Second-order iterative learning control for scaled setpoints

Citation for published version (APA):
Best, de, J. J. T. H., Liu, L., Molengraft, van de, M. J. G., & Steinbuch, M. (2015). Second-order iterative learning control for scaled setpoints. *IEEE Transactions on Control Systems Technology*, *23*(2), 1063-6536. https://doi.org/10.1109/TCST.2014.2324178

DOI:
10.1109/TCST.2014.2324178

Document status and date:
Published: 01/01/2015

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Second-Order Iterative Learning Control for Scaled Setpoints

Jeroen de Best, Lancheng Liu, René van de Molengraft, and Maarten Steinbuch

Abstract—**Iterative learning control (ILC) is a control technique for systems subject to repetitive setpoints or disturbances. However, in many applications, the setpoint is not strictly repetitive, and the learning process should start all over from the beginning if the setpoint changes. In this brief, point-to-point movements with different magnitudes will be considered, which are constructed by scaling a nominal setpoint. Second-order ILC with an adaptive low-pass filter in the trial domain is used to accurately track these scale varying setpoints under the influence of disturbances that are either repetitive or experience the same scaling as the setpoint. Experiments have been carried out to validate the proposed method.**

Index Terms—**Iterative learning control (ILC), motion control.**

I. Introduction

In many manufacturing processes, production steps are carried out on repetitive structures. Examples of repetitive structures are given in Fig. 1. In many of these production steps, the tool is to be aligned with respect to a feature, perform its task, and move toward the next feature. Most conventional control approaches use a feedback controller for plant stabilization and disturbance rejection in combination with a feedforward controller according to a predefined structure (e.g., mass, damping, and coulomb friction) to increase the performance. However, there are limitations using this approach. For example, the closed-loop bandwidth can be limited by system dynamics, whereas the fixed-structure feedforward may not be able to capture the disturbances [1], [2]. For tracking a predefined setpoint over and over again, a control technique called iterative learning control (ILC) [1] can be applied. The ILC reduces the tracking error along a trajectory that is traced repeatedly by the iterative refinement of a feedforward signal. Good surveys of recent ILC research can be found in [3]–[6].

One constraint within ILC is that the setpoint needs to be repetitive every trial. In practice, however, the distance between consecutive features may vary (e.g., due to temperature changes). Hence, the setpoint to be tracked is not strictly repetitive but varies due to these variations. Applying ILC for varying setpoints is one of the challenges in current ILC research. Several methods have been developed to use the knowledge from previous ILC trials to construct feedforward signals for new, different setpoints.

In [7] and [8], time-frequency adaptive ILC is proposed, where different setpoints are generated using a constant velocity phase with variable length. More general trajectories are considered in [9] in which a finite impulse response mapping strategy is proposed based on converged learning forces obtained with learning control at a specific acceleration set-point profile. In [10], basis tasks are learned from which a reference trajectory can be constructed. Direct learning control (DLC) [11] and recursive DLC [12] are developed to generate the control signal for a new setpoint using prestored setpoints and control signals. The methods presented in [9]–[12] need the converged feedforward signals learned from specific setpoints to construct the new control input for a different setpoint. In this brief, we will present a second-order ILC (SOLIC) algorithm in which scale varying setpoints are applied during the learning process and for which the tracking error will be reduced iteratively.

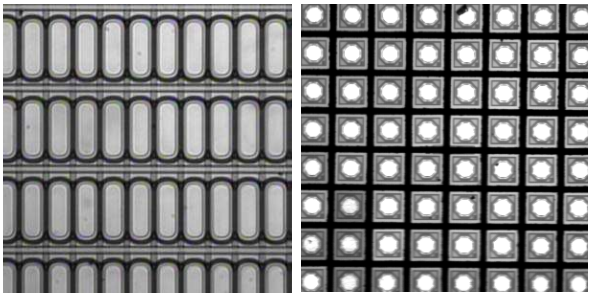

Fig. 1. Repetitive structures: OLED display and diodes on a wafer.

High-order ILC was studied in [13]–[20]. It is shown that high-order ILC is useful to increase the convergence speed [13], [16], [18], reject disturbances that satisfy an *a priori* relation from one trial to the next [19] and has robustness in the presence of external disturbances [18]. In this brief, SOILC will be used to iteratively identify different classes of disturbances, which are used for updating the feedforward signal.

This brief focuses on accurate tracking of setpoints, which are constructed by scaling a nominal setpoint. ILC is used to update the feedforward signal while during iterations scale varying setpoints are applied. We analyze the convergence of the tracking error for situations with and without disturbances. It is assumed that these disturbances are repetitive every trial and/or experience the same scaling as the setpoint. A SOILC strategy will be used to identify these two types of disturbances and compensate for them during iterations. The contributions of this brief are: 1) the design of a SOILC strategy to accurately track scale varying setpoints in which during the learning process these scale varying setpoints are applied and 2) will be implemented on an industrial setup to show the effectiveness.

The rest of this brief is organized as follows. In Section II, the standard ILC method is extended by implementing scaling only, leading to normalized ILC (NILC). SOILC will be derived in Section IV. In Section V, results of experiments will be given, where the different methods will be compared. Section VI will present the conclusions.

II. Standard ILC and NILC

In this section, we will briefly discuss the ILC working principle [21], [22]. To explain ILC, consider the block scheme given in Fig. 2 with a controller K and a plant G, both assumed to be discrete and linear time invariant. A schematic representation of the plant that

Manuscript received March 28, 2014; accepted May 5, 2014. Date of publication June 3, 2014; date of current version February 11, 2015. Manuscript received in final form May 12, 2014. This work was supported by SenterNovem-Innovatiegerichte Onderzoeksprogramma's IOP. Recommended by Associate Editor S. M. Savaresi.

The authors are with the Department of Mechanical Engineering, Control Systems Technology Group, Eindhoven University of Technology, Eindhoven 5600 MB, The Netherlands (e-mail: j.j.t.h.d.best@gmail.com; lancheng.liu@gmail.com; m.j.g.v.d.molengraft@tue.nl; m.steinbuch@tue.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2014.2324178

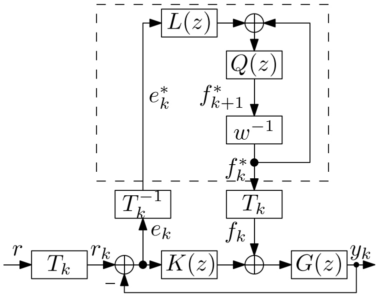

Fig. 2. (N)ILC scheme. Dashed box: offline updating of the feedforward signal.

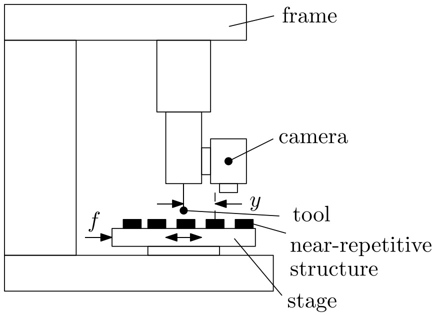

Fig. 3. Schematic representation of the plant. A near-repetitive structure is mounted on a stage driving by the force f. The goal is to position the tool with respect to the features of the near-repetitive structure. The relative position between the tool and the features y is measured via a camera.

is considered in this brief is given in Fig. 3. For now, we take the gain $T_k = 1$. The time shift operator in Fig. 2 is denoted by z, i.e., $z^{-1}x(t) = x(t-1)$, here t represents the sample number. The trial shift operator is denoted by w, i.e., $w^{-1}x_k(t) = x_{k-1}(t)$, where k represents the trial number [19], [20]. The repetitive setpoint is given by r, whereas the output at trial k is denoted by y_k. During trail k, the feedforward signal f_k is applied and the error e_k is measured. Offline, the error signal is filtered with the learning filter L and added to the feedforward f_k. This filter is chosen as an approximation of the inverse of the process sensitivity S_p and can be designed, for instance, using the zero-phase error tracking controller algorithm [23]. Next, the robustness filter Q is applied, which results in the feedforward f_{k+1} that is applied in the next trial $k+1$. The offline updating is mathematically written as

$$f_{k+1} = Q(f_k + Le_k). \tag{1}$$

The tracking error e in trial $k+1$ can be written as

$$e_{k+1} = Sr - S_p f_{k+1} \tag{2}$$

where $S = 1/(1+GK)$ is the sensitivity and $S_p = G/(1+GK)$ is the process sensitivity. Substitution of the update law (1) into (2) leads to

$$e_{k+1} = Sr - QS_p(f_k + Le_k). \tag{3}$$

Similar to (2), we use the fact that $S_p f_k = Sr - e_k$ and substitute this into (3) such that the error at trial $k+1$ becomes a function of the error in the previous trial k

$$e_{k+1} = Q(1 - LS_p)e_k + (1-Q)Sr. \tag{4}$$

The above system is called an linear iterative system for which convergence (under the assumption of infinite trajectory length) is obtained when [20]

$$\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty < 1 \quad \forall\omega \in [-\pi, \pi] \tag{5}$$

is satisfied.

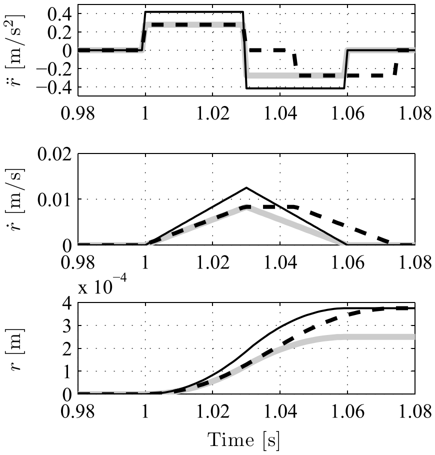

Fig. 4. Setpoint generation for different magnitudes. Gray line: nominal setpoint. Dashed line: stretched setpoint. Black line: scaled setpoint.

As opposed to the repetitive setpoint considered in standard ILC, point-to-point setpoints with varying travel distances will be considered here since small variations are present in the distance between successive features. An example of a nominal setpoint is given in gray in Fig. 4. In general, two ways of setpoint generation for different magnitudes are: 1) include a constant velocity part [7] (dashed line in Fig. 4) and 2) scale the acceleration profile [11], [12] (black line in Fig. 4). In this brief, we handle setpoint variation using the second type and scale a nominal setpoint r by a gain T_k, which results in a setpoint r_k that is used in the trial k, i.e., $r_k = T_k r$. The value of T_k is assumed to be bounded by $T_k \in [\underline{T}, \overline{T}]$, where $\underline{T}, \overline{T} \in \mathbb{R}^+$ are related to the pitch variation present in the repetitive structure. The scaling factor, T_k, can be determined *a priori*. The center of the camera is initially located above the center of a feature, whereas the next neighboring target feature is already in the field of view. From this, the distance between the features can be determined and T_k can be determined. We scale the setpoint such that the time to reach each target is the same. Moreover, the switching times for the acceleration in this case remain the same, such that we can exploit the use of scaling.

As standard ILC can only cope with a repetitive setpoint, the error will not converge if during iterations these scale varying setpoints r_k are applied. To handle scale varying setpoints, we proceed as follows. Standard ILC can be extended by incorporating the gain T_k before and after the ILC update block as depicted in Fig. 2, which will be referred to as NILC in the remainder of this brief. The learning update uses the normalized error $e_k^* = T_k^{-1}e_k$ to construct a normalized feedforward signal f_{k+1}^* as shown in the dashed area in Fig. 2, whereas the applied feedforward signal is given by $f_k = T_k f_k^*$. The same analysis done in (1)–(5) can be carried out to prove that the error is convergent. However, in case disturbances are present in the system, NILC is likely to fail due to the fact that disturbances will not scale in general.

III. Existence of Disturbances

The assumptions behind NILC are that the closed-loop system is LTI and that the error scales with the same gain as the setpoint, which in the presence of disturbances does not hold [24]. In this brief, three types of disturbances will be considered.

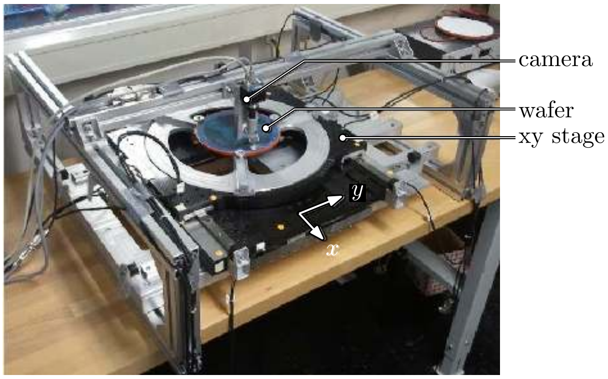

Fig. 5. Industrial application: an xy-wafer stage.

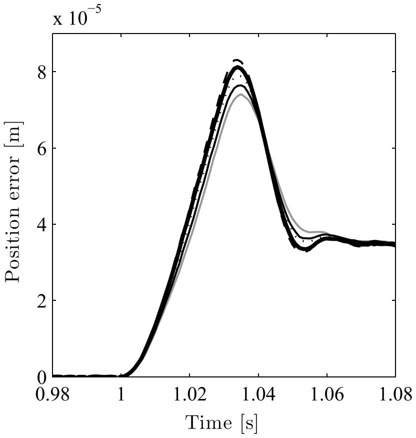

Fig. 6. Measured errors for different gains. Gray line: $T_1 = 0.9$. Black line: $T_2 = 0.95$. Dotted-black line: $T_3 = 1$. Bold: $T_4 = 1.05$. Dashed-black line: $T_5 = 1.1$.

1) Disturbances that experience the same scaling as the setpoints, i.e., $d_k = T_k d$. Unmodeled viscous damping in a mechanical motion system is an example of such a disturbance. Moreover, the reference itself can be seen as one.
2) Disturbances that are repetitive every trial, i.e., $d_{k+1} = d_k = \cdots = d$. These kind of disturbances are for example caused by an amplifier offset, gravity forces, or dry friction.
3) Disturbances that are of a random nature such as sensor noise.

The existence of disturbances is investigated in practice using the industrial application depicted in Fig. 5. The setup consists of an xy-stage carrying a wafer with discrete semiconductors. The stage is actuated by current controlled linear motors. For feedback, the position of the semiconductors is measured via a camera mounted above the stage at 1 kHz. A controller is tuned using classical loopshaping techniques [25] such that the closed-loop system has a bandwidth of 20 Hz (Section V).

Next, five experiments are conducted, in which each time the nominal setpoint of Fig. 4 is scaled by a different gain T_k and applied to the closed-loop control system while measuring the position error. The gains T_k that are used in the experiments are $T_1 = 0.9$, $T_2 = 0.95$, $T_3 = 1$, $T_4 = 1.05$, and $T_5 = 1.1$. The corresponding errors are given in Fig. 6. A first observation is that the five measured errors have a similar shape, but each with different amplitude. From scaling, the errors are expected to satisfy

$$\frac{e_i}{T_i} = \frac{e_j}{T_j}, \quad i, j \in \{1, \ldots, 5\}. \tag{6}$$

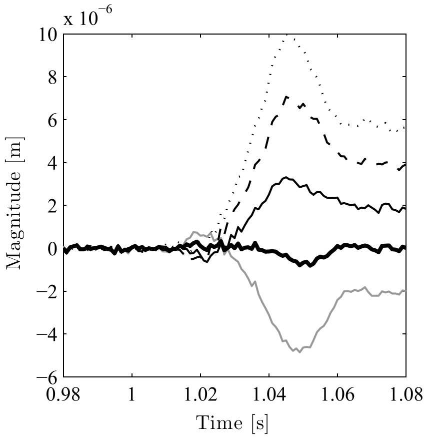

Fig. 7. Differences between measured error e_2 and its approximations. Gray line: $e_2 - (T_2/T_1)e_1$. Black line: $e_2 - (T_2/T_3)e_3$. Dashed-black line: $e_2 - (T_2/T_4)e_4$. Dotted-black line: $e_2 - (T_2/T_5)e_5$. Bold line: difference between measured error e_2 and approximation by combination of e_1 and e_3 given in (9).

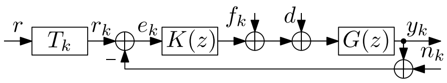

Fig. 8. Feedback control structure with a repetitive input disturbance d.

Using this, the error e_2 for example can be estimated from e_1 by $(T_2/T_1)e_1$. Similar estimates of e_2 can be made from e_3, e_4, and e_5. The differences of the measured error e_2 and these estimated errors are given in Fig. 7. It can be seen that the differences are not exactly zero, since disturbances of types 2 and 3 are present. Considering these types of disturbances, a much more accurate estimate of e_2 can be computed by taking combinations of errors. This is explained using Fig. 8 where, with $f_k = 0$ and $n_k = 0$, the errors are given by $e_i = T_i g + h, i \in \{1, \ldots, 5\}$, with $g = Sr$ and $h = -S_p d$. Note that g originates from a type 1 disturbance, which is the reference r in this case. The signal h originates from a type 2 disturbance, being the repetitive disturbance d. From two measurements, for example, e_1 and e_3, we can estimate g and h by

$$\begin{pmatrix} e_1 \\ e_3 \end{pmatrix} = \begin{pmatrix} T_1 & 1 \\ T_3 & 1 \end{pmatrix} \begin{pmatrix} g \\ h \end{pmatrix} \rightarrow \begin{pmatrix} \tilde{g} \\ \tilde{h} \end{pmatrix} = \begin{pmatrix} T_1 & 1 \\ T_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} e_1 \\ e_3 \end{pmatrix} \tag{7}$$

which results in

$$\tilde{g} = \frac{e_3 - e_1}{T_3 - T_1}, \quad \tilde{h} = \frac{T_3 e_1 - T_1 e_3}{T_3 - T_1}. \tag{8}$$

Using this, e_2 can now be estimated more accurately as

$$T_2 \tilde{g} + \tilde{h} = \frac{T_3 - T_2}{T_3 - T_1} e_1 + \frac{T_2 - T_1}{T_3 - T_1} e_3. \tag{9}$$

The difference of the measured error e_2 and this estimate is also given in Fig. 7 in bold. It can be seen that it is much more accurate than the scaled errors. The accuracy of this estimate is <1 μm, whereas others are as large as 10 μm. Similar results are obtained, when this estimate is constructed with other combinations of errors. In the remainder of this brief, this idea will be extended to ILC by learning the signals g and h, which will lead to SOILC.

IV. Second-Order Iterative Learning Control

In this section, first the principle of SOILC will be explained. Then, it will be analyzed under which conditions the proposed SOILC approach is convergent and what the influence of sensor noise and applying similar setpoints is. Finally, improvements will be presented by adding an adaptive low-pass filter in the trial domain for SOILC.

A. Principle of SOILC

Consider the control scheme in Fig. 8 at this moment without sensor noise $n_k = 0$, where the goal is to design a feedforward signal f_{k+1} in such a way that the error e_{k+1} in trial $k+1$ is zero. Assume we have two trials $k-1$ and k, for which the errors of these two trials can be written as

$$e_{k-1} = T_{k-1}g + h - S_p f_{k-1} \tag{10}$$

$$e_k = T_k g + h - S_p f_k. \tag{11}$$

After these two trials, the repetitive terms g and h can be estimated similar to (8)

$$\tilde{g} = \frac{e_{k-1} - e_k}{T_{k-1} - T_k} + \frac{S_p(f_{k-1} - f_k)}{T_{k-1} - T_k} \tag{12}$$

$$\tilde{h} = \frac{T_{k-1}e_k - T_k e_{k-1}}{T_{k-1} - T_k} + \frac{S_p(T_{k-1}f_k - T_k f_{k-1})}{T_{k-1} - T_k}. \tag{13}$$

Assume the feedforward signal for trial $k+1$ is f_{k+1}, then the error for trial $k+1$ can be estimated as

$$\begin{aligned} e_{k+1} &= T_{k+1}\tilde{g} + \tilde{h} - S_p f_{k+1} \\ &= (1-\alpha)e_{k-1} + \alpha e_k + S_p((1-\alpha)f_{k-1} + \alpha f_k) - S_p f_{k+1} \end{aligned} \tag{14}$$

with α defined as

$$\alpha = \frac{T_{k-1} - T_{k+1}}{T_{k-1} - T_k}, \quad T_{k-1} \neq T_k. \tag{15}$$

Since we want to design a feedforward signal f_{k+1} in such a way that $e_{k+1} = 0$, from (14) we solve f_{k+1}

$$f_{k+1} = (1-\alpha)f_{k-1} + \alpha f_k + \frac{1}{S_p}((1-\alpha)e_{k-1} + \alpha e_k). \tag{16}$$

As in standard ILC, the inverse of the process sensitivity S_p is approximated by L and a robustness filter Q can be added (Section II), leading to the second-order update law

$$f_{k+1} = Q((1-\alpha)f_{k-1} + \alpha f_k + L((1-\alpha)e_{k-1} + \alpha e_k)). \tag{17}$$

Update law (17) can be used for the situation, where types 1 and 2 disturbances are present in the system. However, at this point, three questions remain to be answered. First, is the new linear iterative system convergent? Second, what happens in case $T_k = T_{k-1}$? Third, how does SOILC deal with type 3 disturbances?

B. Analysis of SOILC

In this section, the SOILC approach is analyzed with respect to the three previous mentioned questions.

1) Convergence: From (10) and (11), we obtain

$$S_p f_{k-1} = T_{k-1}g + h - e_{k-1} \tag{18}$$

$$S_p f_k = T_k g + h - e_k. \tag{19}$$

In (14), substitute f_{k+1} by (17) and together with (18) and (19), we obtain

$$e_{k+1} = Q(1 - LS_p)(\alpha e_k + (1-\alpha)e_{k-1}) + (1-Q)(T_{k+1}g + h). \tag{20}$$

To analyze the convergence of error, the system is constructed as a linear iterative system. From (20), it can be seen that e_{k+1} is related to e_k and e_{k-1}. We define $\underline{x}_k = (e_k e_{k-1})^T$ and $\underline{u}_k = T_{k+1}g + h$ such that

$$\underline{x}_{k+1} = A\underline{x}_k + B\underline{u}_k \tag{21}$$

with

$$A = \begin{pmatrix} a_1 & a_2 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1-Q \\ 0 \end{pmatrix} \tag{22}$$

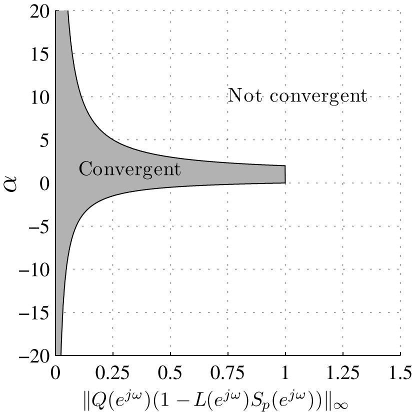

Fig. 9. Computed α values based on $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty$ that make the system convergent.

where $a_1 = Q(1-LS_p)\alpha$ and $a_2 = Q(1-LS_p)(1-\alpha)$. Convergence of this linear iterative system is assessed in the frequency domain using the work of [20]. Transforming the linear iterative system to the frequency domain leads to

$$\underline{X}_{k+1}(\omega) = A(e^{j\omega})\underline{X}_k(\omega) + B(e^{j\omega})\underline{U}_k(\omega) \tag{23}$$

where the signals $\underline{x}_k$ and u_k are transformed to the frequency domain using

$$X(\omega) = \sum_{l=0}^{\infty} x(l)e^{-j\omega l}. \tag{24}$$

Convergence is now obtained if

$$\overline{\rho} = \sup_{\omega \in [0,\pi]} \rho(A(e^{j\omega})) < 1 \tag{25}$$

with $\rho(A(e^{j\omega}))$ denoting the spectral radius of $A(e^{j\omega})$ defined as

$$\rho(A(e^{j\omega})) = \max_{i=\{1,2\}} |\lambda_i(A(e^{j\omega}))|. \tag{26}$$

The eigenvalues of the matrix $A(e^{j\omega})$ are given by

$$\lambda_{1,2}(e^{j\omega}) = \frac{a_1(e^{j\omega}) \pm \sqrt{a_1(e^{j\omega})^2 + 4a_2(e^{j\omega})}}{2}. \tag{27}$$

Therefore, convergence of the linear iterative system (21) is guaranteed if the condition

$$\left\| \frac{a_1(e^{j\omega}) \pm \sqrt{a_1(e^{j\omega})^2 + 4a_2(e^{j\omega})}}{2} \right\|_\infty < 1 \quad \forall\omega \tag{28}$$

is satisfied. Since the phase of $L(e^{j\omega})S_p(e^{j\omega})$ is zero for all frequencies and $Q(e^{j\omega})$ is a zero-phase low-pass filter, $a_1(e^{j\omega})$ and $a_2(e^{j\omega})$ are real-valued, $\forall\omega$. It will be shown that there is a tradeoff between the designed Q and L filter and the maximum allowable values of α, which guarantee convergence. In Fig. 9, the gray area indicates the allowable values of α for different values of $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty$. From this figure, we have the following observations:

1) for convergence, $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty \leq 1$ is necessary;
2) if $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty = 1$, then $0 \leq \alpha \leq 2$; however, from from the definition of α, it is possible that α is negative depending on the values of the gains;
3) the smaller the value of $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty$, the larger the range of possible values of α.

Note that the presented convergence analysis is very strict, since it is expected that the error in the next trial is always smaller than the current error, irrespective of the applied gain with which the setpoint is scaled. However, if the setpoint in the next trial is obtained by scaling with a larger gain than the current setpoint, the error in the next trial is also expected to be larger than the current error. As a result, it is harder to always obtain a smaller error in the next trial than the error in the current trial.

2) Perfect Pitch: If in the previous two trials, the gains T_k and T_{k-1} are the same, by definition, the value of $\alpha = \pm\infty$. Hence, if $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty > 0$, then convergence can not be guaranteed. This is caused by the fact that we can not estimate g and h after two trials with the same setpoint. In fact, after having designed the learning filter L and the low-pass filter Q, the quantity $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty$ can be calculated. Using this result in combination with Fig. 9, the allowable values of α can be determined. Improvements regarding this issue will be discussed Section IV-C.

3) Type 3 Disturbances: If T_{k-1} is close to T_k, then type 3 disturbances highly affect the estimations of g and h. This is explained as follows. By considering sensor noise n_k in Fig. 8, the errors e_{k-1} and e_k can be written as

$$e_{k-1} = T_{k-1}g + h - S_p f_{k-1} - S n_{k-1} \tag{29}$$

$$e_k = T_k g + h - S_p f_k - S n_k. \tag{30}$$

Substitution of (29) and (30) into (12) and (13) leads to

$$\tilde{g} = g - \underbrace{S\frac{n_{k-1} - n_k}{T_{k-1} - T_k}}_{\text{estimation error}},\ \tilde{h} = h - \underbrace{S\frac{T_{k-1}n_k - T_k n_{k-1}}{T_{k-1} - T_k}}_{\text{estimation error}}. \tag{31}$$

Therefore, if T_k is close to T_{k-1}, there will be large estimation errors, since the noise terms are amplified. In this brief, we will deal with this sensor noise by iteratively estimating g and h such that these noise terms will not be amplified. This is done by introducing an adaptive low-pass filter in the trial domain on the estimates of g and h.

C. Improving SOILC

In this section, we will improve the principle of SOILC with respect to: 1) sensor noise and 2) for cases in which $T_k = T_{k-1}$. By introducing an adaptive low-pass filter in the trial domain, SOILC is first made less sensitive to sensor noise. Incorporating sensor noise n_k in the previous analysis leads to the update law

$$\begin{aligned} f_{k+1} = Q((1-\alpha)f_{k-1} + \alpha f_k + L((1-\alpha)e_{k-1} + \alpha e_k \\ + S(1-\alpha)n_{k-1} + S\alpha n_k)). \end{aligned} \tag{32}$$

If the previous two gains, T_k and T_{k-1} are close to each other, the absolute value of α can be large. Hence, the sensor noise is amplified by a large gain and becomes part of the next feedforward signal f_{k+1}, which is not desired and may cause a large error.

In the trial domain, the random type 3 disturbances, such as sensor noise, are changing from trial to trial, whereas the repetitive type 2 disturbances remain the same. Therefore, the sensor noise can be seen as a high-frequency signal in the trial domain, whereas the repetitive disturbances can be seen as a low-frequency signal in the trial domain [26]. This implies that we can use a low-pass filter in the trial domain to reject the sensor noise. We use SOILC with an adaptive low-pass filter in the trial domain to smoothen out the sensor noise. In this way, the estimations of g and h are obtained iteratively and filtered and are then used in the generation of the new feedforward signal f_{k+1}. Define the terms g and h as the true values and $\tilde{g}_k$ and $\tilde{h}_k$ to represent the corresponding estimations after trial k

$$\tilde{g}_k = \frac{e_{k-1} - e_k}{T_{k-1} - T_k} + S_p \frac{f_{k-1} - f_k}{T_{k-1} - T_k} \tag{33}$$

$$\tilde{h}_k = \frac{T_{k-1}e_k - T_k e_{k-1}}{T_{k-1} - T_k} + S_p \frac{T_{k-1}f_k - T_k f_{k-1}}{T_{k-1} - T_k}. \tag{34}$$

The first-order adaptive low-pass filters in the trial domain are chosen as

$$\hat{g}_{k+1} = (1 - \gamma_k)\hat{g}_k + \gamma_k \tilde{g}_k \tag{35}$$

$$\hat{h}_{k+1} = (1 - \gamma_k)\hat{h}_k + \gamma_k \tilde{h}_k \tag{36}$$

where $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$ are the low-pass filtered outputs of the estimations, which are going to be used in the update law. The value of γ_k can be tuned to give a weighting on how much the current estimates of g and h are used for the construction of the new feedforward signal. For stability of these low-pass filters, it is required that $0 \leq \gamma_k \leq 1$. To prevent sensor noise amplification when $T_k \approx T_{k-1}$, we choose γ_k as $\gamma_k = \beta\ |T_{k-1} - T_k|$, where now the scalar β should satisfy $0 \leq \beta \leq 1/\overline{T} - \underline{T}$. In such way, we cancel out the denominator $(T_{k-1} - T_k)$ in (31), so only $\beta\ S(n_{k-1} - n_k)$ and $\beta\ S(T_{k-1}n_k - T_k n_{k-1})$ are considered in the estimations of g and h. A tradeoff is present in this case between convergence speed and sensitivity to random disturbances, as is also discussed in [27]. A larger value of β results in faster convergence but results in a system that is more sensitive to noise and vice versa.

In case $T_{k-1} = T_k$, the value of γ_k becomes zero. As a result, the estimates $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$ are not updated and equal the previous ones $\hat{g}_k$ and $\hat{h}_k$ such that no learning is performed. For the cases where $T_{k-1} = T_k$ and learning is to be performed in trial $k+1$ with $T_{k+1} = T_k$, two options are considered.

1) In case $T_{k-1} = T_k$, a standard ILC update could be applied for trial $k+1$, which updates the feedforward signal and decreases the next tracking error. The update law is in that case given by $f_{k+1} = Q(f_k + Le_k)$. A disadvantage, however, is that the estimates $\hat{g}$ and $\hat{h}$ are not updated while standard ILC is applied. Whenever future gain values differ from T_k such that SOILC can be applied again, the error might increase significantly, since old and possibly nonconverged values of $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$ are used.

2) In order to keep learning and update $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$ while $T_{k+1} = T_k$, we propose the following. Instead of using information of the previous two trials k and $k-1$ to update $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$ in SOILC, we use the information of the previous trial k and trial $k-p$, where $p \geq 1$ is the smallest number for which $T_{k-p} \neq T_k$ to update $\hat{g}$ and $\hat{h}$. Therefore, the update can be written as

$$\hat{g}_{k+1} = (1 - \gamma_k)\hat{g}_k + \gamma_k \tilde{g}_k \tag{37}$$

$$\hat{h}_{k+1} = (1 - \gamma_k)\hat{h}_k + \gamma_k \tilde{h}_k \tag{38}$$

with now $\gamma_k = |T_{k-p} - T_k|$ and

$$\tilde{g}_k = \frac{e_{k-p} - e_k}{T_{k-p} - T_k} + S_p \frac{f_{k-p} - f_k}{T_{k-p} - T_k} \tag{39}$$

$$\tilde{h}_k = \frac{T_{k-p}e_k - T_k e_{k-p}}{T_{k-p} - T_k} + S_p \frac{T_{k-p}f_k - T_k f_{k-p}}{T_{k-p} - T_k}. \tag{40}$$

As an example, consider the gains depicted in Fig. 10. The gains are different for each iteration except for iteration five to eight, where the gains are the same. Hence, for $k = 6$, 7, and 8 the value $T_{k-1} - T_k = 0$. Therefore for iteration $k = 6$, 7 and 8, the values of p in (39) and (40) are 2, 3, and 4, respectively. For iteration $k = 2$, 3, 4, 5, 9, and 10, the value $T_{k-1} - T_k \neq 0$. Therefore, the value of p in

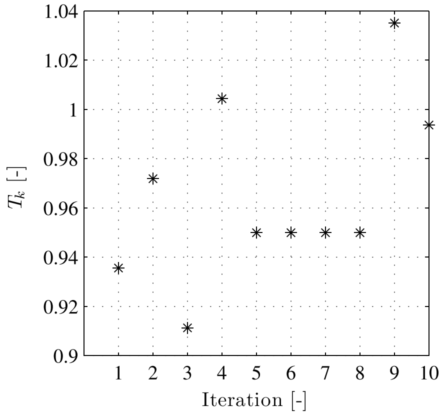

Fig. 10. Example of possible gains. The gains in iterations five to eight are the same, whereas others are different each trial.

(39) and (40) is taken as one, such that it results in its original form of (33) and (34).

The error in trial $k+1$ can now be estimated as

$$e_{k+1} = T_{k+1}\hat{g}_{k+1} + \hat{h}_{k+1} - S_p f_{k+1}. \tag{41}$$

Since the goal is to design a feedforward signal f_{k+1} such that $e_{k+1} = 0$, we derive the new update law

$$f_{k+1} = L(T_{k+1}\hat{g}_{k+1} + \hat{h}_{k+1}). \tag{42}$$

By successive substitution and with the definition of γ_k, we can write $\hat{g}_{k+1}$ as

$$\begin{aligned}\hat{g}_{k+1} = &\ (1 - \beta\ |T_{k-1} - T_k|)\hat{g}_k + \beta\ \text{sgn}(T_{k-1} - T_k)\\ &\times(e_{k-1} - e_k + T_{k-1}\hat{g}_{k-1} + \hat{h}_{k-1} - T_k\hat{g}_k - \hat{h}_k).\end{aligned}$$

Similarly

$$\begin{aligned}\hat{h}_{k+1} = &\ (1-\beta\ |T_{k-1}-T_k|)\hat{h}_k + \beta\ \text{sgn}(T_{k-1}-T_k)(T_{k-1}e_k\\ &-T_k e_{k-1} + T_{k-1}(T_k\hat{g}_k+\hat{h}_k) - T_k(T_{k-1}\hat{g}_{k-1}+\hat{h}_{k-1})).\end{aligned}$$

To filter out the high-frequency components in the measured error, we still use a zero-phase low-pass filter Q as the robustness filter after the updated estimations $\hat{g}_{k+1}$ and $\hat{h}_{k+1}$. Therefore, the first-order low-pass filters in the trial domain (35) and (36) now also include the low-pass filtering in frequency domain, i.e., the new filters now are

$$\hat{g}_{k+1} = Q((1-\gamma_k)\hat{g}_k + \gamma_k\tilde{g}_k) \tag{43}$$

$$\hat{h}_{k+1} = Q((1-\gamma_k)\hat{h}_k + \gamma_k\tilde{h}_k). \tag{44}$$

V. Results

In this section, the performance of standard ILC, NILC, SOILC, and SOILC with an adaptive low-pass filter in the trial domain for scale varying setpoints will be compared. The proposed methods are validated on a *xy*-wafer stage, where the task is to move from one feature to the next (Fig. 5). The frequency response function (FRF) from the input of the motor to the position output is given in Fig. 11. The plant is modeled by a mass-damper system with delay. The obtained model given by

$$G = 1\times10^{-4} \cdot \frac{1\times10^{-3}}{z-1} \cdot \frac{1.34z^2+5.14z+1.23}{z^2(z-0.85)} \tag{45}$$

is also shown in Fig. 11 by its FRF and shows a good match until ~60 Hz. As a consequence, a mismatch between the measured process sensitivity and its model is expected after 60 Hz. Therefore, the Q filter gets a cutoff frequency of 50 Hz. A controller K is tuned which consists of a lead filter with a zero at 6 Hz and a pole at 100 Hz and a second-order low-pass filter with a cutoff frequency of 250 Hz and a damping of 0.6. Finally, a notch is added at 80 Hz. The discrete controller is given by

$$K = 1\times10^4 \cdot \frac{3.3z^5-2.3z^4-7.8z^3+11z^2-3.5z-0.82}{z^5-2.4z^4+2.4z^3-1.2z^2+0.38z-0.070}. \tag{46}$$

The number of iterations that will be performed is 30. For the sake of comparison, the arbitrary gains are chosen the same for the four methods. The bounds are given by $\underline{T} = 0.75$ and $\overline{T} = 1.25$. Fig. 12 shows the applied gains. The proposed methods are applied on the *xy*-wafer stage with the value of β chosen as 1 in this case, such that $0 \leq \beta \leq 1/\overline{T} - \underline{T} = 2$ is satisfied. The maximum errors for each iteration are given in Fig. 13. Since standard ILC does not incorporate the scaling of the setpoint, it is expected that the final error oscillates depending on the applied gains, which also shows in Fig. 13. NILC does incorporate the scaling of the setpoint, however, it is based on the absence of type 2 disturbances. Dry friction is one of the major disturbances present in the experimental setup. It can be seen that if two successive gains are quite different the error of NILC increases, which is caused by the inappropriate scaling of the error. To *a priori*, determine the performance for standard ILC versus NILC for the case given in Fig. 8 consider the standard ILC case where: 1) a fixed disturbance d is present and 2) a setpoint r is applied with (for now) a fixed gain, i.e., T_1. The error of each iteration is given by $e_k = T_1 S r - S_p d - S_p f_k$. After convergence

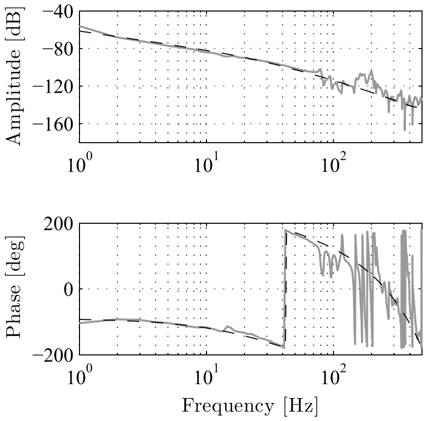

Fig. 11. Measurement FRF and the corresponding fit of the plant: measurement data (gray line), fitted model (dashed black line).

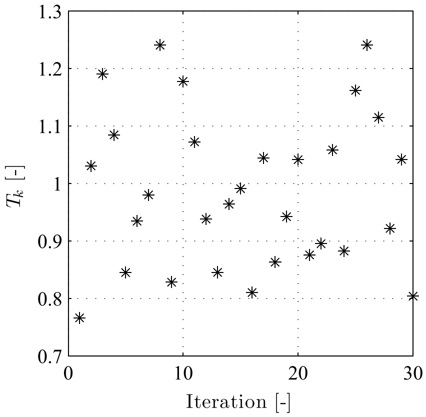

Fig. 12. Gains applied during iterations on the *xy*-wafer stage.

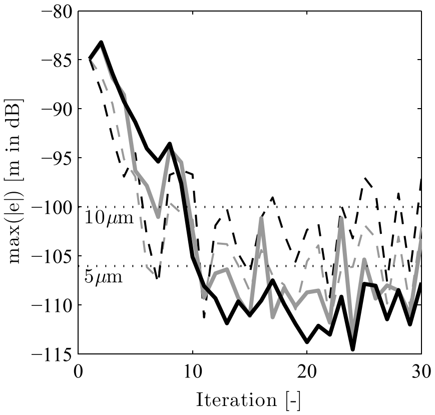

Fig. 13. Maximum absolute error in meters. Dashed-gray line: standard ILC. Dashed-black line: NILC. Gray line: SOILC. Bold line: SOILC with an adaptive low-pass filter in the trial domain.

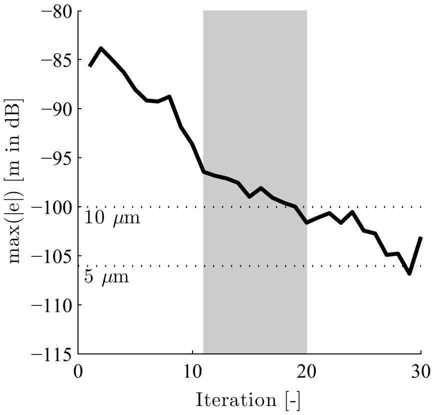

Fig. 14. Maximum absolute error in meters. Bold: SOILC with an adaptive low-pass filter in the trial domain with $\beta = 0.5$. The gains of iteration 11 through 20 (the hatched area) are the same but learning is still present.

($k \to \infty$), the error has approached zero and the feedforward signal is given by $f_\infty = S_p^{-1} T_1 S r - d$. If now a differt gain, say T_2, would be applied to the setpoint then for standard ILC together with the converged feedforward signal f_∞, the error would be $e = T_2 S r - S_p d - S_p f_\infty = T_2 S r - S_p d - S_p(S_p^{-1} T_1 S r - d) = S(T_2 - T_1) r$. On the other hand if NILC would have been applied, then the error would be $e = T_2 S r - S_p d - S_p(T_2/T_1) f_\infty = T_2 S r - S_p d - S_p(T_2/T_1)(S_p^{-1} T_1 S r - d) = S_p((T_2/T_1) - 1) d = S(T_2 - T_1)(G/T_1) d$. In both cases, non zero errors appear. Which one is the least directly depends on the plant G and (the frequency content of) r and d, where the latter one requires additional system knowledge in order to *a priori* compare the performance between standard ILC and NILC. If the frequency content of $(G/T_1)d$ is smaller than the frequency content of r, then NILC is more favorable than standard ILC and vice versa.

SOILC without low-pass filters in the trial domain suffers from the fact that α will become large if $T_{k-1} - T_k$ is small, resulting in an increase of the errors due to sensor noise amplification. This is the case for iteration 16, where the feedforward update (17) is dependent on iterations 14 and 15, which are close to each other (Fig. 12). The same reasoning holds for iteration 23. The most satisfactory results are obtained using SOILC with an adaptive low-pass filter in the trial domain. After 11 iterations, the error is converged to maximum errors of <5 μm. Overall, we see that SOILC has a slower convergence rate in the first iterations. Note, that in this brief, the initial conditions of the feedforward signals f, see Fig. 8, are taken as zero for all experiments. At the first iteration the estimators $\hat{g}$ and $\hat{h}$ for SOILC are zero as well. As opposed to standard ILC and NILC, the SOILC uses a second-order iteration scheme. Therefore, the signals $\hat{g}$ and $\hat{h}$, on which the feedforward depends, can only be estimated after two iterations. So, in fact the first applied feedforward making use of the identified signals $\hat{g}$ and $\hat{h}$ is at the third iteration. This already explains the difference in convergence at the beginning of the iteration process.

For industrial applicability, a better initial condition could be estimated to have a faster convergence. First, the iteration process could be started with a classical tuned feedforward signal f, where typically mass, damping, and dry friction are included [2]. Second, regarding the signal g, we recognize that the setpoint contribution is captured within the signal. So in fact, a better initial estimate for g is advised as Sr, with S being the sensitivity and r the setpoint (Fig. 8). Similarly, for the signal h a better initial estimate for h is advised as $-S_p d$, with S_p being the process sensitivity and d the disturbance (Fig. 8). However, the signal d is to be known in that case, which requires system knowledge, i.e., the specific nature of what is captured within the signal d should be known. The proposed solution in case $T_{k-1} = T_k$ in SOILC with adaptive low-pass filtering in the trial domain is investigated next. During the iterations, the applied gains in this case are the same as in Fig. 12, except that the gains of iterations 11 through 20 are kept the same in this case and equal to the gain of iteration 11. Furthermore, the value of β is taken as 0.5, such that learning is slower. If $\beta = 1$, we saw in the previous results that the error and therefore also the estimates $\hat{g}$ and $\hat{h}$ already converged within 11 iterations. The effect that we want to visualize here is that learning is still present even when the gain values of two successive iterations are the same. The results are given in Fig. 14. The gray hatched area indicates that the gains are the same for these iterations. A first observation is that the error converges slower. This was expected due to the lower value of β. Second, the error converges even when the gains of iterations 11 through 20 are the same. Furthermore, after iteration 20, when the gains deviate again, learning is still present as can be seen by the further reduction of the error. This result shows that the proposed solution in the case where two successive gains are the same is effective.

VI. Conclusion

Three methods, NILC, SOILC, and SOILC with an adaptive low-pass filter in the trial domain have been investigated to handle scale varying setpoints in ILC. Experiments are carried out to validate these methods. NILC achieves a good performance when there is no disturbance at all. SOILC is sensitive to nonrepetitive noise when the previous applied setpoints are almost the same. SOILC with an adaptive low-pass filter in the trial domain can handle the situation when both repetitive disturbances and nonrepetitive noise exist and achieves a good performance. After convergence, the error is reduced to <5 μm. The investigated methods consider disturbances that experience the same scaling as the setpoint, and trial-independent repetitive disturbances. Another class of disturbances are position-dependent disturbances, e.g., cogging. This kind of disturbances cannot be handled in this brief, since scaling cannot be applied. This will, therefore, be subject for future research.

References

[1] K. L. Moore, *Iterative Learning Control for Deterministic Systems*. New York, NY, USA: Springer-Verlag, 1993.

[2] S. Van der Meulen, R. L. Tousain, and O. H. Bosgra, “Fixed structure feedforward controller design exploiting iterative trials: Applied to a wafer stage and a desktop printer,” *ASME J. Dyn. Syst., Meas., Control*, vol. 130, no. 5, pp. 051006-1–051006-16, 2008.

[3] K. L. Moore, "Iterative learning control: An expository overview," *Appl. Comput. Control, Signals Circuits*, vol. 1, no. 1, pp. 151–214, 2008.

[4] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[5] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[6] K. L. Moore, Y. Q. Chen, and H. S. Ahn, "Iterative learning control: A tutorial and big picture view," in *Proc. 45th IEEE Conf. Decision Control*, Dec. 2006, pp. 2352–2357.

[7] I. Rotariu, R. Ellenbroek, G. van Baars, and M. Steinbuch, "Iterative learning control for variable setpoints, applied to a motion system," in *Proc. IEEE Eur. Control Conf.*, May 2003, pp. 1–8.

[8] I. Rotariu, M. Steinbuch, and R. Ellenbroek, "Adaptive iterative learning control for high precision motion systems," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 5, pp. 1075–1082, Sep. 2008.

[9] M. F. Heertjes and M. J. G. Van de Molengraft, "Set-point variation in learning schemes with applications to wafer scanners," *Control Eng. Pract.*, vol. 17, no. 3, pp. 345–356, Mar. 2009.

[10] D. Hoelzle, A. Alleyne, and A. J. W. Johnson, "Basis task approach to iterative learning control with applications to micro-robotic deposition," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 5, pp. 1138–1148, Sep. 2011.

[11] J. X. Xu, "Direct learning of control efforts for trajectories with different magnitude scales," *Automatica*, vol. 33, no. 12, pp. 2191–2195, 1997.

[12] J. X. Xu, J. Xu, and B. Viswanathan, "Recursive direct learning of control efforts for trajectories with different magnitude scales," *Asian J. Control*, vol. 4, no. 1, pp. 49–59, 2002.

[13] Z. Bien and K. M. Huh, "Higher-order iterative learning control algorithm," *IEE Proc. D Control Theory Appl.*, vol. 136, no. 3, pp. 105–112, May 1989.

[14] Y. Chen, C. Wen, and M. Sun, "A robust high-order p-type iterative learning controller using current iteration tracking error," *Int. J. Control*, vol. 68, no. 2, pp. 331–342, 1997.

[15] K. M. Huh, "A study on the robustness of the higher-order iterative learning control algorithm," in *Proc. Asian Control Conf.*, 1997.

[16] Y. Chen, M. Sun, B. Huang, and H. Dou, "Robust higher order repetitive learning control algorithm for tracking control of delayed repetitive systems," in *Proc. 31st IEEE Conf. Decision Control*, vol. 3. Dec. 1992, pp. 2504–2510.

[17] Y. Chen, Z. Gong, and C. Wen, "Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays," *Automatica*, vol. 34, no. 3, pp. 345–353, 1998.

[18] Y. T. Kim, H. Lee, H. S. Noh, and Z. Bien, "Robust higher-order iterative learning control for a class of nonlinear discrete-time systems," in *Proc. IEEE Conf. Syst., Man, Cybern.*, vol. 3. Oct. 2003, pp. 2219–2224.

[19] K. L. Moore and Y. Q. Chen, "On monotonic convergence of high order iterative learning update laws," in *Proc. 15th IFAC Congr. Invited Session High-Order Iterative Learn. Control*, 2002, pp. 1–6.

[20] M. Norrlöf, "Iterative learning control: Analysis, design, and experiments," Ph.D. dissertation, Linköpings University, Linköping, Sweden, 2000.

[21] R. Merry, R. van de Molengraft, and M. Steinbuch, "Iterative learning control with wavelet filtering," *Int. J. Robust Nonlinear Control*, vol. 18, no. 10, pp. 1052–1071, 2008.

[22] M. Steinbuch and M. J. G. van de Molengraft, "Iterative learning control of industrial motion systems," in *Proc. 1st IFAC Conf. Mechatron. Syst.*, Darmstadt, Germany, 2000, pp. 967–972.

[23] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *ASME J. Dyn. Syst., Meas., Control*, vol. 109, no. 1, pp. 65–68, 1987.

[24] R. Merry, R. van de Molengraft, and M. Steinbuch, "The influence of disturbances in iterative learning control," in *Proc. IEEE Conf. Control Appl.*, Aug. 2005, pp. 974–979.

[25] G. F. Franklin, J. D. Powell, A. Emami-Naeini, and J. D. Powell, *Feedback Control of Dynamic Systems*, vol. 3. Reading, MA, USA: Addison-Wesley, 1994.

[26] B. E. Helfrich *et al.*, "Combined H∞-feedback control and iterative learning control design with application to nanopositioning systems," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 336–351, Mar. 2010.

[27] D. Bristow, "Frequency domain analysis and design of iterative learning control for systems with stochastic disturbances," in *Proc. IEEE Amer. Control Conf.*, Jun. 2008, pp. 3901–3907.