Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: enabling large feedforward tasks on a position-dependent printer

Mechatronics 38 (2016) 76–90

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer☆

Jurgen van Zundert[a,*], Joost Bolder[a], Sjirk Koekebakker[b], Tom Oomen[a]

[a] Control Systems Technology, department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[b] Océ Technologies B.V., Venlo, The Netherlands

ARTICLE INFO

Article history:
Received 15 April 2016
Revised 8 July 2016
Accepted 10 July 2016

Keywords:
Iterative learning control
Resource-efficient ILC
Stable inversion
LQ optimal control
Feedforward control design
Industrial application

ABSTRACT

Iterative learning control (ILC) enables high performance for systems that execute repeating tasks. Norm-optimal ILC based on lifted system representations provides an analytic expression for the optimal feedforward signal. However, for large tasks the computational load increases rapidly for increasing task lengths, compared to the low computational load associated with so-called frequency domain ILC designs. The aim of this paper is to solve norm-optimal ILC through a Riccati-based approach for a general performance criterion. The approach leads to exactly the same solution as found through lifted ILC, but at a much smaller computational load ($\mathcal{O}(N)$ vs $\mathcal{O}(N^3)$) for both linear time-invariant (LTI) and linear time-varying (LTV) systems. Interestingly, the approach involves solving a two-point boundary value problem (TPBVP). This is shown to have close connections to stable inversion techniques, which are central in typical frequency domain ILC designs. The proposed approach is implemented on an industrial flatbed printer with large tasks which cannot be implemented using traditional lifted ILC solutions. The proposed methodologies and results are applicable to both ILC and rational feedforward techniques by applying them to suitable closed-loop or open-loop system representations. In addition, they are applied to a position-dependent system, revealing necessity of addressing position-dependent dynamics and confirming the potential of LTV approaches in this situation.

1. Introduction

1.1. Iterative learning control in mechatronic applications

Mechatronic systems can be accurately positioned using control. With feedback control the command signal is updated based on past errors, namely the difference between the measured and desired output. With feedforward control information of the desired output is used to anticipate on future errors which enables accurate positioning. In this paper, the main focus is on learning such a feedforward command signal from data via iterative learning control (ILC).

ILC algorithms often achieve exceptional performance for systems that operate repetitively, i.e., systems that perform the same task over and over again. ILC [1,2] exploits the repetitive behavior of the system by learning from past executions. Many successful ILC applications in mechatronics have been reported, including wafer scanners [3,4], H-drive pick and place machines [5], and printing systems [6,7].

An important class in ILC is norm-optimal ILC [8], where the optimal feedforward is determined on the basis of a performance criterion. When representing the system in the lifted framework [9], an analytic expression can be directly obtained for the optimal ILC controller [10]. However, the implementation of lifted ILC involves multiplication and inversion of $N \times N$-matrices, with N the task length. Since the matrices scale with the task length, lifted ILC is impractical for large tasks. The need for computationally efficient techniques is well-recognized, see, for example, [11]. In addition, efficient techniques for the computation of the ILC convergence condition are developed, see, for example, [12].

Abbreviations: ILC, iterative learning control; LTI, linear time-invariant; LTV, linear time-varying; LQ, linear quadratic; RAM, random-access memory; TPBVP, two-point boundary value problem; ZPETC, zero phase error tracking control.

☆ This research is supported by the Dutch Technology Foundation STW, carried out as part of the Robust Cyber-Physical Systems (RCPS) project (no. 12694); and Innovational Research Incentives Scheme under the VENI grant "Precision Motion: Beyond the Nanometer" (no. 13073) awarded by NWO (The Netherlands Organization for Scientific Research).

* Corresponding author.
E-mail address: j.c.d.v.zundert@tue.nl (J.v. Zundert).

http://dx.doi.org/10.1016/j.mechatronics.2016.07.001
0957-4158/

1.2. Iterative learning control for large tasks

Several approaches in ILC have been used that have a significantly smaller computational burden compared to standard norm-optimal ILC algorithms. On the one hand, norm-optimal ILC has been extended in several ways. Basis functions in lifted ILC [13,14] lead to smaller computational burden. However, these basis functions are typically used to enhance extrapolation properties of ILC, at expense of performance. Alternatively, the Toeplitz/Hankel structure of the involved matrices can be exploited, see [15], as is done in [16].

On the other hand, alternative approaches to ILC are typically based on (non-optimal) two step approaches. These approaches include a learning filter obtained via typically noncausal plant inversion techniques and is completed by a robustness filter to guarantee convergence of the iteration. For instance, preview based control approaches such as zero phase error tracking control (ZPETC) are often applied in ILC. ZPETC was originally developed for noncausal feedforward compensation of non-minimum phase systems [17]. See [18] for a multivariable extension and [19] for related methods. Such ZPETC-related algorithms enable the design of noncausal filters and have computationally complexity $\mathcal{O}(N)$. However, they typically introduce approximation errors and are only applicable to linear time-invariant (LTI) systems.

Interestingly, stable inversion [20–22] has been abundantly used in ILC and feedforward, see, for example, [14,18,23,24] for high-tech motion control applications, and [25] for multivariable extensions. Such inversion methods enable noncausal inverses for square multivariable systems and immediately generalize to linear time-varying (LTV) systems. These methods essentially provide an exact inverse over a bi-infinite time horizon, but still introduce approximation errors for finite tasks. These errors are caused by incompatible initial conditions in the finite time case as a result of mixed causal/noncausal filtering.

1.3. Contributions

Although several frameworks and associated algorithms have been developed for ILC, there seems to be a trade-off between computational requirements and accuracy. One either has to accept a large computational time or approximation errors for non-optimal approaches. The aim of the present paper is to develop an optimal design algorithm for joint design of the learning and robustness filter. The algorithm exploits noncausality, is directly applicable to LTV systems, and addresses the finite time interval behavior through LTV designs for both LTI and LTV systems, while providing computational complexity $\mathcal{O}(N)$. In addition, the underlying solution involves solving a two-point boundary value problem (TPBVP). This is shown to have a direct connection to LTI/LTV stable inversion techniques, revealing very similar underlying mechanisms and proving a unified framework for both approaches.

The main contribution of this paper is to provide a systematic resource-efficient norm-optimal ILC framework which is implementable for large tasks, and is applicable to both LTI and LTV systems. The following five sub-contributions are identified.

I. The resource-efficient ILC approach is presented for LTI and LTV systems with general performance criteria, including derivations and proofs.
II. Connections to stable inversion are established, revealing very similar underlying mechanisms. These inversion techniques have recently received significant attention in ILC and feedforward. This in turn leads to a unified framework of norm-optimal ILC and stable inversion. Hence the techniques in the present paper apply to a large class of LTI/LTV ILC algorithms.
III. Through application of the resource-efficient ILC approach on an industrial flatbed printer with large tasks, the practical relevance is demonstrated as lifting techniques are unsuitable.
IV. The relevance of LTV feedforward and ILC is demonstrated on a position-dependent printer system, confirming the necessity to address position-dependent effects in this situation.
V. The computational load of the lifted ILC and the resource-efficient ILC approach are compared, revealing the significant advantages of the proposed approach.

The proposed approach is foreseen to facilitate the resource-efficient implementation of optimal ILC for LTI/LTV systems. Due to its inherent connection with stable inversion, it also enables the direct implementation of both square and non-square rational feedforward controllers [26,27]. Finally, it is foreseen for use as model inversion technique in frequency domain ILC designs.

1.4. Related results

Several results related to the ones presented here have appeared in the literature. As in typical ILC designs, the proposed approach is based on noncausal feedforward techniques for reference tracking, but then applied iteratively and on a closed-loop system transfer function. This is similar to applying ZPETC and stable inversion. In this paper, the approach is based on the classical linear quadratic (LQ) tracking controller [28, Ch. 9], which is well-known to be noncausal. It builds on the ILC approaches in [29,30] and exploits optimal control theory for computing the optimal feedforward signal. Since the state-space instead of the lifting framework is used for describing the system, the computational burden is much smaller. In particular, it extends the result in [29] for a more general performance criterion (including input weighting) and LTV systems. In [30], a related theoretical development is presented, linked to sampled-data ILC with intersample behavior. Further, related results are developed in [31], where a different LQ criterion is posed and in [32, Ch. 6], where an H_∞-type criterion is employed.

The LTV case is of particular importance for certain systems, including the position-dependent printer considered here, but also in very different applications, including nuclear fusion [33]. More general criteria with dedicated optimizations can be found in [34]. Finally, in [35], stable inversion is extended to deal with non-square systems to address non-square systems with more actuators using infinite horizon LQ theory. In the present paper, possible truncation effects for finite time implementations are explicitly addressed. It thereby possibly extends the results in [35] for situations where these boundary effects have a significant influence on the performance, as occasionally occurs in ILC.

1.5. Outline

The outline of this paper is as follows. In Section 2, ILC and norm-optimal ILC are formulated and the well-known analytic solution of lifted norm-optimal ILC is presented. Analysis of this solution reveals the computational challenges that come with actual implementation and motivates the development of a resource-efficient ILC approach. In Section 3, the resource-efficient ILC approach for LTI and LTV systems is presented based on LQ tracking which forms contribution I. Also, in Section 3, connections to stable inversion and a simulation case study of both approaches are presented, leading to contribution II. Many systems, including the position-dependent flatbed printer considered in this paper, can be modeled as LTV systems. The experimental setup of the flatbed printer is described in Section 4. The developed resource-efficient ILC approach can directly be applied on LTV models which significantly enhances the performance for LTV systems, as shown in

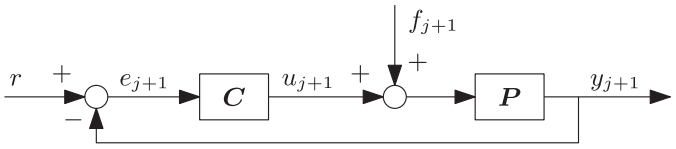


Fig. 1. ILC control diagram. The goal for trial $j+1$ is to minimize the error e_{j+1} by design of feedforward f_{j+1}.

Section 5 which constitutes contribution III. In Section 6, the potential of resource-efficient ILC is demonstrated on the industrial flatbed printer involving large tasks ($N = 100,000$), forming contribution IV. Finally, the computational load of resource-efficient ILC is compared with that of lifted ILC in Section 7 to illustrate the significant saving in computational time, leading to contribution V. Conclusions are given in Section 8.

1.6. Notation

Systems are linear, in discrete time, have n_i inputs and n_o outputs, and are indicated in boldface, e.g., $\boldsymbol{H}$. Let $x[k]$ denote a signal x at time k. Let $h^{ij}[k,l] \in \mathbb{R}^{n_o \times n_i}$ be the impulse response of the time-varying system $\boldsymbol{H}^{ij}[k]$ from the jth input $u^j[l]$ at time l, to the i-th output $y^i[k]$ at time k. The output $y^i[k]$ of the response of $\boldsymbol{H}^{ij}[k]$ to input u^j is given by $y^i[k] = \sum_{l=-\infty}^{\infty} h^{ij}[k,l]u^j[l]$. Let $N \in \mathbb{Z}^+$ denote the trial length, i.e., the number of samples per trial. Many results directly generalize to the continuous time case.

Variables related to the lifted framework, also called supervector notation [2], are underlined. Define the stacked input signal $\underline{u}[k] = \begin{bmatrix} u^1[k] & u^2[k] & \cdots & u^{n_i}[k] \end{bmatrix}^\top \in \mathbb{R}^{n_i \times 1}$ and similarly $\underline{y}[k] \in \mathbb{R}^{n_o \times 1}$ and $\underline{h}[k,l] \in \mathbb{R}^{n_o \times n_i}$. Assuming $\underline{u}[l] = \underline{0}$ for $l < 0$ and $l > N-1$, then the input-output relation in lifted notation is given by

$$\underbrace{\begin{bmatrix} \underline{y}[0] \\ \underline{y}[1] \\ \vdots \\ \underline{y}[N-1] \end{bmatrix}}_{\underline{y}} = \underbrace{\begin{bmatrix} \underline{h}[0,0] & \underline{h}[0,1] & \cdots & \underline{h}[0,N-1] \\ \underline{h}[1,0] & \underline{h}[1,1] & \cdots & \underline{h}[1,N-1] \\ \vdots & \vdots & \ddots & \vdots \\ \underline{h}[N-1,0] & \underline{h}[N-1,1] & \cdots & \underline{h}[N-1,N-1] \end{bmatrix}}_{\underline{H}} \times \underbrace{\begin{bmatrix} \underline{u}[0] \\ \underline{u}[1] \\ \vdots \\ \underline{u}[N-1] \end{bmatrix}}_{\underline{u}}$$

Let $\|\underline{x}\|_{\underline{W}} := \underline{x}^\top \underline{W} \underline{x}$, where $\underline{x} \in \mathbb{R}^{Nn_x}$ and $\underline{W} = \underline{W}^\top \in \mathbb{R}^{Nn_x \times Nn_x}$. $\underline{W}$ is positive definite ($\underline{W} \succ 0$) iff $\underline{x}^\top \underline{W} \underline{x} > 0$, $\forall \underline{x} \neq \underline{0}$ and positive semi-definite ($\underline{W} \succeq 0$) iff $\underline{x}^\top \underline{W} \underline{x} \geq 0$, $\forall \underline{x}$. In case the system $\boldsymbol{H}$ is LTI, then $h^{ij}[k,l]$ reduces to $h^{ij}[k-l]$, i.e., only depends on the relative time, and in which case $\underline{H}$ is Toeplitz.

2. Problem formulation

In this section, the problem is formulated by defining the ILC design problem and the norm-optimal performance criterion, and deriving and analyzing the analytic optimal solution for lifted ILC. This reveals the computational challenges of this solution, which in turn motivates the development of the resource-efficient ILC approach.

2.1. ILC and norm-optimal ILC

Consider the closed-loop configuration depicted in Fig. 1, with $\boldsymbol{P}$ the n_i-input, n_o-output system to control with outputs y_{j+1}, $\boldsymbol{C}$ a stabilizing feedback controller, and $e_{j+1} = r - y_{j+1}$ the error signal to be minimized. For repetitive tasks, the reference signal r has finite length and is independent of j. Each repetition/execution is called a trial and indicated with a subscript $j = 0, 1, 2, \ldots$. In iterative learning control, the goal is to minimize error e_{j+1} by design of the n_i-dimensional feedforward f_{j+1} based on data of previous trials (e_j, f_j). Typically, approximate models $\boldsymbol{P}$ and $\boldsymbol{C}$ are used.

In lifted notation (see Section 1), the error at trial j is

$$\begin{aligned} \underline{e}_j &= \underline{S}\underline{r} - \underline{S}\underline{P}\underline{f}_j \\ &= \underline{\tilde{r}} - \underline{J}\underline{f}_j, \end{aligned}$$

with (output) sensitivity $\underline{S} = \left(\underline{I}_{Nn_o} + \underline{P}\underline{C}\right)^{-1}$, (output) process sensitivity $\underline{J} = \underline{S}\underline{P}$, and $\underline{\tilde{r}} = \underline{S}\underline{r}$. Since $\underline{\tilde{r}}$ is trial-invariant, it follows that

$$\begin{aligned} \underline{e}_{j+1} &= \underline{\tilde{r}} - \underline{J}\underline{f}_{j+1} \\ &= \underline{e}_j - \underline{J}(\underline{f}_{j+1} - \underline{f}_j). \end{aligned} \tag{1}$$

Hence, to minimize $\underline{e}_{j+1}$, $\underline{f}_{j+1}$ can be based on a model $\underline{J}$ and data $\underline{e}_j, \underline{f}_j$.

2.2. Lifted norm-optimal ILC

An important class of ILC is norm-optimal ILC in which $\underline{f}_{j+1}$ follows from minimizing a performance criterion as given in Definition 1.

Definition 1 (Performance criterion)**.** A general performance criterion in norm-optimal ILC is given by

$$\mathcal{J}(\underline{f}_{j+1}) = \|\underline{e}_{j+1}\|_{\underline{W}_e} + \|\underline{f}_{j+1}\|_{\underline{W}_f} + \|\underline{f}_{j+1} - \underline{f}_j\|_{\underline{W}_{\Delta f}}, \tag{2}$$

with $\underline{W}_e \succ 0$, $\underline{W}_f, \underline{W}_{\Delta f} \succeq 0$, and $\underline{e}_{j+1}$ given by (1).

Since $\underline{e}_{j+1}$ is affine in $\underline{f}_{j+1}$, $\mathcal{J}(\underline{f}_{j+1})$ is quadratic in $\underline{f}_{j+1}$ and hence the optimal feedforward signal $\underline{f}^*_{j+1}$ can be computed analytically, as in, e.g. [10], from

$$\left.\frac{d\mathcal{J}(\underline{f}_{j+1})}{d\underline{f}_{j+1}}\right|_{\underline{f}_{j+1} = \underline{f}^*_{j+1}} = 0 \tag{3}$$

with solution provided by Theorem 2.

Theorem 2 (Solution lifted ILC)**.** *Given $\underline{J}^\top \underline{W}_e \underline{J} + \underline{W}_f + \underline{W}_{\Delta f} \succ 0$, an LTI or LTV model $\underline{J}$, and measurement data $\underline{f}_j, \underline{e}_j$, the optimal $\underline{f}^*_{j+1}$ for the performance criterion of Definition 1 is*

$$\underline{f}^*_{j+1} = \underline{Q}\underline{f}_j + \underline{L}\underline{e}_j, \tag{4}$$

with

$$\begin{aligned} \underline{\Gamma} &= \left(\underline{J}^\top \underline{W}_e \underline{J} + \underline{W}_f + \underline{W}_{\Delta f}\right)^{-1}, \\ \underline{Q} &= \underline{\Gamma}\left(\underline{J}^\top \underline{W}_e \underline{J} + \underline{W}_{\Delta f}\right), \\ \underline{L} &= \underline{\Gamma}\,\underline{J}^\top \underline{W}_e. \end{aligned} \tag{5}$$

Proof. Substitute (1) in (2) and solve (3) for $\underline{f}^*_{j+1} = \underline{f}_{j+1}$. □

Two key observations can be made from Theorem 2. First, the solution of Theorem 2 is time-varying ($\underline{Q}$, $\underline{L}$ not Toeplitz), even if model $\boldsymbol{J}$ is LTI ($\underline{J}$ Toeplitz). This is caused by taking the transpose $\underline{J}^\top$ and the inverse $\underline{\Gamma}$. Second, the solution in Theorem 2 is noncausal ($\underline{Q}$, $\underline{L}$ not necessarily lower triangular), even if the model $\boldsymbol{J}$ is causal ($\underline{J}$ lower triangular).

2.3. Computational challenges in lifted ILC

The derivation and calculation of $\underline{f}^*_{j+1}$ in Theorem 2 is elementary and involves straightforward matrix algebra. However, its direct implementation may be challenging from a computational perspective. For instance, the $Nn_i \times Nn_i$ matrix inversion in $\underline{\Gamma}$ via the

Gauss-Jordan method has time complexity $w = 3$, i.e., the computational time grows as $\mathcal{O}(N^3)$, see [36]. Although many methods have been developed to reduce the computational time, currently a time complexity of $w = 2.4$ seems to be the limit [36]. Also the matrix multiplications via Schoolbook matrix multiplication grow as $\mathcal{O}(N^3)$. Even when pre-computing $\underline{Q}$ and $\underline{L}$ off-line, a direct implementation of Theorem 2 is impractical when N becomes large since matrix-vector multiplication (via Schoolbook matrix multiplication) grows as $\mathcal{O}(N^2)$. These observations are experimentally demonstrated in Section 6, and intensively analyzed in Section 7.

In lifted ILC, matrices with dimensions in the order of $N \times N$ are used to describe the system $\boldsymbol{J}$ and the fact that a resource-efficient system with McMillan degree n_x underlies this input-output system is not recognized. Since typically $n_x \ll N$, lifted ILC is a resource-inefficient norm-optimal ILC approach, as will be shown in Section 7. In the following section, an alternative to Theorem 2 is presented that builds on well-known results in optimal control. The approach exploits state-space descriptions and provides a resource-efficient norm-optimal ILC approach.

3. Resource-efficient ILC

In this section, the resource-efficient ILC approach is presented, i.e., contribution I. The approach is an alternative to the lifted ILC approach (see Theorem 2) providing identical optimal performance, but at significantly smaller computational load. This makes resource-efficient ILC practical for experimental implementation of long tasks, as is demonstrated in Section 6. The difference in computational load is analyzed and demonstrated in Section 7. In the current section, also connections and a simulation comparison to stable inversion techniques are presented, constituting contribution II.

Remark 3. In the remainder, the argument k is suppressed for the system matrices. This is done to emphasize that the optimal ILC controller for an LTI system is LTV, which is an important property of such finite time optimal ILC controllers.

3.1. State-space description

In resource-efficient ILC the system $\boldsymbol{J}$ is described using a state-space description as provided by Lemma 4.

Lemma 4. *Let the LTV system $\boldsymbol{P}$ and the feedback controller $\boldsymbol{C}$ in Fig. 1 be described by the state-space realizations*

$$\boldsymbol{P} \overset{s}{=} \left[\begin{array}{c|c} A_P & B_P \\ \hline C_P & D_P \end{array}\right] \quad \text{and} \quad \boldsymbol{C} \overset{s}{=} \left[\begin{array}{c|c} A_C & B_C \\ \hline C_C & D_C \end{array}\right]. \tag{6}$$

Then, a state-space realization of the process sensitivity $\boldsymbol{J}$ is given by

$$\boldsymbol{J} \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right],$$

with

$$A = \begin{bmatrix} A_P - B_P(I_{n_i} + D_C D_P)^{-1} D_C C_P & B_P(I_{n_i} + D_C D_P)^{-1} C_C \\ -B_C(I_{n_o} + D_P D_C)^{-1} C_P & A_C - B_C(I_{n_o} + D_P D_C)^{-1} D_P C_C \end{bmatrix}$$

$$B = \begin{bmatrix} B_P(I_{n_i} + D_C D_P)^{-1} \\ -B_C(I_{n_o} + D_P D_C)^{-1} D_P \end{bmatrix},$$

$$C = \begin{bmatrix} (I_{n_o} + D_P D_C)^{-1} C_P & (I_{n_o} + D_P D_C)^{-1} D_P C_C \end{bmatrix},$$

$$D = (I_{n_o} + D_P D_C)^{-1} D_P,$$

and for $D_P = 0$

$$\boldsymbol{J} \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} A_P - B_P D_C C_P & B_P C_C & B_P \\ -B_C C_P & A_C & 0 \\ \hline C_P & 0 & 0 \end{array}\right].$$

Proof. See Appendix A. □

3.2. Optimal solution

Typically, diagonal performance weights are chosen in Definition 1, see, e.g., [10,13], i.e., $W_e = \mathrm{diag}(w_e[k])$ with $w_e[k] \in \mathbb{R}^{n_o \times n_o}$, $W_f = \mathrm{diag}(w_f[k])$ with $w_f[k] \in \mathbb{R}^{n_i \times n_i}$, and $W_{\Delta f} = \mathrm{diag}(w_{\Delta f}[k])$ with $w_{\Delta f}[k] \in \mathbb{R}^{n_i \times n_i}$. For this choice, Definition 1 is equivalent to Definition 5.

Definition 5 (Performance criterion diagonal weights)**.** The performance criterion with diagonal weights is given by

$$\begin{aligned} \mathcal{J}(f_{j+1}) = \sum_{k=0}^{N-1} & e_{j+1}^\top[k] w_e[k] e_{j+1}[k] + f_{j+1}^\top[k] w_f[k] f_{j+1}[k] \\ & + \big(f_{j+1}[k] - f_j[k]\big)^\top w_{\Delta f}[k] \big(f_{j+1}[k] - f_j[k]\big) \end{aligned} \tag{7}$$

with $w_e[k] > 0$, $w_f[k] \geq 0$, $w_{\Delta f}[k] \geq 0$, $\forall k$.

Resource-efficient ILC determines the optimal feedforward for the performance criterion of Definition 5 and is provided by Theorem 6.

Theorem 6 (Solution resource-efficient LTI/LTV ILC)**.** *Let the model $\boldsymbol{J}$ of the process sensitivity have the LTI/LTV state-space realization (A, B, C, D), with n_i inputs, n_o outputs, and state dimension n_x, see also Lemma 4. Then, for the performance criterion of Definition 5, f_{j+1}^* is the output of the state-space system*

$$\left[\begin{array}{c|cccc} A - BL[k] & -BL_f[k] & BL_e[k] & BL_g[k] \\ \hline -L[k] & I_{n_i} - L_f[k] & L_e[k] & L_g[k] \end{array}\right], \tag{8}$$

with zero initial state for input $\begin{bmatrix} f_j[k] \\ e_j[k] \\ g_{j+1}[k+1] \end{bmatrix}$, where

$$\begin{aligned} L[k] &= \big(\gamma^{-1}[k] + B^\top P[k+1] B\big)^{-1} \big(D^\top w_e[k] C + B^\top P[k+1] A\big), \\ L_f[k] &= \big(\gamma^{-1}[k] + B^\top P[k+1] B\big)^{-1} w_f[k], \\ L_e[k] &= \big(\gamma^{-1}[k] + B^\top P[k+1] B\big)^{-1} D^\top w_e[k], \\ L_g[k] &= \big(\gamma^{-1}[k] + B^\top P[k+1] B\big)^{-1} B^\top, \\ \gamma[k] &= \big(D^\top w_e[k] D + w_f[k] + w_{\Delta f}[k]\big)^{-1}, \end{aligned} \tag{9}$$

with

$$\begin{aligned} g_{j+1}[k] = & \big(A^\top - K_g[k] B^\top\big) g_{j+1}[k+1] + C^\top w_e[k] e_j[k] \\ & + K_g[k] w_f[k] f_j[k], \end{aligned} \tag{10}$$

$$g_{j+1}[N] = 0_{n_x \times 1},$$

where

$$\begin{aligned} K_g[k] = & \big(A^\top - C^\top w_e[k] D \gamma[k] B^\top\big) P[k+1] \\ & \times \big(I_{n_x} + B\gamma[k] B^\top P[k+1]\big)^{-1} B\gamma[k], \end{aligned}$$

and $P[k]$ the solution of the matrix difference Riccati equation

$$\begin{aligned} P[k] = & \big(A^\top - C^\top w_e[k] D \gamma[k] B^\top\big) P[k+1] \\ & \times \Big(I_{n_x} - B\big(\gamma^{-1}[k] + B^\top P[k+1] B\big)^{-1} B^\top P[k+1]\Big) \\ & \times \big(A - B\gamma[k] D^\top w_e[k] C\big) \\ & + C^\top w_e[k] C - C^\top w_e[k] D \gamma[k] D^\top w_e[k] C, \end{aligned}$$

$$P[N] = 0_{n_x \times n_x}. \tag{11}$$

Proof. See Appendix B. □

Remark 7. For $D = 0$ and $w_f[k] + w_{\Delta f}[k] = R$, (11) reduces to

$$\begin{aligned} P[k] = & C^\top w_e[k] C + A^\top P[k+1] A \\ & - A^\top P[k+1] B \big(B^\top P[k+1] B + R\big)^{-1} B^\top P[k+1] A \end{aligned}$$

which is the well-known discrete time dynamic Riccati equation.

Note that time index k is explicitly indicated only for some elements in (8), while it is suppressed for others. The main reason is to illustrate the following interesting aspects, see also Remark 3. If $\boldsymbol{P}$ and $\boldsymbol{C}$ are LTI, then optimal ILC in (8) is LTV, and time-variance of any of the matrices in (6) affects all elements in (8) through $L[k]$, $L_f[k]$, $L_e[k]$, $L_g[k]$. Note that LTV designs are a key advantage for LTI systems, since these essentially handle the boundary effects for finite time trials [37]. To see this, by definition of the optimal feedforward, the result of Theorem 6 is the same as the result of Theorem 2, computed using a different approach. It is well-known that lifted ILC allows for time-varying and noncausal feedforward signals. This is reflected in the matrices $\underline{Q}$ and $\underline{L}$ of (5) being not Toeplitz and not lower-triangular, respectively. These aspects can also be observed in the results of Theorem 6: the dependence on k of the state-space matrices (8) reflects time-variance, whereas solving part of the equations backwards in time reflects noncausality and is closely related to stable inversion techniques.

Algorithm 8 provides a step-by-step procedure for implementing the results of Theorem 6. Note that step 1) can be performed off-line and steps 2) and 3) form the trial update.

Algorithm 8 The resource-efficient f^*_{j+1} is calculated as:

1. Solve the matrix difference Riccati equation (11) backwards in time.
2. Calculate $g_{j+1}[k]$ by solving (10) backward in time.
3. Calculate $f^*_{j+1}[k]$ forward in time as the output of state-space system (8).

It should be noted that the solutions of Theorem 2 and Theorem 6 are exactly the same. Hence, no performance is sacrificed, however the computational approaches do differ. In particular, the calculations in Theorem 6 scale with N instead of N^3 as in Theorem 2, see also Section 2.3. Therefore the resource-efficient ILC approach delivers a significant reduction in computational cost at a small expense of diagonal time-varying weighting filters, see Definition 5. As a result, resource-efficient ILC is well-suited for large tasks as will be demonstrated in Section 6 by implementing the approach on an industrial setup. Next, connections to stable inversion are highlighted.

3.3. Stable inversion

Theorem 6 reveals that solutions for resource-efficient ILC for both LTI and LTV systems are $\mathcal{O}(N)$. Interestingly, the results and proof of Theorem 6 have a very close connection to algorithms used in, i.a., frequency domain ILC designs and rational feedforward control [27]. In particular, in both cases a rational model $\boldsymbol{H}$ has to be inverted as $\boldsymbol{F} = \boldsymbol{H}^{-1}$, where $\boldsymbol{H} = \boldsymbol{P}$ for rational feedforward and $\boldsymbol{H} = \boldsymbol{J}^{-1} = (\boldsymbol{SP})^{-1}$ for the ILC structure in Section 2. Let $\boldsymbol{H}$ be square, invertible, and have state-space realization ($A_H[k]$, $B_H[k]$, $C_H[k]$, $D_H[k]$), then,

$$\boldsymbol{F} \overset{s}{=} \left[\begin{array}{c|c} A_H[k] - B_H[k]D_H^{-1}[k]C_H[k] & B_H[k]D_H^{-1}[k] \\ \hline -D_H^{-1}[k]C_H[k] & D_H^{-1}[k] \end{array}\right]. \tag{12}$$

Note that the system $\boldsymbol{F}$ in (12) may be unstable. For instance, in the case where $\boldsymbol{H}$ is LTI and has non-minimum phase zeros, then $\boldsymbol{F}$ has unstable poles.

A traditional solution in feedforward and ILC to deal with such unstable poles is ZPETC [17], which leads to an approximate inverse that is noncausal with a certain finite preview. However, it is by definition an approximation, see also [19] where different approximations are evaluated, and does not address the finite time aspect of practical feedforward and ILC implementations. In addition, extension to multivariable systems is practically not trivial, see [18] for results in this direction.

In stable inversion, the unstable part is seen as a noncausal operator and solved backwards in time as a stable system. For the general time-varying system (12), this means that the system has to be split in a stable and unstable part, which is not trivial for time-varying systems [38,39]. If such a split is found, Theorem 9 can be applied. When poles are on the unit circle, the techniques in [40,41] can be exploited. Note that if the state matrix of $\boldsymbol{F}$ in (12) is independent of k, then such a split follows directly from an eigenvalue decomposition as in Corollary 10.

Theorem 9. *Let an LTV system be split as*

$$\begin{aligned} x_s[k+1] &= A_{ss}[k]x_s[k] + A_{su}[k]x_u[k] + B_s u[k], \\ x_u[k+1] &= A_{us}[k]x_s[k] + A_{uu}[k]x_u[k] + B_u u[k], \\ y[k] &= \begin{bmatrix} C_s[k] & C_u[k] \end{bmatrix} \begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix} + D[k]u[k], \end{aligned} \tag{13}$$

where $x_s[k]$ is picking up the stable part with $x_s[0] = x_s^0$ and $x_u[k]$ the unstable part with $x_u[N] = 0$. Then, to find the bounded solution $y[k]$, solve for $P[k]$ backward in time using

$$\begin{aligned} P[k] &= (A_{uu}[k] - P[k+1]A_{su}[k])^{-1}(P[k+1]A_{ss}[k] - A_{us}[k]), \\ P[N] &= 0, \end{aligned} \tag{14}$$

and for $g[k]$ backward in time using

$$\begin{aligned} g[k] &= (P[k+1]A_{su}[k] - A_{uu}[k])^{-1} \\ &\quad \times (B_u[k]u[k] - P[k+1]B_s[k]u[k] - g[k+1]), \\ g[N] &= 0. \end{aligned} \tag{15}$$

Then, $x_s[k]$ can be solved forward in time from

$$\begin{aligned} x_s[k+1] &= (A_{ss}[k] + A_{su}[k]P[k])x_s[k] + B_s[k]u[k] + A_{su}[k]g[k] \\ x_s[0] &= x_s^0, \end{aligned}$$

and $x_u[k]$ follows from

$$x_u[k] = P[k]x_s[k] + g[k].$$

Output $y[k]$ follows directly from

$$y[k] = [cC_s[k] \quad C_u[k]] \begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix} + D[k]u[k].$$

Proof. See Appendix C. □

Corollary 10. *For systems $\boldsymbol{F}$ with time-invariant state matrix, the following procedure can be followed.*

1. *Let $\boldsymbol{F}$ have the state-space realization*

$$\begin{aligned} x[k+1] &= Ax[k] + B[k]u[k], \\ y[k] &= C[k]x[k] + D[k]u[k], \end{aligned}$$

with $x[k] = x_0$.

2. *Introduce the state transformation*

$$x[k] = T \begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix},$$

where T contains eigenvectors of A as columns such that

$$\begin{aligned} \begin{bmatrix} x_s[k+1] \\ x_u[k+1] \end{bmatrix} &= \begin{bmatrix} A_s & 0 \\ 0 & A_u \end{bmatrix} \begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix} + \begin{bmatrix} B_s[k] \\ B_u[k] \end{bmatrix} u[k], \\ y[k] &= \begin{bmatrix} C_s[k] & C_u[k] \end{bmatrix} \begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix} + D[k]u[k], \end{aligned}$$

with $\lambda(A_s) \subset \bar{\mathbb{D}}$ and $\lambda(A_u) \cap \bar{\mathbb{D}} = \emptyset$, where $\bar{\mathbb{D}}$ is the closed unit disk and $\lambda(\cdot)$ the set of eigenvalues, i.e., all stable poles are contained in A_s and all unstable poles in A_u.

3. *Solve*

$$x_s[k+1] = A_s x_s[k] + B_s[k]u[k], \quad x_s[0] = x_s^0$$

forward in time,

$$x_u[k+1] = A_u x_u[k] + B_u[k]u[k], \quad x_u[N] = x_u^N$$

backward in time via

$$x_u[k] = (A_u)^{-1}x_u[k+1] - (A_u)^{-1}B_u[k]u[k].$$

Then, $y[k]$ follows directly from

$$y[k] = [C_s[k] \quad C_u[k]]\begin{bmatrix} x_s[k] \\ x_u[k] \end{bmatrix} + D[k]u[k].$$

Remark 11. The result of Corollary 10 is a special case of Theorem 9 with $A_{ss}[k] = A_s$, $A_{su}[k] = 0$, $A_{us}[k] = 0$, $A_{uu}[k] = A_u$, yielding $P[k] = 0$, $\forall k$ and $x_u[k] = g[k]$.

Remark 12. For motion control systems the states typically represent time-derivatives of the position such as velocity and acceleration. Hence, if the system is initially at rest, then $x_s^0 = 0$. The terminal condition x_u^N should ideally be chosen such that $x_u[0]$ matches with $x[0]$. It can, however, not be derived a priori from $x_u[0]$ since this requires simulation of the unstable system.

A key observation is that both the results in Theorem 6, Theorem 9, and Corollary 10 involve a two-point boundary value problem. In addition, a very similar sweep method is used for the actual solution, albeit applied to a system with smaller dimension in case of Theorem 9 and Corollary 10. The following remarks are appropriate.

- The stable inversion approach provides an exact inverse in the case where the initial conditions are taken at $x_s[-\infty]$ and $x_u[\infty]$ in Theorem 9. In the case where they are finite, as is specified in Theorem 9 and Corollary 10, then this may lead to an incorrect initial state, leading to an inexact inverse and boundary effects. These boundary effects depend on the location of the non-minimum phase zeros and can be mitigated if the preview length is extended, i.e., by preceding the input with zeros, see, e.g., [42], or by using an approach as in Theorem 6.
- The stable inversion technique delivers an LTI inverse in case where the system is LTI, whereas the approach in Theorem 6 (i.e., LQ tracking based) generally leads to an LTV solution even if the original system is LTI.
- In case the original system is LTV, then notice that time variance of any of the entries, i.e. A, B, C, or D, implies that the inverse has a time-dependent state matrix, see $\boldsymbol{F}$ in (12), in which case Theorem 9 has to be used instead of the simpler version in Corollary 10. This in turn necessitates the dichotomic split in stable and unstable dynamics in (13). This split is not necessary in the case Theorem 6 is applied, which may be preferred in practical applications, e.g., systems with varying sensor locations or actuators, such as the moving-mass mechatronic stage in [43].
- Both Theorem 9 and Corollary 10 require that D_H is invertible. This requires that $\boldsymbol{H}$ is square. In the case that D_H is non-invertible, additional steps of preview can be added, i.e., forward shift operators, as is done in, e.g., ZPETC [17].
- In contrast to stable inversion in Theorem 9 and Corollary 10, the optimal approach in Theorem 6 does not require the system to be square or strictly proper due to the input weighting. As a result, it also applies to next-generation motion systems where additional sensors and actuators will be exploited [43, sec. 1], [44].
- Theorem 6 is very closely related to LQ optimal control, for which it is well-known that the LQ solution mirrors the unstable poles with respect to the unit disc, see also [45, sec. 6.1] for vanishing input weight. As a result, it is very closely related to the solution of the stable inversion approach in Theorem 9, where essentially a slightly different split is made. Notice that in this case, the TPBVP in Theorem 6 is twice the size of the one in Theorem 9. In this case, both solutions are present in the TPBVP, where a suitable selection is made in standard feedback control. Interestingly, the solution to the stable inversion problem is thus contained in the LQ tracking solution, where a different selection is made.
- Stable inversion can also be directly seen in an input-output setting. In the LTI case, $\boldsymbol{H}$ is decomposed as $\boldsymbol{H}_{stab}$, $\boldsymbol{H}_{unstab}$, which essentially can be solved using a bilateral instead of a unilateral Laplace/Z-transform [46], see also [47, sec. 1.5].

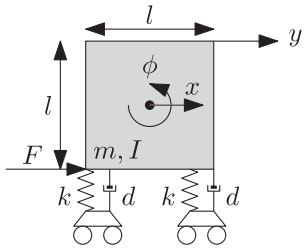


Fig. 2. The flexible cart system, consisting of a single mass mounted on two spring-damper combinations, is subject to input force F and has translation and rotation freedom x and ϕ, respectively. Position y is the output.

Table 1
Parameter values of the flexible cart system.

Parameter	Symbol	Value	Unit
Mass	m	8	kg
Inertia	I	0.0133	kgm^2
Spring constant	k	10^4	N/m
Damping constant	d	10	Ns/m
Length	l	0.1	m

3.4. Resource-efficient ILC and stable inversion

The stable inversion techniques presented in the previous section can be used to determine an exact bounded inverse of a non-minimum phase system. In the previous section, it is highlighted that these techniques are very similar to those used in the proposed resource-efficient ILC solution, see Section 3.2. In particular, for vanishing input weighting, the resource-efficient ILC solution converges to the optimal, possibly noncausal, inverse solution, as is shown in this section via a simulation example.

Consider the mechanical system shown in Fig. 2, with parameters listed in Table 1.

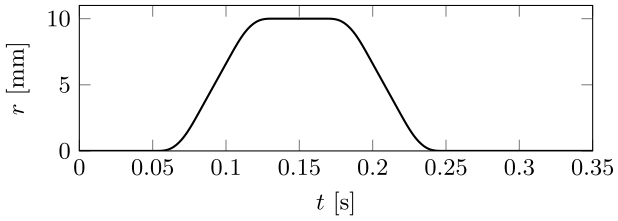
The *continuous*-time state-space realization (A_c, B_c, C_c, D_c) of the linearized system dynamics with input F, state $q = [x \quad \dot{x} \quad \phi \quad \dot{\phi}]^\top$, and output y is

$$\left[\begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2}\frac{kl^2}{I} & -\frac{1}{2}\frac{dl^2}{I} & \frac{1}{2}\frac{l}{I} \\ \hline 1 & 0 & -\frac{1}{2}l & 0 & 0 \end{array}\right].$$

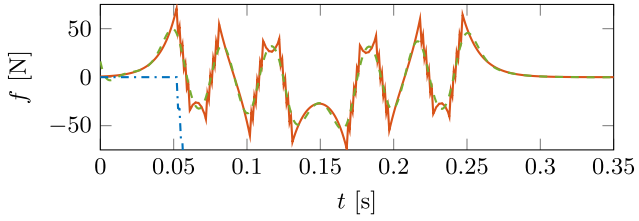
Assuming zero-order-hold on the input, system $\boldsymbol{P}$ has discrete state-space realization

$$\boldsymbol{P} \overset{s}{=} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} e^{A_c h} & A_c^{-1}(A-I)B_c \\ \hline C_c & D_c \end{array}\right],$$

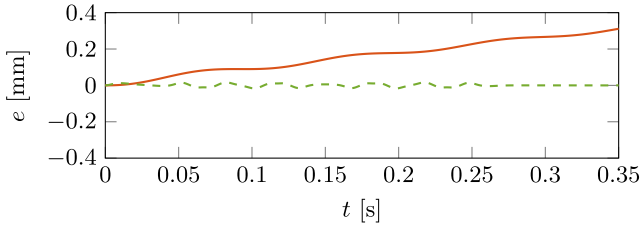
with sampling interval $h = 0.001$ s. The system is in open-loop, i.e., $\boldsymbol{C} = 0$ in Fig. 1, with reference trajectory r as depicted in Fig. 3a.



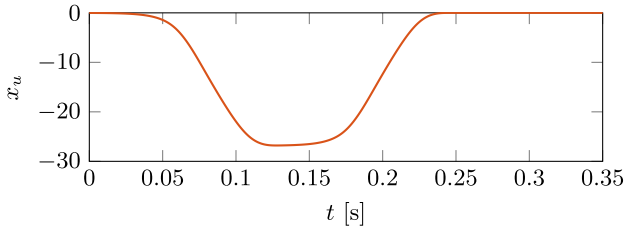
(a) Reference trajectory r is a fourth-order forward-backward movement.



(b) Feedforward f with stable inversion (——) and resource-efficient ILC (- - -) are bounded and almost equal. For regular inversion (-·-·-) f is unbounded.

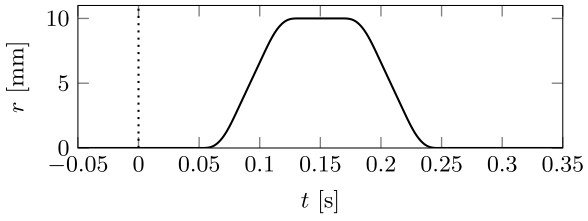


(c) Error e with stable inversion (——) is nonzero, whereas with resource-efficient ILC (- - -) it is almost zero.

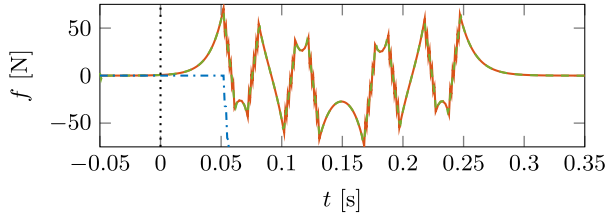


(d) Unstable state for stable inversion converges to zero when $t \to -\infty$.

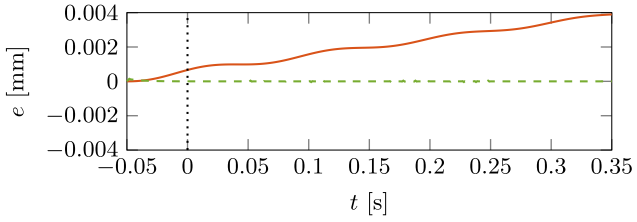
Fig. 3. Regular inversion yields an unbounded f. The error with stable inversion is larger than for resource-efficient ILC due to the nonzero unstable state at $t = 0$.



(a) $N_{pre} = 50$ additional zero samples are added prior to the reference in Fig. 3a.



(b) Feedforward f with stable inversion (——) and resource-efficient ILC (- - -) are more alike than in Fig. 4b. For regular inversion (-·-·-), f is unbounded.



(c) Preview information reduces the error for both stable inversion (——) and resource-efficient ILC (- - -).

Fig. 4. Regular inversion yields an unbounded f. Preview information significantly reduces the error for stable inversion.

The resulting system $\boldsymbol{SP}$ has one non-minimum phase zero and no direct feedthrough ($D = 0$). Since $D = 0$, stable inversion requires additional preview steps, see also the fourth remark at the end of the previous section. Note that preview information is not required for resource-efficient ILC.

Due to the non-minimum phase zero, direct inversion yields an unbounded response f, see Fig. 3b. With stable inversion, i.e., Corollary 10, bounded input f and error e shown in Fig. 3b and Fig. 3c are obtained, respectively. By selecting $w_{\Delta f}[k] = 0,\ \forall k$ in resource-efficient ILC, see Theorem 6 and Algorithm 8, one-step convergence of f is obtained. The result for $w_e[k] = 1,\ w_f[k] = 10^{-12},\ w_{\Delta f}[k] = 0,\ \forall k$, is also shown in Fig. 3b and Fig. 3c. The small error indicates the high quality of the inverse.

Due to the finite-length task, the solution for stable inversion is not exact. The inverse system $\boldsymbol{SP}^{-1}$ contains one unstable state which is simulated as a stable system backwards in time with zero terminal condition, see also Corollary 10. The evolution of the unstable state is depicted in Fig. 3d. For determining the response of system $\boldsymbol{SP}$, zero initial state is assumed. However, since the unstable state of $\boldsymbol{SP}^{-1}$, see Fig. 3d, is not exactly zero at $t = 0$ ($x_u(t = 0) = -0.0158$), this assumption is violated resulting in non-exact output as shown in Fig. 3c.

Consider the case where preview information is available, i.e., the reference is zero for $t < 0$, then the unstable state converges to zero for decreasing t since the system is stable in backward time. The result when $N_{pre} = 50$ samples are introduced prior to the reference task of Fig. 3 is shown in Fig. 4. Here, $x_u(t = -0.05) = -1.73 \cdot 10^{-4}$. This confirms that for $N_{pre} \to \infty$ exact results are obtained.

This simulation case study shows that stable inversion yields non-exact results for finite-length tasks, whereas the proposed resource-efficient approach handles finite time conditions approximately.

4. Experimental setup: industrial flatbed printer

In the next sections, the resource-efficient ILC approach introduced in the previous section is validated on an industrial printer, namely the Océ Arizona 550 GT flatbed printer shown in Fig. 5.

In contrast to conventional consumer printers, the medium on the flatbed printer is fixed on the printing surface using vacuum and the print heads move in two directions. The print heads are located in the carriage which can move in one direction over the gantry, which moves in perpendicular direction over the printing surface. The moving mass of the carriage is approximately

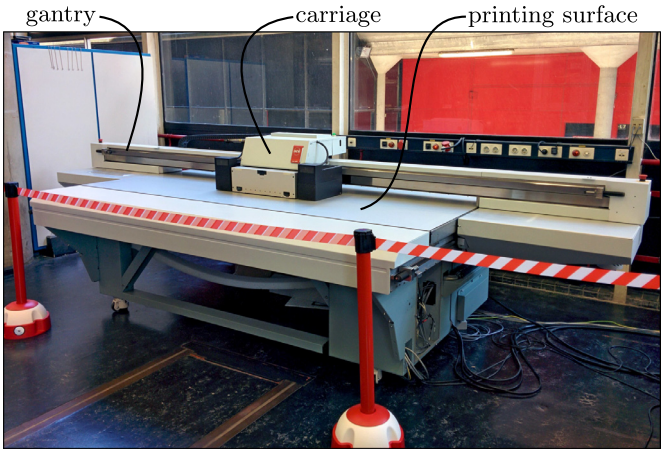


Fig. 5. Industrial flatbed printer. The print heads of the Arizona flatbed printer are located in the carriage which can move over the gantry. The gantry can move in perpendicular direction and rotate over small angles around the vertical axis.

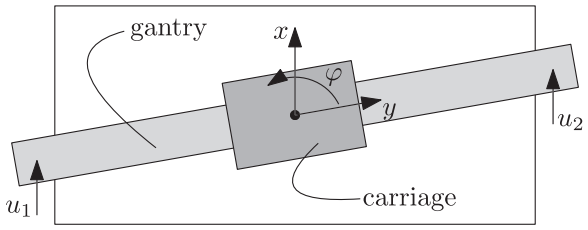


Fig. 6. Schematic top view of the flatbed printer. The gantry translation x and rotation φ are controlled via DC motors u_1, u_2.

32 kg, the maximum medium size is 2.5 × 1.25 m, and the maximum medium thickness is 50.8 mm. A schematic top view of the system is provided in Fig. 6. The system is controlled via Matlab/Simulink that is running on a host computer connected to a separate xPC target computer. On the target computer the application runs in real-time with a sampling frequency of 1000 Hz. After each trial, the ILC algorithm is executed on the host computer and the resulting feedforward signals are uploaded to a lookup table on the target computer via Ethernet. During the real-time execution of the task, the feedforward signals are read from the lookup table.

The validation of resource-efficient ILC is based on the gantry system of the flatbed printer since i) printer tasks are typically large, ii) it is a MIMO system, iii) it is position-dependent, and iv) it is a practically relevant system. The gantry position is controlled through two brushed DC motors (u_1 and u_2) and the position is measured through linear encoders with a resolution of 1 μm. Decoupling into a gantry translation and rotation yields a system with inputs u_x, u_φ ($n_i = 2$) and outputs x, φ ($n_o = 2$). The system operates at a sample frequency of 1 kHz. Depending on the application, different models $\boldsymbol{P}$ of the gantry system are used and introduced when appropriate.

For both the time-varying system model considered in Section 5 as well as the actual experimental system considered in Section 6, the same feedback controller $\boldsymbol{C}$ is used, shown in Fig. 7. The controller achieves a bandwidth of approximately 5 Hz for the diagonal terms. A diagonal controller suffices since for low frequencies the system is decoupled and feedback is only effective until the bandwidth. ILC is effective until much higher frequencies where interaction also plays a significant role. Hence, the full MIMO model is used in ILC.

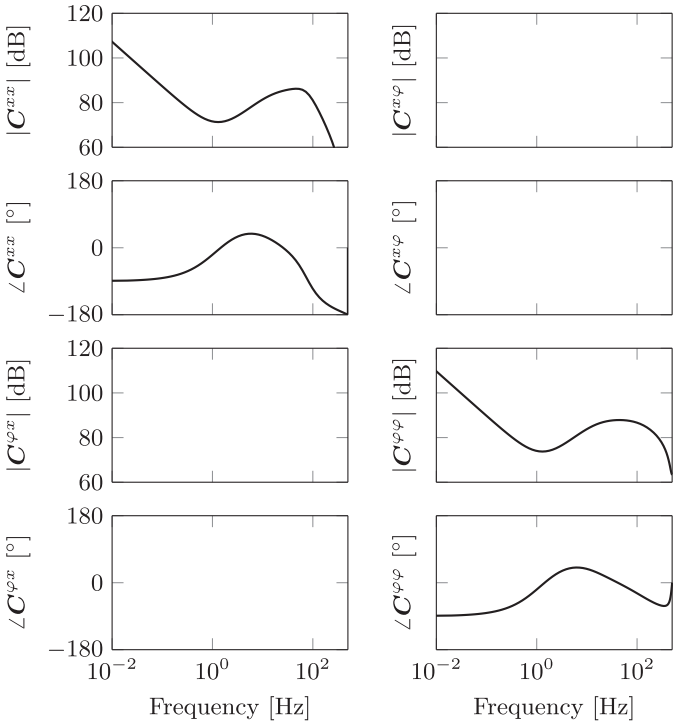


Fig. 7. Bode diagram of diagonal controller $\boldsymbol{C}$. Note that the off-diagonals $\boldsymbol{C}^{x\varphi}$ and $\boldsymbol{C}^{\varphi x}$ are empty.

5. Resource-efficient ILC simulation for the position-dependent printer system

In this section, resource-efficient ILC based on LTI and LTV models is simulated on an LTV model of the flatbed printer, forming contribution III. The system is position-dependent, hence the dynamics vary during motion. The results reveal the potential of using LTV models based on linearization around trajectories when compared to LTI models for fixed positions.

5.1. Position-dependent system

The flatbed printer system introduced in the previous section is considered, see also Figs. 5 and 6. Since the system is inherently position-dependent, a first-principles model is derived to analyze its effect in a simulation study. Due to the moving carriage mass of approximately 32 kg, this model is position-dependent. Given a trajectory y, the first principles model can be linearized around this trajectory resulting in an LTV model of the gantry system. Note that linearization around a trajectory is also done in, for example, [33]. Fig. 8 shows Bode diagrams of the closed-loop gantry system $\boldsymbol{J}$ for different carriage positions y, with $y = 0$ at the left side of the table. The feedback controller in Fig. 7 is used for feedback.

5.2. Reference trajectories and performance weights

The carriage trajectory is designed to cover the whole range area of 3.2 m in y-direction and 0.8 m in x-direction, see Fig. 9. To keep the results insightful, the rotation φ is controlled using feedback only, with $r_\varphi[k] = 0$, $\forall k$, and ILC is only applied in x-direction. The simulated system is thus a multi-input, multi-output system (2 × 2) and ILC is applied to a multi-input, single output system (1 × 2). Note that, although ILC is only applied to x, position y influences the system model through time dependency and rotation φ through the strong cross-coupling (except for $y \approx 1.6$ m), see Fig. 8.

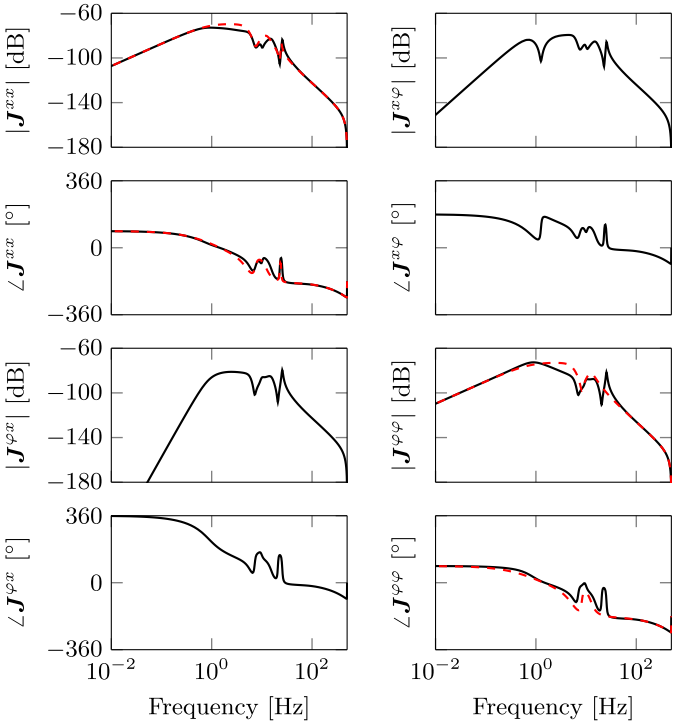


Fig. 8. Parametric closed-loop gantry model $\boldsymbol{J}$ for carriage position $y = 1.6$ m (- - -) and $y = 6$ m (———). For $y = 1.6$ m, there is no cross-coupling.

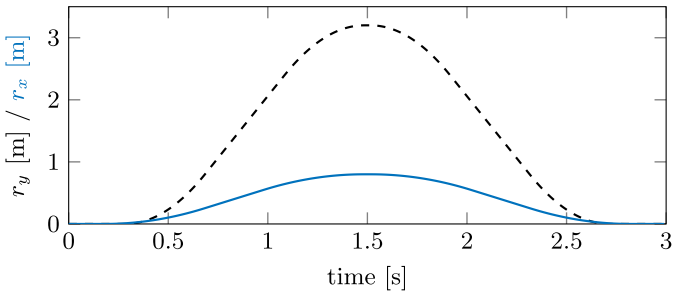


Fig. 9. Reference trajectory r_y (———) introduces position-dependent dynamics due to the moving carriage mass. Reference r_x (——) is a forward-backward movement. Rotation φ is suppressed, i.e., $r_\varphi = 0$.

The weights (see (7)) are selected as $w_e[k] = 10^{10}$, $w_f[k] = 10^{-10}$, $w_{\Delta f}[k] = 0$ in order to achieve high performance in the error norm and fast convergence (after convergence, f_j is in the order of 10^0 and e_j is in the order of 10^{-6}).

5.3. Results

Fig. 10 shows the performance criterion for ILC based on LTI models at several positions y, together with ILC based on the LTV model. When based on the LTV model (-o-), one-step convergence is obtained since the model is exact and $w_{\Delta f} = 0$. In this case, an accurate LTV model is available due to the fact that a first-principles model is derived. As will become clear in Section 6, such models are not straightforward to obtain in practice. Since in y-direction the carriage covers the whole working range (from 0 m to 3.2 m), an obvious choice when using an LTI model would be to use the LTI model with the carriage positioned in the middle of the gantry, i.e., $y = 1.6$ m. With ILC based on this 'averaged' LTI model the convergence is slower (· + ·), due to the model mismatch, but eventually the same high performance as with the LTV model is obtained. If a poor LTI model is chosen, for example at $y = 6$ m, there is no convergence (· ✱ ·). Convergence can be guaranteed by introducing robustness through increasing w_f [48]. Indeed, for $w_f = 10^2$ there is convergence (· ✱ ·), but at the cost of performance. Note that $y = 6$ m is not feasible for the current system, but might become so for larger printing systems. The result stresses the need for identification of accurate position-dependent models which is part of future research.

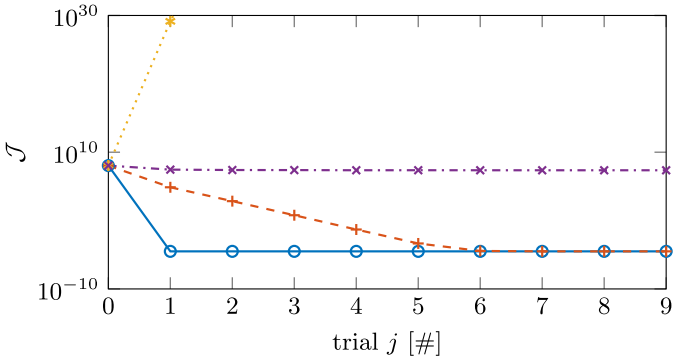


Fig. 10. Performance criterion for simulations on an LTV system with ILC based on: LTV model, $w_f = 10^{-10}$ (-o-); LTI model at $y = 1.6$ m, $w_f = 10^{-10}$ (· + ·); LTI model at $y = 6$ m, $w_f = 10^{-10}$ (· ✱ ·); and LTI model at $y = 6$ m, $w_f = 10^2$ (· ✱ ·). The LTI model at $y = 6$ m requires additional robustness (larger w_f) to converge.

The simulation example shows the benefit of ILC based on an LTV model when the system to control is LTV. For accurate LTI models, high performance is still achievable but at the cost of slower convergence, whereas for inaccurate LTI models performance needs to be sacrificed to guarantee convergence. Importantly, resource-efficient ILC in Algorithm 8 can directly be applied to LTV models, while preserving computational cost $\mathcal{O}(N)$.

6. Experimental implementation

In this section, the resource-efficient ILC approach is applied to the industrial flatbed printer in an experiment with task length $N = 100,000$ which forms contribution IV.

6.1. System modeling

The previous section shows the importance of an accurate system model to obtain both fast convergence and high performance in the error norm. However, the accuracy of the derived position-dependent model based on first principles is limited and identification of accurate position-dependent models is part of ongoing research. Still, for the considered range of operation, the simulation study in Section 5 reveals that an LTI model is sufficiently accurate to guarantee convergence, albeit at a lower rate compared to the LTV model, see Fig. 10. This validates the use of LTI models in the present experimental study. For the experiments in this paper, ILC is based on an LTI model derived from an averaged frequency response measurement, see Fig. 11 for the Bode diagram of the model $\boldsymbol{P}$. For feedback, the feedback controller in Fig. 7 is used.

6.2. Experiment design

Contrary to the simulation case study in Section 5, where ILC was applied to a multi-input, single output system, in the experiments ILC is applied to a multi-input, multi-output system. During printing the gantry position is typically fixed while the carriage with the print heads moves over the gantry. In between the printing, the gantry performs a stepping motion in x-direction in

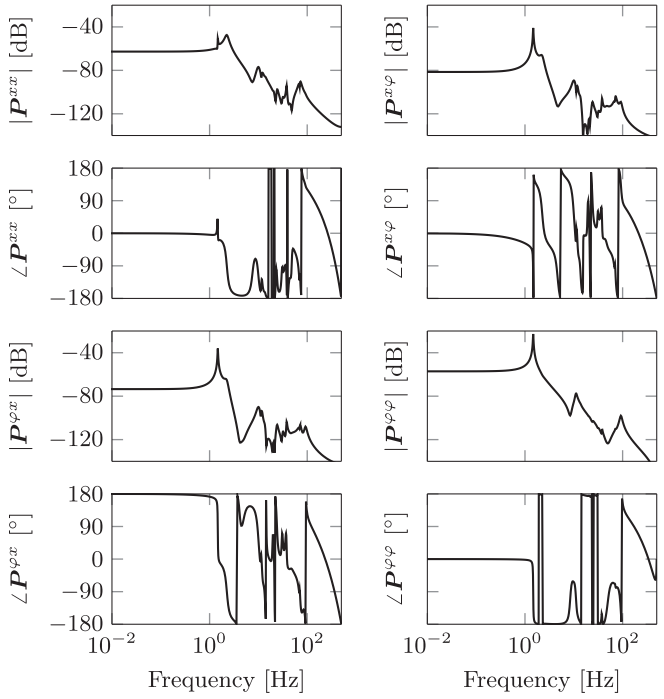


Fig. 11. Bode diagram of the system $\boldsymbol{P}$ for the 2 × 2 Arizona gantry based on an averaged frequency response measurement.

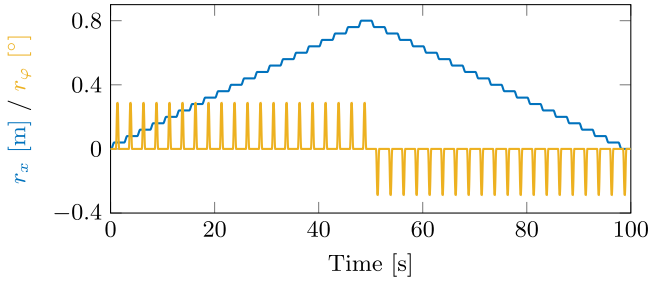


Fig. 12. The gantry performs a stepping movement in x direction (r_x), while small rotations in φ (r_φ) can be used for correcting misalignments. The task length is $N = 100,000$.

order to cover the next part of the medium. Without controlling the carriage, this results in the reference trajectories as shown in Fig. 12, with task length $N = 100,000$. The small rotation in φ during printing can be used for correcting misalignments.

The performance weights in Definition 5 are selected as

$$w_e = \begin{bmatrix} 10^5 & 0 \\ 0 & 5\times10^5 \end{bmatrix}, w_f = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 5\times10^{-5} \end{bmatrix}, w_{\Delta f} = 0,$$

for all k. The choice $w_{\Delta f} = 0$ results in fast convergence of the ILC update, whereas the combination of w_e and w_f ensures minimization of the error, with minimal restriction of the feedforward signal. Note that since an LTI model is used on a position-dependent system, additional robustness ($w_f > 0$) is used to enhance robust convergence properties. As shown in the previous section, this will degrade the performance in terms of the error norm.

6.3. Results

The performance criterion when applying resource-efficient ILC is shown in Fig. 13 for ten trials. Two important aspects are to be noted. First, the decrease in $\mathcal{J}$ indicates convergence of the ILC algorithm, which is enforced by selecting w_f sufficiently high. Second, despite $w_{\Delta f} = 0$, several iteration steps are required to converge to a steady state value due to model mismatches since an LTI model is used.

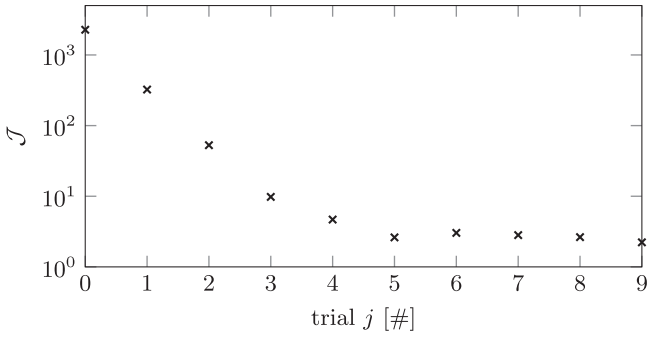
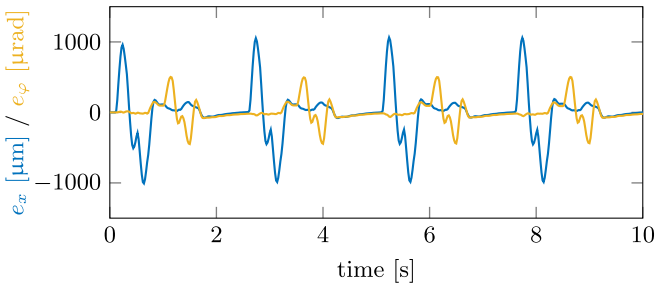
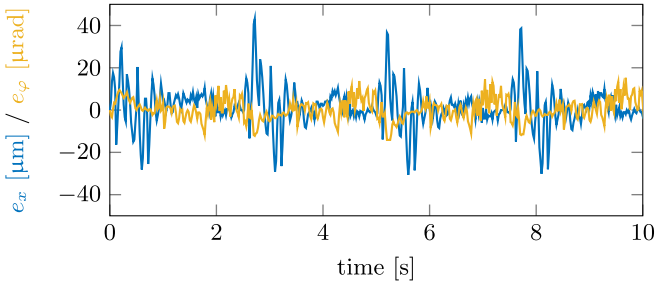


Fig. 13. The performance criterion $\mathcal{J}$ decreases significantly (more than a factor 1000) over the trials indicating convergence and high performance.



(a) Without ILC; feedback only ($j = 0$).



(b) After ten ILC trials ($j = 9$).

Fig. 14. After several ILC trials both the error signal in x direction (e_x) and φ-direction (e_φ) are significantly reduced (note the scales). Only the first ten seconds is shown.

The time domain errors are shown in Fig. 14. In the first trial, $j = 0$, no feedforward is applied (i.e., $f_j[k] = 0,\ \forall k$) yielding $\mathcal{J} = 2272$, $\|e_x\|_\infty = 1122\,\mu\text{m}$, and $\|e_\varphi\|_\infty = 506\,\mu\text{rad}$. After several trials the performance criterion is decreased by a factor 1000 to $\mathcal{J} = 2.2$ at trial $j = 9$, with $\|e_x\|_\infty = 48\,\mu\text{m}$, and $\|e_\varphi\|_\infty = 24\,\mu\text{rad}$.

The results show a significant performance enhancement for the position-dependent printer system, even with an LTI model. The performance may be further increased, where the parameter w_f can be used to tune robustness. Either this has to be chosen at a reasonably high value to guarantee robustness for position-dependent dynamics, or an LTV model of the printer has to be made. The latter is presently under investigation. In addition, a $w_{\Delta f}$ weighting may be introduced to reduce trial-varying disturbances. This is not done in the present research as the focus is on the computation load rather than performance, but can be further optimized.

Importantly, the results show that resource-efficient ILC is practical for large tasks (here $N = 100,000$). For such large tasks, lifted

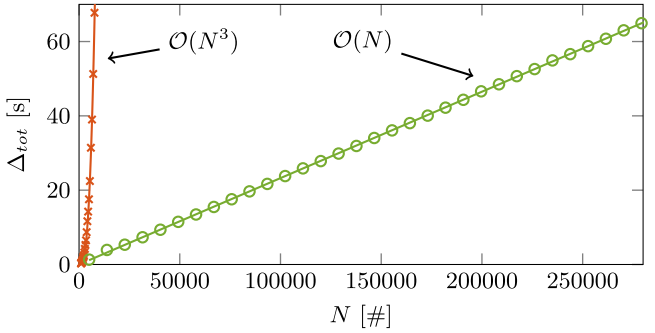


Fig. 15. The computation time $\Delta_{tot} = \Delta_{init} + \Delta_{trial}$ for lifted ILC (×) grows as $\mathcal{O}(N^3)$ (see the fit ——), whereas for resource-efficient ILC (○) it grows as $\mathcal{O}(N)$ (see the fit ——).

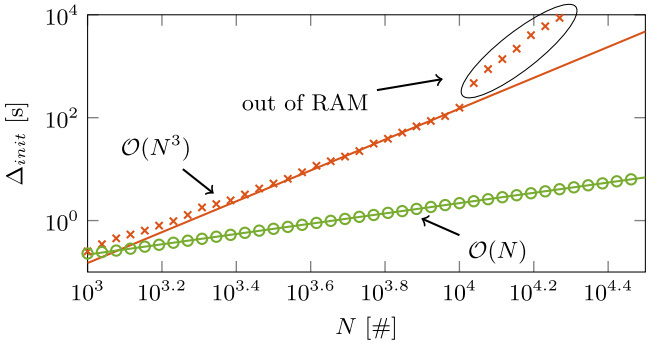


Fig. 16. The initialization time Δ_{init} for lifted ILC (×) evolves as $\mathcal{O}(N^3)$ as shown by the red curve (——). For resource-efficient ILC (○) it evolves as $\mathcal{O}(N)$ as shown by the green curve (——). For large N, there is insufficient RAM available for the initialization of lifted ILC resulting in large computation times.

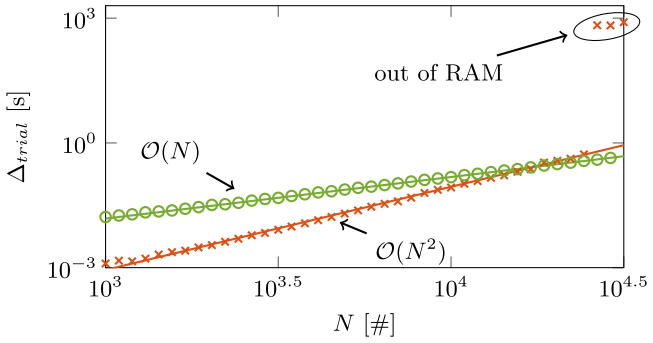


Fig. 17. The trial update time Δ_{trial} for lifted ILC (×) evolves as $\mathcal{O}(N^2)$ as shown by the red curve (——). For resource-efficient ILC (○) it evolves as $\mathcal{O}(N)$ as shown by the green curve (——).

ILC is impractical as is shown in the next section, since it would involve matrices of dimensions $200,000 \times 200,000$ ($n_i = n_o = 2$).

7. Computational requirements

In this section the computational load of lifted ILC and resource-efficient ILC are compared, constituting contribution V. The total computational load Δ_{tot} is split into two parts: $\Delta_{tot} = \Delta_{init} + \Delta_{trial}\, n_{trial}$, where Δ_{init} is the initialization of the algorithm (i.e., all calculations that can be computed a priori off-line), Δ_{trial} the on-line update (i.e., all calculations that need to be executed each trial), and n_{trial} the number of trials.

7.1. Analysis of computational complexity

For lifted ILC (Theorem 2), the initialization is given by (5) ($\underline{Q}$ and $\underline{L}$ are trial-invariant) and the trial update by (4). The initialization (5) is dominated by matrix multiplication and inversion, hence $\Delta_{init}^{lif} \sim \mathcal{O}(N^3)$, when using Schoolbook matrix multiplication and Gauss-Jordan elimination, respectively, see also [49]. The trial update (4) is dominated by matrix-vector multiplication, hence $\Delta_{trial}^{lif} \sim \mathcal{O}(N^2)$, when using Schoolbook matrix multiplication.

For resource-efficient ILC (Theorem 6), the initialization is given by step 1) in Algorithm 8 and the trial update by steps 2) and 3) in Algorithm 8. Note that the state-space matrices of (8) and (10) are trial-invariant and can hence be determined off-line during initialization. The dimensions in all steps are in the order of n_x. Hence, for $n_x \ll N$, $\Delta_{init}^{low} \sim \mathcal{O}(N)$ and $\Delta_{trial}^{low} \sim \mathcal{O}(N)$.

The above derivation is experimentally confirmed in the next section.

7.2. Comparison of computational cost

In this section, the analysis of the previous section is supported by numerical simulations. For the experiment, see Section 6, the initialization and trial update time of both approaches were measured on the full signals ($N = 100,000$) as well as on parts of it, i.e., for smaller N. Results for $\Delta_{tot} = \Delta_{init} + \Delta_{trial}$, i.e., $n_{trial} = 1$, are depicted in Fig. 15. As the analysis in the previous section indicates, the computation time of lifted ILC for large N is dominated by Δ_{init}^{lif} such that $\Delta_{tot}^{lif} \sim \mathcal{O}(N^3)$, see also the fit $\Delta_{tot}^{lif} = c^{lif} N^3$ (——)

Furthermore, the analysis indicates that $\Delta_{tot}^{low} \sim \mathcal{O}(N)$, as is confirmed by Fig. 15, see also the fit $\Delta_{tot}^{low} = c^{low} N$ (——). Hence, especially for large N, the resource-efficient ILC approach is computationally significantly faster than the lifted ILC approach. For comparison, in one hour of calculation time, an experiment with a single trial of length $N \approx 29000$ can be calculated with lifted ILC, and of length $N \approx 15 \cdot 10^6$ with resource-efficient ILC, which is over 530 times as large.

The computation time for the initialization and trial update step are displayed separately in Figs. 16 and 17, respectively. The results confirm the analysis of the previous section with respect to the dependence on N, see the fitted lines. Note that $\mathcal{O}(N^n)$ corresponds to a slope n on the double logarithmic scale. For large N, there is insufficient random-access memory (RAM) available for lifted ILC resulting in large computation times Δ_{init}^{lif}, Δ_{trial}^{lif}.

For the full experiment task length $N = 100,000$, $\Delta_{tot}^{low} = 23.3$ s whereas Δ_{tot}^{lif} is estimated at 40 hours, under the assumption of sufficient RAM.

8. Conclusions

In this paper, a unified approach to resource-efficient ILC techniques for LTI/LTV systems and optimal and general frequency domain designs is developed. In particular, first it is shown that using the lifted framework, an analytic expression for the optimal feedforward signal for generic norm-based performance criteria can be derived by solving a set of linear equations. However, the actual implementation is troublesome for large tasks since the computation load increases as $\mathcal{O}(N^3)$, with N the task length. In this paper, an alternative approach based on optimal control theory is presented that yields the same command signal, but at significantly lower computational cost, namely $\mathcal{O}(N)$, for both LTI and LTV systems.

A further analysis of this solution reveals that it is very similar, both in terms of computational techniques as well as the underlying theoretical developments, to common stable inversion techniques. The connections are explicitly established and analyzed,

leading to a unified solution for many ILC approaches, both lifted and classical frequency domain based, for both LTI and LTV systems.

Practical use is demonstrated by successfully applying resource-efficient ILC on an industrial flatbed printer. Simulation results on a position-dependent model reveal that LTV techniques can be very beneficial when applying ILC on position-dependent systems. Since the required first principles model is not sufficiently accurate for ILC design, an LTI model of the experimental system is used. The proposed algorithm, which is $\mathcal{O}(N)$, can be successfully implemented on a large task (here, $N = 100,000$, with two inputs and two outputs), for which traditional lifted norm-optimal ILC breaks down and is thus impractical to implement.

Ongoing work focuses on further development of feedforward and ILC for position-varying systems, as occurring in, e.g., next-generation motion systems [43]. Indeed, LTV models for these type of systems enable high performance, whereas LTI models require additional robustness at the cost of performance as also shown in experiments. These results motivate the ongoing research to development of new identification techniques for position-dependent systems, see, e.g., [50] for important steps, and development of ILC techniques compatible with these models. Also, next to the finite time results, at present $\mathcal{H}_2/\mathcal{H}_\infty$ optimal finite preview, infinite time results are being developed, see, e.g., [18] for preliminary results in this direction.

Acknowledgements

The authors would like to acknowledge fruitful discussions with and the contributions of Lennart Blanken, Frank Boeren, Okko Bosgra, Harm van Deursen, Janno Lunenburg, Sander Verhoeven, Robbert Voorhoeve, Jeroen van de Wijdeven, Jeroen Willems, and Bart van Willigen that have significantly contributed to the results presented in this paper.

Appendix A. Proof of Lemma 4

In Fig. 1, $\boldsymbol{J}$ is the transfer $f \to y$. Let x_P and x_C denote the state of $\boldsymbol{P}$ and $\boldsymbol{C}$, respectively, then using (6)

$$\begin{aligned} y[k] &= C_P x_P[k] + D_P u[k] + D_P f[k], \\ u[k] &= C_C x_C[k] - D_C y[k], \end{aligned}$$

which can be combined to

$$\begin{aligned} y[k] = {} & (I_{n_o} + D_P D_C)^{-1} \\ & \times (C_P x_P[k] + D_P C_C x_C[k] + D_P f[k]), \\ u[k] = {} & (I_{n_i} + D_C D_P)^{-1} \\ & \times (-D_C C_P x_P[k] + C_C x_C[k] - D_C D_P f[k]). \end{aligned} \tag{A.1}$$

Substitution of these relations into the state equations, rewriting and using the relation $I-(I+X)^{-1}X=(I+X)^{-1}$ yields

$$\begin{aligned} x_P[k+1] &= A_P x_P[k] + B_P u[k] + B_P f[k] \\ &= (A_P - B_P(I_{n_i} + D_C D_P C_P)^{-1} D_C) x_P[k] \\ &\quad + B_P (I_{n_i} + D_C D_P)^{-1} (C_C x_C[k] + f[k]), \\ x_C[k+1] &= A_C x_C[k] - B_C y[k] \\ &= (A_C - B_C (I_{n_o} + D_P D_C)^{-1} D_P C_C) x_C[k] \\ &\quad - B_C (I_{n_o} + D_P D_C)^{-1} (C_P x_P[k] + D_P f[k]). \end{aligned}$$

Combining the above state equations and output equation (A.1), and introducing state $x[k] = \begin{bmatrix} x_P[k] \\ x_C[k] \end{bmatrix}$ yields the state-space realization of $\boldsymbol{J}$.

Appendix B. Proof of Theorem 6

Problem setup

The system dynamics are given by

$$\begin{aligned} x_{j+1}[k+1] &= A x_{j+1}[k] + B f_{j+1}[k], \\ y_{j+1}[k] &= C x_{j+1}[k] + D f_{j+1}[k], \end{aligned}$$

with initial state $x_{j+1}[0] = x_0$ and (A, B, C, D) a state-space representation of the process sensitivity $\boldsymbol{J}$. Define

$$\begin{aligned} \Delta x_{j+1}[k] &:= x_{j+1}[k] - x_j[k], \\ \Delta f_{j+1}[k] &:= f_{j+1}[k] - f_j[k], \end{aligned}$$

then

$$\begin{aligned} \Delta x_{j+1}[k+1] &= A \Delta x_{j+1}[k] + B \Delta f_{j+1}[k], \\ \Delta y_{j+1}[k] &= C \Delta x_{j+1}[k] + D \Delta f_{j+1}[k], \end{aligned}$$

with $\Delta x_{j+1}[0] = 0_{n_x \times 1}$. Since $\tilde{r}$ is trial-invariant,

$$\begin{aligned} e_{j+1}[k] &= \tilde{r}[k] - y_{j+1}[k] \\ &= e_j[k] - C \Delta x_{j+1}[k] - D \Delta f_{j+1}[k]. \end{aligned}$$

In the remainder of the proof the subscript $j+1$ in general, and index $[k]$ for $w_e[k]$, $w_f[k]$, $w_{\Delta f}[k]$ are omitted for notational convenience. Note that this is not a restriction on the developed results.

The optimal input is given by

$$f^* = \arg\min_f \mathcal{J}(f) = f_j + \arg\min_{\Delta f} \mathcal{J}'(\Delta f),$$

where

$$\mathcal{J}'(\Delta f) := \tfrac{1}{2}\mathcal{J}(f_{j+1}) = \sum_{k=0}^{N-1} \mathcal{L}(\Delta x[k], \Delta f[k]),$$

with

$$\begin{aligned} \mathcal{L}(\Delta x[k], \Delta f[k]) = {} & \tfrac{1}{2}(e_j[k] - C\Delta x[k] - D\Delta f[k])^\top w_e \\ & \times (e_j[k] - C\Delta x[k] - D\Delta f[k]) \\ & + \tfrac{1}{2}(\Delta f[k] + f_j[k])^\top w_f (\Delta f[k] + f_j[k]) \\ & + \tfrac{1}{2}(\Delta f[k])^\top w_{\Delta f} (\Delta f[k]). \end{aligned}$$

The steps followed are along the lines of [51, sec. 5.5].

Hamiltonian, state, costate and open-loop optimal control

Let the Hamiltonian be defined as

$$\begin{aligned} &\mathcal{H}(\Delta x[k], \lambda[k+1], \Delta f[k]) \\ &\quad = \lambda^\top[k+1](A\Delta x[k] + B\Delta f[k]) + \mathcal{L}(\Delta x[k], \Delta f[k]). \end{aligned}$$

Let $\mathcal{H}^* = \mathcal{H}(\Delta x^*[k], \lambda^*[k+1], \Delta f^*[k])$, then the optimal state is given by

$$\begin{aligned} \Delta x^*[k+1] &= \frac{\partial \mathcal{H}^*}{\partial \lambda^*[k+1]} \\ &= A\Delta x^*[k] + B\Delta f^*[k], \end{aligned} \tag{B.1}$$

with

$$\Delta x^*[0] = 0_{n_x \times 1}, \tag{B.2}$$

and the optimal costate by

$$\begin{aligned} \lambda^*[k] = {} & \frac{\partial \mathcal{H}^*}{\partial \Delta x^*[k]} \\ & = A^\top \lambda^*[k+1] \\ & - C^\top w_e \big(e_j[k] - C\Delta x^*[k] - D\Delta f^*[k]\big), \end{aligned} \tag{B.3}$$

with

$$\lambda^*[N] = 0_{n_x \times 1}. \tag{B.4}$$

The optimal input satisfies

$$\frac{\partial \mathcal{H}^*}{\partial \Delta f^*[k]} = 0,$$

from which follows

$$\begin{aligned}\Delta f^*[k] = \gamma\,(&D^\top w_e e_j[k] - D^\top w_e C \Delta x^*[k] \\ &- w_f f_j[k] - B^\top \lambda^*[k+1]),\end{aligned} \tag{B.5}$$

with

$$\gamma = \left(D^\top w_e D + w_f + w_{\Delta f}\right)^{-1}.$$

With substitution of (B.5), relations (B.1) and (B.3), with boundary conditions (B.2) and (B.4), form the Hamiltonian system

$$\begin{aligned}&\begin{bmatrix}\Delta x^*[k+1] \\ \lambda^*[k]\end{bmatrix} \\ &\quad= \begin{bmatrix}A - B\gamma D^\top w_e C & -B\gamma B^\top \\ C^\top w_e\left(I - D\gamma D^\top w_e\right)C & A^\top - C^\top w_e D \gamma B^\top\end{bmatrix}\begin{bmatrix}\Delta x^*[k] \\ \lambda^*[k+1]\end{bmatrix} \\ &\qquad+\begin{bmatrix}-B\gamma w_f & B\gamma D^\top w_e \\ -C^\top w_e D\gamma w_f & C^\top w_e D\gamma D^\top w_e - C^\top w_e\end{bmatrix}\begin{bmatrix}f_j[k] & e_j[k]\end{bmatrix}, \\ &\Delta x^*[0] = 0_{n_x\times 1}, \\ &\lambda^*[N] = 0_{n_x\times 1}.\end{aligned} \tag{B.6}$$

Riccati and vector equations

Next, the co-state is eliminated from (B.6) using the sweep method [52] by applying the transformation

$$\lambda^*[k] = P[k]\Delta x^*[k] - g[k], \tag{B.7}$$

which yields

$$\begin{aligned}\Delta x^*[k+1] = &\left(I + B\gamma B^\top P[k+1]\right)^{-1} \\ &\times\left[\left(A - B\gamma D^\top w_e C\right)\Delta x^*[k] + B\gamma D^\top w_e e_j[k]\right. \\ &\left. - B\gamma w_f f_j[k] + B\gamma B^\top g[k+1]\right].\end{aligned} \tag{B.8}$$

Substituting (B.8) and (B.7) in the expression of $\lambda^*[k]$ in (B.6) and rewriting yields

$$\begin{aligned}&\left[P[k] - \left(A^\top - C^\top w_e D\gamma B^\top\right)P[k+1]\right. \\ &\quad\times\left(I + B\gamma B^\top P[k+1]\right)^{-1}\left(A - B\gamma D^\top w_e C\right) \\ &\quad\left. - C^\top w_e C + C^\top w_e D\gamma D^\top w_e C\right]\Delta x^*[k] \\ &= g[k] - \left(C^\top w_e - K_g[k]D^\top w_e - C^\top w_e D\gamma D^\top w_e\right)e_j[k] \\ &\quad-\left(K_g[k]w_f + C^\top w_e D\gamma w_f\right)f_j[k] \\ &\quad-\left(A^\top - C^\top w_e D\gamma B^\top - K_g[k]B^\top\right)g[k+1],\end{aligned} \tag{B.9}$$

where

$$K_g[k] = \left(A^\top - C^\top w_e D\gamma B^\top\right)P[k+1]\left(I + B\gamma B^\top P[k+1]\right)^{-1}B\gamma.$$

Relation (B.9) holds for all values $\Delta x^*[k]$, $\forall k$. Hence, the left-hand side of (B.9) should be zero for all k, leading to

$$\begin{aligned}P[k] = &\left(A^\top - C^\top w_e D\gamma B^\top\right)P[k+1]\left(I + B\gamma B^\top P[k+1]\right)^{-1} \\ &\times\left(A - B\gamma D^\top w_e C\right) + C^\top w_e C - C^\top w_e D\gamma D^\top w_e C,\end{aligned}$$

where the matrix identity

$$(\mathcal{A} + \mathcal{B}\mathcal{C}\mathcal{D})^{-1} = \mathcal{A}^{-1} - \mathcal{A}^{-1}\mathcal{B}\left(\mathcal{C}^{-1} + \mathcal{D}\mathcal{A}^{-1}\mathcal{B}\right)^{-1}\mathcal{D}\mathcal{A}^{-1},$$

leads to the matrix difference Riccati equation (11). Also, the right-hand side of (B.9) should vanish for all k, leading to the vector difference equation (10). Evaluating (B.7) at time instance $k = N$ yields

$$\lambda^*[N] = P[N]\Delta x^*[N] - g[N],$$

which holds for all $\Delta x^*[N]$ and given the boundary condition from (B.4), yields terminal conditions

$$\begin{aligned}&P[N] = 0_{n_x\times n_x}, \\ &g[N] = 0_{n_x\times 1}.\end{aligned}$$

Closed-loop optimal control

The closed-loop optimal control follows by substituting (B.7) at $k+1$ and (B.1) in (B.5), and solving for $\Delta f^*[k]$:

$$\Delta f^*[k] = -L[k]\Delta x^*[k] - L_f[k]f_j[k] + L_e[k]e_j[k] + L_g[k]g[k+1], \tag{B.10}$$

with $L[k]$, $L_f[k]$, $L_e[k]$, and $L_g[k]$ given by (9).

Combining (B.1) with (B.10) yields the state-space system (8) with state $\Delta x^*[k]$, inputs $f_j[k]$, $e_j[k]$, $g[k+1]$, and output $f^*[k] = f_j[k] + \Delta f^*[k]$ as given by (8).

Appendix C. Proof of Theorem 9

The proof is similar to that of [53] for continuous time systems. Since $x_s[k]$ and $x_u[k]$ are linearly coupled, the solution is found by applying the sweep method with

$$x_u[k] = P[k]x_s[k] + g[k], \tag{C.1}$$

which holds for all $x_s[k]$ and since $x_u[N] = 0$,

$$P[N] = 0, \quad g[N] = 0.$$

Evaluating (C.1) at $k+1$ and substituting the dynamics (13) yields

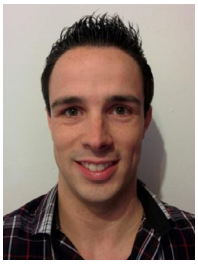
$$\begin{aligned}&\left(A_{us} + A_{uu}P[k] - P[k+1]A_{ss} - P[k+1]A_{su}P[k]\right)x_s[k] \\ &\quad= -A_{uu}g[k] - B_u u[k] + P[k+1]A_{su}g[k] + P[k+1]B_s u[k] + g[k+1],\end{aligned}$$

which holds for all $x_s[k]$ and therefore both sides should vanish. From the left-hand side follows (14) and from the right-hande side follows (15). $x_s[k]$ follows from substituting (C.1) into $x_s[k+1]$ in (13) and solving forward in time. $x_u[k]$ directly follows from (C.1).

References

[1] Bristow D, Tharayil M, Alleyne A. A survey of iterative learning control. Control Syst Mag 2006;26(3):96–114.
[2] Moore KL. Iterative learning control for deterministic systems. Springer-Verlag; 1993.
[3] Mishra S, Coaplen J, Tomizuka M. Precision positioning of wafer scanners: Segmented iterative learning control for nonrepetitive disturbances. Control Syst Mag 2007;27(4):20–5.
[4] de Roover D, Bosgra OH. Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system. Int J Control 2000;73(10):968–79.
[5] Steinbuch M, van de Molengraft R. Iterative learning control of industrial motion systems. 1st IFAC conference on mechatronic systems. Germany: Darmstadt; 2000. p. 967–72.
[6] Sutanto E, Alleyne A. A semi-continuous Roll-to-Roll (R2R) electrohydrodynamic jet printing system. Mechatronics 2015;31:243–54.
[7] Bolder J, Oomen T, Koekebakker S, Steinbuch M. Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer. Mechatronics 2014;24(8):944–53.
[8] Norrlöf M, Gunnarsson S. Time and frequency domain convergence properties in iterative learning control. Int J Control 2002;75(14):1114–26.
[9] Tousain R, van de Meché E, Bosgra O. Design strategy for Iterative Learning Control based on Optimal Control. In: Proceedings of the 40th conference on decision and control, vol. 5. Florida: Orlando; 2001. p. 4463–8.
[10] Gunnarsson S, Norrlöf M. On the design of ILC algorithms using optimization. Automatica 2001;37(12):2011–16.
[11] Barton KL, Bristow DA, Alleyne AG. A numerical method for determining monotonicity and convergence rate in iterative learning control. Int J Control 2010;83(2):219–26.
[12] Rice JK, van Wingerden J-W. Fast calculation of the 'ILC norm' in iterative learning control. Int J Control 2013;86(6):1186–90.
[13] Bolder J, Oomen T. Rational basis functions in iterative learning control - with experimental verification on a motion system. Trans Control Syst Technol 2015;23(2):722–9.

[14] van Zundert J, Bolder J, Oomen T. Optimality and flexibility in iterative learning control for varying tasks. Automatica 2016;67:295–302.
[15] Vandebril R, van Barel M, Mastronardi N. Matrix computations and semiseparable matrices volume 1: linear systems. Baltimore, Maryland: The Johns Hopkins University Press; 2008.
[16] Haber A, Fraanje R, Verhaegen M. Linear computational complexity robust ILC for lifted systems. Automatica 2012;48(6):1102–10.
[17] Tomizuka M. Zero phase error tracking algorithm for digital control. J Dyn Syst Measur Control 1987;109:65–8.
[18] Blanken L, Willems J, Koekebakker S, Oomen T. On design techniques for multivariable Iterative Learning Control. 7th IFAC symposium on mechatronic systems; 2016. [to appear].
[19] Butterworth J, Pao L, Abramovitch D. Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems. Mechatronics 2012;22(5):577–87.
[20] Tien S, Zou Q, Devasia S. Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed AFM operation. Trans Control Syst Technol 2005;13(6):921–31.
[21] Hunt L, Meyer G, Su R. Noncausal inverses for linear systems. Trans Autom Control 1996;41(4):608–11.
[22] Zou Q, Devasia S. Preview-Based Stable-Inversion for Output Tracking of Linear Systems. J Dyn Syst Measur Control 1999;121(4):625–30.
[23] Boeren F, Oomen T, Steinbuch M. Iterative motion feedforward tuning: A data–driven approach based on instrumental variable identification. Control Eng Prac 2015;37:11–19.
[24] Bolder J, van Zundert J, Koekebakker S, Oomen T. Enhancing flatbed printer accuracy and throughput - optimal rational feedforward via iterative learning control. Transactions on Industrial Electronics 2016. [submitted].
[25] Marro G, Prattichizzo D, Zattoni E. Convolution profiles for right inversion of multivariable non-minimum phase discrete-time systems. Automatica 2002;38(10):1695–703.
[26] Boeren F, Bruijnen D, van Dijk N, Oomen T. Joint input shaping and feedforward for point-to-point motion: automated tuning for an industrial nanopositioning system. Mechatronics 2014;24(6):572–81.
[27] Blanken L, Boeren F, Bruijnen D, Oomen T. Rational feedforward tuning: approaches, stable inversion, and experimental comparison. In: Proceedings of the 2016 American control conference. Massachusetts: Boston; 2016. p. 2629–34.
[28] Athans M, Falb PL. Optimal control. New York: McGraw-Hill Inc.; 1966.
[29] Amann N, Owens DH, Rogers E. Iterative learning control using optimal feedback and feedforward actions. Int J Control 1996;65(2):277–93.
[30] Oomen T, van de Wijdeven J, Bosgra OH. System identification and low-order optimal control of intersample behavior in ILC. Trans Autom Control 2011;56(11):2734–9.
[31] Dijkstra B, Bosgra O. Extrapolation of optimal lifted system ILC solution, with application to a waferstage. In: Proceedings of the 2002 American control conference, 4; 2002. p. 2595–600.
[32] van de Wijdeven JJM. Iterative learning control design for uncertain and time-windowed systems. Phd. Eindhoven University of Technology; 2008.
[33] Felici F, Oomen T. Enhancing current density profile control in tokamak experiments using iterative learning control. In: Proceedings of the 54th conference on decision and control. Japan: Osaka; 2015. p. 5370–7.
[34] Zou Q, Liu J. Multi-objective optimal trajectory design and tracking with non-periodic tracking-transition switching for non-minimum phase linear systems. Int J Control 2016:1–20.
[35] Brinkerhoff R, Devasia S. Output Tracking for Actuator Deficient/Redundant Systems: Multiple Piezoactuator Example. AIAA J Guidance, Control Dyn 2000;23(2):370–3.
[36] Sharma G, Agarwala A, Bhattacharya B. A fast parallel Gauss Jordan algorithm for matrix inversion using CUDA. Comput Struct 2013;128:31–7.
[37] Wallén J, Gunnarsson S, Norrlöf M. Analysis of boundary effects in iterative learning control. Int J Control 2013;86(3):410–15.
[38] Devasia S, Paden B. Stable inversion for nonlinear nonminimum-phase time–varying systems. Trans Autom Control 1998;43(2):283–8.
[39] Halanay A, Ionescu V. Time-varying discrete linear systems: input-output operators. Riccati Equations. Disturbance attenuation, 68. Birkhäuser Verlag; 1994.
[40] Devasia S. Output tracking with nonhyperbolic and near nonhyperbolic internal dynamics: Helicopter hover control. J Guidance, Control Dyn 1997;20(3):573–80.
[41] Jetto L, Orsini V, Romagnoli R. Accurate output tracking for nonminimum phase nonhyperbolic and near nonhyperbolic systems. Eur J Control 2014;20(6):292–300.
[42] Middleton RH, Chen JC, Freudenberg JS. Tracking sensitivity and achievable H_∞ performance in preview control. Automatica 2004;40(8):1297–306.
[43] Oomen T, van Herpen R, Quist S, van de Wal M, Bosgra O, Steinbuch M. Connecting system identification and robust control for next-generation motion control of a wafer stage. Trans Control Syst Technol 2014;22(1):102–18.
[44] van Herpen R, Oomen T, Kikken E, van de Wal M, Aangenent W, Steinbuch M. Exploiting additional actuators and sensors for nano-positioning robust motion control. Mechatronics 2014;24(6):619–31.
[45] Anderson BD, Moore JB. Optimal control: linear quadratic methods, vol. 1. Englewood Cliffs, New Jersey: Prentice Hall International, Inc.; 1989.
[46] Sogo T. On the equivalence between stable inversion for nonminimum phase systems and reciprocal transfer functions defined by the two-sided Laplace transform. Automatica 2010;46(1):122–6.
[47] Vinnicombe G. Uncertainty and feedback, $\mathcal{H}_\infty$ loop shaping and the ν-gap metric. London, Great Britain: Imperial College Press; 2001.
[48] van de Wijdeven J, Bosgra O. Using basis functions in iterative learning control: analysis and design theory. Int J Control 2010;83(4):661–75.
[49] Strassen V. Gaussian elimination is not optimal. Numerische Mathematik 1969;13(4):354–6.
[50] Groot Wassink M, van de Wal M, Scherer C, Bosgra O. LPV control for a wafer stage: beyond the theoretical solution. Control Eng Prac 2005;13(2):231–45.
[51] Naidu DS. Optimal control systems. CRC Press; 2003.
[52] Lewis FL, Syrmos VL. Optimal control. 2nd ed. New York: John Wiley & Sons, Inc.; 1995.
[53] Chen D. An iterative solution to stable inversion of nonminimum phase systems. In: Proceedings of the 1993 American control conference. California: San Francisco; 1993. p. 2960–4.



Jurgen van Zundert received the M.Sc. degree (with great appreciation) in Mechanical Engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands in 2014. He is currently pursuing the Ph.D. degree in the Control Systems Technology group within the department of Mechanical Engineering at TU/e. His research interests include feedforward motion control, multi-rate control, and iterative learning control.

Joost Bolder received the M.Sc. degree (cum laude) in Mechanical Engineering from the Eindhoven University of Technology, the Netherlands, in June 2011. In September 2015, he received the PhD degree from the Eindhoven University of Technology. Presently, he is a design engineer at ASML, Veldhoven, The Netherlands. His research interests include iterative learning control, image processing, and mechatronics.

Sjirk Koekebakker received the M.Sc. and Ph.D. degrees in mechanical engineering from Delft University of Technology, Delft, The Netherlands, in 1993 and 2001, respectively. Since 1999, he is a Mechatronic Designer with Oc Technologies B.V., Venlo, The Netherlands, which is part of the Canon group. As of 2009, he has also been a part-time Senior Researcher in the Control Systems Technology Group, Mechanical Engineering Department, Eindhoven University of Technology, Eindhoven, The Netherlands. His main research interests are iterative learning and repetitive control, precision motion control, and printing systems.

Tom Oomen received the M.Sc. degree (cum laude) and Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2005 and 2010, respectively. He was awarded the Corus Young Talent Graduation Award in 2005. He held visiting positions at KTH, Stockholm, Sweden, and at The University of Newcastle, Australia. Presently, he is an assistant professor with the Department of Mechanical Engineering at the Eindhoven University of Technology. He has served as an Associate Editor on the IEEE Conference Editorial Board and as a Guest Editor for IFAC Mechatronics. His research interests are in the field of system identification, robust control, and learning control, with applications in mechatronic systems.