Modified Newton Method based Iterative Learning Control Design for Discrete Nonlinear Systems with Constraints

Slawomir Mandra, Krzysztof Galkowski
Institute of Physics, Faculty of Physics, Astronomy and Informatics,
Nicolaus Copernicus University,
Grudziadzka 5, 87-100 Torun, Poland and
Institute of Control and Computation Engineering,
University of Zielona Gora,
Szafrana 2, 65-516 Zielona Gora, Poland.

Eric Rogers
Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK.
(etar@ecs.soton.ac.uk)

Andreas Rauh, Harald Aschemann,
Chair of Mechatronics,
University of Rostock,
Justus-von-Liebig-Weg 6, D-18059 Rostock, Germany.

Abstract

This paper considers the design of iterative learning control laws for classes of nonlinear dynamics. In particular, a new Newton method design is developed for nonlinear discrete systems in the presence of input constraints, where such constraints will arise in applications. The new design is based on the use of a penalty function and an iterative method for solving an unconstrained nonlinear optimization problem with an algorithm that has monotonic and super linear convergence characteristics. In this new algorithm the input inequality constraints are transformed into equality form by adding auxiliary variables. A cost function is then minimized to produce the new iterative learning control law design. Finally, a simulation based case study is given to illustrate the performance of the new design.

1 Introduction

Many industrial systems are required to undertake the same finite duration task over and over again. In operation, an execution, termed a trial in this paper, is completed over the finite trial length, the system resets to the starting location and the next trial can begin either immediately after the resetting is complete or after a further period of time has elapsed. Once a trial has been completed, all data generated during this trial is available to update the control signal for the next trial and thereby improve performance from trial-to-trial.

Iterative Learning Control (ILC) has been especially developed for such systems. Since the first work [1] it has become an established area of control systems research and application. The survey papers [2, 3] give comprehensive overviews of developments up to their years of publication. Major application areas include robotics and various forms of manufacturing processes, see, e.g., [4], and also a transfer from engineering to healthcare for robotic-assisted upper limb stroke rehabilitation with supporting clinical trials [5, 7], where the Newton method has also been used [8].

Let the integer k denote the trial number, $y_k(p)$ the output and $u_k(p)$ the input signals on this trial. All signals are defined over the finite interval $0 \leq p \leq N{-}1$, where $N < \infty$ denotes the number of sampling instants along the trial and in this paper attention is restricted to single-input single-output (SISO) systems with an immediate generalization to the multiple-input multiple-output case. A reference signal, denoted by $y_d(p),\ 0 \leq p \leq N-1$, is assumed to be available. Given this signal, the error on trial k is $e_k(p) = y_d(p) - y_k(p)$ and the basic ILC design problem is to force the sequence $\{e_k\}_{k\geq 0}$ to converge to zero, or to within an acceptable tolerance, in k where convergence is in terms of the norm on the underlying function space.

A large class of model-based ILC laws are designed using optimization, where the gradient method, see, e.g., [9] has been used. However, gradient-based designs may result in slow convergence speed and low efficiency and this performance issue has also led to use of the conjugate gradient method. For nonlinear plant models, a basic Newton method design does not guarantee that a matrix critical to the whole approach is nonsingular. This problem has led, in the non-ILC literature, to the development of modified quasi Newton methods, such as the BFGS (Broyden, Fletcher, Goldfarb, Shanno) algorithm [10], which given suitable development in the ILC setting, may enable a faster trial-to-trial error convergence rate. Moreover, solving the nonlinear equations defining the entries in the associated Hessian matrix is not required, which greatly reduces the computation and improves the efficiency. Previous work in the non-ILC literature has developed a penalty function method for a class of constrained optimization problems together with convergence analysis, see, e.g., [11].

This paper addresses ILC design based on a modified Newton method and the major novel contributions are:

i) the BFGS optimization algorithm is extended to ILC design for discrete nonlinear systems,

ii) a design algorithm with monotonic and super-linear (which is a desirable general requirement in applications) trial-to-trial error convergence rate where the speed of convergence is a requirement, and

iii) an extension to design in the presence of control input inequality constraints.

The remainder of this paper is organized as follows. Section 2 introduces a class of discrete nonlinear systems considered and writes the ILC dynamics as a set of discrete nonlinear algebraic equations to provide the starting point for Newton method based design In Section 3, a design based on the BFGS algorithm is developed. Section 4 formulates and solves the constrained ILC design problem for constraints on the control input on each trial. The solution is based on transforming the constrained problem to equality form by the use of a penalty functions. In Section 5, the monotonic and super linear convergence characteristics of the design are established and Section 6 gives a numerical example to highlight the new results. Finally, the last section summarizes the results obtained and discusses possible future research.

The notation used throughout this paper is standard. In particular, $\Re^n$ denotes the n-dimensional Euclidean space with norm $||x|| = \sqrt{x^T x}$. A symmetric positive-definite matrix, say Υ, is denoted by $\Upsilon \succ 0$ and I denotes the identity matrix with compatible dimensions.

2 Background and problem formulation

This paper considers single-input single-input discrete nonlinear systems described by the following state-space model in the ILC setting

$$\begin{aligned} x_k(p+1) =& f(x_k(p), u_k(p)), \\ y_k(p) =& h(x_k(p)), \end{aligned} \tag{1}$$

where the nonnegative integer subscript k denotes the trial number, p denotes the sampling instants, $0 \leq p \leq N-1$, $N < \infty$ is the number of samples along the trial (N times the sampling period gives the trial length), $x_k(p) \in \Re^n$, $u_k(p) \in \Re$ and $y_k(p) \in \Re$ represent the system state vector, input and output, respectively, and $f(\cdot)$ and $h(\cdot)$ are vector valued nonlinear functions. Without loss of generality, it is assumed that $x_k(0) = x_{\mathrm{d}}(0)$, i.e., an identical state initial vector on each trial.

The basis of the Newton method for ILC design is to replace the state-space model (1) by a set of algebraic equations in $\Re^N$ and requires the introduction of the input and output time-series vectors

$$u_k = \left[\begin{array}{cccc} u_k(0) & u_k(1) & \cdots & u_k(N-1) \end{array}\right]^T, \; y_k = \left[\begin{array}{cccc} y_k(1) & y_k(2) & \cdots & y_k(N) \end{array}\right]^T.$$

Using (1), the relationships between the input and output time-series can be expressed in terms of the following algebraic functions $g_1, g_2, \ldots, g_N$

$$\begin{aligned}
y_k(1) =& h(x_k(1)) = h(f(x_d(0), u_k(0))) = g_1(x_d(0), u_k(0)),\\
y_k(2) =& h(x_k(2)) = h(f(x_k(1), u_k(1))) = g_2(x_d(0), u_k(0), u_k(1)),\\
&\vdots\\
y_k(N) =& h(x_k(N)) = h(f(x_k(N-1), u_k(N-1))) = g_N(x_d(0), u_k(0), \ldots, u_k(N-1)).
\end{aligned}$$

Also, since the state initial vector on each trial is independent of k, the state-space model (1) can be represented by an algebraic function in $\Re^N$ with the structure

$$y_k = g(u_k), \tag{2}$$

where

$$g(u_k) = \left[\begin{array}{cccc} g_1(x_d(0), u_k(0)) & g_2(x_d(0), u_k(0), u_k(1)) & \ldots & g_N(x_d(0), u_k(0), \ldots, u_k(N-1)) \end{array}\right]^T.$$

The general ILC design problem is to find a control input sequence $\{u_k\}$ such that

$$\lim_{k\to\infty} ||e_k|| = 0,\ \lim_{k\to\infty} ||u_k - u_\infty|| = 0,$$

where u_∞ is termed the learned control and $||\cdot||$ denotes the norm on the underlying function space. In the case considered, the ILC dynamics have now been formulated as the nonlinear equations (2) and the problem of finding the desired input which forces (1) to track the supplied reference signal y_d is equivalent to finding the solution that satisfies (2) with y_k replaced by pre-specified the reference signal $y_d = \left[\begin{array}{cccc} y_d(1) & y_d(2) & \cdots & y_d(N) \end{array}\right]^T$.

Following the developments in, e.g., [12], the Newton-based ILC law is

$$u_{k+1} = u_k + z_{k+1},\ \ G_k(u_k)z_{k+1} = e_k,\ k \geq 1, \tag{3}$$

where $G_k(u_k)$ is the gradient matrix of $g(u_k)$. This law avoids (potentially) complex calculations to form the inverse of the nonlinear system dynamics (2). The inverse computation has been avoided by introducing $u_{k+1} = u_k + z_{k+1}$, where $z_{k+1} = (G_k(u_k))^{-1} e_k$, which is computed by solving $G_k(u_k)z_{k+1} = e_k$. In ILC terms, introducing $G_k(u_k)$ is equivalent to the linearization of (1) on trial k at (u_k, x_k). It can be shown, using properties of the parallel-chord method for solving nonlinear multivariable equations, see, e.g., [13], that, if convergent a Newton-based method exhibits local quadratic convergence, i.e., for the ILC case [12] the convergence of u_k to u_∞ satisfies

$$||u_{k+1} - u_\infty|| \leq c||u_k - u_\infty||^2,\ \ c > 0. \tag{4}$$

As a consequence the Newton-based ILC law (3) has the quadratic convergence property, but slow convergence speed can result. Also some applications require design in the presence of constraints to formulate a physically meaningful control law, see Section 4 for further discussion of an application area where input constraints are particularly required.

To address these issues, this paper develops the standard Newton method based ILC design to obtain a new version that has the super-linear convergence property for applications where fast trial-to-trial error convergence is required (without compromising other requirements). Also the design is extended to allow input constraints, where these are particularly relevant in ILC design since this form of control is based on direct computation of the control input for the next trial using previous trial data. The analysis that follows in this paper assumes that $g(\cdot)$ is a twice continuously differentiable function.

3 Control Law Design

Given the assumptions made, the system (2) can be approximated by a Taylor series expansion truncated after the first two terms and written in the form

$$g(u_k) \approx g(u_{k+1}) - G_{k+1}(u_{k+1})\Delta u_k + \frac{1}{2}\Im(\Delta u_k), \tag{5}$$

where $\Delta u_k = u_{k+1} - u_k$,

$$G_{k+1}(u_{k+1}) = \begin{bmatrix} \frac{\partial g_1(x_d(0),u_{k+1}(0))}{\partial u_{k+1}(0)} & 0 & \cdots & 0 \\ \frac{\partial g_1(x_d(0),u_{k+1}(0))}{\partial u_{k+1}(0)} & \frac{\partial g_2(x_d(0),u_{k+1}(0),u_{k+1}(1))}{\partial u_{k+1}(1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1(x_d(0),u_{k+1}(0))}{\partial u_{k+1}(0)} & \frac{\partial g_2(x_d(0),u_{k+1}(0),u_{k+1}(1))}{\partial u_{k+1}(1)} & \cdots & \frac{\partial g_N(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(N-1))}{\partial u_{k+1}(N-1)} \end{bmatrix},$$

$$\Im(\Delta u_k) = \begin{bmatrix} \Delta u_k^T H_{k+1,1}(u_{k+1})\Delta u_k \\ \Delta u_k^T H_{k+1,2}(u_{k+1})\Delta u_k \\ \vdots \\ \Delta u_k^T H_{k+1,N}(u_{k+1})\Delta u_k \end{bmatrix}, \; H_{k+1}(u_{k+1}) = \begin{bmatrix} H_{k+1,1}(u_{k+1}) \\ H_{k+1,2}(u_{k+1}) \\ \vdots \\ H_{k+1,N}(u_{k+1}) \end{bmatrix},$$

$$H_{k+1,j}(u_{k+1}) = \begin{bmatrix} \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}^2(0)} & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(0)\partial u_{k+1}(1)} & \cdots & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(0)\partial u_{k+1}(N-1)} \\ \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(1)\partial u_{k+1}(0)} & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}^2(1)} & \cdots & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(1)\partial u_{k+1}(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(N-1)\partial u_{k+1}(0)} & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}(N-1)\partial u_{k+1}(1)} & \cdots & \frac{\partial^2 g_j(x_d(0),u_{k+1}(0),\ldots,u_{k+1}(j-1))}{\partial u_{k+1}^2(N-1)} \end{bmatrix},$$

and $j = 1, 2, \cdots, N$. Furthermore, $G_{k+1}(u_{k+1}) \in \Re^{N\times N}$ and $H_{k+1}(u_{k+1}) \in \Re^{N\times N\times N}$ are, respectively, the gradient and Hessian matrices of $g(u_{k+1})$. Moreover, the derivative of the gradient matrix of (5) with respect to u_k is (where for ease of presentation the arguments of matrix-valued functions are dropped from this point onwards and also approximation is replaced by equality in further analysis of (5))

$$G_k = G_{k+1} - H_{k+1}\Delta u_k,$$

or, on setting $\Delta G_k = G_{k+1} - G_k$,

$$\Delta G_k = H_{k+1}\Delta u_k. \tag{6}$$

Cases will arise where the Hessian H_{k+1} cannot be accurately obtained and instead an approximation, Θ_{k+1}, to H_{k+1} is used, where the objectives are $i)$ to ensure that $\Theta_k \approx H_k$ for each trial to obtain a fast convergence rate and $ii)$ the calculation direction of the iterative algorithm approximates to the Newton direction. Also, if $\Theta_k \succ 0$ then the calculation direction of the algorithm is the descent direction of the function $g(\cdot)$ for u_k over the trial number.

At this stage (6) can be rewritten as

$$\Delta G_k \approx \Theta_{k+1}\Delta u_k, \tag{7}$$

where, on introducing (and again replacing approximation by equality in further analysis of (7)),

$$\begin{aligned} \Theta_{k+1} &= \begin{bmatrix} \Theta_{k+1,1}^T & \Theta_{k+1,2}^T & \cdots & \Theta_{k+1,N}^T \end{bmatrix}^T, \ \Theta_{k+1,j} \in \Re^{N\times N}, \\ \Delta G_k &= \begin{bmatrix} \Delta G_{k,1}^T & \Delta G_{k,2}^T & \cdots & \Delta G_{k,N}^T \end{bmatrix}^T, \ \Delta G_{k,j} \in \Re^{N\times 1}, j = 1, 2, \cdots, N, \end{aligned}$$

gives

$$\Delta G_{k,j} = \Theta_{k+1,j}\Delta u_k.$$

Since $\Theta_{k+1,j}$ cannot be obtained directly and is variable with k, introduce the correction term $E_{k,j}$ such that $\Theta_{k+1,j} = \Theta_{k,j} + E_{k,j}$ and hence (7) can be written as

$$\Delta G_{k,j} = (\Theta_{k,j} + E_{k,j})\Delta u_k. \tag{8}$$

The problem of ensuring that $\Theta_{k+1,j} \succ 0$ as k increases has been the subject of previous research, see, e.g., [16]. Moreover, the correction term $E_{k,j}$ in (8) must ensure the positive-definite property of $\Theta_{k+1,j}$ and one method is to choose $E_{k,j}$ to be of the form

$$E_{k,j} = \alpha_j \mu_{k,j}\mu_{k,j}^T + \beta_j v_{k,j} v_{k,j}^T, \tag{9}$$

where α_j and β_j are real numbers and $\mu_{k,j}$ and $v_{k,j}$ are vectors to be determined. Substituting (9) into (8) gives

$$(\alpha_j \mu_{k,j}\mu_{k,j}^{\mathrm{T}} + \beta_j v_{k,j} v_{k,j}^{\mathrm{T}})\Delta u_k = \Delta G_{k,j} - \Theta_{k,j}\Delta u_k,$$

whose solution is non-unique for $\mu_{k,j}$ and $v_{k,j}$. Setting (for column vectors of the same dimension), $(v_{k,j}^{\mathrm{T}}\Delta u_k)v_{k,j} = (v_{k,j}v_{k,j}^{\mathrm{T}})\Delta u_k$ gives

$$(\alpha_j \mu_{k,j}^{\mathrm{T}}\Delta u_k)\mu_{k,j} + (\beta_j v_{k,j}^{\mathrm{T}}\Delta u_k)v_{k,j} = \Delta G_{k,j} - \Theta_{k,j}\Delta u_k.$$

Introduce $\mu_{k,j}$ and $v_{k,j}$ as

$$\mu_{k,j} = \chi_j \Theta_{k,j}\Delta u_k, v_{k,j} = \gamma_j \Delta G_{k,j},$$

where the vectors of $\mu_{k,j}$ and $v_{k,j}$ are parallel, respectively, to $\Theta_{k,j}\Delta u_k$ and $\Delta G_{k,j}$ and χ_j and γ_j are parameters to be determined. To solve this problem, (9) can be written as

$$E_{k,j} = \alpha_j \chi_j^2 \Theta_{k,j}\Delta u_k(\Theta_{k,j}\Delta u_k)^T + \beta_j \gamma_j^2 \Delta G_{k,j}(\Delta G_{k,j})^T. \tag{10}$$

Setting

$$\alpha_j \chi_j^2 = -\frac{1}{(\Theta_{k,j}\Delta u_k)^T \Delta u_k},\ \ \beta_j \gamma_j^2 = \frac{1}{(\Delta G_{k,j})^T \Delta u_k},$$

and substituting in (10) gives, after routine manipulations,

$$E_{k,j} = -\frac{\Theta_{k,j}\Delta u_k (\Theta_{k,j}\Delta u_k)^T}{(\Theta_{k,j}\Delta u_k)^T \Delta u_k} + \frac{\Delta G_{k,j}(\Delta G_{k,j})^T}{(\Delta G_{k,j})^T \Delta u_k}.$$

Hence the correction formula for the Hessian matrix $\Theta_{k+1,j}$ in (7) can be written as

$$\Theta_{k+1,j} = \Theta_{k,j} - \frac{\Theta_{k,j}\Delta u_k (\Theta_{k,j}\Delta u_k)^T}{(\Theta_{k,j}\Delta u_k)^T \Delta u_k} + \frac{\Delta G_{k,j}(\Delta G_{k,j})^T}{(\Delta G_{k,j})^T \Delta u_k}, j = 1, 2, \cdots, N. \quad (11)$$

The following theorem is based on a more general result on the positive-definite properties of approximate Hessian matrices (see [6] for further details) to the ILC problem considered and gives conditions for $\Theta_{k+1,j} \succ 0$.

Theorem 1 *Suppose that the matrix $\Theta_{k,j}$ satisfies $\Theta_{k,j} \succ 0$ for any $k \geq 1$ and $j = 1, 2 \ldots, N$. Then $\Theta_{k+1,j} \succ 0$ if and only if $\Delta G_{k,j}^T \Delta u_k > 0$.*

Proof 1 *Using* (7)

$$\Delta G_{k,j}^T \Delta u_k = \Delta u_k^T \Theta_{k+1,j} \Delta u_k,$$

and if $\Theta_{k+1,j} \succ 0$ then $\Delta G_{k,j}^T \Delta u_k > 0$ and necessity is established.

Using (11)*, there exists an arbitrary vector $\zeta_j \neq 0$ such that*

$$\zeta_j^T \Theta_{k+1,j} \zeta_j = \zeta_j^T \Theta_{k,j} \zeta_j - \frac{(\zeta_j^T \Theta_{k,j} \Delta u_k)^2}{(\Theta_{k,j}\Delta u_k)^T \Delta u_k} + \frac{(\zeta_j^T \Delta G_{k,j})^2}{(\Delta G_{k,j})^T \Delta u_k}. \quad (12)$$

Since $\Theta_{k,j} \succ 0$, there exists a matrix $\Theta_{k,j}^{1/2}$ such that $\Theta_{k,j} = \Theta_{k,j}^{1/2}\Theta_{k,j}^{1/2}$. Applying the Cauchy-Schwarz inequality gives

$$\begin{aligned}(\zeta_j^T \Theta_{k,j} \Delta u_k)^2 &= \Big((\Theta_{k,j}^{1/2}\zeta_j)^T(\Theta_{k,j}^{1/2}\Delta u_k)\Big)^2 \\ &\leq \left\|\Theta_{k,j}^{1/2}\zeta_j\right\|^2 \left\|\Theta_{k,j}^{1/2}\Delta u_k\right\|^2 \\ &= (\Theta_{k,j}^{1/2}\zeta_j)^T(\Theta_{k,j}^{1/2}\zeta_j)(\Theta_{k,j}^{1/2}\Delta u_k)^T(\Theta_{k,j}^{1/2}\Delta u_k) \\ &= (\zeta_j^T \Theta_{k,j}\zeta_j)(\Delta u_k^T \Theta_{k,j} \Delta u_k).\end{aligned} \quad (13)$$

The necessary and sufficient condition for $(\zeta_j^T \Theta_{k,j} \Delta u_k)^2 = (\zeta_j^T \Theta_{k,j}\zeta_j)(\Delta u_k^T \Theta_{k,j} \Delta u_k)$ in (13) is that there must exist a nonzero real number $\tau_{k,j}$ such that $\zeta_j = \tau_{k,j}\Delta u_k$. Hence

$$\zeta_j^T \Theta_{k+1,j} \zeta_j = \frac{\tau_{k,j}^2 (\Delta u_k^T \Delta G_{k,j})^2}{(\Delta G_{k,j})^T \Delta u_k} > 0.$$

Otherwise, if (13) *is a strict inequality,* (12) *gives*

$$\begin{aligned}\zeta_j^T\Theta_{k+1,j}\zeta_j >&\zeta_j^T\Theta_{k,j}\zeta_j+\frac{(\zeta_j^T\Delta G_{k,j})^2}{(\Delta G_{k,j})^T\Delta u_k}-\frac{(\zeta_j^T\Theta_{k,j}\zeta_j)(\Delta u_k^T\Theta_{k,j}\Delta u_k)}{(\Theta_{k,j}\Delta u_k)^T\Delta u_k}\\ =&\frac{(\zeta_j^T\Delta G_{k,j})^2}{(\Delta G_{k,j})^T\Delta u_k}>0.\end{aligned}$$

Hence, for an arbitrary vector $\zeta_j \neq 0$, $\zeta_j^{\mathrm{T}}\Theta_{k+1,j}\zeta_j > 0$ and $\Theta_{k+1,j} \succ 0$ and the proof is complete.

Remark 1 *Given Theorem* 1*, if $\Theta_{0,j} \succ 0$, then the matrices $\Theta_{k,j}, k \geq 1$, generated by* (11) *will be positive-definite if $\Delta G_{k,j}^T\Delta u_k > 0$. Otherwise, set $\Theta_{k+1,j} = \Theta_{k,j}$ to ensure $\Theta_{k+1,j} \succ 0$, i.e., the control direction is unchanged and the latest $\Theta_{k,j}$ is used.*

The structure of Θ_{k+1} could lead to a very complicated ILC law design. As an alternative, introduce $M_{k+1} = \left[\begin{array}{cccc} M_{k+1,1}^T & M_{k+1,2}^T & \cdots & M_{k+1,N}^T \end{array}\right]^T \in \Re^{N\times N\times N}$ and write $\hat{\Theta}_{k+1} = M_{k+1} \times \Theta_{k+1} = \sum_{j=1}^{N} M_{k+1,j}\Theta_{k+1,j}$ where $M_{k+1,j} \in \Re^{N\times N}$ are adjustable weighting matrices. Then using these matrices, a candidate ILC law for (2) is

$$\begin{cases} u_{k+1} = u_k + \varepsilon_k W\delta_k,\\ \hat{\Theta}_k\delta_k = e_k,\\ \hat{\Theta}_k = \mathrm{M}_k \times \Theta_k,\\ \Theta_k = \left[\begin{array}{cccc}\Theta_{k,1}^T & \Theta_{k,2}^T & \cdots & \Theta_{k,N}^T\end{array}\right]^T,\\ \Theta_{k+1,j} = \Theta_{k,j} - \frac{\Theta_{k,j}\Delta u_k(\Theta_{k,j}\Delta u_k)^T}{(\Theta_{k,j}\Delta u_k)^T\Delta u_k} + \frac{\Delta G_{k,j}(\Delta G_{k,j})^T}{(\Delta G_{k,j})^T\Delta u_k}, \end{cases} \tag{14}$$

where $\varepsilon_k \in \Re$ is the updating step factor, $W \in \Re^{N\times N}$ is a relaxation matrix, $\delta_k \in \Re^{N\times 1}$ is the search direction and $\hat{\Theta}_k \in \Re^{N\times N}$ is the parameter matrix to be determined. Also ε_k is related to the change in the control input between two successive trials and is therefore an important parameter in the (trial-to-trial error) convergence analysis. Its calculation during optimization procedure is a very important issue and is considered again in Section 4.

The matrix W is introduced to attach weights to the entries in δ_k and then a new search direction can be constructed to speed up the convergence of the correlation elements in search process. Also, if ILC convergence holds then as $k \to \infty$, $\Delta u_k = u_{k+1} - u_k \to 0$ and hence a singularity would occur in computation. In such a case, sufficiently small positive numbers can be added to the denominator terms.

Remark 2 *The ILC law* (14) *requires the computation of the difference between two gradient matrices $\Delta G_{k,j} = G_{k+1,j}(u_{k+1}) - G_{k,j}(u_k)$. Should difficulties arise in this computation one possible approach is to replace $\Delta G_{k,j}$ in the* *last entry by an approximate representation. Suppose, e.g., that the Lipschitz assumptions*

$$\|G_{k+1}(u_{k+1}) - G_k(u_k)\| \leq k_{\nabla g} \|u_{k+1} - u_k\|, \; k_{\nabla g} \geq 0,$$
$$\|g(u_{k+1}) - g(u_k)\| \leq k_g \|u_{k+1} - u_k\|, \; k_g \geq 0$$

hold. Then there may exist $k_\nabla \geq 0$ such that $\|\Delta G_k\| = \|G_k(u_{k+1}) - G_k(u_k)\| = k_\nabla \|g(u_{k+1}) - g(u_k)\|$, where k_∇ is a trial dependent variable. Additionally, the ILC law (14) only uses ΔG_k, where $\Delta G_{k,j} = \theta_j(g(u_{k+1}) - g(u_k)) = \theta_j((y_d - y_k) - (y_d - y_{k+1})) = -\theta_j \Delta e_k$, where θ_j is a candidate scalar that satisfies $|\theta_j| = k_\nabla$ without loss of generality. Hence in this case the approximation is directly in terms of the trial error and potentially difficult computations in constructing the Hessian matrix are avoided. However, this method requires the choice of, in particular, k_∇, which is non-trivial in general. The remainder of this paper assumes the direct computation of the difference between the two gradient matrices is sufficiently accurate, but this area is an obvious candidate for future research.

Remark 3 *In analysis, no loss of generality arises from selecting W as the identity matrix. (The choice of a non-identity matrix for W in an application is left as a topic for further research.)*

4 Constrained optimization

In some practical applications, it will be necessary to impose constraints on some variables, such as the maximum control effort that can be allowed to prevent damage to, e.g., the actuators and other system components. Hence constrained design is required. The design of constrained ILC laws has been the subject of previous research and in this paper the focus is on input constraints. In previous work, e.g., [14], vector inequality constraints of the form

$$\Lambda u_{k+1} \leq \mathrm{B}, \tag{15}$$

where $\Lambda = [I \quad -I]^T \in \Re^{2N \times N}$, $B \in \Re^{2N \times 1}$ have been considered. This paper develops new results for such constraints for an ILC law of the form (14) and design is by minimizing the cost function

$$\min J_{k+1}(\varepsilon_k) = \|e_{k+1}\|^2 + \gamma \varepsilon_k^2, \tag{16}$$

where $\gamma > 0$ is an adjustable parameter and the inclusion of the ε_k term in the cost function is to prevent excessive change in the control input from one trial to the next.

On setting $\Phi(u_{k+1}) = \Lambda u_{k+1} - B$, the corresponding rows in (15) can be written as

$$\Phi_j(u_{k+1}) = \tilde{\alpha}_j^T u_{k+1} - \tilde{\beta}_j \leq 0,$$

where $\tilde{\alpha}_j^T$ and $\tilde{\beta}_j$, respectively, are the jth rows of Λ and B. Moreover, the inequality constraint can be transformed into an equality condition by introducing the auxiliary variable vector $\eta_{k+1} \in \Re^N$, where $\eta_{k+1,j}$ denotes the jth entry of η_{k+1} on trial $k+1$. Hence the problem to be solved is

$$\begin{cases} \min J_{k+1}(\varepsilon_k) = \|e_{k+1}\|^2 + \gamma\varepsilon_k^2, \\ \Phi_j(u_{k+1}) + \eta_{k+1,j}^2 = 0. \end{cases} \tag{17}$$

Other forms of constrained design can arise in applications and those that are applicable or required in each case should be applied. In ILC for robotic-assisted upper limb stroke rehabilitation, the control input [5, 7] is the stimulation applied on each trial to the muscles involved, e.g., triceps for a reaching motion. Moreover, the stimulation is at higher frequency and this, in turn, increases the risk of muscle fatigue and hence the session must end, which can also be demotivating for the patient and in any case approval for clinical tests will set a maximum level of stimulation that can be applied. This is one applications oriented reason for considering input constraints. (See also the further discussion of this area in the conclusions section of this paper).

The generalized augmented Lagrange performance function based on the penalty function and multiplier method for (17) is

$$\begin{aligned} \tilde{J}_{k+1}(\varepsilon_k, \eta_{k+1}, \lambda_k, \sigma) = \|e_{k+1}\|^2 + \gamma\varepsilon_k^2 - \sum_{j=1}^{N} \lambda_{k,j}(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2) \\ + \sum_{j=1}^{N} \frac{\sigma}{2}(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2)^2, \end{aligned}$$

where λ_k denotes the Lagrange multiplier and $\lambda_{k,j}$ denotes the jth entry of λ_k on trial k, $\sigma > 0$ is the penalty factor and the updating algorithm for λ_k is $\lambda_{k+1,j} = \lambda_{k,j} - \sigma\Phi_j(u_k)$. To eliminate $\eta_{k+1,j}$ from $\tilde{J}_{k+1}(u_{k+1}, \eta_{k+1}, \lambda_k, \sigma)$, the first order partial derivative with respect to $\eta_{k+1,j}$ is computed and set equal to zero for a minimum, resulting in

$$\eta_{k+1,j}\left(-\lambda_{k,j} + \sigma\left(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2\right)\right) = 0.$$

Suppose also that $\lambda_{k,j} - \sigma\Phi_j(u_{k+1}) > 0$ and $\eta_{k+1,j}^2 = \frac{\lambda_{k,j}}{\sigma} - \Phi_j(u_{k+1})$, then

$$-\lambda_{k,j}\left(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2\right) + \frac{\sigma}{2}\left(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2\right)^2 = -\frac{\lambda_{k,j}^2}{\sigma} + \frac{\lambda_{k,j}^2}{2\sigma} = \frac{1}{2\sigma}(-\lambda_{k,j}^2), \tag{18}$$

otherwise, if $\lambda_{k,j} - \sigma\Phi_j(u_{k+1}) \leq 0$ then $\eta_{k+1,j}^2 = 0$. Hence

$$\begin{aligned} -\lambda_{k,j}(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2) + \frac{\sigma}{2}(\Phi_j(u_{k+1}) + \eta_{k+1,j}^2)^2 =& -\lambda_{k,j}\Phi_j(u_{k+1}) + \frac{\sigma}{2}(\Phi_j(u_{k+1}))^2 \\ =& \frac{1}{2\sigma}\left((\sigma\Phi_j(u_{k+1}) - \lambda_{k,j})^2 - \lambda_{k,j}^2\right), \end{aligned} \tag{19}$$

and on combining (18) and (19)

$$\begin{aligned} & -\lambda_{k,j}(\Phi_j(u_{k+1})+\eta_{k+1,j}^2)+\frac{\sigma}{2}(\Phi_j(u_{k+1})+\eta_{k+1,j}^2)^2 \\ =& \frac{1}{2\sigma}\left\{\min\left[0,(\sigma\Phi_j(u_{k+1})-\lambda_{k,j})\right]^2-\lambda_{k,j}^2\right\}. \end{aligned} \tag{20}$$

Substituting (20) into the generalized augmented Lagrange performance function $\tilde{J}_{k+1}(\varepsilon_k,\eta_{k+1},\lambda_k,\sigma)$ gives

$$\tilde{J}_{k+1}(\varepsilon_k,\lambda_k,\sigma)=\|e_{k+1}\|^2+\gamma\varepsilon_k^2+\frac{1}{2\sigma}\sum_{j=1}^{N}\left\{[\min\{0,(\sigma\Phi_j(u_{k+1})-\lambda_{k,j})\}]^2-\lambda_{k,j}^2\right\}. \tag{21}$$

The updating step factor ε_k in (14) can be determined by minimizing the performance function (21). Hence substitute (14) into (21) and set

$$\frac{\partial\tilde{J}_{k+1}(\varepsilon_k,\lambda_k,\sigma)}{\partial\varepsilon_k}=0.$$

Also it is immediate on subtracting y_d (the reference signal) from both sides of (5) that

$$y_k-y_d\approx y_{k+1}-y_d-G_{k+1}(u_{k+1})\Delta u_k+\frac{1}{2}\Im(\Delta u_k)$$

and introducing

$$\tilde{G}_k=G_{k+1}-\frac{1}{2}\Delta u_k^TH_{k+1}=G_k+H_{k+1}\Delta u_k-\frac{1}{2}\Delta u_k^TH_{k+1}\approx G_k+\Theta_{k+1}\Delta u_k-\frac{1}{2}\Delta u_k^T\Theta_{k+1},$$

gives

$$e_{k+1}=e_k-\tilde{G}_k(u_{k+1}-u_k),$$

where $e_k=y_d-y_k$. Moreover, using $u_{k+1}=u_k+\varepsilon_kW\delta_k$ gives

$$e_{k+1}=\left(I-\tilde{G}_k\varepsilon_kW\hat{\Theta}_k^{-1}\right)e_k. \tag{22}$$

The following two cases arise from the analysis above.

Case 1: If $\lambda_{k,j}-\sigma\Phi_j(u_{k+1})>0$, then

$$\frac{\partial\tilde{J}_{k+1}(\varepsilon_k,\lambda_k,\sigma)}{\partial\varepsilon_k}=-2e_k^T(I-\tilde{G}_k\varepsilon_kW\hat{\Theta}_k^{-1})(\tilde{G}_kW\hat{\Theta}_k^{-1})e_k+2\gamma\varepsilon_k=0,$$

or

$$\varepsilon_k^*=\frac{e_k^T\left(\tilde{G}_kW\hat{\Theta}_k^{-1}+\left(\tilde{G}_kW\hat{\Theta}_k^{-1}\right)^T\right)e_k}{2\left(\gamma+\left\|\tilde{G}_kW\hat{\Theta}_k^{-1}e_k\right\|^2\right)}. \tag{23}$$

Case 2: If $\lambda_{k,j} - \sigma\Phi_j(u_{k+1}) \leq 0$, then

$$\begin{aligned}\frac{\partial \tilde{J}_{k+1}(\varepsilon_k, \lambda_k, \sigma)}{\partial \varepsilon_k} = &- 2e_k^T \left(I - \tilde{G}_k\varepsilon_k W\hat{\Theta}_k^{-1}\right)\left(\tilde{G}_k W\hat{\Theta}_k^{-1}\right) e_k + 2\gamma\varepsilon_k \\ &+ \varepsilon_k\sigma\sum_{j=1}^{N}\left\|\tilde{\alpha}_j^T W\hat{\Theta}_k^{-1}e_k\right\|^2 + \sigma\sum_{j=1}^{N}\tilde{\alpha}_j^T W\hat{\Theta}_k^{-1}e_k(\tilde{\alpha}_j^T u_k) \\ &- \sum_{j=1}^{N}\tilde{\alpha}_j^T W\hat{\Theta}_k^{-1}e_k\sigma\tilde{\beta}_j + \lambda_{k,j} = 0,\end{aligned}$$

or

$$\varepsilon_k^* = \frac{e_k^T N_k e_k + \sum\limits_{j=1}^{N}\tilde{\alpha}_j^T W\delta_k\sigma\tilde{\beta}_j + \lambda_{k,j}) - \sum\limits_{j=1}^{N}\sigma\tilde{\alpha}_j^T W\delta_k(\tilde{\alpha}_j^T u_k)}{2\left(\gamma + \left\|\tilde{G}_k W\Theta_k^{-1}e_k\right\|^2\right) + \sum\limits_{j=1}^{N}\sigma\left\|\tilde{\alpha}_j^T W\delta_k\right\|^2}, \tag{24}$$

where $N_k = \tilde{G}_k W\hat{\Theta}_k^{-1} + (\tilde{G}_k W\hat{\Theta}_k^{-1})^T$. Further routine manipulations give

$$\varepsilon_k^* = \frac{e_k^T N_k e_k + (\sigma B + \lambda_k)^T \Lambda W\delta_k - \sigma(\Lambda u_k)^T\Lambda W\delta_k}{2\left(\gamma + \left\|\tilde{G}_k W\hat{\Theta}_k^{-1}e_k\right\|^2\right) + \sigma(\Lambda W\delta_k)^T(\Lambda W\delta_k)}. \tag{25}$$

Substituting the optimal updating step factor ε_k^* from (23) or (25) into (14) gives the ILC law

$$u_{k+1} = u_k + \varepsilon_k^* W\delta_k. \tag{26}$$

5 Convergence analysis

This section analyzes monotonic trial -to-trial error norm reduction for the system obtained by applying the ILC law developed in the previous section. In applications terms, the requirement is to ensure that the tracking accuracy is improved after each trial is complete. This requirement can be stated in terms of the Euclidean norm as the requirement that the tracking error decreases monotonically for each trial, i.e.

$$\|e_{k+1}\| < \|e_k\|, \;\; \forall k > 0. \tag{27}$$

To obtain a condition for this property, consider the system resulting from application of (26) to (2) and hence

$$e_{k+1} = \left(I - \tilde{G}_k\varepsilon_k^* W\hat{\Theta}_k^{-1}\right)e_k. \tag{28}$$

Then the following result establishes how the updating step factor ε_k affects trial-to-trial error convergence.

Theorem 2 *Consider a discrete nonlinear system described by* (1) *in the presence of input constraints defined by* (15)*. Suppose also that the ILC law of the form* (26) *computed using (14) and (21) is applied. Then the resulting controlled system has monotonic trial-to-trial error convergence, i.e,* (27) *is satisfied, if*

$$\lambda_{k,j} - \sigma \Phi_j(u_{k+1}) > 0, \tag{29}$$

holds. If this condition is not satisfied, monotonic trial-to-trial error convergence occurs if both the Lagrange multiplier $\lambda_{k,j}$ *and the penalty factor* σ *satisfy the following constraint*

$$\tilde{\alpha}_j^T u_k - \tilde{\beta}_j - \frac{\lambda_{k,j}}{\sigma} = 0, \ \forall k > 0, \ j = 1, 2, \dots N.$$

Proof 2 *Assume that* (29) *holds. Then on setting* $\gamma > 0$ *as required by* (16)*, the following inequality holds*

$$\|e_{k+1}\|^2 < \|e_{k+1}\|^2 + \gamma {\varepsilon_k^*}^2. \tag{30}$$

Also using (28), (30) *can be rewritten as*

$$\|e_{k+1}\|^2 < \|e_k\|^2 + {\varepsilon_k^*}^2 \left(\gamma + \left\|\tilde{G}_k W \hat{\Theta}_k^{-1} e_k\right\|^2\right) - e_k^T \varepsilon_k^* \left(\tilde{G}_k W \hat{\Theta}_k^{-1} + \left(\tilde{G}_k W \hat{\Theta}_k^{-1}\right)^T\right) e_k$$

and substituting (23) *gives*

$$\|e_{k+1}\|^2 < \|e_k\|^2 - \frac{\left[e_k^T \left(\tilde{G}_k W \hat{\Theta}_k^{-1} + \left(\tilde{G}_k W \hat{\Theta}_k^{-1}\right)^T\right) e_k\right]^2}{4\left(\gamma + \left\|\tilde{G}_k W \hat{\Theta}_k^{-1} e_k\right\|^2\right)} < \|e_k\|^2. \tag{31}$$

Moreover, this last inequality immediately ensures that (27) *holds and hence monotonic trial-to-trial error convergence occurs. Otherwise, if* $\lambda_{k,j} - \sigma\Phi_j(u_{k+1}) \leq 0$,

$$\|e_{k+1}\|^2 < \|e_{k+1}\|^2 + \gamma {\varepsilon_k^*}^2 + \sum_{j=1}^{N} \frac{1}{2\sigma} \left[(\sigma \Phi_j(u_{k+1}) - \lambda_{k,j})^2\right],$$

and hence

$$\begin{aligned}\|e_{k+1}\|^2 < \|e_k\|^2 + {\varepsilon_k^*}^2 &\left(\gamma + \left\|\tilde{G}_k W \hat{\Theta}_k^{-1} e_k\right\|^2 + \frac{\sigma}{2}\sum_{j=1}^{N} \left\|\tilde{\alpha}_j^T W \hat{\Theta}_k^{-1} e_k\right\|^2\right) \\ &- \varepsilon_k^* \left[e_k^T N_k e_k + \sum_{j=1}^{N} \tilde{\alpha}_j^T W \delta_k (\sigma \tilde{\beta}_j + \lambda_{k,j}) - \sum_{j=1}^{N} \sigma \tilde{\alpha}_j^T W \delta_k (\tilde{\alpha}_j^T u_k)\right] \\ &+ \frac{\sigma}{2} \sum_{j=1}^{N} \left[\left(\tilde{\alpha}_j^T u_k - \tilde{\beta}_j - \frac{\lambda_{k,j}}{\sigma}\right)^2\right]\end{aligned}$$

and substituting from (24) *gives*

$$\|e_{k+1}\|^2 < \|e_k\|^2 + \frac{\sigma}{2}\sum_{j=1}^{N}\left[\left(\tilde{\alpha}_j^T u_k - \tilde{\beta}_j - \frac{\lambda_{k,j}}{\sigma}\right)^2\right]$$

$$- \frac{\left[e_k^T N_k e_k + \sum\limits_{j=1}^{N} \tilde{\alpha}_j^T W \hat{\Theta}_k^{-1} e_k \left(\sigma\tilde{\beta}_j + \lambda_{k,j}\right) - \sum\limits_{j=1}^{N} \sigma \tilde{\alpha}_j^T W \hat{\Theta}_k^{-1} e_k (\tilde{\alpha}_j^T u_k)\right]^2}{4\left(\gamma + \left\|\tilde{G}_k W \hat{\Theta}_k^{-1} e_k\right\|^2\right) + 2\sum\limits_{j=1}^{N} \sigma \left\|\tilde{\alpha}_j^T W \hat{\Theta}_k^{-1} e_k\right\|^2}.$$

This last inequality implies that in both cases, i.e., $\tilde{\alpha}_j^T u_k - \tilde{\beta}_j - \frac{\lambda_{k,j}}{\sigma} = 0$ *and* $\tilde{\alpha}_j^T u_k - \tilde{\beta}_j - \frac{\lambda_{k,j}}{\sigma} \neq 0$*, monotonic trial-to-trial convergence property defined by* (27) *is satisfied if the sum of the last two terms on the right-hand side of this inequality is negative. This completes the proof.*

Remark 4 *Other parameters in the ILC law can also be optimized, e.g., the matrix* W*, but in this paper the interest is trial-to-trial error convergence and these other cases are left as areas for possible future research.*

Theorem 2 establishes that the developed ILC law considered has monotonic trial-to-trial error convergence. The analysis given next shows that this ILC law also exhibits super-linear trial-to-trial convergence, i.e., if the sequence of trial control inputs converges to the learned control u_∞ then $\lim\limits_{k\to\infty} \frac{\|u_{k+n+1}-u_\infty\|}{\|u_k-u_\infty\|} = 0, n \geq 0$. Hence this modified Newton method based ILC law can be considered as a special case of the BFGS algorithm and therefore it inherits the properties of this last algorithm, including the super-linear convergence property. In particular, suppose that the inputs u_k are generated by (14) and the following conditions hold:

1) the input sequence of u_k does not terminate, but remains in a closed and bounded, convex set where the function $g(u_k)$ is twice continuously differentiable and $g(u_k)$ has a unique stationary point u_∞;

2) the sequence of Hessian matrices $\{H_{\infty,j}(u_\infty), j = 1, 2, \ldots, N\}$ are positive-definite and $\{H_{k,j}(u_k), j = 1, 2, \ldots, N\}$ is Lipschitz continuous in a neighborhood of u_∞, i.e., $\forall u_k \in \Upsilon(u_\infty, \delta)$;

3) the matrix sequence $\{\Theta_{k,j}\}$ is bounded in norm;

4) the condition $\left|\left|\Delta u_k^T \left(\Delta G_{k,j} - \Theta_{k,j}\Delta u_k\right)\right|\right| \geq r \|\Delta u_k\| \|\Delta G_{k,j} - \Theta_{k,j}\Delta u_k\|$ holds for each trial, where the constant $r \in (0, 1)$.

Then $\lim\limits_{k\to\infty} u_k = u_\infty$ and

$$\lim_{k\to\infty} \frac{\|u_{k+n+1} - u_\infty\|}{\|u_k - u_\infty\|} = 0.$$

Hence the control input sequence $\{u_k\}$ converges to u_∞ at an $(n+1)$-step super-linear convergence rate, for a detailed treatment (in the non ILC case) see [17]. Moreover, it is important to emphasize that super-linear convergence means that the new ILC law can deliver faster trial-to-trial error convergence rate over an alternative that can only achieve the monotonic property.

In summary, the ILC law developed in this paper can be implemented as follows:

Step 1. For the given application, select the initial state $x_k(0)$, the initial input u_0, reference trajectory $y_{\rm d}$, the number of samples along a trial N, the penalty factor σ and the initial Lagrange multiplier λ_0.

Step 2. Specify the maximum output tracking error $\xi_{\max}$ and a maximum number of trials to be completed $k_{\max}$.

Step 3. Given y_k and u_k, use (21) to compute ε_k^* and hence given W and γ, the control input for the next trial.

Step 4. If the tracking error performance on trial $k+1$ is less than $\xi_{\max}$ the ILC law has achieved the performance specification. If not, return to Step 3 and repeat. If the trial number is $k_{\max}$, the procedure ends.

Due to possible inherent NP-hardness of the problem, there is no theoretical way to ensure convergence to a global minimum in this problem. Furthermore, in some cases the desired performance specification on the tracking error cannot be achieved even when the number of possible trials tends to infinity. However, the numerical experiments carried out so far may imply that this new approach is competitive (in terms of the convergence rate) among others that admit gradient evaluation.

6 Simulation case study

To demonstrate the properties and performance of the new ILC law developed in this paper, a simulation case study is given. This study is based on an available model of a Continuous Stirred Tank Reactor (CSTR) where the reaction temperature needs to be controlled. The model of the dynamics under some reaction conditions and the reaction mechanism is based on that given in [15]. Applying the Euler discretization method with sampling time $T = 1$ minute (the chemical reaction is a slowly varying dynamic process) to the governing differential equation results in the following nonlinear discrete model as the basis for this study.

$$
\begin{aligned}
x_k^1(p+1) &= \left(0.2 - 7.2\cdot 10^{10} e^{\frac{-10^4}{x_k^2(p)}}\right) x_k^1(p) + 0.8,\\
x_k^2(p+1) &= \left(-0.8 - e^{-7}\right) x_k^2(p) + \left(1.44\cdot 10^{12} e^{\frac{-10^4}{x_k^2(p)}}\right) x_k^1(p) + \left(1 - e^{-7}\right) u_k(p) + 280,\\
y_k(p) &= x_k^2(p).
\end{aligned}
$$

It is assumed that this CSTR system needs to execute the same reaction task over $0 \leq p \leq 720$ minutes and the reference trajectory is specified as

$$y_{\mathrm{d}}(p) = \begin{cases} 0.2p + 423, & 0 \leq p \leq 250, \\ \frac{1}{7}p + \frac{3061}{7}, & 250 \leq p \leq 320, \\ 483, & 320 \leq p \leq 600, \\ -\frac{19}{60}p + 673, & 600 \leq p \leq 720, \end{cases}$$

see Fig.1, which is a piecewise function modeling four stages of the material temperature control in the reaction process. Also $x_k^1(0) = 1$, $x_k^2(0) = 423$, the initial control input is $u_0(p) = 293$, $0 \leq p \leq 720$, the penalty factor $\sigma = 0.1$, the initial Lagrange multiplier $\lambda_0 = [\ 20 \quad 20 \quad \cdots \quad 20\ ]^T$. The control input constraint is defined by $B = [\ 378, \quad 378, \quad \cdots \quad 378, \quad 273, \quad 273, \quad \cdots \quad 273\ ]^T$ and $\Lambda = [I, -I]^T$. Moreover, to evaluate tracking performance from trial-to-trial, the convergence performance index is the root mean square (RMS) along the trial, i.e.,

$$RMS(k) = \sqrt{\frac{1}{721}\sum_{p=0}^{721} e_k^2(p)},\ k = 0, 1, \cdots, 30.$$

Setting the maximum tracking RMS error $\xi_{\max} = 5.1 \cdot 10^{-3}$, the maximum number of trials $k_{\max} = 30$, $W = I$, $\gamma = 0.0636$, the CSTR system temperature output tracking curve and RMS error curve generated with the ILC law (14) applied are shown, respectively, in Figs. 1 and 2. To compare the trial-to-trial error convergence speed and accuracy of the new ILC law, the Newton-type ILC algorithm used in [8] was applied to this example, giving

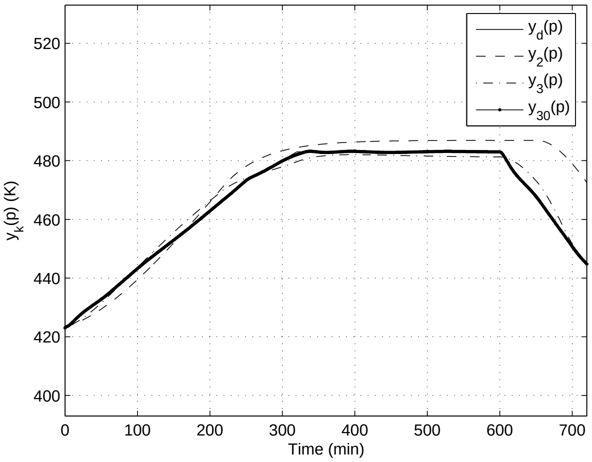

Figure 1: The desired temperature output and tracking curve

$$u_{k+1} = u_k + G_k^{-1} e_k,\ G_k^{-1} = (1 - e^{-7})^{-1}.$$

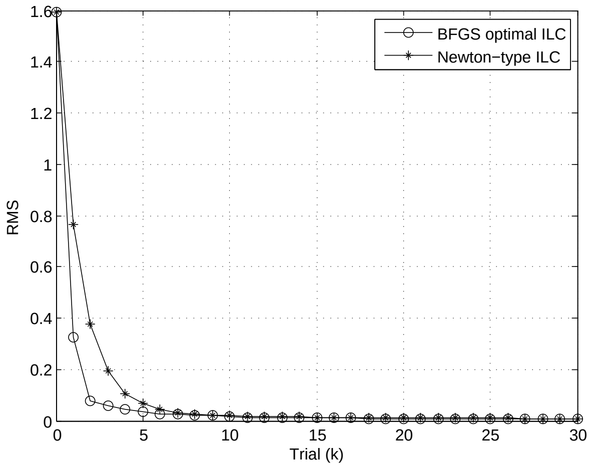

Figure 2: The RMS error curve along the trial axis

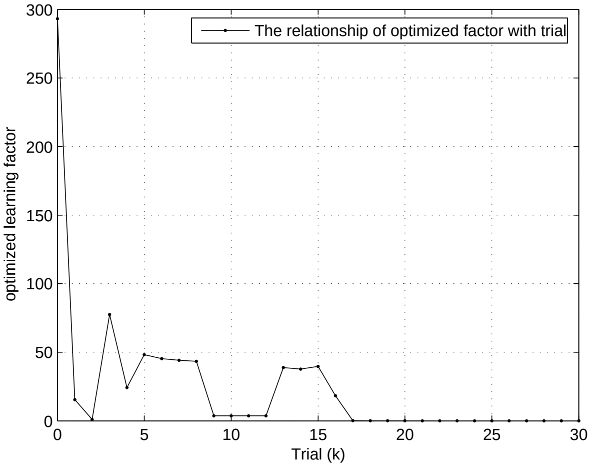

Figure 3: The optimized updating step factor ε_k^* along the trial axis

The results obtained with this law are also shown in Fig. 2 and confirm that the new ILC law outperforms the basic Newton-type ILC law. These plots also confirm that the tracking error is monotonically convergent from trial-to-trial even under the input constraint. Fig. 3 shows that the optimized learning factor ε_k gradually approaches zero. Fig. 4 shows the control inputs for both designs and Fig. 5 the cost function for the constrained design.

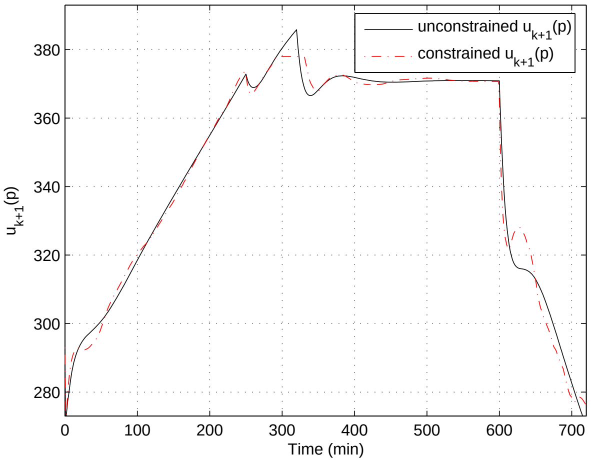

Figure 4: The control input for the unconstrained and constrained designs

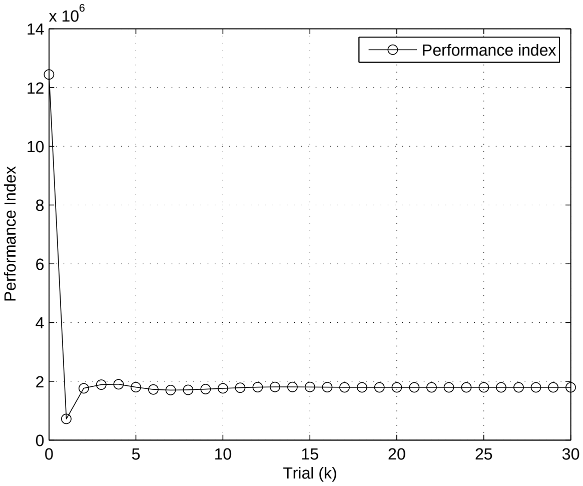

Figure 5: The cost function variation from trial-to-trial for the constrained design

7 Conclusions

This paper has developed a new ILC law for discrete nonlinear systems, including the case of input constraints, based on a modified Newton method. The new design is based on the use of a penalty function and an iterative method for solving an unconstrained nonlinear optimization problem with an algorithm that has monotonic and super linear convergence characteristics, where the input inequality constraints are transformed into equality form by adding auxiliary variables. The super-linear convergence property means that the convergence speed increases rapidly once the input is close enough to the learned control. This explains why this design can converge faster than alternatives, as supported by the example in the previous section where the alternative is standard Newton based ILC. This example also confirms better convergence speed over standard the Newton ILC design due to the super-convergence property.

Topics for future research include a detailed investigation into the optimization of the other parameters in the final control law, i.e., the relaxation matrix W and the search direction δ_k. Also further research is required to extend the constrained design to include other forms of constraints on variables, e.g. the state and output vectors. The extension to multiple-input multiple-output systems, including constraints, should also be investigated. Moreover, refinements of the BFGS algorithm reported in the literature should also be considered for use in ILC design.

References

[1] S. Arimoto, S. Kawamura, F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1984, **1**(2), 123–140.

[2] D. A. Bristow, M. Tharayil, A. Alleyne. A survey of Iterative Learning Control, *IEEE Control Systems Magazine*, 2006, **26(3)**, 96–114.

[3] H. S. Ahn, Y. Q. Chen, K. L. Moore. Iterative learning control: brief survey and categorization, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 2007, **37(6)**, 1109–1121.

[4] I. Lim, D. J. Hoelzle, K. L. Barton. A multi-objective iterative learning control approach for additive manufacturing applications, *Control Engineering Practice*, 2017, **64(11)**, 74–87.

[5] C. T. Freeman, A. M Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, E. Rogers. Iterative learning control of FES applied to the upper extremity for rehabilitation, *Control Engineering Practice*, 2009, **17(3)**, 368–381.

[6] D.G. Luenberger, Y. Yinyu. Linear and Nonlinear Programming, 2015, Springer Publishing Company.

[7] K. L. Meadmore, T. Exell, E. Hallewell, A. M Hughes, C. T. Freeman, M. Kutlu, V. Benson, E.Rogers, J. H. Burridge. The application of precisely controlled functional electrical stimulation to the shoulder, elbow and wrist for upper limb stroke rehabilitation: a feasibility study, *Journal of NeuroEngineering and Rehabilitation*, 2014, **11(105)**.

[8] C.T. Freeman. Newton method based iterative learning control for robot assisted rehabilitation using FES. *Mechatronics*, 2014, **24**(8), 934–943.

[9] D.H. Owens, B. Chu. Combined inverse and gradient iterative learning control: performance, monotonicity, robustness and non-minimum-phase zeros, *International Journal of Robust and Nonlinear Control*, 2014, **24**(3), 406–431.

[10] G.L. Yuan, S.W. Yao. A BFGS algorithm for solving symmetric nonlinear equations. *Optimization*, 2013, **62**(1), 85–99.

[11] X.X. Huang. Convergence of a class of penalty methods for constrained scalar set-valued optimization. *Journal of Global Optimization*, 2013, **56**(1), 1501–1513.

[12] T. Lin, D.H. Owens, J. Hatonen. Newton method based iterative learning control for discrete nonlinear systems, *Iernational Journal of Control*, 2006, **79**(10), 1263–1276.

[13] J.M. Ortega, W.C. Rheinboldt. Iterative Solution of a Nonlinear Equation in Several Variables, 1970, London: Academic Press.

[14] C.T. Freeman, Y. Tan. Iterative Learning Control With Mixed Constraints for Point-to-Point Tracking. *IEEE Transactions on Control Systems Technology*, 2013, **21**(3), 604–616.

[15] L. Magni, G.D. Nicolao, L. Magnani, R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 2001, **37**(9), 1351–1362.

[16] Ž. Povalej. Quasi-Newton's method for multiobjective optimization. *Journal of Computational and Applied Mathematics*, 2014, **255**, 765–777.

[17] J. Nocedal, S.J. Wright. Numerical optimization (second edition), 2006, London: Springer.