

Model-free iterative learning of time-optimal point-to-point motions for LTI systems

Pieter Janssens, Goele Pipeleers and Jan Swevers

Department of Mechanical Engineering, Div. PMA, Katholieke Universiteit Leuven,
 Celestijnenlaan 300B, Heverlee B3001, Belgium

email: Pieter.Janssens@mech.kuleuven.be

Abstract—This paper presents a model-free iterative learning control algorithm, which generates time-optimal point-to-point motions for linear time-invariant systems. The proposed optimization-based algorithm consists of two levels. At the first level, a bisection algorithm determines the fastest possible point-to-point motion, i.e. the motion time is minimized, subject to actuator limitations. At the second level, an iterative learning control algorithm for point-to-point motions learns the system input that results in a point-to-point motion, with the minimal motion time obtained by the bisection algorithm at the first level. Simulation results show that the proposed model-free method is able to learn the time-optimal system input for a given point-to-point motion problem in the presence of measurement noise and repeating disturbances.

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 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]. Using this technique, accurate tracking can be obtained despite model uncertainty and repeating disturbances. Reported applications in the field of motion control include industrial robots [2] and wafer stage motion systems [3].

While ILC was originally developed for tracking control problems, it has recently been adapted to suppress residual vibrations of flexible systems executing point-to-point motion problems [4]. In this case, the exact trajectory from an initial position to a desired end position is undefined. The desired output is only defined at some time instants of the trial, e.g. at the dwell of the point-to-point motion.

This paper presents a model-free ILC algorithm that learns the fastest possible trajectory, subject to actuator constraints, for a point-to-point motion problem. The algorithm consists of two levels. At the first level, a bisection algorithm minimizes the time required to reach the endpoint with a desired accuracy. The result of the first level algorithm is then used by a model-free ILC algorithm for point-to-point motions at the second level. The basic idea behind this ILC algorithm for point-to-point motion problems is adopted from a model-free ILC algorithm for tracking control problems described in previous papers by the authors [5], [6]. The ILC algorithm for point-to-point motions learns the system input that results in a point-to-point motion with the minimal motion time calculated by the bisection algorithm at the first level.

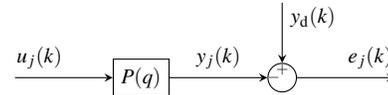


Fig. 1. Open-loop discrete-time LTI system $P(q)$.

Another benefit, besides the minimal motion time, over current ILC algorithms for point-to-point motions [4], is that the presented method does not require a plant model, the only requirements are linearity and time-invariance of the system. Model-free ILC algorithms have the advantage of being applicable to different machines without having to perform an identification experiment on every single machine.

The outline of this paper is as follows. First, the application of the ILC algorithm presented in [5], [6] to point-to-point motion problems is described in section II. This is the second level of the two-level time-optimal ILC algorithm. Consequently, section III describes how this algorithm is extended with a bisection algorithm such that time-optimal point-to-point motions are learned. This is the first level of the two-level time-optimal ILC algorithm. Section IV describes how these two algorithms are combined into a model-free time-optimal ILC algorithm for point-to-point motions. Section V presents simulation results on an accurate model of a linear motor, which includes actuator limitations, cogging disturbances and measurement noise. These results show that the proposed algorithm is able to learn the time-optimal system input for a given point-to-point motion problem. Finally, section VI summarizes the conclusions.

II. MODEL-FREE ILC FOR POINT-TO-POINT MOTIONS

This section presents the model-free ILC algorithm for point-to-point motion problems, which constitutes the second level of the time-optimal ILC algorithm.

A. Basic model-free ILC algorithm

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

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

and output:

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

where subscript $j \in \{0, 1, 2, \dots\}$ denotes the trial number, k refers to the discrete time instants kT_s , T_s denotes the

sampling period, q is the one-sample advance operator, and N denotes the number of samples per trial. Let $y_d(k)$ and \mathcal{K}_d respectively denote the desired output and the set of all time instants at which the desired output is specified. For point-to-point motion problems, only the time instants at which standstill at a certain position is desired are included in \mathcal{K}_d . The bisection algorithm at the first level determines the set \mathcal{K}_d that corresponds with the minimal motion time (see section III). The positioning error is given by:

$$e_j(k) = y_d(k) - y_j(k), \quad k \in \mathcal{K}_d. \quad (1)$$

Contrary to traditional model-based ILC methods, the proposed ILC method relies on the linearity and time-invariance of the system only; a plant model is not required. 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 :

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

In this formula $\alpha_j(k)$ denotes the impulse response of $\alpha_j(q)$, $u_{lc}(k)$ represents a linear combination of the previous trials' input signals $u_0(k), u_1(k), \dots, u_j(k)$, 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 relying on the system's linearity and time-invariance only:

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

where $y_{lc}(k)$ denotes a linear combination of previous trials' output signals $y_0(k), y_1(k), \dots, y_j(k)$ that is composed in the same way as the linear combination $u_{lc}(k)$ used in (2). 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 [7], which is used in the remainder of this paper, the update law (2) is rewritten as:

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

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

$$\hat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{lc} \boldsymbol{\alpha}_j = \mathbf{y}_j + \mathbf{A}_j \mathbf{y}_{lc}, \quad (5)$$

where \mathbf{Y}_{lc} and \mathbf{A}_j respectively denote the lower-triangular Toeplitz matrix of $y_{lc}(k)$ and $\alpha_j(k)$.

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

$$\begin{aligned} & \underset{\boldsymbol{\alpha}_j \in \mathbb{R}^N}{\text{minimize}} && \|\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)\|_2 \\ & \text{subject to} && \hat{\mathbf{y}}_{j+1} = \mathbf{y}_j + \mathbf{Y}_{lc} \boldsymbol{\alpha}_j, \\ & && \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_{lc} \boldsymbol{\alpha}_j, \\ & && |\mathbf{u}_{j+1}| \leq \bar{\mathbf{u}}, \quad |\delta \mathbf{u}_{j+1}| \leq \bar{\delta \mathbf{u}}. \end{aligned} \quad (6)$$

The ℓ_2 -norm of the predicted next trial's positioning error $\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)$ 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 convex optimization problem (6) is solved and thus the optimal FIR-filter $\alpha_j(q)$ is known, the next trial's input signal is computed 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 $u_{lc}(k)$ result in update laws with important advantages. For now, consider the following simple update laws:

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

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

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 value of the objective function, provided that (i) \mathbf{U}_0 is a full-rank matrix, and (ii) no measurement noise or disturbances are present. The first condition is sufficient to ensure that the required change in system input $\mathbf{u}^* - \mathbf{u}_0$, where \mathbf{u}^* denotes the time-optimal system input, is in the column range of \mathbf{U}_0 . The lower-triangular Toeplitz matrix \mathbf{U}_0 is of full rank if and only if $u_0(1) \neq 0$. This way, the condition on the rank of \mathbf{U}_0 restricts the choice of the first trial's input signal $u_0(k)$. The second condition ensures that the predicted output $\hat{\mathbf{y}}_1 = \mathbf{y}_0 + \mathbf{Y}_0 \boldsymbol{\alpha}_0$ is exactly equal to the true output of trial $j=1$, and hence the minimum of optimization problem (6) equals the true minimal value of the considered objective function for the imposed bounds on the actuator input. This global optimum 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), $\mathbf{U}_{lc} = \mathbf{U}_0$ 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 condition to converge to the time-optimal system input might not be satisfied further on in the learning process since $\mathbf{U}_{lc} = \mathbf{U}_j$ results from a previous iteration's

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

B. Dealing with measurement noise

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

Consider an open-loop LTI system $P(q)$ with input $u_j(k)$, true output $y_j(k)$, and measured output $y_j^m(k) = y_j(k) + n_j(k)$, 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 unknown, the predicted plant output $\hat{\mathbf{y}}_{j+1}$ is computed from the measured output signals \mathbf{y}_j^m and \mathbf{y}_0^m in correspondence with update law (7):

$$\begin{aligned}\hat{\mathbf{y}}_{j+1} &= \mathbf{y}_j^m + \mathbf{Y}_0^m \boldsymbol{\alpha}_j, \\ &= \mathbf{y}_j + \mathbf{n}_j + \mathbf{Y}_0 \boldsymbol{\alpha}_j + \mathbf{N}_0 \boldsymbol{\alpha}_j,\end{aligned}\quad (9)$$

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

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

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

$$e_{j+1}^{\text{pr}} = \mathbf{y}_{j+1} - \hat{\mathbf{y}}_{j+1} = -\mathbf{n}_j - \mathbf{N}_0 \boldsymbol{\alpha}_j. \quad (11)$$

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

$$\|\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)\|_2 = \quad (12)$$

$$\|\mathbf{y}_d(\mathcal{K}_d) - \mathbf{y}_{j+1}(\mathcal{K}_d) + e_{j+1}^{\text{pr}}(\mathcal{K}_d)\|_2, \quad (13)$$

and hence the prediction error impedes the model-free ILC algorithm from minimizing the true positioning error. Therefore it is necessary to constrain the prediction error.

From (11) the standard deviation of the prediction error of trial $j+1$ can easily be derived. The largest standard deviation of the prediction error is found at the last sample of the trial, $k=N$:

$$\sigma_{e_{j+1}^{\text{pr}}(N)} = \sigma_n \sqrt{1 + \|\boldsymbol{\alpha}_j\|_2}. \quad (14)$$

This analysis shows that adding the convex constraint

$$\|\boldsymbol{\alpha}_j\|_2 \leq t \quad (15)$$

to optimization program (6) limits the standard deviation of the prediction error and therefore also the influence of the prediction error on the objective function.

Hence, in the presence 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 $u_{j+1}(k)$:

$$\begin{aligned}\text{minimize}_{\boldsymbol{\alpha}_j \in \mathbb{R}^N} & \quad \|\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)\|_2 \\ \text{subject to} & \quad \hat{\mathbf{y}}_{j+1} = \mathbf{y}_j^m + \mathbf{Y}_0^m \boldsymbol{\alpha}_j, \\ & \quad \mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{U}_0 \boldsymbol{\alpha}_j, \\ & \quad |\mathbf{u}_{j+1}| \leq \bar{\mathbf{u}}, \quad |\delta \mathbf{u}_{j+1}| \leq \bar{\delta \mathbf{u}} \\ & \quad \|\boldsymbol{\alpha}_j\|_2 \leq t.\end{aligned}\quad (16)$$

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

Towards the end of the learning process, when the output motion converges to the desired one, the accuracy of the output prediction gets more important whereas the required change in input signal decreases. For this reason, it is advantageous to lower the value of t towards the end of the learning process resulting in fast convergence in the beginning of the learning process and accurate output prediction when necessary, i.e. at the end of the learning process.

C. Dealing with trial-invariant disturbances

In the previous subsection the model-free ILC algorithm was adapted for problems where measurement noise is present. In many applications, however, the output also suffers from trial-invariant disturbances. In this case, convergence speed can be increased by choosing an appropriate update law.

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

$$\mathbf{y}_j = \mathbf{P} \mathbf{u}_j + d, \quad (17)$$

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 (7), (10) and (17) results in the following prediction error:

$$e_{j+1}^{\text{pr}} = \mathbf{y}_{j+1} - \hat{\mathbf{y}}_{j+1} = -\mathbf{D} \boldsymbol{\alpha}_j, \quad (18)$$

where \mathbf{D} denotes the lower-triangular Toeplitz matrix of $d(k)$. Constraining the ℓ_2 -norm of $\boldsymbol{\alpha}_j$ again reduces 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):

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0), \quad (19)$$

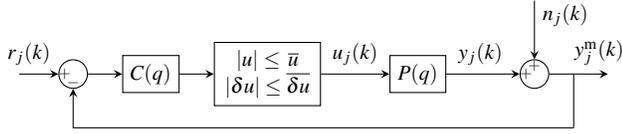


Fig. 2. Closed-loop discrete-time system with actuator constraints.

where γ determines the relative weight of \mathbf{u}_0 with respect to $\mathbf{u}_j - \mathbf{u}_{j-1}$. The predicted output of trial $j+1$ for the system described by (17) is:

$$\begin{aligned} \hat{\mathbf{y}}_{j+1} &= \mathbf{y}_j + \mathbf{A}_j(\mathbf{y}_j - \mathbf{y}_{j-1} + \gamma \mathbf{y}_0), \\ &= \mathbf{P}\mathbf{u}_j + \mathbf{d} + \mathbf{A}_j(\mathbf{P}(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0) + \gamma \mathbf{d}), \end{aligned} \quad (20)$$

whereas the actual output is:

$$\mathbf{y}_{j+1} = \mathbf{P}(\mathbf{u}_j + \mathbf{A}_j(\mathbf{u}_j - \mathbf{u}_{j-1} + \gamma \mathbf{u}_0)) + \mathbf{d}. \quad (21)$$

Consequently, the resulting prediction error is:

$$\mathbf{e}_{j+1}^{\text{pr}} = \mathbf{y}_{j+1} - \hat{\mathbf{y}}_{j+1} = -\gamma \mathbf{A}_j \mathbf{d}. \quad (22)$$

This analysis shows that an appropriate choice of γ in update law (19) leads to a reduced prediction error due to trial-invariant disturbances (for $|\gamma| < 1$), and still allows \mathbf{U}_{lc} to be of full rank at every trial of the learning process (for $\gamma \neq 0$), under the assumption that $\mathbf{u}_1(k)$ satisfies $\mathbf{u}_1(0) \neq 0$. The tuning parameter γ regulates the trade-off between the prediction error due to trial-invariant disturbances and the ability to reduce the next trial's positioning error.

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 $\mathbf{u}_{j+1}(k)$:

$$\begin{aligned} &\underset{\alpha_j \in \mathbb{R}^N}{\text{minimize}} && \|\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)\|_2 \\ &\text{subject to} && \hat{\mathbf{y}}_{j+1} = \mathbf{y}_j^m + (\mathbf{Y}_j^m - \mathbf{Y}_{j-1}^m + \gamma \mathbf{Y}_0^m) \alpha_j \\ &&& \mathbf{u}_{j+1} = \mathbf{u}_j + (\mathbf{U}_j - \mathbf{U}_{j-1} + \gamma \mathbf{U}_0) \alpha_j \\ &&& |\mathbf{u}_{j+1}| \leq \bar{\mathbf{u}}, \quad |\delta \mathbf{u}_{j+1}| \leq \bar{\delta \mathbf{u}} \\ &&& \|\alpha_j\|_2 \leq t. \end{aligned} \quad (23)$$

D. Applications to closed-loop systems with actuator constraints

In many applications, ILC is combined with feedback control since an iterative learning controller cannot compensate for nonrepeating disturbances. Consider the closed-loop system in Fig. 2 with actuator constraints $\bar{\mathbf{u}}$ and $\bar{\delta \mathbf{u}}$, controller $C(q)$, plant $P(q)$, reference signal $r_j(k)$, actuator input $\mathbf{u}_j(k)$, output $\mathbf{y}_j(k)$ and measured output $\mathbf{y}_j^m(k) = \mathbf{y}_j(k) + \mathbf{n}_j(k)$. 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 $\mathbf{y}_d(k)$, taking into account the constraints on the actuator input $\mathbf{u}_j(k)$, whereas in the open-loop case the actuator input itself is updated.

When the actuator constraints are active, the relation between the reference signal $r_j(k)$ and the output $\mathbf{y}_j(k)$ of the closed-loop system in Fig. 2 becomes nonlinear. To avoid this nonlinearity, optimization program (23) is solved first to compute the optimal FIR-filter $\alpha_j(q)$, and herewith the optimal next trial's input signal $\mathbf{u}_{j+1}(k)$ that satisfies the

actuator constraints. However, since the actuator signal can no longer be accessed directly, it is aimed for by applying the following reference signal:

$$\mathbf{r}_{j+1} = \mathbf{C}^{-1} \mathbf{u}_{j+1} + \hat{\mathbf{y}}_{j+1}, \quad (24)$$

where \mathbf{C} denotes the lower-triangular Toeplitz matrix of the impulse response of controller $C(q)$, while \mathbf{u}_{j+1} and $\hat{\mathbf{y}}_{j+1}$ result from the solution of (23). To ensure that the rank condition on \mathbf{U}_{lc} is satisfied (section II-A), the first trial's reference signal must satisfy $r_0(1) \neq 0$.

III. MINIMIZING MOTION TIME USING A BISECTION ALGORITHM

This section discusses the bisection algorithm that calculates the minimal motion time and thus also the time-optimal set of time instants \mathcal{K}_d at which standstill can be imposed. This set is used by the model-free ILC algorithm at the second level to learn a system input that results in the time-optimal point-to-point motion.

Consider a point-to-point motion where a system has to reach the endpoint \mathbf{y}_{end} in m time samples starting from position $\mathbf{y}(0) = 0$. The desired system output $\mathbf{y}_d(k) = \mathbf{y}_{\text{end}}$ is only defined from time sample m to the end of the trial, that is for $k \in \mathcal{K}_d = \{m, m+1, \dots, N\}$. At every iteration of the learning process, the value of m is minimized using a bisection method [8]. This involves solving a series of convex optimization problems of the following form:

$$\begin{aligned} &\underset{\alpha_j \in \mathbb{R}^N}{\text{minimize}} && \|\alpha_j\|_2 \\ &\text{subject to} && \|\mathbf{y}_d(\mathcal{K}_d) - \hat{\mathbf{y}}_{j+1}(\mathcal{K}_d)\|_2 \leq e_{\text{des}} \sqrt{N-m+1} \\ &&& \hat{\mathbf{y}}_{j+1} = \mathbf{y}_j^m + (\mathbf{Y}_j^m - \mathbf{Y}_{j-1}^m + \gamma \mathbf{Y}_0^m) \alpha_j \\ &&& \mathbf{u}_{j+1} = \mathbf{u}_j + (\mathbf{U}_j - \mathbf{U}_{j-1} + \gamma \mathbf{U}_0) \alpha_j \\ &&& |\mathbf{u}_{j+1}| \leq \bar{\mathbf{u}}, \quad |\delta \mathbf{u}_{j+1}| \leq \bar{\delta \mathbf{u}}. \end{aligned} \quad (25)$$

In this optimization problem, e_{des} denotes the desired rms value of the positioning error and $N-m+1$ denotes the number of samples in \mathcal{K}_d . The output prediction is again based on the LTI property of the system as in section II. The aim of solving this optimization problem is determining whether it is possible to achieve the desired positioning accuracy e_{des} for a certain motion time of m time samples, taking into account the actuator constraints. For this reason, the constraint on $\|\alpha_j\|_2$, which limits the update to the actuator input, is removed. The ℓ_2 -norm of α_j is now minimized to avoid that the uncertainty on the predicted output becomes too large.

Depending on the feasibility of optimization problem (25), the value of m , the motion time of the point-to-point motion, is reduced or increased until the minimal value m^* is found. The bisection method finds the time-optimal set $\mathcal{K}_d = \{m^*, m^*+1, \dots, N\}$ for the desired endpoint \mathbf{y}_{end} , the desired positioning error e_{des} , and the actuator limitations $\bar{\mathbf{u}}$ and $\bar{\delta \mathbf{u}}$. At iterations $j \geq 1$, an estimate of the minimal value m^* is known from a previous iteration. This allows the number of steps of the bisection algorithm, and thus also the computation time, to be reduced by limiting the search interval of m^* .

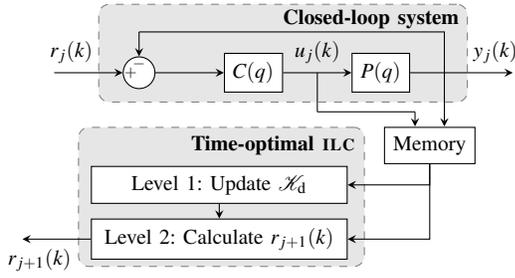


Fig. 3. Schematic representation of the two-level time-optimal ILC algorithm for a closed-loop system.

IV. LEARNING TIME-OPTIMAL POINT-TO-POINT MOTIONS

Fig.3 shows a schematic representation of the two-level time-optimal ILC algorithm for a closed-loop system with plant $P(q)$ and feedback controller $C(q)$. At every iteration j , the following steps are taken:

- First, a reference signal $r_j(k)$ is applied to the closed-loop system. For the first iteration, the reference signal $r_0(k)$ must satisfy $r_0(1) \neq 0$ (see section II).
- The actuator input $u_j(k)$ and the output $y_j(k)$ are stored in a memory.
- Level 1: The bisection algorithm estimates the minimal motion time taking into account the actuator bounds and the desired positioning error.
- Level 2: The ILC algorithm for point-to-point motions calculates the next iteration's reference signal $r_{j+1}(k)$.

As the learning process proceeds, the measured output y_j^m converges to the desired time-optimal trajectory, and the required update $U_{lc}\alpha_j$ to the actuator input decreases from trial to trial. As a result, the l_2 -norm of α_j , which is in the objective function of optimization problem (25), also decreases from trial to trial, resulting in more accurate estimates of the minimal motion time and therefore also convergence of the estimated minimal motion time.

Using the presented time-optimal ILC algorithm, it is also possible to trade off time-optimality with positioning accuracy by adjusting the value of e_{des} . In this case, however, it might be a more natural choice to minimize the l_∞ -norm of the positioning error instead of the l_2 -norm.

V. SIMULATION RESULTS

This section presents the results of a simulation test case on an accurate model of a linear motor. These results demonstrate the ability of the presented algorithm to learn time-optimal motion trajectories in the presence of measurement noise and cogging disturbances.

A. Motor model

The simulations make use of a discrete-time model ($T_s = 0.0025$ s) of a current-controlled linear motor with a position feedback controller. The model also includes cogging and clipping due to actuator constraints. Fig. 4 shows a block diagram of the closed-loop system, with cogging force $d_j(k)$.

Cogging is considered as the main disturbance in permanent-magnet linear motors. It is modelled as a combination of position ripple and force ripple. The position ripple is the force required to keep the carriage of a linear

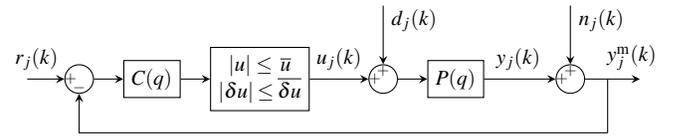


Fig. 4. The closed-loop system with cogging force $d_j(k)$ and actuator limitations \bar{u} and $\bar{\delta u}$.

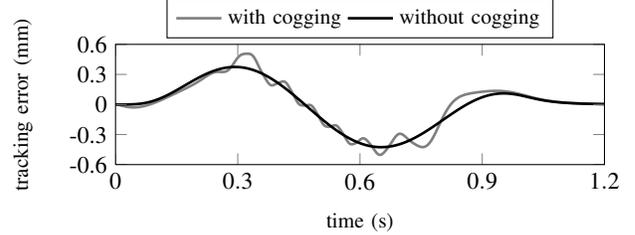


Fig. 5. Tracking error of the closed-loop system, with and without cogging disturbances, to a smooth reference step of 20cm.

motor at a fixed position, with zero motor input current. This disturbance force depends on the position only. 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 [9]. Since the input current is updated from trial to trial, the cogging disturbance is not entirely trial-invariant. The more the algorithm converges to the optimal solution, however, the smaller the trial-to-trial variation of the cogging disturbance will be. The system dynamics, including both force and position ripple, are given by the following equations:

$$\begin{aligned}
 y_j &= PC(I_N + PC)^{-1}(r_j - n_j) \\
 &+ P(I_N + PC)^{-1} \left[\underbrace{a \sin(\omega_y y_j) + b \sin(3\omega_y y_j + \phi_1)}_{\text{position ripple}} \right. \\
 &\left. + \underbrace{u_j(c \sin(\omega_y y_j + \phi_2) + d \sin(3\omega_y y_j + \phi_3))}_{\text{force ripple}} \right], \\
 u_j &= clip(C(r_j - y_j - n_j)),
 \end{aligned}$$

where $clip(\cdot)$ is a nonlinear function that represents the clipping due to the actuator constraints, ω_y is the spatial frequency determined by the width of the permanent magnets, and $\phi_1, \phi_2, \phi_3, a, b, c$ and d are parameters that determine the size and shape of the cogging force. Fig. 5 shows the tracking error to a smooth reference step of 20cm with and without cogging disturbance. This figure shows that a significant part of the tracking error is due to the cogging disturbance.

B. Results

The linear motor is commanded to travel a distance of 20cm time-optimally, satisfying the following actuator constraints:

$$|u_j(k)| \leq 10 \text{ A} \quad \text{and} \quad |\delta u_j(k)| \leq 5 \text{ A}. \quad (27)$$

During the first trial of the learning process a reference signal, satisfying $r_0(1) \neq 0$, is applied to the system. When calculating the second trial's reference signal (trial $j = 1$), update law (7) is used in optimization problems (23) and (25)

TABLE I

ESTIMATED MINIMAL MOTION TIME m^* (IN NUMBER OF SAMPLES) AS A FUNCTION OF THE TRIAL NUMBER AND THE THEORETICAL OPTIMUM FOR THE THREE CONSIDERED TEST CASES.

case	trial 1	trial 2...15	optimum
I	95	95	95
II	96	95	95
III	96	95	95

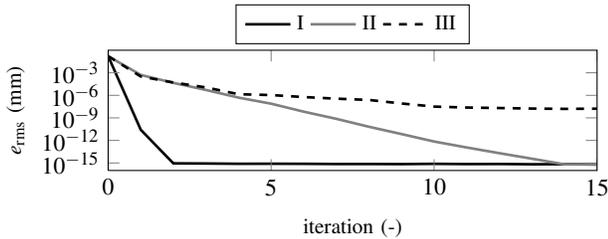


Fig. 6. Rms value of the positioning error as a function of the iteration number for the three considered test cases.

because experimental data from only one previous iteration are available. From then on, both the first and second level of the model-free time-optimal ILC algorithm use the update law given by (19) with $\gamma=0.1$. The upperbound t on $\|\alpha_j\|_2$ in optimization problem (23) is equal to 1.5.

Three cases are considered.

- Case I only considers the closed-loop system with actuator constraints.
- Case II includes cogging disturbances, but the output measurements are free of noise.
- Case III considers both cogging disturbances and noise-corrupted output measurements ($\sigma_n = 0.01 \mu\text{m}$).

The desired positioning accuracy e_{des} is chosen equal to 1 nm for the three cases. A smaller value for e_{des} results in the same motion time, which can be explained by the finite sampling time $T_s = 0.0025 \text{ s}$. For the third case, the minimal positioning error after learning is greater than e_{des} due to the measurement noise.

Table I shows the estimated minimal motion time m^* as a function of the trial number for the three different test cases. The theoretical minimal value of m^* for the simulated discrete-time system is also given. These results show that the estimated minimal motion time m^* converges to the true minimal value in very few iterations.

Fig. 6 shows the rms value of the positioning error as a function of the iteration number for these three situations. The presented algorithm achieves a zero positioning error in case of noise-free output measurements (case I & II). When measurement noise is present (case III), the positioning error after learning has a peak value of $0.033 \mu\text{m}$ and an rms value of $0.015 \mu\text{m}$. Further improvement is hardly possible because of the noise level. Fig. 7 shows the learned time-optimal output motion $y_{15}(k)$ of the 15th trial together with the corresponding actuator input $u_{15}(k)$, which satisfies the actuator limitations given by (27). These results show that the presented algorithm is able to learn time-optimal point-to-point motion trajectories in the presence of measurement noise and cogging disturbances.

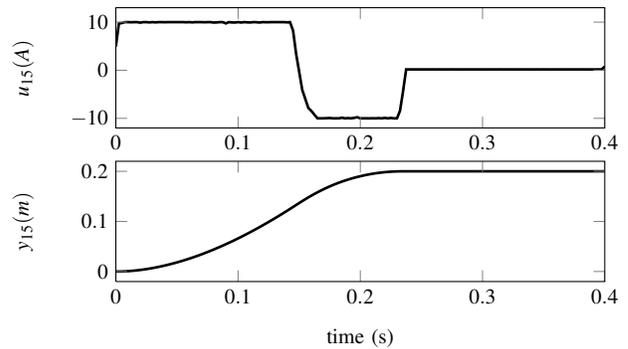


Fig. 7. Actuator input and output motion after 15 trials of learning in case of measurement noise and cogging disturbances (Case III).

VI. CONCLUSION

This paper presents a two-level model-free ILC method that learns time-optimal point-to-point motions for LTI systems. The bisection algorithm at the first level minimizes the motion time subject to actuator limitations. At the second level, a model-free ILC algorithm for point-to-point motions learns the system input that results in a point-to-point motion with the minimal motion time, calculated at the first level. Simulation results using an accurate model of a linear motor show that (i) the minimal motion time is learned in very few iterations, (ii) the ILC algorithm for point-to-point motions learns the optimal system input, even when measurement noise and repeating disturbances are present.

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] D. A. Bristow and A. G. Alleyne, "A manufacturing system for microscale robotic deposition," in *Proceedings of the American Control Conference*, June 4-6 2003.
- [3] M. Heertjes and T. Tso, "Nonlinear iterative learning control with applications to lithographic machinery," *Control Engineering Practice*, vol. 15, pp. 1545–1555, 2007.
- [4] J. van de Wijdeven and O. Bosgra, "Residual vibration suppression using hankel iterative learning control," in *Proceedings of the American Control Conference*, 2006.
- [5] P. Janssens, G. Pipeleers, and J. Swevers, "Model-free iterative learning control for LTI systems with actuator constraints," in *Proceedings of the 18th IFAC World Congress*, 2011.
- [6] —, "Model-free iterative learning control for LTI systems and experimental validation on a linear motor test setup," in *Proceedings of the American Control Conference*, 2011.
- [7] H.-S. Ahn, K. L. Moore, and Y. Chen, *Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems*. Springer-Verlag, 2007.
- [8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [9] 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.

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 (LECO PRO) 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.