2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Model-free iterative learning control for LTI systems and experimental validation on a linear motor test setup

Pieter Janssens, Goele Pipeleers and Jan Swevers

Abstract— **This paper presents a novel model-free iterative learning control algorithm for linear time-invariant systems with actuator constraints. At every trial, a finite impulse response filter to update the system input is computed by solving a convex optimization problem that minimizes the next trial's tracking error while accounting for actuator constraints. The presented iterative learning control algorithm is validated on a linear motor positioning system. Experimental results show the ability of the proposed model-free algorithm to learn the optimal system input in the presence of cogging forces and actuator input constraints.**

I. INTRODUCTION

Iterative learning control (ILC) is an open-loop control strategy that aims at improving the tracking performance of a system executing the same task repeatedly under the same operating conditions. The system input is updated iteratively, i.e. from trial to trial, to improve the accuracy of the desired motion [1], [2]. Using this technique, accurate tracking can be obtained even when the system dynamics are uncertain. Reported applications of ILC include machine tool axes [3] and wafer stage motion systems [4], which involve linear motor systems that typically suffer from cogging disturbances [5]. Besides motion control, ILC has also been successful in other domains such as active noise control [6] and chemical process control [7].

When designing an ILC algorithm the aim is to use the error information from previous trials as efficiently as possible in order to achieve a minimal tracking error in as few iterations as possible. The simplest ILC law uses a PID-type learning filter, which consists of a proportional, integral and derivative gain on the tracking error to update the system input. More advanced learning laws [1] include plant inversion methods, quadratically optimal design (Q-ILC), $\mathcal{H}_\infty$ design methods and other optimization-based approaches [8], [9], [10]. These methods use a plant model and possibly also uncertainty models to ensure (robust) monotonic convergence.

Contrary to the variety of model-based ILC methods, model-free methods are rare in the ILC literature [11]. Model-free ILC algorithms have the advantage of being applicable to different machines without having to perform identification experiments over and over again. Another shortcoming of classical ILC methods is that the desired trajectory is assumed to be realizable given the actuator constraints, whereas in many situations this is not known beforehand. Only few methods consider actuator constraints [12]. This paper presents an ILC algorithm for linear time-invariant (LTI) systems that overcomes both these shortcomings. In addition to the theory behind the novel model-free ILC algorithm, the paper also presents the experimental validation on a permanent magnet linear motor. Two test cases show the ability of the proposed method to deal with actuator constraints and repeating disturbances such as actuator cogging.

Pieter Janssens, Goele Pipeleers and Jan Swevers are with the Department of Mechanical Engineering, Div. PMA, Katholieke Universiteit Leuven, Celestijnenlaan 300B, Heverlee B3001, Belgium Pieter.Janssens@mech.kuleuven.be

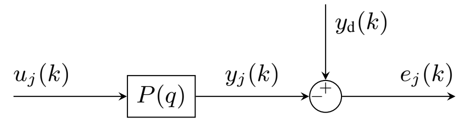

Fig. 1. Open-loop discrete-time LTI system $P(q)$.

The paper is organized as follows. Section II details the theory behind the presented model-free ILC algorithm and analyzes the influence of noise and disturbances on the performance. Section III discusses the linear motor test setup and presents the experimental results. Finally, section IV summarizes the conclusions.

II. A MODEL-FREE ILC ALGORITHM

This section presents the fundamental principle of the model-free ILC algorithm and extends the proposed method for systems with measurement noise and trial-invariant disturbances. At the end of this section the application to closed-loop systems is discussed.

A. Fundamental principle of the model-free ILC algorithm

Consider the open-loop, single-input single-output (SISO), discrete-time, LTI system $P(q)$ in Fig. 1. $P(q)$ has input:

$$u_j(k), \quad k \in \{1, 2, \ldots, N\},$$

output:

$$y_j(k), \quad k \in \{\tau+1, \tau+2, \ldots, \tau+N\},$$

desired output:

$$y_\mathrm{d}(k), \quad k \in \{\tau+1, \tau+2, \ldots, \tau+N\},$$

and tracking error $e_j(k) = y_\mathrm{d}(k) - y_j(k)$, where subscript $j \in \{0, 1, 2, \ldots\}$ denotes the trial number, k refers to the discrete time instants, q is the one-sample advance operator, τ denotes the relative degree of $P(q)$ and N denotes the number of samples per trial.

A widely used ILC update formula is

$$u_{j+1}(k) = Q(q)[u_j(k) + L(q)e_j(k)], \tag{1}$$

978-1-4577-0079-8/11/$26.00 ©2011 AACC

where $Q(q)$ is the robustness filter and $L(q)$ the learning filter. The design of $L(q)$ usually requires nominal plant knowledge since this filter relates $e_j(k)$ to $u_{j+1}(k)$, whereas the design of $Q(q)$ uses uncertainty models to ensure robust monotonic convergence.

Contrary to the design of iterative learning controllers given by (1), the proposed ILC method relies only on the linearity and time-invariance of the system. The system input is updated using a linear combination of previous system inputs convoluted with a trial-varying, but linear time-invariant, causal finite impulse response filter (FIR-filter) $\alpha_j(q)$ of length N:

$$u_{j+1}(k) = u_j(k) + u_{\mathrm{lc}}(k) * \alpha_j(k). \tag{2}$$

In this formula $u_{\mathrm{lc}}(k)$ represents any linear combination of the previous trials' input signals $u_0(k), u_1(k), \ldots, u_j(k)$, $\alpha_j(k)$ denotes the impulse response of $\alpha_j(q)$, and $*$ denotes the discrete-time convolution operator.

When updating the input signal $u_j(k)$ using (2), the corresponding output $y_{j+1}(k)$ is predicted by only relying on the system's linearity and time-invariance:

$$\widehat{y}_{j+1}(k) = y_j(k) + y_{\mathrm{lc}}(k) * \alpha_j(k), \tag{3}$$

where $y_{\mathrm{lc}}(k)$ denotes a linear combination of previous trials' output signals $y_0(k), y_1(k), \ldots, y_j(k)$ that is composed in the same way as the linear combination $u_{\mathrm{lc}}(k)$ used in (2), and $\widehat{y}_{j+1}(k)$ denotes the prediction of the output of trial $j+1$ for input $u_{j+1}(k)$, obtained without the use of a system model. At every iteration, the FIR-filter $\alpha_j(q)$ is computed by solving a convex optimization problem as explained below.

Using the lifted system representation [13], which is used in the remainder of this paper, the update law (2) is rewritten as:

$$\underbrace{\begin{bmatrix} u_{j+1}(1) \\ u_{j+1}(2) \\ \vdots \\ u_{j+1}(N) \end{bmatrix}}_{\mathbf{u}_{j+1}} = \underbrace{\begin{bmatrix} u_j(1) \\ u_j(2) \\ \vdots \\ u_j(N) \end{bmatrix}}_{\mathbf{u}_j} + \underbrace{\begin{bmatrix} u_{\mathrm{lc}}(1) & 0 & \cdots & 0 \\ u_{\mathrm{lc}}(2) & u_{\mathrm{lc}}(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ u_{\mathrm{lc}}(N) & u_{\mathrm{lc}}(N-1) & \cdots & u_{\mathrm{lc}}(1) \end{bmatrix}}_{\mathbf{U}_{\mathrm{lc}}} \underbrace{\begin{bmatrix} \alpha_j(1) \\ \alpha_j(2) \\ \vdots \\ \alpha_j(N) \end{bmatrix}}_{\boldsymbol{\alpha}_j}, \tag{4}$$

where $\mathbf{U}_{\mathrm{lc}}$ denotes the lower-triangular Toeplitz matrix of $u_{\mathrm{lc}}(k)$. Analogous to (4), the predicted output of trial $j+1$ is rewritten as:

$$\widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{\mathrm{lc}} \boldsymbol{\alpha}_j = \mathbf{y}_j + \mathbf{A}_j \mathbf{y}_{\mathrm{lc}}, \tag{5}$$

where

$$\mathbf{Y}_{\mathrm{lc}} = \begin{bmatrix} y_{\mathrm{lc}}(\tau+1) & 0 & \cdots & 0 \\ y_{\mathrm{lc}}(\tau+2) & y_{\mathrm{lc}}(\tau+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ y_{\mathrm{lc}}(\tau+N) & y_{\mathrm{lc}}(\tau+N-1) & \cdots & y_{\mathrm{lc}}(\tau+1) \end{bmatrix}$$

denotes the lower-triangular Toeplitz matrix of $y_{\mathrm{lc}}(k)$ and $\mathbf{A}_j$ denotes the lower-triangular Toeplitz matrix of $\alpha_j(k)$.

Between two trials the trial-varying FIR-filter $\alpha_j(q)$ is computed by solving the following convex optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_{\mathrm{d}} - \widehat{\mathbf{y}}_{j+1}\|_2 \\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{\mathrm{lc}} \boldsymbol{\alpha}_j \\ & \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_{\mathrm{lc}} \boldsymbol{\alpha}_j \\ & |\mathbf{u}_{j+1}| \leq \overline{u} \\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u}. \end{aligned} \tag{6}$$

The ℓ_2-norm of the predicted next trial's tracking error is minimized taking into account linear inequality constraints on $u_{j+1}(k)$ and $\delta u_{j+1}(k) = u_{j+1}(k) - u_{j+1}(k-1)$ to avoid saturation of the actuators. When this convex optimization problem is solved and thus the optimal FIR-filter $\alpha_j(k)$ is known, the next trial's input signal is calculated using (4).

Although any update law of the form (4) allows the output of an LTI system to be predicted without the use of a system model, some particular choices for $u_{\mathrm{lc}}(k)$ result in update laws with important advantages. For now, consider the following simple update laws:

$$u_{\mathrm{lc}}(k) = u_0(k): \quad \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_0 \boldsymbol{\alpha}_j, \tag{7}$$

$$u_{\mathrm{lc}}(k) = u_j(k): \quad \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_j \boldsymbol{\alpha}_j. \tag{8}$$

After the first trial, when computing $\mathbf{u}_1$, both update laws are equivalent and make use of the initial input signal $\mathbf{u}_0$. The proposed method, using update law (7) or (8), can be shown to converge in only one iteration to the minimal tracking error, provided that (i) $\mathbf{Y}_0$ is a full-rank matrix, and (ii) no measurement noise or disturbances are present. The first condition is sufficient to ensure that $\mathbf{e}_0 = \mathbf{y}_{\mathrm{d}} - \mathbf{y}_0$ is in the column range of $\mathbf{Y}_0$. The lower-triangular Toeplitz matrix $\mathbf{Y}_0$ is of full rank if and only if $y_0(\tau+1) \neq 0$, therefore $u_0(1)$ must be nonzero [1]. This way, the condition on the rank of $\mathbf{Y}_0$ restricts the choice of the first trial's input signal $u_0(k)$. The second condition ensures that the predicted output $\widehat{\mathbf{y}}_1 = \mathbf{y}_0 + \mathbf{Y}_0 \boldsymbol{\alpha}_0$ is exactly equal to the true output of trial $j = 1$, and the minimal value of the objective function of the optimization problem is the true minimal rms value of the tracking error for the imposed bounds on the actuator input. This globally optimal solution is found since optimization problem (6) is convex.

In practice, however, measurement noise and disturbances are present and more iterations are needed to converge to the optimal system input. At iterations $j > 1$, update law (7) and (8) and hence the corresponding ILC algorithms are different. In the case of update law (7), $\mathbf{Y}_{\mathrm{lc}} = \mathbf{Y}_0$ is of full rank at every trial along the learning process as long as the initial input signal $u_0(k)$ satisfies $u_0(1) \neq 0$. In the case of update law (8), this necessary condition to converge to the optimal system input might not be satisfied further on in the learning process since $\mathbf{Y}_{\mathrm{lc}} = \mathbf{Y}_j$ results from a previous

[1] It is assumed that the system's initial conditions are zero.

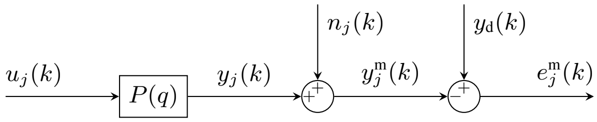

Fig. 2. Open-loop discrete-time system $P(q)$ with measurement noise $n_j(k)$.

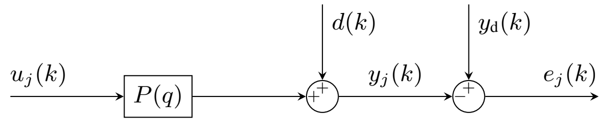

Fig. 3. Open-loop discrete-time system $P(q)$ with a trial-invariant output disturbance $d(k)$.

iteration's optimization problem and is not free to choose. For this reason, update law (7) is preferred.

B. Obtaining accurate output predictions in case of measurement noise

This section discusses the influence of measurement noise on the model-free ILC algorithm and proposes an essential modification to the algorithm.

Consider the LTI system $P(q)$ in Fig. 2 with input $u_j(k)$, true output $y_j(k)$, measured tracking error $e_j^\mathrm{m}(k)$ and measured output $y_j^\mathrm{m}(k) = y_j(k) + n_j(k)$, which is corrupted by zero-mean measurement noise $n_j(k)$ with standard deviation σ_n.

Since the true noise-free output $\mathbf{y}_j$ is not known, the predicted plant output $\widehat{\mathbf{y}}_{j+1}$ is computed from the measured output signals $\mathbf{y}_j^\mathrm{m}$ and $\mathbf{y}_0^\mathrm{m}$:

$$\begin{aligned}\widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j^\mathrm{m} + \mathbf{Y}_0^\mathrm{m}\boldsymbol{\alpha}_j,\\ &= \mathbf{y}_j + \mathbf{n}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j + \mathbf{N}_0\boldsymbol{\alpha}_j,\end{aligned} \tag{9}$$

where $\mathbf{N}_0$ and $\mathbf{Y}_0^\mathrm{m}$ respectively denote the lower-triangular Toeplitz matrices of $n_0(k)$ and $y_0^\mathrm{m}(k)$. The difference between the true output $\mathbf{y}_{j+1}$ and the predicted output $\widehat{\mathbf{y}}_{j+1}$ is called the prediction error of trial $j+1$ and is denoted by $\mathbf{e}_{j+1}^\mathrm{pr}$. The update relation (7) still yields

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j, \tag{10}$$

whereby the prediction error of trial $j+1$ is given by:

$$\mathbf{e}_{j+1}^\mathrm{pr} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = -\mathbf{n}_j - \mathbf{N}_0\boldsymbol{\alpha}_j. \tag{11}$$

Consequently, the objective function of optimization program (6) amounts to

$$\|\mathbf{y}_\mathrm{d} - \widehat{\mathbf{y}}_{j+1}\| = \left\|\mathbf{y}_\mathrm{d} - \mathbf{y}_{j+1} + \mathbf{e}_{j+1}^\mathrm{pr}\right\|, \tag{12}$$

and hence the prediction error hinders the model-free ILC algorithm from further reducing the tracking error. Therefore it is necessary to constrain this error.

From (11) the standard deviation of the prediction error at sample k of trial $j+1$ can easily be derived:

$$\sigma_{e_{j+1}^\mathrm{pr}(k)} = \sigma_\mathrm{n}\sqrt{1 + \|\alpha_j(1,\ldots,k)\|_2}. \tag{13}$$

The largest standard deviation of the prediction error is found at the last sample of the trial, $k = N$:

$$\sigma_{e_{j+1}^\mathrm{pr}(N)} = \sigma_\mathrm{n}\sqrt{1 + \|\boldsymbol{\alpha}_j\|_2}. \tag{14}$$

This analysis shows that adding the convex constraint

$$\|\boldsymbol{\alpha}_j\|_2 \leq t \tag{15}$$

to optimization program (6) limits the standard deviation of the prediction error by $\sigma_\mathrm{n}\sqrt{1+t}$ and hence allows the model-free ILC algorithm to further reduce the tracking error.

Hence, in case of measurement noise, the following convex optimization program is solved between two trials to obtain the optimal FIR-filter $\alpha_j(q)$ and thus also the updated input signal $u_{j+1}(k)$:

$$\begin{aligned}\underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_\mathrm{d} - \widehat{\mathbf{y}}_{j+1}\|_2\\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}_j^\mathrm{m} + \mathbf{Y}_0^\mathrm{m}\boldsymbol{\alpha}_j\\ & \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j\\ & |\mathbf{u}_{j+1}| \leq \overline{u}\\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u}\\ & \|\boldsymbol{\alpha}_j\|_2 \leq t.\end{aligned} \tag{16}$$

In addition to the beneficial effect on the prediction error, the constraint on $\|\boldsymbol{\alpha}_j\|_2$ also limits the change in input signal $\mathbf{U}_0\boldsymbol{\alpha}_j$ from one trial to the next trial. Consequently, the constraint on $\|\boldsymbol{\alpha}_j\|_2$ influences the convergence speed of the ILC algorithm. The upperbound on $\|\boldsymbol{\alpha}_j\|_2$ regulates the trade-off between convergence speed and accuracy of the output prediction.

C. Dealing with trial-invariant disturbances

In the previous section the model-free ILC algorithm was adapted for tracking problems where measurement noise is present. In many real-life applications, however, the output also suffers from trial-invariant disturbances. Contrary to measurement noise, these trial-invariant disturbances can be compensated using iterative learning controllers.

Consider again an LTI system $P(q)$ with a certain trial-invariant output disturbance $d_j(k) = d(k)$, for all $j = 0, 1, 2, \ldots$ (see Fig. 3). Using the lifted-system representation, the system dynamics, including the trial-invariant disturbance $d(k)$, are written as follows:

$$\mathbf{y}_j = \mathbf{P}\mathbf{u}_j + \mathbf{d}, \tag{17}$$

where $\mathbf{P}$ denotes the lower-triangular Toeplitz matrix of the impulse response of the system $P(q)$.

Since the model-free ILC algorithm assumes the system dynamics to be LTI, again a prediction error arises in the objective function of the optimization program when update law (7) is used. Combining (17) and (10) results in the predicted output:

$$\begin{aligned}\widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j + \mathbf{Y}_0\boldsymbol{\alpha}_j,\\ &= \mathbf{P}(\mathbf{u}_j + \mathbf{U}_0\boldsymbol{\alpha}_j) + \mathbf{d} + \mathbf{D}\boldsymbol{\alpha}_j,\end{aligned} \tag{18}$$

where $\mathbf{D}$ denotes the lower-triangular Toeplitz matrix of $d(k)$, whereas the actual output is:

$$\mathbf{y}_{j+1} = \mathbf{P}(\mathbf{u}_j + \mathbf{U}_0 \boldsymbol{\alpha}_j) + \mathbf{d}. \tag{19}$$

The resulting prediction error is:

$$\mathbf{e}^{\text{pr}}_{j+1} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = \mathbf{D}\boldsymbol{\alpha}_j. \tag{20}$$

Constraining the ℓ_2-norm of $\boldsymbol{\alpha}_j$ would again reduce the prediction error at the cost of convergence speed.

In the presence of trial-invariant disturbances, however, more appropriate choices of the update law, resulting in more accurate output predictions and therefore also faster convergence, can be made.

Consider the following specific case of (4):

$$u_{\text{lc}}(k) = u_j(k) - u_{j-1}(k) + \gamma u_0(k), \tag{21}$$

where $\gamma \approx 0.01 \ldots 0.1$, yielding:

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0). \tag{22}$$

The predicted output of trial $j+1$ for the system described by (17) is:

$$\begin{aligned} \widehat{\mathbf{y}}_{j+1} &= \mathbf{y}_j + \mathbf{A}_j(\mathbf{y}_j - \mathbf{y}_{j-1} + \gamma \mathbf{y}_0), \\ &= \mathbf{P}\mathbf{u}_j + \mathbf{d} + \mathbf{A}_j(\mathbf{P}(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0) + \gamma \mathbf{d}), \end{aligned} \tag{23}$$

whereas the actual output is:

$$\widehat{\mathbf{y}}_{j+1} = \mathbf{P}(\mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0)) + \mathbf{d}. \tag{24}$$

Consequently the resulting prediction error is:

$$\mathbf{e}^{\text{pr}}_{j+1} = \mathbf{y}_{j+1} - \widehat{\mathbf{y}}_{j+1} = \gamma \mathbf{A}_j \mathbf{d}. \tag{25}$$

This analysis shows that update law (22) reduces the prediction error due to trial-invariant disturbances and still allows $\mathbf{Y}_{\text{lc}}$ to be of full rank at every trial along the learning process. The scalar γ is the second tuning parameter of the model-free ILC algorithm next to t, the upperbound on the ℓ_2-norm of $\boldsymbol{\alpha}_j$, and regulates the trade-off between the prediction error due to trial-invariant disturbances and the ability to reduce the next iteration's tracking error. Both tuning parameters are allowed to vary from iteration to iteration.

To summarize, in the presence of measurement noise and trial-invariant disturbances the following convex optimization problem is solved to obtain the optimal FIR-filter and hence also the updated input signal $u_{j+1}(k)$:

$$\begin{aligned} \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} \quad & \|\mathbf{y}_{\text{d}} - \widehat{\mathbf{y}}_{j+1}\|_2 \\ \text{subject to} \quad & \widehat{\mathbf{y}}_{j+1} = \mathbf{y}^{\text{m}}_j + (\mathbf{Y}^{\text{m}}_j - \mathbf{Y}^{\text{m}}_{j-1} + \gamma \mathbf{Y}^{\text{m}}_0)\boldsymbol{\alpha}_j \\ & \mathbf{u}_{j+1} = \mathbf{u}_j + (\mathbf{U}_j - \mathbf{U}_{j-1} + \gamma \mathbf{U}_0)\boldsymbol{\alpha}_j \\ & |\mathbf{u}_{j+1}| \leq \overline{u} \\ & |\boldsymbol{\delta u}_{j+1}| \leq \overline{\delta u} \\ & \|\boldsymbol{\alpha}_j\|_2 \leq t. \end{aligned} \tag{26}$$

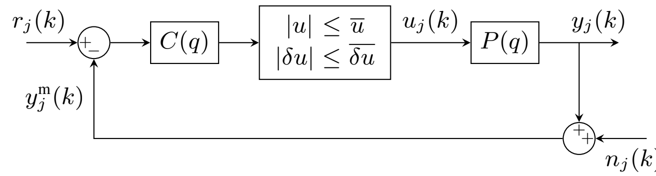

Fig. 4. Closed-loop discrete-time system with actuator constraints.

D. Application to closed-loop systems with actuator constraints

In most applications, ILC is combined with feedback control since an iterative learning controller cannot compensate for nonrepeating disturbances. Consider the closed-loop system in Fig. 4 with actuator constraints $\overline{u}$ and $\overline{\delta u}$, controller $C(q)$, plant $P(q)$, reference signal $r_j(k)$, actuator input $u_j(k)$, output $y_j(k)$ and measured output $y^{\text{m}}_j(k) = y_j(k) + n_j(k)$. The difference with the aforementioned open-loop systems is that in the closed-loop case the reference signal $r_j(k)$ is updated in order to track a given desired output $y_{\text{d}}(k)$, taking into account the constraints on the actuator input $u_j(k)$, whereas in the open-loop case the actuator input itself is updated.

When the actuator constraints are active and thus the input signal $u_j(k)$ is clipped, the relation between the reference signal $r_j(k)$ and the output $y_j(k)$ of the closed-loop system in Fig. 4 becomes nonlinear. Still, even when the actuator constraints are active, the plant $P(q)$ itself is LTI. Therefore exactly the same optimization program as in the open-loop case can be used to obtain an optimal FIR-filter to construct a predicted next trial's actuator input $\widehat{u}_{j+1}(k)$ that satisfies the actuator constraints. The only required adjustment for closed-loop systems is that the next trial's reference trajectory $r_{j+1}(k)$ is calculated after solving convex optimization problem (26). The next trial's reference signal that corresponds to the predicted next trial's actuator input $\widehat{u}_{j+1}(k)$ and the predicted next trial's output $\widehat{y}_{j+1}(k)$ is given by:

$$\mathbf{r}_{j+1} = \mathbf{C}^{-1}\widehat{\mathbf{u}}_{j+1} + \widehat{\mathbf{y}}_{j+1}, \tag{27}$$

where $\mathbf{C}$ denotes the lower-triangular Toeplitz matrix of the impulse response of controller $C(q)$ and $\widehat{\mathbf{u}}_{j+1}$, $\widehat{\mathbf{y}}_{j+1}$ are given by an appropriate update law.

III. EXPERIMENTAL VALIDATION

This section presents the experimental validation of the proposed model-free ILC algorithm on a linear motor positioning system. Section III-A describes the linear motor test setup. Section III-B discusses the experimental results of a first test case where the system is able to track the reference trajectory without hitting the actuator bounds. The results show the ability of the proposed algorithm to deal with actuator cogging. Section III-C presents the experimental results of a second test case where the proposed ILC algorithm manages to achieve a minimal tracking error taking into account the imposed bounds on the actuator input.

Fig. 5. Linear motor test setup.

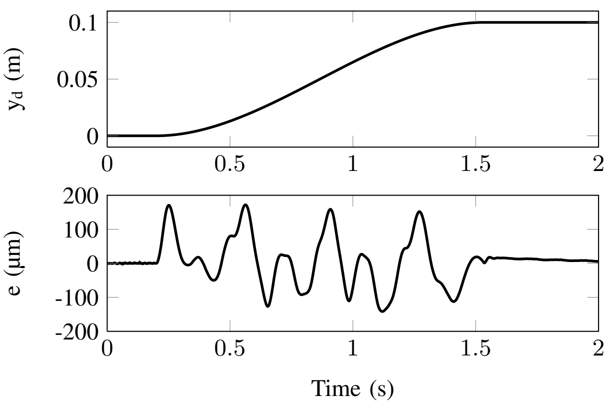

Fig. 6. Tracking error to a smooth reference trajectory.

A. Experimental setup

The experimental validation is performed on a current-controlled permanent-magnet linear motor depicted in Fig. 5. The position of the carriage of the linear motor is measured using a linear encoder and fed back to an already available position controller. The position measurement is subject to output noise with an rms value of approximately $0.12\,\mu$m. The limiting factor concerning the remaining tracking error, however, is not the measurement noise but the repeatability of the closed-loop system, which is approximately $1\,\mu$m. In this paper, all experiments are performed with a sample time of $2\,$ms.

In permanent-magnet linear motors cogging is considered as the main disturbance [14]. Cogging forces are considered as a combination of two types of ripple: position ripple and force ripple. The position ripple is the necessary force to keep the carriage of the linear motor at a certain position, with zero motor input current. This disturbance force only depends on the position and does not depend on the input current. The force ripple is caused by the variation of the motor constant with the position. Therefore this disturbance force is position-dependent and proportional with the motor input current [5]. To illustrate the effect of the cogging disturbance on the measured output, a smooth reference trajectory is applied to the closed-loop system. Fig. 6 shows the applied reference trajectory and the corresponding tracking error. The tracking error contains a repetitive pattern with a spatial frequency that corresponds to the width of the permanent magnets. For this smooth reference trajectory, the cogging forces already result in a peak tracking error of $170\,\mu$m.

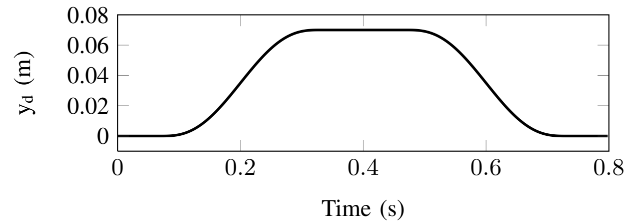

Fig. 7. Desired output $y_d(k)$.

	e_{max} (µm)	e_{rms} (µm)
Feedback only	1736	931
Trial 20	4.3	1.9

TABLE I

COMPARISON OF THE PEAK VALUE AND RMS VALUE OF THE TRACKING ERROR BEFORE AND AFTER LEARNING FOR A FORWARD AND BACKWARD MOTION OF $7\,$cm.

B. Cogging compensation

Since the reference trajectory is updated from trial to trial, the cogging disturbance is not entirely trial-invariant. However, the more the algorithm converges to the optimal solution, the smaller the update of the reference trajectory and hence also the trial-to-trial variation of the cogging disturbance will be. Close to the optimal solution cogging disturbances can be considered trial-invariant, and hence iterative learning controllers are able to compensate for cogging disturbances.

Fig. 7 shows the desired output $y_d(k)$, which is a forward and backward motion of $7\,$cm that needs to be executed in 0.8 seconds. Before every iteration the carriage is returned to the same starting position (within $0.3\,\mu$m) because of the position-dependency of the cogging disturbance.

During the first trial of the learning process a reference signal with a rich frequency content, satisfying $r_0(1) \neq 0$, is applied to the system. When calculating the second trial's reference signal, update law (7) is used because experimental data from only one previous iteration are available. From then on, the model-free ILC algorithm uses the update law given by (22) with $\gamma = 0.05$ and the upperbound t on $\|\alpha_j\|_2$ equal to 0.3. Fig. 8 shows the rms value of the tracking error as a function of the trial number and table I compares the peak value and rms value of the tracking error before and after learning. Fig. 9 shows the remaining tracking error of the 20^{th} trial. Further reduction of the tracking error is hardly possible due to the limited repeatability of the test setup. These results show that the proposed model-free ILC algorithm is able to compensate for cogging disturbances.

C. Optimal reference tracking taking into account actuator constraints

The following experimental results show the ability of the proposed method to learn a reference trajectory leading to the minimal rms value of the tracking error taking into account

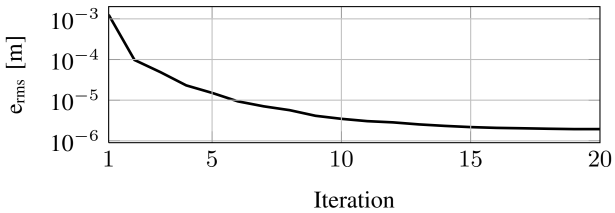

Fig. 8. Rms value of the tracking error as a function of the trial number.

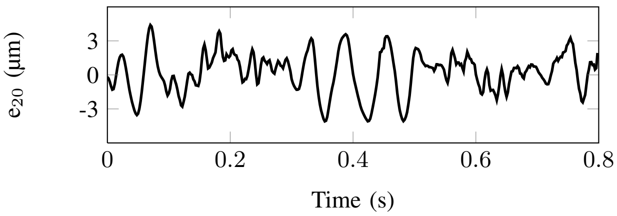

Fig. 9. Tracking error of the 20$^{\text{th}}$ trial.

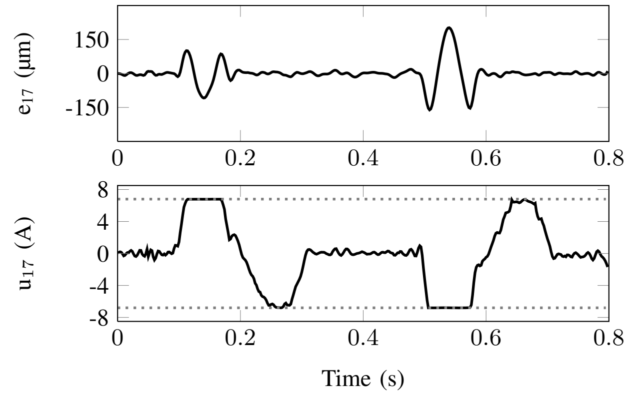

Fig. 10. Tracking error e_{17} and actuator input u_{17} of the 17^{th} trial together with the bounds on the actuator input.

actuator constraints. The desired output is again a forward and backward motion that needs to be executed in $0.8\,\mathrm{s}$ but the distance is increased to $10\,\mathrm{cm}$ such that the constraints on the actuator input become active.

The model-free ILC algorithm, which uses the update law given by (22) with $t = 0.3$ and $\gamma = 0.1$, reaches convergence in 17 iterations. Fig. 10 shows the remaining tracking error e_{17} and the actuator input u_{17} of the 17^{th} iteration. Cogging is compensated at time instants when no actuator constraints are active. Only when the actuator input hits its bounds, the tracking error is an order of magnitude larger than the noise level.

IV. Conclusion

This paper presents a model-free ILC algorithm for LTI systems with actuator constraints. Between two trials the system input is updated using an optimal trial-varying FIR-filter, obtained by solving a convex optimization problem that minimizes the next trial's tracking error. This convex optimization problem allows accounting for linear actuator constraints.

The effect of measurement noise on the next trial's reference signal is reduced by constraining the standard deviation of the prediction error. The upperbound on this convex constraint regulates the trade-off between convergence speed and accuracy of the output prediction. The influence of trial-invariant disturbances on the output prediction is analyzed and learning laws that yield smaller prediction errors and therefore also faster convergence are proposed.

The experimental validation on a linear motor test setup shows the ability of the proposed model-free ILC algorithm to deal with cogging disturbances and actuator constraints.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: a learning based method for high-performance tracking control," *IEEE Control systems magazine*, vol. 26, pp. 96–114, June 2006.

[2] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International journal of control*, vol. 73, pp. 930–954, 2000.

[3] D. Kim and S. Kim, "An iterative learning control method with application for cnc machine tools," *IEEE Transactions on Industry Applications*, vol. 32, no. 1, pp. 66–72, January/February 1996.

[4] M. Heertjes and T. Tso, "Nonlinear iterative learning control with applications to lithographic machinery," *Control engineering Practice*, vol. 15, pp. 1545–1555, 2007.

[5] P. Van den Braembussche, J. Swevers, H. Van Brussel, and P. Vanherck, "Motion control of machine tool axes with linear motor," in *Proceedings of the 27th CIRP International Seminar on Manufacturing Systems*, Ann Arbor, Michigan, USA, May 1995.

[6] B. Stallaert, G. Pinte, P. Sas, W. Desmet, and J. Swevers, "A novel design strategy for iterative learning and repetitive controllers of systems with a high modal density: Application to active noise control," *Mechanical Systems and Signal Processing*, vol. 24, no. 2, pp. 444–454, February 2010.

[7] K. Lee, S. Bang, S. Yi, J. Son, and S. Yoon, "Iterative learning control of heat-up phase for a batch polymerization reactor," *Journal of process control*, vol. 4, pp. 255–262, 1996.

[8] D. Owens and J. Hätönen, "Iterative learning control - an optimization paradigm," *Annual reviews in control*, vol. 29, pp. 57–70, 2005.

[9] D. H. Nguyen and D. Banjerdpongchai, "An LMI approach for robust iterative learning control with quadratic performance criterion," *Journal of process control*, vol. 19, pp. 1054–1060, 2009.

[10] N. Amann, D. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proc.-Control Theory Applications*, vol. 143, pp. 217–224, 1996.

[11] Y. Li and J. Bechhoefer, "Model-free iterative control of repetitive dynamics for high-speed scanning in atomic force microscopy," *Review of scientific instruments*, vol. 80, 2009.

[12] J. A. Frueh and M. Q. Phan, "Linear quadratic optimal learning control (LQL)," in *Proceedings of the 37th IEEE Conference on Decision & Control*, 1998.

[13] M. Phan and R. Longman, "A mathematical theory of learning control for linear discrete multivariable systems," in *Proceedings of the AIAA/AAS Astrodynamics Conference*. Minneapolis, Minnesota: AIAA Publishers, August 1988, pp. 740–746.

[14] H.-S. Ahn, "State-periodic adaptive compensation of cogging and coulomb friction in permanent-magnet linear motors," *IEEE Transactions on magnetics*, vol. 41, pp. 90–98, 2005.

Acknowledgment

Goele Pipeleers is Postdoctoral Fellow of the Research Foundation - Flanders. This work has been carried out within the framework of projects IWT-SBO 80032 (LECOPRO) of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) and G.0422.08 and G.0377.09 of the Research Foundation - Flanders (FWO - Flanders). This work also benefits from K.U.Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC) and from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. Also ETEL is gratefully acknowledged for their support.