# Iterative Learning in Functional Space for Non-Square Linear Systems

Della Santina, C.; Angelini, Franco

**DOI**
10.1109/CDC45484.2021.9683673

**Publication date**
2021

**Document Version**
Final published version

**Published in**
Proceedings of the 60th IEEE Conference on Decision and Control (CDC 2021)

2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

# Iterative Learning in Functional Space for Non-Square Linear Systems

Cosimo Della Santina[1,2], Franco Angelini[3]

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683673

***Abstract*— Many control problems are naturally expressed in continuous time. Yet, in Iterative Learning Control of linear systems, sampling the output signal has proven to be a convenient strategy to simplify the learning process while sacrificing only marginally the overall performance. In this context, the control action is similarly discretized through zero-order hold - thus leading to a discrete-time system. With this paper, we want to investigate an alternative strategy, which is to track sampled outputs without masking the continuous nature of the input. Instead, we look at the whole input evolution as an element of a functional subspace. We show how standard results in linear Iterative Learning Control naturally extend to this context. As a result, we can leverage the infinite-dimensional nature of functional spaces to achieve exact tracking of strongly non-square systems (number of inputs less than outputs). We also show that constraints - like those imposed by intermittent control - can be naturally integrated within this framework.**

## I. Introduction

Iterative Learning Control (ILC) [1] identifies a class of control strategies devised to deal with repetitive motions. By closing the loop in the iteration domain rather than in time, these techniques can learn the feedforward action necessary to perform tasks with high precision. The application of ILC to square Multi-Input-Multi-Output (MIMO) systems is a widely studied topic [2]–[4]. The non-square case received instead far less attention. The few works dealing with this case focus on systems having more inputs than outputs [5], [6]. This condition is far less challenging than the other way around - i.e., more outputs than inputs. Indeed, dropping some of the inputs always brings the system into a square form. To the best of the Authors knowledge, the application and validation of ILC to systems with more outputs than inputs is still missing.

This work aims at tackling this challenge for continuous linear systems with sampled outputs. When applying ILC to such a system, the model is typically discretized using a zero-order hold. This simple action removes any requirement on high order derivatives which would instead be present if attacking the problem directly in continuous time [7]. Nevertheless, by constraining the input to be a piece-wise constant function, we are arbitrarily restraining the space of exploration of a learning strategy. Fig. 1 summarizes the proposed control framework. The idea is to track sampled outputs without imposing a piece-wise behavior on the input. Instead, we select the feedforward control action as an element of a functional subspace. This way, we can always pick a large enough functional space to deal with input and output spaces of any dimension.

This work has been supported by the EU project 101016970 NI.
[1]Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands. [2]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. [3]Centro di Ricerca "Enrico Piaggio" and Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy. Contact emails: c.dellasantina@tudelft.nl, frncangelini@gmail.com.

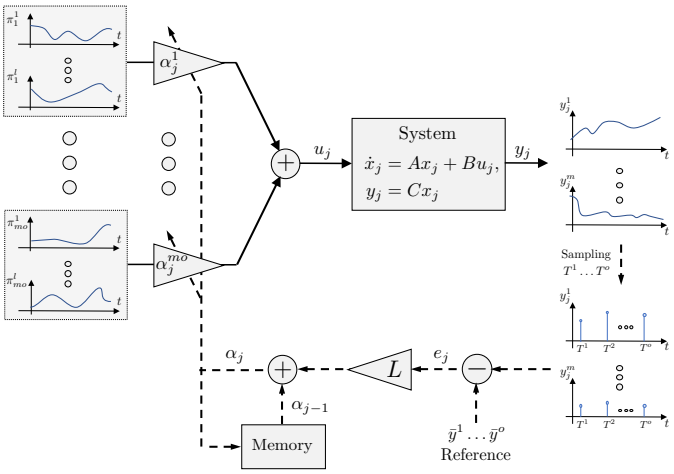



Fig. 1. Block scheme of the proposed Iterative Learning Control algorithm; $j$ is the iteration index. We have access to the continuous input signal $u_j$ and discrete samples of the output $y_j(T^1) \dots y_j(T^o)$. The goal is to learn the former so that the latter is equal to $\bar{y}^1 \dots \bar{y}^o$. We split the learning strategy into two phases. First, we express the control action as a linear combination of iteration-independent basis functions $\pi_1 \dots \pi_{mo}$. Then, we learn iteratively the weights $\alpha_j^1 \dots \alpha_j^{mo}$ through single-stage proportional error feedback.

Learning in a functional space has been investigated in [8]–[10] to improve the extrapolation properties of ILC. These works do not deal with non-square systems. Another key difference is that the functional expansion is operated directly on the discrete system rather than as an alternative to zero-order holding. In terminal ILC (TILC) [11]–[13] and Point-to-Point ILC (P2PILC) [14]–[16], the input signal is still a piece-wise constant function but sampled with higher frequency w.r.t. the output. The control action is updated using only the tracking error measurements of the samples of interest rather than the entire error signal. The extra input points are used to improve the performance of the tracking task. For example, in [17], the sampling of reference points of a P2PILC algorithm is optimized to minimize energy consumption. An analogous concept is proposed by [18], which considers a generic convex cost function. The connection between TILC, P2PILC, and ILC with incomplete information are discussed in [19], [20]. However, none of these works exploit the oversampling of the input to learn control actions for systems with fewer inputs than outputs.

To summarize, this work proposes a novel ILC framework for non-square MIMO systems which learns a continuous control action directly in functional space. We present necessary and sufficient conditions for its convergence. We also provide a general choice of the functional subspace, which assures convergence of the learning process for all controllable linear systems. Finally, we show that we can embed constraints on the input (limited bandwidth, impulsive control, exerting control only in specific intervals) directly into the basis functions.

### A. Units and index notation

We will not explicitly specify units in the rest of the paper. All physical units may be assumed to be expressed in the MKS system and angles in radian. In equations, we use subscripts to refer to iterations and superscripts to identify elements within vectors and matrices.

## II. Functional Iterative Learning Control

### A. Problem statement

Consider the linear non-square continuous system

$$\dot{x}_j = Ax_j + Bu_j, \quad y_j = Cx_j, \tag{1}$$

with iteration index $j$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times l}$, $C \in \mathbb{R}^{m\times n}$, $x_j \in \mathbb{R}^n$, $u_j \in \mathbb{R}^l$, $y_j \in \mathbb{R}^m$, with the usual meaning. We say that a system is non-square if $l \neq m$. The case $l < m$ is particularly challenging, while we can always treat $l > m$ as $l = m$ by dropping some inputs. Consider also a finite set of time instances $\{T^1, \dots, T^o\}$, where the superscript is intended as an index and thus not as a power. We take without loss of generality times ordered from the smaller to the larger, and all greater than 0. We take $T^0 = 0$ as starting time. We call $\bar{y}^1 \dots \bar{y}^o \in \mathbb{R}^m$ the desired values for the output at the given times. Our goal is to find a learning rule for $u_j(t)$ such that

$$\lim_{j\to\infty} y_j(T^k) = \bar{y}^k, \quad \forall k \in 1 \dots o. \tag{2}$$

Note that we do not impose any constraint on the control action at this stage. For example, we do not ask that it is piecewise constant as in standard ILC algorithms.

**Remark 1.** *All the results provided in this paper generalize easily to linear systems subject to iteration-independent disturbances. This happens in the usual way, by incorporating the effect of disturbance with the unforced evolution of the system. The required derivations are therefore not reported here for the sake of space.*

### B. Main Result

Statically, the input-output characteristic of (1) is not square. But in ILC we have the opportunity of looking at the whole evolution of the control action $u_j$ - and the space of all possible continuous functions is infinite-dimensional. This simple perspective shift allows us to make the problem square by selecting the control action from a large enough subspace of the functional space. We can parametrize a generic element of this space as follows

$$u_j(t) = \pi(t)\alpha_j, \tag{3}$$

and

$$\pi = \left[\pi^1 \dots \pi^o\right] \in \mathbb{R}^{l\times mo}, \tag{4}$$

with $\pi^i(t) \in \mathbb{R}^{l\times m}$ and $\alpha_j \in \mathbb{R}^{mo}$. In this way, we have achieved two goals. First, we have separated the components of $u_j(t)$ that are time-dependent from the ones that are iteration-dependent. Second, we recast the challenge of finding a generic $u$ into the more straightforward task of finding a learning rule for the vector $\alpha_j$. The following Theorem introduces such a learning rule and provides conditions for convergence.

**Theorem 1.** *Consider the linear learning rule*

$$\alpha_{j+1} = \alpha_j + L\begin{bmatrix} \bar{y}^1 - y_j(T^1) \\ \vdots \\ \bar{y}^o - y_j(T^o) \end{bmatrix}, \tag{5}$$

*with $L \in \mathbb{R}^{mo\times mo}$. The combination of* (3) *and* (5) *fulfills* (2) *with $x_j(0) = x_0$ for all $j$, if and only if*

$$\rho\left(I - LH\right) < 1, \tag{6}$$

*where $\rho$ is the spectral radius of the argument, and*

$$H = \begin{bmatrix} \int_0^{T^1} Ce^{A(T^1-\tau)}B\pi(\tau)\mathrm{d}\tau \\ \vdots \\ \int_0^{T^o} Ce^{A(T^o-\tau)}B\pi(\tau)\mathrm{d}\tau \end{bmatrix} \in \mathbb{R}^{mo\times mo}. \tag{7}$$

*Proof.* The closed form solution of (1) is

$$y_j(t) = Ce^{At}x(0) + C\int_0^t e^{A(t-\tau)}Bu_j(\tau)\mathrm{d}\tau. \tag{8}$$

We want to express how the $i-$th output of interest is affected by an input as (3). Thus, we evaluate

$$y_j(T^i) = Ce^{AT^i}x(0) + \left(\int_0^{T^i} Ce^{A(T^i-\tau)}B\pi(\tau)\mathrm{d}\tau\right)\alpha_j. \tag{9}$$

In the spirit of the super-vector notation, we collect all the outputs of interest in the vector $Y$, obtaining

$$Y_j = d_j + H\alpha_j, \tag{10}$$

where $d_j^i = Ce^{AT^i}x(0)$, and the $i-$th set of $m$ rows of $H$ is $\int_0^{T^i} Ce^{A(T^i-\tau)}B\pi(\tau)\mathrm{d}\tau$. Now that we have an equation which is formally akin to standard time discrete ILC literature, we can proceed in the typical way. Plugging (10) into (5) yields

$$\begin{aligned} \alpha_{j+1} &= \alpha_j + L(\bar{Y} - d - H\alpha_j) \\ &= (I - LH)\alpha_j + L(\bar{Y} - d), \end{aligned} \tag{11}$$

where $\bar{Y} \in \mathbb{R}^{mo}$ has as elements $(i-1)m+1$ to $im$ the $i-$th desired output $\bar{y}^i$. This equation represents a time discrete system in the only time variable $j$. A linear time discrete system is asymptotically stable if and only if all the eigenvalues of its dynamic matrix are within the unit circle. For (11) this is represented by condition (6).

This is enough to prove that the learning rule is convergent. We now must look for the asymptotic behavior. The steady state equilibrium of (11) is

$$\alpha_\infty = (LH)^{-1}L(\bar{Y} - d), \tag{12}$$

which combined with (10) yields

$$Y_\infty - \bar{Y} = -(\bar{Y} - d) + H(LH)^{-1}L(\bar{Y} - d). \tag{13}$$

Note now that both $L$ and $H$ are full rank by hypothesis. Indeed if this was not the case then $\det(LH) = 0$, and therefore at least one eigenvalue of $I - LH$ would be equal to 1, contradicting (6). Thus $(LH)^{-1} = H^{-1}L^{-1}$, which in turn implies

$$Y_\infty - \bar{Y} = 0. \tag{14}$$

The proof is concluded by noticing that this equation is the direct super-vector reformulation of (2). □

**Remark 2.** *The matrix $H$ can be evaluated from lab experiments without deriving an explicit model of the system.*

*Indeed, the $i-th$ set of $m$ columns can be obtained by exciting the system through $\pi^i$ and recording the responses at $\{T^1, \dots, T^o\}$.*

We have two design choices in defining the learning rule: the learning gains $L$ and the set of functions $\pi(t)$. For what concerns the latter, any choice of $\pi$ such that $H$ is full rank is fine. This leaves open the challenge of finding $L$. Eq. (5) is formally equivalent to classic linear learning rules, and condition (6) bears a clear resemblance to convergence conditions commonly found in the literature [1]. As a consequence, usual ILC algorithms can be ported in this framework with minimal changes. For example, for $\gamma \in \mathbb{R}$, the gain $L = \gamma I$ generates a completely model-free proportional rule, which is convergent if and only if $\rho(I - \gamma H) < 1$. If $H \succ 0$, then a small enough gain exists such that the learning is successful. If we hypothesize a more accurate knowledge of the plant, we can use the following damped pseudo-inverse as learning gain

$$L = (H^\top H + S)^{-1} H^\top, \tag{15}$$

where $S \succeq 0$. Following the same steps as in [21], we can prove that in this way at each step the learning process minimizes the cost function $\left|\left|\sum_{k=1}^{o} (\bar{y}^k - y_j^k)\right|\right|^2 + ||\alpha_j - \alpha_{j-1}||_S^2$. If $S = 0$ then $L = H^{-1}$ which always convergences in one step for a nominal plant (deadbeat learning).

### C. Simulations: $N$-masses system

We test here the proposed strategy in the extreme case of controlling the full state of a system by using a single input. Fig. 2 shows the considered mechanical system, which obeys to (16). This can be regarded as a simple template of soft robot [22]. For these simulations, we take $m = 1$, $\kappa = 2$, $\beta = 1$. The system is heavily under-actuated, having just one input and the full $2N$-dimensional state as output. We sample the output at the three instants $T^1, T^2, T^3$. The latter two are fixed to 18 and 20 seconds respectively. We perform simulations for increasing values of $N$ and $T^1 = 8$ or for $N = 3$ and $T^1 \in \{2, 5, 8, 11, 14\}$. The desired outputs are $\bar{y}^1 = (1/N, 2/N, \dots, 1, 0, \dots, 0)$, $\bar{y}^2 = (0, \dots, 0, 0, \dots, 0)$, and $\bar{y}^3 = \bar{y}^2$. Note that since $y = x$, then the first $N$ outputs are the positions of the masses, and the remaining $N$ are the velocities. As base functions $\pi$ we take a set of $6N$ Gaussians, with variance $T^3/(12N)$ and shifted in time such that they homogeneously cover the interval $[0, T^3]$. The resulting $H$ function is non singular for all the tested values, but its condition number gets too large when $N > 5$. We set the learning gain as prescribed by (15), with $S = 10^{-2} I$. Fig. 3 reports the evolution of the error across iterations. The learning process converges after few iterations to error close to zero in all the conditions tested - with the exception of $T^1 = 2$ and $T^1 = 14$. This can be interpreted by considering that in a two seconds span the number of Gaussians being in their variance range is less than three. Fig. 4 reports the full evolutions for $N = 5$ and $T^1 = 8$. The desired output is matched with an high level of precision, and with a reasonably regular control action.

## III. General selection of the functional space

We already discussed in the previous section that any choice of $\pi$ such that $\det(H) \neq 0$ is fine, and also that a positive defined $H$ is advantageous. A natural question is therefore when and how these conditions can be achieved. The following Lemma provides a general answer to these questions.

**Lemma 1.** *If $[A, B]$ is controllable, then the base functions*

$$\pi^i(t) = \begin{cases} B^\top e^{A^\top (T^i - t)} C^\top & \text{if } T^{i-1} \leq t < T^i \\ 0 \in \mathbb{R}^{l \times m} & \text{otherwise,} \end{cases} \tag{17}$$

*are such that $H$ is a block lower triangular and strictly positive definite matrix.*

*Proof.* Consider the sub-matrix $H^{i,j}$ composed of the $i-$th block of $m$ rows and $j-$th block of $m$ columns. According to the definition, its value is

$$H^{i,j} = \int_0^{T^i} C e^{A(T^i - \tau)} B \pi^j(\tau) \mathrm{d}\tau \in \mathbb{R}^{m \times m}. \tag{18}$$

Plugging (17) in it yields $H^{i,j} = 0$ if $j > i$. This proves that $H$ is lower block triangular. Algebraic manipulations allow to write the other blocks as

$$H^{i,j} = C\, e^{A(T^i - T^j)} \mathcal{G}_{T^j - T^{j-1}}\, C^\top, \tag{19}$$

where we introduced the finite time controllability gramian

$$\mathcal{G}_T = \int_0^T e^{A(T-\tau)} B B^\top e^{A^\top (T-\tau)} \mathrm{d}\tau \in \mathbb{R}^{n \times n}. \tag{20}$$

We have already proven that $H$ is block triangular. Thus, to check the rank of $H$ we can focus on the rank of diagonal blocks $H^{i,i} = C\, \mathcal{G}_{T^i - T^{i-1}}\, C^\top$. Since $[A, B]$ is controllable by hypothesis, then $\mathcal{G}_T$ is full rank and positive definite for all $T > 0$. Furthermore, $C \in \mathbb{R}^{m \times n}$ is full rank by definition, i.e. $\mathrm{Rank}\{C\} = m \leq n$. Thus, $H^{i,i} \succ 0$.

□

**Remark 3.** *Despite having the classical upper triangular structure, $H$ is not block Toeplitz. This only happens when the output sampling rate is constant, i.e. $T^j - T^{j-1} = T$ for all $j$.*

Note that (17) simplifies the assessment of decentralized learning rules. Thanks to the block triangular structure we can say that $\rho(I - \gamma C \mathcal{G}_{T^j - T^{j-1}} C^\top) < 1$, $\forall j \in \{1 \dots l\}$. Furthermore, since $C \mathcal{G}_{T^j - T^{j-1}} C^\top \succ 0$ then a small gain always exists such that the learning process is convergent.

### A. Simulations: $N$-masses system (Cont'd)

We repeat the simulations discussed in Sec. II-C when using (17) instead of Gaussians. The gain $L$ changes consequently. With this choice we can achieve good performance up to $N = 7$, corresponding to 14 outputs and 42 values controlled with a single input. Fig. 5 reports the evolution of the total error at each iteration. Again, the algorithm learns to achieve the goal in few steps. With this choice of $\pi$, the case $T^1 = 14$ does not show any qualitative difference w.r.t. the other considered times. The case $T^1 = 2$ is still challenging for the algorithm, but this time a clear decreasing pattern can be observed. Note that this lower pace is justified by the fact that 2 seconds is a very short time for taking the system so far from its equilibrium and locally stop it there (zero velocity). As a consequence, the necessary control action is high and the optimal gain (15) favours slow convergence over too large

$$\begin{bmatrix} \dot{x}_j^1 \\ \dot{x}_j^2 \\ \dot{x}_j^3 \\ \vdots \\ \dot{x}_j^N \\ \dot{x}_j^{N+1} \\ \dot{x}_j^{N+2} \\ \dot{x}_j^{N+3} \\ \vdots \\ \dot{x}_j^{2N} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 \\ -\frac{2\kappa}{m} & \frac{\kappa}{m} & 0 & \dots & 0 & -\frac{2\beta}{m} & \frac{\beta}{m} & 0 & \dots & 0 \\ \frac{\kappa}{m} & -\frac{2\kappa}{m} & \frac{\kappa}{m} & \dots & 0 & \frac{\beta}{m} & -\frac{2\beta}{m} & \frac{\beta}{m} & \dots & 0 \\ 0 & \frac{\kappa}{m} & -\frac{2\kappa}{m} & \dots & 0 & 0 & \frac{\beta}{m} & -\frac{2\beta}{m} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \dots & -\frac{\kappa}{m} & 0 & 0 & 0 & \dots & -\frac{\beta}{m} \end{bmatrix} \begin{bmatrix} x_j^1 \\ x_j^2 \\ x_j^3 \\ \vdots \\ x_j^N \\ x_j^{N+1} \\ x_j^{N+2} \\ x_j^{N+3} \\ \vdots \\ x_j^{2N} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \frac{1}{m} \end{bmatrix} u_j, \quad y_j = x_j \,. \tag{16}$$

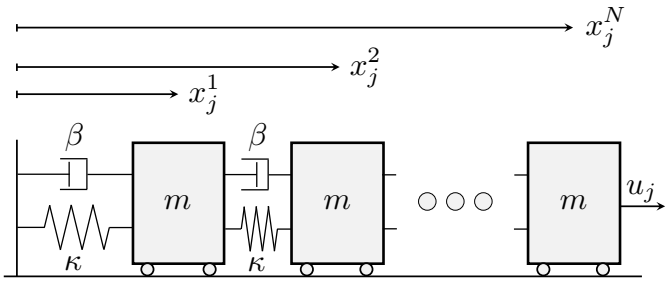
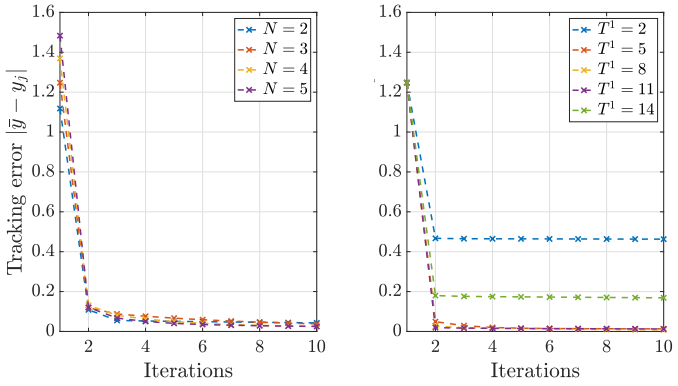



Fig. 2. An interconnection of $N$ masses, interacting through equal linear springs and dampers, and actuated through a single force $u_j$ applied at the $N-$th body. Each mass is free to move only in one direction. The variable $x_j^i$ measures the absolute position of the $i-$th mass. The remaining parameters are the mass $m$, the stiffness $\kappa$, and the damping $\beta$.




(a) Varying $N$ with $T^1 = 8$ (b) Varying $T^1$ with $N = 3$

Fig. 3. Evolution of tracking errors across iterations, when Gaussian base functions are used, together with an optimal tuning of the learning gain. Panel (a) shows the performance when the number of masses $N$ varies, and panel (b) depicts how learning evolves when the timing of the first goal is changed.

changes of $\alpha_j$. Fig. 6 shows the evolution of salient variables during the $10-$th iteration of the algorithm. The evolution of the output is similar to the one in Fig. 4. A main qualitative difference is that the evolution that we obtain now present lower frequency and higher amplitudes oscillations compared to the Gaussian case. The base functions $\pi$ are generated using (17) and as a result they are automatically scaled to the different time windows. The resulting control action is of similar amplitude to the Gaussian case. Again, we observe here low frequency oscillations. Since each time window has its set of base functions, the resulting evolution of $u_{10}$ is only piece-wise smooth. Interestingly though, this property equips

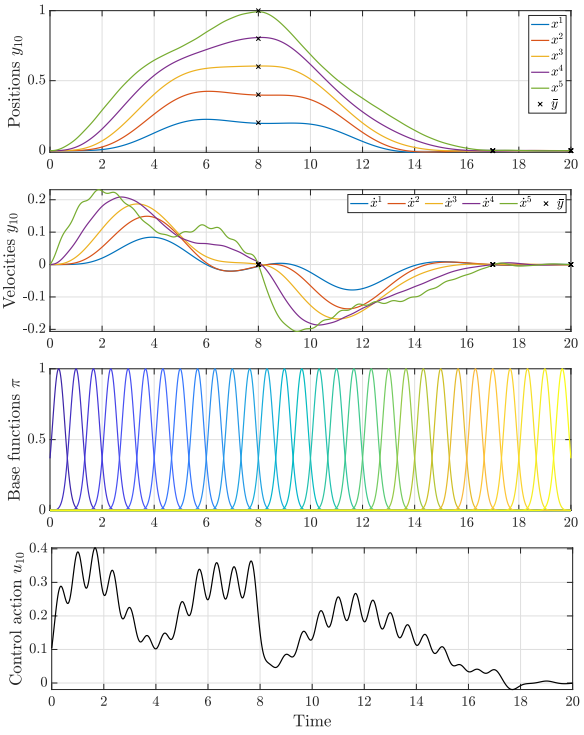
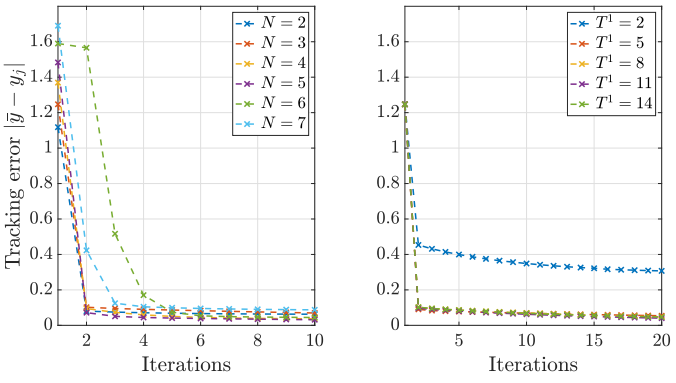



Fig. 4. Evolution of salient variables of the 5-masses systems, at the $10^{\text{th}}$ iteration of the algorithm. Gaussian base functions $\pi$ are used together with an optimal tuning of the learning gain. The first target is placed at $T^1 = 8$. Black crosses refer to the desired values of the outputs.

the algorithm with the ability of understanding that the system is at the equilibrium in $T^2$ and no action is needed to keep it there.

## IV. Constraints on the Input: Intermittent Control

The proposed framework allows imposing some constraints on the input in a natural way. For example, we may want our control action to have a maximum frequency content so that we do not incur in the speed limits of actuators, that the action is continuous up to some derivative order, or that the control is impulsive. These properties can all be achieved by selecting an appropriate family of base functions $\pi$. In the interest of space, we consider here only one type of constraint, which we believe is particularly relevant. In some experimental conditions, the control may be possible only in limited intervals of time. For the sake of space, we consider here the case in which we always have some actuation time from one sample and the




(a) Varying $N$ with $T^1 = 8$ (b) Varying $T^1$ with $N = 3$

Fig. 5. Evolution of tracking errors across iterations, when (17) is used to define the base functions, and the learning gain is optimally tuned. Panel (a) shows the performance when the number of masses $N$ varies, and panel (b) depicts how learning evolves when the timing of the first goal is changed.

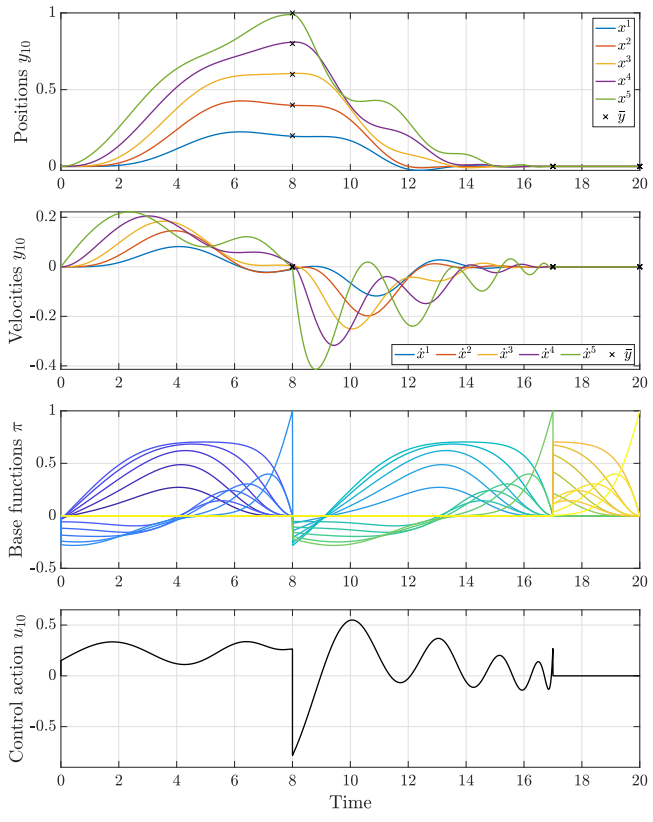



Fig. 6. Evolution of salient variables of the 5-masses systems, at the $10^{\text{th}}$ iteration of the algorithm. We use base functions $\pi$ generated according to (17), together with an optimal tuning of the learning gain. The first target is placed at $T^1 = 8$. Black crosses refer to the desired values of the outputs.

following one. Yet, the result could be easily extended to $a$ subsequent intervals without actuation if $ma \leq n$.

**Corollary 1.** *Consider* $\{\tilde{T}^1, \dots, \tilde{T}^o\}$ *such that* $T^{j-1} < \tilde{T}^j \leq T_j$. *If* $[A, B]$ *is controllable then the base functions*

$$\pi^i = \begin{cases} B^\top e^{A^\top (T^i - t)} C^\top & \text{if } T^{i-1} \leq t < \tilde{T}^i \\ 0 \in \mathbb{R}^{l \times m} & \text{otherwise.} \end{cases} \tag{21}$$

*are such that* $H$ *is an upper triangular, positive definite, full rank matrix.*

*Proof.* The proof follows the one of Lemma 1 up to (19), which becomes

$$H^{i,j} = C\, e^{A(T^i - \tilde{T}^j)} \mathcal{G}_{\tilde{T}^j - T^{j-1}}\, e^{A^\top (T^j - \tilde{T}^j)}\, C^\top. \tag{22}$$

Thus the $i-$th diagonal block is $H^{i,i} = C\, e^{A(T^i - \tilde{T}^i)} \mathcal{G}_{\tilde{T}^i - T^{i-1}}\, e^{A^\top (T^i - \tilde{T}^i)}\, C^\top$. In turn, this matrix is strictly positive definite since $\mathcal{G}_{\tilde{T}^i - T^{i-1}} \succ 0$ and $\det(Ce^{A(\tilde{T}_j - T_{j-1})}) \neq 0$. □

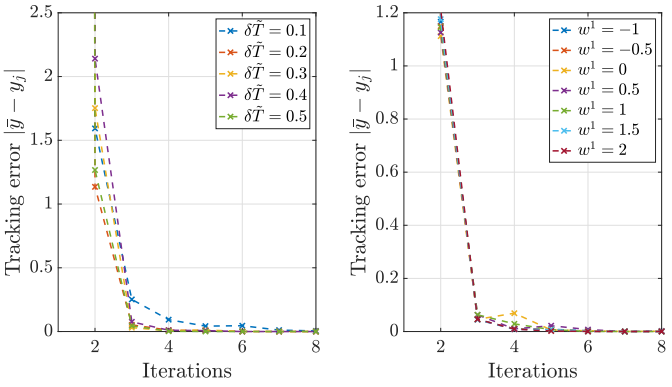
Therefore, we can apply the same base function that we applied before whenever the control is allowed and drop the value to 0 when needed. Note, however, that the final $H$ matrix is different, implying different learning gains $L$.

### A. Simulations: basketball in the wind

We take inspiration from a basketball player who learns how to juggle and dunk when the unknown constant wind is present. Also, the gravity constant is considered unknown. Both are supposed to be iteration-independent. The following equations describe the dynamics

$$\begin{bmatrix} \dot{x}^1 \\ \dot{x}^2 \\ \dot{x}^3 \\ \dot{x}^4 \\ \dot{x}^5 \\ \dot{x}^6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} u_j^1 + w^1 \\ u_j^2 + w^2 \\ u_j^3 - mg \end{bmatrix}, \tag{23}$$

where $m = 0.5$ is the mass of the ball, $w^1$ and $w^2$ are constant forces exerted by the wind, and $g$ is the gravitational constant. We take as output $y = (x_1, x_2, x_3)$, which are the three Cartesian positions expressed w.r.t. the chest of the player. The goal is to learn four juggles followed by a 3-points dunk. We encode this task as $\bar{y}^i = (0, 0, 0)$ for $i \in \{1, 2, 3, 4\}$ and $\bar{y}^5 = (6.75, 0, 1.55)$. The timing is $T^1 = 1$, $T^2 = 2$, $T^3 = 3$, $T^4 = 4$, $T^5 = 5.5$. The control action can be exerted in the time window $[T^i, T^i + \delta\tilde{T}]$. We enforce this behavior by selecting the base functions according to (21). The learning gains $L$ is optimally tuned through (15) with $S = 10^{-2}$. We perform two sets of simulations. In the first set we fix the wind to $(w^1, w^2) = (0.5, -0.9)$ and we vary $\delta\tilde{T}$. In the second, we take $\delta\tilde{T} = 0.5$ and we vary $w^1$. Fig. 7 shows the resulting evolutions of the error across iterations. Fig. 8 shows the evolution of the output (positions) and control action for $\delta\tilde{T} = 0.2$ and $w^1 = 0.5$. The desired targets are closely matched, by relying on very simple intermittent control actions. Note that since the control is intermittent the only way that the controller has to bring back the ball at the initial condition every second is to push it in the air and then rely on the open loop dynamics. The trajectory is coherent with what




(a) Varying $\delta\tilde{T}$ with $w^1 = 0.5$ (b) Varying $w^1$ with $\delta\tilde{T} = 0.2$

Fig. 7. Evolution of tracking errors across iterations for system (23), when (21) is used to define the base functions, and the learning gain is optimally tuned. Panel (a) shows the performance when the time available for exerting control changes, and panel (b) depicts what happens when the direction of the wind is varied. We cut the error at the first iteration since it is two degrees of magnitude higher than the current scale.

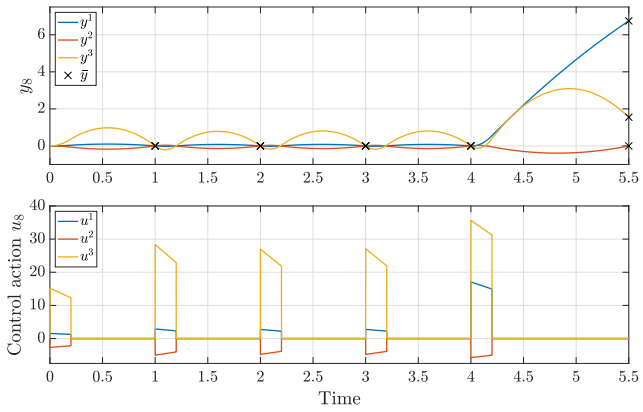



Fig. 8. Evolution of salient variables of the basketball example, at the $8^{\text{th}}$ iteration of the algorithm. We use base functions $\pi$ generated according to (21), together with an optimal tuning of the learning gain. In this simulations the algorithm has $\delta\tilde{T} = 0.5$ seconds to push the ball at the beginning of each interval, and the unkown frontal wind force is $w^1 = 0.5$. Black crosses refer to the desired values of the outputs.

we can observe in real world scenarios. Note that the ball is always pushed slightly side way to compensate for the wind.

## V. Conclusions

This work introduced an iterative learning control framework suited for learning continuous control actions that regulate the output of a linear system at discrete points. By exploiting the virtually infinite degrees of freedom provided by the continuous signals, we can deal with strongly non-square systems. We can also assure that the control action respects some constraints at each iteration, for example, non-null only during prescribed intervals. Future work will extend the theory towards time-varying systems and non-fixed sampling times by using arguments as in [23]. We then plan to test the resulting algorithms in fine-tuning control strategies in robotics.

## References


[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[2] B. J. Driessen and N. Sadegh, "Multi-input square iterative learning control with input rate limits and bounds," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 4, pp. 545–550, 2002.

[3] B. J. Driessen and N. Sadegh, "Convergence theory for multi-input discrete-time iterative learning control with coulomb friction, continuous outputs, and input bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 18, no. 5, pp. 457–471, 2004.

[4] M.-B. Radac, R.-E. Precup, and E. M. Petriu, "Model-free primitive-based iterative learning control approach to trajectory tracking of mimo systems with experimental validation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2925–2938, 2015.

[5] W. Hoffmann and A. G. Stefanopoulou, "Iterative learning control of electromechanical camless valve actuator," in *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, vol. 4. IEEE, 2001, pp. 2860–2866.

[6] L. Noueili, W. Chagra, and M. Ksouri, "New iterative learning control algorithm using learning gain based on $\sigma$ inversion for nonsquare multi-input multi-output systems," *Modelling and Simulation in Engineering*, vol. 2018, 2018.

[7] L. Cenceschi, F. Angelini, C. Della Santina, and A. Bicchi, "Pi$^\sigma$ - pi$^\sigma$ continuous iterative learning control for nonlinear systems with arbitrary relative degree," in *2021 European Control Conference (ECC)*. IEEE, 2021.

[8] J. van de Wijdeven and O. H. Bosgra, "Using basis functions in iterative learning control: analysis and design theory," *International Journal of Control*, vol. 83, no. 4, pp. 661–675, 2010.

[9] J. Bolder and T. Oomen, "Rational basis functions in iterative learning control—with experimental verification on a motion system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 722–729, 2014.

[10] L. Blanken, G. Isil, S. Koekebakker, and T. Oomen, "Flexible ilc: Towards a convex approach for non-causal rational basis functions," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12 107–12 112, 2017.

[11] R. Chi, D. Wang, Z. Hou, and S. Jin, "Data-driven optimal terminal iterative learning control," *Journal of Process Control*, vol. 22, no. 10, pp. 2026–2037, 2012.

[12] R. Chi, Y. Liu, Z. Hou, and S. Jin, "Data-driven terminal iterative learning control with high-order learning law for a class of non-linear discrete-time multiple-input–multiple output systems," *IET Control Theory & Applications*, vol. 9, no. 7, pp. 1075–1082, 2015.

[13] X. Dai, Q. Quan, J. Ren, Z. Xi, and K.-Y. Cai, "Terminal iterative learning control for autonomous aerial refueling under aerodynamic disturbances," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 7, pp. 1577–1584, 2018.

[14] T. D. Son and H.-S. Ahn, "Terminal iterative learning control with multiple intermediate pass points," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 3651–3656.

[15] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative learning control for multiple point-to-point tracking application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 590–600, 2010.

[16] R. Chi, Z. Hou, S. Jin, and B. Huang, "An improved data-driven point-to-point ilc using additional on-line control inputs with experimental verification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 4, pp. 687–696, 2017.

[17] X. Zhao and Y. Wang, "Energy-optimal time allocation in point-to-point ilc with specified output tracking," *IEEE Access*, vol. 7, pp. 122 595–122 604, 2019.

[18] Y. Chen, B. Chu, and C. T. Freeman, "Point-to-point iterative learning control with optimal tracking time allocation," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1685–1698, 2017.

[19] D. Shen and Y. Wang, "Survey on stochastic iterative learning control," *Journal of Process Control*, vol. 24, no. 12, pp. 64–77, 2014.

[20] D. Shen, "Iterative learning control with incomplete information: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 885–901, 2018.

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

[22] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft robots," *Encyclopedia of Robotics, M. Ang, O. Khatib, and B. Siciliano, Eds. Springer*, 2020.

[23] F. Angelini, R. Mengacci, C. Della Santina, M. G. Catalano, M. Garabini, A. Bicchi, and G. Grioli, "Time generalization of trajectories learned on articulated soft robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3493–3500, 2020.