Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Efficient Implementation of a Binary Iterative Learning Control

Florian Arnold * **Daniel Topalovic** * **Rudibert King** *

* *Technische Universität Berlin*
Department of Process Engineering,
Chair of Measurement and Control,
Strasse des 17. Juni 135, 10623 Berlin, Germany
(E-mail: f.arnold@tu-berlin.de, daniel.topalovic@tu-berlin.de,
rudibert.king@tu-berlin.de)

Abstract: Iterative learning control (ILC) is an adequate control approach to handle various types of cyclic control tasks. However, when in each iteration the calculation of the control trajectory requires the solution of a high dimensional constrained quadratic program, the algorithm is bound to be infeasible for real-time applications with very small cycle lengths in the order of milliseconds due to the prohibitively large computational cost.
In this contribution, an approach is presented to reduce the computational burden to solve an optimization-based iterative learning control that is restricted to a binary domain by orders of magnitude. The method is suitable for control trajectories that contain only few 1's, but a large number of 0's in each iteration for a specific class of problems, e.g., for cyclic firing synchronization of combustion tubes.
The presented setup is tested experimentally at an acoustic mock-up of an annular pulse detonation combustor to determine an appropriate fire synchronization. More specifically, it is used to adjust the firing pattern of multiple simulated combustion tubes in order to reduce pressure fluctuations measured downstream in an annular plenum, which is a prerequisite to apply such a new thermodynamically efficient combustion process in a real gas turbine.

Keywords: Learning control, optimal control, quadratic programming, binary control, discrete-time systems, multi-input/multi-output systems

1. INTRODUCTION

Norm optimal iterative learning control (ILC), as described by Amann et al. (1998), is a powerful variant of ILC to handle cyclic control problems. An overview of the wide range of different ILC approaches is given, e.g., by Ahn et al. (2007) and Bristow et al. (2006). If a linear model and unconstrained real-valued control inputs can be assumed for the process considered, a very efficient calculation of the control trajectory results. However, this is not true when constraints have to be respected, and even more so, when the actuation of the system is restricted to a binary domain, see Kochenberger et al. (2014). Now, the optimization problem can no longer be solved analytically, but a binary quadratic program (QP) has to be solved numerically in each iteration, see Arnold et al. (2019). For highly dynamic systems, with a high number of sampling instants and a small cycle duration, this is expensive from the computational point of view and often rules out the real-time application. Similar problems have been addressed in the context of model predictive control by Axehill et al. (2010) and Bürger et al. (2019).

This contribution presents, for a specific class of problems, an approach for handling such high dimensional binary optimization problems efficiently within an ILC. In the proposed method, the size of the optimization problem is reduced significantly by partially dropping out model information in each iteration. As it can be expected that the initial guess of the control trajectory of the ILC has a major impact on the performance of this approach, different initial conditions are investigated. For an application, an obvious choice would be to calculate the open-loop optimal control trajectory as an initial condition prior to the actual operation in an offline phase when restrictions concerning the speed of calculation do not exist. Additionally, this global optimal solution will be exploited here to validate the performance of the controller for varying initial control trajectories.

An experimental test of the presented method is performed at a cold mockup of a new combustion system, namely at an acoustic test rig. Here, loudspeakers mimic the effect of combustion tubes operated in an unsteady fashion and firing into a downstream plenum. For this, a pre-recorded signal of a certain length, mimicking a detonation, is send to the loudspeaker and played. The control input is a binary signal representing the switching-on times of the speakers. The control task for this multi-input/multi-output system is to synchronize the switching-on times so that the measured pressure signals at certain downstream positions in an annular gap are minimized. Such a mitigation will be needed when constant-volume combustion schemes, as a pulsed detonation combustion, will be

implemented in future gas turbines to increase efficiency. This example also exemplifies the class of applications the presented approach might be suitable for.

The organization of this contribution is as follows: Section 2 starts from the descriptions of a general optimal control problem and ILC to then discuss a formulation in the binary domain. Next, a transformation is introduced that will reduce the computational effort in each cycle of the ILC. By this, real-time applicability can be achieved. The acoustic test rig is then introduced in Section 3 together with the mathematical model of the process. Results from the test rig are shown in the ensuing section before a conclusion is drawn in Section 5.

2. OPTIMAL CONTROL

For the following section, it is assumed that the input-output behavior of one complete cycle j of the ILC can approximately be described by a linear model

$$\underline{Y}_j = \mathbf{\Phi}\underline{U}_j, \tag{1}$$

with model transition matrix $\mathbf{\Phi}$. Here, the supervector $\underline{Y}_j \in \mathbb{R}^{n_y}$ contains the outputs of a complete cycle, and $\underline{U}_j \in \mathbb{B}^{n_u}$ represents the corresponding supervector of inputs. The dimensions $n_y = p \cdot ny$ and $n_u = p \cdot nu$ depend on the number of sampling instants p, as well as the number of sensors ny and control inputs nu, respectively. Here, vectors are underlined. Upper and lower case letters distinguish between supervectors and ordinary vectors. Bold style symbols indicate matrices.

An implementation of an ILC based on such a model in real-time can be difficult when the cycle duration is short, but a large number of sampling instants p is needed to describe the dynamics well. To reduce the cost in every ILC iteration, the following sections will introduce a new method to decrease the dimension of the underlying optimization task. Since this approach is expected to require an initial control trajectory $\underline{U}_0$ close to the optimal trajectory $\underline{U}_{opt}$ for satisfactory results, an open-loop optimal control is calculated first, serving as such an initial guess. The optimal solution will be used as well as a benchmark to test the convergence of the approach.

The subsequent sections start with a brief summary of optimal binary control to set the scene before the general ILC setup is introduced. Finally, a transformation of the underlying quadratic program (QP) that reduces the dimension in each ILC iteration is presented as the main contribution of this paper.

2.1 Binary Optimal Control

An open-loop optimal control problem formulation is used to calculate a feasible actuation $\underline{U}^*$ that is optimal for one cycle with respect to a cost function $J(\underline{U})$. For an optimal tracking of the reference, given by the supervector $\underline{R}$, with respect to the weighted $||\cdot||_2$-norm, the cost function reads

$$J(\underline{U}) = \left(\underline{R} - \underline{Y}(\underline{U})\right)^T \mathbf{W}_E \left(\underline{R} - \underline{Y}(\underline{U})\right). \tag{2}$$

The weighting matrix $\mathbf{W}_E$ is assumed to be positive definite and symmetric. If the system output $\underline{Y}(\underline{U})$ can be calculated from a linear model like (1), the cost function results in a quadratic expression with respect to the control trajectory $\underline{U}$:

$$\begin{aligned} J(\underline{U}) &= \left(\underline{R} - \mathbf{\Phi}\underline{U}\right)^T \mathbf{W}_E \left(\underline{R} - \mathbf{\Phi}\underline{U}\right) \\ &= \underline{U}^T\mathbf{\Phi}^T\mathbf{W}_E\mathbf{\Phi}\underline{U} - 2\,\underline{R}^T\mathbf{W}_E\mathbf{\Phi}\underline{U} + \underline{R}^T\mathbf{W}_E\underline{R} \\ &= \underline{U}^T\mathbf{Q}\underline{U} + \underline{q}^T\underline{U} + c. \end{aligned} \tag{3}$$

The optimal control trajectory $\underline{U}^*$ can then be calculated with the binary QP

$$\underline{U}^* = \arg\min \underline{U}^T\mathbf{Q}\underline{U} + \underline{q}^T\underline{U} \tag{4}$$

subject to

$$\underline{U} \in S \cap \mathbb{B}^{n_u}. \tag{5}$$

Besides the binary constraint $\mathbb{B}^{n_u}$, more constraints are included in S to allow for the addition of further requirements. As an example, for the experimental system considered below, the control trajectory is constrained by the assumption that each loudspeaker can only be actuated once within one cycle.

2.2 Binary Iterative Learning Control

As the model (1) is only an approximation of the actual system and disturbances may occur during the operation, an additional closed-loop control is required to guarantee a sufficient control performance. For cyclic processes, norm optimal iterative learning control can be exploited to improve the control trajectory from cycle to cycle in an optimized fashion.

In cycle $j+1$, the control error $\underline{E}_{j+1} = \underline{R} - \underline{Y}_{j+1}$ can be calculated from the cycle-invariant reference $\underline{R}$ and the measured system output over one cycle $\underline{Y}_{j+1}$. The cost for cycle $j+1$ then reads

$$J_{j+1}(\underline{U}_{j+1}) = \underline{E}_{j+1}^T\,\mathbf{W}_E\,\underline{E}_{j+1} + \Delta\underline{U}_{j+1}^T\mathbf{W}_{\Delta U}\Delta\underline{U}_{j+1}. \tag{6}$$

In the general norm optimal ILC formulation the change of the control trajectory $\underline{\Delta U}_{j+1} = \underline{U}_{j+1} - \underline{U}_j$ is penalized with the positive definite and symmetric weight $\mathbf{W}_{\Delta U}$ to avoid overshoots that might lead to oscillations. As the error $\underline{E}_{j+1}$ is unknown at the end of cycle j when the control trajectory $\underline{U}_{j+1}$ needs to be calculated, the process model (1) is used. Moreover, besides the reference trajectory $\underline{R}$, disturbances $\underline{D}$ are assumed to be cycle-invariant as well in what follows.

The control error in cycle j is given by

$$\underline{E}_j = \underline{R} - \underline{Y}_j. \tag{7}$$

It is now assumed that the measured system output $\underline{Y}_j$ can be approximated with the model Eq. (1) and a cycle-invariant disturbance $\underline{D}$:

$$\underline{E}_j = \underline{R} - \left(\underline{Y}_j + \underline{D}\right) = \underline{R} - \left(\mathbf{\Phi}\underline{U}_j + \underline{D}\right). \tag{8}$$

Reusing this equation for cycle $j+1$ likewise leads to the prediction of the control error in cycle $j+1$ as a function of $\underline{U}_{j+1}$ and known quantities at the end of cycle j

$$\begin{aligned} \underline{R} - \underline{D} = \underline{E}_j + \mathbf{\Phi}\underline{U}_j &= \underline{E}_{j+1} + \mathbf{\Phi}\underline{U}_{j+1} \\ \underline{E}_{j+1} &= \underline{E}_j + \mathbf{\Phi}\underline{U}_j - \mathbf{\Phi}\underline{U}_{j+1}. \end{aligned} \tag{9}$$

Hence, the cost function in (6) can be written as a quadratic expression depending only on $\underline{U}_{j+1}$:

$$J_{j+1} = \underline{U}_{j+1}^T \mathbf{Q} \underline{U}_{j+1} + \underline{q}^T \underline{U}_{j+1} + c, \tag{10}$$

with

$$\mathbf{Q} = \mathbf{\Phi}^T \mathbf{W}_E \mathbf{\Phi} + \mathbf{W}_{\Delta U} \tag{11}$$

$$\underline{q}^T = -2 \left(\underline{U}_j^T \left(\mathbf{\Phi}^T \mathbf{W}_E \mathbf{\Phi} + \mathbf{W}_{\Delta U} \right) + \underline{E}_j^T \mathbf{W}_E \mathbf{\Phi} \right) \tag{12}$$

$$c = \underline{E}_j^T \mathbf{W}_E \left(\underline{E}_j + 2 \mathbf{\Phi} \underline{U}_j \right) + \underline{U}_j^T \left(\mathbf{\Phi}^T \mathbf{W}_E \mathbf{\Phi} + \mathbf{W}_{\Delta U} \right) \underline{U}_j. \tag{13}$$

For a real-valued, unconstrained control domain, the optimal control trajectory $\underline{U}_{j+1}^*$ could be easily calculated by taking the derivative of J_{j+1} with respect to $\underline{U}_{j+1}$ and equating it to zero. However, as the control domain considered in this contribution is restricted to binary values, the optimal control trajectory has to be solved numerically.

Without any precautions, such a numerical solution of the binary problem can easily rule out any real-time applicability. A worst case scenario for the solution, for example with a *branch-and-bound* approach, would require the exploration of all n_u possible solutions. Consequently, the complexity would be exponential $\mathcal{O}(2^{n_u})$ if the complexity of each exploration is bounded $\mathcal{O}(1)$ with respect to n_u. For the experimental example considered below, the length of the design variable $\underline{U}_{j+1}$ amounts to $\underline{U}_{j+1} \in \mathbb{B}^{2500}$, while a cycle length as short as 50 milliseconds is considered. Solving the binary program directly on an actual personal computer, takes several hours for a single iteration. Thus, such an ILC setup is not suitable for real-time applications as the one addressed here.

To reduce the computational complexity, the cost function is modified first by translating the part of the cost accounting for the change of the control trajectory from cycle to cycle into a modified additional constraint S_{j+1}, which will be cycle-variant. Hence, $\mathbf{W}_{\Delta U}$ is set to zero in (11) and (12). The motivation for this reformulation comes from the class of problems considered here. The large, binary supervector of control inputs $\underline{U}_{j+1}$ will mainly consist of 0's. A 1 will indicate that at a specific point in time a specific control input is used to switch on the actuator addressed by this entry. In the next time instant, this entry will be 0 again as switching-on only occurs once in a period. As a result, the supervector $\underline{U}_{j+1}$ will consist of large related portions with identical entries 0. If it is now assumed that the pattern of the control trajectory in cycle j is not too far away from the optimal solution for $j+1$, the 1's appearing in $\underline{U}_{j+1}$ will be in the vicinity of the 1's in $\underline{U}_j$. As mentioned above, in a general binary optimization formulation, however, all possible changes of entries have to be calculated for an exhaustive search, e.g., in a *branch-and-bound* approach, even for the portions where it is unlikely to find a solution. For this reason, we propose a new cycle-variant constraint S_{j+1} that limits the number of possible control trajectories $\underline{U}_{j+1}$ that have to be considered. This finally leads to the following modified QP

$$\begin{aligned} \underline{U}_{j+1}^* = \arg\min \, & \underline{U}_{j+1}^T \mathbf{Q} \underline{U}_{j+1} + \underline{q}^T \underline{U}_{j+1} \\ \text{subject to} \quad & \underline{U}_{j+1} \in S_{j+1} \\ \text{with} \quad & \mathbf{Q} = \mathbf{\Phi}^T \mathbf{W}_E \mathbf{\Phi} \\ & \underline{q}^T = -2 \left(\underline{U}_j^T \left(\mathbf{\Phi}^T \mathbf{W}_E \mathbf{\Phi} \right) + \underline{E}_j^T \mathbf{W}_E \mathbf{\Phi} \right). \end{aligned} \tag{14}$$

The construction of S_{j+1}, which includes the binary constraint as well, will be detailed in the next section.

2.3 QP Transformation

The binary control trajectory

$$\underline{U}_j = [u_{j,1}, \ldots, u_{j,l}, \ldots, u_{j,n_u}]^T \tag{15}$$

of a cycle j consists of a certain number $n_{u0}(\underline{U}_j)$ of zeros and $n_{u1}(\underline{U}_j) = n_u - n_{u0}(\underline{U}_j)$ ones. As the system output $\underline{Y}_j$ in (1) is a linear combination of the columns of the transition matrix

$$\mathbf{\Phi} = [\underline{\phi}_1, \ldots, \underline{\phi}_l, \ldots, \underline{\phi}_{n_u}], \tag{16}$$

a column $\underline{\phi}_l$ that corresponds to a vanishing entry $u_{j,l} = 0$ in $\underline{U}_j$ will not have any impact on the output $\underline{Y}_j$. Hence, if $n_{u0}(\underline{U}_j) \gg n_{u1}(\underline{U}_j)$ a lot of information hold in $\mathbf{\Phi}$ is not used for the model prediction in cycle j and accordingly also for solving the QP (14). However, this might change in the upcoming cycle $j+1$. So unused columns from cycle j might become relevant again. The idea proposed in this contribution to reduce the computational complexity is to restrict the number of columns in $\mathbf{\Phi}$ that might become relevant for the next cycle $j+1$. This can be done with a vicinity argument, as for the experimental example, or by more refined approaches.

As described above, the change of the control trajectory $\Delta \underline{U}_{j+1} = \underline{U}_{j+1} - \underline{U}_j$ from one cycle to the next is limited by constraints that result in a solution space

$$S_{j+1} = \left\{ \hat{\underline{U}}_{j+1}^1, \ldots, \hat{\underline{U}}_{j+1}^h, \ldots, \hat{\underline{U}}_{j+1}^{n_s} \right\}. \tag{17}$$

The set consists of a finite number $n_s = |S|$ of potential control trajectories

$$\hat{\underline{U}}_{j+1}^h = [\hat{u}_{j+1,1}^h, \ldots, u_{j+1,l}^h, \ldots, u_{j+1,n_u}^h]^T \in \mathbb{B}^{n_u} \tag{18}$$

for the upcoming cycle $j+1$ including $\underline{U}_j \in \mathbb{B}^{n_u}$. From these potential control trajectories a vector

$$\underline{P}_{j+1} = [p_{j+1,1}, \ldots, p_{j+1,l}, \ldots, p_{j+1,n_u}]^T \tag{19}$$

is built with

$$p_{j+1,l} = \begin{cases} 1 & if \quad \exists \hat{\underline{U}}_{j+1}^h \in S_{j+1} \mid \hat{u}_{j+1,l}^h = 1 \\ 0 & else \end{cases}. \tag{20}$$

Using this vector, a transformation matrix $\mathbf{T}_{j+1}^T$ can be constructed starting from the diagonal matrix

$$\mathbf{D}_{j+1} = diag(\underline{P}_{j+1}) = [\underline{d}_1, \ldots, \underline{d}_l, \ldots, \underline{d}_{n_u}] \tag{21}$$

and cutting out all columns with $\underline{d}_l = \underline{0}$, where $\underline{0}$ represents a vector of zeros. Using the transformation matrix $\mathbf{T}_{j+1}$, the original control trajectory $\underline{U}_{j+1}$ can then be transformed into a surrogate control vector

$$\tilde{\underline{U}}_{j+1} = \mathbf{T}_{j+1} \underline{U}_{j+1} \in \mathbb{B}^m, \tag{22}$$

which condenses all those entries into $\tilde{\underline{U}}_{j+1}$ that are allowed to change during optimization. If $\underline{U}_{j+1} \in S_{j+1}$, (22) can be rearranged to invert the transformation:

$$\begin{aligned} \mathbf{T}_{j+1}^T \tilde{\underline{U}}_{j+1} &= \mathbf{T}_{j+1}^T \mathbf{T}_{j+1} \underline{U}_{j+1} \\ \mathbf{T}_{j+1}^T \tilde{\underline{U}}_{j+1} &= \mathbf{D}_{j+1} \underline{U}_{j+1} \\ \mathbf{T}_{j+1}^T \tilde{\underline{U}}_{j+1} &= \underline{U}_{j+1}. \end{aligned} \tag{23}$$

$\mathbf{D}_{j+1} \underline{U}_{j+1} = \underline{U}_{j+1}$ is true, as by construction all vanishing diagonal elements of $\mathbf{D}_{j+1}$ relate to zero elements in $\underline{U}_{j+1} \in S_{j+1}$. Therefore, $\mathbf{D}_{j+1}$ can be substituted for by the identity matrix. Equation (23) can now be used to adapt the QP (14) as follows:

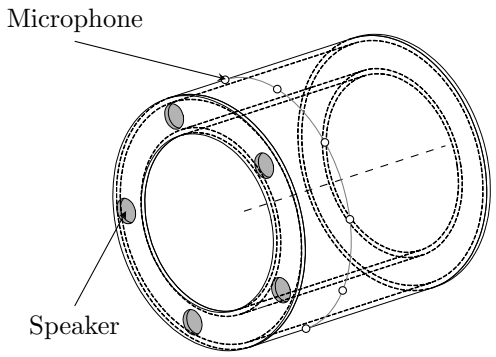

Fig. 1. Sketch of the acoustic test rig

$$\underline{\tilde{U}}^*_{j+1} = \underset{\underline{\tilde{U}} \in \mathbb{B}^m}{\arg\min} \quad \underline{\tilde{U}}^T_{j+1} \tilde{\mathbf{Q}} \underline{\tilde{U}}_{j+1} + \underline{\tilde{q}}^T \underline{\tilde{U}}_{j+1} \tag{24}$$
$$\text{with} \quad \tilde{\mathbf{Q}} = \mathbf{T}_{j+1} \mathbf{Q} \mathbf{T}^T_{j+1}$$
$$\underline{\tilde{q}}^T = \underline{q}^T \mathbf{T}^T_{j+1}.$$

An additional constraint is not required as the information about the solution space S_{j+1} is contained in the transformation, but $\underline{\tilde{U}} \in \mathbb{B}^m$ must hold. The optimal surrogate control trajectory $\underline{\tilde{U}}^*$ obtained, which can be re-transformed into the real domain

$$\underline{U}^* = \mathbf{T}^T_{j+1} \underline{\tilde{U}}^*, \tag{25}$$

can then be applied to the actual system.

If the number of zeros $n_{u0}(\underline{P}_{j+1})$ is large and a small set S_{j+1} is chosen, this transformation will reduce the dimension of the optimization problem significantly, and consequently allows for more efficient computation. The transformation increases the complexity in each iteration, as a matrix multiplication with $\mathcal{O}(n_s \cdot n_u)$ is required. But the more relevant complexity of binary QP-exploration is decreased $\mathcal{O}(2^{n_s}) \ll \mathcal{O}(2^{n_u})$, since $n_s \ll n_u$. A too small solution space S_{j+1}, however, can result in convergence issues, especially if the initial control trajectory $\underline{U}_0$ is far away from the optimal solution. Therefore, the presented reduction should only be used if an educated initial guess for $\underline{U}_0$, e.g., from an offline calculated optimal control is available.

3. EXPERIMENTAL SETUP

To give an impression of the capability of the presented approach, the control setup is tested on a real application. The experimental setup comprises an acoustic system consisting of five loudspeakers and an annular gap, see Fig. 1. It is similar to the setup used by Wolff et al. (2016) and Wolff and King (2019). This acoustic system is a mock-up of a combustion system with five tubes running a pulsed detonation combustion. The loudspeakers mimic the combustion by introducing a pre-recorded pressure disturbance into the annular gap. The pressure inside the gap is measured by microphones. More detailed information on the setup can be found in the aforementioned literature.

The goal of the presented controller is to synchronize the firing events to manipulate the pressure field inside the annular gap. A cyclic firing pattern of the loudspeakers is desired that yields improved pressure signals at $ny = 10$ measurement positions with respect to a cost function. The actuation of the system is done by binary values, where inside a cycle a firing event in the i-th tube/loudspeaker at discrete time k is marked by $u_i(k) = 1$. An optimal input synchronization is achieved by solving the binary optimization program based on a model of the process. An alternative way to synchronize such multi-input, linear, discrete-time dynamical systems is a multi-agent approach as used by Chen et al. (2015) or Heemels and Donkers (2013).

3.1 Model Description

For the optimal control as well as for the iterative learning control an appropriate model description of the acoustic system, which is compliant to (1), is required. The presented model is similar to the one found in Wolff and King (2019). The approach is based on the assumption that a constant periodic pressure signature with a fixed frequency is obtained after a number of cycles when a certain firing pattern is applied repeatedly with the same frequency. For the acoustic system considered here, it takes about five cycles at a cycle-frequency of 20 Hz to get to a proper cyclic operation.

As the experimental system can be described by linear acoustics, superposition holds. As a result, the model is constructed by superposition of responses obtained from experiments with a cyclic impulse excitation for each loudspeaker separately. It should be noted that an impulse means here that a fixed, pre-recorded loudspeaker signal is initiated. For this reason, the model obtained describes the acoustic system combined with the specific real continuous loudspeaker input applied.

The sampling rate is set to $f_s = 10\,\text{kHz}$ for the measurement as well as for the actuation. With a firing frequency of 20 Hz, a single period is split up into 500 discrete time steps. The binary actuation for the $nu = 5$ loudspeakers of the system can be described by the vector $\underline{u}(k) = [u_1(k), u_2(k), \dots u_5(k)]^T \in \mathbb{B}^5$ for the k-th time step of a period. As mentioned above, a value $u_i(k) = 1$ indicates a firing event of the speaker i at time k. The pressure measurements $\underline{y}(k) \in \mathbb{R}^{10}$ for time k are obtained at $ny = 10$ positions. For a complete cycle of $p = 500$ time steps, the measurements of all microphones, as well as the actuation signals of all loudspeakers are expressed as supervectors

$$\underline{Y} = \begin{bmatrix} \underline{y}(1) \\ \underline{y}(2) \\ \vdots \\ \underline{y}(500) \end{bmatrix} \quad \text{and} \quad \underline{U} = \begin{bmatrix} \underline{u}(1) \\ \underline{u}(2) \\ \vdots \\ \underline{u}(500) \end{bmatrix}. \tag{26}$$

To identify the model, each loudspeaker signal i is switched on for multiple cycles. This is represented in the model by a unit impulse $u_i(1) = 1$ as the physical actuation signal, i.e., the real voltage applied, has been assumed to be part of the system/model. After a transient phase, the cyclic response $\underline{I}_i$ for a complete cycle is measured. With these impulse responses, the input-output behavior for one cycle can be approximated by

$$\underline{Y} = \underbrace{\left[\underline{I}^0_1, \dots, \underline{I}^0_5, \underline{I}^1_1, \dots, \underline{I}^1_5, \dots, \underline{I}^{499}_1, \dots, \underline{I}^{499}_5\right]}_{\mathbf{\Phi}} \underline{U}. \tag{27}$$

The vector $\underline{I}_i^q$ is the measured impulse response $\underline{I}_i$ circularly shifted by q steps. The linear model (27) describes the influence of all 5 inputs on all 10 measurement positions for one period and can be used for the optimal control as well as for the ILC.

The model built from impulse responses gives a sufficient fit. Nevertheless, it should be noted that the system transition matrix Φ can also be obtained by other approaches, e.g., by a least squares fit, for systems for which an impulse-like signal cannot be applied due to technical reasons.

4. RESULTS

For the following results, the reference is set to $\underline{R} = \underline{0}$, resulting in a control that aims to reduce the combined $||\cdot||_2$-norm of all measured microphone signals. To set the solution space S_{j+1}, a vicinity argument is chosen for simplicity. Of course, more refined approaches are possible. The solution space S_{j+1} for each cycle is constructed by allowing a shift in the switching-on times by only a discrete time steps for each input. For $a = 2$, and with respect to the original number of time steps in one period, i.e., $p = 500$, the dimension of the binary QP that has to be solved for each cycle of the ILC is reduced from $n_u = 2500$ to $n_{\tilde{u}} = 25$. For a solution space with $a = 5$, the dimension of the QP is increased and grows to $n_{\tilde{u}} = 55$. As the size of the solution space $|S_{j+1}| = n_s = n_{\tilde{u}} = 5 \cdot (2 \cdot a + 1)$ depends linearly on the parameter a, it directly impacts the computational cost and the convergence behavior. A small a ensures efficient computation but might hold the risk of unpredictable convergence behavior. So for the actual application the setting of a is a trade-off between computational cost and predictability of the convergence behavior.

To get a first impression of the performance of the described control approach, experiments were run with varying initial control trajectories. The weight in (14) is set to $\mathbf{W}_E = \mathbf{I}$. Consequently, the cost $J(\underline{U})$ amounts to the sum of squared error. Figure 2 displays the mean square error over the control cycles starting with two generic patterns as well as a control trajectory close to the optimal solution. The results confirmed that the approach is highly sensitive to the initial control trajectory. Even though a significant improvement could be achieved for all initial trajectories, only the initial actuation close to the optimal solution resulted in a converged control error that was very close to the minimum error, which is given by the broken line. This is the trade off caused by the strong reduction of the original large scaled optimization problem. By dropping out large parts of the model information the controller might only converge to a local optimum. The reduced model information prohibits a further change in the control trajectory. However, other approaches to set the solution space S_{j+1} might yield better results.

The convergence behavior can be improved with an increased solution space. Figure 3 shows the error for a solution space with $a = 2$ and $a = 5$ again starting with an equidistantly distributed pattern as well as a simultaneous one. For these generic firing patterns, an improvement with a larger solution space is visible. In contrast, the results for an initial control trajectory close to the optimal

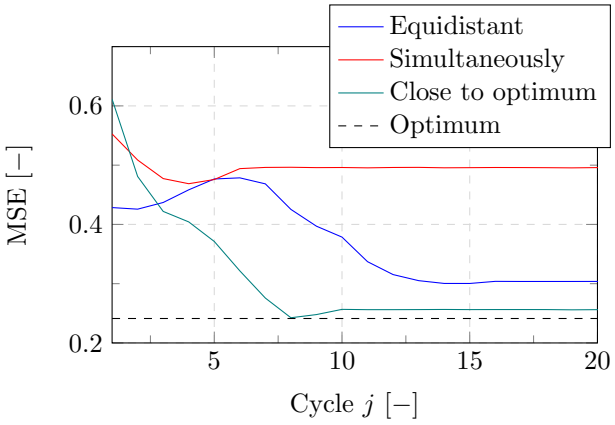

Fig. 2. Evolution of the mean squared error (MSE) over all 10 measurement positions starting with two generic (equidistant, simultaneous) firing patterns and one close to the optimal solution.

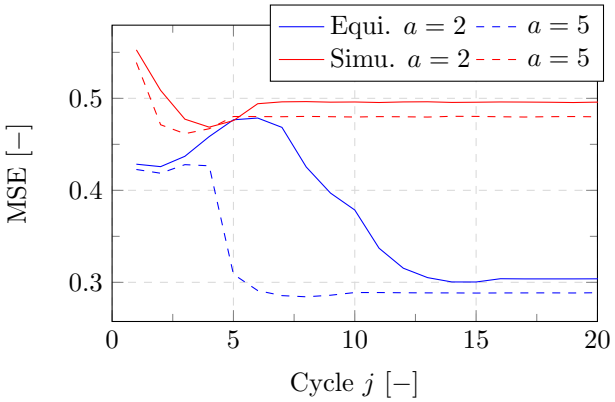

Fig. 3. Evolution of the mean squared error (MSE) over all 10 measurement positions starting with an equidistant respectively a simultaneous firing pattern using varying cardinality of the solution space

solution could not be further improved and, therefore, is not displayed again in Fig. 3. Obviously, the final step to the optimal solution requires a step that is not contained within the solution space S.

For the real setup with detonation tubes, pressure fluctuations will be much higher, violating the assumption of linear acoustics. As a preliminary robustness test of the method for such kind of problems, a static output nonlinearity is added to the system. Each actual measurement $y(t)$ of the acoustic test rig is transformed into a new output $y_n(t)$ by the static map

$$y_n(t) = 5 \cdot \mathrm{sgn}(y(t)) \cdot y^2(t). \tag{28}$$

The signum function $\mathrm{sgn}(y(t))$ is added to preserve the sign information of the original signal. This allows the controller to compensate for positive pressure disturbances with negative signals. Although superposition does not hold for this output, the model transition matrix Φ was constructed again using the impulse responses as above as an approximation. Then, the optimal binary control trajectory was calculated.

The results for this nonlinear setup are shown in Fig. 4. For the equidistant initial pattern, the mean squared error was considerably improved. The number of cycles to find

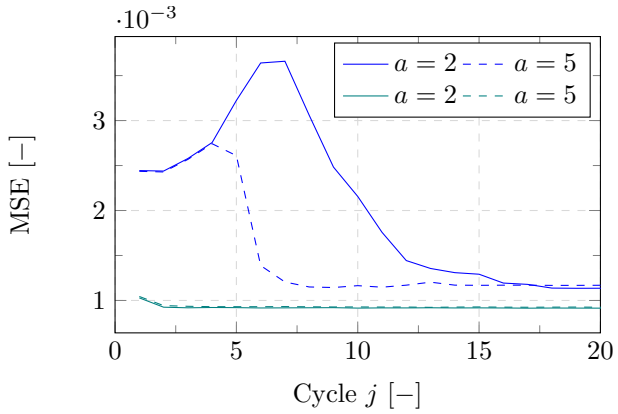

Fig. 4. Evolution of the mean squared error (MSE) over all 10 measurement positions with an applied static output nonlinearity starting with a generic firing pattern and one close to the calculated optimal solution

a local optimum was again smaller for a larger solution space S_{j+1}. From a detailed inspection of the control trajectories obtained, it became clear that even though the reached mean squared errors were in a same range, totally different trajectories were found by the algorithm. Obviously, different local minima were present. This again indicates the major impact of the solution space on the convergence behavior and the overall performance of the approach.

For the nonlinear experiment with an initial control trajectory close to the optimal solution, only a small improvement can be seen within the first three iterations. This applies for both investigated cases of the solution space, i.e., $a = 2$ and $a = 5$. Nevertheless, the overall level of the mean squared error is smaller than for the equidistant initial pattern.

5. CONCLUSION

The presented results indicate that for the class of systems considered here the introduced approach offers a suitable solution to reduce the computational cost of ILC applications with high dimensional binary control domains. Even though the control performance is slightly impaired by the used transformation, the controller still provides a sufficient reduction of the control error. With an initial control trajectory close to the real optimum the converged control error is very close to the minimal achievable error and is kept there due to the closed-loop action. Also for a nonlinear system, the reduced ILC setup showed robust results, which is promising with respect to experimental applications where linearity cannot be assumed.

Nevertheless, the introduced approach still holds potential for further improvements. Especially the selection of the solution space S is expected to impact the performance strongly. It might be possible to use information from the original large QP like the orientation described by the eigenvectors of the quadratic objective Q. An optimal compilation of S will be part of future investigations.

Furthermore, the convergence behavior of binary and more general integer valued ILC setups will be part of upcoming research as it is directly linked to possible performance improvements by means of reduced computational effort. This is necessary as conventional convergence analysis for ILC, like the one described by Ardakani et al. (2017) or Meng and Moore (2017), are not directly applicable for the systems with a discrete actuation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) as part of collaborative research center SFB 1029 'Substantial efficiency increase in gas turbines through direct use of coupled unsteady combustion and flow dynamics' in project A05.

REFERENCES

Ahn, H.S., Chen, Y.Q., and Moore, K.L. (2007). Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6), 1099–1121.

Amann, N., Owens, D.H., and Rogers, E. (1998). Predictive optimal iterative learning control. *International Journal of Control*, 69(2), 203–226.

Ardakani, M.M.G., Khong, S.Z., and Bernhardsson, B. (2017). On the convergence of iterative learning control. *Automatica*, 78, 266–273.

Arnold, F., Neuhäuser, K., and King, R. (2019). Experimental comparison of two integer valued iterative learning control approaches at a stator cascade. *Journal of Engineering for Gas Turbines and Power*, (GTP-19-1479).

Axehill, D., Vandenberghe, L., and Hansson, A. (2010). Convex relaxations for mixed integer predictive control. *Automatica*, 46(9), 1540–1545.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(June), 96–114.

Bürger, A., Zeile, C., Altmann-Dieses, A., Sager, S., and Diehl, M. (2019). Design, implementation and simulation of an MPC algorithm for switched nonlinear systems under combinatorial constraints. *Journal of Process Control*, 81, 15–30.

Chen, M.Z., Zhang, L., Su, H., and Li, C. (2015). Event-based synchronisation of linear discrete-time dynamical networks. *IET Control Theory and Applications*, 9(5), 755–765.

Heemels, W.P. and Donkers, M.C. (2013). Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3), 698–711.

Kochenberger, G., Hao, J.K., Glover, F., Lewis, M., Lü, Z., Wang, H., and Wang, Y. (2014). The unconstrained binary quadratic programming problem: A survey. *Journal of Combinatorial Optimization*, 28(1), 58–81.

Meng, D. and Moore, K.L. (2017). Convergence of iterative learning control for SISO nonrepetitive systems subject to iteration-dependent uncertainties. *Automatica*, 79, 167–177.

Wolff, S., Schäpel, J.S., and King, R. (2016). Application of Artificial Neural Networks for Misfiring Detection in an Annular Pulsed Detonation Combustor Mockup. *Journal of Engineering for Gas Turbines and Power*, 139(4), 041510.

Wolff, S.D. and King, R. (2019). Optimal Control for Firing Synchronization in an Annular Pulsed Detonation Combustor Mockup by Mixed-Integer Programming. In *AIAA Scitech Forum*, 2019-1742.