International Journal of Control
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcon20

Multivariable norm optimal iterative learning control with auxiliary optimisation

David H. Owens [a b c], Chris T. Freeman [b] & Bing Chu [b]

[a] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, Mappin Street, UK

[b] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, Highfield, UK

[c] Department of Advanced Robotics, Instituto Italiano di Tecnologia, 30, 16163, via Morego, Genova

Published online: 05 Apr 2013.

To cite this article: David H. Owens , Chris T. Freeman & Bing Chu (2013): Multivariable norm optimal iterative learning control with auxiliary optimisation, International Journal of Control, DOI:10.1080/00207179.2013.771822

To link to this article: http://dx.doi.org/10.1080/00207179.2013.771822

International Journal of Control, 2013
http://dx.doi.org/10.1080/00207179.2013.771822

Multivariable norm optimal iterative learning control with auxiliary optimisation

David H. Owens[a,b,c], Chris T. Freeman[b,*] and Bing Chu[b]

[a]*Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK;* [b]*Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK;* [c]*Department of Advanced Robotics, Instituto Italiano di Tecnologia, via Morego, 30, 16163, Genova*

(*Received 25 June 2012; final version received 28 January 2013*)

The paper describes a substantial extension of norm optimal iterative learning control (NOILC) that permits tracking of a class of finite dimensional reference signals whilst simultaneously converging to the solution of a constrained quadratic optimisation problem. The theory is presented in a general functional analytical framework using operators between chosen real Hilbert spaces. This is applied to solve problems in continuous time where tracking is only required at selected intermediate points of the time interval but, simultaneously, the solution is required to minimise a specified quadratic objective function of the input signals and chosen auxiliary (state) variables. Applications to the discrete time case, including the case of multi-rate sampling, are also summarised. The algorithms are motivated by practical need and provide a methodology for reducing undesirable effects such as payload spillage, vibration tendencies and actuator wear whilst maintaining the desired tracking accuracy necessary for task completion. Solutions in terms of NOILC methodologies involving both feedforward and feedback components offer the possibilities of greater robustness than purely feedforward actions. Results describing the inherent robustness of the feedforward implementation are presented and the work is illustrated by experimental results from a robotic manipulator.

Keywords: iterative learning control; learning control; optimisation; linear systems; soft constraints; auxiliary optimisation

1. Introduction

Iterative learning control (ILC) is a methodology applicable to systems which repeatedly perform the same operation over a finite time interval $0 \leq t \leq T$. Each repetition is termed an iteration or trial. The aim is to sequentially improve the performance of the operation as the trial index, k, increases through suitable use of data recorded over previous trials of the task, often in combination with current-trial information. Originally conceived for robotic applications (Arimoto, Miyazaki, & Kawamura, 1984), ILC has since been applied to wide variety of application fields, with research interest in both theoretical and application domains continuing to expand year-on-year. Recent overviews of the literature are given in Ahn, Chen, and Moore (2007) and Bristow, Tharayil, and Alleyne (2006). A mature algorithmic framework has evolved for the case of linear ILC applied to linear plants, which include gradient based algorithms whose convergence and robustness properties have been extensively studied (Furuta & Yamakita, 1987; Kinosita, Sogo, & Adachi, 2002; Owens, Hätönen, & Daley, 2009). A highly cited example is norm optimal ILC (NOILC) (Amann, Owens, & Rogers, 1996b), which calculates the input of trial k to minimise a quadratic cost function comprising weighted norms of the current trial error vector and the difference in the control input vector on two successive trials. A general Hilbert space and hence general theoretical development of NOILC appeared originally in Amann et al. (1996b) and was shown to lead to a control realisation as a state feedback plus feedforward (predictive) term. The ability to achieve control over error and input (mean square) norm evolution has led to NOILC receiving significant attention in the ILC community e.g. Gunnarsson and Norrlof (2001), Lee, Lee, and Kim (2000) and Barton and Alleyne (2011). The robustness and performance afforded by this combined control structure was subsequently verified experimentally on applications including an accelerator based free electron laser (Rogers et al., 2010), multi-axis laser facility (Barton & Alleyne, 2011) and within stroke rehabilitation (Freeman, Rogers, Hughes, Burridge, & Meadmore, 2012). The general formulation also permitted further theoretical developments including the extension to the discrete time case (Amann, Owens, & Rogers, 1996a), an N-iteration ahead predictive solution (Amann, Owens, & Rogers, 1998), acceleration mechanisms (Owens & Chu, 2009), the inclusion of convex input constraints (Chu & Owens, 2010) and a detailed analysis of the effect of non-minimum phase zeros in the continuous (Owens, Chu, Rogers, Freeman, & Lewin, 2012) and discrete time (Owens & Chu, 2010) cases.

*Corresponding author. Email: cf@ecs.soton.ac.uk

Over the years, some of the original postulates in Arimoto et al. (1984) defining the task and underlying plant have been relaxed, one of which is the stipulation that the task is that of following a motion profile defined at *all* points $0 \leq t \leq T$. For example, a recent body of work (Chen & Xu, 1997; Gauthier & Boulet, 2008, 2009; Wang & Hou, 2011; Xu & Huang, 2008; Xu, Chen, Lee, & Yamamoto, 1999) considered the case where a specified position must be reached at time $t = T$. Extensions to this have recently been considered in discrete (Freeman, 2012; Freeman & Tan, 2012; Son & Ahn, 2011) and continuous time (Owens, Freeman, & Dinh, 2012; Son, Ahn, & Nguyen, 2011a; Son, Nguyen, & Ahn, 2011b; Son & Ahn, 2012), which enforce tracking only at specified intermediate times, thereby providing an ILC framework for use with a broad class of applications including production line automation, crane control, satellite positioning and robotic 'pick and place' tasks in which the system output (e.g. payload position) is only critical at a finite set of prescribed time instants. A further application is within stroke rehabilitation. Here electrical stimulation is applied to assist patients' movements precisely to encourage voluntary effort whilst ensuring accurate movement, but where the task is naturally specified in the form of a point-to-point problem (Freeman et al., 2012). Whilst the point-to-point problem can be tackled by the standard ILC framework by employing a reference, which passes through the required points, its formulation as an optimisation problem which does not enforce tracking over unnecessary time periods clearly holds the possibility of more rapid convergence and superior transient performance, whilst enabling more transparent design and analysis. This was illustrated in Owens et al. (2012) using an extension of NOILC to deal with the intermediate point tracking problem. Experimental results indicated the relative benefits of feedforward plus feedback implementations as compared to the purely feedforward implementation.

In general, tracking at intermediate points only implies that the desired tracking control signal will be non-unique. It is natural in these circumstances to ask whether or not an ILC algorithm can be constructed that not only achieves the desired tracking at intermediate points but also converges to a solution that minimises an auxiliary quadratic objective function dependent on the input signal and a defined set of auxiliary variables e.g. accelerations and inter-sample velocities. Within robotic manipulation, such an approach would limit system vibration as well as spillage or damage of/to the payload. The designer may also wish to limit the control effort in order to reduce energy needs and actuator wear. The aim of this paper is to show that solutions do indeed exist for this problem and that they can be approached, constructed and analysed using techniques similar to those used in the well-developed framework of NOILC. Two formulations of algorithms based on 'switching' between two different optimisation problems are presented, the second being a relaxation of the first that requires more iterations but displays superior robustness in practice. In addition, as the computations are a combination of matrix manipulations and quadratic optimisation similar to NOILC, the algorithms can be implemented in a feedforward form or a combined feedforward plus feedback form.

This paper is structured as follows: in Section 2 the problem is defined and modelled in a general product Hilbert space setting. This approach simplifies notation and indicates the generality of the algorithms to be derived later. The notational simplification is analogous to that obtained from the use of transfer functions in classical control. The generality comes from the ability to analyse and construct algorithms for continuous time, discrete time and other classes of systems under the same framework. Section 3 explains how the operator formulation can be applied to the case of linear, continuous state space systems, linear discrete systems and multi-rate systems. In Section 4, two new switching algorithms are derived and key information on convergence rates and the character of the limit input function are proven. In Section 5, a theoretical analysis of a feedforward implementation is discussed in detail and indicates the potential for considerable robustness of convergence in practice. This is followed by a practical evaluation using a robotic experimental facility, which indicates that the theory is an excellent guide to convergence and robustness properties in practice. Conclusions are given in Section 7.

2. Iterative tracking with auxiliary optimisation: a general model

In this section, a general theory is presented that provides a framework for the solution of a class of linear system ILC problems that not only have a tracking requirement but also require that a solution is found simultaneously that minimises a selected quadratic performance criterion (also called an objective function). The framework chosen is that used by Amann et al. (1996b), namely functional analysis in real Hilbert spaces. This has the benefit of compactness of notation and proof plus a generality that permits wide application to a number of problems including time-invariant, time-varying continuous time, discrete time and multi-rate ILC problems. The presentation starts with this problem formulation and then, in later sections, the precise form of a proposed algorithm is described. The convergence properties of the algorithm are then investigated and a modification is described that provides flexibility for implementation purposes. Proofs of robustness properties are then given. All use the modelling tools described in this section.

2.1. Systems models, signal spaces, operators and the problem statement

The system model to be used has signals consisting of inputs $u \in \mathcal{U}$, outputs $y \in \mathcal{Y}$ and auxiliary variables $z \in \mathcal{Z}$ where

$\mathcal{U}, \mathcal{Y}, \mathcal{Z}$ are real Hilbert spaces. The inner product in $\mathcal{U}$ is denoted $<u, v>_{\mathcal{U}}$ with induced norm $\|u\|_{\mathcal{U}} = \sqrt{<u, u>_{\mathcal{U}}}$ with similar notation used for $\mathcal{Y}$ and $\mathcal{Z}$.

The underlying systems dynamics is described by a linear relation

$$y = Gu + d \qquad \text{(Plant Dynamics)}, \tag{1}$$

where G is a bounded linear operator mapping $\mathcal{U}$ into $\mathcal{Y}$ and d is an iteration independent term describing initial condition effects and any exactly repeated disturbances or bias in the signal. Implicit in this assumption therefore is that the initial condition on each iteration is the same. This model plays an indirect role in the general theory but, as will be seen from the state space examples in the next section, it plays a crucial role in the construction of the actual tracking problem. This tracking problem requires two ingredients, namely a *tracking requirement* and *auxiliary variables* defined as follows. Examples describing formulations in terms of state space models are given in later sections (e.g. Section 3.1.1).

The tracking requirement is defined using a signal $r^e \in \mathcal{R}_e$, where $\mathcal{R}_e$ is also a Hilbert space. The tracking sought is specified by the linear relation

$$y^e = r^e \text{ where } y^e = G_0 u + d_0 \quad \text{(Tracking Requirement)}, \tag{2}$$

where G_0 is a bounded linear operator mapping $\mathcal{U}$ into $\mathcal{R}_e$ and $d_0 \in \mathcal{R}_e$ is an iteration independent term again describing the effects of initial conditions and repeated disturbances or bias.

An auxiliary signal z is selected as a suitable measure of desired control performance and is specified by a linear relation

$$z = G_1 u + d_1 \qquad \text{(Auxiliary Variable Dynamics)}, \tag{3}$$

where G_1 is a bounded linear operator mapping $\mathcal{U}$ into a real Hilbert space $\mathcal{Z}$ and $d_1 \in \mathcal{Z}$ is an iteration independent term again describing the effect of initial conditions and exactly repeated disturbances or bias. The precise ILC tracking problem is defined as follows:

Problem definition: *Given a desired reference $r^e \in \mathcal{R}_e$ and an initial input signal $u_0 \in \mathcal{U}$, construct an ILC algorithm for the plant with dynamics $y = Gu + d$ to find a system input signal in $\mathcal{U}$ that satisfies the tracking requirement $y^e = r^e$ where $y^e = G_0 u + d_0$ whilst simultaneously minimising the objective function,*

$$J(z, u) = \|z - z_0\|^2_{\mathcal{Z}} + \|u - u_0\|^2_{\mathcal{U}}, \tag{4}$$

where (z_0, u_0) is the solution pair of the auxiliary equation for the given input u_0. That is, the ILC algorithm converges to a solution of the constrained optimisation problem,

$$u_\infty = argmin\{J(z, u) : u \in \mathcal{U}, \quad r_e = G_0 u + d_0, z = G_1 u + d_1\}. \tag{5}$$

A special case: *If $G_1 = 0$ and $d_1 = 0$, then (z_0, u_0) = (0, u_0) is a solution of the auxiliary system (for all choices of u_0) and, in this case, the auxiliary optimisation problem is the minimisation of the input energy measure $\|u - u_0\|^2_{\mathcal{U}}$.*

Next, the following assumptions are also used to ensure invertibility of relevant 'matrix' operators later in the development in this paper.

Important dimensionality and invertibility assumptions: *It is assumed throughout the paper that*

(1) *the Hilbert space $\mathcal{R}_e$ is finite dimensional and also that*
(2) *the kernel (null space) of the adjoint operator satisfies $ker[G_0^*] = \{0\}$ (equivalently, the range of G_0, denoted $\mathcal{R}[G_0]$, is exactly $\mathcal{R}_e$).*

The important aspects of Hilbert space theory relevant to the theoretical development are summarised in Appendix 1.

Note: The condition on the range of G_0 is necessary to ensure that all signals in $\mathcal{R}_e$ can be tracked. In practice, if $\mathcal{R}[G_0]$ is a proper subspace of $\mathcal{R}_e$, the apparent problem of failing to satisfy the assumptions can be removed in a formal way by regarding $\mathcal{R}[G_0]$ as a Hilbert space in its own right and setting $\mathcal{R}_e$ equal to this subspace.

2.2. Invertibility and spectral properties

The following results follow from the above construction and assumptions and are important to the following theoretical development. The first easily proven result provides information on invertibility of a special operator.

Lemma 2.1: *Given the above assumptions, let $L : \mathcal{U} \to \mathcal{U}$ be any positive definite or semi-definite, self adjoint operator in $\mathcal{U}$. Then the following self-adjoint operator*

$$G_0(I + L)^{-1} G_0^* : \mathcal{R}_e \to \mathcal{R}_e$$

can be represented by a non-singular, positive definite (in the topology of $\mathcal{R}_e$), square matrix of dimension equal to the dimension of the space $\mathcal{R}_e$. In particular, choosing $L = 0$, the matrix representation of $G_0 G_0^*$ is square and non-singular.

The following results characterise the spectrum of an operator relevant to the problem. It uses an important alternative topology for $\mathcal{R}_e$ defined by the result.

Lemma 2.2: *Using the above assumptions and construction, suppose that a new Hilbert topology in $\mathcal{R}_e$ is defined by the inner product $<u, v>_0 = <u, (G_0G_0^*)^{-1}v>_{\mathcal{R}_e}$ and associated induced norm. Then this topology is topologically equivalent to that induced by $<\cdot, \cdot>_{\mathcal{R}_e}$.*

Proof: Note also that all norm topologies in finite dimensional spaces are equivalent. The bilinear form $<u, v>_0 = <u, (G_0G_0^*)^{-1}v>_{\mathcal{R}_e}$ is indeed an inner product as $<u, v>_0 = < v, u>_0 (\forall u, v)$ and, without loss of generality, it is always possible to choose a basis for $\mathcal{R}_e$ in which the matrix representation of the product $<u, v>_{\mathcal{R}_e} = u^T v$. The norm in $\mathcal{R}_e$ is then Euclidean. In this basis, the self-adjoint operator $G_0G_0^*$ is a positive-definite, symmetric matrix so the associated quadratic form $<u, u>_0 = <u, (G_0G_0^*)^{-1}u>_{\mathcal{R}_e}$ is positive definite. □

Lemma 2.3: *Using the above assumptions and construction, suppose that L is self-adjoint and positive semi-definite on $\mathcal{U}$ with $\underline{\delta}^2 I \le L \le \overline{\delta}^2 I$ for some scalars $0 \le \underline{\delta}^2 \le \overline{\delta}^2$. Define $W_\infty = I - G_0(I + L)^{-1}G_0^*(G_0G_0^*)^{-1}$. Then W_∞ is self adjoint and positive semi-definite in the topology defined by the inner product $<\cdot, \cdot>_0$ and its eigenvalues are real and lie in the half-open interval [0, 1). More precisely, every eigenvalue λ satisfies the inequality,*

$$0 \le \frac{\underline{\delta}^2}{1+\underline{\delta}^2} \le \lambda \le \frac{\overline{\delta}^2}{1+\overline{\delta}^2} < 1. \tag{6}$$

Moreover, its spectral radius $r(W_\infty)$ is equal to its induced norm $\|W_\infty\|_0$, which is hence characterised by the relation

$$r(W_\infty) = \|W_\infty\|_0 = \min\{<v, W_\infty v>_0 : <v, v>_0 = 1\}. \tag{7}$$

Proof: First, write $W_\infty = I - \Gamma$. It is easily seen that Γ is self-adjoint and positive definite in the $<\cdot, \cdot>_0$ topology. Its spectrum (eigenvalues) is (are) hence real and strictly positive. Let $\Gamma v = \lambda v$ with $v \neq 0$ and write, using algebraic manipulation and the Cauchy Schwarz inequality.

$$\begin{aligned}\frac{1}{1+\overline{\delta}^2}\|v\|_0^2 \le \lambda\|v\|_0^2 &= <v, \Gamma v>_0 \\ &= <G_0^*(G_0G_0^*)^{-1}v, (I+L)^{-1}G_0^*(G_0G_0^*)^{-1}v>_{\mathcal{U}} \\ &\le \frac{1}{1+\underline{\delta}^2}\|v\|_0^2,\end{aligned} \tag{8}$$

where the bounds on L have been used to write $\frac{1}{1+\overline{\delta}^2} I \le <u, (I+L)^{-1}u>_{\mathcal{U}} \le \frac{1}{1+\underline{\delta}^2} <u, u>_{\mathcal{U}}, \forall$ non-zero $u \in \mathcal{U}$. The result follows by converting this inequality into bounds on $1 - \lambda$. □

Note: The results can be refined if L is positive definite and $\underline{\delta}^2 I < L \le \overline{\delta}^2 I$ with $0 \le \underline{\delta}^2 < \overline{\delta}^2$. In this case the spectrum lies in the open interval (0, 1) and each eigenvalue satisfies

$$0 \le \frac{\underline{\delta}^2}{1+\underline{\delta}^2} < \lambda \le \frac{\overline{\delta}^2}{1+\overline{\delta}^2} < 1. \tag{9}$$

Finally, the bounded operator G_0 has some simple properties of value in the following analysis.

Lemma 2.4: *If $\mathcal{R}_e$ has dimension n_r, then for any chosen basis of $\mathcal{R}_e$, there exists vectors g_j, $1 \le j \le n_r$ such that the mapping $u \to G_0u$ maps u into the coordinates of G_0u following the rule:*

$$u \to \begin{bmatrix} <g_1, u>_{\mathcal{U}} \\ \vdots \\ <g_{n_r}, u>_{\mathcal{U}} \end{bmatrix} \tag{10}$$

Furthermore, let $\{u_k\}_{k \ge 0}$ be an infinite sequence in $\mathcal{U}$. Then, a sufficient condition for G_0u_k to converge to zero in the norm topology of $\mathcal{R}_e$ is that $u_k \to 0$ in the weak topology of $\mathcal{U}$.

Proof: Boundedness of G_0 is equivalent to continuity. The result then follows from the definitions of norm and weak topologies in Hilbert spaces and the Riesz Representation Theorem (see, for example, Luenberger, 1969). □

3. Linear state space systems: intermediate point tracking problems

The problem definition described in the previous section has the advantage of generality but needs to be tailored to meet the needs of a specific application. In the next subsections, an application to continuous state space systems is defined in detail and application to single and multi-rate discrete systems outlined. The presentation starts with the construction of the tracking system from a basic underlying state space model $S(A, B, C)$, proceeds to define the relevant spaces, the operators G, G_0, G_0^* and G_1 and then to compute $G_0G_0^*$ which is a matrix that is needed in the resultant algorithms.

3.1. Continuous time systems

3.1.1. A general intermediate point tracking system model

The following description demonstrates how the intermediate point tracking objective of attaining desired output values at specific points in a time interval $[0, T]$ can be formulated as a special case of the general problem with specific operators G, G_0, G_1 and spaces $\mathcal{Y}, \mathcal{U}, \mathcal{Z}, \mathcal{R}_e$. Let $S(A, B, C)$ be a *strictly* proper m-output, ℓ-input, state dimension n, linear, time-invariant system written in the form,

$$\dot{x}(t) = Ax(t) + Bu(t), x(0) = x_0, y(t) = Cx(t) \tag{11}$$

or, equivalently, in the operator form $y = Gu + d$, where $G : L_2^\ell[0, T] \to L_2^m[0, T]$, $y, d \in L_2^m[0, T]$ and $u \in L_2^\ell[0, T]$ i.e. $\mathcal{U} = L_2^\ell[0, T]$ and $\mathcal{Y} = L_2^m[0, T]$. Here $T < \infty$ is assumed to be fixed. The relevant convolution operator G and signal d appearing in Equation (1) are hence defined by

$$(Gu)(t) = \int_0^t Ce^{A(t-s)}Bu(s)\,\mathrm{d}s,$$
$$d(t) = Ce^{At}x_0, \quad t \in [0, T]. \tag{12}$$

Note: The output y(t) is assumed to contain all measured variables whose values will be specified by the tracking problem at some or all intermediate times. For example, if positions y and velocities $\dot{y}$ are to be specified in a mechanical problem (e.g. a robot required to be momentarily stationary at each location in a component assembly task) and CB = 0 then the velocities can be added to an output y(t) = Cx(t) by the map,

$$y \to \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, C \to \begin{bmatrix} C \\ CA \end{bmatrix} \text{ and } m \to 2m. \tag{13}$$

Note: G is always bounded as T is finite even if it is unstable. It is natural to assume however that S(A, B, C) is asymptotically stable as algorithms will normally be implemented in conjunction with a stabilising feedback controller.

Let $0 < t_1 < t_2 < \cdots < t_M = T$ be M distinct intermediate points in $[0, T]$ needed to define the task. For any $f \in L_2^m[0, T]$, consider the linear map $f \mapsto f^e$ with

$$f^e = \begin{bmatrix} F_1 f(t_1) \\ \vdots \\ F_M f(t_M) \end{bmatrix} \in R^{f_1} \times \cdots \times R^{f_M} = \mathcal{R}_e$$

where each F_j is $f_j \times m$ and of full row rank. Clearly $\mathcal{R}_e$ can be identified with R^{n_r} with $n_r = f_1 + \cdots f_M$.

Note: Including the F_j permits the designer to require only that selected elements or linear combinations of elements of f that are important in the tracking objective be specified at each time $t = t_j$, e.g. a system may need to arrive at the end-time at a specific spatial point with a desired velocity but need only to pass through a number of given spatial points at some intermediate times (velocities unspecified).

With this notation, the 'extended output' y^e from the plant is defined to be the values $F_j y(t_j)$, $1 \le j \le M$ at intermediate points, i.e. the dynamics in Equation (2) can be modelled using

$$y^e = G_0 u + d_0,$$

$$G_0 u = \begin{bmatrix} G_1^e u \\ \vdots \\ G_M^e u \end{bmatrix}, \quad d_0 = \begin{bmatrix} F_1 d(t_1) \\ \vdots \\ F_M d(t_M) \end{bmatrix} \tag{14}$$

with $G_0 : L_2^\ell[0, T] \to R^{f_1} \times \cdots \times R^{f_M}$ a linear operator. As there is no need to define a reference trajectory for all points of $[0, T]$, the reference signal is defined as $r^e = [(r_1^e)^T, \ldots, (r_M^e)^T]^T \in \mathcal{R}_e$ where r_j^e is the desired value of $F_j y(t_j)$ at time t_j, $1 \le j \le M$. This notation is consistent with the above by noting that r^e is generated by the map $r \to r^e$ for any nominal $r(t)$ that has the desired values at the intermediate points. With this definition, the tracking error is constructed from $e^e = r^e - y^e = (r - y)^e$. Each operator $G_j^e : L_2^\ell[0, T] \to R^{f_j}$ in Equation (14) is constructed from G and defined by the relation,

$$G_j^e u = F_j(Gu)(t_j) = F_j \int_0^{t_j} Ce^{A(t_j - t)}Bu(t)\,\mathrm{d}t. \tag{15}$$

3.1.2. *Computing the adjoint operators and $G_0 G_0^*$*

First the relevant adjoint operator G_0^* of G_0 is computed. Take the product space $\mathcal{R}_e = R^{f_1} \times \cdots \times R^{f_1}$ to be a Hilbert space with inner product,

$$<(v_1, \ldots, v_M), (w_1, \ldots, w_M)>_{[Q]} = \sum_{j=1}^{M} v_j^T Q_j w_j,$$

where the $f_j \times f_j$ matrices Q_j, $1 \le j \le M$ are symmetric and positive definite. $[Q]$ is used to denote the data set $\{Q_1, \ldots, Q_M\}$ and the squared error norm is $\|e^e\|_{[Q]}^2 = <e^e, e^e>_{[Q]}$. Also the input signal space $\mathcal{U}$ is a real Hilbert space with inner product and norm

$$<u, v>_R = \int_0^T u^T(t)Rv(t)\,\mathrm{d}t \quad \text{where} \quad R = R^T > 0$$
$$\text{and} \quad \|u\|_R = \sqrt{\int_0^T u^T(t)Ru(t)\,\mathrm{d}t}. \tag{16}$$

Note: For notational transparency in the state space examples, the inner products have been given subscripts corresponding to the weighting matrices used.

Noting that

$$<(w_1, \ldots, w_M), G_0 u>_{[Q]} = <G_0^*(w_1, \ldots, w_M), u>_R \tag{17}$$

by definition, then the adjoint is computed from adjoints of G_j^e, $1 \le j \le M$.

(1) *Adjoint operator of G_j^e:* Consider G_j^{e*} via the identity

$$w_j^T Q_j F_j \int_0^{t_j} Ce^{A(t_j-t)}Bu(t)\,\mathrm{d}t = \int_0^{t_j} (R^{-1}B^T e^{A^T(t_j-t)}C^T F_j^T Q_j w_j)^T Ru(t)\,\mathrm{d}t.$$

It can hence be deduced that

$$(G_j^{e*}w_j)(t) = \left\{ \begin{array}{ll} R^{-1}B^T e^{A^T(t_j-t)}C^T F_j^T Q_j w_j; & 0 \le t \le t_j \\ 0; & t > t_j \end{array} \right\}, \tag{18}$$

which can be computed from the relation $P_j(t) = 0$, $t \in (t_j, T]$ and, on $[0, t_j]$, from

$$\begin{aligned} \dot{P}_j(t) &= -A^T P_j(t),\ P_j(t_j) \\ &= C^T F_j^T Q_j, (G_j^{e*}w_j)(t) \\ &= R^{-1}B^T P_j(t) w_j. \end{aligned} \tag{19}$$

(2) *Adjoint operator of G_0:* Using the above representations, the adjoint operator of G_0 is the map $(w_1, \ldots, w_M) \mapsto u$ defined by

$$\begin{aligned} u(t) &= \sum_{j=1}^{M} (G_j^{e*}w_j)(t) \\ &= R^{-1}B^T \sum_{j=1}^{M} P_j(t) w_j. \end{aligned} \tag{20}$$

(3) *Finally, the matrix $G_0G_0^*$* is obtained as an $n_r \times n_r$ block matrix with $f_i \times f_j$ (i, j)th block given by

$$\begin{aligned} (G_0G_0^*)_{ij} &= \int_0^{\min(t_i,t_j)} F_i Ce^{A(t_i-t)} \\ &\quad \times BR^{-1}B^T e^{A^T(t_j-t)}C^T F_j^T Q_j\,\mathrm{d}t \\ &= Q_i^{-1} \int_0^{\min(t_i,t_j)} P_i^T(t) R P_j(t)\,\mathrm{d}t. \end{aligned} \tag{21}$$

3.1.3. The auxiliary system

The auxiliary system $z = G_1u + d_1$ of Equation (3) is assumed to be described by an ℓ-input, m_z-output *proper* state space model $S(A_z, B_z, C_z, D_z)$ of state dimension n_z, i.e.

$$\begin{gathered} \dot{x}_z(t) = A_z x_z(t) + B_z u(t),\ \ z(t) = C_z x_z(t) + D_z u(t) \\ x_z(t) \in \mathcal{R}^{n_z},\ x_z(0) = x_{z0} \end{gathered} \tag{22}$$

with G_1 and d_1 defined as

$$\begin{aligned} (G_1u)(t) &= D_z u(t) + \int_0^t C_z e^{A_z(t-s)} B_z u(s)\,\mathrm{d}s, \\ d_1(t) &= C_z e^{A_z t} x_{z0}, \quad t \in [0, T]. \end{aligned} \tag{23}$$

Choosing $\mathcal{Z} = L_2^{m_z}[0, T]$ with inner product and norm defined by

$$\begin{aligned} <z_1, z_2>_Q &= \int_0^T z_1^T(t) Q z_2(t)\,\mathrm{d}t \ \ \& \\ \|z\|_Q &= \sqrt{\int_0^T z^T(t) Q z(t)\,\mathrm{d}t}, \end{aligned} \tag{24}$$

(where the $m_z \times m_z$ matrix $Q = Q^T > 0$) then the auxiliary optimisation problem is the minimisation, subject to both the auxiliary dynamic equations and the tracking requirement, of the objective function,

$$\begin{aligned} J(z, u) = \int_0^T \big[&(z(t) - z_0(t))^T Q (z(t) - z_0(t)) \\ &+ (u(t) - u_0(t))^T R (u(t) - u_0(t))\big]\,\mathrm{d}t \end{aligned} \tag{25}$$

indicating the secondary objective of minimising a weighted combination of an input energy measure and the variation of the auxiliary signal. The relative weights of these objectives are reflected in the choices of the $m_z \times m_z$ matrix Q and $\ell \times \ell$ matrix R.

In summary, for the continuous time case the tracking requirement (2), auxiliary dynamics (3) and objective function (4) are, respectively, represented by Equations (14)–(15), (23) and (25).

3.2. Discrete time state space systems

The relevant models and operators for this case are summarised in Appendix where it is seen that both uni-rate (Appendix 1) and multi-rate (Appendix 2) systems can be included in the general operator formulation favoured in this paper. In particular, the finite dimensionality assumptions are automatically valid in these cases even if full tracking at *all* sample points is required. The invertibility assumptions do still need to be checked however and the computations for operators and adjoints, etc. differ in computational detail. Other useful practical observations include:

(1) Discrete time tracking of a reference signal at each sample point is just NOILC (Amann et al., 1996b), which fits into the formulation described by choosing $G_0 = G$ and $\mathcal{R}_e = \mathcal{Y}$. The idea of auxiliary optimisation only makes sense here if the input u that enables the tracking to be achieved is non-unique. This is the case if $\ell > m$.

(2) The situation of discrete time delay systems is easily included in the formulation if delays are synchronised with the samples and every delay is an integer multiple of the sample length. In such a case the delay is removed by replacing it by an extended state description.

3.3. *A comment on the choice of a state space auxiliary system*

The auxiliary system choice is infinite (in both continuous time and discrete time cases) and must reflect the needs of the application. A number of possible other interpretations are illustrated below using the notation of the above and Appendix.

(1) G_1 may reflect the map between inputs and other states of interest to the control problem but this is not necessary for applications.
(2) More generally, the choice of G_1 could be based upon a perceived need to shape the frequency content of the input signal. This can be seen by assuming that $d_1 = 0$ and $u_0 = 0$ and writing the auxiliary objective function in the form,

$$J(z,u) = \|z - z_0\|^2_{\mathcal{Z}} + \|u - u_0\|^2_{\mathcal{U}} = <u, (I + G_1^* G_1)u>_U. \quad (26)$$

Assuming that it is possible to do the factorisation $(I + G_1^* G_1) = G_f^* G_f$, then the auxiliary optimisation problem can be seen to be that of minimising the norm of the 'filtered' input $G_f u$. This argument can be reversed, starting with a desired G_f and using it to construct G_1. Details of the exploitation of this idea are left for future research output.
(3) In both discrete time and continuous time state space situations, suppose that $u_0 = 0$, $d_1 = 0$ and $Q = R_z$ where R_z is symmetric and positive definite. Also assume that G_1 can be identified with the matrix identity $G_1 = I$. Then the auxiliary objective function becomes simply $J = \|u\|^2_{R+R_z}$. In this case the methodology can be interpreted as an algorithm that uses an input weight R during iterations to achieve tracking whilst minimising control energy as measured by $J = \|u\|^2_{R+R_z}$.

4. Switching algorithms - general theory

The paper now considers the iterative solution of the constrained optimisation problem (5) in a manner that meets the needs of ILC. The structure of the proposed algorithms is designed to address, in one algorithm, the TWO objectives of eliminating the tracking errors $e^e = r^e - y^e$ (where $y^e = G_0 u + d_0$ is the intermediate point output achieved by the input signal u) whilst also achieving the desired auxiliary minimisation of $J(z, u)$. In addition, the aim is to design algorithms that use simple computational tools from linear quadratic optimisation that, for state space systems, can be realised as familiar linear quadratic optimal control methodologies. Both on-line and off-line computational implementations are then possible.

The two algorithms now presented are based on the idea of switching between constraints in the sense that each iteration consists of two components. The two components of each iteration for the first algorithm (Switching Algorithm 1) respectively solve the tracking problem exactly and then minimise a modified auxiliary objective function. The second algorithm (Switching Algorithm 2) is a relaxation of the first that allows approximate satisfaction of the tracking constraint at each iteration whilst retaining the asymptotic convergence properties of the first algorithm. This approximation takes the form of replacing the exact solution by a series of $k_0 \geq 1$ iterations of a NOILC problem based on G_0 and a start condition obtained from the auxiliary system iterations.

4.1. *Switching Algorithm 1*

The algorithm is given as follows and presupposes that r^e, Q, $[Q]$ and R have been specified, that the two factors u_0, z_0 appearing in objective function (4) have been chosen and that G_0^* and $G_0 G_0^*$ have been computed:

Switching Algorithm 1

1. Set the iteration index $k = 0$ and choose $u_k = u_0$
2. Find the plant output $y_k^e = G_0 u_k + d_0$ and compute $e_k^e = r^e - y_k^e$.
3. Solve the optimisation problem

$$\tilde{u}_{k+1} = \text{argmin}\{\|u - u_k\|^2 : r^e = G_0 u + d_0\} \text{ using the off-line formula } \tilde{u}_{k+1} = u_k + G_0^*(G_0 G_0^*)^{-1} e_k^e. \quad (27)$$

Note the need for invertibility of the matrix $G_0 G_0^*$.
4. Solve the optimisation problem

$$(u_{k+1}, z_{k+1}) = \text{argmin}\big\{\|z - z_k\|^2_{\mathcal{Z}} + \|u - \tilde{u}_{k+1}\|^2_{\mathcal{U}} : z = G_1 u + d_1\big\}. \quad (28)$$

5. Replace k by $k + 1$ and go to step 2.

Note that step 4 is computationally identical to the first iteration of a NOILC control problem with 'output' equal to the auxiliary signal z, a reference signal 'z_k' and first input choice of '$\tilde{u}_{k+1}$'. It can be computed off-line or on-line, and in the latter case a combined feedback and (predictive) feedforward structure has been found to be more robust

to model uncertainty (Owens et al., 2012). The required formulation is provided by Amann et al. (1996b), and for clarity the exact formulae involved are given by Equations (77)–(79).

Finally, there is an option to terminate the algorithm at any point if required but, for analysis purposes, it is assumed that the iterations are not terminated. For simplicity, the algorithm is written assuming that the control input u_{k+1} is the new input applied to the plant on iteration $k+1$ but the input $\tilde{u}_{k+1}$ could be used as an alternative.

4.2. Analysis of Switching Algorithm 1

To analyse the behaviour of the algorithm, note that steps 3 and 4 give rise to the characterisation,

$$\begin{aligned} u_{k+1} &= \tilde{u}_{k+1} + G_1^*(z_k - z_{k+1}) \\ &= u_k + G_1^*(z_k - z_{k+1}) + G_0^*(G_0G_0^*)^{-1}e_k^e \\ &= u_k + G_1^*G_1(u_k - u_{k+1}) + G_0^*(G_0G_0^*)^{-1}e_k^e, \end{aligned} \quad (29)$$

which gives

$$u_{k+1} = u_k + (I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}e_k^e \quad (30)$$

and hence

$$\begin{aligned} e_{k+1}^e &= W_\infty e_k^e, \quad \text{where} \\ W_\infty &= (I - G_0(I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}), \end{aligned} \quad (31)$$

which is a matrix iteration in the finite dimensional space $\mathcal{R}_e$. Using Lemmas 2.1–2.3 and identifying L with $G_1^*G_1$ immediately yields the convergence theorem

Theorem 4.1: *Using the above construction, W_∞ has a real spectrum lying in the half-open interval [0, 1) and the tracking error sequence $\{e_k^e\}_{k\geq 0}$ satisfies the condition,*

$$e_\infty^e = \lim_{k\to\infty} e_k^e = 0 \quad \left(\textit{i.e. } y_\infty^e = \lim_{k\to\infty} y_k^e = r^e\right) \quad (32)$$

in any norm topology of $\mathcal{R}_e$. In particular,

$$\|e_{k+1}^e\|_0 \leq \lambda\|e_k^e\|_0 \forall k \geq 0 \ \ (\textit{where } \lambda = \|W_\infty\|_0 < 1) \quad (33)$$

and e_k^e converges to zero as $\|e_k^e\|_0 \leq \lambda^k\|e_0^e\|_0 \to 0$ as $k \to \infty$. Finally,

$$\lim_{k\to\infty} \|\tilde{u}_k - u_k\|_{\mathcal{U}} = 0. \quad (34)$$

(1) Note that the tracking error is monotonic in the error norm $\|e\|_0$, which depends upon the original inner product chosen for $\mathcal{R}_e$ modified by $(G_0G_0^*)^{-1}$. It is therefore not necessarily monotonic in the commonly used Euclidean norm although the user may influence convergence properties through choice of $[Q]$, Q and R.

(2) Also, as each $\tilde{u}_k$ solves the tracking problem, an important outcome of this result is that of consistency i.e. either of the sequences $\{u_k\}_{k\geq 0}$ or $\{\tilde{u}_k\}_{k\geq 0}$ asymptotically solve the problem.

To derive additional properties of the algorithm, write the input iteration in the form

$$u_{k+1} = u_k + (I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}W_\infty^k e_0^e \quad (35)$$

which, as W_∞ has spectral radius <1, proves the existence of the limit $u_\infty = \lim_{k\to\infty} u_k$ generated from the formula,

$$\begin{aligned} u_\infty - u_0 &= (I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1} \\ \sum_{k=0}^{\infty} W_\infty^k e_0^e &= (I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}(I - W_\infty)^{-1}e_0^e, \end{aligned} \quad (36)$$

where u_∞ hence generates the auxiliary signal $z_\infty = G_1u_\infty + d_1$ where

$$\begin{aligned} \lim_{k\to\infty} z_k = z_\infty = z_0 + G_1(I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1} \\ \times (I - W_\infty)^{-1}e_0^e. \end{aligned} \quad (37)$$

Writing the above as

$$\begin{aligned} (I + G_1^*G_1)(u_\infty - u_0) = G_0^*\lambda_0 \quad \text{where} \\ \lambda_0 = (G_0G_0^*)^{-1}(I - W_\infty)^{-1}e_0^e \in \mathcal{R}_e \end{aligned} \quad (38)$$

or using $z = G_1u + d_1$

$$u_\infty - u_0 = G_0^*\lambda_0 + G_1^*\lambda_1, \quad \text{where } z_\infty - z_0 = -\lambda_1. \quad (39)$$

It is immediately concluded that

Theorem 4.2: *Under the condition defined above, the pair (u_∞, z_∞) is a stationary point of the Lagrangian*

$$\begin{aligned} \mathcal{L}(u, \lambda_0) = J(z, u) + 2 < \lambda_0, r^e - G_0u - d_0 >_{\mathcal{R}_e} \\ + 2 < \lambda_1, z - G_1u - d_1 >_{\mathcal{Z}} \end{aligned} \quad (40)$$

(with multipliers $\lambda_0 \in \mathcal{R}_e$, $\lambda_1 \in \mathcal{Z}$) for the linearly constrained optimisation problem with

$$\textit{Objective Function } J(z, u) = \|z - z_0\|_{\mathcal{Z}}^2 + \|u - u_0\|_{\mathcal{U}}^2 \quad (41)$$

and Constraints $\{u \in \mathcal{U}, r_e = G_0u + d_0, z = G_1u + d_1\}$. (42)

Furthermore, the sequence $\{u_k\}_{k\geq 0}$ (and hence $\{\tilde{u}_k\}_{k\geq 0}$) asymptotically solves the associated ILC tracking problem with the auxiliary optimisation criterion posed in Section 2.

Proof: The linearity of the constraints and the assumption that $\mathcal{R}[G_0] = \mathcal{R}_e$ indicate that all points of $\mathcal{Z} \times \mathcal{U}$ are *regular points* (see, for example, Luenberger, 1969) of the constraints. This condition, together with the observation that the objective function is convex and the constraint set is both closed and convex in $\mathcal{Z} \times \mathcal{U}$, then proves that (u_∞, z_∞) solves the required, constrained optimisation problem. □

4.3. Switching Algorithm 1: convergence rates and parameter choices

The norms in $\mathcal{U}$ and $\mathcal{Z}$ are assumed to be chosen by the control design engineer to reflect the desired relative weighting between $\|z - z_0\|^2_{\mathcal{Z}}$ and $\|u - u_0\|^2_{\mathcal{U}}$ in the objective function (4). To illustrate the influence of this relative weighting replace (4) by the modified weighting

$$J_\epsilon(z, u) = \|z - z_0\|^2_{\mathcal{Z}} + \epsilon^2 \|u - u_0\|^2_{\mathcal{U}}, \tag{43}$$

where $\epsilon^2 > 0$ is a parameter introduced to reveal the effects of a simple change in the relative weights of the two terms in the objective function.

The effect on the previous theory is simply that the operator G_1^* is replaced by $\epsilon^{-2}G_1^*$ and, in particular, W_∞ is replaced by

$$W_\infty(\epsilon) = (I - G_0(I + \epsilon^{-2}G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}). \tag{44}$$

$W_\infty(\epsilon)$ has all the properties of W_∞. It also has relevant properties for algorithm performance as reflected in the following result:

Theorem 4.3:

$$\lim_{\epsilon\to\infty} W_\infty(\epsilon) = 0 \tag{45}$$

and, if ker[G_1] = {0},

$$\lim_{\epsilon\to 0+} W_\infty(\epsilon) = I \quad (\textit{the Identity}) \tag{46}$$

In particular, the spectral radius r($W_\infty(\epsilon)$) then has the properties

$$\lim_{\epsilon\to\infty} r(W_\infty(\epsilon)) = 0 \ \ \& \ \ \lim_{\epsilon\to 0+} r(W_\infty(\epsilon)) = 1 \tag{47}$$

(which is interpreted as stating that convergence will be slow if ϵ is 'very small' and very rapid if ϵ is 'large', i.e. the algorithm convergence is, in general, expected to be faster if greater emphasis is placed on control input energy reduction in the auxiliary optimisation problem).

Proof: The first limit follows by first letting $\epsilon^2 > \|G_1^*G_1\|_{\mathcal{U}}$ and then using the expansion $(I + X)^{-1} = I + X + X^2 + X^3 + \cdots$ for any bounded operator X with norm strictly less than unity to show that $W_\infty(\epsilon) = O(\epsilon^{-2}) \to 0$ as $\epsilon \to \infty$. To prove the second limit, let $r \in \mathcal{R}_e$ be arbitrary and nonzero. Set $v = G_0^*(G_0G_0^*)^{-1}r \in \mathcal{U}$ and examine the relation $u_\epsilon = (I + \epsilon^{-2}G_1^*G_1)^{-1}v$ in the form $\epsilon^2 u_\epsilon + G_1^*G_1u_\epsilon = \epsilon^2 v$ i.e., using the Cauchy Schwarz inequality,

$$\epsilon^2\|u_\epsilon\|_{\mathcal{U}} + \|G_1u_\epsilon\|^2_{\mathcal{Z}} = \epsilon^2 <u_\epsilon, v>_{\mathcal{U}} \leq \epsilon^2\|u_\epsilon\|_{\mathcal{U}}\|v\|_{\mathcal{U}}$$

so that $\|u_\epsilon\|_{\mathcal{U}} \leq \|v\|_{\mathcal{U}} \forall \epsilon$. Also, it follows that $\|G_1u_\epsilon\|^2_{\mathcal{Z}} \leq \epsilon^2\|u_\epsilon\|_{\mathcal{U}}\|v\|_{\mathcal{U}} \leq \epsilon^2\|v\|^2_{\mathcal{U}} \to 0\,(\text{as}\,\epsilon \to 0)$. Now let $\psi \in \mathcal{Z}$ be arbitrary and examine the relation $<\psi, G_1u_\epsilon>_{\mathcal{Z}} = <G_1^*\psi, u_\epsilon>_{\mathcal{U}} \to 0\,(\text{as}\,\epsilon \to 0)$. The assumption that $\ker[G_1] = \{0\}$ implies that G_1^* has dense range (see, for example, Luenberger, 1969). This, together with the proved fact that $\|u_\epsilon\|_{\mathcal{U}} \leq \|v\|_{\mathcal{U}} \forall \epsilon$ then proves that u_ϵ converges to zero (as $\epsilon \to 0$) in the weak topology of $\mathcal{U}$. Writing

$$(I - W_\infty(\epsilon))r = G_0(I + G_1^*G_1)^{-1}G_0^*(G_0G_0^*)^{-1}r = G_0u_\epsilon$$

and using the finite dimensionality of $\mathcal{R}_e$ to identify G_0 as a matrix with 'rows' in $\mathcal{U}$ (see Lemma 2.4) indicates that

$$\lim_{\epsilon\to 0+} W_\infty(\epsilon)r = r, \tag{48}$$

which ends the proof of the result as r is arbitrary and $\mathcal{R}_e$ is finite dimensional. □

4.4. Switching Algorithm 2

Algorithm 1 assumes that the optimisation in step 3 will be solved using the given formula (27) for $\tilde{u}_{k+1}$ (or an equivalent computation). This requirement can be relaxed by replacing this step by a finite iteration process to yield the modified algorithm written in the form of ILC 'outer' iterations with index k for the sequence $\{u_k\}_{k\geq 0}$, etc. but, where each such iteration consists of k_0 'inner iterations' (with index j) used in the computation of $\tilde{u}_{k+1}$. This is still an ILC process but the use of two iteration indices simplifies the presentation of the methodology. In addition to providing control over the input change, the use of inner iterations means that the tracking problem can be addressed on-line over several trials, e.g. by exploiting a feedback plus (predictive) feedforward implementation. Intuitively, this could lead to greater robustness in practice in dealing with plant-model mismatch. The algorithm proposed is as follows.

The 'outer iteration' process:

1. Set the iteration index $k = 0$ and choose $u_k = u_0$ and an integer $k_0 \geq 1$.
2. Undertake k_0 'inner iterations' using the NOILC problem for the system $y^e = G_0 u + d_0$ with reference signal r^e and initial input choice u_k (a more detailed description is given following this algorithm statement). These iterations can be undertaken on-line.
3. Set $\tilde{u}_{k+1}$ to be equal to the final control signal $\tilde{u}_{k+1}^{(k_0)}$ generated by this process.
4. As in Switching Algorithm 1, solve the optimisation problem

$$(u_{k+1}, z_{k+1}) = \text{argmin}\{\|z - z_k\|_{\mathcal{Z}}^2 + \|u - \tilde{u}_{k+1}\|_{\mathcal{U}}^2 : z = G_1 u + d_1\} \quad (49)$$

   for the control signal u_{k+1}.
5. Replace k by $k + 1$ and go to step 2.

The Inner Iterations: Step 2 in the above can be described in more detail as follows:

1. Set an inner iteration index $j = 0$ and initiate inner iterations from the starting condition $\tilde{u}_{k+1}^{(j)} = u_k$ and $e_{(k+j)}^e = e_k^e$.
2. Compute the new input signal

$$\tilde{u}_{k+1}^{(j+1)} = \tilde{u}_{k+1}^{(j)} + G_0^*(I + G_0 G_0^*)^{-1} e_{(k+j)}^e \quad (50)$$

   obtained by minimising

$$\left\|e_{(k+j+1)}^e\right\|_{\mathcal{R}_e}^2 + \left\|\tilde{u}_{k+1}^{(j+1)} - \tilde{u}_{k+1}^{(j)}\right\|_{\mathcal{U}}^2, \quad (51)$$

   where the new output $y_{(k+j+1)}^e = G_0 \tilde{u}_{k+1}^{(j+1)} + d_0$ and the resultant tracking error $e_{(k+j+1)}^e = r^e - y_{(k+j+1)}^e$.
3. Replace j by $j + 1$.
4. If $j = k_0$, end the inner iteration process. Otherwise return to step 2.

Note: Step 2 of the inner iterations can be done either off-line by simulation or on-line using experimentation. In the latter case, a combined feedback and (predictive) feedforward structure embeds the potential for greater robustness to model uncertainty. A solution has recently been published (Owens et al., 2012), which makes this possible in the continuous time case, and, for completeness, the exact formulae required are given by (82)–(86).

4.5. Analysis of Switching Algorithm 2

The analysis is similar to that of Switching Algorithm 1. The crucial difference is the inner iteration process. It is easily seen from NOILC that, $\forall j \geq 0$,

$$\begin{aligned} \tilde{u}_{k+1}^{(j+1)} &= \tilde{u}_{k+1}^{(j)} + G_0^* e_{(k+j+1)}^e \quad \text{and} \quad \text{hence} \\ e_{(k+j+1)}^e &= (I + G_0 G_0^*)^{-1} e_{(k+j)}^e, \\ \text{i.e.} \quad e_{(k+j)}^e &= (I + G_0 G_0^*)^{-j} e_k^e \end{aligned} \quad (52)$$

Hence, using induction, and $\tilde{u}_{k+1}^{(0)} = u_k$

$$\tilde{u}_{k+1} = \tilde{u}_{k+1}^{(k_0)} = u_k + G_0^* \sum_{j=0}^{k_0 - 1} (I + G_0 G_0^*)^{-(j+1)} e_k^e. \quad (53)$$

Whenever the matrix $G_0 G_0^*$ is positive definite, all eigenvalues of $(I + G_0 G_0^*)^{-1}$ are strictly less than unity and hence, summing the geometric series,

$$\begin{aligned} \tilde{u}_{k+1} = u_k &+ G_0^*(I + G_0 G_0^*)^{-1}(I - (I + G_0 G_0^*)^{-1})^{-1} \\ &\times (I - (I + G_0 G_0^*)^{-k_0}) e_k^e, \end{aligned} \quad (54)$$

which yields

$$\tilde{u}_{k+1} = u_k + G_0^*(G_0 G_0^*)^{-1}(I - (I + G_0 G_0^*)^{-k_0}) e_k^e. \quad (55)$$

As $\lim_{k_0 \to \infty}(I + G_0 G_0^*)^{-k_0} = 0$, it is seen that Algorithm 2 is equivalent to Algorithm 1 when $k_0 = \infty$.

The use of finite values of k_0 can however affect performance. More precisely, using the above input signal in step 4 of Algorithm 2 and using similar algebra to that used in Algorithm 1 yields the results that, $\forall k \geq 0$,

$$e_{k+1}^e = W_{k_0} e_k^e \quad \text{and hence } e_k^e = W_{k_0}^k e_0^e \quad (56)$$

where the matrix

$$\begin{aligned} W_{k_0} &= I - G_0(I + G_1^* G_1)^{-1} G_0^*(G_0 G_0^*)^{-1} \\ &\quad \times (I - (I + G_0 G_0^*)^{-k_0}) \\ \text{i.e. } W_{k_0} &= W_\infty + (I - W_\infty)(I + G_0 G_0^*)^{-k_0} \end{aligned} \quad (57)$$

The rest of the analysis is identical to that of Switching Algorithm 1 with W_∞ replaced by W_{k_0} with identical results and properties provided that all eigenvalues of W_{k_0} have modulus strictly less than unity. Noting that

$$\lim_{k_0 \to \infty} W_{k_0} = W_\infty, \quad (58)$$

it follows that all of the properties of Switching Algorithm 1 are retained if k_0 is large enough. However, if k_0 is too small, then it might be expected that W_{k_0} could have eigenvalues ≥ 1 and hence that the algorithm could diverge. The following result proves that this is not the case.

Theorem 4.4: *Under the conditions defined above, every eigenvalue λ of W_{k_0} satisfies the inequality,*

$$|\lambda| \leq r(W_\infty) + (1 - r(W_\infty)) \left(\frac{(r((G_0 G_0^*)^{-1}))}{(1 + r((G_0 G_0^*)^{-1}))} \right)^{k_0} < 1. \tag{59}$$

That is $r(W_{k_0}) < 1$ for all $k_0 > 0$ and Switching Algorithm 2 converges for all choices of $k_0 > 0$.

Note: The bound on $r(W_{k_0})$ is monotonically decreasing to the value $r(W_\infty) = \|W_\infty\|_0$. This suggests that, although k_0 has no effect on final convergence, increasing k_0 will tend to increase convergence rate.

Proof: For any eigenvalue λ, using $\|K\|_0$ to denote the induced norm of an operator $K : \mathcal{R}_e \to \mathcal{R}_e$ in the topology induced by $<\cdot\,\cdot>_0$,

$$\begin{aligned} |\lambda| \leq \|W_{k_0}\|_0 &\leq \|W_\infty\|_0 + \|I - W_\infty\|_0 \|(I + G_0 G_0^*)^{-1}\|_0^{k_0} \\ &= r(W_\infty) + (1 - r(W_\infty)) \|(I + G_0 G_0^*)^{-1}\|_0^{k_0}. \end{aligned}$$

It is easily proved that $(I + G_0 G_0^*)^{-1}$ is positive definite and self-adjoint in the $<\cdot, \cdot>_0$ topology. Its norm is hence equal to its spectral radius which is just $\frac{g^2}{1+g^2}$, where g^2 is largest eigenvalue of $(G_0 G_0^*)^{-1}$. This completes the proof. □

Finally, the discussion of convergence rates and parameter choices also remains valid. In particular, if the parameter ϵ^2 is introduced into the inner iteration objective function, i.e.,

$$\begin{aligned} &\left\| e^e_{(k+j+1)} \right\|^2_{\mathcal{R}_e} + \left\| u^{(j+1)}_{k+1} - u^{(j)}_{k+1} \right\|^2_{\mathcal{U}} \\ &\quad \to \left\| e^e_{(k+j+1)} \right\|^2_{\mathcal{R}_e} + \epsilon^2 \left\| u^{(j+1)}_{k+1} - u^{(j)}_{k+1} \right\|^2_{\mathcal{U}}, \end{aligned} \tag{60}$$

then G_0^* is replaced by $\epsilon^{-2} G_0^*$ in all formulae. W_∞ is unchanged by this modification but

$$\lim_{\epsilon \to 0} W_{k_0}(\epsilon) = W_\infty. \tag{61}$$

That is, smaller values of k_0 can be used to achieve a given convergence rate if the weighting of the control energy in the inner iteration optimisation problem is reduced.

5. A discussion of algorithm robustness

A characterisation of robustness is possible in the special case where all computations are undertaken off-line but using observed plant tracking error data, i.e. a specific feedforward implementation.

Feedforward assumption for robustness analysis: *Suppose that both the computation of $\tilde{u}_{k+1}$ from u_k and e^e_k and the auxiliary optimisation problem generating u_{k+1} from z_k and $\tilde{u}_{k+1}$ are undertaken off-line but that each u_k is used on-line to generate the tracking error data e^e_k from the plant.*

It follows that differences between the plant and the model G_0 will affect each signal y^e_k, and hence e^e_k, in an 'unpredictable' way. In contrast, although there may be modelling errors in G_1 (as a representation of the behaviour of the actual values of z generated by the plant), these errors have no impact on the algorithm as the 'real' z measurements are not made and are not required for the computations. The algorithm may still converge to the desired tracking signal, but, if it does indeed converge, it is natural to ask the question whether or not it converges to a solution of the defined ILC problem with auxiliary optimisation defined in terms of the *model-based* values of the auxiliary variable z.

Using the feedforward assumption and, for simplicity, examining the case of Switching Algorithm 1, the main change to the theoretical analysis is that the data e^e_k used is generated from the plant. Let G_0 now denote the operator deduced from a model of the plant and denote the actual value of this operator by $G_0 + \Delta G_0$ where, again, ΔG_0 maps $\mathcal{U}$ into $\mathcal{R}_e$ and is bounded. In a similar manner, suppose that the plant replaces the model value d_0 by $d_0 + \Delta d_0$. As a consequence, the plant output $y^e_k = (G_0 + \Delta G_0)u_k + (d_0 + \Delta d_0)$ and the observed *on-line* plant error is now

$$e^e_k = r^e - y^e_k = r^e - (G_0 + \Delta G_0)u_k - (d_0 + \Delta d_0) \forall k \geq 0. \tag{62}$$

The relevant formulae used for analysis are as follows, namely, if u_k generates the error e^e_k from the actual plant *on-line*, then the next input u_{k+1} can be characterised via *off-line* computation of $\tilde{u}_{k+1}$,

$$\tilde{u}_{k+1} = u_k + G_0^*(G_0 G_0^*)^{-1} e^e_k$$

using the model G_0. This is then followed by the model-based *off-line* NOILC minimisation problem, which has the formal solution

$$\begin{aligned} u_{k+1} &= \tilde{u}_{k+1} + G_1^*(z_k - z_{k+1}) \\ &= u_k + G_1^* G_1 (u_k - u_{k+1}) + G_0^*(G_0 G_0^*)^{-1} e^e_k \\ \text{i.e. } u_{k+1} &= u_k + (I + G_1^* G_1)^{-1} G_0^*(G_0 G_0^*)^{-1} e^e_k \end{aligned}$$

It follows trivially, by operating on this with the real plant operator $G_0 + \Delta G_0$, that

$$e^e_{k+1} = (W_\infty + \Delta W_\infty) e^e_k, \forall k \geq 0, \tag{63}$$

where ΔW_∞ is the change induced by the plant-model mismatch. Note that the change to d_0 has no effect on the general form of the error evolution.

Analysis of the algorithm has already proved that the spectral radius $r(W_\infty)$ of the matrix W_∞ is strictly less

than unity and considerably less than unity if the weighting on control energy in the auxiliary optimisation problem is large.

The matrix perturbation ΔW_∞ takes the form,

$$\Delta W_\infty = -\Delta G_0(I + G_1^* G_1)^{-1} G_0^*(G_0 G_0^*)^{-1}. \quad (64)$$

Again, using $\|K\|_0$ to denote the induced norm of $K : \mathcal{R}_e \to \mathcal{R}_e$ in the topology induced by the inner product $<\cdot, \cdot>_0$, it is deduced that

Theorem 5.1: *Under the defined conditions above, the feedforward implementation of the Switching Algorithm 1 converges to a zero tracking error if the induced norm,*

$$\|\Delta W_\infty\|_0 = \|\Delta G_0(I + G_1^* G_1)^{-1} G_0^*(G_0 G_0^*)^{-1}\|_0 < 1 - r(W_\infty). \quad (65)$$

In particular, it is sufficient that

$$\|\Delta G_0\| = (sup\{<\Delta G_0 v, \Delta G_0 v>_0 : v \in \mathcal{U} \ \& \ <v, v>_{\mathcal{U}} = 1\})^{1/2} < (1 - r(W_\infty))(1 + \delta^2) \quad (66)$$

where δ^2 is any positive scalar in the range $0 \le \delta^2 \le \underline{\delta}^2$ where

$$\underline{\delta}^2 = inf\{<G_1 v, G_1 v>_{\mathcal{Z}} : v \in \mathcal{U} \ \& \ <v, v>_{\mathcal{U}} = 1\}. \quad (67)$$

Proof: The relation $e^e_{k+1} = (W_\infty + \Delta W_\infty) e^e_k$ indicates that it is only necessary to prove that, under the defined conditions, all eigenvalues of $W_\infty + \Delta W_\infty$ have modulus strictly less than unity. Note that all eigenvalues satisfy the inequality,

$$\begin{aligned} |\lambda| &\le \|W_\infty + \Delta W_\infty\|_0 \\ &\le \|W_\infty\|_0 + \|\Delta W_\infty\|_0 \\ &= r(W_\infty) + \|\Delta W_\infty\|_0 \end{aligned} \quad (68)$$

i.e. the first inequality is sufficient to ensure that $|\lambda| < 1$. Next, note that a sufficient condition for this to be true is that

$$\|\Delta G_0\| < \frac{1 - r(W_\infty)}{\|(I + G_1^* G_1)^{-1}\| \|G_0^*(G_0 G_0^*)^{-1}\|} \quad (69)$$

From the definitions,

$$\begin{aligned} \|G_0^*(G_0 G_0^*)^{-1}\|^2 &= \sup\{<G_0^*(G_0 G_0^*)^{-1} v, \\ &\quad G_0^*(G_0 G_0^*)^{-1} v>_{\mathcal{U}} : <v, v>_0 = 1\} \\ &= \sup\{<v, (G_0 G_0^*)^{-1} v>_{\mathcal{R}_e} : <v, v>_0 = 1\} = 1, \\ &\quad \text{and} \|(I + G_1^* G_1)^{-1}\| \le (1 + \underline{\delta}^2)^{-1}, \end{aligned}$$

which proves the result. □

Three useful observations can easily be made, namely:

(1) It is always possible to choose the value $\delta = 0$. This choice removes the need to consider the auxiliary system dynamics in the robustness analysis.
(2) If G_1 is generated from a continuous state space model $S(A_z, B_z, C_z)$, then, as high frequency inputs generate infinitesimally small outputs, it is expected that $\underline{\delta}^2 = 0$ so $\delta = 0$ is the only choice.
(3) Choosing $\delta^2 = 0$ and using the lower bound

$$\begin{aligned} 1 - r(W_\infty) &= \inf\{<v, (I - W_\infty)v>_0 : \\ &\quad v \in \mathcal{R}_e, <v, v>_0 = 1\} \\ &\ge \tfrac{1}{1+\overline{\delta}^2} \text{ for any} \\ \overline{\delta}^2 \ge \|G_1\|^2 &= \sup\{<G_1 v, G_1 v>_{\mathcal{Z}} : \\ &\quad v \in \mathcal{U} \ \& <v, v>_{\mathcal{U}} = 1\}, \end{aligned}$$

the robustness condition can be replaced by the simple inequality,

$$\|\Delta G_0\| < \frac{1}{1 + \overline{\delta}^2}, \quad (71)$$

which bounds permissible modelling errors simply in terms of a bound on the operator norm of G_1 in $\mathcal{U}$. G_0, R and $[Q]$ play a role in the numerical values appearing in the inequality as the defined measure of magnitude of ΔG_0 is defined in terms of the $<\cdot, \cdot>_0$ topology. Also Q and R affect the numerical value of $\overline{\delta}^2$.

(4) It is noted that these formulae may make it possible to provide more explicit numerical bounds on robustness. For example, if G_1 is an asymptotically stable ℓ-input, m_z output continuous state space system as in Section 3, then causality on [0, T], monotonicity of norms with respect to T and Parseval's theorem on [0, ∞) suggests the choice of $\overline{\delta}^2$ as

$$\|G_1\|^2 \le \overline{\delta}^2 = \sup_{\omega \ge 0} r(R^{-1} G_1^T(-i\omega) Q G_1(i\omega)), \quad (72)$$

where $G_1(s)$ is used to denote the transfer function matrix of the state space model of the operator G_1. The general effects of Q and R on robustness are now revealed, i.e. robustness tends to reduce as the input weighting R reduces (or Q increases). It also suggests that, for a given choice of weights, the form of the frequency response $G_1(i\omega)$ will have an effect, e.g. significant resonances will reduce robustness by increasing $\overline{\delta}^2$.

The final question to be answered is 'what is the nature of the limit of the algorithm in the presence of such modelling errors?' This is answered by the following theorem.

Theorem 5.2: *Under the conditions of the previous theorem, the feedforward implementation of Switching Algorithm 1 in the presence of the defined modelling error converges to a solution (u_∞, z_∞, λ_0, λ_1) of the equations,*

$$\begin{aligned} u_\infty - u_0 &= G_0^*\lambda_0 + G_1^*\lambda_1, \\ z_\infty - z_0 &= -\lambda_1, z_\infty = G_1 u_\infty + d_1 \end{aligned} \tag{73}$$

and

$$r^e - (G_0 + \Delta G_0)u_\infty - (d_0 + \Delta d_0) = 0. \tag{74}$$

Proof: The proof follows in a similar manner to Theorem 4.2, replacing W_∞ by $W_\infty + \Delta W_\infty$ to prove the existence of the limits to the input sequence and auxiliary variable sequence. The relevant value of λ_0 is just

$$\lambda_0 = (G_0 G_0^*)^{-1}(I - W_\infty - \Delta W_\infty)^{-1} e_0^e,$$

which completes the outline proof. □

These equations are just perturbations of the stationary point conditions for the Lagrangian associated with the optimisation problem,

$$\min_{u\in\mathcal{U}}\{J(z,u) : u \in \mathcal{U}, r_e = G_0 u + d_0, z = G_1 u + d_1\} \tag{75}$$

suggesting that the converged solution no longer minimises the auxiliary objective function but, if modelling errors are small, will be a good approximation.

6. Experimental verification using intermediate point tracking with auxiliary optimisation

The paper now considers the experimental evaluation of the algorithms for the continuous time intermediate point problem considered in Section 3. The results are evaluated in this section by first summarising the equations that must be solved/implemented in each iteration and then presenting the results of experiments that not only verify the predictions of the theory but also support the theoretical prediction of good algorithm robustness.

6.1. Computational procedures for continuous systems

Once the intermediate points have been identified, the F_j identified, the desired auxiliary variables have been chosen, models have been obtained and Q and R selected, the computations are as follows:

6.1.1. Switching Algorithm 1

Step 3 requires the calculation of $\tilde{u}_{k+1}$ from u_k and e_k^e given by Equation (27). Here all that is needed is the computation of G_0^*, $G_0 G_0^*$ and its inverse as defined in Section 3. In step 4 the calculation (28) of the next control input signal u_{k+1} requires the minimisation of

$$\begin{aligned} \int_0^T &[(z(t) - z_k(t))^T Q(z(t) - z_k(t)) \\ &+(u(t) - \tilde{u}_{k+1}(t))^T R(u(t) - \tilde{u}_{k+1}(t))]\,\mathrm{d}t \end{aligned} \tag{76}$$

subject to the auxiliary equation dynamics $S(A_z, B_z, C_z, D_z)$ of Equation (22). In the case of $D_z = 0$, this is just the first iteration of the NOILC solution defined by Amann et al. (1996b) with reference $z_k(t)$ and initial input $\tilde{u}_{k+1}(t)$. That is the feedback plus feedforward solution,

$$u_{k+1}(t) = \tilde{u}_{k+1}(t) - R^{-1}B_z^T\left[K_z(t)x_z(t) + \xi_{z,k+1}(t)\right], \tag{77}$$

where the feedback gain $K_z(t)$ is the unique solution of the Riccatti equation,

$$\begin{aligned} &\dot{K}_z(t) + K_z(t)A_z + A_z^T K_z(t) - K_z(t)B_z R^{-1} B_z^T K_z(t) \\ &\quad + C_z^T Q C_z = 0, K_z(T) = 0 \end{aligned} \tag{78}$$

and the feedforward term is generated from

$$\begin{aligned} &\dot{\xi}_{z,k+1}(t) + A_z^T \xi_{z,k+1}(t) - K_z(t)B_z R^{-1} B_z^T \xi_{z,k+1}(t) \\ &\quad - K_z(t)B_z\tilde{u}_{k+1}(t) + C_z^T Q z_k(t) = 0, \\ &\quad \xi_{z,k+1}(T) = 0. \end{aligned} \tag{79}$$

The structure of this solution allows either off-line computation or on-line implementation.

6.1.2. Switching Algorithm 2

Step 4 of this algorithm has the same solution procedure for the auxiliary optimisation (49) as given above. The main difference lies within step 2, where k_0 inner iterations are used to compute u_{k+1} each of which is a problem of the form,

$$\begin{aligned} \min_{u\in L_2^r[0,T]} \Bigg(&\sum_{p=1}^{M}(r_p^e - y_p^e)^T Q_p (r_p^e - y_p^e) \\ &+ \int_0^T \left(u(t) - \tilde{u}_{k+1}^{(j)}(t)\right)^T R\left(u(t) - \tilde{u}_{k+1}^{(j)}(t)\right)\mathrm{d}t \Bigg) \end{aligned} \tag{80}$$

for $\tilde{u}_{k+1}^{(j+1)}$, subject to the constraint $y^e = G_1 u + d_1$ in the form of its state space model. There are two possible solution techniques for this

A feedforward solution: uses G_0^* and $G_0 G_0^*$ to write

$$\tilde{u}_{k+1}^{(j+1)} = \tilde{u}_{k+1}^{(j)} + G_0^*(I + G_0 G_0^*)^{-1} e_{(k+j)}^e, \quad j = 0, 1, 2, \ldots, k_0 - 1 \tag{81}$$

A feedback/feedforward solution: A solution for this has recently been published (Owens et al., 2012) in the form

$$\tilde{u}_{k+1}^{(j+1)}(t) = \tilde{u}_{k+1}^{(j)}(t) - R^{-1} B^T \left[K(t) x_z(t) + \xi_{k+1}^{(j+1)}(t) \right], \tag{82}$$

where the feedback gain $K(t)$ is now the unique solution of the Riccatti equation,

$$\dot{K}(t) + K(t)A + A^T K(t) - K(t) B R^{-1} B^T K(t) + C^T Q C = 0, \; K(T) = C^T F_M^T Q_M F_M C^T \tag{83}$$

with *jump conditions*

$$K(t_p-) - K(t_p+) = C^T F_p^T Q_M F_p C^T, \; 1 \le p < M \tag{84}$$

at the intermediate points and the feedforward term is generated from

$$\dot{\xi}_{k+1}^{(j+1)}(t) + A^T \xi_{k+1}^{(j+1)}(t) - K(t) B R^{-1} B^T \xi_{k+1}^{(j+1)}(t) - K(t) B \tilde{u}_{k+1}^{(j)}(t) = 0, \quad \text{with } \xi_{k+1}^{(j+1)}(T) = C^T F_M^T Q_M F_M r_M \tag{85}$$

and *jump conditions*

$$\xi_{k+1}^{(j+1)}(t_p-) - \xi_{k+1}^{(j+1)}(t_p+) = C^T F_p^T Q_p F_p r_p^e, \; 1 \le p < M. \tag{86}$$

The possibility of on-line implementation is clear from the structure of the solution.

6.2. Control of a robotic manipulator system

The ILC approaches developed have been experimentally 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. Each servomotor includes cascaded current and velocity control loops with a control input supplied by a dSPACE ds1103 real-time board with communication 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 Figure 1. Frequency response tests have established that the linear model (87) adequately represents the system dynamics, with angular input and output specified in degrees.

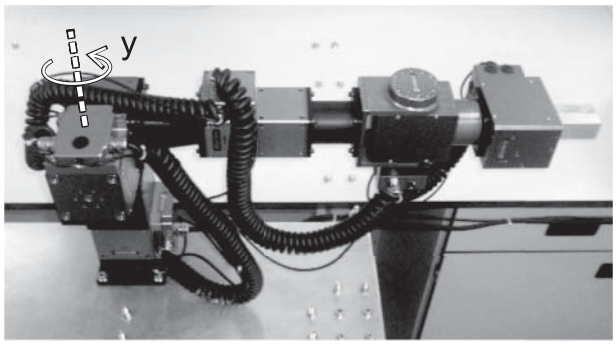

Figure 1. Robotic manipulator system showing actuated output, y.

$$G(s) = \frac{400788.1582(s + 12.14)(s + 24.01)^2}{(s + 31.52)(s + 22.97)(s + 2.178)(s^2 + 36.59s + 363.7)(s^2 + 124.5s + 4076)} \tag{87}$$

This can be represented in continuous-time by a state space model (11) whose matrices (A, B, C) are used for computation and simulation.

The intermediate point task replicates an industrial multiple 'pick and place' process in which payloads are manipulated during an assembly operation. The selected reference is given by the vector of output values

$$r^e = \begin{bmatrix} 20, & -30, & 10, & 20 \end{bmatrix}^T \tag{88}$$

at the $M = 4$ time points

$$t_1 = 1, \quad t_2 = 3, \quad t_3 = 5, \quad t_4 = T = 6. \tag{89}$$

on the interval [0, 6]. Using Equation (89) and the above representation of the plant G, G_0 is then given by (14) with $F_j = 1$, $1 \le j \le M$, whose elements (15) map the plant input to its output at time t_j.

To produce a smooth output response at all points $t \in [0, 6]$, and hence reduce vibration and potential payload damage/spillage between the specified intermediate points, some degree of minimisation of the mean square joint acceleration is proposed. The auxiliary variable is hence $z(t) = \ddot{y}(t)$, and, since system (87) has relative degree >2, G_1 can be written in continuous-time state-space form using Equation (22) with $S(A_z, B_z, C_z, D_z) = S(A, B, CA^2, 0)$.

The solutions u_∞ and z_∞ of the objective function (4) calculated using the nominal plant model are shown in Figure 2 for $R = I$, and the cases $Q = 0$ and $Q \to \infty$. The input signal for the latter contains significant initial oscillatory behaviour, which reduces as Q is decreased and is entirely absent when $Q = 0$. It is noted that, as $Q \to \infty$, $z_\infty(t)$ is

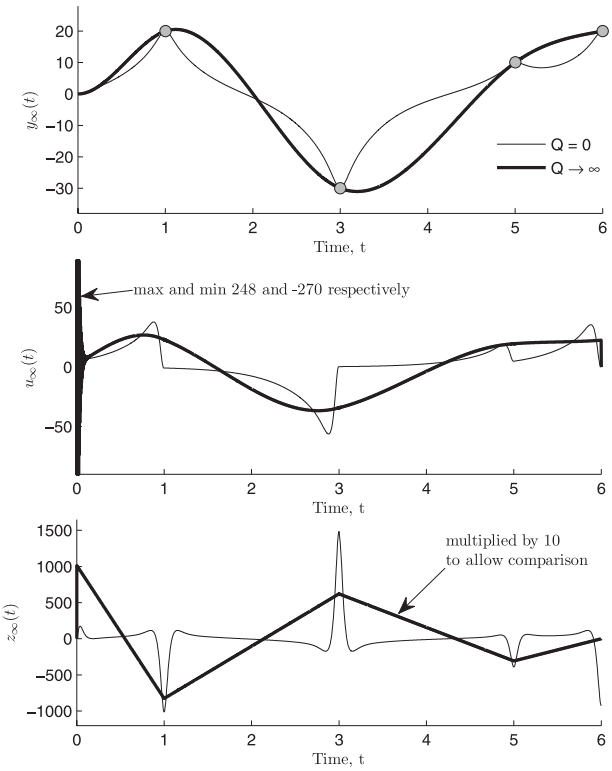

Figure 2. Limiting solutions to the objective function minimisation calculated using the nominal plant model and $R = I$.

almost linear between sample points, hence providing a less impulsive acceleration response. In the results which follow, all algorithms have been calculated in continuous-time and then discretised using a sampling time of 500 Hz.

6.3. *Switching Algorithm 1 results*

Switching Algorithm 1 has been implemented using the procedure described in Section 4.1. In step 3 the feedforward update (27), which involves the non-causal operator $G_0^*(G_0G_0^*)^{-1}$, is calculated using Equations (20) and (21). To solve Equation (28) in step 4 the algorithms in Section 6.1.1 have been employed. The best performance has been achieved by implementing feedback plus state feedback NOILC to solve the auxiliary problem experimentally in step 4 using (77)–(79). 200 trials have been performed and results are shown in Figure 3. Norm results comparing performance during the initial convergence stage are shown in Figure 4 and it can be seen that negligible degradation of performance occurs as the number of trials increases.

To quantify convergence rates, the iteration index, $k*$, is recorded that corresponds to when the intermediate point tracking criterion,

$$\|r^e - y_k^e\| < \epsilon \|r^e\|, \tag{90}$$

is first met. Here y_k^e is the output of the constraint trial in step 4. A value of $\epsilon = 0.01$ has been taken in all the results that follow. Experimental norm values recorded on step 4

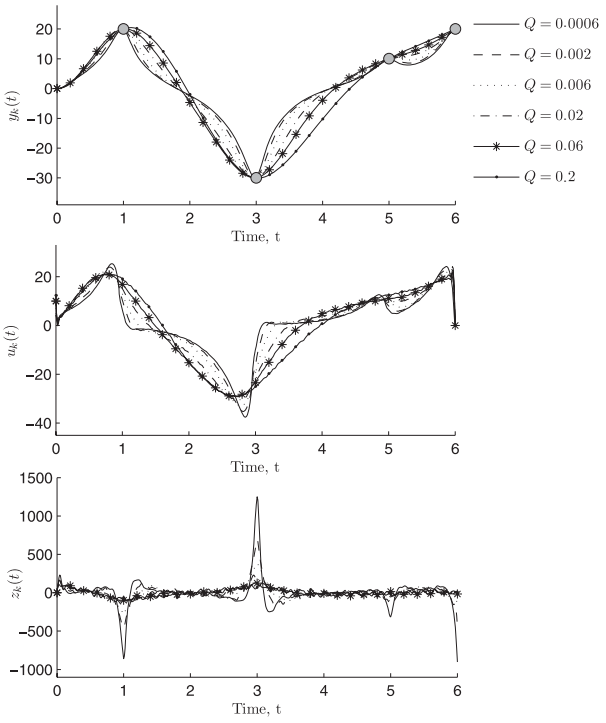

Figure 3. Switching Algorithm 1—experimental results recorded during step 4. Final trial input, output and auxiliary signals for a range of Q.

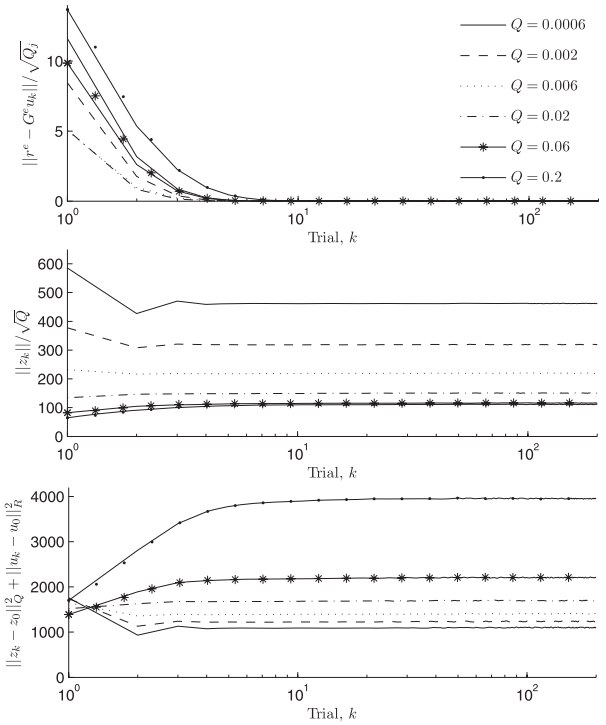

Figure 4. Switching Algorithm 1—experimental results recorded during step 4. Intermediate point tracking, auxiliary signal and cost function norms for a range of Q.

Table 1. Switching Algorithm 1—experimental results recorded during step 4. Convergence data using $\epsilon = 0.01$, and corresponding auxiliary and cost function norms. Predicted values for the limit $k \to \infty$ using the nominal plant model are shown on the right-hand side.

Q	k^*	$\|\|u_{k^*}\|\|$	$\|\|z_{k^*}\|\|/\sqrt{Q}$	$J(z_{k^*}, u_{k^*})$	$\|\|u_\infty\|\|$	$\|\|z_\infty\|\|/\sqrt{Q}$	$J(z_\infty, u_\infty)$
0.0006	4	30.774	457.994	1072.946	34.650	321.143	1262.510
0.002	3	32.256	304.160	1225.506	35.451	228.599	1361.341
0.006	3	33.462	207.854	1378.941	36.652	164.190	1505.176
0.02	3	35.369	141.851	1653.406	38.413	120.152	1764.333
0.06	4	37.215	112.847	2149.056	40.071	102.355	2234.326
0.2	6	38.681	108.103	3833.482	41.653	95.925	3575.325

of trial k^* are given in Table 1 for a variety of Q values and compared with the theoretical limit values obtained from the model. The excellent agreement is self-evident.

As Q increases, the theoretical optimal value of $J(u_\infty)$ increases. At the same time the theoretical optimal value of $||z_\infty||/\sqrt{Q}$ decreases as more weight is given to it in the optimisation problem (4). The experimental results match the predicted results reasonably well, although it has been found that for high values of Q ($\gg$1) there is a reduction in performance, with fluctuation in norms and eventual lack of convergence. This reflects Theorem 5.1 in Section 5, which bounds the admissible plant uncertainty as $\|\Delta G_0\| < (1 - r(W_\infty))(1 + \underline{\delta}^2)$. The calculated bounds are shown in Table 2.

Since robustness reduces as Q increases, excessive values should be avoided, especially as such Q values produce negligible improvement in cost, as reflected in the fact that the theoretical value of $||z_\infty||/\sqrt{Q}$ corresponding to $Q = 0.2$ (95.925) is close to the limiting value as $Q \to \infty$ (94.266). As Q is increased, the convergence of the plant outputs shown in Figure 3 to the limiting case as $Q \to \infty$ given in Figure 2 is clear. To mitigate the effect of noise and model uncertainty, it is common practice in ILC to apply a zero-phase low-pass filter to the control input u_{k+1} (and/or $\tilde{u}_{k+1}$) prior to its application in the subsequent trial (Longman, 2000), however in the present case this has not been done in order to assess the effects of noise and uncertainty.

6.4. *Switching Algorithm 2 results*

Switching Algorithm 2 is implemented using the procedure of Section 4.4 with the algorithms given in Section 6.1.2.

Table 2. Robustness bounds.

Q	$r(W_\infty)$	$(1 - r(W_\infty))(1 + \underline{\delta}^2)$
0.0006	0.0660	0.9340
0.002	0.1372	0.8628
0.006	0.2320	0.7681
0.02	0.3763	0.6239
0.06	0.5461	0.4545
0.2	0.7403	0.2608

Experimental implementation of both steps 2 and 4 has been found to provide the best performance. Within step 2, $k_0 = 2$ inner iteration trials with all $Q_j = 100$ have been used, and the feedforward plus state feedback implementation of NOILC, given by Equations (82)–(86), has been found to provide superior results than purely feedforward implementation. Step 4 is implemented as in Switching Algorithm 1, using Equations (77)–(79).

Two hundred trials of the switching algorithm procedure have been performed with norm plots shown in Figure 5, and summary convergence data is given in Table 3.

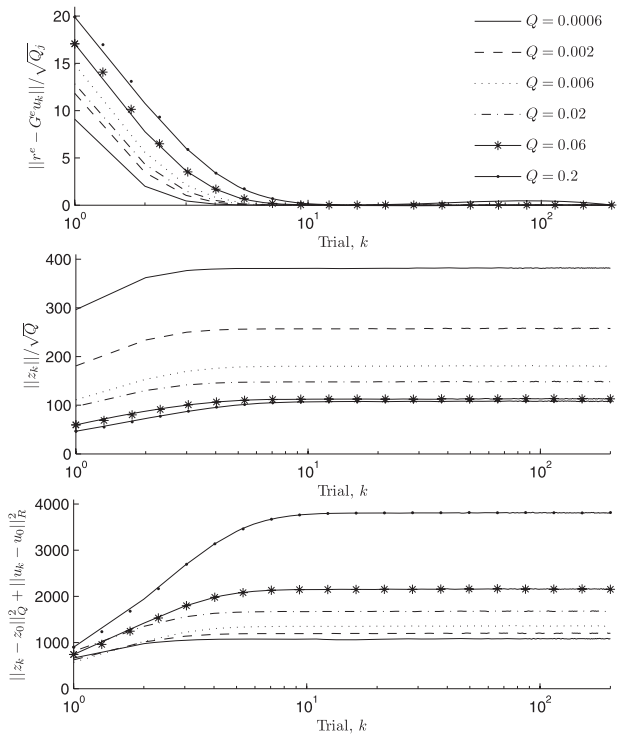

Figure 5. Switching Algorithm 2—experimental results recorded during step 4. $Q_j = 100$ and $k_0 = 2$ inner iterations are used in step 2, using combined feedback and feedforward NOILC. Intermediate point tracking, auxiliary signal and cost function norms for a range of Q.

Table 3. Switching Algorithm 2—experimental results recorded during step 4. $Q_j = 100$ and $k_0 = 2$ inner iterations are used in step 2, using combined feedback and feedforward NOILC. Convergence data using $\epsilon = 0.01$, and corresponding auxiliary and cost function norms. Predicted values for the limit $k \to \infty$ using the nominal plant model are shown on the right-hand side.

Q	k^*	$\|\|u_{k^*}\|\|$	$\|\|z_{k^*}\|\|/\sqrt{Q}$	$J(z_{k^*}, u_{k^*})$	$\|u_\infty\|$	$\|\|z_\infty\|\|/\sqrt{Q}$	$J(z_\infty, u_\infty)$
0.0006	4	31.323	379.843	1067.727	34.650	321.143	1262.510
0.002	4	32.334	254.567	1175.111	35.451	228.599	1361.341
0.006	5	33.753	178.199	1329.856	36.652	164.190	1505.176
0.02	5	35.598	127.863	1594.241	38.413	120.152	1764.333
0.06	6	37.044	110.756	2109.921	40.071	102.355	2234.326
0.2	8	38.532	105.772	3722.299	41.653	95.925	3575.325

Table 4. Switching Algorithm 2—experimental results recorded during step 4. $Q_j = 50$ and $k_0 = 2$ inner iterations are used in step 2, using combined feedback and feedforward NOILC. Convergence data using $\epsilon = 0.01$, and corresponding auxiliary and cost function norms. Predicted values for the limit $k \to \infty$ using the nominal plant model are shown on the right-hand side.

Q	k^*	$\|\|u_{k^*}\|\|$	$\|\|z_{k^*}\|\|/\sqrt{Q}$	$J(z_{k^*}, u_{k^*})$	$\|u_\infty\|$	$\|\|z_\infty\|\|/\sqrt{Q}$	$J(z_\infty, u_\infty)$
0.0006	5	31.221	378.558	1060.760	34.650	321.143	1262.510
0.002	6	32.398	255.212	1179.898	35.451	228.599	1361.341
0.006	7	33.794	178.471	1333.160	36.652	164.190	1505.176
0.02	8	35.580	127.784	1592.562	38.413	120.152	1764.333
0.06	10	37.115	111.240	2120.042	40.071	102.355	2234.326
0.2	11	38.194	105.977	3705.055	41.653	95.925	3575.325

Table 5. Switching Algorithm 2—experimental results recorded during step 4. $Q_j = 50$ and $k_0 = 10$ inner iterations are used in step 2, using combined feedback and feedforward NOILC. Convergence data using $\epsilon = 0.01$, and corresponding auxiliary and cost function norms. Predicted values for the limit $k \to \infty$ using the nominal plant model are shown on the right-hand side.

Q	k^*	$\|\|u_{k^*}\|\|$	$\|\|z_{k^*}\|\|/\sqrt{Q}$	$J(z_{k^*}, u_{k^*})$	$\|u_\infty\|$	$\|\|z_\infty\|\|/\sqrt{Q}$	$J(z_\infty, u_\infty)$
0.0006	3	31.152	377.251	1055.838	34.650	321.143	1262.510
0.002	3	32.140	254.629	1162.651	35.451	228.599	1361.341
0.006	4	33.576	177.346	1316.057	36.652	164.190	1505.176
0.02	5	35.394	126.844	1574.523	38.413	120.152	1764.333
0.06	6	36.948	111.334	2108.870	40.071	102.355	2234.326
0.2	7	38.071	105.142	3660.369	41.653	95.925	3575.325

Switching Algorithm 2 attains lowest norm values when Q_j is chosen small enough to produce accurate convergence (i.e. Q_j equals a value that performs well using standard intermediate point NOILC (Owens et al., 2012) without any switching). If Q_j is chosen accordingly, the value of k^* then determines the convergence rate of the combined switching algorithm, as predicted by Theorem 4.4. To illustrate this, Table 4 shows results using $Q_j = 50$ with $k_0 = 2$ inner iterations, and displays similar norm values to the $Q_j = 100$ case, but slower convergence. If k_0 is then increased, the convergence rate using $Q_j = 50$ can then be returned back to similar levels seen in the $Q_j = 100$ results. This is shown in Table 5 with results for $Q_j = 50$ with $k_0 = 10$ inner iterations. The relatively large number of inner iterations leads to accurate intermediate point tracking during step 2, and hence to rapid convergence of the overall switching algorithm to low norm levels. The price paid for this is a greater overall number of experimental trials.

7. Conclusions

A powerful new and general functional analytic framework has been developed which models a novel class of ILC problems which require tracking of a specified finite dimensional reference signal whilst minimising a quadratic objective function of input and a defined auxiliary variable. Although the tracking problem is finite dimensional, the underlying dynamics and auxiliary dynamics can be infinite dimensional. It is shown that this formulation can be used to solve the abstract tracking problem whilst simultaneously minimising a general class of quadratic objective function representing secondary control objectives. General applications to solutions of intermediate point discrete and continuous time tracking are described in detail and indications of applications to multi-rate and discrete delay systems are outlined. The ideas and algorithms have a broad range of operations within fields such as industrial automation and robotics require such a framework to, for

example, move payloads to preset positions during manufacturing tasks whilst reducing spillage, vibration, contact forces and actuator wear.

This paper is the first to derive general solutions employing the norm optimal ILC framework, with implementation achieved using either a feedforward update structure, or a combined feedback plus feedforward form which has the practically confirmed benefit of additional robustness to modelling uncertainty. The general analysis and algorithms proposed, expressed in terms of operator notation, has the advantage of efficiency and generality of proofs plus the great benefit that the results on performance and robustness are directly applicable to many system types including linear discrete-time, continuous-time and time-varying systems. The algorithm structure is based on systematic switching between objectives and permits design flexibility in terms of choice of a number of parameters $[Q_j]$, Q, R, k_0 and the choice of whether off-line updates are applied to the experimental plant or updates are generated on-line. A rigorous robustness analysis has been performed which provides explicit uncertainty bounds, which ensure convergence to a solution which precisely achieves the tracking objective despite the presence of plant-model mismatch. The various options have been explored using experimental tests on a robotic manipulator system, with results confirming a high degree of performance in practice.

References

Ahn, H. S., Chen, Y., & Moore, K. L. (2007). Iterative learning control: Brief survey and categorization 1998–2004. *IEEE Transactions on Systems Man and Cybernetics Part C*, 37, 1099–1121.

Amann, N., Owens, D. H., & Rogers, E. (1996a). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proceedings - Control Theory Applications*, 143, 217–224.

Amann, N., Owens, D. H., & Rogers, E. (1996b). Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 65, 277–293.

Amann, N., Owens, D. H., & Rogers, E. (1998). Predictive optimal iterative learning control. *International Journal of Control*, 69, 203–226.

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

Barton, K. L., & Alleyne, A. G. (2011). A norm optimal approach to time-varying ILC with application to a multi-axis robotic testbed. *IEEE Transactions on Control Systems Technology*, 19, 166–180.

Bristow, D. A., Tharayil, M., & Alleyne, A. G. (2006). A survey of iterative learning control: A learning-based method for high-performance tracking control. *IEEE Control Systems Magazine*, 26, 96–114.

Chen, Y., & Xu, J. X. (1997). 'A high-order terminal iterative learning control scheme,' in *Proceedings of the 36th IEEE conference on decision and control* (pp. 3771–3772), San Diego, California.

Chu, B., & Owens, D. H. (2010). Iterative learning control for constrained linear systems. *International Journal of Control*, 83, 1397–1413.

Freeman, C. T. (2012). Constrained point-to-point iterative learning control with experimental verification. *Control Engineering Practice*, 20, 489–498.

Freeman, C. T., Rogers, E., Hughes, A. M., Burridge, J. H., & Meadmore, K. L. (2012). Iterative learning control in healthcare: Electrical stimulation and robotic-assisted upper limb stroke rehabilitation. *IEEE Control Systems Magazine*, 32, 18–43.

Freeman, C. T., & Tan, Y. (2012). Iterative learning control with mixed constraints for point-to-point tracking. *IEEE Transactions on Control System Technology*, 1–13. doi:10.1109/TCST.2012.2187787

Furuta, K., & Yamakita, M. (1987). 'The design of a learning control system for multivariable systems,' in *Proceedings of the IEEE international symposium on intelligent control* (pp. 371–376), Philadelphia, Pennsylvania.

Gauthier, G., & Boulet, B. (2008). Robust design of terminal ILC with H_∞ mixed sensitivity approach for a thermoforming oven. *Journal of Manufacturing Science and Engineering*, 289391. doi:10.1155/2008/289391

Gauthier, G., & Boulet, B. (2009). 'Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens,' in *Proceedings of the American control conference* (pp. 1640–1645), St. Louis, Missouri.

Gunnarsson, S., & Norrlof, M. (2001). On the design of ILC algorithms using optimization. *Automatica*, 37, 2011–2016.

Kinosita, K., Sogo, T., & Adachi, N. (2002). Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control*, 4, 60–67.

Lee, J., Lee, K., & Kim, W. (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36, 641–657.

Longman, R. W. (2000). Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73, 930–954.

Luenberger, D. G. (1969). *Optimization by vector space methods*. Hoboken, New Jersey: Wiley.

Owens, D. H., & Chu, B. (2009). Accelerated norm-optimal iterative learning control algorithms using successive projection. *International Journal of Control*, 82, 1469–1484.

Owens, D. H., & Chu, B. (2010). Modelling of non-minimum phase effects in discrete-time norm optimal iterative learning control. *International Journal of Control*, 83, 2012–2027.

Owens, D. H., Chu, B., Rogers, E., Freeman, C. T., & Lewin, P. L. (2013). The influence of non-minimum-phase zeros on the performance of optimal continuous-time iterative learning control. *IEEE Transactions on Control System Technology*, In press.

Owens, D. H., Freeman, C. T., & Dinh, T. V. (2012). Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation. *IEEE Transactions on Control System Technology*, 1. doi: 10.1109/TCST.2012.2196281

Owens, D. H., Hätönen, J. J., & Daley, S. (2009). Robust monotone gradient-based discrete-time iterative learning control. *International Journal of Robust and Nonlinear Control*, 19, 634–661.

Rogers, E., Owens, D. H., Werner, H., Freeman, C. T., Lewin, P. L., Kichhoff, S., ...Lichtenberg, G. (2010). Norm optimal iterative learning control with application to problems in accelerator based free electron lasers and rehabilitation robotics. *European Journal of Control*, 16, 497–524.

Son, T. D., & Ahn, H. S. (2011). 'Terminal iterative learning control with multiple intermediate pass points,' in *Proceedings of the American control conference* (pp. 3651–3656), San Francisco, California.
Son, T. D., & Ahn, H. S. (2012). Optimal iterative learning control with uncertain reference points. In *Proceedings of the IEEE multi-conference on systems and control* (pp. 1244–1248), Dubrovnik, Croatia.
Son, T. D., Ahn, H. S., & Nguyen, D. H. (2011a). 'An interpolation method for multiple terminal iterative learning control,' in *Proceedings of the IEEE multi-conference on systems and control* (pp. 1528–1532), Denver, Colorado.
Son, T. D., Nguyen, D. H., & Ahn, H. S. (2011b, December). Iterative learning control for optimal multiple-point tracking. In *Proceedings of the 50th IEEE conference on decision and control and European control conference* (pp. 6025–6030), Orlando, Florida.
Wang, Y., & Hou, Z. (2011). Terminal iterative learning control based station stop control of a train. *International Journal of Control*, 84, 1263–1274.
Xu, J. X., Chen, Y., Lee, T., & Yamamoto, S. (1999). Terminal iterative learning control with an application to RTPCVD thickness control. *Automatica*, 35, 1535–1542.
Xu, J. X., & Huang, D. (2008). Initial state iterative learning for final state control in motion systems. *Automatica*, 44, 3162–3169.

Appendix 1. A summary of relevant Hilbert space concepts

The following aspects of Hilbert space theory are relevant in the theoretical development used in the paper:

(1) The adjoint operator of a bounded linear operator K mapping a Hilbert space H_1 into a Hilbert space H_2 is the uniquely defined bounded operator $K^*: H_2 \to H_1$ satisfying the equality

$$<u, Kv>_{H_2} = <K^*u, v>_{H_1} \ \forall v \in H_1 \ \forall u \in H_2. \quad \text{(A1)}$$

The induced operator norm of K is defined to be

$$\|K\| = sup\{\|Ku\|_{H_2} : u \in H_1 \ \& \ \|u\|_{H_1} = 1\}. \quad \text{(A2)}$$

(2) If $H_1 = H_2$ then $\|K^*\| = \|K\|$. K is said to be self-adjoint if $H_1 = H_2$ and $K = K^*$. If this is the case then K is positive semi-definite (respectively positive definite) if and only if $<v, Kv> \geq$ (resp. $>$) $0 \ \forall$ non-zero$v \in H_1$. In the case of $H_1 = H_2$ being finite dimensional with K positive semi-definite or positive definite, the spectral radius $r(K)$ of K has the same value as the norm of K and is then given as the solution of the optimisation problem

$$r(K) = \|K\|_{H_1} = max\{<v, Kv>_{H_1} : v \in H_1, \|v\|_{H_1} = 1\}. \quad \text{(A3)}$$

(3) The condition on the range of G_0 is necessary to ensure that all signals in $\mathcal{R}_e$ can be tracked. In practice, if $\mathcal{R}[G_0]$ is a proper subspace of $\mathcal{R}_e$, the apparent problem of failing to satisfy the assumptions can be removed in a formal way by regarding $\mathcal{R}[G_0]$ as a Hilbert space in its own right and setting $\mathcal{R}_e$ equal to this subspace.

Appendix 2. Discrete time systems modelling

In this appendix, the models and operators relevant to the case of discrete time systems are summarised. Both uni-rate and multi-rate systems are considered.

A2.1. ***Identical input and output sample rates***

The relevant formulae for an ℓ-input, m-output discrete time system $y = Gu + d$ represented by a discrete time state space model $S(A, B, C)$ of the form,

$$x(t+1) = Ax(t) + Bu(t), x(0) = x_0, y(t) = Cx(t), \quad \text{(on the interval } t = 0, 1, 2, \ldots, N) \quad \text{(A4)}$$

can be derived in a similar way, with details outlined below. The relevant real Hilbert spaces $\mathcal{Y}, \mathcal{U}, \mathcal{Z}$ are now all spaces of vector time series on the defined time interval (0, 1, 2, …, N) with inner products (assuming constant Q and R for simplicity),

$$\begin{aligned} U: \quad <u, v>_R &= \sum_{j=0}^{N} u^T(t) R v(t), \quad \text{so that } \|u\|_R \\ &= \sqrt{<u, u>_R} \end{aligned} \quad \text{(A5)}$$

$$\begin{aligned} \mathcal{Z}: \quad <z_1, z_2>_Q &= \sum_{j=0}^{N} z_1^T(t) Q z_2(t), \quad \text{so that } \|z\|_Q \\ &= \sqrt{<z, z>_Q}. \end{aligned}$$

The space $\mathcal{R}_e$ and its inner product remain as for the continuous time case.

The underlying dynamics of Equation (1) are defined by the discrete convolution,

$$\begin{aligned} (Gu)(0) = 0, (Gu)(t) &= \sum_{s=0}^{t-1} CA^{t-s-1} Bu(s) \ \forall t \geq 1, \ \& \ d(t) \\ &= CA^t x_0 \forall t \geq 0. \end{aligned} \quad \text{(A6)}$$

The notation defining the tracking problem (2) is the same as for the continuous time case but each G_j^e in Equation (14) is now given by

$$G_j^e u = \sum_{s=0}^{t_j - 1} F_j CA^{t_j - s - 1} Bu(s) \quad \text{(A7)}$$

so that its adjoint is the map whose operation on $w_j \in \mathcal{R}^{f_j}$ is given by the time series

$$(G_j^{e*} w_j)(t) = \left\{ \begin{array}{ll} R^{-1} B^T (A^T)^{t_j - t - 1} C^T F_j^T Q_j w_j; & 0 \leq t \leq t_j - 1; \\ 0; & t_j \leq t \leq N \end{array} \right\}. \quad \text{(A8)}$$

In particular $G_0 G_0^*$ has (i, j)th block

$$\begin{aligned} (G_0 G_0^*)_{ij} &= G_i^e G_j^{e*} \\ &= \sum_{s=0}^{\min(t_i, t_j) - 1} F_i CA^{t_i - s - 1} BR^{-1} B^T (A^T)^{t_j - s - 1} \\ &\quad \times C^T F_j^T Q_j. \end{aligned} \quad \text{(A9)}$$

The auxiliary optimisation problem (3) for the switching algorithms is constructed from an auxiliary discrete system $S(A_z, B_z, C_z, D_z)$ defined as

$$\begin{aligned} &x_z(t+1) = A_z x_z(t) + B_z u(t), x_z(0) = x_{z0}, z(t) \\ &= C_z x_z(t) + D_z u(t), \\ &\quad \text{(on the interval } t = 0, 1, 2, \ldots, N) \end{aligned} \tag{A10}$$

so that G_1 and d_1 are represented as

$$\begin{aligned} (G_1 u)(0) = 0, (G_1 u)(t) = \sum_{s=0}^{t-1} C_z A_z^{t-s-1} B_z u(s) \forall t \geq 1, \& \\ d_1(t) = C_z A_z^t x_{z0} \forall t \geq 0 \end{aligned} \tag{A11}$$

'and the cost (4) is given by

$$\begin{aligned} J(z, u) = \sum_{t=0}^{N} \Big[&(z(t) - z_0(t))^T Q(z(t) - z_0(t)) \\ &+ (u(t) - u_0(t))^T R(u(t) - u_0(t)) \Big]. \end{aligned} \tag{A12}$$

In summary, for the discrete time case the tracking requirement (2) is represented by Equations (14) and (A7), the auxiliary dynamics (3) is represented by Equation (A11), and the objective function (4) is represented by Equation (A12).

A2.2. ***A note on multi-rate sampled systems***

Consider the case of multi-rate sampled systems where the output sampling interval is an integer multiple of the input sampling interval and it is desired that the sampled output tracks a desired reference at the output sample points. The auxiliary variable z is assumed to be defined and of interest at sample points synchronous with the input sample times. This situation can be modelled using the discrete modelling tools above by regarding the output sample points as the intermediate points $\{t_j\}_{1 \leq j \leq M}$. In this case M and hence n_r will be typically very large. This does not alter the principles described in the algorithms and could be used to control the inputs and auxiliary variable between output sample instances. If auxiliary variable measurements at the faster input sampling speed are not available, then the auxiliary optimisation problem will necessarily be model-based and entirely off-line. Alternatively an on-line implementation could be considered using a model and observer to reconstruct z from input output data.