Setpoint variation in iterative learning schemes

Baggen, M.C.J.; Heertjes, M.F.; van de Molengraft, M.J.G.

Published: 01/01/2006

Document Version

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

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):


General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 12. Dec. 2018
Abstract—Iterative Learning Control (ILC) in motion control systems is often hampered by the fact that variations in the desired setpoint trajectory are not accounted for. When the setpoint changes, different dynamics are excited causing the learned signal to be less effective in reducing the tracking error. In this paper, two methods to deal with setpoint variation are discussed. In the first method, the learned feedforward signal is decomposed in different force tables corresponding to particular parts of the setpoint. The second method utilizes the learned data from different setpoints to create a finite impulse response (FIR) mapping between setpoint and feedforward input. The potential of both methods is demonstrated through experiment on a lithographic motion system.

Index Terms—Feedforward, Iterative Learning Control, Setpoint Tracking Control, Finite Impulse Response.

I. INTRODUCTION

Motion control systems performing repetitive tasks are frequently encountered in industry. Examples of such systems are pick and place modules in component mounting machines, moving stages in lithographic machinery or welding robots in assembly lines. This class of systems generally requires high position accuracy combined with short settling times. Being exposed to large acceleration levels, a combined feedforward and feedback design then becomes necessary to achieve performance.

Iterative Learning Control (ILC) [1],[2],[3] provides the means to improve tracking accuracy of systems under repetitive motion. Numerous ILC algorithms have been proposed since the early publication of Arimoto et al. [4]. The initial frequency-domain approach (also known as classical ILC) evolved to time-domain approaches which are nowadays often presented in so-called 'lifted' system description [5], [6], [7], [8]. Either linear or nonlinear, a stable ILC scheme produces a feedforward signal that compensates for repetitive error. Key is the fact that the underlying setpoint is strictly repeatable. Herein small variations induce different tracking errors, which makes the previously learned signal less effective in reducing such error, i.e., the learning process has to resume.

To deal with setpoint variation several approaches are known from literature. Re-using learned information in case of specific setpoint variation with varying time- and amplitude scales is discussed in [9], [10]. The learning process becomes an explicit function (e.g. B-splines, neural networks) of the setpoint signal in [11], [12]. In this paper we adopt the perspective of setpoint commonalities. Based on third-order setpoint profiles, where the acceleration and jerk (derivative of acceleration) levels are fixed, a setpoint can be expressed as a superposition of different jerk phases with varying time shifts. To account for the time shifts, two strategies are employed: a multi-table approach and a finite impulse response (FIR) mapping. In the multi-table approach a feedforward signal from a previously learned table is used to construct a feedforward signal applicable to any setpoint from the considered class of setpoint profiles. Compared to [13] this requires only one force table. Hence multi-table merely reflects the fact that a single table can be activated multiple times. Alternatively, the FIR approach is an extension of the work presented in [14]. Learned data obtained during various repetitive tasks is used to train a FIR mapping between the setpoint signal and this data. Two feedforward FIR filters are obtained, one based on position setpoint signals and the other based on acceleration setpoint signals. Key to both approaches is the fact that learning is only applied in as a calibration of the tables and filters respectively. Once calibrated, the system performance is determined by the worst trial during subsequent processing steps. The main purpose of this paper is to investigate whether an industrial lithographic motion system can benefit from the considered approaches.

The paper is organized as follows. In Section II the considered motion system is described. Next a brief description of the used learning algorithm is given in Section III. The multi-table approach is presented in Section IV, followed by the FIR approach in section V. Both methods are validated and compared in experiment. In Section VI the main conclusions are summarized.

II. MOTION CONTROL SYSTEMS

In the production of IC's a wafer scanner is used to print a pattern on a silicon disk called wafer. A schematic architecture of a wafer scanner is shown in Fig. 1. Light passes through a mask (reticle) and lens onto the wafer. Both reticle and wafer are positioned by accurately controlled stages that perform scanning moves. By means of example we will restrict ourselves to a single stage of the dual stage reticle stage dynamics. A typical control structure for such a stage is depicted in Fig. 2. Herein the tracking error \( e \) is defined by the difference between the setpoint \( r \) and the measured output position \( y \). The error is fed into a PID-based feedback controller \( C \). The feedforward controller \( F \) exploits the a-priori knowledge of the setpoint \( r \) and the plant \( P \) to create an additional input \( u_f \) after which the total input \( u \) is used to control the plant \( P \). For reasons of simplicity the controlled stage dynamics are considered in a Single-Input Single-Output (SISO) framework.
A. Motion systems

The considered mechanics is described by double integrator behavior over a large frequency range, an example of which is shown in Fig. 3. Besides the double integrator behavior of the rigid body dynamics, several resonance modes can be recognized in the high-frequency range. Additionally the phase characteristics show a time delay resulting from computation time and zero-order-hold effects.

B. Feedback control design

To control the mechanics, the feedback controller $C$ is based on a series connection of a PID-controller, a notch filter and a low-pass filter. A model in the Laplace domain corresponding to a continuous-time model is given by the following transfer function:

$$C(s) = PID(s) N(s) LP(s),$$  \hspace{1cm} (1)

where $s$ represents the Laplace variable. The PID part is defined by:

$$PID(s) = \frac{k_p (s^2 + \omega_d s + \omega_i \omega_d)}{s},$$  \hspace{1cm} (2)

where $\omega_i$ is the cut-off frequency of the integral action, $\omega_d$ is the cut-off frequency of the differential action, and $k_p$ is a static gain. The second-order notch filter has the following structure:

$$N(s) = \left(\frac{\omega_p^2}{\omega_z^2}\right) \frac{\omega_z^2 s^2 + 2 \beta_z \omega_z s + \omega_z^2}{s^2 + 2 \beta_p \omega_p s + \omega_p^2},$$  \hspace{1cm} (3)

Herein $\omega_z$ and $\omega_p$ represent the zero and pole corner frequencies, $\beta_z$ and $\beta_p$ the corresponding damping coefficients. The second-order low-pass filter is defined as:

$$LP(s) = \frac{\omega_{lp}^2}{s^2 + 2 \beta_{lp} \omega_{lp} s + \omega_{lp}^2},$$  \hspace{1cm} (4)

where $\omega_{lp}$ is the cut-off frequency of the low-pass filter and $\beta_{lp}$ is its dimensionless damping coefficient. On the basis of a sampling frequency of $f_s = 5$ kHz, a discrete-time equivalent of the controller is implemented. The controlled behavior of the mechanics (Fig. 3) with the controller parameters from Table I is represented by the frequency response of the open-loop characteristics $OL(s) = C(s) P(s)$ as shown in Fig. 4. Here a bandwidth of approximately 250 Hz along with a phase margin of 30 degrees and a gain margin of -5 dB can be distinguished.
TABLE I

<table>
<thead>
<tr>
<th>filter parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{TD} )</td>
<td>( k_p ) 1.9 \cdot 10^{-6} \text{ [N]}</td>
</tr>
<tr>
<td>( \omega_0 )</td>
<td>2\pi \cdot 130 \text{ [rad/s]}</td>
</tr>
<tr>
<td>( \omega_1 )</td>
<td>2\pi \cdot 63 \text{ [rad/s]}</td>
</tr>
<tr>
<td>( \omega_{fr} )</td>
<td>2\pi \cdot 1450 \text{ [rad/s]}</td>
</tr>
<tr>
<td>( \beta_0 )</td>
<td>0.65</td>
</tr>
<tr>
<td>( \omega_s )</td>
<td>2\pi \cdot 543 \text{ [rad/s]}</td>
</tr>
<tr>
<td>( \beta_s )</td>
<td>10^{-2} [-]</td>
</tr>
<tr>
<td>( \omega_{fr} )</td>
<td>2\pi \cdot 543 \text{ [rad/s]}</td>
</tr>
<tr>
<td>( \beta_{fr} )</td>
<td>5 \cdot 10^{-2} [-]</td>
</tr>
</tbody>
</table>

C. Feedforward control

Given the feedback and feedforward connection as shown in Fig. 2, the closed loop transfer function between \( r \) and \( e \) is given by:

\[
e (s) \quad r (s) = \frac{1-P (s) F (s)}{1+P (s) C (s)}
\]

From it can be concluded that an ideal feedforward controller \( F \) equals the inverse of the plant \( P \). Unfortunately \( P \) does not necessarily induce a stable inverse, for example because of non-minimum phase behavior. Since the system under consideration exhibits dominant rigid-body behavior over a large range of frequencies, however a straightforward approach is to invert the double-integrator part giving an acceleration feedforward, i.e.

\[
F (s) = \hat{m} s^2,
\]

where \( \hat{m} \) is the estimated plant mass. Though giving significant performance improvement, acceleration feedforward still tends to limit the achievable performance for the system at hand. Discontinuities in the acceleration setpoint profile excite higher-order dynamics which hamper performance for example in terms of settling times.

III. ITERATIVE LEARNING CONTROL

Iterative learning control is used to improve upon the feedforward contribution (\( f_{fr} \) in Fig. 2) in particular in terms of settling times. Typically a learning scheme computes a force update for the next trial based on the tracking error and input of the previous trial. Here a trial represents a duty cycle exhibiting a sufficient amount of repetitiveness. The tracking performance is improved through repetition of trials (learning). A learning scheme as described in for example [15], [6] is presented in Fig. 5 where \( e^k \in \mathbb{R}^{n \times 1} \) represents an \( n \)-sample error column of the \( k \)-th iteration. The learned force is represented by \( f_{fr}^k \in \mathbb{R}^{n \times 1} \). The learned force, \( \hat{m} \) a one-trial delay operator in the trial domain, \( I \in \mathbb{R}^{n \times n} \) a unitary matrix and \( \mathbf{P}_s \in \mathbb{R}^{n \times n} \) a matrix representing the closed-loop process sensitivity dynamics \( \mathbf{P}_s \), or in transfer function notation:

\[
\mathbf{P}_s (s) = \frac{y (s)}{f_{fr} (s)} = \frac{P (s)}{1+C (s)P (s)},
\]

Matrix \( \mathbf{P}_s \) has a Toeplitz structure, or

\[
\mathbf{P}_s = \begin{pmatrix}
    h_0 & 0 & \cdots & 0 \\
    h_1 & h_0 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    h_n & h_{n-1} & \cdots & h_0
\end{pmatrix},
\]

where the parameters \( h_0, h_1, \ldots, h_n \) are the so-called Markov parameters of the impulse response. Similarly a Toeplitz matrix \( \mathbf{S} \in \mathbb{R}^{n \times n} \) can be derived that represents the closed-loop sensitivity dynamics \( \mathbf{S} \), which is defined in transfer function notation by

\[
\mathbf{S} (s) = \frac{e (s)}{r (s)} = \frac{1}{1+C (s) P (s)}.
\]

To obtain the Markov parameters in (8) impulse response measurements are performed. The controlled system is subjected to a 100 N force pulse. The average output response of 100 measurements scaled with a factor 1/100 is shown in Fig. 6. Here the Markov parameters are indicated by the data points (dots) at the sampling instants. Furthermore in Fig. 5, the vectors \( \mathbf{r} \in \mathbb{R}^{n \times 1} \) and \( \mathbf{d}^k \in \mathbb{R}^{n \times 1} \) contain the trial-invariant setpoint signal and an output disturbance signal, which may consist of a trial-invariant and trial-variant part respectively. On the basis of a pseudo-inverse [16] a stable estimate of the learning matrix \( \mathbf{L} \) is given by:

\[
\mathbf{L} = (\mathbf{P}_s^T \mathbf{P}_s + \beta I)^{-1} \mathbf{P}_s^T,
\]

where \( \beta \) represents a tuning parameter needed to balance the trade-off between convergence and noise amplification. Note that if \( \mathbf{P}_s \) is invertible and \( \beta = 0 \) it follows that \( \mathbf{L} = \mathbf{P}_s^{-1} \).

Stability of the learning scheme can by studied using the evolution of the error and force signals:

\[
e^k = \mathbf{S} (r - \mathbf{d}^k) - \mathbf{P}_s \cdot f_{fr}^k,
\]

Fig. 5. Representation of an ILC control scheme in lifted system description.

Fig. 6. Average measured impulse response.
Fig. 7. Time-series using ILC. Top: initial error signal with only acceleration feedforward (black) and converged error signal after 20 iterations (grey); Scaled acceleration profile (dashed); Bottom: converging rms error values against trial number.

Fig. 8. Setpoint profiles; a jerk phase is indicated by the grey area. Table profile \( r_{\text{table}} \) (black), overall profile \( r \) (dashed).

In this Section a multi-table approach is considered as a means to deal with the considered class of setpoint variation.

### IV. Multi-Table

#### A. Principle

The multi-table approach is motivated by the fact that the considered set of setpoint profiles consist of fixed building blocks in terms of jerk phases, i.e., the jerk level and its length are fixed. Assuming linear time-invariant (LTI) behavior, whereas the system is initially at rest, the superposition principle now states that when an input can be decomposed as a sum of signals, the overall system response can be expressed as the sum of corresponding responses. Under these conditions it is sufficient to learn only one force \( f_{\text{table}} \) with ILC corresponding to the setpoint \( r_{\text{table}} \) of only one jerk table.

Commonly used setpoint profiles describe varying point-to-point motions in industrial practice. Herein a third-order setpoint is often encountered which is constructed from a superposition of different time-shifted jerk tables. These tables contain jerk values for each sampling moment which are given either by a zero or by the maximum positive or negative jerk level. The length of a jerk phase is fixed resulting in more or less constant frequency contents induced by the setpoint signal throughout the set of considered setpoint variations. A typical setpoint profile is shown in Fig. 8. Here it can be seen that the position and velocity signals are smooth, although the acceleration and jerk signals are not. Note that acceleration, velocity, and position signals are obtained by integration.

For an arbitrary setpoint profile in the considered class of profiles the overall feedforward signal can be constructed from superposition of time shifted tables:

\[
\begin{align*}
    r &= \sum_{i=1}^{k} \alpha_i r_{\text{table}} (t - \tau_i) \Rightarrow f_{\text{ff}} &= \sum_{i=1}^{k} \alpha_i f_{\text{table}} (t - \tau_i). 
\end{align*}
\]  

Herein \( \tau_i \) represents a time shift and \( \alpha_i \in \{+1, -1\} \) a sign correction according to the jerk table at hand. Note that it is
assumed that the setpoint \( r \) is the largest source of tracking error. A possible implementation of the multi-table approach is shown in Fig. 9. Each time a new jerk phase starts, a new force table is initiated. Different from the approach in [13], where the displacement length is varied by inserting zeros in between learned signals, here only one learned table corresponding to the first jerk phase is needed using a proper superposition argument.

![Fig. 9. Control scheme with multi-table approach.](image)

**B. Experimental verification**

An experimental verification is performed with the high-performance motion system as described in Section II. The learning controller described in Section III is first used to learn a force \( f_{ilc} \) at a fixed setpoint. Measured results are shown in Fig. 10. Here it can be seen that learning is most important around the jerk phases where unwanted vibrations excited by the setpoint must be suppressed. The learned force signal corresponding to the first jerk phase starting at \( t = 0 \) s can only be obtained in a truncated form. Velocity restrictions limit the maximum acceleration time, i.e., a second jerk phase has to follow the first one at \( t = 0.03 \) s. The first jerk phase induces a level of constant acceleration from which the effect is shown in Fig. 10. The learned force contains an offset proportional to the acceleration signal, which corresponds to a mismatch in acceleration feedforward. After the first jerk phase (indicated by the grey area in Fig. 8) the truncated signal \( f_{ilc} \) corresponding to the first jerk phase is extended with this constant offset value to approximate \( f_{table} \).

The performance of the multi-table approach is evaluated at a different setpoint then the one used for learning \( f_{table} \). Measured results are shown in Fig. 11 where a force is constructed from \( f_{table} \) (black) according to (15). As a reference a learned force at the new setpoint is also depicted (grey). The multi-table approach appears to be capable of reducing the tracking error and settling time, although the performance of the newly learned ILC signal cannot be met. Nonlinear behavior like amplifier hysteresis, position-dependent dynamics and nonlinear actuator characteristics result in different dynamics at the subsequent jerk phases. In this regards the application of two tables may be considered: one for positive- and one for negative acceleration. Note that during the first jerk phase there is a very good match between \( f_{table} \) and \( f_{ilc} \). This could be expected, since \( f_{table} \) is derived during this jerk phase.

![Fig. 10. Time-series using ILC; Top: initial error (black), error after learning (grey). Bottom: Learned force \( f_{ilc} \) (grey), Force table corresponding to first jerk phase (black); Scaled acceleration profile (dashed).](image)

![Fig. 11. Time-series multi-table approach; Top: initial error, error with multi-table approach, error with ILC. Bottom: Learned force \( f_{ilc} \) (grey), Multi-table force \( f_{m} \) (black), Scaled acceleration profile (dashed).](image)

**V. FIR MAPPING**

A second approach to account for setpoint variations without further learning is to map the relation between setpoint \( r \) and learned force \( f_{ilc} \) through a finite impulse response model, see also [14]. As a result a filter is obtained that not only allows for application at the learned setpoint, but which is also applicable for other setpoints.

**A. Principle**

To describe the relation between setpoint \( r \) and \( f_{ilc} \), a discrete-time finite impulse response (FIR) mapping is used,
which is given by the following difference equation:

\[ f_{\text{ilc},k} = g_{-n_1} r_{k+n_1} + g_{-n_1+1} r_{k+n_1-1} + \ldots + g_{n_2} r_{k-n_2+1}. \]  

Equation (16)

Note that a non-causal mapping is obtained since \( f_{\text{ilc},k} \) can be a function of \( r_{k+n_1} \) future setpoint samples. Eq. (16) can be written as:

\[ f_{\text{ilc},k} = \sum_{i=-n_1}^{n_2} g_i \cdot r_{k-i}. \]  

The complete set of equations can now be written in matrix form:

\[
\begin{pmatrix}
  f_{\text{ilc},k_1} \\
  \vdots \\
  f_{\text{ilc},k_2}
\end{pmatrix} = \begin{pmatrix}
  r_{k_1+n_1} & r_{k_1-n_2} \\
  \vdots & \vdots \\
  r_{k_2+n_1} & r_{k_2-n_2}
\end{pmatrix} \begin{pmatrix}
  g_{-n_1} \\
  \vdots \\
  g_{n_2}
\end{pmatrix},
\]  

where the setpoint \( r \) is used to fill a matrix \( R \in \mathbb{R}^{k_2-k_1 \times n_2+n_2} \), having a Toeplitz structure. The vector \( g \in \mathbb{R}^{n_1+n_2+1} \) contains the FIR filter coefficients \( g_i \) that need to be determined. Furthermore \( n = n_1+n_2+1 \) is the order of the filter, \( n_1 \geq 0 \) is the look-ahead horizon, \( n_2 \geq 0 \) is the lookbehind horizon, \( k = k_2-k_1 \) is the data length of \( f_{\text{ilc}} \). The filter coefficients \( g_i, i = -n_1, \ldots, n_2 \) are obtained through a least squares fit on the \( \{f_{\text{ilc}}, r\} \) data set obtained from subsequent ILC trials. Under the assumption of LTI behavior, a sufficiently exciting training set of \( r \) is chosen such that the resulting FIR filter is applicable to an arbitrary \( r \) in the considered class of setpoint profiles. 

A Least Squares approximation is obtained by solving the set of normal equations:

\[ R^T R \cdot g = R^T f_{\text{ilc}}, \]  

which can be seen as the projection of \( f_{\text{ilc}} \) on the column space of \( R \). The solution of (19) satisfies the least squares solution in the sense that the induced matrix norm \( \| R \cdot g - f_{\text{ilc}} \|_2 \) is minimized. If \( R^T R \) is non-singular then (19) has a unique least squares solution given by:

\[ g = (R^T R)^{-1} R^T f_{\text{ilc}}. \]  

To use the complete data set \( \{f_{\text{ilc}}, r\} \) the following is proposed. The first force signal value \( f_{\text{ilc},k_1} \) is a function of \( n_2 \) samples of setpoint values \( r_{k-n_2} \) from the past which are generally not known. Zeros are inserted for the unknown signal values \( r \), i.e. filling up the upper-right triangular part of \( R \). Once the FIR coefficients \( g \) are determined, a relation between setpoint \( r \) and feedforward force \( f_{\text{ilc}} \) is found which is applicable at other setpoints. It is important to realize that \( k \gg n \) such that apart from the setpoint variation advantage, the FIR filter approach induces a significant data reduction step which is advantageous from an implementation point of view. The obtained filter \( F_{\text{fir}}(g) \) is implemented parallel to the existing feedforward path as shown in Fig. 12. To gain more insight in its structure, a transfer function in the \( z \)-domain can be derived:

\[ F_{\text{fir}}(z) = g_{-n_1} z^{-n_1} + g_{-n_1+1} z^{-n_1-1} + \ldots + g_{n_2} z^{-n_2+1}. \]  

Experiments are performed with the high-performance motion system such as described in Section II. Given the class of setpoints, three sets are used for training while yet another setpoint is used for qualification as shown in Table II. To evaluate performance, feedforward forces are learned using ILC for different training setpoints. When the FIR filter performance is evaluated at the same setpoint at which it is trained, a \( k^{th} \) order filter can reproduce the \( k \) data samples of the learned force in an exact manner. This is not desirable because any noise or position-dependent behavior contained in the measured signals is entirely incorporated in the FIR filter (overfitting). To avoid
overfitting, a cross validation is performed for increasing filter order to test whether the filter can describe feedforward for setpoints other than the training set. The filter is evaluated at the 'qualification' setpoint which is shown in Fig. 14. For the case at hand an optimum appears to occur around the relatively low filter-order of approximately \( n = 20 \). The corresponding filter coefficients \( g_i \) are plotted in Fig. 15. Here the sum of the filter coefficients \( \sum g_i = 3.5 \) equals the DC-gain of the filter. The look-ahead horizon \( n_1 = 5 \) is chosen empirically, which seems enough to compensate delays and non-minimum phase behavior in the controlled plant dynamics. Typically this allows the filter to compute derivatives without any phase shift.

Frequency response characteristics of the filter can be studied by substituting \( z = e^{j2\pi f/f_s} \) in the transfer function (21). The frequency response is evaluated on the interval \( 0 \leq f \leq f_s/2 \). The corresponding frequency domain characteristics of \( F_{\text{fir}} \) are shown in Bode representation in Fig. 16. Clearly inverse plant behavior can be recognized (compare this figure with Fig. 3). The sum of both filters \( F_{\text{fir}} \) and the standard feedforward controller \( F \) should approximate the inverse of the plant \( P \). The filter performance is evaluated at the 'qualification setpoint'. Resulting tracking errors after application of the FIR filter are plotted in Fig. 17. By means of reference, both the initial tracking error and the tracking error after learning are depicted. Again improvement is obtained compared to the initial tracking error, but the results are not satisfactory. The FIR filter trained with position setpoint signals has difficulties approximating the dominant peaks related to the snap (second derivative of acceleration). The slope of the force signal in the constant velocity region (indicated by the dotted line in Fig. 17) corresponds to a stiffness compensation of \( \Delta f_{\text{fir}} \approx 3.5 [N/m] \). This value equals the filter DC-gain. To improve upon performance, a design is proposed incorporating two FIR filters in parallel: one based on the position setpoint signal \( r \), and one based on the acceleration setpoint signal \( a \). This gives:

\[
f_{\text{ILC}} = R \cdot g, \tag{22}
\]
where:
\[
R = \begin{pmatrix}
  r_{k1+n1} & r_{k1-n2} & a_{k1+n1} & a_{k1-n2} \\
  \vdots & \vdots & \vdots & \vdots \\
  r_{k2+n1} & r_{k1+n1} & a_{k1+n1} & a_{k1-n2} \\
  \vdots & \vdots & \vdots & \vdots 
\end{pmatrix}
\]
\[
f_{\text{ile}} = \left( \begin{array}{c}
  f_{\text{ile},k1} \\
  \vdots \\
  f_{\text{ile},k2} 
\end{array} \right), \quad g = \left( \begin{array}{c}
  g_{r,-n1} \\
  \vdots \\
  g_{a,-n1}
\end{array} \right)
\]

Herein \(r\), denotes the position-based FIR coefficients, and \(g\), the acceleration-based FIR coefficients. Note that different filter orders can be assigned to the position- and acceleration-based filter parts. Results are shown in Fig. 18. Incorporating the acceleration part makes the FIR filter better capable of fitting snap-related peaks. Improvement in tracking performance is obtained as compared to the position-based FIR filter in Fig. 17 leaving performances of both FIR mapping and multi-table competitive. Note that the combination of position- and acceleration filters is not restrictive; for example plants with significant damping contributions may benefit from velocity filters.

**VI. COMPARISON AND CONCLUSIONS**

In this paper two methods are studied to deal with setpoint variation using iterative learning schemes. Both methods demonstrate their potential in reducing tracking errors and settling times of a high-performance industrial motion system. A further performance improvement is limited by nonlinear behavior of the controlled plant (e.g. position dependency, direction dependency and saturation in the actuator characteristics) for which is not accounted given the perspective of LTI behavior. Furthermore the setpoint is assumed to be the largest source of tracking error thereby neglecting any other disturbances. The given implementation of multi-table performs best at the jerk phases for which is learned. Contrary such a distinction is not encountered with the proposed FIR mapping which is here based on a broader variety of jerk phases. As a consequence an averaged filter is obtained over the set of training data. Finally it should be mentioned that the FIR mapping is less constrained with respect to the multi-table approach regarding setpoint variation. Moreover, from an implementation point of view, the limited number of FIR filter coefficients as compared to the number of multi-table entries gives a data-reduction advantage.

**REFERENCES**


Fig. 18. Time-series using position- and acceleration-based FIR mapping ILC; Top: initial error, error with ILC, error with FIR; Bottom: ILC force (grey), FIR force (black). Scaled acceleration profile (dashed)

acceleration filters is not restrictive; for example plants with significant damping contributions may benefit from velocity filters.