right]^{-1} \\ \cdot \left(\mathbf{R}_{yy}(k) + \mathbf{R}_{yv}(k)\right)\Gamma_a\theta_0\end{aligned} \tag{30}$$

where Γ_a is a matrix consisting of all coefficients corresponding to the n_a basis functions.

$$\Gamma_a = \begin{bmatrix} a_1^0 & \dots & a_{n_a}^0 \\ \vdots & & \vdots \\ a_1^{n_\psi} & \dots & a_{n_a}^{n_\psi}\end{bmatrix} \in \mathbb{R}^{n_\psi \times n_a} \tag{31}$$

Furthermore, the matrices $\mathbf{R}_{yy}(k) \in \mathbb{R}^{n_\psi \times n_\psi}$ and $\mathbf{R}_{vy}(k) \in \mathbb{R}^{n_\psi \times n_\psi}$ are estimates of the auto-correlation matrix and cross-correlation matrix on the basis of k data samples respectively. Also note that the autocorrelation matrix of the noise $\mathbf{R}_{vv}(k) = \sigma_v^2 I_{n_\psi}$ for $k \to \infty$.

The proof is omitted due to space restrictions.

In [17] it is shown that under mild conditions

$$\mathbf{R}_{yy} = \lim_{k\to\infty} \frac{1}{k}\mathbf{R}_{yy}(k) \quad \mathbf{R}_{vy} = \lim_{k\to\infty} \frac{1}{k}\mathbf{R}_{vy}(k), \tag{32}$$

this, together with Theorem 1 allows to further analyze the setting where input-output data is collected in a closed-loop fashion, and online estimation is used to determine the feedforward parameter as in Fig. 2, which is also exploited in [13]. First, note that

$$y_0 = SPC_{fb}(r - v) \tag{33}$$

implying that correlation is present between y_0 and v due to feedback. Next using Theorem 1 it becomes evident that the obtained estimates will be biased due to measurement noise, since both $\mathbf{R}_{vy} \neq 0$ and $\mathbf{R}_{vv} \neq 0$ resulting in $\mathbb{E}\,\theta^* \neq \theta_0$. This leads to severe performance degradation of not accounted for as will be shown in section IV.

In the following Corollaries, the ideal setting without noise and the case where input-output data is collected in open-loop with noise are considered.

Corollary 1 (Noise free optimization)
Consider the situation where measurement noise is not present, hence the noise related terms in (30) vanish resulting in

$$\mathbb{E}\,\theta^* = \left[\mathbf{R}_{yy}\Gamma_a\right]^{-1}\mathbf{R}_{yy}\Gamma_a\theta_0 = \theta_0. \tag{34}$$

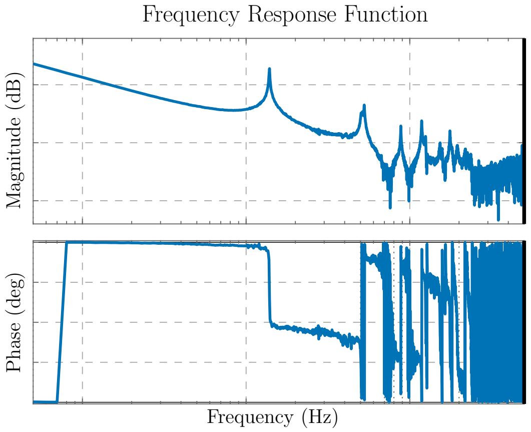

Fig. 3. Frequency response function of a typical motion system.

From which it follows that the estimate θ^* is an unbiased estimate of θ_0 if noise is not present.

Corollary 2 (Noisy optimization in open-loop) Consider the situation where noise is present and the input output data is collected in open-loop, i.e., without a feedback controller being present. This implies that v and y_0 are uncorrelated using (33), i.e., $\mathbf{R}_{vy} = 0$, therefore the estimate (30) simplifies to

$$\mathbb{E}\,\theta^* = \left[\left(\mathbf{R}_{yy} + \sigma_v^2 I_{n_\psi}\right)\Gamma_a\right]^{-1}\mathbf{R}_{yy}\Gamma_a\theta_0. \tag{35}$$

This shows that the estimate becomes biased $\mathbb{E}\,\theta^* \neq \theta_0$ in open-loop due to the effect of measurement noise.

From this analysis it follows that noise does influence the estimate in both the open-loop case as in a closed-loop case. In the following section an example is provided to illustrate the necessity of appropriately dealing with measurement noise in estimation problems to be able to successfully implement feedforward control.

IV. Illustrative Case Study

Typical motion systems including; wafer stages, printers and pick-and-place robot, are described by rigid-body behavior in addition to flexible modes [18].

$$P(s) = M^{-1}\frac{1}{s^2} + \sum_{i=1}^{\infty}\frac{(c_{mqi} + sc_{mvi})b_{mi}}{s^2 + 2\zeta_i\omega_i s + \omega_i^2} \tag{36}$$

An example of a typical frequency response function of such a motion system is depicted in Fig. 3, where a clear mass behavior is observed untill the first flexible mode. For controller design, the bandwidth is usually a factor 3 below the first flexible mode, i.e., where the rigid-body behavior is dominant. Hence, rigid-body dynamics are leading for performance, whereas flexible modes are taken into account for stability reasons.

A. *Plant and feedback controller*

In the remainder of this section, simulations are performed to confirm theoretical conclusions, i.e., illustrate the effect of measurement noise on the tracking performance. For the sake of illustration a single mass system $P(s) = 1/ms^2$ is used.

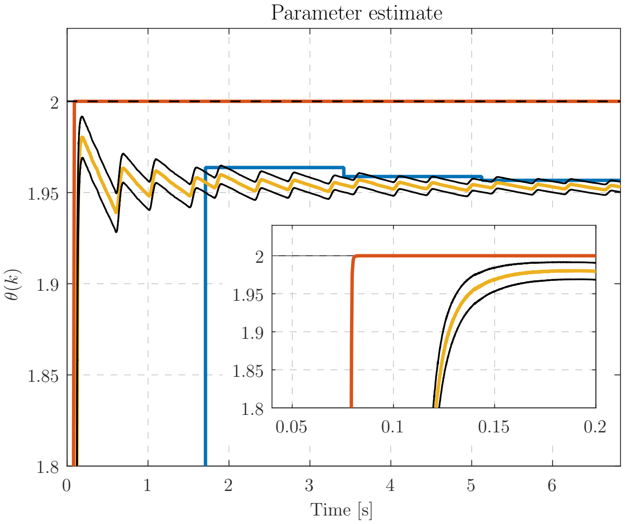

Fig. 4. Online parameter estimates, i.e., θ^{noise} with noise (–) with 99.7 % confidence interval (–), without noise $\theta^{\text{noise-free}}$ (–) and the true parameter θ_0 (- -). Batch-wise estimate is also provided (–). A zoom for $t \in [0.04, 0.2]$ shows the convergence of $\theta^{\text{noise-free}}$.

First, consider the following discretized transfer function of the plant,

$$P(z^{-1}) = \frac{T_s^2}{2}\frac{(z^{-1} + z^{-2})}{m_0(1 - 2z^{-1} + z^{-2})} \tag{37}$$

where $T_s = 0.001$ is the sample time and $m_0 = 2$ the mass. Second, a feedback controller is designed

$$C_{fb}(s) = 1.9 \cdot 10^8 \times \frac{s + 15.71}{s^2 + 1.3 \cdot 10^4 s + 3.2 \cdot 10^6} \tag{38}$$

realizing a bandwidth of 10 Hz. Where the bandwidth is defined as the cross-over frequency of the open-loop transfer function, denoted with $L = PC_{fb}$. Next, consider the feedforward parameterization for this example to obtain the setting in Fig. 2.

B. *Feedforward controller*

According to Definition 2, the input shaper is given by the plant nominator polynomial, i.e.,

$$C_r(q) = \frac{1}{2}(q^{-1} + q^{-2}) \tag{39}$$

and $C_{ff}(q,\theta) = \psi(q)\theta$ with basis function,

$$\psi(q) = \frac{1 - 2q^{-1} + q^{-2}}{T_s^2}. \tag{40}$$

that depends on the plant denominator polynomial. The goal is to optimize the feedforward controller parameter θ using the optimization problem (9), in which $\phi(k) = \Psi^\top y(k)$ where Ψ consists of the single basis function (40) in this example, i.e., $\phi(k) = \psi(q)y(k)$.

C. *Results and conclusions*

Next, the feedforward controller is optimized online using Procedure 1, and simulations are conducted with and without noise to shown the influence. Furthermore, batch-wise optimization is included to show the benefit of direct learning.

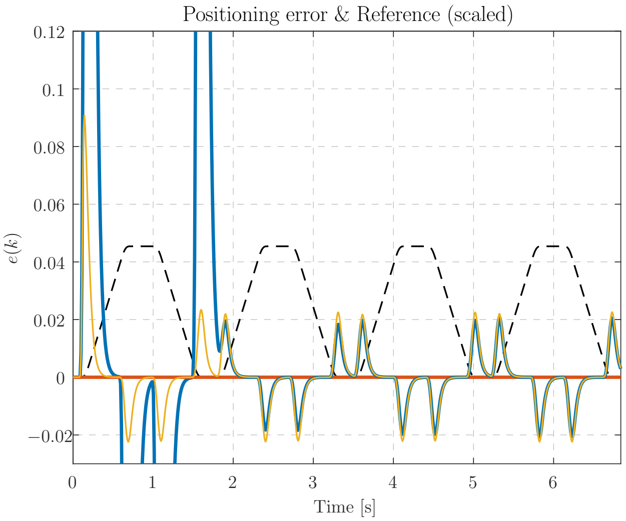

Fig. 5. Positioning error for the case; online learning with noise (–), online learning without noise (–) and batch-wise learning with noise (–). Furthermore, a scaled version of the reference (- -) is given.

The reference is takes as a sequence of fourth-order point-to-point tasks, see Fig. 5, and white noise is used with variance $\sigma_v = 10^{-13}$. The resulting parameter estimates are depicted in Fig. 4 and the obtained positioning error in Fig. 5.

From these results the following observations can be made.

- In the noise-free case, the feedforward parameter converges to the true mass of the system, i.e., $\theta^{\text{noise-free}} \rightarrow \theta_0$. This confirms that an unbiased estimate is obtained if measurement noise is not present, as shown in Corollary 1. Note that after a small time interval, (4) is satisfied since the positioning error converges to zero.
- In case that measurement noise is present, although the variance decreases, the parameter estimate does not converge to the true system parameter $\theta^{\text{noise}} \not\rightarrow \theta_0$, i.e., the obtained estimate is biased confirming the theoretical conclusions in Section III. It can be seen that due to the biased estimate the positioning error remains non-zero, i.e., (4) is not satisfied.
- The immediate benefit of direct learning is observed. For the batch-wise implementation a large error is present during the first task, whereas, for the online approach the estimate converges within a fraction of a task, see zoom in Fig. 4. Hence, the online approach is more robust towards variations in system parameters.

Finally, it can be concluded that the simulations confirm the theoretical conclusions in this paper.

V. CONCLUSIONS

In this paper, a unified framework is provided for automatic feedforward controller tuning. The framework exploits both batch-wise tuning as well as online tuning, leading to new opportunities and insights for feedforward control. A linear least squares optimization problem with an analytic solution is developed to optimize the feedforward controller parameters. Furthermore, a detailed statistical analysis is provided in the case where measurement noise is present, indicating that biased estimates are obtained in earlier approaches [13]. An example confirms the theoretical conclusions, i.e., indicating the possible hazard of using biased estimates for feedforward control resulting in a deterioration of the positioning performance. Finally, benefits of online learning for parameter varying systems are indicated.

REFERENCES

[1] P. Lambrechts, M. Boerlage, and M. Steinbuch, “Trajectory planning and feedforward design for electromechanical motion systems,” *Control Engineering Practice*, no. 13, pp. 145–157, 2005.

[2] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, June 2006.

[3] S. van der Meulen, R. Tousain, and O. Bosgra, “Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer.” *Journal of Dynamics Systems, Measurements and Control*, vol. 130, Sept. 2008.

[4] J. van Zundert and T. Oomen, “Inverting nonminimum-phase systems from the perspectives of feedforward and ILC.” *20th World Congress of the Int. Federation of Automatic Control*, pp. 12 607 – 12 612, 2017.

[5] X. Gao and S. Mishra, “An iterative learning control algorithm for portability between trajectories,” in *2014 American Control Conference*, June 2014, pp. 3808–3813.

[6] J. van de Wijdeven and O. Bosgra, “Using basis functions in iterative learning control: analysis and design theory,” *International Journal of Control*, vol. 83, no. 4, pp. 661–675, 2010.

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

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

[9] J. Bolder, J. van Zundert, S. Koekebakker, and T. Oomen, “Enhancing flatbed printer accuracy and throughput: Optimal rational feedforward controller tuning via iterative learning control,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4207–4216, May 2017.

[10] D. Hoelzle, A. Alleyne, and A. W. Johnson, “Basis task approach to iterative learning control with applications to micro-robotic deposition,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1138–1148, 2011.

[11] 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.

[12] F. Boeren, D. Bruijnen, N. van Dijk, and T. Oomen, “Joint input shaping and feedforward for point-to-point motion: Automated tuning for an industrial nanopositioning system.” *Mechatronics*, 2014.

[13] H. Butler, “Adaptive feedforward for a wafer stage in a lithographic tool,” *IEEE Transactions on Control Systems Technology*, 2012.

[14] K. J. Åström and B. Wittenmark, *Adaptive Control: Second Edition*. Dover Publications, INC. Mineola, New York, 2013.

[15] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[16] R. Bitmead and B. Anderson, “Performance of adaptive estimation algorithms in dependent random environments,” *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 788–794, 1980.

[17] T. Söderström and P. Stoica, *System Identification*. Prentive Hall, Hemel Hempstead UK, 1989.

[18] W. K. Gawronski, *Advanced Structural Dynamics & Active Control of Strucutres*, F. F. Ling, Ed. Springer, 2004, vol. 411 p.

[19] T. Söderström and P. Stoica, “Instrumental variable methods for system identification,” *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 1–9, 2002.

[20] M. Gilson, H. Garnier, P. C. Young, and P. M. Van den Hof, “Optimal instrumental variable method for closed-loop identification,” *IET control theory & appl.*, vol. 5, no. 10, pp. 1147–1154, 2011.

[21] L. Ljung and T. Söderström, *Theory and practice of recursive identification*. MIT press, 1983.

[22] A. Padilla, H. Garnier, P. Young, and J. Yuz, “Recursive online IV method for identification of continuous-time slowly time-varying models in closed loop,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4008–4013, 2017.