Constrained Iterative Learning Control for Linear Time-Varying Systems With Experimental Validation on a High-Speed Rack Feeder

Article *in* Control Systems Technology, IEEE Transactions on · November 2021

DOI: 10.1109/TCST.2021.3123744

5 authors, including:

Bing Chu
University of Southampton
48 PUBLICATIONS **470** CITATIONS

Andreas Rauh
Carl von Ossietzky Universität Oldenburg
280 PUBLICATIONS **1,743** CITATIONS

David H Owens
Zhengzhou University
582 PUBLICATIONS **7,231** CITATIONS

Constrained Iterative Learning Control for Linear Time-Varying Systems with Experimental Validation on a High-Speed Rack Feeder

Bing Chu, Andreas Rauh, Harald Aschemann, Eric Rogers, and David H Owens

***Abstract*—Iterative learning control (ILC) applies to systems required to track the desired trajectory of finite duration repeatedly. This paper considers constrained ILC design for linear time-varying systems, a problem with limited, in relative terms, results in the literature but not uncommon in practical applications. Different design algorithms are developed and their convergence properties established. An extension of these designs to point-to-point tracking tasks is given. A high-speed rack feeder typically used in automated warehouses is considered to verify the designs. It represents a flexible beam structure subject to kinematic constraints such as a maximum velocity and a maximum acceleration with a vertically moving mass causing the time-varying characteristics. Experimental results demonstrate the effectiveness of the designs.**

***Index Terms*—iterative learning control, constrained control, experimental validation.**

I. Introduction

Iterative learning control (ILC) is a high-performance control design method capable of achieving accurate reference tracking for systems undertaking the same finite duration task repeatedly. Each execution is termed a trial (or iteration or pass) in the literature, and its finite duration is known as the trial (or iteration or pass) length. Once a trial has ended, information, e.g., state (if it can be measured or else estimated), input, and output, generated by the dynamics over the complete trial length, is available for use in constructing the control input for the next trial. The aim is to improve performance from trial-to-trial.

Originating in robotics [1], ILC has found successful applications in a wide range of areas. One possible starting point for the literature is the survey papers [2], [3] and the latest monograph [4]. Recent examples include various forms of additive manufacturing processes, e.g., [5], [6], nanopositioning, e.g., [7], path following for center-articulated industrial vehicles [8]. In the general area of healthcare, ILC has found applications in, e.g., robotic-assisted upper limb stroke rehabilitation with clinical trials, e.g. [9], and ventricular assist devices [10].

These and many other applications have, in the main, used, after modifications if necessary, model-based linear time-invariant ILC designs. Examples include inverse-model based methods, see, e.g., [11], gradient-type ILC, see, e.g., [12], optimisation based designs, see, e.g., [13] which form the control input by optimising a performance index. A typical example in the latter case is norm-optimal ILC (NOILC) designs (for background see, e.g., [14]), which uses a cost function that is the sum of a quadratic term in the current trial error and one in the difference between the control inputs on two successive trials. There has also been productive research on robust ILC with supporting experimental validation, see, e.g., [15]–[17].

The application areas discussed above assume that, for discrete dynamics, the reference trajectory passes through all sample instants. In some applications, however, this may be restrictive or not required. Point-to-point ILC relaxes the requirement to one where the reference trajectory is specified only at specified sample instants. This form of ILC has been developed, based on linear time-invariant dynamics, to the stage of experimental validation, see, e.g., [18]–[20]. Terminal ILC is a particular case of point-to-point where only the starting and final values of the trajectory are specified, for a recent application with experimental support see [21].

Motivated by practical relevance, ILC design in the presence of system constraints, e.g., input amplitude limits, has also seen productive research. In [22] a novel nonlinear controller for process systems with input constraints was developed. In [23] an ILC problem with soft constraints was studied where Lagrange multiplier methods are used. The results in [24] are based on quadratic optimal design to formulate a constrained ILC problem and suggest that such a design has the capability to deal with constraints. Also [25] used a constrained convex optimization setting and an interior-point-type method to solve the constrained ILC problem for linear systems with saturation and rate constraints.

A large majority of the published ILC literature is based on linear time-invariant (LTI) dynamics. Design for linear time-varying (LTV) systems, which are not uncommon in practice, is relatively limited. In [26] a D-type ILC in combination with a feedback loop and saturation action was considered to deal with saturation constraints for both input and output for LTV systems. The results in [24] apply to time-varying systems though the computation load can be problematic. An NOILC design was developed in [27] for unconstrained LTV systems with time varying weighting matrices. In [28] an iterative learning identification and learning control design (using a NOILC approach) was developed for time-varying systems in the absence of system constraints.

This paper develops a general constrained ILC design

Bing Chu and Eric Rogers are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK {b.chu, etar}@ecs.soton.ac.uk

Andreas Rauh is with Carl von Ossietzky Universität Oldenburg, Department of Computing Science, Group: Distributed Control in Interconnected Systems, 26111 Oldenburg, Germany andreas.rauh@uni-oldenburg.de

Harald Aschemann is with the Chair of Mechatronics, University of Rostock, Germany Harald.Aschemann@uni-rostock.de

David H Owens is a Principal Scientist with the Department of Automation, Zhengzhou University, Zhengzhou, 450001, P. R. China and an Emeritus Professor at the Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD d.h.owens@shef.ac.uk

framework for LTV systems to address the limitations discussed above. The design problem is formulated in abstract operator form in Hilbert space, and new constrained ILC algorithms are then developed using a successive projection framework, previously applied to LTI systems, see, e.g., [20], [29], [30]. The algorithms' convergence properties are established rigorously and their implementation procedures developed. Also the effectiveness of the new designs are validated experimentally on a high-speed rack-feeder system.

The analysis and results in this paper begin in the next section where the system dynamics are introduced, the constrained ILC design problem is formulated, and the necessary background on the successive projection framework is given. In the following sections III and IV, respectively, different constrained ILC design algorithms for LTV systems are derived, their convergence properties established together with procedures for their implementation. These algorithms are further extended to point-to-point ILC design in Section V. Section VI gives the experimental validation results from the rack feeder rig. Finally, Section VII provides the conclusions of this research and discusses possible future research.

II. Constrained ILC for LTV Systems

A. Representation of the System Dynamics

This paper considers single-input single-output (SISO) systems with an immediate extension to multiple-input multiple-output cases. Consider a discrete-time LTV system described in the ILC setting by the following state-space model

$$\begin{aligned} x_k(t+1) &= A_t x_k(t) + B_t u_k(t) \\ y_k(t) &= C_t x_k(t) \end{aligned} \tag{1}$$

where t is the time index (i.e., sample number), $k \geq 0$ is the trial number, $u_k(t), x_k(t), y_k(t)$ are the input, state vector and output, respectively, of the system on trial k, A_t, B_t, C_t are (time varying) system matrices with compatible dimensions. The initial state vector $x_k(0) = x_0, k = 1, 2, \cdots$ is the same for each trial, and no loss of generality arises from setting $x_0 = 0$. The control task is to track a given reference signal $r(t)$ defined on a finite interval $t \in [0, N]$ where $N > 0$ is the trial length. In operation, a trial is completed, the system is reset and a new trial begins. The ILC design uses information from previous trial(s) to compute the control input for the next trial in a manner that improves the tracking performance from trial-to-trial.

Defining the supervectors for the input and output on each trial as in the time-invariant case, i.e.,

$$\begin{aligned} u_k &= \left[\begin{array}{cccc} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{array}\right]^T \\ y_k &= \left[\begin{array}{cccc} y_k(1) & y_k(2) & \cdots & y_k(N) \end{array}\right]^T, \end{aligned} \tag{2}$$

leads to the lifted representation for the dynamics, i.e.,

$$y_k = G u_k \tag{3}$$

where the system operator matrix G is given by

$$G = \left[\begin{array}{ccccc} C_1 B_0 & 0 & \cdots & 0 & 0 \\ C_2 A_1 B_0 & C_2 B_1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & 0 \\ C_N A_{N-1} \cdots A_1 B_0 & \cdots & \cdots & \cdots & C_N B_{N-1} \end{array}\right].$$

Also introduce the lifted representation of $r(t)$ as

$$r = \left[\begin{array}{cccc} r(1) & r(2) & \cdots & r(N) \end{array}\right]^T \tag{4}$$

Then the lifted model representation of the error on trial k is

$$e_k = r - G u_k \tag{5}$$

where entry i, $1 \leq i \leq N$, in e_k is $r(i) - y_k(i)$.

In the LTI case, G is non-singular (when the system relative degree is one, i.e., $CB \neq 0$) and hence there is a unique solution to the tracking problem. However, for LTV systems, G can be singular due to the time-varying nature of the matrices A_t, B_t, and C_t. Consequently, there could be no solution to the reference tracking problem. To ensure that the control design problem is feasible, it is assumed in this paper that the reference signal r lies in the range of the matrix G, $r \in \mathrm{range}(G)$, i.e., there exists an input u^* such that $r = Gu^*$. When this condition holds but G is not invertible, there will be an infinite number of solutions to the reference tracking problem. Note that the assumption $r \in \mathrm{range}(G)$ is more general and less restrictive than assuming G is invertible as considered in existing literature (e.g., [24]).

B. Constrained ILC Design for LTV Systems

Constraints are often encountered in application areas and therefore design in the presence of such constraints will be needed as otherwise there is no a priori guarantee that the control signal will be applicable to the system dynamics. Different forms of constraints arise in applications, e.g., on the input itself, or rate constraints and state or output constraints. Also constraints can be divided into two classes termed, respectively, hard and soft. Hard constraints are due to physical limitations of the system, e.g., those on input magnitude(s) of actuators at each point in time. Soft constraints are those related to performance requirement, e.g., constraints on the total productivity. This paper considers hard input constraints, but the results extend directly to state and output constraints.

Suppose the input is constrained to be in a set Ω, taken as a closed convex set in a Hilbert space H. In practice, the set Ω is often of simple structure. For example, the following scenarios are often encountered, again stated for the SISO case:

- input saturation constraint:
$$\Omega = \{u \in H : |u(t)| \leq M(t),\ 0 \leq t \leq N-1\}$$
- input rate constraint:
$$\Omega = \{u \in H : |u(t) - u(t-1)| \leq \lambda(t),\ 1 \leq t \leq N-1\}$$
- input sign constraint:
$$\Omega = \{u \in H : 0 \leq u(t),\ 0 \leq t \leq N-1\}$$
- input energy constraint:
$$\Omega = \{u \in H : \sum_{t=0}^{N-1} u^2(t) \leq M\}$$

where $M(t), \lambda(t)$ and M are some positive scalars defining the input constraint bounds.

The constrained ILC design problem can now be formulated as finding a control law of the following general form

$$u_{k+1} = f\left(e_{k+1}, \ldots, e_{k-s}, u_k, \cdots, u_{k-q}\right), \tag{6}$$

such that the output y_k tracks the desired reference r, and that the generated input sequence satisfies the constraint requirements, i.e.,

$$u_k \in \Omega, \quad \forall k \geq 0.$$

Note that when $s > 0$ or $q > 0$, (6) is termed a higher order updating law. This paper only considers algorithms of the form $u_{k+1} = f\left(e_k, u_k\right)$. For higher order algorithms, refer to, e.g., [31] and/or the relevant references in the survey papers [2], [3].

In what follows, a successive projection framework previously used to develop a constrained ILC algorithm for LTI systems is summarized and then will be used in the next two sections to derive constrained ILC algorithms for LTV systems.

C. Constrained ILC for LTV Systems using Projection

The successive projection approach was recently used to develop several constrained ILC algorithms for LTI systems [20], [29], [30]. The key idea is to consider ILC design problem as iteratively finding a point in the intersection of two closed convex sets in some Hilbert spaces, which can then be solved using the well-known projection method shown in the theorem below.

Theorem 1. *[32], [33] Let S_1 and S_2 be two closed convex sets in a Hilbert space X. Define projection operators $P_{S_1}(\cdot)$ and $P_{S_2}(\cdot)$ as*

$$P_{S_1}(x) = \arg\min_{\hat{x}\in S_1} \|\hat{x} - x\|_X^2, \tag{7}$$

$$P_{S_2}(x) = \arg\min_{\hat{x}\in S_2} \|\hat{x} - x\|_X^2, \tag{8}$$

where $\|\cdot\|$ is the induced norm in X. Then given the initial estimate $x_0 \in X$, the sequences $\{\tilde{x}_k\}_{k\geq 0}$ and $\{x_k\}_{k\geq 0}$ generated by

$$\tilde{x}_{k+1} = P_{S_1}(x_k),\ x_{k+1} = P_{S_2}(\tilde{x}_{k+1}),\ k \geq 0 \tag{9}$$

are uniquely defined for each $x_0 \in X$ and satisfy the following monotonic convergence condition

$$\|\tilde{x}_{k+2} - x_{k+1}\|_X^2 \leq \|\tilde{x}_{k+1} - x_k\|_X^2. \tag{10}$$

If either set is compact or finite dimensional, the sequences $\{\tilde{x}_k\}_{k\geq 0}$ and $\{x_k\}_{k\geq 0}$ converge to fixed points $\tilde{x}^* \in S_1$ and $x^* \in S_2$, i.e.,

$$\lim_{k\to\infty} \tilde{x}_k = \tilde{x}^*, \ \lim_{k\to\infty} x_k = x^* \tag{11}$$

defining the minimum distance between two sets, i.e.,

$$\|\tilde{x}^* - x^*\|_X^2 = \min_{\tilde{x}\in S_1, x\in S_2} \|\tilde{x} - x\|_X^2. \tag{12}$$

Furthermore, if $S_1 \cap S_2 \neq \emptyset$, the following convergence condition is satisfied

$$\|x_{k+1} - x\|_X^2 \leq \|x_k - x\|_X^2, \ \forall x \in S_1 \cap S_2, k \geq 0. \tag{13}$$

In a similar way to ILC design for LTI systems, constrained ILC design for LTV systems can be formulated as iteratively finding a point in the intersection of S_1 and S_2 in the Hilbert space $H = \mathcal{Y} \times \mathcal{U}$, where

$$\begin{aligned} S_1 &:= \{(e,u) \in H : e = r - Gu, u \in \Omega\} \\ S_2 &:= \{(e,u) \in H : e = 0\} \end{aligned} \tag{14}$$

or the following two sets

$$\begin{aligned} S_1 &:= \{(e,u) \in H : e = r - Gu\} \\ S_2 &:= \{(e,u) \in H : e = 0, u \in \Omega\}, \end{aligned} \tag{15}$$

where the Hilbert spaces $\mathcal{Y} = \mathbb{R}^N$ and $\mathcal{U} = \mathbb{R}^N$ are equipped with the following inner products and associated induced norms

$$\begin{aligned} \langle y, x\rangle_Q &= y^T Q x, \quad \forall y, x \in \mathcal{Y} \\ \langle u, v\rangle_R &= u^T R v, \quad \forall u, v \in \mathcal{U} \end{aligned}$$

where Q and R are positive definite matrices of compatible dimensions. Using the above method to solve the constrained ILC design problem will lead to different algorithms that are described in the following two sections.

Before developing the constrained ILC algorithms, it is worth pointing out that when there are no constraints in the design problem, the succussive projection framework can also be applied, resulting in the following algorithm (which recovers the well known NOILC design in [14]):

Algorithm 0. *Given any initial input u_0 with corresponding error e_0, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by*

$$u_{k+1} = \arg\min_u \left\{\|r - Gu\|_Q^2 + \|u - u_k\|_R^2\right\} \tag{16}$$

solves the unconstrained ILC design problem for LTV systems.

This algorithm guarantees monotonic trial-to-trial error convergence and can be implemented using offline computations or online using a feedforward and feedback structure based on solutions of Riccati equations (where the tracking error e_k from previous trial is embedded in the computation by replacing r by $e_k + Gu_k$ in (16)). It also has a distinct feature for LTV systems (compared to LTI systems), as shown in the following proposition.

Proposition 1. *Given any initial input u_0 and corresponding error e_0, Algorithm 0 achieves monotonic trial-to-trial convergence to zero, i.e.,*

$$\|e_{k+1}\|_Q \leq \|e_k\|_Q, \quad \forall k \geq 0$$

and

$$\lim_{k\to\infty} e_k = 0.$$

Moreover, the input converges to the solution of the following optimization problem

$$\lim_{k\to\infty} u_k = u^* = \arg\min_u \{\|u - u_0\|_R, s.t.\ r = Gu\}$$

Consequently, when $u_0 = 0$, the minimum energy solution is obtained.

Proof. Monotonic convergence follows directly from Theorem 1 and the minimum energy solution follows a similar proof as that in [19] and hence the details are omitted. □

As discussed previously, for LTI systems there is only one solution for the ILC design problem considered but for LTV systems there may be infinite number of solutions if G is not invertible. For such cases, the algorithm above only finds one solution, which is also the best possible (this is often appealing in practice).

III. Constrained Iterative Learning Control Algorithm

This section begins the development of the constrained ILC algorithms for LTV systems, starting from the constraint formulation (14).

A. Algorithm Description and Convergence Properties

Consider the constrainted problem formulation of (14). Then application of Theorem 1 gives the following constrained ILC design algorithm for LTV systems.

Algorithm 1. *Given any initial input u_0 satisfying the constraint with corresponding tracking error e_0, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by*

$$u_{k+1} = \arg\min_{u\in\Omega} \left\{ \|r - Gu\|_Q^2 + \|u - u_k\|_R^2 \right\} \tag{17}$$

satisfies the constraint and iteratively solves the constrained ILC design problem for LTV systems.

In the case when perfect tracking of the reference trajectory is possible, i.e., $S_1 \cap S_2 \neq \emptyset$, Algorithm 1 has the desirable property that the tracking error norm will decrease monotonically to zero, as shown in the following theorem.

Theorem 2. *When perfect tracking is possible, Algorithm 1 achieves monotonic trial-to-trial error convergence to zero, i.e.,*

$$\|e_{k+1}\|_Q \leq \|e_k\|_Q, k = 0, 1, \cdots . \tag{18}$$

and

$$\lim_{k\to\infty} e_k = 0, \lim_{k\to\infty} u_k = u^*. \tag{19}$$

Moreover,

$$\|u_{k+1} - u^*\|_R \leq \|u_k - u^*\|_R, \ \forall k \geq 0, \tag{20}$$

i.e., the input iterates approach the converged solution u^* monotonically in norm.

Proof. Equations (18) and (20) follow from Theorem 1 (and Theorem 1 in [32]). Strong convergence follows as the Hilbert space considered is finite dimensional and the set S_1 is closed, bounded and therefore compact. □

Trial-to-trial error convergence to zero may not be possible due to the input constraints Ω, i.e., $S_1 \cap S_2 = \emptyset$. In such a case, the algorithm computes a 'best' approximation as shown next.

Theorem 3. *In the case when trial-to-trial error convergence to zero is not possible, Algorithm 1 converges to a solution u_s^* of the following optimisation problem*

$$u_s^* = \arg\min_{u\in\Omega} \|r - Gu\|_Q, \tag{21}$$

i.e., the algorithm converges to the minimum tracking error norm that can be achieved. Moreover, this convergence is monotonic in the error norm, i.e.,

$$\|e_{k+1}\|_Q \leq \|e_k\|_Q, k = 0, 1, \cdots . \tag{22}$$

Proof. By Theorem 1, the algorithm converges strongly to a fixed point of $P_{S_2}P_{S_1}$, i.e., a point $(0, u_s^*) \in S_2$ satisfying

$$\begin{aligned} &(0, u_s^*) = P_{S_2}P_{S_1}(0, u^*) \\ &= P_{S_2}\{(e, u) = \arg\min_{u\in\Omega} \|r - Gu\|_Q^2 + \|u - u^*\|_R^2\} = (0, u). \end{aligned}$$

Hence from this last equation

$$u_s^* = \arg\min_{u\in\Omega} \|r - Gu\|_Q.$$

Monotonic convergence of the trial-to-trial error norm follows in a similar manner to the case of Theorem 2 and hence the details are omitted. □

This algorithm has the monotonic trial-to-trial error convergence property and hence improvement of the error from trial-to-trial occurs. To realize this property requires an efficient procedure for solving the constrained optimization problem. Moreover, the choice of the weighting matrices Q and R affects the convergence properties of the algorithm. Consider, for ease of presentation, the SISO case. Then a smaller weighting R will allow more aggressive input changes and hence will normally result in a faster error convergence rate. Conversely, a smaller weighting Q on the error norm will put more emphasis on the input change during the optimization. This, in turn, will result in more 'caution' in updating the input signal and hence to slower trial-to-trial error convergence.

B. Implementation of Constrained Algorithm 1 using Receding Horizon Approach

Implementation of this algorithm requires the solution of the constrained quadratic programming (QP) problem (17) after each trial. This will not be a problem for many applications as this computation is conducted off line during two consecutive trials and there are many efficient QP solvers available, see, e.g., [34], [35]. However, there will also be applications where the computational load incurred could be problematic, e.g., when the dimension of the time series u_k is very large due to a high sampling rate and/or long trial length. In such cases solving the QP problem will be very computationally demanding or even unmanageable.

Next, an alternative algorithm is developed using a receding horizon approach to reduce the computational complexity and increase robustness against model uncertainty, which also applies to LTI systems.

Algorithm 2. *Given any initial input u_0 satisfying the constraint with associated error e_0, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by the following procedure also satisfies*

the constraints and iteratively solves the constrained ILC design problem.

- *At time t and for the current system state $x_{k+1}(t)$, use the LTV system model to predict the system output $y_{k+1}(t)$ over a fixed future interval $[t+1, t+N_y]$*
- *Solve the following optimal control problem over a fixed future interval, say $[t, t+N_u-1]$, taking into account the constraints*

$$u_{k+1,t}^{opt} = \arg\min_{u_{k+1,t}\in\Omega}\left\{\sum_{i=t+1}^{t+N_y}\|e_{k+1}(i)\|_Q^2 + \sum_{i=t}^{t+N_u-1}\|u_{k+1,t}(i)-u_k(i)\|_R^2\right\}, \quad (23)$$

where

$$u_{k+1,t} = \begin{bmatrix} u_{k+1}(t) & \cdots & u_{k+1}(t+N_u-1)\end{bmatrix}^T.$$

- *Apply only the first step input in the resulting optimal control sequence; measure or estimate the system state reached at sample instant $t+1$.*
- *Repeat the above steps until $t = N-1$, starting from the (new) current state.*

In common with receding horizon control, the selection of the control and prediction horizons N_u and N_y, respectively, can significantly affect the algorithm's trial-to-trial error convergence properties. Sufficiently large horizons should be used to give more accurate solutions and achieve as good performance as Algorithm 1. In the extreme case when $N_u = N_y = N$, Algorithm 2 recovers the Algorithm 1.

Conversely, selecting values for these parameters that are too small could lead to relatively poorer error convergence or even non-monotonic error norm convergence. Larger horizons will require more computational power at each time step. The balance between the computational power available and the obtained performance is application/user-specific. A rigorous analysis of control and prediction horizons' effect on the algorithm's performance (e.g., monotonic/asymptotic convergence) is nontrivial and left as a topic for further research.

IV. An Alternative Constrained Iterative Learning Control Algorithm

Use of the second abstract setting (15), i.e.,

- $S_1 = \{(e,u)\in H : e = r - Gu\}$
- $S_2 = \{(e,u)\in H : e = 0, u\in\Omega\}$

is the starting point for the development of the following alternative constrained ILC design algorithm for LTV system.

A. Algorithm Description and Convergence Properties

Algorithm 3. *Given any initial input u_0 satisfying the constraint with associated error e_0, the input sequence $u_{k+1}, k = 0,1,2,\cdots$, defined by the solution of the input unconstrained NOILC optimisation problem*

$$\tilde{u}_k = \arg\min_u\left\{\|r-Gu\|_Q^2 + \|u-u_k\|_R^2\right\} \quad (24)$$

followed by the simple input projection

$$u_{k+1} = \arg\min_{u\in\Omega}\|u-\tilde{u}_k\|_R \in \Omega \quad (25)$$

also satisfies the constraint and iteratively solves the constrained ILC problem.

The next two results give the convergence properties of this last algorithm for the cases when, respectively, $S_1\cap S_2 \neq \emptyset$ and $S_1\cap S_2 = \emptyset$.

Theorem 4. *When perfect tracking is possible, i.e., $S_1\cap S_2\neq\emptyset$, Algorithm 3 solves the constrained ILC problem in the sense that*

$$\lim_{k\to\infty} e_k = 0, \lim_{k\to\infty} u_k = u^*. \quad (26)$$

Moreover, this convergence is monotonic with respect to the performance index

$$J_k = \|Ee_k\|_Q^2 + \|Fe_k\|_R^2 \quad (27)$$

where

$$\begin{aligned} E &= I - G\left(G^TQG+R\right)^{-1}G^TQ \\ F &= \left(G^TQG+R\right)^{-1}G^TQ. \end{aligned} \quad (28)$$

Additionally,

$$\|u_{k+1}-u^*\|_R \leq \|u_k-u^*\|_R, \forall k\geq 0, \quad (29)$$

i.e., the input iterates approach the converged solution u^* monotonically in norm.

Proof. The proof is similar to that for Algorithm 1 and hence the details are omitted. □

This algorithm first computes the NOILC solution and then projects it onto the constraint set, which is computationally much simpler than Algorithm 1. However, this does not come for free since by Theorem 2, Algorithm 1 achieves monotonic trial-to-trial error convergence in the tracking error norm but this latest algorithm can only achieve this convergence property in the 'weighted' tracking error norm as shown in previous theorem.

When perfect tracking is not possible, i.e., $S_1\cap S_2=\emptyset$, an approximation of the 'best' solution (21) can be achieved, as shown next.

Theorem 5. *When perfect tracking is not possible, i.e., $S_1\cap S_2=\emptyset$, Algorithm 3 converges to a solution u_s^* of the following optimisation problem,*

$$u_s^* = \arg\min_{u\in\Omega}\left\{\|Ee\|_Q^2 + \|Fe\|_R^2\right\}. \quad (30)$$

Moreover, this convergence is monotonic with respect to the following performance index,

$$J_k = \|Ee_k\|_Q^2 + \|Fe_k\|_R^2 \quad (31)$$

where

$$\begin{aligned} E &= I - G\left(G^TQG+R\right)^{-1}G^TQ \\ F &= \left(G^TQG+R\right)^{-1}G^TQ. \end{aligned} \quad (32)$$

Proof. The proof is similar to that of Theorem 3 and is hence omitted here. □

In common with the first constrained ILC algorithm in this paper, the choice of weighting parameters in this last algorithm affects its convergence properties. In particular, the choices made will affect the converged solution when perfect tracking is not possible. To illustrate this, consider the simplest case of scalar weights, i.e, $Q = q \times I$ and $R = r \times I$, where $q, r > 0$. A small weighting, i.e., $r \to 0$ in (30) (in Algorithm 3) results in convergence to

$$u_s^* = \arg\min_{u\in\Omega} \|H_d V(u - u_{uc}^*)\|_R^2. \tag{33}$$

where H_d is a diagonal matrix with the first m diagonal elements all equal to one and the rest are zero, where m denotes the rank of G, V is a unitary matrix in the singular values decomposition of G (i.e., $G = U\Sigma V$) and $u_{uc}^* = G^\dagger r$ is the unconstrained ILC minimum energy solution in which $G^\dagger$ denotes the pseudoinverse of G.

If $m = N$, i.e., when G is full rank, the above is just simple a 'clipping' of the unconstrained solution u^*, with corresponding (possibly) large error. Conversely, if r is large, the convergence may be slow for both algorithms, but Algorithm 3 for $r \to \infty$ will converge to

$$u_s^* = \arg\min_{u\in\Omega} \|e\|_Q \tag{34}$$

which is the 'best' solution that can be obtained. Thus in Algorithm 3, there is a compromise between the convergence rate and the final converged tracking error norm. This compromise, however, does not exist for Algorithm 1 which will always converge to the best solution (34).

B. Implementation of Constrained Algorithm 3

To implement this last algorithm, the first step requires the solution of the unconstrained NOILC problem discussed previously in this paper. The second step of Algorithm 3 requires the solution of (25). In practice the input constraint Ω is often point-wise and the solution of (25) is easily computed. For example, when $\Omega = \{u \in \mathbb{R}^N : |u(t)| \le M(t)\}$, the solution is

$$u_{k+1}(t) = \begin{cases} M(t) & : \tilde{u}_k(t) > M(t) \\ \tilde{u}_k(t) & : |\tilde{u}_k(t)| \le M(t) \\ -M(t) & : \tilde{u}_k(t) < -M(t) \end{cases}, \tag{35}$$

for $t = 0, \cdots, N-1$; when $\Omega = \{u \in \mathbb{R}^N : \sum_{t=0}^{N-1} u^2(t) \le M\}$, the solution is

$$u_{k+1}(t) = \frac{\tilde{u}_k(t)}{\sum_{t=0}^{N-1} \tilde{u}_k^2(t)} \cdot M, \quad t = 0, \cdots, N-1.$$

Note that Algorithm 3 only requires two simple steps, both of which are easy to complete even for large scale problems.

V. Extension to Point-to-Point Tracking of Constrained LTV systems

In the previous sections, constrained ILC algorithms were developed for LTV systems with a reference defined over the whole trial interval. In this section the extension to point to point tracking problems is developed.

A. Point-to-point tracking

The tracking task considered so far is for the output to track a reference defined over the whole horizon $[0, N]$. However, there are applications where only the reference tracking at a subset of the trial interval is of interest. For example, in robotic pick and place tasks, only tracking at the pick and place instants is required. This type of tracking is termed point-to-point. Point-to-point ILC design for LTI systems has been well-studied, e.g., [19], [20], [30]. Constrained point-to-point ILC design for LTV systems has not been explored.

To formulate the problem, suppose the system is now required to track P intermediate point at $t = t_1, t_2, \ldots, t_P$ with reference $r_{\mathrm{ref}}(t_i)$ respectively. Using the lifted form representation of discrete ILC dynamics, the system output at the above time instants can be written as

$$y_k^e = G_e u_k$$

where

$$\begin{array}{c} y_k^e = \left[\begin{array}{cccc} y_k^e(t_1) & y_k^e(t_2) & \cdots & y_k^e(t_P) \end{array}\right]^T, \\ r^e = \left[\begin{array}{cccc} r_{\mathrm{ref}}(t_1) & r_{\mathrm{ref}}(t_2) & \cdots & r_{\mathrm{ref}}(t_P) \end{array}\right]^T \end{array} \tag{36}$$

and G_e is constructed by extracting the t_ith rows from the system matrix G. The point-to-point tracking error e^e is defined as

$$e^e = r^e - y^e.$$

The constrained point to point ILC design problem can now be formulated as finding a control law of the following general form

$$u_{k+1} = f\left(e_k^e, u_k\right), \tag{37}$$

such that the output y_k^e tracks desired reference r^e, and that the generated input sequence satisfies the constraint, i.e.,

$$u_k \in \Omega, \quad \forall k \ge 0.$$

B. Constrained point-to-point ILC for LTV systems

It turns out that the constrained point to point to ILC design for LTV systems can also be solved using the projection method. Following similar procedures as in the previous two sections gives the following two algorithms and associated convergence properties. The proofs are omitted here for ease of presentation.

Algorithm 4. *Given any initial input u_0 satisfying the constraint with associated point-to-point tracking error e_0^e, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by*

$$u_{k+1} = \arg\min_{u\in\Omega} \left\{ \|r^e - G_e u\|_Q^2 + \|u - u_k\|_R^2 \right\} \tag{38}$$

also satisfies the constraint and iteratively solves the constrained point-to-point ILC design problem.

Algorithm 4 is an extension of Algorithm 1 to the point-to-point ILC design and has the following properties.

Theorem 6. *When perfect tracking is possible, Algorithm 4 achieves monotonic convergence to zero tracking error, i.e.,*

$$\|e_{k+1}^e\|_Q \le \|e_k^e\|_Q, \; k = 0, 1, \cdots \tag{39}$$

and

$$\lim_{k\to\infty} e_k^e = 0, \lim_{k\to\infty} u_k = u^*. \tag{40}$$

Moreover,

$$\|u_{k+1} - u^*\|_R \leq \|u_k - u^*\|_R, \, \forall k \geq 0, \tag{41}$$

i.e., the input iterates approach the converged solution u^* monotonically in norm.

When perfect tracking is not possible, Algorithm 4 converges to a solution u_s^* of the following optimisation problem

$$u_s^* = \arg\min_{u\in\Omega} \|r^e - G_e u\|_Q, \tag{42}$$

i.e., the algorithm converges to the minimum tracking error norm that can be achieved. Moreover, this convergence is monotonic in the tracking error norm, i.e.,

$$\|e_{k+1}^e\|_Q \leq \|e_k^e\|_Q, k = 0, 1, \cdots. \tag{43}$$

An alternative design algorithm follows as an extension of Algorithm 3 and is given next.

Algorithm 5. *Given any initial input u_0 satisfying the constraint with associated error e_0^e, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by the solution of the input unconstrained point-to-point NOILC optimisation problem*

$$\tilde{u}_k = \arg\min_u \left\{ \|r^e - G_e u\|_Q^2 + \|u - u_k\|_R^2 \right\} \tag{44}$$

followed by the simple input projection

$$u_{k+1} = \arg\min_{u\in\Omega} \|u - \tilde{u}_k\| \in \Omega \tag{45}$$

also satisfies the constraint and iteratively solves the constrained point-to-point ILC design problem.

The following result gives the convergence properties of this last algorithm.

Theorem 7. *When perfect tracking is possible, i.e., $S_1 \cap S_2 \neq \emptyset$, Algorithm 5 solves the point-to-point ILC problem in the sense that*

$$\lim_{k\to\infty} e_k^e = 0, \lim_{k\to\infty} u_k = u^*. \tag{46}$$

Moreover, this convergence is monotonic with respect to the following performance index,

$$J_k = \|Ee_k^e\|_Q^2 + \|Fe_k^e\|_R^2 \tag{47}$$

where

$$\begin{aligned} E &= I - G_e \left(G_e^T Q G_e + R\right)^{-1} G_e^T Q \\ F &= \left(G_e^T Q G_e + R\right)^{-1} G_e^T Q. \end{aligned} \tag{48}$$

Additionally,

$$\|u_{k+1} - u^*\|_R \leq \|u_k - u^*\|_R, \, \forall k \geq 0, \tag{49}$$

i.e., the input iterates approach the converged solution u^* monotonically in norm.

When perfect tracking is not possible, i.e., $S_1 \cap S_2 = \emptyset$, Algorithm 5 converges to a solution u_s^* of the following optimisation problem,

$$u_s^* = \arg\min_{u\in\Omega} \left\{ \|Ee^e\|_Q^2 + \|Fe^e\|_R^2 \right\}. \tag{50}$$

Moreover, this convergence is monotonic with respect to the following performance index,

$$J_k = \|Ee_k^e\|_Q^2 + \|Fe_k^e\|_R^2 \tag{51}$$

where

$$\begin{aligned} E &= I - G_e \left(G_e^T Q G_e + R\right)^{-1} G_e^T Q \\ F &= \left(G_e^T Q G_e + R\right)^{-1} G_e^T Q. \end{aligned} \tag{52}$$

The implementation details of the above two algorithms are similar to those explained in the previous two sections and are omitted here for brevity.

VI. Experimental Validation on a High-Speed Rack Feeder System

In this section, the developed algorithms will be experimentally verified on a high-speed rack feeder system.

A. Control-Oriented Modeling

This subsection describes the control-oriented modeling of the prototypic high-speed rack feeder system shown in Figure 1.

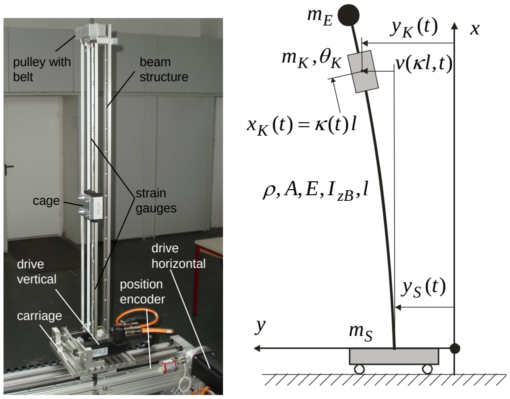


Fig. 1. The high-speed rack feeder system (left) and its corresponding elastic multibody model (right).

The setup consists of a carriage that is driven by an electric DC servo motor via a toothed belt, implementing the basic control input for the motion in horizontal direction. An elastic double-beam structure is vertically mounted on this carriage. On the beam structure, a cage (with variable load mass) can be moved in the vertical direction. The position of this cage, described by the coordinates $y_K(t)$ in horizontal and $x_K(t)$ in vertical direction, represents the tool center point of the rack feeder that should track desired trajectories as accurately as possible with small tracking errors in transient phases and without remaining oscillations at standstill.

The movable cage is driven by a toothed belt with an electric DC servo motor in common with the carriage. Measured data are available from encoders for the actuator angles. Also, the horizontal carriage position is determined by a magnetostrictive transducer. Strain gauges are used to determine the beam deflection during system operation. The horizontal and vertical axes can be operated by fast underlying velocity controllers, running directly on the current converters. Consequently, the corresponding velocities are the control inputs.

The ILC algorithms developed in this paper will be employed to guarantee that predefined limitations on the amplitude of the control signal (the desired carriage velocity in the subsequent model) and constraints on its first time derivative (constraints on the carriage acceleration) are not violated. For simplicity, it is assumed that the vertical cage position $x_K(t)$ is known (but time-varying). As described below, this term leads to a control design problem for time-varying systems, where the time-varying dynamics arise in the vertical motion. This latter feature was not considered in [36], where this preliminary investigation did not consider changes in the vertical position.

In previous work [37], [38], it is shown that a control-oriented elastic multibody model can be employed effectively for the control of the rack feeder system in Figure 1. These control laws used gain-scheduled feedback control design and model predictive trajectory control. Next, a brief summary of the system model is given, with a detailed treatment in [37], [38]. This previous work has established that this model structure has sufficient accuracy to achieve both tracking control of desired trajectories of the cage position in the horizontal direction and active oscillation damping through the system's main drive.

In the control-oriented modeling, the rack feeder is represented by a multibody model (Figure 1, right) with three rigid bodies — the carriage with mass m_S, the cage (mass m_K, mass moment of inertia θ_K, position $x_K(t)$) movable vertically on the beam structure, and a point mass m_E at the tip of the beam — and an elastic Bernoulli beam (density ρ, cross-section A, Young's modulus E, second moment of area I_{zB}, and length l). In the following, the vertical cage position is denoted by the dimensionless parameter

$$\kappa\left(t\right) := \frac{x_K\left(t\right)}{l}.$$

The elastic degrees of freedom of the beam with respect to its bending deflection are represented by the Ritz ansatz

$$\begin{aligned} v\left(x_K(t), t\right) &= v\left(\kappa(t)l, t\right) = \bar{\bar{v}}_1\left(\kappa(t)l\right) \cdot v_1\left(t\right) \\ &= \left(\frac{3}{2}\kappa^2(t) - \frac{1}{2}\kappa^3(t)\right) v_1\left(t\right), \end{aligned} \tag{53}$$

in which only the first bending mode is considered. Here, the ansatz function for the bending deflection has been evaluated directly at the vertical cage position $x_K(t)$ to obtain a mathematical model of the desired system output.

The equations of motion for the rack feeder with the generalized coordinates $q := \left[\begin{array}{cc} y_S\left(t\right) & v_1\left(t\right) \end{array}\right]^T$ (assuming that the vertical cage position $x_K(t)$ is known) and the input vector $h = \left[\begin{array}{cc} 1 & 0 \end{array}\right]^T$ can then be stated as

$$M_R \ddot{q}\left(t\right) + D\dot{q}\left(t\right) + Kq\left(t\right) = h \cdot \left(F_M\left(t\right) - F_F\left(\dot{y}_S\left(t\right)\right)\right) \tag{54}$$

where F_M and F_F denote the motor and friction forces, respectively,

$$M_R = \left[\begin{array}{cc} m_S + \rho A l + m_K + m_E & m_{12} \\ m_{12} & m_{22} \end{array}\right], \tag{55}$$

is the mass matrix and

$$m_{12} = \frac{3}{8}\rho A l + \frac{m_K \kappa^2(t)}{2}\left(3 - \kappa(t)\right) + m_E$$

and

$$\begin{aligned} m_{22} = \frac{33}{140}\rho A l &+ \frac{6\rho I_{zB}}{5l} + \frac{m_K \kappa^4(t)}{4}\left(3 - \kappa(t)\right)^2 \\ &+ \frac{9\theta_K \kappa^2(t)}{l^2}\left(1 - \kappa(t) + \frac{\kappa^2(t)}{4}\right) + m_E \end{aligned}$$

hold. Also the stiffness and damping matrices K and D, respectively, are given by

$$K = \left[\begin{array}{cc} 0 & 0 \\ 0 & k_{22} \end{array}\right] \text{ and } D = \left[\begin{array}{cc} 0 & 0 \\ 0 & \frac{3k_d E I_{zB}}{l^3} \end{array}\right] \tag{56}$$

with

$$\begin{aligned} k_{22} = &\frac{3EI_{zB}}{l^3} - \frac{3}{8}\rho A g \\ &- \frac{3m_K g \kappa^3(t)}{l}\left(1 + \frac{3\kappa^2(t)}{20} - \frac{3\kappa(t)}{4}\right) - \frac{6m_E g}{5l}. \end{aligned}$$

For the experimental control implementation used in the next section to obtain experimental data, it is assumed that the electric drive for the horizontal motion is operated in an underlying velocity control regime, implemented on its current converter, with the resulting dynamics

$$T_{1y}\ddot{y}_S\left(t\right) + \dot{y}_S\left(t\right) = v_S\left(t\right). \tag{57}$$

Including the first-order lag dynamics in the overall system model gives the equations of motion

$$\begin{aligned} \dot{x}_y &= \left[\begin{array}{cc} 0 & I \\ -M_y^{-1}K_y & -M_y^{-1}D_y \end{array}\right] \cdot x_y + \left[\begin{array}{c} 0 \\ M_y^{-1}h_y \end{array}\right] v_S \\ &=: A_y x_y + b_y u_y, \end{aligned} \tag{58}$$

with the control input $u(t) := v_S$, the continuous-time state vector

$$x_y = \left[\begin{array}{c} q \\ \dot{q} \end{array}\right], \tag{59}$$

and modified mass and damping matrices

$$M_y = \left[\begin{array}{cc} T_{1y} & 0 \\ \frac{3}{8}\rho A l + \frac{m_K \kappa^2(t)}{2}\left(3 - \kappa(t)\right) + m_E & m_{22} \end{array}\right] \tag{60}$$

and

$$D_y = \left[\begin{array}{cc} 1 & 0 \\ 0 & \frac{3k_d E I_{zB}}{l^3} \end{array}\right]. \tag{61}$$

Note that the above is a linear time-varying model as $\kappa(t)$ is changing over time. The discrete-time linear time-varying system model (1) is then obtained numerically by computing the matrices of the corresponding state-space representation with the zero-order hold method provided by MATLAB with piecewise constant control inputs and a sampling time of 50 ms. The corresponding system output, namely, the horizontal cage position, corresponds to $y(t) = y_S(t) + v\left(x_K(t), t\right)$.

B. Simulation study

In this subsection, the performance of the constrained algorithms developed in sections III and IV (i.e., Algorithms 1,2, and 3) are compared in simulation using the rack feeder model. The discussion is divided into two parts: inactive constraints and active constraints. A slightly modified reference signal to that used in experiments detailed later in this paper is used to clearly illustrate their performance difference.

In the simulation, the trial duration is $12\,\mathrm{s}$ with a sampling time of $0.05\,\mathrm{s}$, i.e., trial length $N = 240$. The weighting matrices are chosen as $Q = I$ and $R = 0.1 \times I$ respectively. Zero initial input, i.e., $u_0 = 0$, is used for all the algorithms. For Algorithm 2, the control and prediction horizons are chosen as $N_y = N_u = 40$.

1) Inactive constraints: First, the case where the constraints on the inputs (i.e., input amplitude constraint of $7.5\,\mathrm{m/s}$ and input change rate constraint of $30\,\mathrm{m/s^2}$) are inactive for the desired reference signal is considered. In other words, perfect tracking of the reference is possible under constraints. The tracking error norm convergence over 30 trials and outputs on the last trial (together with the reference) for all three algorithms are given in Figs. 2 and 3 respectively. Note that the results in the figures are on top of each other and almost indistinguishable.

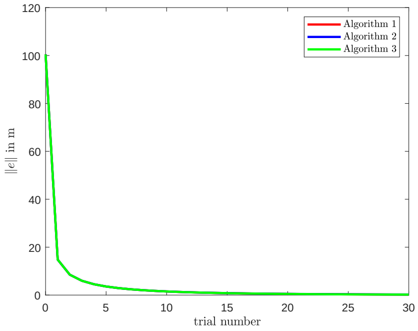


Fig. 2. Simulation results: Comparison of the error norm convergence over 30 trials for inactive constraints.

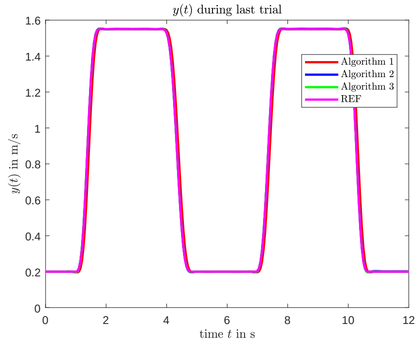


Fig. 3. Simulation results: Comparison of the output signals during the 30th trial for inactive constraints.

From the figures, it can be seen that all three algorithms achieve perfect tracking of the reference signal. Furthermore in this particular set of simulations, their convergence performance are almost identical to each other and not distinguishable in these figures. Finally, inspection on the obtained inputs shows that they all satisfy the constraint requirements, the results of which are omitted here for space reason.

2) Active constraints: We now consider the case where the constraints on the inputs (i.e., input amplitude constraint of $1.5\,\mathrm{m/s}$ and input change rate constraint of $2\,\mathrm{m/s^2}$) are active for the desired reference signal, i.e., perfect tracking of the reference is not possible. The tracking error norm convergence over 30 trials, outputs on the last trial (together with the reference), and inputs on the last trial for all three algorithms are given in Figs. 4, 5 and 6 respectively. Note that the results for Algorithm 1 and Algorithm 2 in the figures are on top of each other and almost indistinguishable.

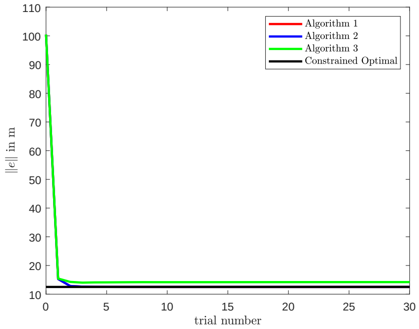


Fig. 4. Simulation results: Comparison of the error norm convergence over 30 trials for active constraints.

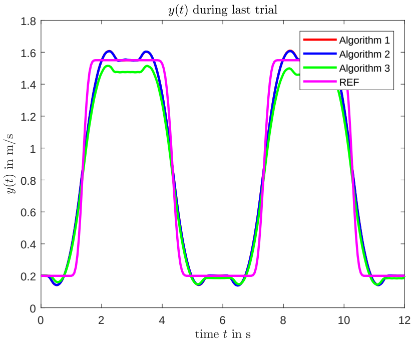


Fig. 5. Simulation results: Comparison of the output signals during the 30th trial for active constraints.

From the figures, it can be seen that as expected Algorithm 1 (red line) converges monotonically in the tracking error norm to the minimum possible value (black line in Fig. 4, determined by Theorem 3). Algorithm 2, though with significantly reduced computational load, achieves almost identical convergence performance as Algorithm 1, which is particularly appealing for applications with moderate computational resources. Algorithm 3 has the simplest computational structure. It converges but to a slightly bigger final tracking error norm (compared to the previous two algorithms). This difference can also been seen from the output and input signals on the last trial.

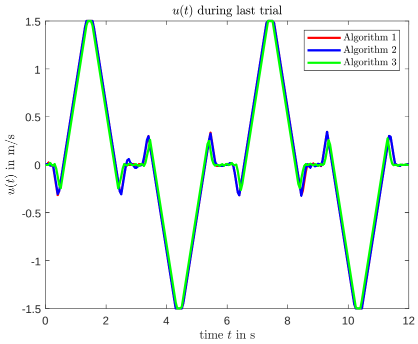


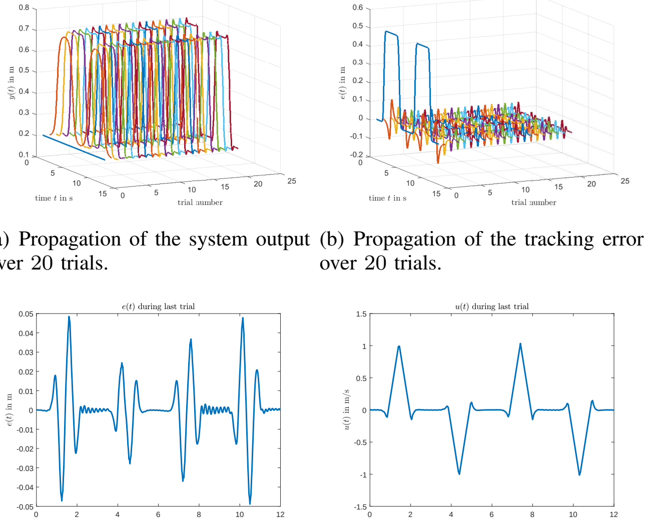
Fig. 6. Simulation results: Comparison of the input signals during the 30th trial for active constraints.

The above observations shed lights on how to choose a particular algorithm in practice. Depending on the computational resource available and the requirements of a particular practical application, the following guidelines for choosing the algorithms are given: i) Algorithm 1 should always be the first choice due to its appealing convergence properties, if the trial length is sufficiently short, or the resetting time between two consecutive trials is sufficiently long, or if the available computation hardware is powerful enough; ii) Algorithm 2 with sufficiently large horizons should be considered as the second preferable choice if the conditions above are not met, and finally iii) Algorithm 3 as the last choice for scenarios where the computation power is limited.

C. Experimental Results

In this subsection, the performance of the constrained ILC algorithms developed in Section III is verified experimentally on the high-speed rack feeder test rig. For application, the algorithms in Section III have better convergence properties (e.g., monotonic convergence in the tracking error norm), whereas the one of Section IV has better computational efficiency. The computations for the rack feeder are not intensive, and hence the focus of the experimental tests is on the convergence properties of the algorithms in Section III.

1) Algorithm 1: QP based Constrained ILC: This algorithm has been applied to two different scenarios, firstly, it has been used for the case where predefined constraints of the maximum carriage velocity and its maximum acceleration are activated during tracking of the desired reference signal. The experimental results obtained are in Figure 7, where the control signal has been initialized with the value zero at the beginning of the experiments. Hence, the tracking error signal coincides with the desired trajectory for $k = 0$, and this control algorithm leads to a rapid decrease of the tracking error after only a few trials. To prevent an increase in the error as the trial number increased, which can often be detected in experimentally implemented ILC algorithms, the error signal is filtered by applying a zero-phase filter before the computation of the next trial input. This is possible due to the finite trial length and is undertaken in the time between the end of a trial and the start of the next. In this application, a fourth-order Butterworth filter with a cutoff frequency of $5\,\mathrm{Hz}$ and no additional real-time constraints are imposed.



(a) Propagation of the system output over 20 trials.

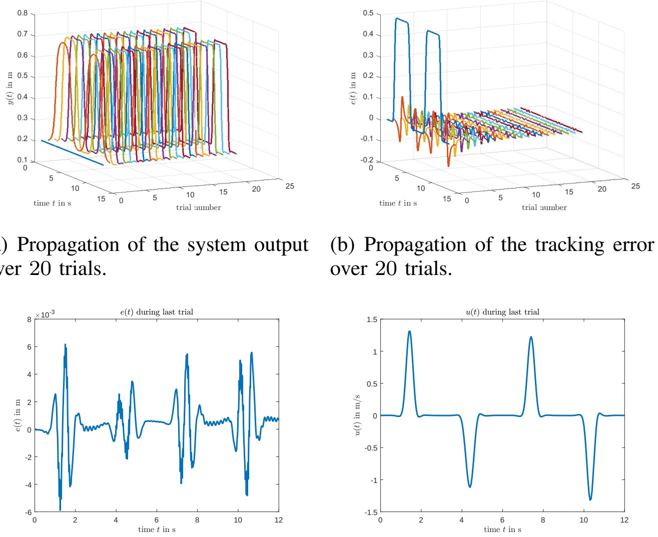
(b) Propagation of the tracking error over 20 trials.

(c) Tracking error (20th trial).

(d) Control signal (20th trial).

Fig. 7. Experimental results for Algorithm 1 with active constraints (max. carriage velocity $1.5\,\mathrm{m/s}$, max. acceleration $2.0\,\mathrm{m/s^2}$).

In the second scenario, the corresponding limits are set to values that are not reached during the complete sequence of experiments. As shown in Fig. 8, the controlled error is reduced significantly from trial-to-trial in cases when none of the input and input rate constraints are active. As in the first scenario, the error is again zero-phased filtered after each trial, leading to smooth control action with excellent damping of remaining oscillations at standstill positions of the rack feeder.



(a) Propagation of the system output over 20 trials.

(b) Propagation of the tracking error over 20 trials.

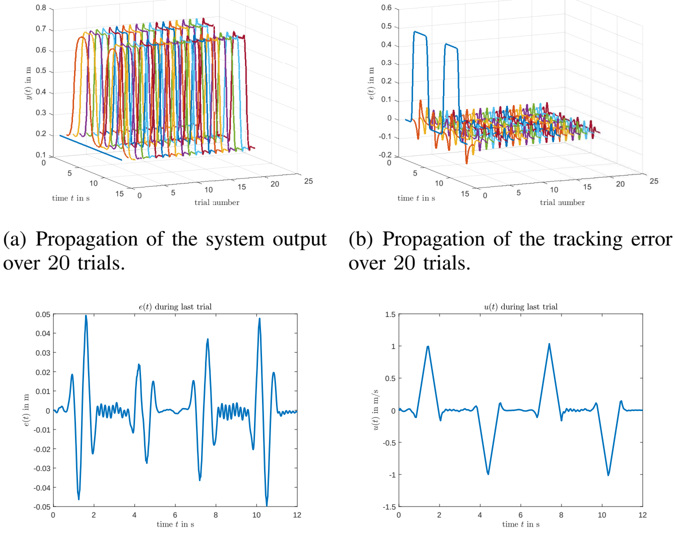
(c) Tracking error (20th trial).

(d) Control signal (20th trial).

Fig. 8. Experimental results for Algorithm 1 with inactive constraints.

2) Algorithm 2: Receding Horizon Approach based Constrained ILC: Figures 9 and 10 give the results of repeating the previously described experiments for Algorithm 2. Its computational requirement are less demanding ($N_u = N_y = 23$) –

it only requires the solution of a QP problem with a decision variable in $\mathbb{R}^{23}$ rather than $\mathbb{R}^{240}$ as for Algorithm 1 in the previous subsection. As can been seen in Figure 9 when the input and input rate constraints are active, both control algorithms show a very similar behavior in the experimental validation. Additionally, the choice of the parameters N_u and N_y again leads to suitable damping of the elastic vibrations at standstill.



(a) Propagation of the system output over 20 trials.

(b) Propagation of the tracking error over 20 trials.

(c) Tracking error during the 20th trial.

(d) Control signal during the 20th trial.

Fig. 9. Experimental results for Algorithm 2 with active constraints (max. carriage velocity $1.5\,\mathrm{m/s}$, max. acceleration $2.0\,\mathrm{m/s^2}$).

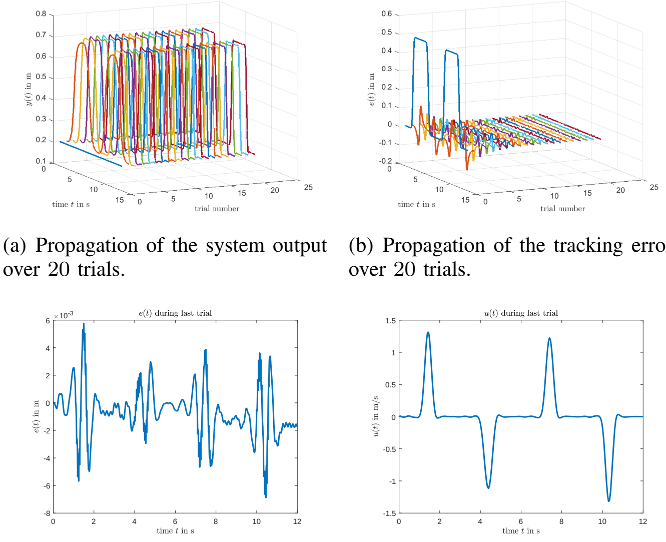
Repeating the experiment with inactive constraints results in similar performance as those observed for the first algorithm (as shown in Figure 10).

To compare the performance of the two algorithms, the error norm convergence of the four experiments and the converged inputs on the last trial are given in Figures 11 and 12, respectively, showing that: i) the tracking error norms converge monotonically for Algorithm 1 (as predicted by the theory), ii) Algorithm 2 produces similar performance as Algorithm 1 (results in the figures for both algorithms are in very close agreement and indistinguishable), and that iii) all kinematic constraints are satisfied and best exploited.

VII. Conclusions

In this paper, a constrained ILC design framework for linear time-varying systems has been developed. The analysis is based on an operator-based description in a Hilbert space setting. Using this setting, different constrained ILC algorithms are developed, and their convergence properties established. These algorithms have very different convergence properties and computational requirements, and implementation procedures have been developed for all of them. Also, an extension to the more challenging case of point-to-point tracking tasks for LTV systems has been developed. Supporting experimental results are given for the standard case with point-to-point experimental validation left as an area for future research, where these focus on the convergence properties since computational complexity is not an issue for the rack feeder.

The design approach developed in this paper is very general. While this paper has addressed discrete-time dynamics, the results can in theory be extended to continuous-time and time-delayed dynamics, as well as to directly include the filtering actions (e.g., zero phase filter used in the experiments) into the design. In addition, the experimental as well as additional simulation studies suggest that the new algorithms have a certain degree of robustness against the inevitable modeling uncertainties and noise/disturbances. A rigourous theoretical analysis on this, particularly the level of model uncertainties and non-repeatable disturbances that can be tolerated (using possibly the operator theory from [4]), is on the way. Future research will also consider the development of algorithms for robust ILC of LTV systems, and experimental validation of the new constrained point-to-point tracking algorithms for LTV systems.



(a) Propagation of the system output over 20 trials.

(b) Propagation of the tracking error over 20 trials.

(c) Tracking error during the 20th trial.

(d) Control signal during the 20th trial.

Fig. 10. Experimental results for Algorithm 2 with inactive constraints.

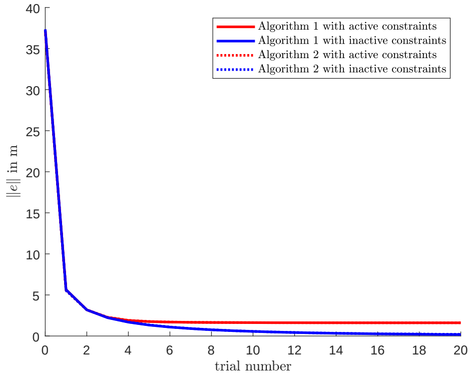


Fig. 11. Comparison of the error norm convergence over 20 trials.

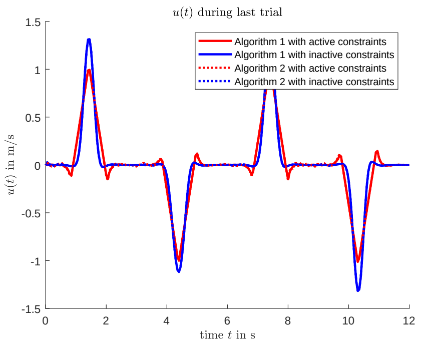


Fig. 12. Comparison of the control signals for *Algorithm 2* during the 10th trial.

References

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operations of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 1, pp. 123–140, 1984.

[2] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control: A learning-based method for high-performance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[3] H. S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 37, no. 6, pp. 1109–1121, 2007.

[4] D. H. Owens, *Iterative learning control: an optimization paradigm*. Springer, 2015.

[5] W. Rafajlowicz, P. Jurewicz, J. Reiner, and E. Rafajlowicz, "Iterative learning of optimal control for nonlinear processes with applications to laser additive manufacturing," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2647–2654, 2019.

[6] I. Lim, D. J. Hoelzle, and K. L. Barton, "A multi-objective iterative learning control approach for additive manufacturing applications," *Control Engineering Practice*, vol. 64, pp. 74–87, 2017.

[7] R. de Rozario, A. Fleming, and T. Oomen, "Finite-time learning control using frequency response data with application to a nanopositioning stage," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2085–2096, 2019.

[8] L. G. Dekker, J. A. Marshall, and J. Larsson, "Experiments in feedback linearized iterative learning-based path following for center-articulated industrial vehicles," *Journal of Field Robotics*, vol. 36, no. 5, pp. 955–972, 2019.

[9] C. T. Freeman, E. Rogers, A.-M. Hughes, J. H. Burridge, and K. L. Meadmore, "Iterative learning control in healthcare electrical stimulation and robotic-assisted upper limb stroke rehabilitation," *IEEE Control Systems Magazine*, vol. 32, no. 1, pp. 18–43, 2012.

[10] M. Ketelhut, S. Stemmler, J. Gesenhues, M. Hein, and D. Abel, "Iterative learning control of ventricular assist devices with variable cycle durations," *Control Engineering Practice*, vol. 83, pp. 33–44, 2019.

[11] R. W. Longman, Y. Peng, T. Kwon, H. Lus, R. Betti, and J. Juang, "Adaptive inverse iterative learning control," *Journal of the Chinese Society of Mechanical Engineers*, vol. 32, no. 6, pp. 493–506, 2011.

[12] D. H. Owens, J. Hatonen, and S. Daley, "Robust monotone gradient-based discrete-time iterative learning control," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 6, pp. 634–661, 2009.

[13] E. Rogers, D. H. Owens, H. Werner, C. T. Freeman, P. L. Lewin, S. Kichhoff, and G. Lichtenberg, "Norm optimal iterative learning control with application to problems in accelerator based free electron lasers and rehabilitation robotics," *European Journal of Control*, vol. 16, no. 5, pp. 497–524, 2010.

[14] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proceedings-Control Theory and Applications*, vol. 143, no. 2, pp. 217–224, 1996.

[15] J. van de Wijdeven, T. Donkers, and O. Bosgra, "Iterative learning control: Robust monotonic convergence analysis," *Automatica*, vol. 45, no. 10, pp. 2383–2391, 2009.

[16] ——, "Iterative learning control for uncertain systems: noncausal finite time interval robust control design," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 14, pp. 1645–1666, 2011.

[17] W. Paszke, E. Rogers, K. Galkowski, and Z. Cai, "Robust finite frequency range iterative learning control design and experimental verification," *Control Engineering Practice*, vol. 21, no. 10, pp. 1310–1320, 2013.

[18] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 500–510, 2011.

[19] D. H. Owens, C. T. Freeman, and T. V. Dinh, "Norm-optimal iterative learning control with intermediate point weighting: theory, algorithms, and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 999–1007, 2013.

[20] Y. Chen, B. Chu, and C. T. Freeman, "Generalized iterative learning control using successive projection: algorithm, convergence, and experimental verification," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2079–2091, 2020.

[21] S. V. Johansen, M. R. Jensen, B. Chu, J. D. Bendtsen, and E. Rogers, "Broiler FCR optimization using norm optimal terminal iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 580–592, 2021.

[22] C. T. Chen and S. T. Peng, "Learning control of process systems with hard input constraints," *Journal of Process Control*, vol. 9, no. 2, pp. 151–160, 1999.

[23] S. Gunnarsson and M. Norrlof, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011–2016, 2001.

[24] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," *Automatica*, vol. 36, no. 5, pp. 641–657, 2000.

[25] S. Mishra, U. Topcu, and M. Tomizuka, "Optimization-based constrained iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1613–1621, 2011.

[26] G. Sebastian, Y. Tan, and D. Oetomo, "Convergence analysis of feedback-based iterative learning control with input saturation," *Automatica*, vol. 101, pp. 44–52, 2019.

[27] K. L. Barton and A. G. Alleyne, "A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 166–180, 2011.

[28] N. Liu and A. Alleyne, "Iterative learning identification/iterative learning control for linear time-varying systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 10, 2016.

[29] B. Chu and D. H. Owens, "Iterative learning control for constrained linear systems," *International Journal of Control*, vol. 83, no. 7, pp. 1397–1413, 2010.

[30] B. Chu, C. T. Freeman, and D. H. Owens, "A novel design framework for point-to-point ILC using successive projection," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1156–1163, 2014.

[31] Z. Bien and K. M. Huh, "Higher-order iterative learning control algorithm," *IEE Proceedings, Part D: Control Theory and Applications*, vol. 136, no. 3, pp. 105–112, 1989.

[32] D. H. Owens and R. P. Jones, "Iterative solution of constrained differential/algebraic systems," *International Journal of Control*, vol. 27, no. 6, pp. 957–974, 1978.

[33] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point of convex sets," *U.S.S.R. Computational Mathematics and Mathematical Physics*, vol. 7, pp. 1–24, 1967.

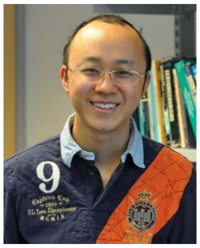
[34] J. More and G. Toralodo, "Algorithms for bound constrained quadratic-programming problems," *Numerische mathematik*, vol. 55, no. 4, pp. 377–400, 1989.

[35] A. Friedlander, J. M. Martinez, and M. Raydan, "A new method for large-scale box constrained convex quadratic minimization problems," *Optimisation Methods and Software*, vol. 5, no. 1, pp. 57–74, 1995.

[36] B. Chu, A. Rauh, H. Aschemann, and E. Rogers, "Experimental validation of constrained ilc approaches for a high speed rack feeder," in *Proceedings of the 2015 American Control Conference*, 2015, pp. 3631–3636.

[37] H. Aschemann and D. Schindele, "Model predictive trajectory control for high-speed rack feeders," in *Model Predictive Control*. InTech, 2010, pp. 183–198.

[38] H. Aschemann and J. Ritzke, "Gain-scheduled tracking control for high-speed rack feeders," in *Proceedings of the 1st Joint Intl. Conference on Multibody System Dynamics (IMSD)*, Lappeenranta, Finland, 2010.



Bing Chu received the B.Eng. degree in automation and the M.Sc. degree in control science and technology from Tsinghua University, Beijing, China, in 2004 and 2007, respectively, and the Ph.D. degree in automatic control and systems engineering from The University of Sheffield, Sheffield, U.K., in 2009.

He was a Post-Doctoral Researcher with the University of Oxford, Oxford, U.K., from 2010 to 2012. He is currently an Associate Professor in electronics and computer science with the University of Southampton, Southampton, U.K. His current research interests include iterative learning and repetitive control, analysis and control of large-scale networked systems, applied optimization theory, and their applications to robotics, power electronics, and next-generation healthcare. Dr. Chu was a recipient of a number of awards, including the Best Paper Prize of the 2012 United Kingdom Automatic Control Council International Conference on Control and the Certificate of Merit for the 2010 IET Control and Automation Doctoral Dissertation Prize.

Andreas Rauh was born in Munich, Germany, on March 25, 1977. He received his diploma degree in electrical engineering and information technology from the Technische Universität München, Munich, Germany, in 2001, his PhD degree (Dr.-Ing.) from the University of Ulm, Germany, in 2008, and his habilitation (Dr.-Ing. habil.) from the University of Rostock, Germany, in 2017. After a one-year research stay at ENSTA Bretagne (Brest, France), Andreas was appointed Full Professor in Oct. 2021 at the Carl von Ossietzky Universität Oldenburg, Germany, where he is heading the group *Distributed Control in Interconnected Systems*. He has published more than 170 articles and chapters in edited books, international conferences and peer-reviewed journals with a focus on modeling, control, as well as state and parameter estimation for systems with stochastic and set-valued uncertainty. Since 2008 he has been a member of the IEEE 1788 Working Group for the Standardization of Interval Arithmetic.

Harald Aschemann received his diploma degree in mechanical engineering from the University of Hanover, Germany, in 1994. After two years of work in research and development with a leading company in machine tools, he joined the Department of Measurement, Control, and Microtechnology at the University of Ulm, Germany. He completed his PhD in 2001 and served as a research associate and lecturer at the same department. Since 2006, Harald Aschemann has been a full professor and the head of the Chair of Mechatronics at the University of Rostock, Germany. His research interests involve control-oriented modelling, identification, optimization as well as nonlinear control and estimation with application to mechatronic, robotic and thermo-fluidic systems.

Eric Rogers is currently Professor of Control Systems Theory and Design at the School of Electronics and Computer Science, University of Southampton, UK. His major research interests are in iterative learning control theory and applications in both engineering and healthcare, multidimensional systems theory and applications, long range autonomy for autonomous underwater vehicles and flow control. He is the Editor-in-Chief of the International Journal of Control and Multidimensional Systems and Signal Processing and has an extensive record of advisory work for funding agencies and industries in the UK and elsewhere.

David H. Owens received the B.Sc. and A.R.C.S. degrees in physics and the Ph.D. degree in control engineering from the Imperial College London, London, U.K, in 1969 and 1973, respectively.

He is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China. He is also with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, U.K. His current research interests include robust optimization-based iterative learning control, repetitive control, and their applications in mechanical systems, rehabilitation robotics, and manufacture.