

**Document Version**

Final published version

**Citation (APA)**

Ickenroth, T., Van Haren, M., Kon, J., Van Meer, M., Van Hulst, J., & Oomen, T. (2025). Automatic Basis Function Selection in Iterative Learning Control: A Sparsity-Promoting Approach Applied to an Industrial Printer. In *Proceedings of the 2025 American Control Conference, ACC 2025* (pp. 2931-2936). (Proceedings of the American Control Conference). IEEE. <https://doi.org/10.23919/ACC63710.2025.11107864>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.  
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

**Sharing and reuse**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

**Green Open Access added to [TU Delft Institutional Repository](#)  
as part of the Taverne amendment.**

More information about this copyright law amendment  
can be found at <https://www.openaccess.nl>.

Otherwise as indicated in the copyright section:  
the publisher is the copyright holder of this work and the  
author uses the Dutch legislation to make this work public.

# Automatic Basis Function Selection in Iterative Learning Control: A Sparsity-Promoting Approach Applied to an Industrial Printer

Tjeerd Ickenroth<sup>1</sup>, Max van Haren<sup>1</sup>, Johan Kon<sup>1</sup>, Max van Meer<sup>1</sup>, Jilles van Hulst<sup>1</sup>, and Tom Oomen<sup>1,2</sup>

**Abstract**—Iterative learning control (ILC) techniques are capable of improving the tracking performance of control systems that repeatedly perform similar tasks by utilizing data from past iterations. The aim of this paper is to design a systematic approach for learning parameterized feedforward signals with limited complexity. The developed method involves an iterative learning control in conjunction with a data-driven sparse subset selection procedure for basis function selection. The ILC algorithm that employs sparse optimization is able to automatically select relevant basis functions and is validated on an industrial flatbed printer.

## I. INTRODUCTION

Learning in control systems allows to improve tracking performance for systems that need to perform multiple motion tasks. By effectively exploiting large amounts of data that is currently available control systems can achieve outstanding performance. The data contains valuable information about the to be controlled system that subsequently can be used to learn important system properties. These properties are key to be able to effectively control mechatronic systems up to high precision when performing multiple tasks. Iterative Learning Control (ILC) has demonstrated considerable success in improving performance in systems that operate repetitively. By utilizing data from previous iterations, ILC refines the control input to effectively reject disturbances that repeat with each trial. ILC has been successfully implemented in real world applications in varying industries, e.g. robotics [1], mechatronics [2] [3], and manufacturing [4]. In spite of excellent reported tracking accuracy, the learned control signal is optimal for a fixed reference trajectory only. Since extrapolation to other references can lead to significant accuracy degradations [5], it is essential that extrapolation capabilities are introduced in ILC to achieve improvements for multiple tasks.

Since ILC is traditionally non-robust to varying references, several directions have been explored to develop such ILC algorithms that are flexible to task variations. Recent successful developments in ILC with basis functions include a physics based approach [6], Bayesian priors [7], polynomial basis functions [8] [9] [10], and rational basis functions [11] [12], which all aim to approximate the inverse system. An alternative approach is shown in [13] that requires a predefined

set of reference tasks.

The specification of basis functions on the basis of physical insight highly depends on the application at hand and the quality of the prior, hence the performance of such approaches varies. In particular, the results in [6] assume a very specific system structure, which has shown exceptional results for varying references in industrial applications in case the system is in the model class. If the system is not in the model class, other basis functions may be preferred, which heavily depend on the quality of the model of the system.

Although important developments have been made in task-flexible ILC and several successful applications have been reported, the basis function parameterization for performance and task flexibility is an open question. The aim of this paper is to perform automatic basis function selection for ILC.

Basis function selection for ILC involves several unique requirements, including task flexibility, performance, and low-order parameterizations for implementation and noise attenuation. Sparse subset selection methods in dynamic systems have been investigated in the field of system identification [14] [15] [16] where it is a common problem to estimate an accurate mathematical model of a real system using as few parameters as possible. Many techniques for sparse estimation have been successfully used for model structure selection in linear regression settings. For example, in *Forward Selection* regressors are added one by one according to how statistically significant they are [17]. Another approach to efficiently enforce sparsity is by the use of  $\ell_1$ -regularization, i.e., a penalty on the size of the parameter vector is added to the cost function and is referred as *Least Absolute Shrinkage and Selection Operator* (LASSO) [18]. *Least Angle Regression and Shrinkage* (LARS) is able to efficiently solve a LASSO problem along the solution path [19]. First adaptation of sparsity in ILC has been investigated in [20] where it has been used to construct sparse command signals. However, the use of such techniques to select basis functions is immature and not developed.

The main contribution of this paper is to develop a framework for automatic basis function selection based on sparse optimization for ILC, including experimental implementation and comparison.

The paper is structured as follows. The learning control problem is formally stated in Section II. In Section III the solution is proposed. Section IV elaborates on the parameterization of feedforward filters. In Section V, an experimental case study is presented that reveals the advantages of automatic basis function in ILC.

<sup>1</sup>Tjeerd Ickenroth (corresponding author), Max van Haren, Johan Kon, Max van Meer, Jilles van Hulst, and Tom Oomen are with the Control Systems Technology Section, department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.  
t.s.ickenroth@tue.nl

<sup>2</sup>Tom Oomen is with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

*Notation:* Throughout,  $\|x\|_{\ell_p}$  denotes the usual  $\ell_p$ -norm,  $p \in \mathbb{N}$ . Also,  $\|x\|_0 = \sum_i (x_i \neq 0)$ , i.e., the cardinality of  $x$ . Note that  $\|x\|_0$  is not a norm, since it does not satisfy the homogeneity property. It relates to the general  $\ell_p$ -norm by considering the limit  $p \rightarrow 0$  of  $\|x\|_p$ . A vector  $x \in \mathbb{R}^N$  is called sparse if many of its components are zero, in which case  $\|x\|_0 \ll N$ . Let  $\mathcal{H}(z)$  denote a discrete-time, linear time-invariant (LTI), single-input, single-output system. Vectors are denoted as lowercase letters and matrices with uppercase letters, e.g.,  $x$  and  $X$ . The columns and rows with index  $i$  of a matrix  $X$  are denoted by  $\text{col}_i(X)$  and  $\text{row}_i(X)$ , respectively. Let  $h(k) \forall k \in \mathbb{Z}$  be the impulse response coefficients of  $\mathcal{H}(z)$ , with the infinite impulse response  $y(k) = \sum_{\tau=-\infty}^{\infty} h(\tau)u(k-\tau)$ . Let  $u(k) = 0$  for  $k < 0$  and  $k \geq N$  to obtain the finite-time convolution

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix}}_y = \underbrace{\begin{bmatrix} h(0) & h(-1) & \cdots & h(-N+1) \\ h(1) & h(0) & \cdots & h(-N+2) \\ \vdots & \vdots & \ddots & h(-1) \\ h(N-1) & h(N-2) & \cdots & h(0) \end{bmatrix}}_H \underbrace{\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}}_u,$$

with  $y, u \in \mathbb{R}^N$  and  $H \in \mathbb{R}^{N \times N}$  is the finite-time convolution matrix corresponding to  $\mathcal{H}(z)$ .

## II. PROBLEM FORMULATION

In this section, the problem is formulated. First, the problem setup is presented. Second, the experimental setup is introduced. Finally, the problem that is addressed in this paper is defined.

### A. Control setup

Consider the closed-loop system shown in Fig. 1. The system and stabilizing feedback controller are denoted by  $P$  and  $C$ , respectively. Both are considered to be discrete time, single-input single-output (SISO), and linear time-invariant (LTI) systems. The associated ILC system, in this so-called lifted notation, is

$$e_j = S(r_j - P f_j) - S v_j, \quad (1)$$

where  $S = (I_N + PC)^{-1}$  is the sensitivity,  $j \in \mathbb{N}$  the trial, or experiment, of length  $N \in \mathbb{N}$ ,  $e_j \in \mathbb{R}^N$  the error signal to be minimized,  $r_j \in \mathbb{R}^N$  the reference signal that is possibly trial-varying,  $f_j \in \mathbb{R}^N$  the command signal, and  $v_j \in \mathbb{R}^N$  represents trial-varying disturbances, including measurement noise.

Similarly to (1), the error at trial  $j+1$  is given by

$$e_{j+1} = S(r_{j+1} - P f_{j+1}) - S v_{j+1}, \quad (2)$$

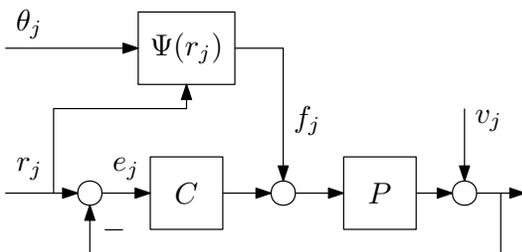


Fig. 1. Control structure considered.

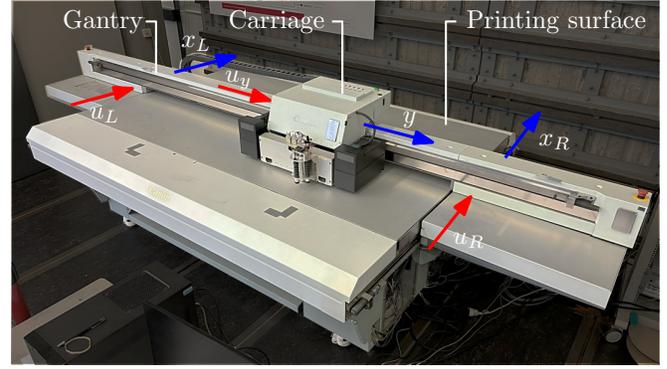


Fig. 2. Experimental setup: Canon Production Printing Arizona 550 GT flatbed printing system, with the inputs indicated in red and outputs in blue.

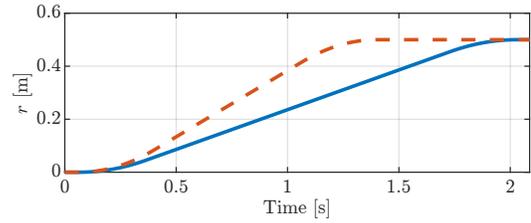


Fig. 3. The first (—) and second (---) faster motion tasks being applied to the Arizona flatbed printer.

and under the assumption that the reference is trial invariant, i.e.,  $r_{j+1} = r_j$ , the error propagation in lifted notation is given by

$$e_{j+1} = e_j - J(f_{j+1} - f_j) - S(v_{j+1} - v_j), \quad (3)$$

where  $J = PS \in \mathbb{R}^{N \times N}$  the finite-time convolution matrix of the process sensitivity.

### B. Application example

The mechatronic system considered in this paper is the flatbed printer shown in Fig. 2. In contrast to standard consumer printers, the medium is fixed on the printing surface. The carriage, which contains the print heads, translates in  $y$ -direction along the gantry, which translates in  $x$  and rotates in  $R_z$ . Hence, the printer is able to hover over the printing surface in both  $x$  and  $y$  directions. In this work a single-input single-output situation is considered, in which only the carriage is moved.

The printer must be able to perform multiple print tasks on sheets of varying widths and varying materials with high precision. The duration of a print task depends on the material to be printed and the type of print, e.g., glossy or matt. Therefore, the motion tasks that are applied to the carriage are not one single non-changing reference, but a set of references that require high accuracy tracking. Two examples of references with different printing speed and carriage acceleration are shown in Fig. 3.

### C. Learning problem

To allow flexibility in the motion task, the feedforward signal  $f_j$  is parameterized by a set of basis functions, denoted by  $\Psi(r_j) \in \mathbb{R}^{N \times N_{\theta}}$ , and learning parameters  $\theta_j \in \mathbb{R}^{N_{\theta}}$ ,

resulting in

$$f_j = \Psi(r_j)\theta_j, \quad (4)$$

as shown in Fig. 1. Substituting (4) into (3) the error propagation becomes

$$e_{j+1} = e_j - \mathbf{J}\Psi(r)(\theta_{j+1} - \theta_j) - \mathcal{S}(v_{j+1} - v_j), \quad (5)$$

which shows that  $e_{j+1}$  is linear in the parameters  $\theta_{j+1}$ . The objective of feedforward control in ILC is to minimize the tracking error of the next trial,  $e_{j+1}$  in (5), by designing  $\theta_{j+1}$  based on measured data of the previous trial, i.e.,  $e_j$  and  $\theta_j$ . The parameters  $\theta_j$ , along with the basis  $\Psi(r_j)$ , aim to describe the inverse system  $P^{-1}$ , i.e.,  $\Psi(r_j)\theta_j \approx P^{-1}r_j$  to obtain zero tracking error in the absence of noise, see (1). However, the tracking performance is limited by the ability of  $\Psi(r)\theta$  to describe the inverse system and heavily relies on the selection of basis  $\Psi(r)$ .

Therefore, the problem addressed in this paper is to develop an ILC algorithm that learns the optimal learning parameters  $\theta_j$  while at the same time automatically selects a subset of basis functions  $\Psi_S(r) \subseteq \Psi(r)$  that contribute to high tracking performance. This is summarized in the following requirements:

- R1) the iterations (5) lead to a small error  $e_j$  in the presence of both trial-varying reference  $r_j$  and disturbances  $v_j$ ;
- R2) a rich basis  $\Psi(r) \in \mathbb{R}^{N \times N_\theta}$  can be posed, i.e.,  $N_\theta$  can be large, whereas the algorithm automatically selects a (sparse) subset  $\Psi_S(r) \in \mathbb{R}^{N \times n_\theta} \subseteq \Psi(r)$  with  $n_\theta \ll N_\theta$  parameters, thereby controlling the computational load of the algorithm and be robust to trial-varying effects.

### III. SPARSE BASIS FUNCTION SELECTION

In this section the Sparse Basis Function Iterative Learning Control (SBF-ILC) algorithm is provided that satisfies Requirements R1 and R2. First, it is shown how to enforce exact sparsity in the learning parameters  $\theta$ . Second, the SBF-ILC algorithm is provided.

#### A. Enforcing exact sparsity in the learning parameters

The objective of feedforward control in ILC is to minimize the tracking error of the next trial,  $e_{j+1}$  in (5), by learning the parameters  $\theta_{j+1}$  based on measured data of the previous trial for a given set of basis functions  $\Psi(r_j)$ . The optimization problem is defined as

$$\mathcal{J}(\theta_{j+1}) = \frac{1}{2} \|e_{j+1}(\theta_{j+1})\|_{W_e}^2 + \frac{1}{2} \|\Psi(r_j)\theta_{j+1}\|_{W_f}^2 + \frac{1}{2} \|\Psi(r_j)(\theta_{j+1} - \theta_j)\|_{W_{\Delta f}}^2, \quad (6)$$

where  $\|x\|_W^2 = x^\top W x$ , and  $W_e, W_f, W_{\Delta f} \in \mathbb{R}^{N \times N}$  are symmetric positive (semi-)definite weighting matrices. Problem (6) is quadratic in the optimization variables  $\theta_{j+1}$ , and hence, has a minimizer that is analytically computed as

$$\theta_{j+1} = Q\theta_j + L e_j, \quad \text{with} \quad (7)$$

$$L = \left( \Psi^\top (J^\top W_e J + W_f + W_{\Delta f}) \Psi \right)^{-1} \Psi^\top J^\top W_e,$$

$$Q = \left( \Psi^\top (J^\top W_e J + W_f + W_{\Delta f}) \Psi \right)^{-1} \Psi^\top (J^\top W_e J + W_{\Delta f}) \Psi,$$

with learning and robustness matrices  $L \in \mathbb{R}^{N_\theta \times N}$  and  $Q \in \mathbb{R}^{N_\theta \times N_\theta}$ , respectively. Here the reference dependency has been removed for brevity.

In accordance with requirements R1 and R2, and given a large set of basis functions  $\Psi(r_j) \in \mathbb{R}^{N \times N_\theta}$ , a (sparse) subset

$$\Psi_S(r_j) \in \mathbb{R}^{N \times n_\theta} \subseteq \Psi(r_j)$$

with  $n_\theta$  learning parameters need to be automatically selected to control the computational load that at the same time leads to small tracking error  $e_j$ . Sparsity of a vector is measured directly through the amount of nonzero elements in this vector, i.e.,  $\|\theta_{j+1}\|_0$ . The addition of the  $\|\theta_{j+1}\|_0$  constraint in (6) results, however, in a non-convex problem, which in fact is NP-hard [21]. To overcome this, the problem is relaxed by an  $\ell_1$ -norm that promotes sparsity in the parameters  $\theta_{j+1}$ , generally referred as LASSO, and is given by the following optimization problem

$$\min_{\theta_{j+1}} \mathcal{J}(\theta_{j+1}) \quad \text{s.t.} \quad \|\theta_{j+1}\|_1 \leq s, \quad (8)$$

where  $s \in \mathfrak{R}_{>0}$  is an upper bound on the  $\ell_1$ -norm of the parameters: a small  $s$  leads to highly sparse results. The Lagrangian of (8) is the non-smooth objective

$$\min_{\theta_{j+1}} \mathcal{J}(\theta_{j+1}) + \lambda \|\theta_{j+1}\|_1 \quad (9)$$

where  $\lambda > 0$  can be interpreted as a regularization parameter: a large penalty  $\lambda$  corresponds to a small bound  $s$ , and vice versa.

The level of sparsity is controlled by the value of  $s$  in (8) (or  $\lambda$  in (9)) which is illustrated in Fig. 4. Here, the contours of the unconstrained objective (6), along with the contours of the  $\ell_0$ ,  $\ell_1$ , and  $\ell_2$  constraint sets are shown. It is geometrically clear that the  $\ell_0$  constraint defines a non-convex set which is challenging to solve, and that the  $\ell_1$  constraint promotes a sparse solution.

Exact sparsity in the learning parameters can be obtained

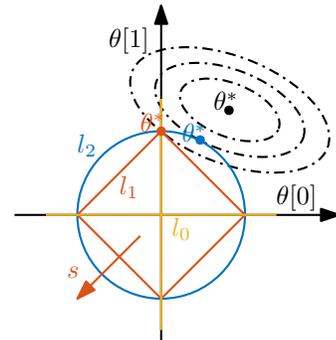


Fig. 4. Enforcing sparsity in ILC. Assuming  $N_\theta = 2$ , hence  $\theta$  contains two elements. The ellipsoidal contour lines of objective (6) are plotted in (---). The  $\ell_0$  constraint set is plotted in (—), which is, obviously, non-convex. The  $\ell_1$  constraint set is plotted in (—). The  $\ell_2$  constraint set is plotted in (—).  $\theta^*$  denotes the unconstrained optimal solution to objective function (6). The optimal constrained solution is found at the point where the contour line of the objective function first touches the constraint set, for the  $\ell_1$ -case, this implies that  $\theta[0] = 0$ , hence  $\theta^*$  is sparse. In contrast, for the ridge regression including the  $\ell_2$  constraint, the solution  $\theta^*$  is not sparse.

---

**Algorithm 1** Sparse Basis Function ILC

---

- 1: Initialize diagonal weighting matrices  $W_e, W_f, W_{\Delta f} \succ 0$ ,
  - 2: Initialize number of parameters  $N_\theta$  and cardinality  $n_\theta$ ,
  - 3: Construct  $\Psi(r_j)$  and  $X$  (see (10)),
  - 4: Initialize parameters  $\theta_{j \in 1, \dots, N_{\text{trials}}} = \mathbf{0}^{N_\theta \times 1}$ .
  - 5: **for**  $j \in 1, \dots, N_{\text{trials}}$  **do**
  - 6:   Update feedforward  $f_j = \Psi(r_j)\theta_j$ ;
  - 7:   Run an experiment  $e_j = Sr_j - Jf_j$ ;
  - 8:   Construct  $Y_j$  (see (10));
  - 9:   Find sparse vector  $\tilde{\theta}_{j+1} = \text{Lars}(X, Y_j, n_\theta)$  (*biased*).
  - Debiasing:
  - 10:   Find  $n_\theta$  non-sparse indices  $\tilde{\theta}_{j+1}[i] \neq 0$ ;
  - 11:   Construct active set  $X_A = \text{col}_i(X) \subseteq X$ ;
  - 12:   Calculate the least squares solution  $\text{row}_i(\theta_{j+1}) = (X_A^\top X_A)^{-1} X_A^\top Y_j$  (*debiased*).
  - 13: **end for**
- 

by solving regularization problem (9) along the regularization path via Least Angle Regression and Shrinkage [19]. Shrinkage refers to the process where the selected parameters are systematically reduced in magnitude, resulting in biased estimates of the obtained learning parameters. Hence, it is crucial to take one additional debiasing step to obtain the true learning parameters as shown in [20].

### B. SBF-ILC algorithm

The SBF-ILC algorithm is capable of efficiently selecting a (sparse) subset of basis functions, i.e.  $\Psi_S(r) \subseteq \Psi(r)$  with corresponding learning parameters  $\theta$  that result in the lowest tracking error. The SBF-ILC algorithm that satisfies requirements R1 and R2 is provided in Algorithm 1. Background on a LARS implementation can be retrieved from [22]. The expressions for the response vector  $Y_j \in \mathbb{R}^N$  and predictor matrix  $X \in \mathbb{R}^{N \times N_\theta}$  are obtained by rewriting the objective in (9) as

$$\|Y_j - X\theta_{j+1}\|_2^2 + \lambda \|\theta_{j+1}\|_1.$$

Substituting  $e_{j+1}(\theta_{j+1}) = e_j - J\Psi(r_j)(\theta_{j+1} - \theta_j)$  in (9) and using that  $\|\cdot\|_W^2 = \|\sqrt{W}\cdot\|_2^2$  for  $W$  diagonal, the matrices  $Y_j$  and  $X$  are as follows

$$Y_j = \begin{bmatrix} \sqrt{W_e}(e_j + J\Psi(r_j)\theta_j) \\ \mathbf{0}_{N \times 1} \\ -\sqrt{W_{\Delta f}}\Psi(r_j)\theta_j \end{bmatrix}, \quad X = \begin{bmatrix} \sqrt{W_e}J \\ -\sqrt{W_f} \\ -\sqrt{W_{\Delta f}} \end{bmatrix} \Psi(r_j). \quad (10)$$

**Remark.** Note that the SPILC algorithm in [20] is recovered as a special case where  $\Psi(r) = I_N$  to obtain sparse command signals instead of sparse basis function selection.

## IV. BASIS FUNCTION DESIGN

This section provides a possible parameterization for the set of basis functions. First, a parameterization based on physics is depicted that is often used in industry. Second, a parameterization of non-causal basis functions is introduced that allows to parameterize a large set of functions. To be able to achieve good tracking performance, it is required to learn an accurate estimate of the inverse plant

$P^{-1}$ . The fit accuracy is determined by the selected set of basis functions  $\Psi(r)$  together with their corresponding learning parameters  $\theta$ . The choice for basis  $\Psi(r)$  has major influence on the fitting capabilities and is therefore extremely important. As mentioned in Section II-C the basis functions with learning parameters aim to describe  $\Psi(r_j)\theta_j \approx P^{-1}r_j$  to obtain zero tracking error. Hence, the set of basis functions as input to the SBF-ILC algorithm must be rich to explore the full potential of the approach.

### A. Physical basis functions

In industry, basis functions are often based on physical insight on the system. The basis contains three physical interpretable parameters covering viscous friction, mass- and snap feedforward, i.e.,

$$\Psi(r) = [v \quad a \quad s], \quad (11)$$

where  $v$  the velocity,  $a$  the acceleration, and  $s$  the snap [6]. This parameterization leads to desirable applicable feedforward inputs but due to the limited amount of parameters in these basis functions the capability of approximating the inverse system remains limited, especially for complex mechatronic systems, resulting in unsatisfactory tracking performance.

### B. Non-causal basis functions

Under the assumption that the system under consideration, i.e.,  $P$ , is linear, it is known that a finite impulse response (FIR) filter is able to accurately describe the non-causal inverse system [7]. A non-causal FIR filter is described by

$$f_j[k] = \sum_{i=-n_p}^{N_\theta - n_p - 1} \theta_j[i] r[k - i], \quad (12)$$

where  $f_j[k]$  the feedforward signal at trial  $j$  at time instant  $k$ , and  $n_p$  the number of non-causal samples. Under the assumption that  $r[k] = 0$  for  $k < 0$ , and  $n_p = 0$ , then the expression in (12) is written in lifted notation as

$$f_j = \begin{bmatrix} f_j[0] \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_j[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} r[0] & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \vdots & & & r[0] \\ \vdots & & & \vdots \\ r[N-1] & \dots & \dots & r[N-N_\theta] \end{bmatrix}}_{\Psi_{\text{FIR}}(r)} \underbrace{\begin{bmatrix} \theta_j[0] \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \theta_j[N_\theta - 1] \end{bmatrix}}_{\theta_j}, \quad (13)$$

with  $\Psi_{\text{FIR}}(r)$  the toeplitz basis based on the finite impulse response. This FIR filter feedforward is able to outperform basis function ILC because of the better fitting capabilities of the inverse system due to an increased number of parameters. The hyperparameters of a FIR are the total number of parameters  $N_\theta$  and the number of non-causal parameters  $n_p$  and require a precise selection. For a large number of parameters  $N_\theta$  and the presence of noise in the error signal  $e_j$ , this structure is sensitive to overfitting measurement noise which is a common problem in ILC [20].

The SBF-ILC algorithm has automatic selection until  $n_\theta$  parameters have been found. Hence, it is allowed to select a wide range in causal and non-causal samples and it is only required to provide the total number of parameters  $n_\theta$  that need to be selected.

**Remark.** Note that an alternative parameterization for the set of basis functions can be formulated, such as an extension to basis functions which are nonlinear in the reference. The SBF-ILC algorithm would still perform an automatic subset selection.

## V. EXPERIMENTAL VALIDATION

In this section the proposed SBF-ILC algorithm is demonstrated using an industrial flatbed printer. First, the experimental setup is introduced. Second, the classes of ILC are presented together with the experimental outline. Last, the experimental results for all classes are presented.

### A. Experimental setup

The proposed sparse ILC algorithm is validated using the industrial Arizona flatbed printer, shown in Fig. 2. The input  $u_y$  is the force acting on the carriage, and the output  $y$  is its measured position. The translation and rotation of the gantry have a reference equal to zero, and no feedforward is applied in these directions.

### B. Experimental approach

To validate robustness to trial-varying tasks, hence, satisfying requirement R1, two fourth-order polynomial reference signals are designed using the approach in [6] with  $N = 2088$  samples and are shown in Fig. 3. The first reference is applied for 10 consecutive trials and is then switched to the second reference for another 10 consecutive trials.

To demonstrate the working principle of the SBF-ILC algorithm, hence satisfying requirement R2, the set of basis functions  $\Psi(r_j)$  are constructed based on a FIR filter, see (13), where the total number of parameters  $N_\theta = 200$  with  $n_p = 6$  samples preview. To validate performance of the algorithm it is compared to three different ILC methods that are generally applied. An overview of the experiments are listed below.

- E1) *SBF-ILC*: for  $\Psi = \Psi_{\text{FIR}}$ , as in (13), where  $N_\theta = 200$  and  $n_p = 6$ . Experiments have been performed for increasing values of desired sparsity, i.e., number of nonzero parameters  $n_\theta$ , and is ranging from  $n_\theta \in [1, 12]$ .
- E2) *Basis function (BF) ILC*: as is often considered in industry. Hence, the basis contains three physical parameters, i.e.,  $N_\theta = 3$ , and is equal to (11).
- E3) *Norm optimal (NO) FIR*: with  $\Psi = \Psi_{\text{FIR}}$ , as in (13), where  $n_p = 6$  and for two different number of parameters:
  - a)  $N_\theta = 100$ ,
  - b)  $N_\theta = 200$ .
- E4) *Norm optimal (NO) ILC*: where  $\Psi = I_N$  in (6), hence,  $N_\theta = N = 2088$ .

In all experiments the diagonal weighting matrices  $W_e$ ,  $W_f$ , and  $W_{\Delta f}$  were held constant to ensure a fair comparison. Besides this, a compensation has been done to overcome the nonlinear effect of static friction, and is applied the same way for all experiments, hence it does not result in any advantage for either approach.

### C. Validation results

The error 2-norm for all experiments E1-E4 at the last trial  $j = 20$  are shown in Fig. 5. Table I shows the performance loss factor after switching reference at trial  $j = 11$ . In Fig. 6 the resulting command signal  $f_j$  is shown at the final trial using SBF-ILC with  $n_\theta = 9$  and when using a FIR basis with  $N_\theta = 200$ .

The following observations are made:

- Fig. 5 shows that SBF-ILC outperforms basis functions ILC when using  $n_\theta \geq 4$  parameters. Note that 4 FIR filter parameters are required to reconstruct the same basis that is used in E2. Besides this, the performance improvement saturates when increasing the number of parameters. The result shows that using SBF-ILC with  $n_\theta = 7$  parameters achieves the same performance when using FIR ILC with  $N_\theta = 200$  parameters.
- Table I shows that the proposed SBF-ILC approach is robust to a reference change, since the loss factor is small compared to the other approaches.
- Fig. 5 shows that SBF-ILC with only  $n_\theta \in [7, 12]$  parameters obtains equal performance compared to FIR ILC having  $N_\theta \in \{100, 200\}$  parameters. However, the command signals  $f_j$  that are obtained after learning are dissimilar and are shown in Fig. 6. Here it is shown that

TABLE I  
PERFORMANCE LOSS FACTORS AFTER A REFERENCE CHANGE.

	SBF-ILC	BF-ILC	NO FIR	NO ILC
Parameters $n_\theta$	9	3	200	2088
$\ e\ _2$ loss factor at $r$ -switch	1.67	2.32	1.72	42.66

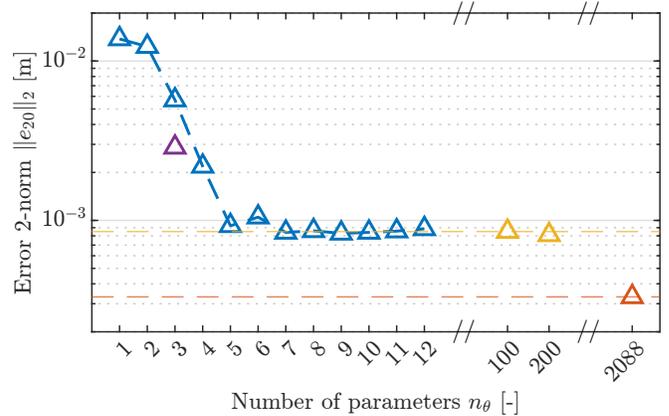


Fig. 5. Experimental error 2-norm at trial  $j = 20$  for basis function ILC with  $N_\theta = 3$  in ( $\Delta$ ), SBF-ILC for  $n_\theta \in [1, 12]$  in ( $\triangleleft$ ), NO FIR for  $N_\theta = \{100, 200\}$  in ( $\Delta$ ), and NO ILC with  $N_\theta = 2088$  in ( $\Delta$ ). Indeed, increasing the number of parameters results in better performance up to an extent, as after  $n_\theta = 7$  the performance saturates.

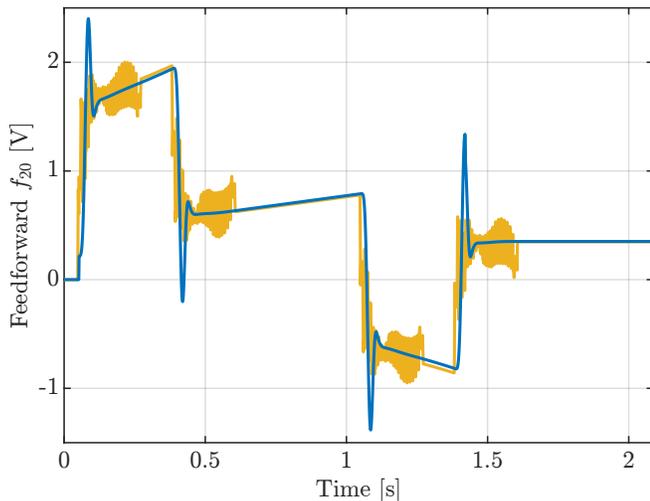


Fig. 6. Experimentally learned feedforward at trial  $j = 20$  for NO FIR with  $N_\theta = 200$  parameters in (—), and for SBF-ILC with  $n_\theta = 9$  parameters in (—). SBF-ILC learns a smooth command signal using significantly less parameters compared to NO FIR, while both obtain the same performance.

the FIR feedforward contains high frequency content which is due to overfitting the parameters, whereas the SBF-ILC feedforward is a smooth signal. This makes the SBF-ILC result much more favorable to apply on a real mechatronic setup.

## VI. CONCLUSION

In this paper, a sparse basis function iterative learning control (SBF-ILC) algorithm has been developed, which integrates automatic basis function selection to achieve both high tracking performance and robustness to varying tasks. The algorithm accommodates a large set of candidate basis functions, from which a sparse subset is selected by solving a LASSO optimization problem. This selection process identifies the basis functions that contribute most to minimizing the tracking error, with the level of sparsity directly controlled by the user. The SBF-ILC algorithm is validated on an industrial flatbed printer where it is compared to traditional basis function ILC, FIR ILC, and norm optimal ILC. Results demonstrate that SBF-ILC outperforms traditional basis function ILC, and can achieve equivalent performance to FIR ILC with only a fraction of the learning parameters. Consequently, SBF-ILC offers high tracking accuracy with reduced computational cost, producing smooth control signals suitable for industrial mechatronic applications.

## ACKNOWLEDGEMENT

This project is co-financed by Holland High Tech, top sector High-Tech Systems and Materials, with a PPP innovation subsidy for public-private partnerships for research and development. The authors gratefully acknowledge the contributions to this paper through a challenge-based learning project by Tim Aarts, Remco Bertels, Matthijs van Brunshot, Armando Cerullo, Yuri Copal, Hein van Dal, Roel Drenth, Hessel van Gemert, Bas Klis, Olaf van Lamsweerde,

Gijs van Meerbeek, Tom Minten, Gert Vankan, and Teun Wijfjes.

## REFERENCES

- [1] J. W. Axehill, I. Dressler, S. Gunnarsson, A. Robertsson, and M. Norrlöf, "Estimation-based ILC applied to a parallel kinematic robot," *Control Engineering Practice*, vol. 33, pp. 1–9, 2014.
- [2] J. Bolder, S. Kleinendorst, and T. Oomen, "Data-driven multivariable ILC: enhanced performance by eliminating L and Q filters," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 12, pp. 3728–3751, 2018.
- [3] D. Huang, J. X. Xu, V. Venkataramanan, and C. T. Huynh, "High-performance tracking of piezoelectric positioning stage using current-cycle iterative learning control with gain scheduling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1085–1098, 2014.
- [4] D. J. Hoelzle and K. L. Barton, "On Spatial Iterative Learning Control via 2-D Convolution: Stability Analysis and Computational Efficiency," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1504–1512, 2016.
- [5] F. Boeren, A. Bareja, T. Kok, and T. Oomen, "Frequency-Domain ILC Approach for Repeating and Varying Tasks: With Application to Semiconductor Bonding Equipment," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2716–2727, 2016.
- [6] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, 2005.
- [7] L. Blanken and T. Oomen, "Kernel-based identification of non-causal systems with application to inverse model control," *Automatica*, vol. 114, 2020.
- [8] J. Bolder, T. Oomen, S. Koekebakker, and M. Steinbuch, "Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer," 2014.
- [9] S. H. Van Der Meulen, R. L. Tousain, and O. H. Bosgra, "Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 130, no. 5, pp. 0510061–05100616, 2008.
- [10] 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.
- [11] J. Bolder, J. Van Zundert, S. Koekebakker, and T. Oomen, "Enhancing flatbed printer accuracy and throughput: Optimal rational feedforward controller tuning via iterative learning control," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4207–4216, 2017.
- [12] J. Van Zundert, J. Bolder, and T. Oomen, "Optimality and flexibility in Iterative Learning Control for varying tasks," *Automatica*, vol. 67, pp. 295–302, 2016.
- [13] D. J. Hoelzle, A. G. Alleyne, and A. J. W. Johnson, "Basis Task Approach to Iterative Learning Control With Applications to Micro-Robotic Deposition," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1138–1148, 2011.
- [14] C. R. Rojas and H. Hjalmarsson, "Sparse estimation based on a validation criterion," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2825–2830, 2011.
- [15] L. Ljung, H. Hjalmarsson, and H. Ohlsson, "Four encounters with system identification," *European Journal of Control*, vol. 17, no. 5–6, pp. 449–471, 2011.
- [16] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse Identification of Nonlinear Dynamics with Control (SINDYC)," in *IFAC-PapersOnLine*, vol. 49, pp. 710–715, 2016.
- [17] C. A. Ingene and S. Weisberg, "Applied Linear Regression," *Journal of Marketing Research*, vol. 18, no. 3, p. 389, 1981.
- [18] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *J. R. Statist. Soc. B*, no. 1, pp. 267–288, 1996.
- [19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [20] T. Oomen and C. R. Rojas, "Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility," *Mechatronics*, vol. 47, pp. 134–147, 2017.
- [21] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.
- [22] K. Sjöstrand, L. H. Clemmensen, G. Einarsson, R. Larsen, and B. Ersboll, "SpaSM: a MATLAB Toolbox for Sparse Statistical Modeling," *Journal of Statistical Software*, vol. 84, no. 10, 2018.