Norm Optimal Iterative Learning Control with Auxiliary Optimization
– An Inverse Model Approach

David H Owens1,2,3, Chris T Freeman2 and Bing Chu2

Abstract—An Iterative Learning Control (ILC) algorithm is derived to address the problem in which tracking is only required at selected intermediate points within the time interval while an auxiliary function is simultaneously minimized. This is done by the means of robotic automation tasks where point-to-point motion control is combined with a need to reduce payload spillage, vibration tendencies and actuator wear. Experimental results confirm practical utility and theoretical performance.

I. INTRODUCTION

Iterative Learning Control (ILC) enables systems which repeatedly perform a tracking task to learn from past experience in order to increase performance. Recently the requirement that all points over the trial duration, \( T < \infty \), are tracked has been relaxed, with schemes allowing tracking of any finite subset of points along \( T \), in both discrete [1], [2], [3] and continuous-time [4]. These provide a framework applicable to a broad class of practical problems in which the system output (e.g. payload position) is only critical at an isolated number of prescribed time instants (e.g. production line automation, crane control, satellite positioning, stroke rehabilitation and robotic ‘pick and place’ tasks).

Tracking a subset of points corresponds to a non-unique control signal, and it is possible to introduce an auxiliary objective function involving a (set of) auxiliary variable(s). Practical examples include output jerk, acceleration, intersample velocity and control effort. Within robotic manipulation, such an approach would limit system vibration as well as spillage or damage of/to the payload. While it is possible to tackle this in the standard ILC framework through careful reference selection, the formulation introduced in this paper embeds faster convergence, superior transient performance, greater robustness and broader design and analysis transparency. Note that the need for tracking to be completed with zero error means tracking and additional constraints cannot be combined within the same objective function.

This paper develops the first solution to intermediate point tracking with auxiliary optimization using algorithms which enforce convergence at a pre-specified rate. Bounds are also derived on the permissible modeling uncertainty.

II. PROBLEM STATEMENT

The ILC problem is now formulated for a general class of linear plants, including time-invariant, time-varying continuous time, discrete time and multi-rate systems.

A. System Models, Signal Spaces & Problem Statement

Let \( \mathcal{U}, \mathcal{Y}, \mathcal{Z} \) be real Hilbert spaces, with the inner product of \( \mathcal{U} \) denoted \( \langle u, v \rangle_\mathcal{U} \) with induced norm \( \| u \|_\mathcal{U} = \sqrt{\langle u, u \rangle_\mathcal{U}} \) with similar notation used for \( \mathcal{Y} \) and \( \mathcal{Z} \). Signals \( u \in \mathcal{U} \) and \( y \in \mathcal{Y} \) represent inputs and outputs respectively, with \( z \in \mathcal{Z} \) an auxiliary variable important in the problem.

In ILC algorithm design is usually based on a model of underlying plant dynamics described by

\[
y = Gu + d \quad \text{(Underlying Plant Dynamics)} \tag{1}
\]

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 repeated disturbances or bias. \( G \) is used to construct the tracking problem, consisting of a specific tracking requirement and auxiliary variable optimization.

The tracking requirement is defined using a signal \( r^e \in \mathcal{R}_e \), also a Hilbert space. The tracking sought is specified by

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

where \( G_0 \) is a bounded linear operator mapping \( \mathcal{U} \) into \( \mathcal{R}_e \) and is constructed to form an extended output \( y^e \) representing the desired tracking variables and their dependence on the input. The iteration independent term \( d_0 \in \mathcal{R}_e \) describes the effects of initial conditions and repeated disturbances or bias.

The auxiliary signal \( z \) is chosen as a suitable measure of desired control performance and is specified by the relation

\[
z = G_1 u + d_1 \quad \text{(Auxiliary Variable Dynamics)} \tag{3}
\]

where \( G_1 \) is a bounded linear operator mapping \( \mathcal{U} \) into \( \mathcal{Z} \) and \( d_1 \in \mathcal{Z} \) is an iteration independent term again describing initial condition and repeated disturbance effects.

Problem Definition: Given desired reference \( r^e \in \mathcal{R}_e \) and initial input \( u_0 \in \mathcal{U} \), find an ILC law which generates an input which solves the tracking requirement \( y^e = r^e \), while also minimizing the objective function

\[
J(z, u) = \| z - z_0 \|^2_2 + \| u - u_0 \|_\mathcal{U}^2 \tag{4}
\]

where \( (z_0, u_0) \) is the solution pair of the auxiliary equation for the input \( u_0 \). That is, the ILC law converges to a solution of the constrained optimization problem \( (z_\infty, u_\infty) = \text{argmin}\{ J(z, u) : u \in \mathcal{U}, r^e = G_0 u + d_0, \ z = G_1 u + d_1 \} \)

Dimensionality and Invertibility Assumptions: It is assumed throughout the paper that

1) the Hilbert space \( \mathcal{R}_e \) is finite dimensional (i.e. that the number of tracking requirements is finite) and also that
2) the kernel of the adjoint operator satisfies \( \ker^* [G_0^*] = \{0\} \).
B. Linear Continuous State Space Systems

To illustrate the problem, the operators are now specified for continuous linear time-invariant (LTI) systems described in state space form, with intermediate point tracking of desired output values at specific points in \([0, T]\).

1) Intermediate Point Tracking System Model: Let \(S(A, B, C)\) be a strictly proper \(m\)-output, \(\ell\)-input, \(n\)-state, LTI system

\[
\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad x(t) \in \mathbb{R}^n
\]

\[
y(t) = Cx(t), \quad t \in [0, T], \quad T < \infty
\]

then the relevant operators in (1) are

\[
(Gu)(t) = \int_0^t Ce^{A(t-s)}Bu(s)ds, \quad d(t) = Ce^{A}x_0, \quad t \in [0, T]
\]

with \(\mathcal{U} = L_2^m[0, T]\) and \(\mathcal{Y} = L_2^n[0, T]\). Note: The output \(y(t)\) can be augmented to suit the application, e.g., if \(CB = 0\), \(\dot{y}\) can also be specified in the tracking task by the map

\[
y \rightarrow \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \quad C \rightarrow \begin{bmatrix} C \\ CA \end{bmatrix} \quad \text{and} \quad m \rightarrow 2m
\]

(7)

Let \(0 < t_1 < t_2 < \cdots < t_M = T\) be \(M\) distinct intermediate points in \([0, T]\). For any \(f \in L_2^m[0, T]\), consider the linear map \(f \rightarrow f^e\) 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, and allows only selected elements of \(f\) to be specified at each time \(t = t_j\). Clearly \(\mathcal{R}_e\) can be identified with \(R^{n_r}\) with \(n_r = f_1 + \cdots + f_M\). With this notation, the "extended output" \(y^e\) comprises the values \(F_j\dot{y}(t_j), 1 \leq j \leq M\) at intermediate points i.e.

\[
y^e = G_0u + d_0,
\]

\[
G_0u = \begin{bmatrix} G_1^e u \\ \vdots \\ G_M^e u \end{bmatrix}, \quad d_0 = \begin{bmatrix} F_1 d(t_1) \\ \vdots \\ F_M d(t_M) \end{bmatrix}
\]

(8)

with \(G_0 : L_2^m[0, T] \rightarrow R^{f_1} \times \cdots \times R^{f_M}\) a linear operator with components constructed from \(G\)

\[
G_j^e u = F_j(Gu)(t_j) = F_j \int_0^{T_j} Ce^{A(t-j)}Bu(t)dt
\]

(9)

The reference signal is defined as \(r^e = ([r_1^e]^T, \cdots, [r_M^e]^T)^T \in \mathcal{R}_e\) where \(r_j^e\) is the desired value of \(F_jy(t_j)\) at time \(t_j, 1 \leq j \leq M\). The tracking error is \(e^e = r^e - y^e = (r - y)^e\).

2) Adjoint Operators and \(G_0G_0^e\): The product space \(\mathcal{R}_e = R^{f_1} \times \cdots \times R^{f_M}\) is a Hilbert space with inner product

\[
<(x_1, \cdots, x_M), (w_1, \cdots, w_M)>_{\mathcal{Q}} = \sum_{j=1}^M v_j^T Q_j w_j
\]

where the \(f_j \times f_j\) matrices \(Q_j, 1 \leq j \leq M\) are symmetric and positive definite. \([Q]\) denotes the data set \(\{Q_1, \cdots, Q_M\}\) and the squared error norm is \(|e^e|^2_{[Q]} = <e^e, e^e>_{[Q]}\).

The space \(\mathcal{U}\) is a real Hilbert space with inner product and norm

\[
<u, v>_R = \int_0^T u^T(t)Rv(t)dt \quad \text{where} \quad R = R^T > 0 \quad \text{and}
\]

\[
\|u\|_R = \sqrt{\int_0^T u^T(t)Ru(t)dt}
\]

Noting that, by definition,

\[
<(w_1, \cdots, w_M), G_0u>_{[Q]} = <G_0^*(w_1, \cdots, w_M), u>_R
\]

the adjoint is computed from adjoints of \(G_j^e, 1 \leq j \leq M\).

Adjoint Operator of \(G_j^e\): Consider \(G_j^e\) via the identity

\[
w_j^T Q_j F_j \int_0^{T_j} Ce^{A(t-j)}Bu(t)dt = \int_0^{T_j} (R^{-1}B^T e^{A(t-j)}C^TF_j Q_j w_j)T R u(t)dt.
\]

It can hence be deduced that \((G_j^{e*} w_j)(t) = \)

\[
\begin{cases}
R^{-1}B^T e^{A(t-j)}C^TF_j Q_j w_j & ; 0 \leq t \leq t_j \\
0 & ; t > t_j
\end{cases}
\]

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

\[
\dot{P}_j(t) = -A^T P_j(t), \quad P_j(t_j) = C^TF_j Q_j,
\]

\((G_j^{e*} w_j)(t) = R^{-1}B^TP_j(t)w_j
\]

Adjoint Operator of \(G_0\): The adjoint operator of \(G_0\) is then the map \((w_1, \cdots, w_M) \mapsto u)

\[
u(t) = \sum_{j=1}^M (G_j^{e*} w_j)(t) = R^{-1}B^T \sum_{j=1}^M P_j(t) w_j
\]

(10)

Finally, the matrix \(G_0G_0^e\) is obtained as an \(n_r \times n_r\) block matrix with \(f_1 \times f_j (i, j)^{th}\) block given by \((G_0G_0^e)_{ij} = \)

\[
\int_0^{\min(t, t_j)} F_i Ce^{A(t-i)}BR^{-1}B^T e^{A(t-i)}C^TF_j Q_j dt = Q_j^{-1} \int_0^{\min(t, t_j)} F_i(t) R P_j(t) dt
\]

(12)

3) The Auxiliary System: The auxiliary system (3) is assumed to be described by the \(\ell\)-input, \(m_z\)-output, \(n_z\)-state proper model \(S(A_z, B_z, C_z, D_z)\)

\[
\dot{x}_z(t) = A_z x_z(t) + B_z u(t), \quad x_z(0) = x_0, \quad x_z(t) \in \mathbb{R}^{n_z}
\]

\[
z(t) = C_z x_z(t) + D_z u(t), \quad t \in [0, T]
\]

with \(G_1\) and \(d_1\) defined as \(d_1(t) = C_z e^{A_z t} x_0,\)

\[
(G_1 u)(t) = D_z u(t) + \int_0^t C_z e^{A_z (t-s)} B_z u(s)ds, \quad t \in [0, T]
\]

So that \(Z = L_2^{m_z}[0, T]\) with inner product and norm

\[
<z_1, z_2> = \int_0^T z_1^T(t)Q z_2(t)dt \quad \text{and}
\]

\[
\|z\|_Q = \sqrt{\int_0^T z^T(t)Qz(t)dt}
\]

where \(Q = Q^T > 0\).
The objective function (4) is

\[ J(z, u) = \int_{0}^{T} \left[ (z(t) - z_0(t))^T Q(z(t) - z_0(t)) + (u(t) - u_0(t))^T R(u(t) - u_0(t)) \right] dt \]

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 \( R \).

### III. INVERSE MODEL ALGORITHM

Recall the tracking error \( e^e = r^e - y^e \) and let \( f_k \) denote the signal \( f \) on iteration \( k \). The proposed inverse model algorithm is based on the formal solution of (5) stated next.

**Theorem 1:** The solution of the Iterative Learning Control Problem with Auxiliary Optimization is given by the formula

\[ u_\infty = u_0 + L e^e_0 \]

where the operator \( L : \mathcal{R}_e \rightarrow \mathcal{U} \) is defined by

\[ L = L_0 (G_0 L_0)^{-1} \quad \text{with} \quad L_0 = (I + G_1^* G_1)^{-1} G_0^* : \mathcal{R}_e \rightarrow \mathcal{U} \]

and is a right inverse as \( G_0 L = I \) (the identity on \( \mathcal{R}_e \)).

**Proof:** See [5]. \( \square \)

This theorem leads immediately to the conceptual feedforward "inverse model" algorithm defined by

\[ u_{k+1} = u_k + L e^e_k \]  \hspace{1cm} (13)

which, starting from the initial control signal \( u_0 \), converges in one step to the required control \( u_\infty \) (see [5] for proof).

Model uncertainty makes robustness an important property of any ILC algorithm. To provide a mechanism to limit step size, slowing down convergence but improving robustness, the conceptual result can be converted into the feedforward variable parameter "inverse model" algorithm defined by

\[ u_{k+1} = u_k + \beta_{k+1} L e^e_k \]  \hspace{1cm} (14)

starting from the initial control signal \( u_0 \) with an iteration dependent parameter (learning gain) \( \beta_{k+1} \). The term \( \beta_{k+1} \) can be computed to minimize the objective function

\[ J_\beta(\beta_{k+1}) = \| e^e_{k+1} \|_{\mathcal{R}_e}^2 + w_{k+1} \beta_{k+1}^2 \]  \hspace{1cm} (15)

where \( w_{k+1} \) is a strictly positive weight. A suitable form is

\[ w_{k+1} = \tilde{w}_1 + \tilde{w}_2 \| e^e_k \|_{\mathcal{R}_e}^2 \quad \tilde{w}_1 \geq 0 \quad \tilde{w}_2 \geq 0 \quad \tilde{w}_1 + \tilde{w}_2 > 0 \]  \hspace{1cm} (16)

Substituting into \( J_\beta \) then gives the following monotonic convergence theorem:

**Theorem 2:** Suppose that, in the variable gain "inverse-model" algorithm defined by the iteration

\[ u_{k+1} = u_k + \beta_{k+1} L e^e_k \]  \hspace{1cm} (17)

the learning gain \( \beta_{k+1} \), is computed to minimize \( J_\beta(\beta_{k+1}) \). Then the required gain value is

\[ \beta_{k+1} = \beta(e^e_k) \quad \text{where} \quad \beta(e^e) = \frac{\| e^e \|_{\mathcal{R}_e}^2}{w_{k+1} + \| e^e \|_{\mathcal{R}_e}^2} \quad \forall k \geq 0 \]  \hspace{1cm} (18)

and monotonic error norm reduction is achieved

\[ \| e^e_{k+1} \|_{\mathcal{R}_e} < \| e^e_k \|_{\mathcal{R}_e} \quad \forall k \geq 0 \]  \hspace{1cm} (19)

Moreover, if \( \tilde{w}_1 > 0 \), the consequent iteration sequence satisfies the following limiting behaviours

\[ \lim_{k \to \infty} e^e_k = 0 \quad \& \quad \sum_{k \geq 0} \beta_{k+1} < \infty. \]  \hspace{1cm} (20)

In particular, \( \beta_{k+1} \to 0^+ \) as \( k \to \infty \) and the algorithm again converges monotonically to the required solution of the tracking problem with auxiliary optimization i.e.

\[ \lim_{k \to \infty} u_k = u_\infty. \]  \hspace{1cm} (21)

**Proof:** See [5]. \( \square \)

The following observations concern parameter choice:

- In the special case of \( \tilde{w}_1 = 0 \), the resultant gains are constant at the fixed value \( \beta_{k+1} = \frac{1}{\tilde{w}_2+1} \).
- If \( \tilde{w}_1 > 0 \), then the algorithm behaves like a fixed gain algorithm whenever \( \tilde{w}_2 \approx \| e^e_k \|_{\mathcal{R}_e}^2 \), but, as the norm reduces, gains reduce and ultimately go to zero. As a guide, when \( \tilde{w}_1 \approx \tilde{w}_2 \| e^e_k \|_{\mathcal{R}_e}^2 \), then \( \beta_{k+1} \approx \frac{1}{\tilde{w}_2+1} \).

### IV. COMPUTATIONAL IMPLEMENTATION

**A. Computation of the Operator \( L \)**

The operator \( L \) can be computed by the following NOILC-like optimization problems:

- **Step 1:** As \( \mathcal{R}_e \) is finite dimensional of dimension \( n_e \), then choose a basis in \( \mathcal{R}_e \) and represent vectors in \( \mathcal{R}_e \) and operators in this basis.
- **Step 2:** As \( G_0^* : \mathcal{R}_e \rightarrow \mathcal{U} \), identify \( G_0^* \) with a matrix \( G_0^* = [g_1, \ldots, g_{n_e}] \) of vectors \( g_j \in \mathcal{U} \).
- **Step 3:** Compute \( L_0 = (I + G_1^* G_1)^{-1} G_0^* = [(I + G_1^* G_1)^{-1} g_1, \ldots, (I + G_1^* G_1)^{-1} g_{n_e}] \) where each \( g_j \) can be computed as \( g_j - \varphi_j \) where \( \varphi_j = G_1^* v_j \) solves the optimization problem \( \varphi_j, v_j = \arg \min \{ \| g_j - \varphi \|_{\mathcal{U}} + \| v \|_{\mathcal{Z}} : v \in \mathcal{Z}, \varphi = G_1^* v \} \).
- **Step 4:** Compute the matrix \( G_0 L_0 = [G_0 k_1, \ldots, G_0 k_{n_e}] \) and set \( L = L_0 (G_0 L_0)^{-1} \).

**B. Linear Continuous State Space Systems**

The above procedure is now specialized to LTI continuous-time state space systems with operators given in section II-B.

- **Step 1:** The choice of basis in \( \mathcal{R}_e \) is trivial for this application as the basis is implicit in the matrix description used to define \( A, B, C \) in (6).
- **Step 2:** From (10), the operator \( G_0^* \) is the map \( (w_1, \ldots, w_M) \rightarrow u \) defined by

\[ u(t) = \sum_{j=1}^{M} (G_0^* w_j)(t) = R^{-1} B^T \sum_{j=1}^{M} P_j(t) w_j \]

where each \( P_j(t) = 0 \), \( t \in (t_j, T) \) and, on \( [0, t_j] \),

\[ P_j(t) = -A^T P_j(t), \quad P_j(t_j) = C^T F^T Q_j, \]

\[ (G_0^* w_j)(t) = R^{-1} B^T P_j(t) w_j \]

Hence \( G_0^* \) is simply the time varying \( \ell \times n_e \) matrix

\[ [R^{-1} B^T P_j(t), \ldots, R^{-1} B^T P_M(t)] \]

with \( g_j(t) \) the \( j \)th column of this matrix.
Step 3: The operator \( L_0 \) is a \( \ell \times n_r \) time varying matrix whose \( j^{th} \) column, written \( g_j(t) - \varphi_j^*(t) \), is obtained from

\[
(\varphi_j^*, v_j^*) = \arg\min \left[ \int_0^T \left( (g_j(t) - \varphi_j^*(t))^T R(g_j(t) - \varphi_j^*(t)) + v_j^T(t) Q v_j(t) \right) dt \right]
\]

subject to the dynamic constraints \( \varphi_j = G_1^* v_j \) which is just the \( r \)-output, \( m_z \)-input system constraint

\[
\begin{align*}
\dot{\eta}_k(t) &= -A^T \eta_k(t) - C^T_z Q v_j(t), \\
\varphi_j(t) &= R^{-1}(B^T \eta_k(t) + D^T_z Q v_j(t))
\end{align*}
\]

with terminal boundary condition \( \eta_k(T) = 0 \). If \( \tilde{f}(t) = f(T-t) \) for any signal \( f \), then the solution is obtained by one trial of NOILC [6] using initial input \( v = 0 \), plant model \( S(A^T, C^T_z Q, R^{-1} B^T, R^{-1} D^T_z Q) \) and reference \( \bar{y}_j \).

Step 4: The \( j^{th} \) column of \( G_0 L_0 \) is the response of the extended output \( y^e \) to the \( j^{th} \) column of \( L_0(t), 1 \leq j \leq n_r \) after which \( L(t) = L_0(t)(G_0 L_0)^{-1} \) is routinely computed.

V. MODEL PERTURBATIONS & ROBUSTNESS

A. Additive Perturbations

Suppose \( G_0 \) is deduced from a model of the plant and \( G_0 + \Delta G_0 \) is its actual value, and similarly that \( d_0 \) is actually \( d_0 + \Delta d_0 \). The observed plant output is then \( y^e_k = (G_0 + \Delta G_0) u_k + (d_0 + \Delta d_0) \) and the observed on-line plant error

\[
e^e_k = r^e - y^e_k = r^e - (G_0 + \Delta G_0) u_k - (d_0 + \Delta d_0) \quad \forall \ k \geq 0.
\]

If \( u_k \) generates \( e^e_k \) from the actual plant on-line, then the next input \( u_{k+1} \) is computed off-line using

\[
u_{k+1} = u_k + \beta_{k+1} L e^e_k = u_k + \beta_{k+1} L_0(G_0 L_0)^{-1} e^e_k
\]

where \( L \) is computed using models \( G_0 \) and \( G_1 \). It is assumed that \( \beta_{k+1} \) is defined by \( \beta_{k+1} = \beta(e^e_k) \in (0,1) \) whenever \( \|e^e_k\|_{R_e} \neq 0 \). It follows that

\[
e^e_{k+1} = ((1 - \beta_{k+1}) I - \beta_{k+1} U) e^e_k, \quad \forall \ k \geq 0. \tag{23}
\]

where \( U \) is the change induced by the plant-model mismatch

\[
U = \Delta G_0 L \tag{24}
\]

Note \( d_0 \) has no effect on the result.

Theorem 3: With the above construction, the inverse-model algorithm reduces the error norm on each iteration if the induced norm of \( U \) in \( R_e \) satisfies

\[
\|U\|_{R_e} < 1 \tag{25}
\]

Moreover, the error norm of \( e^e_k \) converges to zero as \( k \to \infty \).

Proof: See [5]. \( \Box \)

Condition (25) can be replaced by the more conservative

\[
\|\Delta G_0\| < \left. \frac{1}{\|L_0(G_0 L_0)^{-1}\|} \right.
\]

which separates the model effects from the mismatch. This can be improved using the following result:

Lemma 1: Suppose that \( L_1 \) is a bounded operator, self-adjoint and positive semi-definite on \( U \) with \( \delta^2 I \leq L_1 \leq \bar{\delta}^2 I \) for some scalars \( 0 \leq \delta^2 \leq \bar{\delta}^2 \). Then

\[
\| (I + L_1)^{-1} G_0^*(G_0(I + L_1)^{-1} G_0^*)^{-1} \| \geq \| G_0^*(G_0 G_0^*)^{-1} \| = \frac{1}{\lambda^2_{\min}(G_0 G_0^*)}
\]

where \( \lambda_{\min}(G_0 G_0^*) \) is the smallest eigenvalue of \( G_0 G_0^* \). In particular, the norm of \( (I + L_1)^{-1} G_0^*(G_0(I + L_1)^{-1} G_0^*)^{-1} \) has a minimum when \( L_1 = 0 \). Also

\[
\left( \frac{1}{1 + \delta^2} \lambda^2_{\min}(G_0 G_0^*) \right)^{1/2} \geq \| (I + L_1)^{-1} G_0^*(G_0(I + L_1)^{-1} G_0^*)^{-1} \|
\]

Proof: See [5]. \( \Box \)

Setting \( L_1 = G_1^* G_1 \) leads to the sufficient condition

\[
\|\Delta G_0\| < \lambda^1_{\min}(G_0 G_0^*) \left( \frac{1 + \bar{\delta}^2}{1 + \delta^2} \right)^{1/2}, \tag{27}
\]

which separates the effects of tracking and auxiliary optimization requirements, relating robustness to the spectrum of \( G_0 G_0^* \) and upper and lower bounds on \( G_1^* G_1 \) respectively.

For stable, LTI continuous time state space systems

\[
\delta^2 = 0 \quad \& \quad \bar{\delta}^2 = \sup_{\omega \geq 0} \lambda_{\max}(R^{-1} G_1(-i\omega) Q G_1(i\omega))
\]

where \( \lambda_{\max}(\cdot) \) is the largest eigenvalue of its argument and \( G_1(s) \) denotes the transfer function matrix of operator \( G_1 \).

The next theorem shows that the right-hand side of (27) increases as points are removed from the tracking task.

Theorem 4: Using the above discussion suppose that the \( n_r \times n_r \) matrix \( G_0 G_0^* \) has eigenvalues \( \lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n_r} \). Then the removal of any single tracking requirement within \( r^e \) generates a new \( (n_r - 1) \times (n_r - 1) \) matrix \( G_0 G_0^* \) with eigenvalues \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n_r-1} \) satisfying the interlacing condition

\[
\lambda_{\min}(G_0 G_0^*) = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n_r-1} \leq \lambda_{n_r}
\]

In particular, the smallest eigenvalue increases in value and the robustness bound (27) predicts an increase in robustness.

Proof: See [5]. \( \Box \)

B. Multiplicative Perturbations

Now \( U \) plays the role of a multiplicative modelling error, with nominal value \( I \), and the error evolution is

\[
e^e_{k+1} = (I - \beta_{k+1} U) e^e_k \quad \text{with} \quad U = (G_0 + \Delta G_0) L \tag{28}
\]

Conditions for monotone robust convergence are next given.

Theorem 5: Suppose that \( U \) is invertible and \( \beta_{k+1} \in (0,1) \). Then the inequality \( \|e^e_{k+1}\|_{R_e} < \|e^e_k\|_{R_e} \) holds iff

\[
U^{-1} + (U^*)^{-1} > \beta_{k+1} I \quad \text{(i.e.} \quad U + U^* > \beta_{k+1} U^* U) \tag{29}
\]

where \( U^* \) denotes the adjoint of \( U \) in the \( R_e \) topology. In particular, the inverse model law is robustly convergent if

\[
U^{-1} + (U^*)^{-1} > \beta_1 I
\]
where $\beta_1$ is the gain used on the first iteration.

Proof: See [5]. □

Notes: Whatever the value of $\beta_1$ to be used, $U + U^*$ must be strictly positive in the topology of $\mathcal{R}_{w}$ for robust monotone convergence to be achieved. As $\beta_1$ decreases, the algorithm can tolerate a greater range of uncertainty $U$. Suppose that $U + U^* > 0$ and hence define $\beta^*(n_r) = \sup \{ \beta : U + U^* \geq \beta U^* U \} = \lambda_{\text{min}}(U^{-1} + (U^*)^{-1}) > 0$.

In effect, $\beta^*(n_r) > 0$ is the lowest upper bound on those $\beta_1$ such that the robustness condition of the theorem is satisfied. The theorem is hence equivalent to the statement $\beta_1 < \beta^*(n_r)$ which provides a simple bound on the permitted magnitude of the initial error norm with links to the chosen weights $\overline{w}_1, \overline{w}_2$ i.e. writing $\beta_1 < \beta^*(n_r)$ as

$$(1 - \beta^*(n_r)(1 + \overline{w}_2))\|e_0\|^2_{\mathbb{R}_+} < \beta^*(n_r)\overline{w}_1$$

VI. EXPERIMENTAL EVALUATION

Experimental evaluation is now used to verify performance and support algorithm robustness.

A. Problem Statement and Preliminary Analysis

The approach has been tested on a 6 degree of freedom robotic arm whose 5 rotary joints are composed of Power Cubes (Schunk GmbH & Co.) incorporating brushless servomotors with integrated power electronics and transmission. Results are presented for the first joint which is aligned in the horizontal plane as shown in Fig. 1. Frequency response tests have identified the linear model (30), with angular input and output in degrees. A continuous-time state space model $S(A, B, C)$ is used for computation.

The intermediate point task selected replicates an industrial ‘pick and place’ process in which payloads are manipulated during assembly. The reference is given by

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

(31)

defining the desired outputs at the $M = 4$ time points

$$t_1 = 1, \quad t_2 = 3, \quad t_3 = 5, \quad t_4 = T = 6.$$  (32)

on the time interval $[0, 6]$. Using (32), $G_0$ is then given by (8) with $F_j = 1, 1 \leq j \leq M$, whose elements (9) map the plant input to its output at time $t_j$. $G_0$ is calculated using (10), with equal weighting at all points, $Q_j = 1, 1 \leq j \leq M$, which hence governs convergence of the output tracking (2).

To reduce vibration and potential payload damage/spillage, minimization of the joint acceleration is considered. The auxiliary variable is hence $\bar{z}(t) = \dot{y}(t)$, and $G_1$ can be written using $S(A_2, B_2, C_2) = S(A, B, C^2)$. The weight $Q$ hence governs the relative emphasis on minimizing joint acceleration compared with control effort, via objective function (4).

Table I shows the robustness bound (27) which closely approximates that of (26) at low $Q$ values. By separating the effects of tracking and auxiliary minimization objectives, (27) provides a transparent method of assessing robustness.

Note: The ‘standard’ inverse algorithm [7] designed for the situation where $r^e$ is specified at all points along the trial, $G_0 = G$ and $G_1 = 0$ yields a value of $\|L_0(G_0L_0)^{-1}\|^{-1} = 1.4 \times 10^{-7}$, indicating poor robustness, confirming that robustness increases as points are removed from the tracking objective. Note: $G_1 = 0$ provides an upper limit on $\|L_0(G_0L_0)^{-1}\|^{-1}$ of 1.1628, corresponding to greatest robustness. In the results which follow, all algorithms have been calculated in continuous-time using the results of section II-B and section IV-B and applied to the system using zero-order hold discretization at 500 Hz.

B. Inverse algorithm results

The inverse algorithm has been implemented on the robotic test facility for the cases where $Q = 0.0006, 0.002, 0.006, 0.02, 0.06$ and 0.2. The weight $\bar{w}_1$ has taken the values 0, 0.2, 0.5, 1, 2 and $\bar{w}_2$ the values 0.1, 0.5, 1, 2, 10. To investigate long term stability, 500 trials have been carried out. Selecting $\beta_1$ on the $k^{th}$ trial to minimize (15) resolves the problem of instability while maintaining fast convergence, as long as $\bar{w}_1$ is sufficiently high. Figure 2 shows norm results using $Q = 0.2$, $\bar{w}_2 = 0.1$, and various $\bar{w}_1$. Here $\beta_1$ is $\approx 0.9091$ which clearly is not robust if fixed at this value over multiple trials, however choosing $\bar{w}_2 \geq 0.5$ resolves the problem of instability and leads to rapid learning. This hence confirms Theorem 3. Final trial input and output signals appear in Fig. 3 for a range of $Q$. Choice of larger $\bar{w}_2$ reduces $\beta_1$ to avoid a lack of robustness, as shown in Figure 4. Here $\beta_1 = 0.5$ and all values of $\bar{w}_1$ considered produce convergent learning. The convergent rate is slightly slower compared with using smaller $\bar{w}_2$, however the reduction in error and auxiliary signal norms is less oscillatory for all $\bar{w}_1$ values. These results hence confirm the observations made in section V-B.

VII. CONCLUSIONS

A powerful framework has been developed for ILC problems requiring tracking of a finite dimensional reference
\[
G(s) = \frac{400788.1582(s + 12.14)(s + 24.01)^2}{(s + 31.52)(s + 22.97)(s + 2.178)(s^2 + 36.59s + 363.7)(s^2 + 124.5s + 4076)}
\]  
(30)

while minimizing an objective function. The proposed algorithms have been tested experimentally on a robotic manipulator, with results confirming robustness and performance.

**REFERENCES**


