

# Iterative Learning Control of the Electrostatic Microbridge Actuator

Błażej Cichy<sup>1</sup>, Krzysztof Gałkowski<sup>1</sup>, Andreas Rauh<sup>2</sup> and Harald Aschemann<sup>2</sup>

**Abstract**—In this paper, we consider the control of an electrostatic microbridge actuator with a distributed electrostatic force input and distributed displacement sensing. A bounded desired trajectory is achieved by using an Iterative Learning Control (ILC) method based on discrete linear repetitive processes with the use of Linear Matrix Inequalities (LMI). Space and time discretization is accomplished by using a finite difference approach based on the so-called Crank-Nicolson method.

**Index Terms**—Iterative Learning Control; distributed systems; Linear Matrix Inequalities.

## I. INTRODUCTION

Iterative Learning Control (ILC) has been developed especially to improve the performance of systems that operate in a repetitive manner, where the task is to follow some given trajectory in a specified finite time interval with high precision. The finite time interval is also known as a pass or as a trial in the literature. The novel principle behind ILC is to suitably use information from previous trials, often in combination with appropriate current trial information, to select the system input in the current trial in such a way that the controller's performance is improved from trial to trial. In particular, the aim is to improve the performance from trial to trial in the sense that the tracking error, i.e., the difference between the output during the whole trial and the specified reference trajectory, is sequentially reduced either to zero in the ideal case or to some suitably small value. Since the original work [1], the general area of ILC has been the subject of intense research work, where initial sources for the literature can be found in the survey paper [2].

ILC algorithms propagate information from trial to trial and along the trial, respectively, and hence can be treated as a 2D (or more generally  $nD$ ) system, or particularly as a repetitive process [3], which is characterized by a series of sweeps, termed passes, through a set of dynamics defined over a fixed finite duration that is known as the pass length. Recently, ILC algorithms, which are designed in the repetitive process framework, have been experimentally tested with results that clearly show how the trade-off between trial-to-trial error convergence, on the one hand, and the performance along a trial, on the other hand, can be treated in this setting [4].

This work is partially supported by the National Science Centre in Poland, grant No. 2011/01/B/ST7/00475, as well as by the German Research Foundation (Deutsche Forschungsgemeinschaft DFG), grant No. AS132/3-1.

<sup>1</sup>B. Cichy and K. Gałkowski are with the Institute of Control and Computation Engineering, University of Zielona Góra, ul. Ogrodowa 3b, 65-246 Zielona Góra, Poland {B.Cichy, K.Galkowski}@issi.uz.zgora.pl

<sup>2</sup>A. Rauh and H. Aschemann are with the Chair of Mechatronics, University of Rostock, Justus-von-Liebig-Weg 6, D-18059 Rostock, Germany {Andreas.Rauh, Harald.Aschemann}@uni-rostock.de

Currently, the vast majority of work reported on ILC considers lumped parameter systems. However, there is also some research dealing with the application of ILC to distributed parameter systems governed by Partial Differential Equations (PDEs), for example, [5], [6]. In terms of the development of ILC for PDEs, an obvious approach is to work directly with the defining equations, where, for example, [7] considers the design of P-type and D-type control laws for parabolic PDEs, such as the controlled heat equation, using semigroup theory. In [8], a number of other possible application areas is considered, such as the velocity and tension control for axially moving materials and electrostatic microbridge actuators, which are the subject of this paper. A common physical constraint in many distributed systems is given by the fact that only boundary control strategies can be employed. However, distributed sensors and/ or actuators also have a long history in numerous areas. More recent developments in supporting technologies have led to a renewed activity to consider distributed sensors and/ or actuators effectively in applications, see, for example, [9]. Finally, ILC has been applied to the control of heating systems in [10].

Natural connections exist between distributed parameter and multidimensional systems ( $nD$  systems), see e.g. [11]. These connections motivate the analysis in this paper, where the ILC method is applied to an electrostatic microbridge actuator, see [8]. In particular, the governing PDEs are first discretized by an implicit discretization scheme based on the Crank-Nicolson method [12] which has advantageous numerical stability properties. Moreover, this allows us to meet another physical requirement, namely, to use a control input which is homogeneously distributed in the space coordinate. Once the discretized model has been constructed, it can be written as a discrete linear repetitive process state-space model which is then used for the development of control laws that can be computed using Linear Matrix Inequalities (LMIs).

Throughout this paper,  $M \succ 0$  (resp.  $\prec 0$ ) denotes a real symmetric positive definite (resp. negative definite) matrix. The operator  $r(\cdot)$  denotes the spectral radius of a matrix. Furthermore, null and identity matrices with compatible dimensions are denoted by  $0$  and  $I$ , respectively. Finally, the following abbreviations are introduced

$$\text{tri}_\kappa(\beta, \gamma, \xi) = \begin{bmatrix} \gamma & \beta & & 0 \\ \xi & \gamma & \beta & \\ & \ddots & \ddots & \ddots \\ & & \xi & \gamma & \beta \\ 0 & & & \xi & \gamma \end{bmatrix}_{\kappa \times \kappa}, \Lambda_\kappa = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{\kappa \times 1}.$$

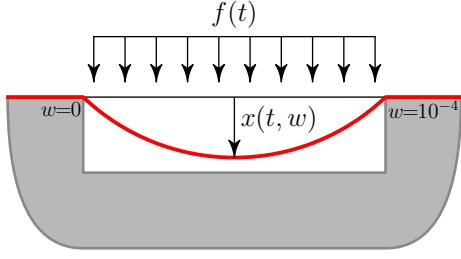


Fig. 1. Schematic diagram of the electrostatic microbridge actuator.

## II. TOWARDS A DISCRETE MODEL OF THE ELECTROSTATIC MICROBRIDGE ACTUATOR

Consider the electrostatic microbridge actuator which is schematically depicted in Fig. 1. The process behavior is described by the partial differential equation

$$\rho \frac{\partial^2 x(t, w)}{\partial t^2} + c \frac{\partial x(t, w)}{\partial t} - \mathcal{P} \frac{\partial^2 x(t, w)}{\partial w^2} = f(t), \quad (1)$$

where  $x(t, w)$  is the transversal displacement,  $\rho$  is the linear mass density,  $\mathcal{P}$  is the residual tension,  $L$  is the length,  $c$  is the squeeze film damping coefficient, and  $f(t)$  is a uniformly distributed electrostatic force, which represents the control input, and  $t$  and  $w$  are the time and space variables, respectively. The force actuator is assumed to be characterized by a first-order lag behavior according to

$$T_1 \frac{df(t)}{dt} + f(t) = u(t), \quad (2)$$

where  $u(t)$  represents the actual input signal. To fully characterize the problem, the physically motivated boundary and initial conditions are given as

$$x(t, 0) = x(t, L) = 0 \quad \forall t \geq 0 \quad \text{and} \quad f(0) = 0. \quad (3)$$

The design of a digital control law and its actual implementation require the construction of an appropriate approximation of the dynamics in terms of a set of difference equations. Hence, a finite difference approach is applied where signals are considered only at the discrete time and space instants  $t = lT$  and  $w = ph$ , with the time and space discretization periods  $T$  and  $h$ , respectively. Moreover,  $L = \alpha h$  is introduced, where  $\alpha$  is a fixed finite natural number. This is the basis for the formulation of a discrete dynamical system model, enabling efficient computations to be performed. If a direct discretization method is applied to spatio-temporal dynamics, there is the need to ensure numerical stability by selection of the sampling period(s). An advantageous alternative is to use an unconditionally numerically stable discretization scheme. This stability requirement is frequently met when using one of the implicit discretization methods, namely, the so-called Crank-Nicolson scheme [12]. For example, the Crank-Nicolson method leads to unconditionally stable discrete models for the heat transfer equation, see e.g. [10]. The equation (1) is of second order but resembles the heat transfer equation and hence one can suppose that its Crank-Nicolson discretization remains also unconditionally stable, which can be proved by using

the von Neumann stability analysis approach accompanied by the use of the bilinear transform, see [13].

Note also that an explicit discretization scheme does not apply at all in this case due to the fact that the actuator force  $f$  does not depend on the space variable and, therefore, is constant along this coordinate. A model generated by using an explicit discretization scheme, however, requires that the control is a function of both the time and space variables. In cases, when the control action only depends on time in the original continuous system, it becomes hard (or even impossible) for the explicit approximation to account for the property that the input does not depend on the space variable.

### A. Crank-Nicolson Discretization for PDEs

The equation (1) is discretized by using the Crank-Nicolson method which uses the following discrete approximations

$$\begin{aligned} x(t, w) &\approx \frac{x(l+2, p) + x(l+1, p) + x(l, p)}{3}, \\ \frac{\partial x(t, w)}{\partial t} &\approx \frac{x(l+2, p) - x(l, p)}{2T}, \\ \frac{\partial^2 x(t, w)}{\partial t^2} &\approx \frac{x(l+2, p) - 2x(l+1, p) + x(l, p)}{T^2}, \\ \frac{\partial^2 x(t, w)}{\partial w^2} &\approx \sum_{i=0}^2 \frac{x(l+i, p+1) - 2x(l+i, p) + x(l+i, p-1)}{3h^2}, \\ f(t) &\approx f(l+1), \end{aligned} \quad (4)$$

with  $x(l, p) = x(lT, ph)$  and  $f(l) = f(lT)$ . Furthermore, to discretize the input equation (2) for the force actuator, the zero-order hold method (corresponding to an implicit Euler discretization) is used, i.e.,

$$\frac{df(t)}{dt} \approx \frac{f(l+1) - f(l)}{T}, \quad (5)$$

which yields

$$u(l+1) = \left(1 + \frac{T_1}{T}\right) f(l+1) - \frac{T_1}{T} f(l). \quad (6)$$

Here, the time constant  $T_1$  is chosen to satisfy

$$T_1 \in [3T, 10T]. \quad (7)$$

Then, the approximation of (1) over  $l = 0, 1, \dots, N$ ,  $p = 1, 2, \dots, \alpha - 1$ , is given by

$$\begin{aligned} &A_1 x(l+2, p+1) + B_1 x(l+2, p) + C_1 x(l+2, p-1) \\ &+ A_2 x(l+1, p+1) + B_2 x(l+1, p) + C_2 x(l+1, p-1) \\ &+ A_3 x(l, p+1) + B_3 x(l, p) + C_3 x(l, p-1) = f(l+1), \end{aligned} \quad (8)$$

where

$$\begin{aligned} A_1 = A_2 = A_3 = C_1 = C_2 = C_3 &= -\frac{P}{3h^2}, \\ B_1 &= \frac{\rho}{T^2} + \frac{2P}{3h^2} + \frac{c}{2T}, \quad B_2 = \frac{2P}{3h^2} - \frac{2\rho}{T^2}, \\ B_3 &= \frac{\rho}{T^2} + \frac{2P}{3h^2} - \frac{c}{2T} \end{aligned}$$

hold with the associated boundary conditions

$$\begin{aligned} x(0, p) &= g_0(p), \quad x(1, p) = g_1(p), \quad 1 \leq p \leq \alpha - 1, \\ x(l, 0) &= 0, \quad x(l, \alpha) = 0, \quad l \geq 0, \end{aligned} \quad (9)$$

where  $g_0(p)$  and  $g_1(p)$  are vectors of dimension  $n \times 1$  with entries which are known functions of  $p$ .

*Remark 1:* In a first simplified version, it is assumed that the time constant  $T_1$  of (2) is exactly known during the control synthesis and hence  $f(l+1)$  can be considered as a virtual control input and that the actual input  $u(l)$  will be recalculated using (6). If this assumption does not hold, it is necessary to differentiate equation (1) with respect to time and to consider it together with (2) to obtain the full higher-order model with the input variable  $u(t)$ . Another possibility for the approximation is to shift back  $l$  to  $l-1$  in (8) and to employ (6) together with the original equation (8) to obtain a higher-order difference equation describing the process with the discrete input variable  $u(l)$ . Then, uncertainty in  $T_1$  can be addressed during control synthesis, which is the subject of ongoing work.

The discrete approximation of (8) is in the form of an implicit 2D equation that cannot be used directly to construct a discrete recursive model approximation to the process dynamics. Consequently, we introduce the stacking vectors

$$\mathcal{X}(l) = [x(l, 1)^T, x(l, 2)^T, \dots, x(l, \alpha - 1)^T]^T. \quad (10)$$

Then, zero boundary conditions in (9) yield that (8) can be rewritten as

$$\mathcal{A}_1 \mathcal{X}(l+2) = -\mathcal{A}_2 \mathcal{X}(l+1) - \mathcal{A}_3 \mathcal{X}(l) + \mathcal{B}f(l+1), \quad (11)$$

where the control matrix is given by  $\mathcal{B} = \Lambda_{\alpha-1}$  and  $x(l, 0) = 0$ ,  $x(l, \alpha) = 0$  and  $\mathcal{X}(l) = [g_\iota(1)^T, g_\iota(2)^T, \dots, g_\iota(\alpha-1)^T]^T$ ,  $\iota = 0, 1$ , are known boundary conditions and

$$\mathcal{A}_i = \text{tri}_{\alpha-1}(\mathcal{A}_i, B_i, C_i) \quad (12)$$

holds for  $i = 1, 2, 3$ . Note also that  $\mathcal{A}_i$ ,  $i = 1, 2, 3$ , are tridiagonal block Toeplitz matrices.

Now, it is clear that, due to lifting along the space variable, the dynamics along this dimension has been hidden and that there is no dependency on the variable  $p$ . In turn, the control action does not depend on the variable  $p$  either, which is a requirement for this case, as the function  $f(t)$  in (1) is independent of the space variable  $w$ . However, to obtain the state-space model from (11), it is necessary to invert the matrix  $\mathcal{A}_1$ , which is always possible as  $\mathcal{A}_1$  is tridiagonal or block tridiagonal (for the MIMO case) and hence is nonsingular. Therefore, (11) can be rewritten as

$$\mathcal{X}(l+2) = -\widehat{\mathcal{A}}_1 \mathcal{X}(l+1) - \widehat{\mathcal{A}}_2 \mathcal{X}(l) + \widehat{\mathcal{B}}f(l+1), \quad (13)$$

where  $\widehat{\mathcal{A}}_1 = \mathcal{A}_1^{-1} \mathcal{A}_2$ ,  $\widehat{\mathcal{A}}_2 = \mathcal{A}_1^{-1} \mathcal{A}_3$ , and  $\widehat{\mathcal{B}} = \mathcal{A}_1^{-1} \mathcal{B}$  hold. This is however not in the state-space form yet as it is a second-order equation, but there are commonly known approaches to transform a second-order difference equations into the first-order form. Introduce hence

$$\mathbf{X}(l) = \begin{bmatrix} \mathcal{X}(l+1) \\ \mathcal{X}(l) \end{bmatrix}, \quad \mathbf{F}(l) = f(l+1) \quad (14)$$

which allows us to rewrite (13) in the form

$$\mathbf{X}(l+1) = \mathbf{A} \mathbf{X}(l) + \mathbf{B} \mathbf{F}(l), \quad (15)$$

where

$$\mathbf{A} = \begin{bmatrix} -\widehat{\mathcal{A}}_1 & -\widehat{\mathcal{A}}_2 \\ I & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \widehat{\mathcal{B}} \\ 0 \end{bmatrix} \quad (16)$$

which is clearly first-order, and hence in the state-space form.

### III. ILC PROBLEM FORMULATION

In ILC, a major objective is to achieve convergence of the trial-to-trial error. In [4], it has been shown how such an ILC design can be performed for linear time-invariant dynamics by writing the design problem in terms of a discrete linear repetitive process [3]. Note that there is a well-known equivalence (see Chapter 3 of [3]) between these processes and 2D discrete linear systems.

In this paper, the objective is to achieve the prescribed reference signal  $x(l, p) = x^*(l, p)$  in space and time over the finite rectangle  $l = 0, 1, \dots, N$ ,  $p = 1, 2, \dots, \alpha - 1$  represented in the stacked vector form as  $\mathcal{X}^*(l) = [x^*(l, 1), x^*(l, 2), \dots, x^*(l, \alpha-1)]^T$ . As we do not specify the whole vector  $\mathbf{X}(l)$  but only half on it, e.g.  $\mathcal{X}^*(l+1)$ , an output equation can be added to the model in the form

$$\mathbf{Y}(l) \hat{=} \mathcal{X}(l+1) = \mathbf{C} \mathbf{X}(l) \quad (17)$$

with

$$\mathbf{C} = [I \quad 0]. \quad (18)$$

For the description of the ILC scheme, a new positive integer variable  $k$  is defined to denote the trial-to-trial update. Then, (13) and (17) can be rewritten for the  $k$ -th trial as

$$\begin{aligned} \mathbf{X}(k, l+1) &= \mathbf{A} \mathbf{X}(k, l) + \mathbf{B} \mathbf{F}(k, l), \\ \mathbf{Y}(k, l) &= \mathbf{C} \mathbf{X}(k, l). \end{aligned} \quad (19)$$

Now, define the tracking error  $E(k, l)$  over  $0 \leq l \leq N$  as

$$E(k, l) \hat{=} \mathcal{X}^*(l+1) - \mathcal{X}(k, l+1) = \mathbf{Y}^*(l) - \mathbf{Y}(k, l). \quad (20)$$

It is straightforward to see that

$$E(k+1, l) - E(k, l) = -(\mathbf{Y}(k+1, l) - \mathbf{Y}(k, l)). \quad (21)$$

Introduce also the state and control input increments

$$\begin{aligned} \Theta(k+1, l+1) &= \mathbf{X}(k+1, l) - \mathbf{X}(k, l), \\ \Delta \mathbf{F}(k+1, l) &= \mathbf{F}(k+1, l) - \mathbf{F}(k, l). \end{aligned} \quad (22)$$

Then,

$$\Theta(k+1, l+1) = \mathbf{A} \Theta(k+1, l) + \mathbf{B} \Delta \mathbf{F}(k+1, l-1). \quad (23)$$

From (21), when taking into account (19) and (22), we obtain

$$\begin{aligned} E(k+1, l) &= E(k, l) - \mathbf{C} \mathbf{A} \Theta(k+1, l) \\ &\quad - \mathbf{C} \mathbf{B} \Delta \mathbf{F}(k+1, l-1). \end{aligned} \quad (24)$$

As noted previously,  $\mathbf{F}(k, l)$  is first considered as a virtual control input, while the actual input  $u(k, l)$  is recalculated

by (6). Consider next the application of a control law of the form

$$\Delta \mathbf{F}(k+1, l) = \mathbf{K}\Theta(k+1, l+1) + K_2 E(k, l+1) . \quad (25)$$

Applying this control law to (23) and (24) and introducing

$$\begin{aligned} \hat{A} &= \mathbf{A} + \mathbf{BK}, & \hat{B}_0 &= \mathbf{BK}_2 , \\ \hat{C} &= -\mathbf{C}(\mathbf{A} + \mathbf{BK}), & \hat{D}_0 &= I - \mathbf{CBK}_2 , \end{aligned} \quad (26)$$

yields the final ILC scheme in the form

$$\begin{aligned} \Theta(k+1, l+1) &= \hat{A}\Theta(k+1, l) + \hat{B}_0 E(k, l) , \\ E(k+1, l) &= \hat{C}\Theta(k+1, l) + \hat{D}_0 E(k, l) . \end{aligned} \quad (27)$$

This is in the form of a discrete linear repetitive process state-space model, presented in the next section, and hence the stability theory for this class of 2D systems can be used to design the control law in the same manner as, e.g. in [4] for a finite-dimensional discrete linear state-space model.

#### IV. LINEAR REPETITIVE PROCESSES

The full treatment of repetitive processes, including stability analysis and stabilization, together with the representation of ILC algorithms for lumped parameter systems can be found in [3]. Here, the essential issue for further analysis is stability, for which we revisit briefly its basic notions. The stability theory for linear repetitive processes with constant pass length consists of two distinct concepts. Asymptotic stability demands *bounded-input bounded-output (BIBO) stability* over the fixed finite pass length  $\alpha > 0$ . This notion, however, does not guarantee an appropriate process dynamics along the pass. To avoid this situation, a stronger notion, the so-called *stability along the pass* is introduced assuming arbitrary pass length  $0 < \alpha \leq \infty$  (cf. [3]).

Consider the case of discrete dynamics along the pass and let  $\alpha < \infty$  denote the pass length and  $k \geq 0$  the pass number or index. Such processes evolve over the subset of the positive quadrant in the 2D plane defined by  $\{(p, k) : 0 \leq p \leq \alpha - 1, k \geq 0\}$ , and the most basic discrete linear repetitive process state-space model [3] has the following form

$$\begin{aligned} x_{k+1}(p+1) &= Ax_{k+1}(p) + Bu_{k+1}(p) + B_0 y_k(p) , \\ y_{k+1}(p) &= Cx_{k+1}(p) + Du_{k+1}(p) + D_0 y_k(p) . \end{aligned} \quad (28)$$

Here on pass  $k$ ,  $x_k(p) \in \mathbb{R}^n$  is the state vector,  $y_k(p) \in \mathbb{R}^m$  is the pass profile vector, and  $u_k(p) \in \mathbb{R}^r$  is the vector of control inputs. To complete the process description, it is necessary to specify the boundary conditions, that is,

$$\begin{aligned} x_{k+1}(0) &= d_{k+1}, \quad k \geq 0 , \\ y_0(p) &= f(p), \quad 0 \leq p \leq \alpha - 1 , \end{aligned} \quad (29)$$

where the  $n \times 1$  vector  $d_{k+1}$  has known constant entries and  $f(p)$  is an  $m \times 1$  vector of known functions of  $p$ .

*Theorem 1:* [3] A discrete linear repetitive process described by (28) and (29) is stable along the pass if, and only if, i)  $r(D_0) < 1$ , ii)  $r(A) < 1$ , and iii) all eigenvalues of

$$G(z) = C(zI - A)^{-1}B_0 + D_0$$

have modulus strictly less than unity for all  $|z| = 1$ .

The first two conditions are simple and can be checked by direct application of standard tests for linear systems. However, the third one becomes very difficult and cannot be applied for a control law design to achieve a desired performance. An alternative approach for the control design, that is also used in the following, is described in [3]. It makes use of a Lyapunov function method in combination with an LMI control design technique.

#### V. ILC DESIGN

As the starting point, consider the candidate for a Lyapunov function

$$V(k, l) = V_1(k, l) + V_2(k, l)$$

with

$$V_1(k, l) = \Theta(k+1, l)^T \mathbf{P} \Theta(k+1, l)$$

and

$$V_2(k, l) = E(k, l)^T P_2 E(k, l) ,$$

where  $\mathbf{P} = \text{diag}(P_0, P_1)$ ,  $P_i \succ 0$ ,  $i = 0, 1, 2$ . The associated increment of  $V(k, l)$  is given by

$$\Delta V(k, l) = V_1(k, l+1) - V_1(k, l) + V_2(k+1, l) - V_2(k, l) .$$

The process is stable along the trial if  $\Delta V(k, l) < 0$  holds for all  $k$  and  $l$ , which is equivalent to the requirement that

$$\Phi^T P \Phi - P \prec 0 , \quad (30)$$

where

$$\Phi = \begin{bmatrix} \hat{A} & \hat{B}_0 \\ \hat{C} & \hat{D}_0 \end{bmatrix}, \quad P = \text{diag}(\mathbf{P}, P_2)$$

and (30) is the 2D Lyapunov matrix inequality, see [3].

It is possible to represent (30) in terms of an LMI condition and to obtain the following result for stability along the trial under control action together with explicit formulas for computing the matrices in the control law (25).

*Theorem 2:* An ILC scheme of the form (27) is stable along the trial over  $R = \{(l, p) : l = 0, 1, \dots, N, p = 1, 2, \dots, \alpha - 1\}$  for any choice of the positive integers  $N$  and  $\alpha > 1$  if there exist matrices  $\mathbf{X} = \text{diag}(X_0, X_1) \succ 0$ ,  $X_2 \succ 0$ ,  $\mathbf{R} = [R_0 \ R_1]$  and  $R_2$  such that the following LMI is feasible

$$\begin{bmatrix} -\mathbf{X} & 0 & * & * \\ 0 & -X_2 & * & * \\ \mathbf{AX} + \mathbf{BR} & \mathbf{BR}_2 & -\mathbf{X} & * \\ -\mathbf{CAX} - \mathbf{CBR} & X_2 - \mathbf{CBR}_2 & 0 & -X_2 \end{bmatrix} \prec 0 \quad (31)$$

with

$$X_i = I_{\alpha-1} \otimes x_i, \quad R_i = r_i \Lambda_{\alpha-1}^T, \quad i = 0, 1, 2 \quad (32)$$

and the scalar values  $x_i$  and  $r_i$ . Here, the sign  $*$  denotes a symmetric block entry, i.e., entries with the index  $(i, j)$  are identical to  $(j, i)$ .

If (31) holds, the control matrices  $\mathbf{K}$  and  $K_2$  can be computed by using

$$\mathbf{K} = \mathbf{R}\mathbf{X}^{-1}, \quad K_2 = R_2 X_2^{-1} , \quad (33)$$

where the matrix  $\mathbf{K}$  is partitioned according to

$$\mathbf{K} = [K_0 \quad K_1] = [R_0 X_0^{-1} \quad R_1 X_1^{-1}]$$

with  $K_i = k_i \Lambda_{\alpha-1}^T$  and the scalar values  $k_i$ ,  $i = 0, 1, 2$ .

*Proof:* By the 2D Lyapunov inequality, (27) is stable along the trial if there exists a matrix  $P = \text{diag}(\mathbf{P}, P_2) \succ 0$  such that

$$\Phi^T P \Phi - P \prec 0, \quad (34)$$

where

$$\Phi = \begin{bmatrix} \hat{A} & \hat{B}_0 \\ \hat{C} & \hat{D}_0 \end{bmatrix}. \quad (35)$$

Note here that to avoid further controller non-causality we need to limit our attention to the case of  $\mathbf{P} = \text{diag}(P_0, P_1)$ , where both  $P_0$  and  $P_1$  are equally dimensioned square matrices. An obvious application of the Schur complement formula to (34) yields

$$\begin{bmatrix} -\mathbf{P} & * & * & * \\ 0 & -P_2 & * & * \\ \hat{A} & \hat{B}_0 & -\mathbf{P}^{-1} & * \\ \hat{C} & \hat{D}_0 & 0 & -P_2^{-1} \end{bmatrix} \prec 0. \quad (36)$$

Now introduce

$$\mathbf{X} = \mathbf{P}^{-1}, \quad X_2 = P_2^{-1} \quad (37)$$

and pre- and post-multiply (36) by  $\text{diag}(\mathbf{X}, X_2, I, I)$  to obtain

$$\begin{bmatrix} -\mathbf{X} & * & * & * \\ 0 & -X_2 & * & * \\ \hat{A}\mathbf{X} & \hat{B}_0 X_2 & -\mathbf{X} & * \\ \hat{C}\mathbf{X} & \hat{D}_0 X_2 & 0 & -X_2 \end{bmatrix} \prec 0. \quad (38)$$

Substituting the expression in (26) for the corresponding matrices leads to

$$\begin{bmatrix} -\mathbf{X} & * & * & * \\ 0 & -X_2 & * & * \\ \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{K}\mathbf{X} & \mathbf{B}\mathbf{K}_2 X_2 & -\mathbf{X} & * \\ -\mathbf{C}\mathbf{A}\mathbf{X} - \mathbf{C}\mathbf{B}\mathbf{K}\mathbf{X} & X_2 - \mathbf{C}\mathbf{B}\mathbf{K}_2 X_2 & 0 & -X_2 \end{bmatrix} \prec 0. \quad (39)$$

Finally, the definitions

$$\mathbf{R} = \mathbf{K}\mathbf{X}, \quad R_2 = K_2 X_2, \quad (40)$$

lead to the LMI given in (31). Hence, the control matrices defined in (33) can be calculated from (40). This completes the proof.  $\blacksquare$

It turns out that solving the LMI (31) frequently leads to very slow convergence. Hence, it is suggested to design an LMI optimization procedure which removes this disadvantage. One of the possible solutions to this problem is shown further in the numerical example section.

The virtual control signal, i.e. the electrostatic force that is produced by the actuator can be computed according to

$$\begin{aligned} f(k, l+1) &= f(k-1, l+1) \\ &+ K_0(\mathcal{X}(k, l+1) - \mathcal{X}(k-1, l+1)) \\ &+ K_1(\mathcal{X}(k, l) - \mathcal{X}(k-1, l)) \\ &+ K_2(\mathcal{X}^*(l+2) - \mathcal{X}(k-1, l+2)), \end{aligned} \quad (41)$$

while the actual input signal results from (6) as

$$u(k, l+1) = \left(1 + \frac{T_1}{T}\right) f(k, l+1) - \frac{T_1}{T} f(k, l) \quad (42)$$

with  $f(k, 0) = 0$  and  $u(k, 0) = 0$ ,  $\forall k \geq 0$ .

## VI. NUMERICAL EXAMPLE

Consider the microbridge process (1) over  $0 \leq t \leq 10^{-3}$  (given in s) and  $0 \leq w \leq 10^{-4}$  (given in m), with  $\rho = 9.32 \times 10^{-9}$  (in kg/m),  $c = 7 \times 10^{-5}$  (in Ns/m),  $\mathcal{P} = 1 \times 10^{-7}$  (in N), and apply the Crank-Nicolson discretization scheme with  $T = 1 \times 10^{-5}$  (in s) and  $h = 1.25 \times 10^{-6}$  (in m), which represent feasible parameters since, as has been mentioned previously, the Crank-Nicolson discretization is unconditionally numerically stable for this case. Hence,  $\alpha = 80$  and  $N = 100$  are obtained and the coefficients in the model (8) are

$$A_i = C_i = -2.1333 \times 10^4, \quad i = 1, 2, 3,$$

$$B_1 = 4.2763 \times 10^4, \quad B_2 = 4.248 \times 10^4, \quad B_3 = 4.2756 \times 10^4$$

with the boundary conditions for the ILC scheme (19)

$$\mathbf{X}(0, l) = 0, \quad \mathbf{F}(0, l) = 0, \quad 0 \leq l \leq N,$$

$$\mathbf{X}(k, 0) = 0, \quad k \geq 1,$$

$$x(k, l, p) = 0, \quad p \in \{0, \alpha\}, \quad 0 \leq l \leq N, \quad k \geq 0.$$

The reference signal under investigation, which satisfies all boundary conditions, is depicted in Fig. 2.

Using only the Theorem 2 can lead to an unsatisfactory design and, in particular, the required control action can be excessive and convergence speed can be too small. To avoid such a situation, we can solve the following LMI optimization problem

$$\begin{aligned} &\text{maximize } r_0 + r_1 + r_2 \\ &\text{subject to the LMI (31)} \end{aligned} \quad (43)$$

with the additional constraint  $x_2 \leq r_2 \leq 3x_2$  obtained in a heuristic way, i.e., by trial and error. If (43) holds, the matrices of the stabilizing control law in (25) are given as in (33).

Solving this LMI optimization problem gives the following parameters of the control matrices  $K_i$ ,  $i = 0, 1, 2$  in (33)

$$k_0 = -2.0508, \quad k_1 = 1.6245, \quad k_2 = 2.3439.$$

The control signal  $u(k, l)$  resulting from (41) and (42) with  $T_1 = 6T$  is presented in Fig. 3.

Now, the controller's robustness against the actuator uncertainty is checked by simulations. For this reason, we assume  $T_1 = 6T$  as the nominal value and check the system behavior for  $T_1 \in [3T; 10T]$ . Hence, keeping the control action  $u(k, l)$  invariant, we recalculate the electrostatic force

$$f(k, l+1) = \left(\frac{T_1}{T+T_1}\right) f(k, l) + \left(\frac{T}{T+T_1}\right) u(k, l+1), \quad (44)$$

which comes from (42) for  $T_1 \in [3T; 10T]$ . Moreover, we assume that  $f(k, 0) = 0$  holds for all  $k \geq 0$ .

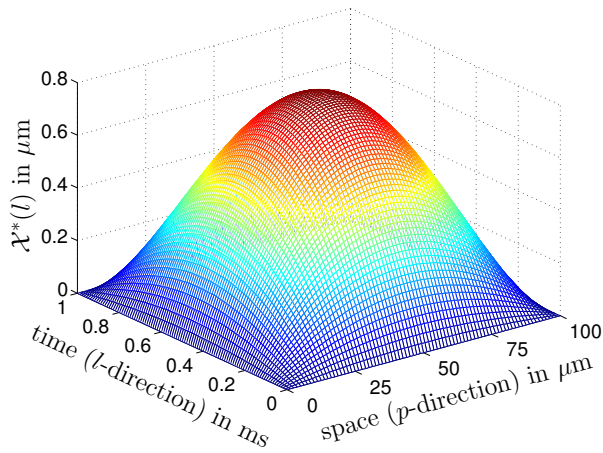


Fig. 2. Reference signal  $\mathcal{X}^*(l)$ .

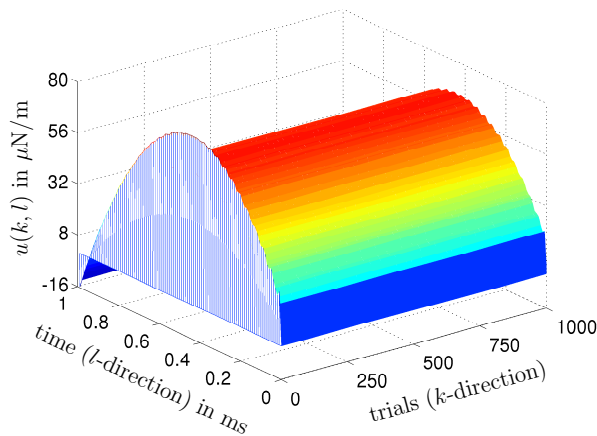


Fig. 3. Control input for  $T_1 = 6T$  across trials and time.

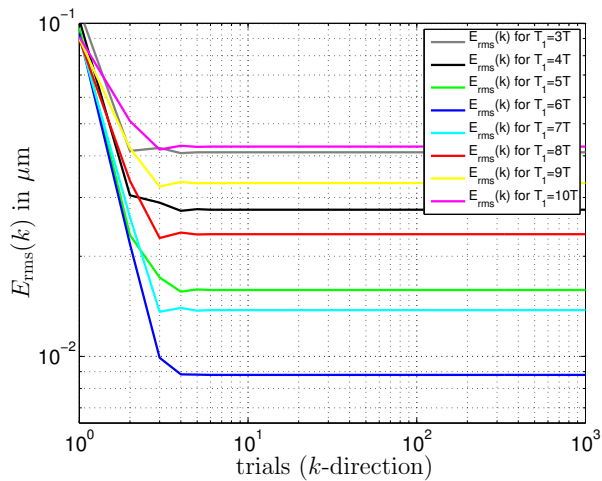


Fig. 4. Error  $E_{rms}(k)$  over trials and a ratio  $\frac{T_1}{T}$ .

An averaged measure of the error signal, i.e., a Root Mean

Square (RMS) for these cases is presented in Fig. 4, which shows that changing the ratio  $\frac{T_1}{T}$  in the prescribed range does not destabilize the system, however, slightly increases the steady state error.

## VII. CONCLUSIONS

In this paper, an ILC method based on using repetitive processes and LMIs has been applied to the model of an electrostatic microbridge actuator. Due to the need for distributed control signals, which are constant along the space variable, the Crank-Nicolson discretization has been applied, which further ensures unconditional numerical stability for this case. Further work aims at increasing the controller's robustness with respect to parameter uncertainty.

## REFERENCES

- [1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.
- [2] H.-S. Ahn, Y.-Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.
- [3] E. Rogers, K. Gałkowski, and D. H. Owens, *Control systems theory and applications for linear repetitive processes*, ser. Lecture Notes in Control and Information Sciences. Berlin Heidelberg: Springer-Verlag, 2007, vol. 349.
- [4] Ł. Hladowski, K. Gałkowski, Z. Cai, E. Rogers, C. T. Freeman, and P. L. Lewin, "Experimentally supported 2D systems based iterative learning control law design for error convergence and performance," *Control Engineering Practice*, vol. 18, no. 4, pp. 339–348, 2010.
- [5] J. Choi, B. J. Seo, and K. S. Lee, "Constrained digital regulation of hyperbolic PDE systems: A learning control approach," *Korean Journal of Chemical Engineering*, vol. 18, no. 5, pp. 606–611, 2001.
- [6] K. L. Moore and Y.-Q. Chen, "Iterative learning approach to a diffusion control problem in an irrigation application," in *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 2006, pp. 1329–1334.
- [7] Xu Chao, R. Arastoo, and E. Schuster, "On iterative learning control of parabolic distributed parameter systems," in *Proceedings of the 17th Mediterranean Conference on Control and Automation*, 2009, pp. 510–515.
- [8] H. Zhao, "Passive, iterative, and repetitive control for flexible distributed parameter systems," Ph.D. dissertation, Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, Pennsylvania, 2005.
- [9] N. D. Maxwell and S. F. Asokanathan, "Modal characteristics of a flexible beam with multiple distributed actuators," *Journal of Sound and Vibration*, vol. 269, no. 1-2, pp. 19–31, 2004.
- [10] B. Cichy, K. Gałkowski, and E. Rogers, "Iterative learning control for spatio-temporal dynamics using Crank-Nicolson discretization," *Multidimensional Systems and Signal Processing*, vol. 23, no. 1, pp. 185–208, 2012.
- [11] R. Rabenstein and L. Trautmann, "Towards a framework for continuous and discrete multidimensional systems," *International Journal of Applied Mathematics and Computer Science*, vol. 13, no. 1, pp. 73–86, 2003.
- [12] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 43, pp. 50–67, 1947.
- [13] R. Rabenstein and P. Steffen, "Stability analysis for implicit second order finite difference schemes," in *7th International Workshop on Multidimensional (nD) Systems (nDS)*, Poitiers, France, 2011, pp. 1–6.