Iterative Learning Control with Variable Pass Length Applied to Trajectory Tracking on a Crane with Output Constraints

Mickaël Guth, Thomas Seel, Jörg Raisch

***Abstract*— A typical application of Iterative Learning Control (ILC), namely trajectory tracking on a lab-scale gantry crane, is considered. However, the load is only allowed to move in the close proximity of the reference trajectory. Since these output constraints lead to disrupted trials, the pass length in this ILC system is not constant. In this contribution, we present new results on convergence and new combinations of methods for this class of ILC systems and apply them to the given application. Simulation and experimental results are provided which demonstrate that both maximum pass length and small tracking error can be achieved in very few iterations even in the presence of tight constraints and model uncertainties.**

I. ITERATIVE LEARNING CONTROL ON A GANTRY CRANE

Iterative Learning Control (ILC) is a control method which aims at improving the performance of systems that execute the same task repeatedly. To this end, an input trajectory, which is typically applied in a feedforward fashion during an iteration, is updated from iteration to iteration based on error information from the last trial. ILC has been successfully used in many application such as robotics [4], [11], biomedical systems [3], [9], [12], chemical processes [7], steel production systems [8] and many more. A particularly challenging ILC application is trajectory tracking on a gantry crane (or bridge crane). In [5], e.g., ILC for pick-and-place tasks on a three-dimensional gantry robot is considered. Satisfactory results are obtained within five iterations using norm-optimal ILC and within twenty iterations using P-type ILC. In [6], an ILC controller designed by linear matrix inequalities is presented. The results are validated experimentally on a gantry robot. Satisfactory trajectory tracking is achieved within less than twenty trials using a constant learning gain.

A classical and obvious approach to trajectory tracking is a standard state feed-back control. However, a feed-back loop can only be as fast as the model is precise. In this paper, we will face bad model knowledge and ask the system to achieve performance that is not achievable with a linear-quadratic regulator [1]. ILC has been chosen to increase the performance of such a system. This is a special problem that is relevant to a number of applications and we examine it on a specific example, i.e. trajectory tracking on a lab-scale gantry crane.

In the vast majority of the applications mentioned above, it is assumed that, even before the learning step, the system is able to complete a trial and that the complete output trajectory of that first trial can be obtained. In the often used analogy of human motor learning, this refers to a basketball player improving his hand and arm movement from throw to throw, or to a rally driver trying to get closer to the optimal trajectory from lap to lap. However, when young humans learn to walk, or when most people try to water ski for the first time, then they must first learn to complete the task and then improve performance rather than only doing the latter. Similarly, in some ILC application systems, the initial input, i.e., the input that is applied in the first trial, drives the system to violate constraints before the trial has completed and consequently, that trial is aborted before it is finished. In that case, the incomplete output information must nonetheless be used to improve the input trajectory such that, in the second trial, the constraint is not violated, or at least violated at a later (cycle) time.

Little to no attention has been given to such systems so far. E.g., it is not clear under which circumstances a learning algorithm leads to completion of a trial in a finite number of learning steps. Besides this, the presence of limiting constraints and an initial input that leads to their violation poses many research questions about the dynamics of the pass length and the tracking error in such systems. Most of these questions cannot be answered in the brevity of this contribution. We will, however, undertake a first step towards analyzing such systems by applying recent results [9] on variable pass length (VPL) dynamics to a standard ILC application. More precisely, we consider trajectory tracking with output constraints on a lab-scale gantry crane, as depicted in Figure 1. In Section IV, this application example is used to demonstrate how an ILC algorithm might learn from the incomplete information of a disrupted trial and still converge even with large disturbances. Prior to this, in Sections II and III, we will model the system, define the control task, present the different algorithms that are applied in Section IV. In that section, we analyze the resulting pass length and error dynamics using recent results on variable pass length ILC.

II. PRECISE PATH FOLLOWING ON GANTRY CRANES

The gantry crane (see Figure 1) is produced by InTeCo[1] and consists of a cart that can be moved in x- and y-direction. A load with mass m is attached at the end of a rope, which is attached to the cart. The rope can be wound up and down, thus allowing the load to move in a three-dimensional space.

M. Guth, T. Seel and J. Raisch are with Control Systems Group, Technical University, Berlin, Germany. M. Guth and J. Raisch are also with MPI, Magdeburg, Germany.
{guth,seel,raisch}@control.tu-berlin.de

[1] http://www.inteco.com.pl/

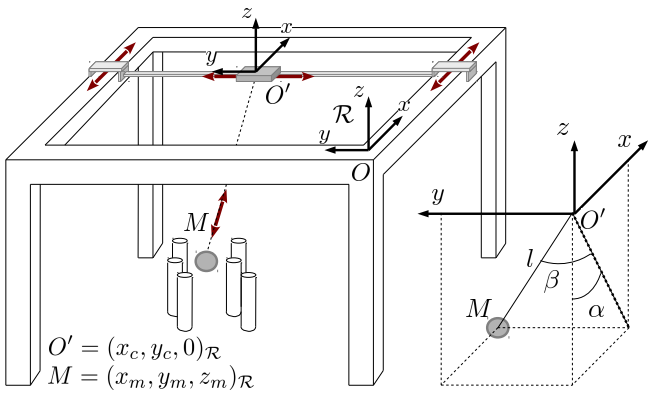

Fig. 1. Representation of the gantry crane that is used in the ILC experiments. The cart must be moved in x and y direction such that the load that is attached to the end of the rope travels through the obstacles without touching them. Whenever the load gets too close to an obstacle, the rope is wound up and the entire task must be repeated.

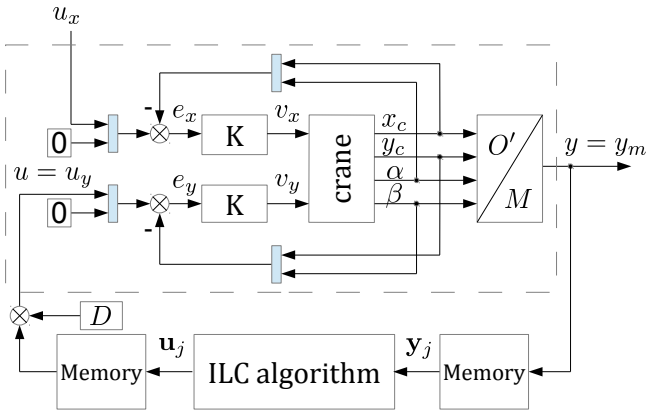

Fig. 2. Cascaded control structure. The block *crane* represents the dynamics of the crane and has two inputs, velocities v_x and v_y, which are computed by two identical linear-quadratic regulators K. The measured outputs are the positions of the cart and angles of the rope. They are used to compute the load's y-position. After each iteration j , the output is stored in a vector $\mathbf{y}_j$ and used to update the input $\mathbf{u}_j$ for the next trial $(j+1)$. The difference between a classical and a variable pass length (VPL) ILC lies in the *ILC algorithm* block. In the first case, the entire error vector is available, and in the second case, the algorithm learns from incomplete error information. The block D represents the possible disturbance that could occur after the *ILC algorithm* block. See Section IV for sensitivity analysis.

The x-y-position $(x_c, y_c)^T$ of the cart as well as the length l of the rope and the two-dimensional angle $(\alpha, \beta)^T$ with respect to vertical axis are measured online at a frequency of $f_s = 20\,\text{Hz}$. At the same rate, the speed of the cart in both x- and y-direction can be set. To simplify the dynamics that the ILC must face, we design a feedback controller [1] for the cart position. For this, a non-linear model of the crane, which was provided along with the crane, is derived and then linearized. Subsequently, a linear-quadratic regulator [1] K, consisting of a pole-placement based linear observer and state feedback, is computed (by solving a Riccati equation [1]) and implemented, as depicted in Figure 2.

For the ILC problem, we consider the discrete iteration time $k \in [1, \overline{n}]$, where $\overline{n}$ is called the (full) pass length or iteration duration. A reference path $y_d(x)$ is defined

TABLE I

NUMERICAL VALUES OF $G(z)$, THE TRANSFER FUNCTION GIVEN IN (1)

a_0	1	a_4	$-4.87 \cdot 10^{-1}$	b_0	$1.64 \cdot 10^{-3}$
a_1	-2.16	a_5	$-2.77 \cdot 10^{-3}$	b_1	$-1.90 \cdot 10^{-3}$
a_2	1.27	a_6	$3.04 \cdot 10^{-2}$	b_2	$1.81 \cdot 10^{-3}$
a_3	0.289	a_7	$5.94 \cdot 10^{-2}$	b_3	$1.52 \cdot 10^{-2}$
		a_8	$7.40 \cdot 10^{-2}$	b_4	$-1.55 \cdot 10^{-2}$
		a_9	$-2.18 \cdot 10^{-2}$	b_5	$1.46 \cdot 10^{-4}$
		a_{10}	$-3.78 \cdot 10^{-2}$	b_6	$8.64 \cdot 10^{-3}$

from a starting position (x_0, y_0) to a final position (x_f, y_f). This path shall be tracked by the load with constant rope length $l = 0.5\,\text{m}$ within a given time $T = 4\,\text{s}$. Therefore, $\overline{n} = f_s T = 80$. Any reference path $y_d(x)$ corresponds to an infinite number of pairs $(x(k), y_d(k))$. We chose an $x(k)$ that is monotonously increasing in time, implementable with the available actuators and fulfills $x(T) = x_\text{f}$. Subsequently, we compute the (unique) $y_d(k)$ which, in combination with $x(k)$, yields the desired path $y_d(x)$. Thus, the two-dimensional path following problem is reduced to a one-dimensional trajectory tracking task, i.e., learning to follow the desired y-profile $y_d(k)$.

In this ILC task, the input $u_j(k)$ that the ILC applies in the j^{th} trial is the reference for the load's y-position supplied to the inner control loop, see Figure 2. Accordingly, the measurement information for the ILC algorithm is $y_j(k)$,the load's y-position in the j^{th} trial, which is computed from the cart's y-position y_c, the rope angles $(\alpha, \beta)^T$ and length l. To use model-based ILC algorithms, we use a simple step test and standard system identification methods to approximate the input-output dynamics of the dashed box in Figure 2 by a linear time-discrete transfer function $G(z)$:

$$G(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_6 z^{-6}}{a_0 + a_1 z^{-1} + \ldots + a_{10} z^{-10}}. \tag{1}$$

The parameters a_i, b_i are calculated via least squares methods, see e.g. [10]. The numerical values are summarized in Table I. A comparison of the simulated and measured step response is shown in Figure 3. For each trial j, we require the cart and load to rest at (x_0, y_0). Furthermore, we stack the values of the input profile $u_j(k)$, $k = 1, ..., \overline{n}$, and output profile $y_j(k)$, $k = 1 + \phi, ..., \overline{n} + \phi$, into lifted [2] vectors $\mathbf{u}_j \in \mathbb{R}^{\overline{n}}$ and $\mathbf{y}_j \in \mathbb{R}^{\overline{n}}$, respectively, where $\phi \in \mathbb{N}$ is the number of samples that $\mathbf{u}_j$ is shifted backward[2] in time[3]. ϕ is called *phase lead* and is used to compensate for delays in the system dynamics. Once the lifted vectors are built, we use $G(z)$ to derive the following lifted system model of the input-output trial dynamics:

$$\mathbf{y}_j = \mathbf{P}\mathbf{u}_j + \mathbf{v} \;\forall j, \tag{2}$$

where $\mathbf{P} \in \mathbb{R}^{\overline{n} \times \overline{n}}$ is the lifted system matrix containing the system's Markov parameters p_i, $i = 1, ..., \overline{n} + \phi$, and $\mathbf{v} \in \mathbb{R}^{\overline{n}}$ is an unknown but iteration-invariant signal representing output disturbances as well as the effects of initial conditions.

[2]when needed, the input is set to $u_j(k) = 0 \,\forall k \in [\overline{n} + 1, \overline{n} + \phi]$

[3]i.e., $\mathbf{u}_j := [u_j(1), ..., u_j(\overline{n})]^T$ while $\mathbf{y}_j := [y_j(1+\phi), ..., y_j(\overline{n}+\phi)]^T$

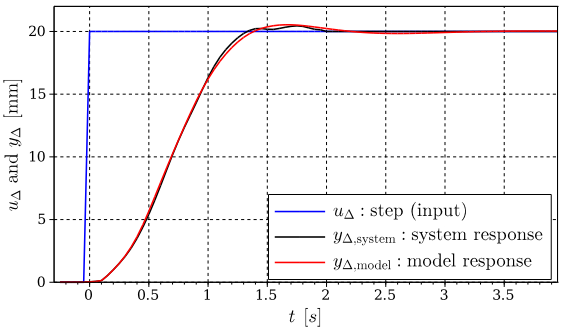

Fig. 3. Identification of the input-output dynamics of the dashed box in Figure 2. A step u_Δ is applied as input on the position control loop of the real system. We use standard least squares methods to fit the parameters of a linear transfer function $G(z)$ such that its response $y_{\Delta,\text{model}}$ to the same step is close to the real system's step response $y_{\Delta,\text{system}}$.

The Markov parameters p_i, $i = 1, ..., \overline{n}+\phi$, can be obtained from the impulse response of $G(z)$. The lifted system matrix is then constructed as follows:

$$\mathbf{P} = \begin{pmatrix} p_{1+\phi} & \cdots & p_1 & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & & 0 \\ \vdots & & & & \ddots & \vdots \\ p_{\overline{n}+\phi} & \cdots & & \cdots & & p_{1+\phi} \end{pmatrix}. \tag{3}$$

Finally, we also define a lifted reference vector $\mathbf{y_d} \in \mathbb{R}^{\overline{n}}$ that contains all samples of the desired y-profile $y_d(k)$, $k = 1+\phi, ..., \overline{n}+\phi$. In a standard ILC problem, the controller would try to find the input $\mathbf{u_d}$ that yields $\mathbf{y}_j = \mathbf{y_d}$. Typically, an input update law of the following form is employed:

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{L}(\mathbf{y_d} - \mathbf{y}_j)\ \forall j, \tag{4}$$

where $\mathbf{L}$ is the learning gain matrix that describes how tracking errors are mapped to changes in the input trajectory. Many design techniques for various convergence properties can be found in the literature, see e.g. [2]. However, all classic ILC methods assume that the trials are of constant length $\overline{n}$ and that each input update is based on full error information, i.e., that all samples of $(\mathbf{y_d} - \mathbf{y}_j)$ are known. In the following section, we examine what happens if this assumption is dropped.

III. ITERATIVE LEARNING CONTROL WITH OUTPUT CONTSTRAINTS

In this section, the desired path $y_d(x)$ is assumed to be surrounded by obstacles which the load must not have contact with. In order to avoid contact, output constraints are defined, as depicted in Figure 4. These bounds are translated into time dependent constraints $\underline{y}(k) < y(k) < \overline{y}(k)\ \forall k$ on the y-coordinate. Whenever these constraints are violated, the load is wound up and the input of the current ILC trial is no longer applied to the cart. We further assume that the constraints are so tight that even with a well-chosen initial input trajectory u_0, a violation occurs in the first trial. Thus, we obtain a constrained ILC problem in which the controller must learn from incomplete trial information.

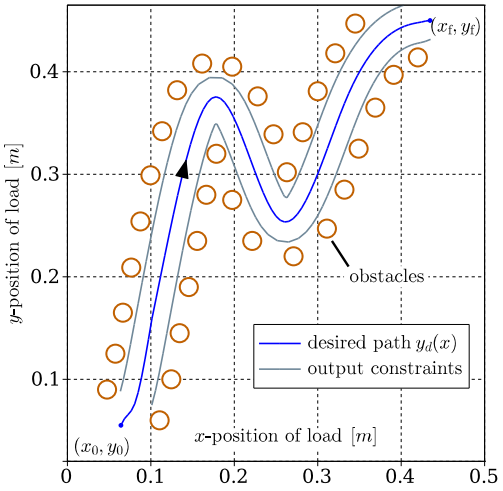

Fig. 4. The path which the load shall track in $T = 4\,\text{s}$ is defined from (x_0, y_0) to (x_f, y_f). The obstacles form output constraints. A band with radius $r = 2\,\text{cm}$ is defined, in which the load is supposed to travel when the variable pass length ILC algorithm is applied.

A. Variable Pass Length ILC Dynamics

In this variable pass length ILC system, the pass length n_j may vary from trial to trial within $n_j \in [1, \overline{n}]$, $\overline{n} = 80$. Since n_j is not known in advance, the controller must provide a full-length input $\mathbf{u}_j \in \mathbb{R}^{\overline{n}}$ in each trial. Nevertheless, each trial will end as soon as one of the constraints is violated or the maximum pass length $\overline{n}$ is reached. Therefore, the output $\mathbf{y}_j$ contains only n_j measurement samples, while the last $\overline{n} - n_j$ samples are unknown or undefined.

The key question in this context is how to update the input. In classic ILC systems, the controller learns from the lifted error vector $(\mathbf{y_d} - \mathbf{y}_j)$. Since this full error information is not available in our case, we use $\hat{\mathbf{e}}_j$ instead, which is defined as the first n_j samples of that error vector completed by the last $\overline{n} - n_j$ samples being zero:

$$\hat{\mathbf{e}}_j := \begin{bmatrix} y_d(1+\phi) - y_j(1+\phi) \\ \vdots \\ y_d(n_j+\phi) - y_j(n_j+\phi) \\ \\ \mathbf{0}_{(\overline{n}-n_j)\times 0} \end{bmatrix}. \tag{5}$$

We then use a standard ILC update equation of the form

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \mathbf{L}\hat{\mathbf{e}}_j. \tag{6}$$

This approach of setting the last $\overline{n} - n_j$ samples to zero has been used before in [9]. By doing so, we restrict the learning to the part of the output trajectory that we gained information about, i.e., the first n_j samples. If $\mathbf{L}$ is chosen adequately, then the output y_{j+1} of the next trial will improve in these samples, and the constraint violation will occur, if at all, at a larger cycle time. In other words, the ILC is "told" to improve tracking on the first n_j samples without paying attention to the last $\overline{n} - n_j$ samples of the output trajectory.

The issue of proper controller design and convergence analysis for variable pass length systems has been addressed

in [9]. There it was explained that, although only the first n_j samples of the error are used in the input update law, it is reasonable to use the *maximum pass length* (MPL) error for convergece analysis. More precisely, to examine monotonic convergence for a given $\mathbf{P}$ and $\mathbf{L}$ one should consider, for each trial, the error that would have been measured if that trial would not have been disrupted. It was found that, unlike the error vector $\hat{\mathbf{e}}_j$, the MPL error $\mathbf{e}_j$ is a proper measure of controller perfomance, since its values do not depend on the trial duration n_j. We use the following trial-to-trial MPL error dynamics that were derived in [9] and can be found by combining (2), (5), (6), and $\mathbf{e}_j := \mathbf{y_d} - \mathbf{y}_j$:

$$\mathbf{e}_{j+1} = (\mathbf{I} - \mathbf{PLH}_{n_j})\mathbf{e}_j, \tag{7}$$

where $\mathbf{H}_{n_j} = \text{blockdiag}\{\mathbf{I}_{n_j}, \mathbf{0}_{\overline{n}-n_j}\}$ and $\mathbf{I}_{n_j}$ and $\mathbf{0}_{\overline{n}-n_j}$ are, respectively, the identity matrix of dimension $n_j \times n_j$ and the zero matrix of dimension $(\overline{n}-n_j)\times(\overline{n}-n_j)$. Here, $\mathbf{H}_{n_j}$ is used to restrict the learning to the first n_j samples of the MPL error $\mathbf{e}_j$. In [9], it is found that the 1-norm[4] $||\mathbf{e}_j||_1$ of the MPL error is monotonic convergent if the induced matrix norm $\gamma := ||\mathbf{I} - \mathbf{PL}||_1$ is less than or equal to one. This guarantees that the MPL error does not increase from trial to trial. But it has not been examined how the error $\hat{\mathbf{e}}_j$ that is actually measured evolves from trial to trial. For the present application, it is of particular interest whether the output improves on the first n_j samples. The following theorem addresses this question.

Theorem 1 *In a variable pass length ILC system* (2)*,* (5)*,* (6)*, the error* $\hat{\mathbf{e}}_j$ *decreases on the first* n_j *samples by at least a factor* $\gamma := ||\mathbf{I} - \mathbf{PL}||_1$*, i.e.*

$$||\mathbf{H}_{n_j}\hat{\mathbf{e}}_{j+1}||_1 \leq \gamma||\hat{\mathbf{e}}_j||_1 \ \forall j. \tag{8}$$

Proof: From (7), we obtain:

$$\begin{aligned}\mathbf{H}_{n_j}\mathbf{e}_{j+1} &= \mathbf{H}_{n_j}(\mathbf{I}_{\overline{n}} - \mathbf{PLH}_{n_j})\mathbf{e}_j \\ &= \left[\mathbf{H}_{n_j}(\mathbf{H}_{n_j} - \mathbf{PLH}_{n_j}) + \mathbf{H}_{n_j}(\mathbf{I}_{\overline{n}} - \mathbf{H}_{n_j})\right]\mathbf{e}_j\end{aligned}$$

However, $\mathbf{H}_{n_j}(\mathbf{I}_{\overline{n}} - \mathbf{H}_{n_j}) = \mathbf{0}_{\overline{n}}\ \forall n_j$, therefore:

$$\mathbf{H}_{n_j}\mathbf{e}_{j+1} = \mathbf{H}_{n_j}(\mathbf{H}_{n_j} - \mathbf{PLH}_{n_j})\mathbf{e}_j = \mathbf{H}_{n_j}(\mathbf{I}_{\overline{n}} - \mathbf{PL})\hat{\mathbf{e}}_j$$

The induced matrix 1-norm is sub-multiplicative, thus:

$$\begin{aligned}||\mathbf{H}_{n_j}\mathbf{e}_{j+1}||_1 &\leq ||\mathbf{H}_{n_j}||_1||\mathbf{I}_{\overline{n}} - \mathbf{PL}||_1||\hat{\mathbf{e}}_j||_1 \\ &\leq \gamma||\hat{\mathbf{e}}_j||_1 \ \text{, because } ||\mathbf{H}_{n_j}||_1 = 1\end{aligned}$$

Note that if $n_{j+1} \geq n_j$, then $\mathbf{H}_{n_j}\hat{\mathbf{e}}_{j+1} = \mathbf{H}_{n_j}\mathbf{e}_{j+1}$, and if $n_{j+1} < n_j$, then $||\mathbf{H}_{n_j}\hat{\mathbf{e}}_{j+1}||_1 \leq ||\mathbf{H}_{n_j}\mathbf{e}_{j+1}||_1$. In both cases, $||\mathbf{H}_{n_j}\hat{\mathbf{e}}_{j+1}||_1 \leq ||\mathbf{H}_{n_j}\mathbf{e}_{j+1}||_1$. ■

The matrix $\mathbf{L}$ can be designed such that γ is minimal. (8) guarantees that the error (in the 1-norm) on the first n_j samples decreases by a factor γ. Therefore, as the deviation between $\mathbf{y}_j$ and $\mathbf{y_d}$ decreases on the first n_j samples, it is reasonable to assume that the constraints are violated at a later point in iteration time, i.e., n_j increases until $n_j = \overline{n}$. Once full pass length is achieved, the ILC algorithm learns from complete trials thus becoming a classical ILC problem, which we will call a constant pass length (CPL) ILC.

[4]the 1-norm for $x \in \mathbb{R}^n$ is given by $||x||_1 = \sum_{i=1}^{n}|x_i|$. The matrix norm induced by the 1-vector norm is $||A||_1 = \max_{1\leq j\leq n}\sum_{i=1}^{n}|a_{i,j}|$ for $A \in \mathbb{R}^{n\times n}$

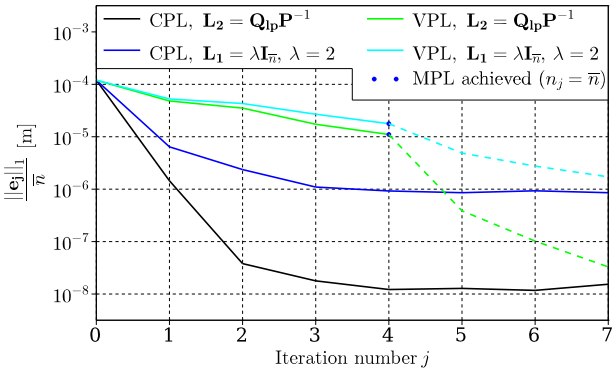

Fig. 5. Simulation results for the non-linear model. Scaled 1-norm of the error e_j depending on the iteration number j. As expected, when no constraint disrupts the trial (CPL or classical case), the convergence rate is very fast. However, when output constraints are considered (VPL case), the error decreases more slowly from iteration to iteration. As soon as the maximum pass length $n_j = \overline{n}$ is achieved, the convergence rate becomes the same as in the CPL case. Both learning gain matrices lead to a maximum pass length in four iterations but $\mathbf{L} = \mathbf{L_2} = \mathbf{P}^{-1}$ leads to significantly better precision than $\mathbf{L} = \mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$.

B. Design of Learning Gain Matrices

We design two learning gain matrices, $\mathbf{L_1}$ and $\mathbf{L_2}$. First, we choose a standard diagonal approach $\mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$. Here, λ is a scalar parameter that can be tuned to achieve fast convergence. The second learning gain matrix $\mathbf{L_2}$ is taken as the inverse of the lifted system matrix $\mathbf{P}$. The idea behind this inversion-based approach is that for $\mathbf{L} = \mathbf{P}^{-1}$, convergence on the first n_j samples is achieved in one iteration since $\gamma = 0$ in this case, which should lead to better results than for the diagonal learning gain approach. However, the inversion-based approach is known to be less robust with respect to uncertainties in the system dynamics (see e.g. [2]) and, for the variable pass length case, leads to high frequencies in the input trajectories. Therefore, we additionally apply a non-causal[5] (with respect to k) low-pass filter to $\mathbf{L_2}$, which results in $\mathbf{L_2} = \mathbf{Q}_{\text{lp}}\mathbf{P}^{-1}$. Unlike the classic Q-filter approach that is often considered in ILC (see e.g. [2]), this low-pass filtering refers to including a multiplicative factor in the learning gain matrix and, therefore, does not affect the steady-state error. To assess the performance of $\mathbf{L_1}$ and $\mathbf{L_2}$, the non-linear model is simulated and the results are provided in Figure 5 where the transient behavior and steady-state values of the error are evaluated for $\mathbf{L_1}$ and $\mathbf{L_2}$ in both the CPL (constant pass length, or classical ILC) and VPL (variable pass length) case.

[5]non-causality does not pose a problem in this context, as the entire update of the input profile is being done between iterations. Therefore, all of u_j is evaluated to compute u_{j+1}.

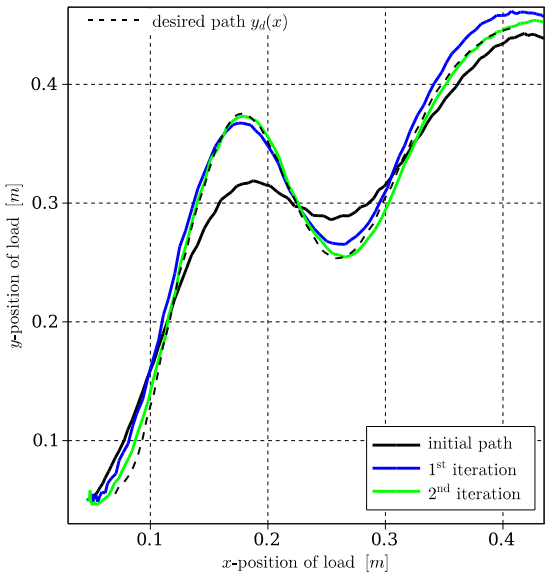

Fig. 6. Experiment of the constant pass length (or classical) ILC algorithm based on plant inversion $\mathbf{L_2} = \mathbf{Q}_{\text{lp}}\mathbf{P}^{-1}$ on the lab-scale gantry crane (no output constraints). The trajectory tracking is satisfactory after two iterations. This algorithm is slightly more precise than for $\mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$.

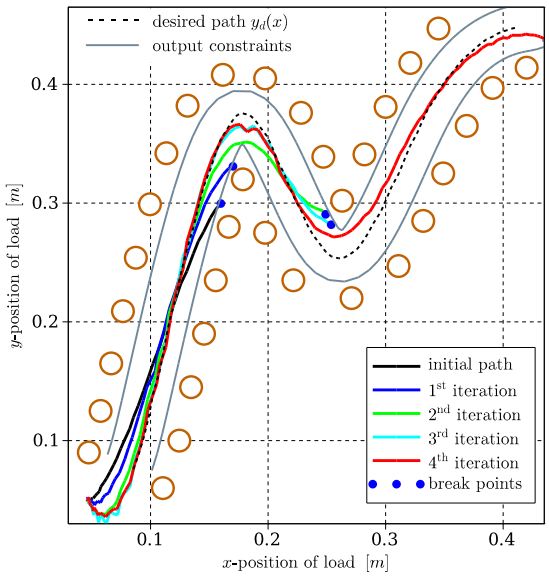

Fig. 7. Experiment of the variable pass length ILC algorithm with diagonal learning gain matrix $\mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$ on the lab-scale gantry crane ($\lambda = 2.0$, $r = 2\,\text{cm}$). The maximum pass length $n_j = \overline{n}$ is achieved in four iterations. This algorithm is slightly less precise than the one based on plant inversion.

IV. EXPERIMENTAL RESULTS

The ILC controllers are used on the real gantry crane. Since the inner controller is a position control loop (LQR), it is natural to choose $u_0(k) = y_d(k)$ for the first trial. If the inner loop were fast enough, no ILC would be needed and the desired path would be followed with satisfactory precision and without infringing the constraints. However, we ask the system to follow a path in a short time ($T = 4\,\text{s}$). Therefore the system dynamics are too slow and applying $u_0(k) = y_d(k)$ yields a first output $y_0(k)$, which is very different from $y_d(k)$.

The algorithms, for both $\mathbf{L_1}$ and $\mathbf{L_2}$, are first tested without output constraints. The trials are not disrupted and the algorithm learns from full trajectories. This corresponds to the classical approach of ILC. The ILC parameters are tuned properly to obtain fast convergence and precision. Satisfactory tracking is achieved after a few iterations. Figure 6 shows the results of this algorithm for $\mathbf{L_2} = \mathbf{Q}_{\text{lp}}\mathbf{P}^{-1}$, which leads to similar but slightly better results than for $\mathbf{L} = \mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$, in terms of precision (see Figure 8). Note that the performance of $\mathbf{L_2}$ (but not of $\mathbf{L_1}$) has deteriorated drastically when compared to the simulation results in Figure 5. This can be attributed to the known lack of robustness of the inversion-based approach with respect to model uncertainties.

Once this CPL (or classical) ILC problem is solved, we include constraints, as explained in Figure 4. In this case, the trial is stopped as soon as the constraints ($\underline{y}(k) < y(k) < \overline{y}(k)\,\forall k$) are infringed. Since a disruption occurs on the first trial, it is expected that the convergence rate will be smaller than for the case where no output constraints are included, as predicted by the simulation in Section III Figure 5. The convergence conditions $||\mathbf{e}_{j+1}||_1 \leq ||\mathbf{e}_j||_1$ and $n_{j+1} \geq n_j$ need to be checked experimentally to make a statement on the monotonic convergence of this algorithm. In Figure 8, the scaled error $\frac{||\mathbf{e}_{j+1}||_1}{\overline{n}}$ is plotted over j.

In the experimental results, for both $\mathbf{L_1}$ and $\mathbf{L_2}$, the same initial input is applied ($u_0(k) = y_d(k)$) as well as the same constraints ($r = 2\,\text{cm}$). Figure 7 shows these results for the first controller, with $\mathbf{L} = \mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$, where $\lambda = 2.0$. The trial is disrupted as soon as the load leaves the band (break points). After four iterations, the maximum pass length ($n_j = \overline{n}$) is achieved, as predicted by the simulation (see Figure 5). We can conclude that the algorithm learns from trial to trial how to track the desired profile $y_d(k)$.

The same experiment is made for the second controller, with $\mathbf{L} = \mathbf{L_2} = \mathbf{Q}_{\text{lp}}\mathbf{P}^{-1}$. The radius for the constraints is unchanged. The convergence is slightly faster than for the first controller, but the maximum pass length is achieved in four iterations, like for the diagonal learning gain $\mathbf{L_1}$. In terms of precision, this algorithm is slightly more efficient, as shown in Figure 8. It shows the convergence rates of both algorithms with (VPL case) and without (CPL case) output constraints. As expected, when no constraint disrupts the trial, the convergence rate is very fast. When output constraints are considered, the error is reduced more slowly from trial to trial and as soon as $n_j = \overline{n}$, the convergence rate becomes the same as for the CPL case. Unlike in the simulation results (Figure 5), the inversion-based algorithm is only slightly more efficient in terms of precision in the experiment than the diagonal learning gain approach. Again this can be attributed to the known lack of robustness of this approach with respect to model uncertainties.

It is interesting to look at the effect of a disturbance D on the learning process. We focus our sensitivity analysis on

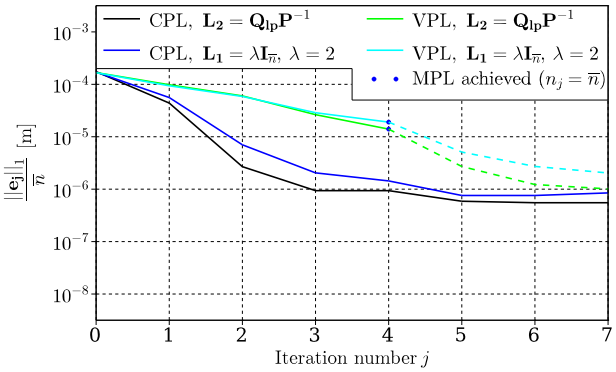

Fig. 8. Scaled 1-norm of the error e_j depending on the iteration number j. As expected, when no constraint disrupts the trial (CPL case), the convergence rate is very fast. However, when output constraints are considered (VPL case), the error decreases more slowly from iteration to iteration. As soon as the maximum pass length $n_j = \overline{n}$ is achieved, the convergence rate becomes the same as in the CPL case. Both learning gain matrices lead to a maximum pass length in four iteration but $\mathbf{L} = \mathbf{L_2} = \mathbf{P}^{-1}$ leads to slightly better precision than $\mathbf{L} = \mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$.

a constant disturbance in the form of an artificial offset, as depicted in Figure 2. Same experimental conditions are set: radius $r = 2\,\text{cm}$ for the output constraints, same VPL-ILC algorithm, same learning gain matrices[6]. Table II shows the evolution of the number of iteration (n_{MPL}) needed to achieve maximum pass length, for different values of the disturbance.

TABLE II

SENSITIVITY ANALYSIS OF THE EXPERIMENTAL RESULTS

D	$-2r$	$-r$	$-\frac{r}{2}$	0	$\frac{r}{2}$	r	$2r$
n_{MPL}	10	6	5	4	4	6	10

For a disturbance D as described in Figure 2, the number n_{MPL} of iterations needed to obtain maximum pass length ($n_j = \overline{n}$) is examined. Even for a disturbance twice as large as the radius r of the output constraints, the ILC algorithm with variable pass length is (monotonically) convergent. A small disturbance has very little impact on the convergence speed.

For small disturbances, the convergence speed of the VPL-ILC algorithm is almost unchanged (four iterations, as in Figure 7). For larger disturbances, the algorithm is still monotonically convergent but more iterations are needed to obtain maximum pass length. What appears to be a symmetry may originate from the almost symmetrical definition of the output constraints (constant radius r around the desired path $y_d(x)$). Our proposed algorithm apparently shows sufficiently low sensitivity with respect to disturbances.

A high definition video presentation of our experimental setup and control task is available at [13]. It shows how the rope is wound up as soon as the constraints are infringed. Therein, the maximum pass length $n_j = \overline{n}$ is achieved after eight iterations because the constraints are more restrictive ($r = 1.7\,\text{cm}$) than for the results shown in this paper.

[6]The results of the sensitivity analysis are the same for both learning gain matrices L_1 and L_2.

V. CONCLUSION

Two Iterative Learning Control (ILC) algorithms are considered for the case where output constraints are included. These constraints force the system to stop the trial when they are infringed, thus allowing the ILC controller to learn only from incomplete information. More work needs to be done on variable pass length (VPL) ILC. Few theoretical criteria have been found for monotonic convergence yet. However, a new result is presented on the evolution of the error on the first n_j samples, i.e. the error that is measured until the trial is disrupted. This VPL ILC is then applied to a laboratory-scale gantry crane and compared to the classical ILC approach, without constraints. The inversion based algorithm ($\mathbf{L_2} = \mathbf{P}^{-1}$) appears to be more precise than the algorithm based on a diagonal learning gain matrix ($\mathbf{L_1} = \lambda\mathbf{I}_{\overline{n}}$). In both cases the maximum pass length $n_j = \overline{n}$ is achieved after four iterations and monotonic convergence is observed even for very tight constraints. The proposed approach shows sufficiently low sensitivity with respect to constant and iteration invariant disturbances.

For VPL-ILC problems, new methods and criteria are to be found to guarantee monotonic convergence of the error.

REFERENCES

[1] B.D.O. Anderson and J.B. Moore, "Linear Optimal Control," *Prentice Hall* (Englewood Cliffs, NJ), 1971.

[2] D.A. Bristow, M. Tharayil and A.G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96-114, 2006.

[3] A. Duschau-Wicke, J. von Zitzewitz, R. Banz and R. Riener, "Iterative Learning Synchronization of Robotic Rehabilitation Tasks", *IEEE 10th Int. Conf. on Rehabilitation Robotics*, pp. 335-340, 2007

[4] H. Elci, R.W. Longman, M.Q Phan, J.N. Juang and R. Ugoletti, "Simple learning control made practical by zero-phase filtering: applications to robotics", *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 753-767, 2002.

[5] C. Freeman, P. Lewin, E. Rogers and J. Ratcliffe, "Iterative Learning Control Applied to a Gantry Robot and Conveyor System", *Transactions of the Institute of Measurement and Control*, vol. 32, no. 3, pp. 251-264, 2010.

[6] L. Hladowski, C. Zhonglun, K. Galkowski, E. Rogers, C.T. Freeman, P.L. Lewin and W. Paszke, "Repetitive process based iterative learning control designed by LMIs and experimentally verified on a gantry robot", *American Control Conference*, pp. 949-954, 2009.

[7] M. Hui-hai, "Design of the Low Density Mixture Liquid Control System Based on Iterative Learning Control Theory", *Second Int. Conf. on Intelligent Computation Technology and Automation*, vol. 2, pp. 849-851, 2009.

[8] M. Pandit and K. H. Buchheit, "Optimizing iterative learning control of cyclic production processes with application to extruders", *IEEE Tran. on Control Systems Technology*, vol. 7, no. 3, pp. 382-390, 1999.

[9] T. Seel, T. Schauer and J. Raisch, "Iterative learning control for variable pass length systems", *Preprints of the 18th IFAC World Congress (Milan, Italy)*, pp. 4880-4885, 2011.

[10] A. Simpkins , "System Identification: Theory for the User, 2nd Edition (Ljung, L.; 1999) [On the Shelf]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 95-96, 2012.

[11] J. Wang, X. Liu, and X. Xue, "Nonlinear Model Predictive Iterative Learning Control for Robotic System", *Int. Conf. on System Science and Engineering (ICSSE)*, pp. 258-263, 2012.

[12] Y. Wang, E. Dassau and F.J. Doyle, "Closed-Loop Control of Artificial Pancreatic β -Cell in Type 1 Diabetes Mellitus Using Model Predictive Iterative Learning Control", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 211-219, 2010.

[13] `http://tinyurl.com/vpl-ILC-crane`