2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

# Point-to-Point Iterative Learning Control with Mixed Constraints

Chris Freeman and Ying Tan

***Abstract*— Iterative learning control is concerned with tracking a reference trajectory defined over a finite time duration, and is applied to systems which perform this action repeatedly. In this paper iterative learning schemes are developed to address the case in which the output is only critical at certain time instants. This freedom makes it possible to incorporate both hard and soft constraints into the control scheme. Experimental results confirm practically and performance.**

## I. INTRODUCTION

Iterative Learning Control (ILC) is applicable to systems which repeatedly track a reference, $\mathbf{y}_d(t)$, defined over a finite interval $0 \leq t \leq T$. In many applications, however, the goal is to repeatedly follow a motion profile in which the system output is only critical at a finite set of prescribed time instants. Examples include production line automation, crane positioning, and robotic 'pick and place' tasks. Instead of tracking $\mathbf{y}_d(t)$, strategies for point-to-point ILC typically involve the generation of a more suitable motion profile, $\hat{\mathbf{y}}_d(t)$, in advance. The standard ILC framework is clearly able to tackle such problems simply by using $\hat{\mathbf{y}}_d(t)$. Since tracking performance at times between those important time instants is not specified, extra freedom can be gained in ILC design, allowing additional performance demands to be addressed. In particular, when soft and hard constraints are considered, point-to-point ILC may lead to a feasible solution while the original system is infeasible for those constraints. In [1], [2], [3] ILC was employed in point-to-point motion control to suppress residual vibrations, but these involved design of a static reference over $0 \leq t \leq T$. In contrast, Terminal ILC required only an end-point demand, employing a time-invariant input vector [4]. These approaches, however, are formulated for a single point-to-point movement, as opposed to a multiple sequence of point-to-point movements. Moreover, they cannot simultaneously address general forms of objective function, such as minimizing the norm of the input, output or states, or their derivatives. These drawbacks were overcome in [5] where an approach to multiple point-to-point ILC was developed. Here tracking was only required over an arbitrary set of time points, with the reference comprising the corresponding output values to be tracked. This simplified design and implementation, and allowed soft constraints to be specified in response to performance requirements. However, hard constraints are generally more critical and arise due to actuator saturation, physical limitations, or imposed safety restrictions. This paper therefore focuses on hard input constraints and shows how they can be addressed in the ILC point-to-point framework, either alone, or in combination with soft constraints.

C. Freeman is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. cf@ecs.soton.ac.uk. Y. Tan is with the Electrical and Electronic Engineering Department, University of Melbourne, Parkville, VIC 3010, Australia. y.tan@ee.unimelb.edu.au

## II. PROBLEM FORMULATION

The set of real numbers is denoted as $\mathbb{R}$, and the set of integers as $\mathbb{N}$. The symbol $k$ denotes the trial number and $k \in \mathbb{N}_{\geq 0}$. For any vector $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$. For any matrix $A \in \mathbb{R}^{n\times n}$, $\|A\|_2$ is the induced norm of the vector norm. Consider the following nonlinear discrete-time system

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \\ \mathbf{y}(t) &= \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \qquad \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{1}$$

defined over the finite time interval $t \in [0, 1, 2, \dots, N-1]$. Here $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are continuously differentiable, and $\mathbf{x}(\cdot) \in \mathbb{R}^n$, $\mathbf{u}(\cdot) \in \mathbb{R}^m$, $\mathbf{y}(\cdot) \in \mathbb{R}^p$ are the state, input and output vectors respectively. The input and output sequences are given by

$$\begin{aligned} \mathbf{u} &= [\mathbf{u}(0)^T, \mathbf{u}(1)^T, \dots, \mathbf{u}(N-1)^T]^T \quad \in \mathbb{R}^{mN} \\ \mathbf{y} &= [\mathbf{y}(0)^T, \mathbf{y}(1)^T, \dots, \mathbf{y}(N-1)^T]^T \quad \in \mathbb{R}^{pN} \end{aligned}$$

The standard ILC framework constructs a series of inputs which drives the system to track a reference sequence

$$\mathbf{y}_d = [\mathbf{y}_d(0)^T, \mathbf{y}_d(1)^T, \dots, \mathbf{y}_d(N-1)^T]^T \quad \in \mathbb{R}^{pN}$$

Let $\mathbf{u}_k$ and $\mathbf{y}_k$ be the input and output vectors respectively on the $k^{th}$ trial, with $\mathbf{e}_k = \mathbf{y}_d - \mathbf{y}_k$ the tracking error. Then it is necessary to find a sequence of control inputs that satisfies

$$\lim_{k\to\infty} \|\mathbf{e}_k\|_2 = 0, \qquad \lim_{k\to\infty} \|\mathbf{u}_k - \mathbf{u}_d\|_2 = 0$$

where $\mathbf{u}_d$ is the desired input signal corresponding to $\mathbf{y}_d$. Over the $k^{th}$ trial the relationship between the input and output time-series is expressed by the algebraic functions

$$\begin{aligned} \mathbf{y}_k(0) &= \mathbf{h}(\mathbf{x}_k(0), \mathbf{u}_k(0)) = \mathbf{g}_0(\mathbf{x}_k(0), \mathbf{u}_k(0)) \\ \mathbf{y}_k(1) &= \mathbf{h}(\mathbf{x}_k(1), \mathbf{u}_k(1)) = \mathbf{h}\left(\mathbf{f}(\mathbf{x}_k(0), \mathbf{u}_k(0)), \mathbf{u}_k(1)\right) \\ &= \mathbf{g}_1\left(\mathbf{x}_k(0), \mathbf{u}_k(0), \mathbf{u}_k(1)\right) \\ &\vdots \\ \mathbf{y}_k(N-1) &= \mathbf{h}(\mathbf{x}_k(N-1), \mathbf{u}_k(N-1)) \\ &= \mathbf{h}\left(\mathbf{f}(\mathbf{x}_k(N-2), \mathbf{u}_k(N-2)), \mathbf{u}_k(N-1)\right) \\ &= \mathbf{g}_{N-1}(\mathbf{x}_k(0), \mathbf{u}_k(0), \mathbf{u}_k(1), \dots, \mathbf{u}_k(N-1)) \end{aligned} \tag{2}$$

This allows a precise connection to be made between ILC and techniques from nonlinear optimization employed subsequently. Since $\mathbf{x}_k(0) = \mathbf{x}_0$, using the relations (2), the system (1) can be represented by the algebraic function $\mathbf{g}(\cdot) : \mathbb{R}^{mN} \to \mathbb{R}^{pN}$ in the following form

$$\mathbf{y}_k = \mathbf{g}(\mathbf{u}_k),\ \mathbf{g}(\cdot) = [\mathbf{g}_0(\cdot)^T, \mathbf{g}_1(\cdot)^T, \dots, \mathbf{g}_{N-1}(\cdot)^T]^T. \tag{3}$$

978-1-4577-0079-8/11/$26.00 ©2011 AACC 

In the point-to-point problem the plant output is specified at a fixed number, $M \leq N$, of sample instants given by $1 \leq n_1 < n_2 < \cdots < n_M \leq N$. Let the prescribed values of the output at these instants be $\mathbf{y}_r(0), \mathbf{y}_r(1), \ldots, \mathbf{y}_r(M-1)$, where $\mathbf{y}_r(i) \in \mathbb{R}^p$. ILC can be considered an iterative numerical solution to the problem of finding a control input which minimizes the point-to-point error norm

$$\min_{\mathbf{u}} J(\mathbf{u}), \qquad J(\mathbf{u}) = \|\mathbf{y}_r - \Phi \mathbf{g}(\mathbf{u})\|_2^2 \tag{4}$$

where the $pM \times pN$ matrix $\Phi$ has block-wise components

$$\Phi_{i,j} = \begin{cases} I_p & j = n_i, \quad i = 1, 2 \ldots M \\ 0_p & \text{otherwise} \end{cases} \tag{5}$$

where $I_p$ and $0_p$ are the $p \times p$ identity and zero matrices respectively, and

$$\mathbf{y}_r = \left[\mathbf{y}_r(0)^T, \mathbf{y}_r(1)^T, \cdots, \mathbf{y}_r(M-1)^T\right]^T \quad \in \mathbb{R}^{pM}. \tag{6}$$

The control objective is to find a sequence of control inputs $\{\mathbf{u}_k\}$ such that $\lim_{k \to \infty} \Phi \mathbf{g}(\mathbf{u}_k) = \mathbf{y}_r$, thus minimizing the optimization criterion (4).

As $\mathbf{g}(\cdot)$ is a static mapping, the design of ILC becomes a special case of numerical analysis in which iterative updating laws are employed to solve a nonlinear equation. Many numerical algorithms are available (see [6]) to solve such a problem, and in this paper the gradient descent and Newton methods are used. Standard approaches will also be utilised to address additional inequality and equality constraints.

## III. GRADIENT DESCENT POINT-TO-POINT ILC

The gradient descent method is frequently used to tackle optimization problems of the form of (4), and in this case the iterative update is

$$\begin{aligned} \mathbf{u}_{k+1} &= \mathbf{u}_k - \frac{\beta}{2} \nabla_{\mathbf{u}} J(\mathbf{u}_k) \\ &= \mathbf{u}_k + \beta \left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k) \end{aligned} \tag{7}$$

Since $\left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k)$ is a descent direction for problem (4), there exists a scalar gain $\beta > 0$ which guarantees reduction of the error norm. The rigid connection established between the gradient descent method and ILC ensures that the algorithm has the local property of a linear convergence rate to zero error. The derivative $\mathbf{g}'(\mathbf{u}_k)$ is equivalent to the system linearisation around $\mathbf{u}_k$ and is represented by the $pN \times mN$ matrix

$$\mathbf{g}'(\mathbf{u}_k) = \begin{bmatrix} \frac{\partial \mathbf{g}_0}{\partial \mathbf{u}_k(0)} & \frac{\partial \mathbf{g}_0}{\partial \mathbf{u}_k(1)} & \cdots & \frac{\partial \mathbf{g}_0}{\partial \mathbf{u}_k(N-1)} \\ \frac{\partial \mathbf{g}_1}{\partial \mathbf{u}_k(0)} & \frac{\partial \mathbf{g}_1}{\partial \mathbf{u}_k(1)} & \cdots & \frac{\partial \mathbf{g}_1}{\partial \mathbf{u}_k(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{N-1}}{\partial \mathbf{u}_k(0)} & \frac{\partial \mathbf{g}_N}{\partial \mathbf{u}_k(1)} & \cdots & \frac{\partial \mathbf{g}_{N-1}}{\partial \mathbf{u}_k(N-1)} \end{bmatrix} \tag{8}$$

since $\mathbf{g}_i(\cdot)$ is not a function of $\mathbf{u}_k(j), \quad j > i$, the upper triangular elements are zero.

*Remark 1:* The linearized system $\tilde{\mathbf{y}} = \mathbf{g}'(\mathbf{u}_k)\tilde{\mathbf{u}}$ corresponds to the linear time-varying system

$$\begin{aligned} \tilde{\mathbf{x}}(t+1) &= A(t)\tilde{\mathbf{x}}(t) + B(t)\tilde{\mathbf{u}}_k(t) \\ \tilde{\mathbf{y}}(t) &= C(t)\tilde{\mathbf{x}}(t) + D(t)\tilde{\mathbf{u}}_k(t) \qquad t = 0, 1, \ldots, N-1 \end{aligned} \tag{9}$$

with

$$\begin{aligned} A(t) &= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)}, & B(t) &= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)} \\ C(t) &= \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)}, & D(t) &= \left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)} \end{aligned} \tag{10}$$

The term $\left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k)$ in (7) can be efficiently generated using the co-state representation of system (9). More specifically, it is equal to the output $\tilde{\mathbf{y}}$ of the system

$$\begin{aligned} \tilde{\mathbf{x}}(t+1) &= A^T(t)\tilde{\mathbf{x}}(t) + C^T(t)\tilde{\mathbf{u}}(N-1-t) \\ \tilde{\mathbf{y}}(N-1-t) &= B^T(t)\tilde{\mathbf{x}}(t) + D^T(t)\tilde{\mathbf{u}}(N-1-t) \\ & \qquad t = 0, 1, \ldots, N-1 \end{aligned} \tag{11}$$

with the input $\tilde{\mathbf{u}} = \Phi^T (\mathbf{y}_r - \Phi \mathbf{y}_k)$. □

*Remark 2:* If the reference is defined at every sample, then $M = N$, $\Phi = I_{pN}$, $\mathbf{y}_r = \mathbf{y}_d$, and the update (7) becomes

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta \mathbf{g}'(\mathbf{u}_k)^T \mathbf{e}_k$$

□

*Remark 3:* In the special case that the underlying system is linear time-invariant, (1) may be replaced by

$$\begin{aligned} \mathbf{x}(t+1) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t) \qquad t = 0, 1, \ldots, N-1 \end{aligned} \tag{12}$$

and the gradient ILC point-to-point algorithm (7) becomes

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta \left(\Phi G\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k) \tag{13}$$

with

$$G = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & CA^{N-4}B & \cdots & D \end{bmatrix} \tag{14}$$

The algorithm (13) is shown in [5] to possess extremely desirable robust convergence properties which are inherited from its standard linear ILC counterpart

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta G^T \mathbf{e}_k \tag{15}$$

analyzed by several authors [7], [8], and rigorously tested using a gantry robot facility [9], a non-minimum phase testbed [10], and in stroke rehabilitation [11]. □

### A. Inequality constraints

Consider vector inequality constraints on the system input of the form $\Lambda \mathbf{u} \preceq \mathbf{b}$, where $\Lambda \in \mathbb{R}^{c \times mN}$ and $\mathbf{b} \in \mathbb{R}^c$, where $c$ is the number of imposed constraints. The point-to-point problem is then

$$\min_{\mathbf{u}} \|\mathbf{y}_r - \Phi \mathbf{g}(\mathbf{u})\|_2^2 \quad \text{subject to} \quad \Lambda \mathbf{u} \preceq \mathbf{b} \tag{16}$$

This can be solved using an interior-point approach to inequality constrained minimisation, termed the barrier function [12]. Employing a logarithmic barrier function produces

$$\min_{\mathbf{u}} \left\{ \|\mathbf{y}_r - \Phi \mathbf{g}(\mathbf{u})\|_2^2 - \frac{1}{\tau}\sum_{i=1}^{c} \log(b_i - \mathbf{a}_i^T \mathbf{u}) \right\} \qquad (17)$$

where $\mathbf{a}_i$, $b_i$ are the $i^{th}$ rows of $\Lambda$, $\mathbf{b}$ respectively. The solution via the gradient method is

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta_k \left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k) - \frac{1}{\tau}\Lambda^T \mathbf{d} \quad (18)$$

where the elements of $d \in \mathbb{R}^c$ are given by $d_i = 1/(b_i - \mathbf{a}_i^T \mathbf{u}_k)$. This must be implemented with the scalar $\tau$ progressively taking larger values to result in a hard limit. An appropriate update strategy is to select the highest value of $\beta_k$ that results in a feasible input, that is

$$\max_{\beta_k \in (0 \quad \beta]} \beta_k \quad \text{s.t} \quad \Lambda \left(\mathbf{u}_k + \beta_k \left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k)\right) \preceq \mathbf{b} \qquad (19)$$

and then update $\tau_{k+1}$ according to

$$\tau_{k+1} = \begin{cases} \epsilon \tau_k & \text{if } \beta_k = \beta \\ \tau_k & \text{otherwise} \end{cases} \qquad (20)$$

The multiplier $\epsilon > 1$ is chosen to effect a compromise between convergence to the hard constraint, and robustness.

*Remark 4:* Since only a few points are important, it is possible to select a modified reference $\hat{\mathbf{y}}_d$ such that

$$\hat{\mathbf{y}}_d(t) = \begin{cases} \mathbf{y}_d(t) & \text{if} \quad t = n_i, \quad i = 1, \cdots, M. \\ \mathbf{y}_s(t) & \text{else} \end{cases} \qquad (21)$$

where $\mathbf{y}_s(t)$ is a smooth interpolation between $\mathbf{y}_d(n_i)$ and $\mathbf{y}_d(n_{i+1})$ which satisfies the constraints $\mathbf{a}_i^T \mathbf{u}(t) \le b_i, \forall t \in (n_i, n_{i+1})$. This is feasible as $\mathbf{u}(t)$ can be set to 0 during the interval $(n_i, n_{i+1})$. Therefore, the constraints $\Lambda \mathbf{u}(t) \le \mathbf{b}$ can be relaxed as $\Lambda \Phi \mathbf{u}(t) \le \mathbf{b}$. This implies that by using the point-to-point ILC algorithm, the feasible region of the optimization problem with constraints (16) is enlarged.

### B. Mixed constraints

The optimization problem (4) considers only tracking performance at the desired time instants, leading to a solution

$$\mathbf{y}_r = \Phi \mathbf{g}(\mathbf{u}), \qquad (22)$$

Depending on the number of time instants specified, there will exist many feasible solutions for $\mathbf{u}$. In addition to inequality constraints on the input, this freedom can also be used to tackle a wide range of other performance indices, such as reducing the output derivative at certain times to provide smoother movements. In particular, consider minimizing a general function, $\mathbf{p}(\cdot) \in \mathbb{R}^q$, of the input, output, and states, leading to the problem

$$\min_{\mathbf{u}} \|\mathbf{W}\mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x})\|_2^2 \qquad \text{subject to} \quad \begin{cases} \mathbf{y}_r = \Phi \mathbf{g}(\mathbf{u}) \\ \Lambda \mathbf{u} \preceq \mathbf{b} \end{cases} \qquad (23)$$

where $\mathbf{W} = \text{diag}\{w_1, w_2, \ldots, w_q\}$ is a weighting matrix. From (1) each component of the state time series

$$\mathbf{x} = [\mathbf{x}(0)^T, \mathbf{x}(1)^T, \ldots, \mathbf{x}(N-1)^T]^T \qquad \in \mathbb{R}^{nN} \qquad (24)$$

has the form

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{x}_0 = \boldsymbol{\sigma}_0(\mathbf{x}(0), \mathbf{u}(0)) \\ \mathbf{x}(1) &= \mathbf{f}(\mathbf{x}(0), \mathbf{u}(0)) = \boldsymbol{\sigma}_1(\mathbf{x}(0), \mathbf{u}(0)) \\ &\vdots \\ \mathbf{x}(N-1) &= \mathbf{f}(\mathbf{x}(N-2), \mathbf{u}(N-2)) \\ &= \boldsymbol{\sigma}_{N-1}(\mathbf{x}(0), \mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(N-2)) \end{aligned} \qquad (25)$$

so that the states can be written as an algebraic function $\boldsymbol{\sigma}(\cdot) : \mathbb{R}^{mN} \to \mathbb{R}^{nN}$

$$\mathbf{x} = \boldsymbol{\sigma}(\mathbf{u}), \boldsymbol{\sigma}(\cdot) = [\boldsymbol{\sigma}_0(\cdot)^T, \boldsymbol{\sigma}_1(\cdot)^T, \boldsymbol{\sigma}_2(\cdot)^T, \ldots, \boldsymbol{\sigma}_{N-1}(\cdot)^T]^T$$

then the vector quantity $\mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x})$ in (23) can be expressed in terms of the input vector, $\mathbf{u}$, by the time series

$$\mathbf{p}(\mathbf{u}, \mathbf{g}(\mathbf{u}), \boldsymbol{\sigma}(\mathbf{u})) = \tilde{\mathbf{p}}(\mathbf{u}), \qquad \tilde{\mathbf{p}}(\cdot) : \mathbb{R}^{mM} \to \mathbb{R}^q$$

Furthermore

$$\begin{aligned} \tilde{\mathbf{p}}'(\mathbf{u}) \quad = \quad & \nabla_{\mathbf{u}} \mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x}) + \nabla_{\mathbf{y}} \mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x}) \mathbf{g}'(\mathbf{u}) \\ & + \nabla_{\mathbf{x}} \mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x}) \boldsymbol{\sigma}'(\mathbf{u}) \end{aligned}$$

Following a standard approach to the equality constrained minimization problem [12], the equality constraint in (23) is first linearized about $\mathbf{u}$ to give $\mathbf{y}_r = \Phi \mathbf{g}'(\mathbf{u})\mathbf{u}$. Then take $\hat{\mathbf{u}}$ as any solution satisfying $\Phi \mathbf{g}(\hat{\mathbf{u}}) = \mathbf{y}_r$ and introduce $F \in \mathbb{R}^{mN \times m(N-M)}$ satisfying

$$F \in \ker(\Phi \mathbf{g}'(\mathbf{u}))$$

Denote $J(\mathbf{u}) := \|\mathbf{W}\mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x})\|_2^2$, then the minimization (23) is locally replaced by the inequality constrained problem

$$\begin{aligned} \min_{\mathbf{z}} \tilde{J}(\mathbf{z}) \;&=\; J(F\mathbf{z} + \hat{\mathbf{u}}) \\ &=\; \|\mathbf{W}\tilde{\mathbf{p}}(F\mathbf{z} + \hat{\mathbf{u}})\|_2^2 \quad \text{s. t.} \quad \Lambda(F\mathbf{z} + \hat{\mathbf{u}}) \preceq \mathbf{b} \end{aligned} \qquad (26)$$

with $\mathbf{z} \in \mathbb{R}^{m(N-M)}$. On trial $k$ the applied control input $\mathbf{u}_k$ is then obtained from

$$\mathbf{u}_k = F\mathbf{z}_k + \hat{\mathbf{u}} \qquad (27)$$

Applying the barrier function method to solve (26) yields

$$\min_{\mathbf{z}} \left\{ \|\mathbf{W}\tilde{\mathbf{p}}(F\mathbf{z} + \hat{\mathbf{u}})\|_2^2 - \frac{1}{\tau}\sum_{i=1}^{mN} \log(b_i - \mathbf{a}_i^T(F\mathbf{z} + \hat{\mathbf{u}})) \right\}$$

This has solution via the gradient method

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{z}_k - \beta \nabla_{\mathbf{z}} \tilde{J}(\mathbf{z}_k) - \frac{1}{\tau}(\Lambda F)^T \mathbf{d} \\ &= \mathbf{z}_k - \beta(\tilde{\mathbf{p}}'(\mathbf{u}_k)F)^T \mathbf{W}^2 \tilde{\mathbf{p}}(\mathbf{u}_k) - \frac{1}{\tau}(\Lambda F)^T \mathbf{d} \end{aligned}$$

where the elements of $d \in \mathbb{R}^c$ are given by $d_i = 1/(b_i - \mathbf{a}_i^T(F\mathbf{z}_k + \hat{\mathbf{u}}))$. Within the ILC framework, $\tilde{\mathbf{p}}(\mathbf{u}_k)$ is replaced with the experimentally obtained counterpart $\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k)$ to give

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \beta(\tilde{\mathbf{p}}'(\mathbf{u}_k)F)^T \mathbf{W}^2 \hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) - \frac{1}{\tau}(\Lambda F)^T \mathbf{d}$$

Over each trial this locally solves the minimization component of the optimization problem (26), but the point-to-point component may not be satisfied since it relies on $\hat{\mathbf{u}}$ continuing to satisfy $\Phi \mathbf{g}(\hat{\mathbf{u}}) = \mathbf{y}_r$ within (27) with changing $\mathbf{z}_k$. This is addressed by updating $\hat{\mathbf{u}}$ using the unconstrained point-to-point update of (7). In addition, $F$ must also be updated to ensure that the problem (23) may still be locally replaced by (26). The final update sequence on each trial is

$$\mathbf{u}_{k+1} = F_k \mathbf{z}_{k+1} + \hat{\mathbf{u}}_{k+1} \tag{28}$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \beta(\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^T \mathbf{W}^2 \hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) - \frac{1}{\tau_{k+1}}(\Lambda F_k)^T \mathbf{d} \tag{29}$$

$$\hat{\mathbf{u}}_{k+1} = \hat{\mathbf{u}}_k + \alpha\left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T(\mathbf{y}_r - \Phi \mathbf{y}_k) \tag{30}$$

$$F_k \in \ker(\Phi \mathbf{g}'(\mathbf{u}_k)) \tag{31}$$

As before, the scalar $\tau$ must be increased each trial in order to produce a hard bound on the input signal. To ensure that the constraint engages productively with the minimization component of the update (29), we again apply the $\tau$ update procedures of Section III-A, replacing (19) with

$$\max_{\beta_k \in (0\ \beta]} \beta_k \quad \text{s.t} \quad \Lambda\Big(F_k\Big(\mathbf{z}_k - \beta_k(\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^T \mathbf{W}^2 \hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) - \frac{1}{\tau_k}(\Lambda F_k)^T \mathbf{d}\Big) + \hat{\mathbf{u}}_{k+1}\Big) \preceq \mathbf{b}$$

and then applying (20) to update $\tau$.

## IV. NEWTON METHOD BASED ILC

Whilst providing a high level of robustness to plant uncertainty, the gradient descent approach provides only a linear convergence rate. This section exchanges it for the Newton method which potentially delivers quadratic convergence. Application to (4) gives rise to the ILC update

$$\begin{aligned}\mathbf{u}_{k+1} &= \mathbf{u}_k - \nabla_{\mathbf{u},\mathbf{u}} J(\mathbf{u}_k)^{-1} \nabla_{\mathbf{u}} J(\mathbf{u}_k)\\ &= \mathbf{u}_k + ((\Phi \mathbf{g}'(\mathbf{u}_k))^T \Phi \mathbf{g}'(\mathbf{u}_k))^{-1} (\Phi \mathbf{g}'(\mathbf{u}_k))^T (\mathbf{y}_r - \Phi \mathbf{y}_k)\\ &= \mathbf{u}_k + (\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k) \end{aligned} \tag{32}$$

where $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger$ denotes the pseudoinverse of $\Phi \mathbf{g}'(\mathbf{u}_k)$. Since its computation involves inverse and derivative operations, the update is difficult to implement, especially for large values of $N$. It may contain excessive amplitudes and high frequencies, and, depending on the system relative degree, $\mathbf{g}'(\mathbf{u}_k)$ is likely to be singular. However, it is shown in [5] that $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k)$ is the solution, $\mathbf{u}$, to

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \qquad \text{subject to} \quad \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u} = \mathbf{y}_r - \Phi \mathbf{y}_k \tag{33}$$

which is further shown to be equal to the solution to

$$\min_{\mathbf{u}} \|\mathbf{y}_r - \Phi \mathbf{y}_k - \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u}\|_2^2 \tag{34}$$

via the unconstrained gradient descent method of Section III which always yields the minimum input energy solution. In particular, the substitutions $\mathbf{y}_r \Leftrightarrow \mathbf{y}_r - \Phi \mathbf{y}_k$ and $\mathbf{g}(\mathbf{u}) \Leftrightarrow \mathbf{g}'(\mathbf{u}_k)\mathbf{u}$ are made in (4), with associated iterative update

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha\left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k - \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u}_j) \tag{35}$$

applied to the plant $\Phi \mathbf{g}'(\mathbf{u}_k)$, whose solution approximates $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k)$ after sufficient iterations. The number of trials is chosen to effect a compromise between excessively high amplitudes/frequencies in the update, robustness, and subsequent performance. The total update sequence is

(a) apply input $\mathbf{u}_k$ to the real plant and record output $\mathbf{y}_k$
(b) solve (34) through repeated application of (35) to the system $\Phi \mathbf{g}'(\mathbf{u}_k)$ to obtain a suitable approximation to $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k)$
(c) use the resulting input to form the next input to the Newton update (32). Go to (a)

*Remark 5:* If the reference is defined at every point, then $M = N$, $\Phi = I_{pN}$, $\mathbf{y}_r = \mathbf{y}_d$, and the update (32) becomes

$$\mathbf{u}_{k+1} \quad = \quad \mathbf{u}_k + \mathbf{g}'(\mathbf{u}_k)^\dagger \mathbf{e}_k \tag{36}$$

with (34) becoming

$$\min_{\mathbf{u}} \|\mathbf{e}_k - \mathbf{g}'(\mathbf{u}_k)\mathbf{u}\|_2^2 \tag{37}$$

The problem (37) now fits within the standard ILC framework, with $\mathbf{g}'(\mathbf{u}_k)$ realized in state-space form by the system (9). Therefore the solution to finding the input $\mathbf{u}$ that drives the system to track $\mathbf{e}_k$ can be solved by any convergent ILC approach for linear time-varying systems. In [13] it was assumed that the plant had an equal number of inputs and outputs, and the Norm Optimal ILC law was employed between each trial to solve this problem, and hence provide the term $\mathbf{g}'(\mathbf{u}_k)^\dagger \mathbf{e}_k = \mathbf{g}'(\mathbf{u}_k)^{-1} \mathbf{e}_k$ used in the construction of the next control input. □

### A. Inequality constraints

Consider again the constrained problem (16)

$$\min_{\mathbf{u}} \|\mathbf{y}_r - \Phi \mathbf{g}(\mathbf{u})\|_2^2 \qquad \text{subject to} \quad \Lambda \mathbf{u} \preceq \mathbf{b} \tag{38}$$

This can be solved via the Newton method by imposing the inequality constraint in the iterative calculation of the descent direction, $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k)$, given by the solution to (33). The constraint $\Lambda \mathbf{u}_{k+1} \preceq \mathbf{b}$ then translates to $\Lambda \mathbf{u} \preceq \mathbf{b} - \Lambda \mathbf{u}_k$ and the descent direction is given by

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \qquad \text{subject to} \quad \left\{ \begin{array}{rcl} \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u} &=& \mathbf{y}_r - \Phi \mathbf{y}_k \\ \Lambda \mathbf{u} &\preceq& \mathbf{b} - \Lambda \mathbf{u}_k \end{array}\right.$$

whose solution is then used in (32). The solution to this mixed constraint problem can be addressed within the ILC framework using the approach developed in Section III-B. From previous discussion this is equivalent to applying the gradient method to solve

$$\min_{\mathbf{u}} \|\mathbf{y}_r - \Phi \mathbf{y}_k - \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u}\|_2^2 \qquad \text{subject to} \quad \Lambda \mathbf{u} \preceq \mathbf{b} - \Lambda \mathbf{u}_k$$

This is the form addressed in Section III-A, with corresponding update

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha\left(\Phi \mathbf{g}'(\mathbf{u}_k)\right)^T (\mathbf{y}_r - \Phi \mathbf{y}_k - \Phi \mathbf{g}'(\mathbf{u}_k)\mathbf{u}_j) - \tfrac{1}{\tau}\Lambda^T \mathbf{d} \tag{39}$$

applied to the plant $\Phi \mathbf{g}'(\mathbf{u}_k)$, where the elements of $\mathbf{d} \in \mathbb{R}^c$ are given by $d_i = 1/(b_i - \mathbf{a}_i^T(\mathbf{u}_j + \mathbf{u}_k))$. The highest value of $\alpha$ is used which results in a feasible input, and $\tau$ is updated as in (20).

The full update sequence is therefore

(a) apply input $\mathbf{u}_k$ to the real plant and record output $\mathbf{y}_k$
(b) construct suitable approximation to $(\Phi \mathbf{g}'(\mathbf{u}_k))^\dagger (\mathbf{y}_r - \Phi \mathbf{y}_k)$ satisfying $\Lambda \mathbf{u}_{k+1} \preceq \mathbf{b}$ when used in Newton update (32), through repeated application of (39)
(c) use the resulting input to form the next input to the Newton update (32). Go to (a)

### B. Mixed constraints

As in (23), introduce an additional objective function whilst satisfying the point-to-point tracking requirement through use of an equality constraint

$$\min_{\mathbf{u}} \|\mathbf{W}\mathbf{p}(\mathbf{u},\mathbf{y},\mathbf{x})\|_2^2 \quad \text{s. t.} \quad \begin{cases} \mathbf{y}_r = \Phi\mathbf{g}(\mathbf{u}) \\ \Lambda\mathbf{u} \preceq \mathbf{b} \end{cases} \tag{40}$$

As in Section III-B, the equality constraint is first removed by defining

$$F \in \ker(\Phi\mathbf{g}'(\mathbf{u}))$$

which allows the problem to be locally replaced by the inequality constrained problem (26). The control input applied to the plant is

$$\mathbf{u}_k = F_k\mathbf{z}_k + \hat{\mathbf{u}}_k \tag{41}$$

where $\hat{\mathbf{u}}_k$ and $F_k$ are also updated to ensure the point-to-point tracking objective is satisfied. If the inequality constraint is omitted from (26), its solution using the Newton method is

$$\mathbf{z}_{k+1} = \mathbf{z}_k + (\mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^{\dagger}\mathbf{W}\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) \tag{42}$$

where the Newton descent direction $(\mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^{\dagger}\mathbf{W}\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k)$ is itself the solution to

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \quad \text{s. t.} \quad \mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k\mathbf{u} = \mathbf{W}\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) \tag{43}$$

via the gradient method. Now solve (26) via the Newton method by imposing the inequality constraint on (43). In terms of the control input, on trial $k+1$, the constraint enforces $\Lambda\mathbf{u}_{k+1} \preceq \mathbf{b}$, which, assuming $\hat{\mathbf{u}}_k$ and $F_k$ have not yet been updated, can be written as $\Lambda(F_k\mathbf{z}_{k+1} + \hat{\mathbf{u}}_k) \preceq \mathbf{b}$. In terms of the Newton descent direction, this translates to $\Lambda(F_k(\mathbf{z}_k + \mathbf{u}) + \hat{\mathbf{u}}_k) \preceq \mathbf{b}$. Hence, (43) becomes

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \text{ s. t. } \begin{cases} \mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k\mathbf{u} = \mathbf{W}\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) \\ \Lambda F_k\mathbf{u} \preceq \mathbf{b} - \Lambda\hat{\mathbf{u}}_k - \Lambda F_k\mathbf{z}_k \end{cases}$$

This is equivalent to

$$\min_{\mathbf{u}} \|\mathbf{W}(\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) - \tilde{\mathbf{p}}'(\mathbf{u}_k)F_k\mathbf{u})\|_2^2 \quad \text{s. t.} \quad \Lambda F_k\mathbf{u} \preceq \mathbf{b} - \Lambda\hat{\mathbf{u}}_k - \Lambda F_k\mathbf{z}_k$$

with corresponding update

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \beta(\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^T\mathbf{W}^2(\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k) - \tilde{\mathbf{p}}'(\mathbf{u}_k)F_k\mathbf{u}_j) - \frac{1}{\tau}((\Lambda F_k)^T\mathbf{d}) \tag{44}$$

applied to the plant $\mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k$, where the elements of $\mathbf{d} \in \mathbb{R}^c$ are given by $d_i = 1/(b_i - \mathbf{a}_i^T(F_k\mathbf{z}_k + \hat{\mathbf{u}}_k + \mathbf{u}_j))$.

As previously, the term $\hat{\mathbf{u}}_k$ in (41) must also be updated so that the point-to-point tracking objective is satisfied with changing $\mathbf{z}_k$. This is achieved using the Newton ILC update (32) with the constraint $\Lambda(F_k\mathbf{z}_{k+1} + \hat{\mathbf{u}}_{k+1}) \preceq \mathbf{b}$ where we have assumed that $\mathbf{z}_{k+1}$ has just been updated via (42) as discussed. The unconstrained Newton ILC descent direction, $(\Phi\mathbf{g}'(\mathbf{u}_k))^{\dagger}(\mathbf{y}_r - \Phi\mathbf{y}_k)$, in (32) is the solution to

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \quad \text{s. t.} \quad \Phi\mathbf{g}'(\mathbf{u})\mathbf{u} = \mathbf{y}_r - \Phi\mathbf{y}_k \tag{45}$$

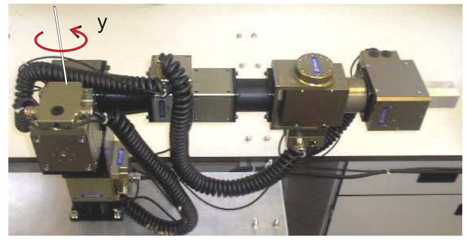


Fig. 1. Robotic manipulator system.

so that the corresponding required constraint is $\Lambda(F_k\mathbf{z}_{k+1} + \hat{\mathbf{u}}_k + \mathbf{u}) \preceq \mathbf{b}$. This therefore produces

$$\min_{\mathbf{u}} \|\mathbf{u}\|_2^2 \quad \text{s. t.} \quad \begin{cases} \Phi\mathbf{g}'(\mathbf{u})\mathbf{u} = \mathbf{y}_r - \Phi\mathbf{y}_k \\ \Lambda\mathbf{u} \preceq \mathbf{b} - \Lambda F_k\mathbf{z}_{k+1} - \Lambda\hat{\mathbf{u}}_k \end{cases}$$

which is equivalent to

$$\min_{\mathbf{u}} \|\mathbf{y}_r - \Phi\mathbf{y}_k - \Phi\mathbf{g}'(\mathbf{u}_k)\mathbf{u}\|_2^2 \text{ s. t. } \Lambda\mathbf{u} \preceq \mathbf{b} - \Lambda F_k\mathbf{z}_{k+1} - \Lambda\hat{\mathbf{u}}_k$$

with the corresponding gradient descent update

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \alpha(\Phi\mathbf{g}'(\mathbf{u}_k))^T(\mathbf{y}_r - \Phi\mathbf{y}_k - \Phi\mathbf{g}'(\mathbf{u}_k)\mathbf{u}_j) - \tfrac{1}{\tau}(\Lambda^T\mathbf{d}) \tag{46}$$

applied to the plant $\Phi\mathbf{g}'(\mathbf{u}_k)$, where the elements of $\mathbf{d} \in \mathbb{R}^c$ are given by $d_i = 1/(b_i - \mathbf{a}_i^T(F_k\mathbf{z}_{k+1} + \hat{\mathbf{u}}_k + \mathbf{u}_j))$.

The total update sequence is therefore

(a) apply input $\mathbf{u}_k$ to the real plant and record output $\mathbf{y}_k$
(b) construct a suitable approximation to the term $(\mathbf{W}\tilde{\mathbf{p}}'(\mathbf{u}_k)F_k)^{\dagger}\mathbf{W}\hat{\tilde{\mathbf{p}}}(\mathbf{u}_k)$ which satisfies $\Lambda F_k\mathbf{z}_{k+1} \preceq \mathbf{b} - \Lambda\hat{\mathbf{u}}_k$ when used in Newton update (42), through repeated application of (44)
(c) use the resulting input to form the next update (42)
(d) construct suitable approximation to $(\Phi\mathbf{g}'(\mathbf{u}_k))^{\dagger}(\mathbf{y}_r - \Phi\mathbf{y}_k)$ satisfying $\Lambda\hat{\mathbf{u}}_{k+1} \preceq \mathbf{b} - \Lambda F_k\mathbf{z}_{k+1}$ when used in Newton update (32), through repeated application of (46)
(e) use the resulting input to form the next update (32)
(f) use the new $\mathbf{u}_{k+1}$ and $\mathbf{z}_{k+1}$ values to form the next control input using (41). Go to (a)

The number of updates of (44) and (46) are chosen to effect a compromise between excessive amplitudes/frequencies present in the update $\mathbf{u}_{k+1}$, robustness, and the subsequent performance achieved.

## V. EXPERIMENTAL RESULTS

The approaches developed have been tested on a six degree of freedom anthropomorphic robotic arm whose five rotary joints are composed of PowerCubes (Schunk GmbH & Co.) incorporating brushless servomotors with integrated power electronics and transmission. These communicate with a dSPACE ds1103 control board via a CAN bus at a rate of 500 kbit/s. Results are presented for the first joint which is aligned in the horizontal plane as shown in Fig. 1. Each joint has been identified using frequency response tests and includes a feedback loop to provide base-line performance. A model of the first link has been identified as $G(s) = \frac{500985.1977(s+12.14)(s+24.01)^2}{(s+29.15)(s+23.07)(s+3.403)(s^2+34.8s+355.4)(s^2+127.4s+4310)}$ and a sampling time of 200Hz has been used.

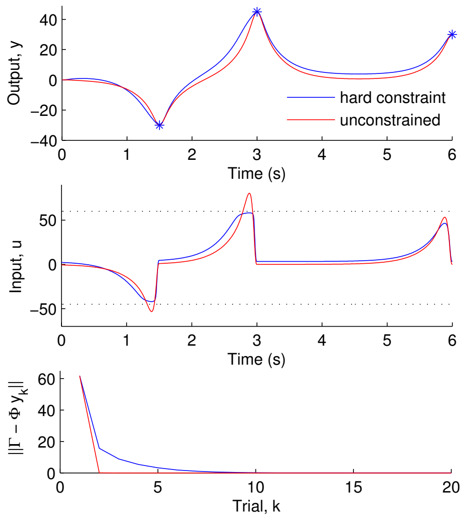


Fig. 2. Newton method-based point-to-point ILC with inequality constraint.

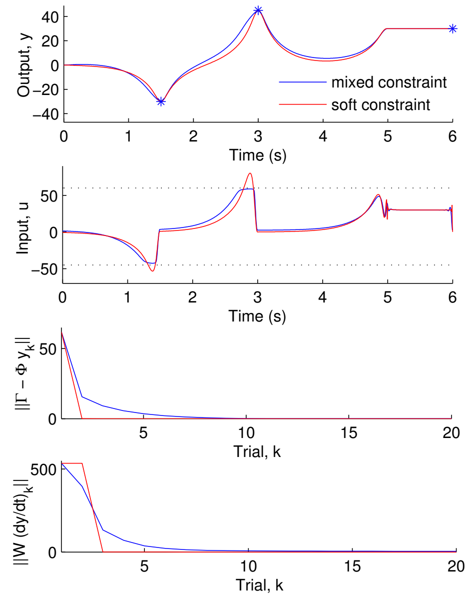


Fig. 3. Newton method-based point-to-point ILC with mixed constraints.

First the Newton approach with inequality constraints of Section IV-A has been implemented. Constraints of $-45 \leq \mathbf{u}(t) \leq 60$ have been employed through selection of $\Lambda = [I,\ -I]^T$, $\mathbf{b} = [60,\ 60,\ldots,60,\ 45,\ 45,\ldots,\ 45]^T$ in (38). Results are shown in Fig. 2 where 25 repetitions of (39) have been used between trials to calculate the required descent direction appearing in (32), with a value of $\epsilon = 1.04$ employed in the updating of $\tau$. For comparison, results for the unconstrained case are also shown, where (35) is used in place of (39) (yielding the minimum input energy solution). A high level of performance can be seen in both cases. Next a soft constraint is applied in which the output derivative is minimized over the period $5 \leq t \leq 6$. In this case $\mathbf{p}(\cdot) = D\mathbf{y}$ in (40) where $D$ is the differential operator, together with $\mathbf{W} = \text{diag}\{0,\ldots,0,1,\ldots,1\}$. This gives $\tilde{\mathbf{p}}(\mathbf{u}) = D\mathbf{g}(\mathbf{u})$, $\tilde{\mathbf{p}}'(\mathbf{u}) = D\mathbf{g}'(\mathbf{u})$, and the soft constraint component (42) becomes $\mathbf{z}_{k+1} = \mathbf{z}_k + \beta(\mathbf{W}D\tilde{\mathbf{g}}'(\mathbf{u}_k)F_k)^\dagger\mathbf{W}\dot{\mathbf{y}}_k$. Furthermore (44) becomes $\mathbf{u}_{j+1} = \mathbf{u}_j + \beta(D\tilde{\mathbf{g}}'(\mathbf{u}_k)F_k)^T\mathbf{W}^2(\dot{\mathbf{y}}_k - D\mathbf{g}'(\mathbf{u}_k)F_k\mathbf{u}_j) - \frac{1}{\tau}(\Lambda F_k)^T\mathbf{d})$ applied to the plant $\mathbf{W}D\mathbf{g}'(\mathbf{u}_k)F_k$. Results are shown in Fig. 3 where 25 trials of (44) are used with $\epsilon = 1.04$ to produce the required descent direction. It can be seen that the point-to-point task is accomplished to high accuracy whilst satisfying both hard and soft constraints.

## VI. Conclusions and Future Work

The requirement for point-to-point motion control arises in many practical applications, including industrial automation, robotics and rehabilitation engineering. It has been shown how the ILC framework can provide solutions which exploit the freedom available to simultaneously address both hard constraints on the input, and soft constraints on the input, output and states. Experimental results confirm the practical utility and performance of the proposed approaches.

Future work will consider the inclusion of prescribed variation in the temporal point-to-point locations to provide more flexibility and faster convergence properties. Constraints linking two or more outputs will also be considered, allowing co-ordinated movements to be performed.

## References

[1] J. Park, P. H. Chang, H. S. Park, and E. Lee, "Design of learning input shaping technique for residual vibration suppression in an industrial robot," *IEEE/ASME Trans Mech*, vol. 11, no. 1, pp. 55–65, 2006.

[2] J. van de Wijdeven and O. Bosgra, "Residual vibration suppression using hankel iterative learning control," *Int J Rob Nonlinear Control*, vol. 18, pp. 1034–1051, 2008.

[3] H. Ding and J. Wu, "Point-to-point control for a high-acceleration positioning table via cascaded learning schemes," *IEEE Trans Ind Elect*, vol. 54, no. 5, pp. 2735–2744, 2007.

[4] G. Gauthier and B. Boulet, "Robust design of terminal ILC with $H_\infty$ mixed sensitivity approach for a thermoforming oven," *J Control Sci and Eng*, p. Article ID 289391, 2008.

[5] C. T. Freeman, "Constrained point-to-point iterative learning control for discrete-time nonlinear systems," *IEEE Trans Control Sys Tech*, vol. Submitted, 2010.

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

[7] K. Chen and R. W. Longman, "Stability issues using FIR filtering in repetitive control," *Advances Astr Sci*, vol. 206, pp. 1321–1339, 2002.

[8] D. H. Owens, J. J. Hätönen, and S. Daley, "Robust monotone gradient-based discrete-time iterative learning control," *Int J Robust and Nonlinear Control*, vol. 19, pp. 634–661, 2009.

[9] J. D. Ratcliffe, J. J. Hätönen, P. L. Lewin, E. Rogers, and D. H. Owens, "Robustness analysis of an adjoint optimal iterative learning controller with experimental verification," *Int J Robust and Nonlinear Control*, vol. 18, no. 10, pp. 1089–1113, 2008.

[10] C. T. Freeman, P. L. Lewin, and E. Rogers, "Further results on the experimental evaluation of iterative learning control algorithms for non-minimum phase plants," *Int J Control*, vol. 80, no. 4, pp. 569–582, 2007.

[11] C. T. Freeman, A. M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers, "Iterative learning control of FES applied to the upper extremity for rehabilitation," *Cont Eng Pract*, vol. 17, no. 3, pp. 368–381, 2009.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2005.

[13] T. Lin, D. H. Owens, and J. J. Hätönen, "Newton method based iterative learning control for discrete non-linear systems," *Int J Control*, vol. 79, no. 10, pp. 1263–1276, 2006.