Citation	<p>Pieter Janssens, Wannes Van Loock, Goele Pipeleers, Frederik Debrouwere and Jan Swevers</p> <p>Iterative learning control for optimal path following problems</p> <p>IEEE Conference on Decision and Control edition:52 location:Florence, Italy date:10-13 December 2013</p>
Archived version	<p>Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher</p>
Published version	<p>insert link to the published version of your paper</p> <p>...</p>
Journal homepage	<p>insert link to the journal homepage of your paper ...</p>
Author contact	<p>your email Frederik.Debrouwere@kuleuven.be</p> <p>your phone number + 32 (0)16 322539</p>
IR	<p>url in Lirias https://lirias.kuleuven.be/handle/123456789/414165</p>

(article begins on next page)

Iterative learning control for optimal path following problems

Pieter Janssens, Wannes Van Loock, Goele Pipeleers, Frederik Debrouwere and Jan Swevers
Department of Mechanical Engineering, Div. PMA, Katholieke Universiteit Leuven,
Celestijnenlaan 300B, Heverlee B3001, Belgium
email: Jan.Swevers@mech.kuleuven.be

Abstract—In optimal path following problems the motion along a given geometric path is optimized according to a desired objective while accounting for the system dynamics and system constraints. In the case of time-optimal path following, for example, the system input to move along the geometric path in minimal time is computed. In practice however, due to model-plant mismatch, (i) the geometric path is not followed exactly, and (ii) the optimized trajectory might be suboptimal, or even infeasible for the true plant. Assuming that the system performs the task repeatedly, this paper proposes an iterative learning control approach to improve the path following performance. The proposed learning algorithm is experimentally validated for a time-optimal path following problem on an XY-table. The results show that the developed ILC approach improves both the execution time and the accuracy significantly.

I. INTRODUCTION

In contrast to tracking control where a reference specified in time has to be tracked, in path following problems the reference to be followed is a geometric path without any preassigned timing information. The aim of optimal path following is to compute the optimal timing of the motion along the geometric path according to a desired objective while accounting for the system dynamics and system constraints.

Optimal path following problems arise in many industrial applications such as robotic welding, painting, or laser cutting [1], [2], [3]. In addition, path following is also used as a low level control strategy in decoupled optimal control [4], [5], [6]. In this case, a high level path planning algorithm determines a geometric path taking into account geometric constraints, e.g. obstacle avoidance, while a low level path following algorithm optimizes the motion along the path taking into account the system dynamics and constraints. Examples of the latter application of optimal path following include control of autonomous vehicles [5], [7], as well as control of chemical reactors [6].

In any practical application, model-plant mismatch is inevitable and as a result (i) the geometric path is not followed exactly, and (ii) the optimized motion trajectory as a function of time might be suboptimal, or even infeasible for the true plant. In the case of time-optimal path following, for example, either the motion along the geometric path can be executed faster by the true plant than by the assumed system dynamics, or the optimized motion trajectory is infeasible for the true plant.

In the field of tracking control, the performance of systems

executing the same task under the same operating conditions repeatedly can typically be improved using an iterative learning control (ILC) algorithm [8]. The system input is updated iteratively, i.e. from iteration to iteration, to improve the tracking accuracy of the desired motion trajectory. A conventional ILC algorithm, however, can only be used to improve the tracking performance, not to update the timing information optimized by the path following algorithm. This means that even with the use of such an ILC algorithm, either there is still room for improvement with respect to the executing time of the time-optimal path following task, or the motion trajectory is still not tracked exactly due to the system limitations.

This paper proposes an ILC approach for path following problems that improves both the tracking accuracy and the timing information assigned to the geometric path through learning. The proposed algorithm is inspired by the observation in [9] that an ILC input update is equivalent to first computing a nonparametric model correction and subsequently solving an optimal control problem for the corrected system dynamics. Similarly, we propose to iteratively compute a nonparametric model correction based on the model-plant mismatch observed during the previous iteration, and subsequently solve the same optimal path following problem for the corrected model. This way, both the system input and the timing information assigned to the geometric path are updated between every two iterations.

The idea of using ILC in path following applications is already introduced in [10]. The spatial-based ILC algorithm proposed in [10] updates the control signals using the tracking error as a function of the progress along the path instead of using the tracking error as a function of time, as is the case in conventional ILC. Contrary to the ILC proposed in this paper, the learned inputs are limited to be on-off, bang-bang velocity references, and the ILC algorithm in [10] is not based on optimal path following problems. As a result, no other system limitations than velocity constraints are taken into account, and it is not possible to trade off the duration of the task against other objectives such as input effort.

Optimal path following problems for robotic manipulators can be solved very efficiently. In [1], the optimal path following problem for the specific model structure of robotic manipulators was reformulated as a convex optimization problem through a nonlinear change of variables. This ap-

proach is extended in [11], [12] for the more general case of differentially flat systems [13], however, the resulting problem is no longer convex.

In this paper, the proposed ILC approach for optimal path following problems is developed for the specific case of an XY-table. The model structure of the XY-table is simple and allows to use the convex reformulation of [1]. Future work includes extending the proposed idea to a broader class of systems by combining it with the path following algorithm in [11] for differentially flat systems.

This paper is organized as follows. Section II presents the optimal path following problem for the system dynamics of an XY-table and gives some background on the reformulation of path following problems described in [1]. Next, Section III discusses the nonparametric model correction based on the measured system information from the previous iteration. The experimental validation in Section IV discusses a time-optimal contouring problem on an XY-table. Finally, Section V summarizes the conclusions of this paper.

II. OPTIMAL PATH FOLLOWING

This section first describes the optimal path following problem for an XY-table consisting of two linear motors, and subsequently reformulates the optimization problem into a convex optimization problem [1].

A. Problem description

The dynamics of a linear motor are modelled as a mass m subject to a driving force F , a position-dependent disturbance force D , and a Coulomb friction force, characterized by a Coulomb friction coefficient C . The decoupled equations of motion of the XY-table are given by:

$$\underbrace{\begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix}}_{\mathbf{F}(t)} = \underbrace{\begin{bmatrix} m_x & 0 \\ 0 & m_y \end{bmatrix}}_{\mathbf{m}} \underbrace{\begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix}}_{\ddot{\mathbf{q}}(t)} + \underbrace{\begin{bmatrix} C_x & 0 \\ 0 & C_y \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \text{sgn}(\dot{x}(t)) \\ \text{sgn}(\dot{y}(t)) \end{bmatrix}}_{\text{sgn}(\dot{\mathbf{q}}(t))} + \underbrace{\begin{bmatrix} D_x(x(t)) \\ D_y(y(t)) \end{bmatrix}}_{\mathbf{D}(\mathbf{q}(t))}, \quad (1)$$

where subscripts x and y refer to the X-motor and Y-motor, respectively.

The 2-D geometric path is denoted by $\mathbf{q}_r(s) = [x_r(s), y_r(s)]^T$ and is expressed as a function of a scalar path coordinate s . Hence, the spatial geometry of the path is determined by $\mathbf{q}_r(s)$, whereas the path's time dependency follows from the relation $s(t)$ between the path coordinate s and time t . Without loss of generality, it is assumed that the trajectory starts at $t = 0$, ends at $t = T$, and that $s(0) = 0 \leq s(t) \leq 1 = s(T)$. Furthermore, it is assumed that everywhere along the path $\dot{s}(t) > 0$, apart from a finite set of points where $\dot{s} = 0$, for instance at the beginning or end of the path.

The aim of solving an optimal path following problem is to determine the optimal timing $s(t)$ and the corresponding system inputs $\mathbf{F}(s(t)) = [F_x(s(t)), F_y(s(t))]^T$ such that:

- the geometric reference $\mathbf{q}_r(s) = [x_r(s), y_r(s)]^T$ is followed exactly by the system outputs $x(t)$ and $y(t)$,
- without violating the system limitations such as actuator constraints, velocity constraints, etc.,
- while minimizing the desired cost function. Typically, the cost function weighs the duration T of the path following task against other performance criteria, such as energy consumption or regularization terms, for instance to avoid large torque jumps [1].

For the considered example of time-optimal path following on an XY-table subject to lower and upper bounds on the input force, the optimal path following problem is defined as:

$$\underset{T, s(\cdot), \mathbf{F}(\cdot)}{\text{minimize}} \quad T \quad (2a)$$

$$\text{subject to} \quad \mathbf{F}(t) = \mathbf{m}\ddot{\mathbf{q}}_r(s(t)) + \mathbf{C} \text{sgn}(\dot{\mathbf{q}}_r(s(t))) + \mathbf{D}(\mathbf{q}_r(s(t))), \quad (2b)$$

$$s(0) = 0, \quad (2c)$$

$$\dot{s}(0) = 0, \quad (2d)$$

$$s(T) = 1, \quad (2e)$$

$$\dot{s}(T) = 0, \quad (2f)$$

$$\dot{s}(t) \geq 0, \quad (2g)$$

$$\underline{\mathbf{F}} \leq \mathbf{F}(t) \leq \overline{\mathbf{F}}, \quad (2h)$$

$$\text{for } t \in [0, T].$$

In (2), the duration T is minimized subject to a set of linear equality constraints containing the system dynamics (2b), the initial conditions (2c), (2d), and the terminal conditions (2e), (2f), which impose that the motion starts and ends with zero velocity. In addition, (2) contains a set of linear inequality constraints to ensure that $\dot{s}(t) \geq 0$ (2g), and that the actuator input is within its bounds (2h).

B. Problem reformulation

This subsection discusses the problem reformulation in [1] that yields a convex optimization problem for the model structure (1) of the considered system. For a given geometric path $\mathbf{q}_r(s)$, the velocities $\dot{\mathbf{q}}_r(s(t)) = [\dot{x}_r(s(t)), \dot{y}_r(s(t))]$ and accelerations $\ddot{\mathbf{q}}_r(s(t)) = [\ddot{x}_r(s(t)), \ddot{y}_r(s(t))]$ of the two linear motors are rewritten as a function of $\dot{s}(t)$ and $\ddot{s}(t)$ using the chain rule:

$$\dot{\mathbf{q}}_r(s(t)) = \mathbf{q}'_r(s)\dot{s}(t), \quad (3a)$$

$$\ddot{\mathbf{q}}_r(s(t)) = \mathbf{q}'_r(s)\ddot{s}(t) + \mathbf{q}''_r(s)\dot{s}(t)^2, \quad (3b)$$

where $\mathbf{q}'_r(s) = \frac{\partial \mathbf{q}_r(s)}{\partial s}$, and $\mathbf{q}''_r(s) = \frac{\partial^2 \mathbf{q}_r(s)}{\partial s^2}$. Combining (2b) and (3) results in the following expression for the equations of motion as a function of $\ddot{s}(t)$ and $\dot{s}(t)^2$:

$$\mathbf{F}(s) = \mathbf{m}\mathbf{q}'_r(s)\ddot{s}(t) + \mathbf{m}\mathbf{q}''_r(s)\dot{s}(t)^2 + \mathbf{C} \text{sgn}(\mathbf{q}'_r(s)) + \mathbf{D}(\mathbf{q}_r(s)), \quad (4)$$

where $\text{sgn}(\dot{\mathbf{q}}_r(s(t)))$ is replaced by $\text{sgn}(\mathbf{q}'_r(s))$ under the assumption that $\dot{s}(t) > 0$.

Next, the new optimization variables

$$a(s) = \ddot{s}(t), \quad (5a)$$

$$b(s) = \dot{s}(t)^2, \quad (5b)$$

are introduced together with the additional constraint

$$b'(s) = 2a(s), \quad (6)$$

which follows from the relations

$$\dot{b}(s(t)) = b'(s)\dot{s}(t), \quad (7a)$$

$$\dot{b}(s(t)) = \frac{d(\dot{s}(t)^2)}{dt} = 2\ddot{s}(t)\dot{s}(t) = 2a(s)\dot{s}(t). \quad (7b)$$

Finally, the objective function (2a) is rewritten by changing the integration variable from t to s :

$$T = \int_0^T 1dt = \int_{s(0)}^{s(T)} \frac{1}{\dot{s}(t)} ds = \int_0^1 \frac{1}{\sqrt{b(s)}} ds. \quad (8)$$

Using (4), (5), (6), and (8), the time-optimal path following optimization problem (2) for an XY-table subject to lower and upper bounds on the input force is now reformulated as the following optimization problem:

$$\begin{aligned} & \underset{a(\cdot), b(\cdot), \mathbf{F}(\cdot)}{\text{minimize}} \quad \int_0^1 \frac{1}{\sqrt{b(s)}} ds, \end{aligned} \quad (9a)$$

$$\begin{aligned} & \text{subject to} \quad \mathbf{F}(s) = \mathbf{m}\mathbf{q}_r'(s)a(s) + \mathbf{m}\mathbf{q}_r''(s)b(s) \\ & \quad + \mathbf{C} \operatorname{sgn}(\mathbf{q}_r'(s)) + \mathbf{D}(\mathbf{q}_r(s)), \end{aligned} \quad (9b)$$

$$b(0) = \dot{s}_0^2, \quad (9c)$$

$$b(1) = \dot{s}_T^2, \quad (9d)$$

$$b'(s) = 2a(s), \quad (9e)$$

$$b(s) \geq 0, \quad (9f)$$

$$\underline{\mathbf{F}} \leq \mathbf{F}(s) \leq \overline{\mathbf{F}}, \quad (9g)$$

$$\text{for } s \in [0, 1].$$

This optimization problem is convex since all constraints (9b)-(9g) are linear in the optimization variables, and the objective function is convex in $b(s)$ [14]. A numerical solution method for (9), and a discussion on other objective functions, regularization terms and constraints that preserve convexity of the optimal path following problem can be found in [1].

Once the optimal solution $a(s)$, $b(s)$, $\mathbf{F}(s)$ is obtained, the relation between the path coordinate and time can be constructed from the inverse relation $t(s)$, which is calculated as:

$$t(s) = \int_0^s \frac{1}{\sqrt{b(\sigma)}} d\sigma. \quad (10)$$

In order to perform an experiment with a fixed sampling rate, the optimal system inputs $F_x(s(t))$, $F_y(s(t))$ and corresponding desired motion trajectories $x_r(s(t))$, $y_r(s(t))$ are interpolated on an equidistant time grid. Note that the resulting interpolation error is negligible provided that the sampling period is sufficiently small.

III. ITERATIVE LEARNING NONPARAMETRIC MODEL CORRECTION

As shown in [9], a classical ILC input update is equivalent to first computing a nonparametric model correction and subsequently solving an optimal control problem for the corrected system dynamics. The proposed ILC approach for optimal path following problems is based on this insight and performs the following four steps in each iteration.

- 1) First, the applied input signals and the measured output signals are used to update the nonparametric model correction $\alpha(k)$ as a function of discrete time.
- 2) Next, the nonparametric model correction is converted to path coordinate domain by interpolating $\alpha(k)$ on the time instances $t(s)$ obtained in step 3 of the previous iteration. This means that the nonparametric model correction is considered to be a position-dependent correction force.
- 3) Then, optimal path following problem (9) is solved for the corrected model equation:

$$\begin{aligned} \mathbf{F}(s) = & \mathbf{m}\mathbf{q}_r'(s)a(s) + \mathbf{m}\mathbf{q}_r''(s)b(s) \\ & + \mathbf{C} \operatorname{sgn}(\mathbf{q}_r'(s)) + \mathbf{D}(\mathbf{q}_r(s)) + \boldsymbol{\alpha}(\mathbf{q}_r(s)), \end{aligned} \quad (11)$$

and $t(s)$ is computed using (10).

- 4) Finally, the optimal system inputs $F_x(s(t))$, $F_y(s(t))$ and corresponding desired motion trajectories $x_r(s(t))$, $y_r(s(t))$ are interpolated on an equidistant time grid to perform an experiment with a fixed sampling rate.

This section details the first two steps of this ILC approach whereas the final two steps were discussed in Section II.

After each experiment, the applied input signals and the measured output signals are used to update $\boldsymbol{\alpha}(\mathbf{q}_r(s))$ in the model equation (11). However, these are discrete-time signals whereas (11) is a model equation in path coordinate domain derived from a continuous-time model. For this reason, the continuous-time system model (1) needs to be discretized. In order to do so, we first define the net forces u_x , and u_y applied to the mass of the X- and Y-motor, respectively:

$$u_x(t) = F_x(t) - C_x \operatorname{sgn}(\dot{x}(t)) - D_x(t) - \alpha_x(t), \quad (12a)$$

$$u_y(t) = F_y(t) - C_y \operatorname{sgn}(\dot{y}(t)) - D_y(t) - \alpha_y(t), \quad (12b)$$

and rewrite the continuous-time system equations (1):

$$u_x(t) = m_x \ddot{x}(t), \quad (13a)$$

$$u_y(t) = m_y \ddot{y}(t). \quad (13b)$$

Next, the zero-order hold method is used to discretize the continuous-time system equations (13), resulting in the following difference equations:

$$\begin{aligned} x(k) - 2x(k-1) + x(k-2) = & \frac{T_s^2}{2m_x} (u_x(k-1) + u_x(k-2)), \\ y(k) - 2y(k-1) + y(k-2) = & \frac{T_s^2}{2m_y} (u_y(k-1) + u_y(k-2)), \end{aligned} \quad (14)$$

where

$$\begin{aligned} u_x(k) &= F_x(k) - C_x \operatorname{sgn}(x(k) - x(k-1)) \\ &\quad - D_x(k) - \alpha_x(k), \\ u_y(k) &= F_y(k) - C_y \operatorname{sgn}(y(k) - y(k-1)) \\ &\quad - D_y(k) - \alpha_y(k), \end{aligned} \quad (15)$$

are the net forces, interpolated on the discrete time grid.

The applied system inputs $F_x(k)$, $F_y(k)$ and the measured outputs $x(k)$, $y(k)$ are substituted in (15) and in the discretized system model equations (14). In the case of perfect plant knowledge, evidently, the residual of the model equations is zero. In the case of model-plant mismatch, however, the residual of the model equations, which is a measure for the model-plant mismatch, is non-zero. However, the residual, and thus also the model-plant mismatch, can be reduced by allowing the nonparametric model corrections $\alpha_x(k)$ and $\alpha_y(k)$ in (15) to vary from their estimate in the previous iteration. Although this nonparametric model correction is assumed to be position-dependent, the model-plant mismatch or the unmodelled dynamics they aim to compensate are not necessarily position-dependent as well. Obviously, convergence of the ILC approach can only be expected when the original model (1) captures the main part of the dynamics of the system. To what extent the model is allowed to vary from the true system dynamics for the ILC still to converge is a topic for future research.

The updates $\Delta\alpha_x(k)$, $\Delta\alpha_y(k)$ to the nonparametric model corrections $\alpha_x(k)$, $\alpha_y(k)$ are obtained by solving the following least-squares optimization problems:

$$\begin{aligned} \text{minimize}_{\Delta\alpha_x, R_x, u_x} \quad & \|R_x\|_2 + \lambda_{\Delta\alpha_x} \|\Delta\alpha_x\|_2 + \lambda_{\alpha_x} \|\alpha_x + \Delta\alpha_x\|_2 \\ & (16a) \end{aligned}$$

$$\begin{aligned} \text{subject to} \quad & u_x(k) = F_x(k) - C_x \operatorname{sgn}(x(k) - x(k-1)) - \\ & D_x(k) - (\alpha_x(k) + \Delta\alpha_x(k)), \\ & \text{for } k \in \{0, \dots, N-1\}, \\ & R_x(k) = x(k) - 2x(k-1) + x(k-2) - \\ & \frac{T_s^2}{2m_x} (u_x(k-1) + u_x(k-2)), \\ & \text{for } k \in \{2, \dots, N-1\}. \end{aligned} \quad (16b) \quad (16c)$$

$$\begin{aligned} \text{minimize}_{\Delta\alpha_y, R_y, u_y} \quad & \|R_y\|_2 + \lambda_{\Delta\alpha_y} \|\Delta\alpha_y\|_2 + \lambda_{\alpha_y} \|\alpha_y + \Delta\alpha_y\|_2 \\ & (17a) \end{aligned}$$

$$\begin{aligned} \text{subject to} \quad & u_y(k) = F_y(k) - C_y \operatorname{sgn}(y(k) - y(k-1)) - \\ & D_y(k) - (\alpha_y(k) + \Delta\alpha_y(k)), \\ & \text{for } k \in \{0, \dots, N-1\}, \\ & R_y(k) = y(k) - 2y(k-1) + y(k-2) - \\ & \frac{T_s^2}{2m_y} (u_y(k-1) + u_y(k-2)), \\ & \text{for } k \in \{2, \dots, N-1\}. \end{aligned} \quad (17b) \quad (17c)$$

In (16b) and (17b), $x(k)$ and $y(k)$ for $k < 0$ are assumed to be equal to $x(0)$ and $y(0)$, respectively.

Fig. 1. Control configuration of the x-motor.

The updated estimate of the position-dependent disturbance forces $\alpha_x(k) + \Delta\alpha_x(k)$ and $\alpha_y(k) + \Delta\alpha_y(k)$, respectively calculated by least-squares problems (16) and (17), which are computationally efficient by exploiting the structure, minimize the cost functions (16a) and (17a), respectively. These cost functions weigh the residuals R_x , R_y of the discretized model equations (14) against regularization terms on the iteration domain change in nonparametric model correction and the magnitude of the nonparametric model correction. Similarly to the tuning of a norm-optimal ILC [15], the tuning parameters $\lambda_{\Delta\alpha_x}$, $\lambda_{\Delta\alpha_y}$ regulate the trade-off between the convergence speed of the proposed iterative model updating procedure, on the one hand, and the sensitivity of the model correction to noise and iteration-varying disturbances, on the other hand. In some cases, an additional penalty term on the time domain difference of the nonparametric model correction might be required to further reduce the sensitivity of the model correction to measurement noise. The regularization terms on the magnitude of the total nonparametric model correction increase robustness by avoiding the model correction to grow too large, on the one hand, but also limit the performance after convergence, on the other hand.

IV. EXPERIMENTAL VALIDATION ON AN XY-TABLE

This section describes the experimental validation of the proposed algorithm on an XY-table. First, the system dynamics and constraints are discussed in Subsection IV-A. Then, Subsection IV-B presents the results of the experimental validation for a time-optimal contouring task.

A. Test setup

The XY-table consists of two position controlled linear motors. Fig. 1 shows the control configuration of the x-motor, the configuration of the y-motor is the same. The control scheme has two inputs, the optimized motor input $F_x(k)$ and the corresponding motion trajectory $x_r(k)$. Both signals are obtained by interpolating the solution of the optimal path following algorithm in Section II on an equidistant time grid with a sampling period of 1 ms. The output position $x(k)$ is measured by a linear encoder and fed back to a PI-lead position controller. In the case of model-plant mismatch or unmodelled disturbances, the output position $x(k)$ differs from the estimated output $x_r(k)$ such that the position controller adds a feedback signal to the optimized system input $F_x(k)$ resulting in a total applied motor input $F_x^{\text{tot}}(k) \neq F_x(k)$. Note that the nonparametric model correc-

	m/K_m (kg A N ⁻¹)	C/K_m (A s m ⁻¹)	$\bar{F}$ (A)
X-motor	0.17	0.14	2
Y-motor	0.065	0.08	0.8

TABLE I

NUMERICAL VALUES FOR THE SYSTEM PARAMETERS AND SYSTEM CONSTRAINTS OF THE XY-POSITIONING STAGE.

Fig. 2. Geometric path, starting position, and initial contouring error ($\times 100$) projected on the path.

tion step requires the total applied motor input $F_x^{\text{tot}}(k)$ and the measured output position $x(k)$.

The motor input is a current reference but can be considered as an input signal proportional to the motor force since the dynamics of the current loop are negligible. As a result, the system dynamics can be described by (1). Table I shows the numerical values of the model parameters m and C divided by the motor constant K_m for the two linear motors together with the numerical values of the input constraint. The model parameters are obtained using identification experiments.

B. Time-optimal contouring

The XY-table is commanded to track an elliptic geometric path, with a major axis of 24 cm in the Y-direction and a minor axis of 20 cm in the X-direction in minimal time. The carriage of the XY-table starts and arrives at the same position with zero velocity. The time-optimal path following algorithm optimizes the timing information assigned to the geometric path for the given system dynamics and constraints yielding an execution time of 0.674 s and a contouring error of 116 μm (rms). Fig. 2 shows the geometric path, the starting position, and the initial contouring error ($\times 100$) projected on the path.

The proposed ILC approach iteratively updates the system equations resulting in better tracking performance and a better timing of the motion along the geometric path. Table II

Fig. 3. Rms-value of the contouring error and estimated minimal execution time as a function of the iteration number.

Fig. 4. Remaining contouring error during the 20th iteration.

shows the numerical values of the tuning parameters in the least-squares optimization problems (16) and (17), for the X-motor and Y-motor, respectively. Fig. 3 shows the rms-value of the contouring error, and the estimated minimal execution time T as a function of the iteration number.

After 15 iterations of learning, the proposed ILC algorithm for time-optimal path following problems converged yielding an execution time of 0.547 s and a contouring error of 8 μm (rms). In an attempt to further reduce the contouring error, the desired motion trajectories $x_r(k)$ and $y_r(k)$ are kept fixed, and a norm-optimal ILC algorithm for constrained systems is used [16]. After 5 more iterations of learning using the constrained norm-optimal ILC algorithm, the contouring error is further reduced to only 3.5 μm (rms). Fig. 4 shows the remaining contouring error of the 20th iteration while Fig. 5 shows the contouring error ($\times 100$) projected on the path before and after learning. Fig. 6 shows the learned system inputs for the X-motor and Y-motor. These signals are non-

	λ_α (m ² A ⁻²)	$\lambda_{\Delta\alpha}$ (m ² A ⁻²)
X-motor	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$
Y-motor	$5 \cdot 10^{-6}$	$1.5 \cdot 10^{-5}$

TABLE II

NUMERICAL VALUES FOR THE TUNING PARAMETERS OF THE PROPOSED ILC APPROACH.

Fig. 5. Comparison of the contouring error ($\times 100$) projected on the path before and after learning.

Fig. 6. Learned input signals $F_x(k)$ and $F_y(k)$ for the 20th trial.

smooth to counter the cogging forces which are treated as a disturbance. Note that the learned system inputs resemble a generalized bang-bang solution.

V. CONCLUSION

This paper presents an iterative learning control approach for optimal path following problems, and hereby extends the applicability of iterative learning control. Contrary to classical iterative learning control algorithms, the proposed algorithm iteratively optimizes both the tracking performance, and the timing of the motion along the geometric path through learning. The proposed algorithm is experimentally validated for a time-optimal contouring task on an XY-table. The results show the benefits of iterative learning control for optimal path following problems. Both the accuracy and the execution time of the contouring task are improved significantly through learning from previous iterations. Future work

includes extending the proposed idea to a broader class of systems.

ACKNOWLEDGMENT

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.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), from the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, and from K.U.Leuven's Concerted Research Action GOA/10/11 "Global real-time optimal control of autonomous robots and mechatronic systems". Goele Pipeleers is Postdoctoral Fellow of the Research Foundation - Flanders.

REFERENCES

- [1] D. Verscheure, "Contributions to contact modeling and identification and optimal robot motion planning," Ph.D. dissertation, KU Leuven, 2009.
- [2] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.
- [3] F. Debruyere, W. Van Loock, G. Pipeleers, Q. Tran Dinh, M. Diehl, J. De Schutter, and J. Swevers, "Time-optimal path following for robots with trajectory jerk constraints using sequential convex programming," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013.
- [4] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.
- [5] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter trajectory tracking control," *Electrical Engineering*, vol. 44, no. August, pp. 1–14, 2008.
- [6] T. Faulwasser, V. Hagenmeyer, and R. Findeisen, "Optimal exact path-following for constrained differentially flat systems," in *Proc. of 18th IFAC World Congress, Milano, Italy*, 2011, pp. 9875–9880.
- [7] R. Skjetne, T. I. Fossen, and P. V. Kokotovic, "Robust output maneuvering for a class of nonlinear systems," *Automatica*, vol. 40, no. 3, pp. 373 – 383, 2004.
- [8] 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.
- [9] M. Volckaert, M. Diehl, and J. Swevers, "A two step optimization based iterative learning control algorithm," in *Dynamic Systems and Control Conference*, Cambridge, Massachusetts, 2010.
- [10] K. L. Moore, M. Ghosh, and Y. Chen, "Spatial-based iterative learning control for motion control applications," *Meccanica*, vol. 42, pp. 167–175, 2007.
- [11] W. Van Loock, G. Pipeleers, M. Diehl, and J. Swevers, "Optimal path following for differentially flat systems," *IEEE Transactions on Robotics*, 2012, submitted.
- [12] W. Van Loock, S. Bellens, G. Pipeleers, J. De Schutter, and J. Swevers, "Time-optimal parking and flying: Solving path following problems efficiently," in *IEEE International Conference on Mechatronics*, Vicenza, Italy, 2013.
- [13] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.
- [14] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.
- [15] R. Tousain, E. van der Meché, and O. Bosgra, "Design strategies for iterative learning control based on optimal control," *Selected Topics in Signals, Systems and Control*, vol. 12, September 2001.
- [16] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Transactions on Control Systems Technology*, 2012.