Sparsity-Promoting Iterative Learning Control for Resource-Constrained Control Systems

Burak Demirel, Euhanna Ghadimi, and Daniel E. Quevedo

Abstract— We propose novel iterative learning control algorithms to track a reference trajectory in resource-constrained control systems. In many applications, there are constraints on the number of control actions, delivered to the actuator from the controller, due to the limited bandwidth of communication channels or battery-operated sensors and actuators. We devise iterative learning techniques that create sparse control sequences with reduced communication and actuation instances while providing sensible reference tracking precision. Numerical simulations are provided to demonstrate the effectiveness of the proposed control method.

Index Terms— Iterative learning control; Sparse control; Convex optimization

I. INTRODUCTION

A multitude of techniques are now available in the literature for precise control of mechatronic systems; see, e.g., [1]. Iterative Learning Control (ILC) is one of the well-known techniques for accurately tracking reference trajectories in industrial systems, which repetitively executes a predefined operation over a finite duration; see, e.g., [2]–[4]. The key idea of iterative learning control relies on the use of the information gained from previous trails to update control inputs to be applied to the plant on the next trial. Iterative learning control was first introduced by Arimoto et al. [5] to achieve high accuracy control of mechatronic systems. Since the original work was published in 1984, it has been successfully practiced in various areas, including additive manufacturing machines [6], robotic arms [7], printing systems [8], electron microscopes [9], and wafer stages [10].

Modern industrial systems, which employ a large number of spatially distributed sensors and actuators to monitor and control physical processes, suffer from resource – control, communication, and computation – constraints. To provide a guaranteed performance or even preserve the stability of the closed-loop systems, it is necessary to take these limitations into account while designing and implementing control algorithms. Sometimes the limited bandwidth of legacy communication networks imposes a constraint on the rate of data transmissions. Besides, when the feedback loop is closed over wireless networks, a further resource constraint becomes apparent due to the use of battery-powered sensors and actuators [11]. The reduced actuator activity also prolongs the lifetime of actuators or improves the fuel efficiency. Therefore, it is desirable to have either sparse or sporadically changing control commands to reduce the use of actuators.

Sparsity-promoting techniques, which is borrowed from compressive sensing literature, have been successfully applied to a number of control problems to tackle the resource constraints mentioned above; see e.g., [12]–[17]. The authors of [12]–[14] modified the original model predictive control cost with an ℓ_1-penalty term to promote the sparsity in the control input trajectory. The authors of [15]–[17] designed energy-aware control algorithms to limit the actuator activity while providing an attainable control performance. Their design is also based on sparse optimization using ℓ_1-norm. To the best of our knowledge, the design of iterative learning control algorithms for resource-constrained systems has not been addressed in the literature and is subject of this paper.

Contributions. In this paper, we develop a Sparsity-promoting Iterative Learning Control (S-ILC) technique for resource-constrained control systems. The main departure from the standard ILC approach is that we introduce a regularization term into the usual ℓ_2-norm cost functions to render the resulting control inputs sparse. The sparsity here is in the cardinality of changes in control values applied to a finite horizon. Moreover, we include additional constraints to model the practical limits on the magnitude of applied control signals. The resulting control problem is then solved using a backward-forward splitting method which trades off between minimizing the tracking error and finding a sparse control input that optimizes the cost with respect to regularizer term. We demonstrate the monotonic convergence of the technique in lack of modeling imperfections. Moreover, we develop an accelerated algorithm to reduce the number of trials required for S-ILC to converge to optimality.

Outline. The remainder of this paper is organized as follows: Section II introduces the problem definition. Section III presents the sparse iterative learning control problem and associated algorithms to solve it. A numerical study is performed in Section IV. Finally, Section V presents concluding remarks. The appendix provides proofs of the main results

Notation. The n-dimensional real space is represented by $\mathbb{R}^n$. $\mathbb{E}$ denotes a finite dimensional euclidean space with inner product $\langle \cdot, \cdot \rangle$. For $u \in \mathbb{R}^n$, its ℓ_1 and ℓ_2 norms are

$$\| u \|_1 := \sum_{i=1}^{n} |u_i|, \quad \| u \| := \left(\sum_{i=1}^{n} u_i^2 \right)^{\frac{1}{2}}.$$

The spectral radius of the real square matrix $M \in \mathbb{R}^{n \times n}$ is denoted by $\rho(M)$. The Euclidean projection of $u \in \mathbb{R}^n$ into

B. Demirel and D. E. Quevedo are with the Chair for Automatic Control (EIM-E), Paderborn University, Warburger Straße 100, 33098, Paderborn, Germany `burak.demirel@protonmail.com` `dquevedo@ieee.org`. E. Ghadimi is with the Scania Group, SE-15187 Södertälje, Sweden, `euhanna.ghadimi@scania.com`.

the compact convex set $\mathcal{U}$ is denoted by $\Pi_{\mathcal{U}}(\cdot)$.

II. PROBLEM FORMULATION

A. System model

We consider the following discrete-time, single-input-single-output (SISO), stable, linear time-invariant (LTI) system $P(z)$ with state space representation:

$$x_k[t+1] = Ax_k[t] + Bu_k[t] , \qquad (1)$$
$$y_k[t] = Cx_k[t] , \qquad (2)$$

where $t \in \mathbb{N}_0$ is the time index (i.e., sample number), $k \in \mathbb{N}_0$ is the iteration number, $x_k[t] \in \mathbb{R}^n$ is the state variable, $u_k[t] \in \mathbb{R}$ is the control input, $y_k[t] \in \mathbb{R}$ is the output variable, and A, B and C are matrices of appropriate dimensions. The initial condition $x[0] = x_0$ is also assumed to be given, and these initial conditions are the same at the beginning of each trial. The input-output behavior of the system in (1) and (2), can be described via a convolution of the input with the impulse response of the system:

$$y_k[t] = CA^t x_0 + \sum_{\tau=0}^{t-1} CA^{t-\tau-1} Bu_k[\tau] . \qquad (3)$$

The coefficients $CA^t B$ for any $t \in \{0, 1, \cdots, \mathrm{T}\}$ are referred to as the Markov parameters of the plant $P(z)$, provided in (1) and (2).

B. Lifted system model

Since we focus on a finite trial length T, it is possible to evaluate (3) for all $t \in \{0, 1, \cdots, \mathrm{T}\}$ and, similar to [18], write its lifted version as

$$y_k = Gu_k + d , \qquad (4)$$

where

$$G = \begin{bmatrix} CA^{t^*-1}B & 0 & \cdots & 0 \\ CA^{t^*}B & CA^{t^*-1}B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{\mathrm{T}-1}B & CA^{\mathrm{T}-2}B & \cdots & CA^{t^*-1}B \end{bmatrix} ,$$
$$d = \begin{bmatrix} CA^{t^*} x_0 & CA^{t^*+1} x_0 & \cdots & CA^{\mathrm{T}} x_0 \end{bmatrix}^\top .$$

The vectors of inputs and output series are defined as

$$u_k = \begin{bmatrix} u_k[0] & u_k[1] & \cdots & u_k[\mathrm{T}-t^*] \end{bmatrix}^\top ,$$
$$y_k = \begin{bmatrix} y_k[t^*] & y_k[t^*+1] & \cdots & y_k[\mathrm{T}] \end{bmatrix}^\top .$$

The relative degree of the transfer function $P(z)$ is denoted by $t^* > 0$. Notice that the matrix G has a Toeplitz structure.

C. Trajectory tracking problem

In this paper, we focus on the reference trajectory tracking problem. It is assumed that a reference trajectory $r[t]$ is given over a finite time-interval between 0 and T. The objective is, here, to determine a control input trajectory $\{u[t]\}_{t=0}^{\mathrm{T}-t^*}$ that minimizes the tracking error:

$$\| e \|^2 \triangleq \| r - y \|^2 = \| r - Gu \|^2 , \qquad (5)$$

where

$$r = \begin{bmatrix} r[t^*] & r[t^*+1] & \cdots & r[\mathrm{T}] \end{bmatrix}^\top ,$$
$$e_k = \begin{bmatrix} e_k[t^*] & e_k[t^*+1] & \cdots & e_k[\mathrm{T}] \end{bmatrix}^\top .$$

The control sequence, which results in an output sequence $\{y[t]\}_{t=t^*}^{\mathrm{T}}$ that perfectly tracks the reference trajectory $\{r[t]\}_{t=t^*}^{\mathrm{T}}$, can be computed via solving the linear equation:

$$u^\star = G^{-1}(r - d) . \qquad (6)$$

Without loss of generality, one can assume that $x_0 = 0$, and, equivalently, $d = 0$. Hence, (6) can be rewritten as

$$u^\star = G^{-1}r . \qquad (7)$$

As argued in [19], the direct inversion of G is not practical in general since it requires having the exact information of G. Besides, instead of inverting the entire matrix G, it is sufficient to compute the pre-image of r under G.

D. Gradient-based iterative learning algorithm

There are various techniques in the literature to solve the unconstrained optimization problem (5) iteratively. The gradient-based iterative learning control algorithm has been received an increasing attention (see, e.g., [18]–[20]) due to its simplicity and light-weight computations compared to higher-order techniques. This algorithm generates the control inputs to be used in the next iteration using the relation:

$$u_{k+1} = u_k + \gamma G^\top e_k ,$$

where $\gamma > 0$ is the learning gain. Using this update law, the error evolves as

$$e_{k+1} = (I - \gamma GG^\top) e_k.$$

Using the norm inequality, provided in [21], we have:

$$\| e_{k+1} \| = \| (I - \gamma GG^\top) e_k \| \leq \| I - \gamma GG^\top \| \| e_k \| .$$

For minimum phase systems, the smallest singular value of the matrix G is nonzero and if one picks $0 < \gamma \leq 2/\rho(GG^\top)$, then $\|I - \gamma GG^\top\| < 1$ holds. Consequently, $\|e_k\|$ converges to zero linearly as $k \to \infty$.

It is worth noting that, for non-minimum phase systems, the matrix G has some singular values that are very close to zero; therefore, it might be significantly ill-conditioned, leading $\|I - \gamma GG^\top\|$ to become nearly one. Taking into account the typical rounding errors that exists in numerical solvers, it is safe to assume that the matrix G has zero singular values in order to avoid convergence issues due to mis-estimation of the optimal learning gain parameter.

E. Trajectory tracking problem with sparsity constraint

Trading off the accuracy of trajectory tracking for the sparsity in control signals amounts to solve

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \| r - Gu \|^2 \\ \text{subject to} \quad & \| Tu \|_0 \leq M \\ & u \in \mathcal{U}, \end{aligned} \qquad (8)$$

where $M \leq N$ with $M \in \mathbb{N}_0$ and $T \in \mathbb{R}^{N-1 \times N}$ is the difference matrix

$$T = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix},$$

and $\mathcal{U}$ is a compact and convex set which represents the practical limits on the input signal. For example, limits on the magnitude of input signal can be modeled by either by a box constraint or an upper bound on ℓ_∞-norm of control input u. With the cardinality constraint in (8), one limits the number of changes in control input values compared to the initial value $u[0]$, thereby promoting sparsity in the frequency of applying control input .

However, due to the cardinality constraint, the problem (8) is non-convex and difficult to solve. A common heuristic method in the literature relies on the ℓ_1-regularized problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2} \| r - Gu \|^2 + \lambda \| Tu \|_1 , \\ \text{subject to} \quad & u \in \mathcal{U}, \end{aligned} \tag{9}$$

where the second term is referred as *total variation* of signal u and the problem (9) is often called *total variation denoising* in signal processing literature [22].

III. SPARSE ITERATIVE LEARNING CONTROL

In this section, we develop a first-order method to solve the regularized control problem, proposed in (9), iteratively. Our technique is based on backward-forward splitting method [23], which is applied to the composite problem:

$$\text{minimize } F(u) := f(u) + g(u) \tag{10}$$

where $f : \mathbb{E} \to \mathbb{R}$ is a differentiable convex function with Lipschitz continuous gradient L satisfying

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{E},$$

and $g : \mathbb{E} \to (-\infty, +\infty]$ a proper closed convex function. Given a scalar $t > 0$ the *proximal* map associated to g is defined as

$$\text{prox}_t(g)(x) := \underset{u}{\text{argmin}} \left\{ g(u) + \frac{1}{2t}\|u - x\|^2 \right\}. \tag{11}$$

An important property of the proximal map is that this point (for any proper closed convex function g) is the unique solution to the associated minimization problem, and as a consequence, one has [24, Lemma 3.1]:

$$(I + t\partial g)^{-1}(x) = \text{prox}_t(g)(x), \ \forall x \in \mathbb{E}. \tag{12}$$

This result can be used to find the following optimality condition for (10)

$$\begin{aligned} 0 &\in t\nabla f(x^\star) + t\partial g(x^\star) \\ x^\star &= (I + t\partial g)^{-1}(I - t\nabla f)(x^\star), \end{aligned}$$

and then further developed to obtain a backward-forward splitting based method to solve (10)

$$\begin{aligned} x_{k+1} &= \text{prox}_\gamma(g)(x_k - \gamma \nabla f(x_k)) \\ &= \underset{x}{\text{argmin}} \left\{ g(x) + \frac{1}{2\gamma}\|x - (x_k - \gamma \nabla f(x_k))\|^2 \right\}. \end{aligned} \tag{13}$$

For instance, if $f = \|Ax - b\|^2$ and $g = \|x\|_1$, then the famous Iterative-Shrinkage-Thresholding Algorithm (ISTA) is recovered; see e.g., [25]. We use backward-forward splitting method to solve (9). In particular, let

$$f(u) := \frac{1}{2}\|Gu - r\|^2, \quad g(u) := \lambda\|Tu\|_1 + \mathcal{I}_{\mathcal{U}}(u), \tag{14}$$

with $\mathcal{I}_{\mathcal{U}}$ denoting the indicator function on $\mathcal{U}$; i.e., $\mathcal{I}_{\mathcal{U}}(u) = 0$ if $u \in \mathcal{U}$ and $\mathcal{I}_{\mathcal{U}}(u) = \infty$ otherwise. Applying the backward-forward splitting, the sparse ILC update rule is given by

$$\begin{aligned} u_{k+1} &= \text{prox}_{\lambda/\gamma}(g)(u_k + \gamma G^\top e_k) \\ &= \underset{u}{\text{argmin}} \left\{ \|Tu\|_1 + \mathcal{I}_{\mathcal{U}}(u) + \frac{1}{2\lambda\gamma}\|u - (u_k + \gamma G^\top e_k)\|^2 \right\} \end{aligned} \tag{15}$$

Unlike the ISTA algorithm with simple ℓ_1-norm regularization, the sparse ILC iterations (15) involve a proximal map that does not admit a closed-form solution. To tackle this problem, we develop an iterative *dual*-based approach for the proximal step. In particular, we are interested in solving

$$\underset{u \in \mathcal{U}}{\text{minimize}} \left\{ \lambda\|Tu\|_1 + \frac{1}{2}\|u - b\|^2 \right\}, \tag{16}$$

by using a first-order method. We have the following result:

Lemma 1 *Denote* $\mathcal{P}^{n-1} \subset \mathbb{R}^{n-1}$ *as the* $n - 1$ *dimensional real space bounded by unit infinity norm (i.e.,* $p \in \mathcal{P}^{n-1}$ *then* $\|p\|_\infty \leq 1$*) and* $\mathcal{L} \in \mathbb{R}^{n \times n-1}$ *given as*

$$\mathcal{L} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}.$$

Let $p \in \mathcal{P}^{n-1}$ *be the optimal solution of*

$$\begin{aligned} \underset{p \in \mathcal{P}^{n-1}}{\text{minimize}} \ h(p) :=& -\|\Pi_{\mathcal{U}}(b - \lambda \mathcal{L}p) - (b - \lambda \mathcal{L}p)\|^2 \\ & + \|b - \lambda \mathcal{L}p\|^2 . \end{aligned} \tag{17}$$

Then, the optimal solution of (16) *is given by*

$$u = \Pi_{\mathcal{U}}(b - \lambda \mathcal{L}p). \tag{18}$$

Next, we present the smoothness properties of (17).

Lemma 2 *The cost function in* (17) *is continuously differentiable, and its gradient is given by*

$$\nabla h(p) := -2\lambda \mathcal{L}^\top \Pi_{\mathcal{U}}(b - \lambda \mathcal{L}p) . \tag{19}$$

Moreover, its Lipschitz constant is bounded by

$$\begin{aligned} \|\nabla h(p) - \nabla h(p\prime)\| &\leq 2\lambda^2 \|\mathcal{L}^\top\|^2 \|p - p\prime\| \\ &= 2\lambda^2 \rho(\mathcal{L}^\top \mathcal{L})\|p - p\prime\|, \ \forall p, p\prime \in \mathcal{P}^{n-1}. \end{aligned} \tag{20}$$

Moreover, it follows $\rho(\mathcal{L}^\top\mathcal{L}) \leq 4$.

We are now ready to form an accelerated projected gradient-based method to solve (17) and (18). Algorithm 1 solves the problem by employing a Nesterov-like acceleration applied to the dual domain. The technique offers a better convergence rate $O(1/k^2)$ as opposed to a gradient-based technique that converges at rate $O(1/k)$; see e.g., [24].

Algorithm 1 Accelerated Projected Gradient

1: Let (N, λ, b) be given as input. Set $q_1 = 0$.
2: **for** $k = 1, \ldots, N$ compute **do**
3:

$$p_k = \Pi_{\mathcal{P}^{n-1}}\left[q_k + \frac{1}{\lambda\rho(\mathcal{L}^\top\mathcal{L})}\mathcal{L}^\top\Pi_{\mathcal{U}}(b - \lambda\mathcal{L}q_k)\right]$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$$

$$q_{k+1} = p_k + \frac{t_k - 1}{t_{k+1}}(p_k - p_{k-1})$$

where $\Pi_{\mathcal{P}^{n-1}}(x)_i = \dfrac{x_i}{\max\{1, |x_i|\}}$ for $i = 1, \ldots, n - 1$.

4: **end for**
5: Return $(x^\star, p^\star) = (\Pi_{\mathcal{U}}(b - \lambda\mathcal{L}p_N), p_N)$.

One can implement the sparse iterative learning control updates u_{k+1} in (15) by first taking a gradient step on u_k and then computing the proximal step via Algorithm 1. Algorithm 2 proposes the gradient-based S-ILC method.

Algorithm 2 Gradient-based S-ILC

1: Let $(N_1, N_2, \lambda, \mathcal{U})$ be given as input. Set u_0, e_0 to vector 0, and $\gamma = 1/\rho(G^\top G)$.
2: **for** $k = 1, \ldots, N_1$ **do**
3:     Set $b_k = u_{k-1} + \gamma G^\top e_{k-1}$.
4:     Run **Algorithm 1** with $(N_2, \gamma\lambda, b_k, \mathcal{U})$ and obtain $u_k \in \mathcal{U}$.
5:     Apply u_k to the plant and receive e_k.
6: **end for**
7: Return $u^\star = u_{N_1}$.

Next lemma confirms that the gradient-based S-ILC results in a non-increasing sequence.

Lemma 3 *Consider the sequence* $\{u_k\}_{k\geq 0}$ *generated by S-ILC. The associated functional values*

$$F(u_k) := \frac{1}{2}\|Gu_k - r\|^2 + \lambda\|Tu_k\|$$

is non-increasing. That is, for all $k \geq 1$,

$$F(u_{k+1}) \leq F(u_k).$$

Moreover, from [24, Theorem 3.1], it follows that

$$F(u_k) - F(u^\star) \leq \frac{\rho(G^\top G)\|u_0 - u^\star\|}{2k}, \ u_0 \in \mathcal{U};$$

where $u^\star$ is the optimal control input while $k \geq 1$ is the number of outer-loop iterations in the gradient-based S-ILC.

To accelerate the convergence of Algorithm 2, Nesterov-like iterations can be applied to its outer-loop. The straightforward application of the Nesterov's method leads to the following updates:

$$b_k = y_k - \gamma G^\top(Gy_k - r),$$
$$y_{k+1} = u_k + \frac{t_k - 1}{t_{k+1}}(u_k - u_{k-1}), \quad (21)$$

where u_k is the control input obtained from inner-loop Algorithm 1. However, this requires the access to the reference signal r, which is not practical in ILC application. We rewrite these updates to find a feasible formulation for ILC. Let $\Delta e_k := e_k - e_{k-1}$ and $y_{k+1} := u_k + \tau_{k+1}\Delta u_k$ where $\tau_{k+1} := (t_k - 1)/t_{k+1}$ and $\Delta u_k = u_k - u_{k-1}$. Now, we rewrite the b_k-th update in (21) as

$$b_{k+1} = y_{k+1} - \gamma G^\top(Gy_{k+1} - r)$$
$$= u_k + \tau_{k+1}\Delta u_k - \gamma G^\top(Gu_k + \tau_{k+1}G\Delta u_k - r),$$
$$= u_k + \tau_{k+1}\Delta u_k + \gamma G^\top(e_k + \tau_{k+1}\Delta e_k),$$

which relates the auxiliary variable b_k to – the readily available – control input and error signals.

Algorithm 3 Accelerated S-ILC

1: Let $(N_1, N_2, \lambda, \mathcal{U})$ be given as input. Set u_{-1}, u_0, e_{-1}, e_0 to vector 0, $t_0 = 0$, $t_1 = 1$, and $\gamma = 1/\rho(G^\top G)$.
2: **for** $k = 1, \ldots, N_1$ **do**
3:     Set

$$t_k = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 4t_{k-1}^2}, \quad \tau_k = \frac{t_{k-1} - 1}{t_k},$$

$$b_k = u_{k-1} + \tau_k\Delta u_{k-1} + \gamma G^\top(e_{k-1} + \tau_k\Delta e_{k-1}).$$

4:     Run **Algorithm 1** with $(N_2, \gamma\lambda, b_k, \mathcal{U})$ and obtain $u_k \in \mathcal{U}$.
5:     Apply u_k to the plant and receive e_k.
6: **end for**
7: Return $u^\star = u_{N_1}$.

Algorithm 3 presents the accelerated Nesterov-like iterates to solve S-ILC. From [26, Theorem 4.4], it yields

$$F(u_k) - F(u^\star) \leq \frac{2\rho(G^\top G)\|u_0 - u^\star\|}{(k+1)^2}, \ u_0 \in \mathcal{U};$$

where $k \geq 1$ is the outer-loop counter of Algorithm 3.

IV. NUMERICAL EXAMPLE

To demonstrate the effectiveness of S-ILC algorithms, we consider a robot arm (see [7]) with one rotational degree-of-freedom as schematically shown in Fig. 1. The input is the torque τ applied to the arm at the joint and is limited to the range of $\pm 12\,\mathrm{Nm}$, whereas the output θ is the angle of the arm measured as seen in Fig. 1. The dynamics of the robotic arm can be described by the following differential equation:

$$\ddot{\theta} = -\frac{g}{l}\sin\theta - \frac{c}{ml^2}\dot{\theta} + \frac{1}{ml^2}\tau, \quad (22)$$

where the arm length is $l = 1.0\,\mathrm{m}$, the payload mass is $m = 1.0\,\mathrm{kg}$, the viscous friction coefficient is $c = 2.0\,\mathrm{Nms/rad}$,

Fig. 1. A schematic drawing of the robot arm.

and the gravitational acceleration is $g = 9.81$ m/s². Changing the variables $x^{(1)} \triangleq \theta$, $x^{(2)} \triangleq \dot{\theta}$, $u \triangleq \tau$, and $y \triangleq \theta$, the nonlinear system (22) is sampled by using zero-order-hold and a sampling time of $T_s = 0.005$ s. The resulting discrete-time system becomes

$$x^{(1)}[t+1] = x^{(1)}[t] + T_s x^{(2)}[t] ,$$

$$x^{(2)}[t+1] = -\frac{gT_s}{l}\sin(x^{(1)}[t]) + \left(1 - \frac{cT_s}{ml^2}\right)x^{(2)}[t]$$
$$+ \frac{T_s}{ml^2}u[t] ,$$

$$y[t] = x^{(1)}[t] .$$

To construct the gradient of the cost function (5), which is equal to $G^\top$, the discrete-time non-linear plant model is linearized around the stationary point $x^{(1)} = \theta = 0$, resulting in the linear approximation:

$$x[t+1] = \begin{bmatrix} 1 & T_s \\ -\frac{gT_s}{l} & 1 - \frac{cT_s}{ml^2} \end{bmatrix} x[t] + \begin{bmatrix} 0 \\ \frac{T_s}{ml^2} \end{bmatrix} u[t] , \quad (23)$$

$$y[t] = \begin{bmatrix} 1 & 0 \end{bmatrix} x[t] . \quad (24)$$

Note that the linearized plant model is used to compute the gradient of the cost function (5), whereas the nonlinear model is employed in actual trials. The trial length is 6 s and the desired trajectory of the robot arm, illustrated in Fig. 2, is

$$r[t] = \frac{\pi}{5}\sin\left(\frac{\pi T_s t}{3}\right) + \frac{2\pi}{25}\sin\left(\pi T_s t\right)$$

for all $t \in \{0, 1, \cdots, 1200\}$.

TABLE I

COMPARISON OF REGULARIZATION PARAMETERS

$\lambda/\rho(G^\top G)$	$\| r - Gu \|_2$	$\| Tu \|_1$	$\| Tu \|_0$
0	1.0694	42.4495	1155
0.5	1.0845	38.0014	799
2.5	1.1406	34.5145	754
5	1.2117	33.0654	463

The simulation is carried out over 50 trials, and the results are displayed in Fig. 2. The optimization problem (9) becomes a least square problem with a box constraint when

Fig. 2. Tracking performance of the S-ILC algorithm for various values of the regularization parameter λ.

$\lambda = 0$. The resulting control input sequence provides the smallest tracking error possible. As seen in Fig. 2, when the regularization parameter λ increases, the control input sequence becomes more and more sparse at the expense of the increased tracking error. Similarly, Table I numerically illustrates the trade-off between the sparsity and the tracking performance. These experiments also demonstrate robustness against non-linearities of the plant. The change in the dynamics does not result in a divergence of the S-ILC algorithm.

Fig. 3 shows the error decay rate of the gradient-based and accelerated S-ILC algorithms over 50 trials. Moreover, we tried a multi-step technique called the heavy-ball method which is obtained by adding a momentum term $\beta(u_k - u_{k-1})$ to the b_{k+1}-update in Algorithm 2 where $\beta \in [0, 1)$ is a scalar parameter. The superior convergence properties of the heavy-ball method compared to the gradient

Fig. 3. The convergence of the error residual for different S-ILC methods.

method is known for twice continuously differentiable cost functions [27]. For the class of composite convex cost functions (10), however, the optimal algorithm parameters and associated convergence rate of the heavy-ball technique is still unknown [28]. Here, we evaluated the heavy-ball algorithm with $\beta = 0.4$. Numerical tests indicate that both Nesterov based (Algorithm 3) and heavy-ball methods improve the convergence of the gradient-based S-ILC Algorithm 2.

It is noteworthy to mention that unlike the gradient-based algorithm, the accelerated and heavy-ball methods are not monotonic, that is, the function values (10) are not guaranteed to be non-increasing. In our evaluations, we observed if the inner-loop Algorithm 1 is performed for a few iterations (so that it does not reach close to optimality), then the accelerated S-ILC problem (Algorithm 3) might become significantly non-monotonic to the point that it might diverge. Finding a monotone converging accelerated S-ILC algorithm is an interesting open problem.

V. CONCLUSIONS

This paper has presented novel iterative learning algorithms to follow reference trajectories under a limited resource utilization. The proposed techniques promote sparsity by solving an ℓ_2-norm optimization problem regularized by a total variation term. With proper selection of regularization parameter, our algorithms can strike a desirable trade-off between the accuracy of target tracking and the reduction of variations in actuation commands. Simulation results validated the efficacy of the proposed methods.

VI. APPENDIX

Proof of Lemma 1: The result can be derived in a similar way as [24, Proposition 4.1]. Here, for completeness, we include the proof. First, note $|x| = \underset{p}{\text{maximize}} \{px : |p| \leq 1\}$, and, similarly, $\|Tu\|_1 = \sum_{i=1}^{n-1} |u_i - u_{i+1}|$ can be written as

$$\underset{p}{\text{maximize}} \{\sum_{i=1}^{n-1} p_i(u_i - u_{i+1}) : |p_i| \leq 1\} = \underset{p \in \mathcal{P}^{n-1}}{\text{maximize}} \mathcal{L}p^\top u.$$

Accordingly, the problem (16) becomes

$$\underset{u \in \mathcal{U}}{\text{minimize}} \; \underset{p \in \mathcal{P}^{n-1}}{\text{maximize}} \; \frac{1}{2}\|u - b\|^2 + \lambda \mathcal{L}p^\top u.$$

Since this problem is convex in u and concave in p, the order of minimization and maximization can be changed to obtain

$$\underset{p \in \mathcal{P}^{n-1}}{\text{maximize}} \; \underset{u \in \mathcal{U}}{\text{minimize}} \; \frac{1}{2}\|u - b\|^2 + \lambda \mathcal{L}p^\top u.$$

Using the basic relation

$$\|x - b\|^2 + 2c^\top x = \|x - b + c\|^2 - \|b - c\|^2 + \|b\|^2,$$

with $x = u$ and $c = \lambda \mathcal{L}p$ results in the equivalent form

$$\underset{p \in \mathcal{P}^{n-1}}{\text{maximize}} \; \underset{u \in \mathcal{U}}{\text{minimize}} \; \|u - (b - \lambda \mathcal{L}p)\|^2 - \|b - \lambda \mathcal{L}p\|^2 + \|b\|^2.$$

The optimal solution of the minimization problem, readily, is given by (18). Instituting the optimal value of u, we arrive at the following dual problem

$$\underset{p \in \mathcal{P}^{n-1}}{\text{maximize}} \; \|\Pi_\mathcal{U}(b - \lambda \mathcal{L}p) - (b - \lambda \mathcal{L}p)\|^2 - \|b - \lambda \mathcal{L}p\|^2 \;,$$
(25)

which completes the proof. ∎

Proof of Lemma 2: Denote $s(x) := \frac{1}{2}\|x - \Pi_\mathcal{U}(x)\|^2$ and note that according to the *proximal* map the following identity holds:

$$s(x) = \inf_y \{\Pi_\mathcal{U}(y) + \frac{1}{2}\|y - x\|^2\}.$$

From [24, Lemma 3.1] it follows that $s(\cdot)$ is continuously differentiable with

$$\nabla s(x) := x - \Pi_\mathcal{U}(x). \tag{26}$$

The gradient of $h(p)$ then reads

$$\begin{aligned}
\nabla h(p) &= \nabla(-2s(b - \lambda \mathcal{L}p) + \|b - \lambda \mathcal{L}p\|^2) \\
&= 2\lambda \mathcal{L}^\top(b - \lambda \mathcal{L}p - \Pi_\mathcal{U}(b - \lambda \mathcal{L}p)) - 2\lambda \mathcal{L}^\top(b - \lambda \mathcal{L}p) \\
&= -2\lambda \mathcal{L}^\top \Pi_\mathcal{U}(b - \lambda \mathcal{L}p).
\end{aligned}$$

For any $p, p\prime \in \mathcal{P}^{n-1}$ we have

$$\begin{aligned}
&\|\nabla h(p) - \nabla h(p\prime)\| \\
&\quad = \|2\lambda \mathcal{L}^\top(\Pi_\mathcal{U}(b - \lambda \mathcal{L}p) - \Pi_\mathcal{U}(b - \lambda \mathcal{L}p\prime))\| \\
&\quad \leq 2\lambda \|\mathcal{L}^\top\| \|\Pi_\mathcal{U}(b - \lambda \mathcal{L}p) - \Pi_\mathcal{U}(b - \lambda \mathcal{L}p\prime)\| \\
&\quad \leq 2\lambda^2 \|\mathcal{L}^\top\| \|\mathcal{L}(p - p\prime)\| \\
&\quad \leq 2\lambda^2 \|\mathcal{L}^\top\| \|\mathcal{L}\| \|p - p\prime\| \\
&\quad = 2\lambda^2 \rho(\mathcal{L}^\top \mathcal{L}) \|p - p\prime\|.
\end{aligned}$$

The matrix $\mathcal{L}^\top \mathcal{L} \in \mathbb{R}^{n-1 \times n-1}$ is given by

$$\mathcal{L}^\mathsf{T}\mathcal{L} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{bmatrix}.$$

Following the Gershgorin circle theorem [21] one concludes the largest eigenvalue of this matrix follows $\rho(\mathcal{L}^\top \mathcal{L}) \leq \max_i \sum_{j=1}^{n-1} |[\mathcal{L}^\top \mathcal{L}]_{ij}| \leq 4$. ∎

Proof of Lemma 3: Let $f(u)$ and $g(u)$ be defined as (14) and

$$P_{\lambda/\rho}(u) := \text{prox}_{\lambda/\rho(G^\top G)}(g)(u - \frac{1}{\rho(G^\top G)}G^\top(Gu - r)).$$

Then the S-ILC algorithm (with converging inner-loop and no modeling error) can be rewritten as

$$u_{k+1} = P_{\lambda/\rho}(u_k).$$

For a convex Lipschitz continuous gradient function f and convex function g, define [26]:

$$Q_L(x,y) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2 + g(x)$$

Then, it can be seen that $P_{\lambda/\rho}(u) = \underset{x}{\text{argmin}}\, Q_{\lambda/\rho}(x, u)$. Furthermore, we have

$$
\begin{aligned}
F(u_k) &\geq Q_{\lambda/\rho}(u_k, u_k) \geq Q_{\lambda/\rho}(P_{\lambda/\rho}(u_k), u_k) \\
&= f(u_k) + \langle \nabla f(u_k), P_{\lambda/\rho}(u_k) - u_k \rangle \\
&\quad + \frac{\rho(G^\top G)}{2}\|P_{\lambda/\rho}(u_k) - u_k\|^2 + g(P_{\lambda/\rho}(u_k)) \\
&\geq f(P_{\lambda\rho}(u_k)) + g(P_{\lambda/\rho}(u_k)) \\
&= F(u_{k+1}),
\end{aligned}
$$

where the last inequality holds for Lipschitz continuous f. ∎

REFERENCES

[1] L. Güvenç, B. Aksun-Güvenç, B. Demirel, and M. T. Emirler, *Control of Mechatronic Systems*. London, UK: the Institution of Engineering and Technology, 2017.

[2] K. L. Moore, *Iterative Learning Control for Deterministic Systems*, ser. Advances in Industrial Control. Springer-Verlag, 1993.

[3] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, June 2006.

[4] D. H. Owens, *Iterative Learning Control: An Optimization Paradigm*, ser. Advances in Industrial Control. London, UK: Springer-Verlag, 2016.

[5] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronic systems," in *Proceeding of the 23rd IEEE Conference on Decision and Control*, 1984.

[6] K. Barton, D. Hoelzle, A. Alleyne, and A. Johnson, "Cross-coupled iterative learning control of systems with dissimilar dynamics: design and implementation," *International Journal of Control*, vol. 84, no. 7, pp. 1223–1233, 2011.

[7] M. Volckaert, M. Diehl, and J. Swevers, "Iterative learning control for nonlinear systems with input constraints and discontinuously changing dynamics," in *Proceeding of the American Control Conference*, 2011.

[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," *Mechatronics*, vol. 24, no. 8, pp. 944–953, Dec. 2014.

[9] G. M. Clayton, S. Tien, K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *ASME Journal of Dynamic Systems Measurement and Control*, vol. 131, no. 6, p. 061101, Oct. 2009.

[10] S. Mishra, J. Coaplen, and M. Tomizuka, "Precision positioning of wafer scanners: Segmented iterative learning control for nonrepetitive disturbances," *Control Systems Magazine*, vol. 27, no. 4, pp. 20–25, Aug. 2007.

[11] B. Demirel, "Architectures and performance analysis of wireless control systems," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, May 2015.

[12] M. Gallieri and J. M. Maciejowski, "ℓ_{asso} MPC: Smart regulation of overactuated systems," in *American Control Conference*, 2012.

[13] E. N. Hartley, M. Gallieri, and J. M. Maciejowski, "Terminal spacecraft rendezvous and capture with LASSO model predictive control," *International Journal of Control*, vol. 86, no. 11, pp. 2104–2113, Aug. 2013.

[14] M. Chyba, S. Grammatico, V. T. Huynh, J. Marriott, B. Piccoli, and R. N. Smith, "Reducing actuator switchings for motion control for autonomous underwater vehicle," in *Proceeding of the American Control Conference*, 2013.

[15] M. Nagahara, D. E. Quevedo, and J. Østergaard, "Sparse packetized predictive control networked control over erasure channels," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1899–1905, July 2014.

[16] M. Nagahara, J. Østergaard, and D. E. Quevedo, "Discrete-time hands-off control by sparse optimization," *EURASIP Journal on Advances in Signal Processing*, vol. 76, 2016.

[17] M. Nagahara, D. E. Quevedo, and D. Nesic, "Maximum hands-off control: A paradigm of control effort minimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 735–747, 2016.

[18] D. H. Owens, J. J. Hatonen, and S. Daley, "Robust monotone gradient-based discrete-time iterative learning control," *International Journal of Robust Nonlinear Control*, vol. 19, pp. 634–661, June 2009.

[19] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control using optimal feedback and feedforward actions," *International Journal of Control*, vol. 65, no. 2, pp. 277–293, 1996.

[20] B. Chu and D. Owens, "Singular value distribution of non-minimum phase systems with application to iterative learning control," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, USA: Cambridge University Press, 2013.

[22] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992.

[23] G. B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in hilbert space," *Journal of Mathematical Analysis and Applications*, vol. 72, no. 2, pp. 383 – 390, 1979.

[24] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, Nov 2009.

[25] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168 – 1200, 2005.

[26] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[27] B. T. Polyak, *Introduction to Optimization*. Optimization Software, 1987.

[28] E. Ghadimi, "Accelerating convergence of large-scale optimization algorithms," Ph.D. dissertation, KTH, Automatic Control, 2015.