

Norm-Optimal Iterative Learning Control with Application to Problems in Accelerator-Based Free Electron Lasers and Rehabilitation Robotics

Eric Rogers^{1,*}, David H. Owens², Herbert Werner³, Chris T. Freeman¹, Paul L. Lewin¹, Stefan Kichhoff³, Christian Schmidt⁴, Gerwald Lichtenberg³

¹ School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, United Kingdom;

² Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom;

³ Institute of Control Systems, Hamburg University of Technology, 21071 Hamburg, Germany;

⁴ DESY Hamburg, Notkestrasse 85, 22607 Hamburg, Germany

This article gives an overview of the theoretical basis of the norm-optimal approach to iterative learning control followed by results that describe more recent work which has experimentally benchmarked the performance that can be achieved. The remainder of the article then describes its actual application to a physical process and a very novel application in stroke rehabilitation.

Keywords: Iterative learning control, Theory, Algorithms, Experimental application

1. Introduction

Iterative Learning Control (ILC) has been especially developed to improve the performance of systems that operate in a repetitive manner where the task is to follow some specified trajectory in a specified finite time interval, also known as a pass or a trial in the literature, with high precision. The novel principle behind ILC is to suitably use information from previous trials, often in combination with appropriate current trial information, to select the current trial input to sequentially improve performance from trial-to-trial. In particular, the aim is to improve performance from trial-to-trial in the sense that

the tracking error, the difference between the output on a trial and the specified reference trajectory, is sequentially reduced to either zero, ideal case, or some suitably small value.

The original work in this area is credited to [6] and since then there have been substantial developments in both systems theoretic and application terms. For an overview of the algorithm development side see, for example, [1] where this reference has the added feature of a categorization of algorithms developed up to 2004. Application areas where ILC has been successfully applied include robotics, automated manufacturing plants, and food processing. For more details, including some of those where there is clear potential for significant added benefit from fully developed ILC, one possible source is the survey article [9].

In general, current research and development in ILC can, as in other areas, be broadly partitioned into starting from either a linear or nonlinear model of the plant dynamics but here we restrict attention to the former where there are still many open problems. This is especially true at the interface between theory and applications. Given the diverse range of algorithms, which have been developed over the years, there is a clear need to develop tools and case studies that allow for valid comparison of competing designs. This article begins by giving the basic

*Correspondence to: E. Rogers; E-mail: etar@ecs.soton.ac.uk

Received 30 September 2009; Accepted 18 May 2010
Recommended by Hyo-Sung Ahn, D.W. Clarke

ideas and results that led to the development of the norm-optimal class of algorithms [3, 4, 5]. The development then continues to describe experimental benchmarking using laboratory-based facilities especially designed and constructed for this purpose. This is followed by more recent results where such algorithms have been applied, using a computationally more efficient implementation resulting from the laboratory-based experimental benchmarking, to free electron lasers (FELs). Following this, a novel application of norm-optimal ILC to stroke rehabilitation is described and the article concludes by a brief overview of some areas of ongoing work and possible future research directions.

2. Norm-Optimal ILC Control Theory

The mathematical definition of ILC used in this article has the following general form, where the results in this section are mainly from [3].

Definition 1: Consider a dynamic system with input u and output y . Let \mathcal{Y} and \mathcal{U} be the output and input function spaces, respectively, and let $r \in \mathcal{Y}$ be a desired reference trajectory for the system. An ILC algorithm is successful if, and only if, it constructs a sequence of control inputs $\{u_k\}_{k \geq 0}$ which, when applied to the system (under identical experimental conditions), produces an output sequence $\{y_k\}_{k \geq 0}$ with the following properties of convergent learning

$$\lim_{k \rightarrow \infty} y_k = r, \quad \lim_{k \rightarrow \infty} u_k = u_\infty$$

Here convergence is interpreted in terms of the topologies assumed in \mathcal{Y} and \mathcal{U} , respectively.

A major advantage of this general statement of the problem is that it allows a simultaneous description of linear and nonlinear dynamics, continuous or discrete plant with either time-invariant or time-varying dynamics.

Let the space of output signals \mathcal{Y} be a real Hilbert space and also let \mathcal{U} be a real, and possibly distinct, Hilbert space of input signals. The respective inner products, denoted by $\langle \cdot, \cdot \rangle$, and norms $\| \cdot \|^2 = \langle \cdot, \cdot \rangle$ are indexed in a way that reflects the particular space concerned, for example, $\|x\|_{\mathcal{Y}}$ denotes the norm of $x \in \mathcal{Y}$. The Hilbert space structure induced by the inner product is essential in what follows but is not restrictive, as specific choices of this structure enables the analysis of, for example, continuous or discrete-time systems.

The dynamics of the systems considered here are assumed to be linear and represented in operator form as

$$y = Gu + z_0 \quad (1)$$

where $G : \mathcal{U} \rightarrow \mathcal{Y}$ is the system input/output operator (assumed to be bounded and typically a convolution operator) and z_0 represents the effects of system initial conditions. If $r \in \mathcal{Y}$ is the reference trajectory or desired output, the tracking error is defined as

$$e = r - y = r - Gu - z_0 = (r - z_0) - Gu \quad (2)$$

Hence, without loss of generality, it is possible to replace r by $r - z_0$ and consequently assume that $z_0 = 0$.

The ILC procedure, if convergent, solves the problem $r = Gu_\infty$ for u_∞ . If G is invertible, then the formal solution is just $u_\infty = G^{-1}r$. A basic assumption in ILC is that direct inversion of G is not acceptable as, for example, this would require exact knowledge of the plant and involve derivatives of the reference trajectory. This high-frequency gain characteristic would make the approach sensitive to noise and other disturbances. Also inversion of the whole plant G is unnecessary as the solution only requires finding the pre-image of the reference trajectory r under G .

The problem considered here is easily seen to be equivalent to finding the minimizing input u_∞ for the optimization problem

$$\min_u \{ \|e\|_{\mathcal{Y}}^2 : e = r - y, \quad y = Gu \} \quad (3)$$

The optimal error $\|r - Gu_\infty\|_{\mathcal{Y}}^2$ is a measure of how well the ILC procedure has solved the inversion problem. It also represents the best the system can do in tracking the signal r . The particular interest here is the case when the optimal error is exactly zero, i.e. when u_∞ is a solution of $r = Gu_\infty$ and hence solves the ILC problem.

There are many iterative procedures to solve the optimization problem (3) but there is a clear advantage in the use of descent algorithms of a suitable type as considered for learning systems by, for example, [20]. The gradient-based ILC algorithm class generates the control input to be used on trial $k + 1$ as

$$u_{k+1} = u_k + \epsilon_{k+1} G^* e_k$$

where $G^* : \mathcal{Y}^* \rightarrow \mathcal{U}^*$ is the adjoint operator of G and ϵ_{k+1} is a step length to be chosen on each trial. This approach suffers from the need to choose a step length and the feedforward structure of the control law in k that takes no account of current trial effects, including disturbances and plant modeling errors.

The norm-optimal ILC algorithm class has the following two important properties.

- Automatic choice of step size.
- Potential for improved robustness through the use of causal feedback of current trial data and feedforward of data from previous trials.

This is achieved by, on completion of trial k , computing the control input on trial $k + 1$ as the solution of the minimum norm optimization problem

$$u_{k+1} = \arg \min_{u_{k+1}} \{J_{k+1}(u_{k+1}) : e_{k+1} = r - y_{k+1}, \\ y_{k+1} = Gu_{k+1}\}$$

where the performance index used is

$$J_{k+1}(u_{k+1}) := \|e_{k+1}\|_{\mathcal{Y}}^2 + \|u_{k+1} - u_k\|_{\mathcal{U}}^2 \quad (4)$$

The initial control $u_0 \in \mathcal{U}$ can be arbitrary in theory but, in practice, will be a good first guess at the solution of the problem.

The solution of this ILC minimum norm optimization problem can be interpreted as the determination of the control input on trial $k + 1$ with the properties that: (i) the tracking error is reduced in an optimal way; and (ii) this new control input does not deviate too much from the control input used on trial k . The relative weighting of these two objectives can be absorbed into the definitions of the norms in \mathcal{Y} and \mathcal{U} .

The benefits of this approach are immediate from the simple interlacing result

$$\|e_{k+1}\|_{\mathcal{Y}}^2 \leq J_{k+1}(u_{k+1}) \leq \|e_k\|_{\mathcal{Y}}^2, \quad \forall k \geq 0 \quad (5)$$

which follows from optimality and the fact that the nonoptimal choice of $u_{k+1} = u_k$ would lead to the relation $J_{k+1}(u_k) = \|e_k\|_{\mathcal{Y}}^2$. Hence we are considering a descent algorithm as the norm of the error is monotonically non-increasing in k . Also equality holds if, and only if, $u_{k+1} = u_k$, i.e. when the algorithm has converged and no more input-updating takes place.

The control law on trial $k + 1$ is obtained by Fréchet differentiation of (4) with respect to u_{k+1} to obtain the stationary point and substitution from (1) and (2) to obtain

$$u_{k+1} = u_k + G^* e_{k+1}, \quad \forall k \geq 0 \quad (6)$$

where G^* denotes the adjoint of G . This equation is the formal update relation for the class of norm-optimal ILC algorithms.

Using $e = r - Gu$ gives the tracking error update relation

$$e_{k+1} = (I + GG^*)^{-1} e_k, \quad \forall k \geq 0$$

and the recursive relation for the input evolution

$$u_{k+1} = (I + G^* G)^{-1} (u_k + G^* r), \quad \forall k \geq 0$$

Norm-optimal ILC has a number of other useful properties. For example, monotonicity immediately shows that the following limits exist

$$\lim_{k \rightarrow \infty} \|e_k\|_{\mathcal{Y}}^2 = \lim_{k \rightarrow \infty} J_k(u_k) =: J_{\infty} \geq 0$$

Also an inductive argument and the inequality $\|y\| \leq \|G\| \|u\|$ yields the relations

$$\sum_{k \geq 0} \|u_{k+1} - u_k\|_{\mathcal{U}}^2 < \|e_0\|_{\mathcal{Y}}^2 - J_{\infty} < \infty$$

$$\sum_{k \geq 0} \|e_{k+1} - e_k\|_{\mathcal{Y}}^2 < \|G\|^2 (\|e_0\|_{\mathcal{Y}}^2 - J_{\infty}) < \infty \quad (7)$$

and hence

$$\lim_{k \rightarrow \infty} \|u_{k+1} - u_k\|_{\mathcal{U}}^2 = 0, \quad \lim_{k \rightarrow \infty} \|e_{k+1} - e_k\|_{\mathcal{Y}}^2 = 0 \quad (8)$$

The properties given in (8) show that the algorithm has an implicit choice of step size, the first distinguishing property of norm-optimal ILC noted above, as the incremental input converges to zero. This asymptotic slow variation is a prerequisite for convergence. Also the summation of the energy costs from the first to the last trial is bounded, as shown by (7).

The following result gives the proof of convergent learning, where $\mathcal{R}(\mathcal{A})$ is used to denote the range of an operator \mathcal{A} . This is proved as Theorem 1 in [3].

Theorem 1: *If either $r \in \mathcal{R}(G)$ or $\mathcal{R}(G)$ is dense in \mathcal{Y} , then the ILC tracking error sequence $\{e_k\}_k$ converges in norm to zero in \mathcal{Y} , i.e. the ILC algorithm has guaranteed convergence of learning.*

The guaranteed convergence together with the monotonicity of the tracking error sequence represent powerful properties of the algorithm. To be applicable to an example, obviously requires a causal implementation where the presence of this property is not immediately obvious here as the relation $u_{k+1} = u_k + G^* e_{k+1}$, although apparently of a feedback form, suggests that the relationship is not causal. For example, if G is the convolution operator in $L_2^m[0, T]$, with the inner product $\langle w, v \rangle_{L_2^m[0, T]} = \int_0^T w^T(t) v(t) dt$, described by the relation $(Gu)(t) := \int_0^t K(t-\tau) u(\tau) d\tau$, then $(G^* e)(t) = \int_t^T K^T(\tau-t) e(\tau) d\tau$. This means that evaluation of $G^* e_{k+1}$ requires knowledge of future values of the tracking errors. Such data are not, of course, available in practice. The special causality structure of ILC allows, however, the transformation of the algorithm into a causal procedure, as detailed for one case of particular interest below.

In mathematical terms, the trial-to-trial error always goes to zero but this does not imply convergence of the input sequence in \mathcal{U} unless this space is chosen appropriately. As an example, suppose that both \mathcal{Y} and \mathcal{U} are L_2 type spaces and G arises from a linear time-invariant

system described by a state-space model. Then if the state initial vector $x(0)$ does not generate an output that matches the value of the reference vector at $t = 0$, the required u_∞ will contain distributions such as the Dirac delta function. Hence $u_\infty \notin \mathcal{U}$ and a proof that $u_k \rightarrow u_\infty$ in \mathcal{U} is impossible. Consequently the following results are conditional on additional assumptions on either the input sequence applied or the plant itself. Here we use the latter and the following result deals with general convergence of the input (the proof is that for Theorem 3 in [3]).

Theorem 2: *The sequence $\{u_k\}_{k \geq 0}$ satisfies*

$$\lim_{k \rightarrow \infty} \|G^*(r - Gu_{k+1})\|_{\mathcal{U}} = 0 \quad (9)$$

*Also if G^*G has a bounded inverse in \mathcal{U} the input sequence converges in norm to*

$$u_\infty = (G^*G)^{-1}G^*r \in \mathcal{U} \quad (10)$$

and if $\sigma := \frac{1}{\|G\|_{\mathcal{U}}} > 0$ (where $\|\cdot\|$ denotes the induced operator norm) the convergence is bounded by a geometric relation of the form

$$\|u_{k+1} - u_\infty\|_{\mathcal{U}} \leq \frac{1}{1 + \sigma^2} \|u_k - u_\infty\|_{\mathcal{U}} \quad (11)$$

This last result only holds with the boundedness assumption imposed on the plant inverse by the assumption that $\sigma^2 > 0$. The following result, whose proof is that for Theorem 4 in [3] relaxes this assumption.

Theorem 3: *If the sequence $\{u_k\}_{k \geq 0}$ is bounded in \mathcal{U} , the desired, or learned control, input $u_\infty \in \mathcal{U}$ and G^*G has range dense in \mathcal{U} , then this sequence converges to u_∞ in the weak topology in \mathcal{U} .*

For finite-dimensional spaces, weak convergence is equivalent to convergence in norm. In such cases, we have also proved convergence in norm. This fact includes discrete-time systems, which are the setting for applications to physical processes.

The modified ILC law

$$u_{k+1} = \alpha u_k + G^*e_k \quad (12)$$

provides a link to almost singular optimal control [16]. Here α is a relaxation parameter, as used commonly in numerical analysis to improve algorithm robustness or a forgetting factor in adaptive control. Setting $\alpha = 1$ recovers the case considered above and substituting from $e = r - Gu$ in the expression for $e_{k+1} - e_k$ gives

$$e_{k+1} = (I + GG^*)^{-1}[\alpha e_k + (1 - \alpha)r] \quad (13)$$

Theorem 1 has proved convergence when $\alpha = 1$. When $\alpha \neq 1$, it is possible to use results from the stability theory

of linear repetitive processes [38] to show that convergence in this case holds if, and only if, $|\alpha| < 1$ with non-zero limit error $\hat{e}_\infty \in \mathcal{Y}$ given by

$$\hat{e}_\infty = \left(I + \frac{GG^*}{1 - \alpha}\right)^{-1} r \quad (14)$$

Using the plant equation, it follows that the input sequence also satisfies

$$u_{k+1} = (I + GG^*)^{-1}(\alpha u_k + G^*r) \quad (15)$$

The norm of the recursion operator $(I + G^*G)^{-1}\alpha$ in this case is $|\alpha|$, and it follows immediately that the input sequence converges geometrically in norm in \mathcal{U} if, again, $|\alpha| < 1$ to the limit

$$\hat{u}_\infty = ((1 - \alpha)I + G^*G)^{-1}G^*r \quad (16)$$

with geometric constant $|\alpha|$. Note that the control input vector in this case converges in norm if relaxation is used. Also for convergence to a solution close to u_∞ , α must be chosen to be close to, but slightly less than, unity. This follows from the fact that \hat{u}_∞ and \hat{e}_∞ are the solutions of the optimization problem

$$\min_u [\hat{J}(u) = \|e\|_{\mathcal{Y}}^2 + (1 - \alpha)\|u\|_{\mathcal{U}}^2 : y = Gu, e = r - y] \quad (17)$$

It now follows that it is possible to make $\|e\|_{\mathcal{Y}}^2$ arbitrarily small with controls $u \in \mathcal{U}$ and hence the minimum value of \hat{J} tends to zero as α approaches one from below. The following result can therefore be established (Theorem 5 in [3]).

Theorem 4: *Under the assumptions of Theorem 1, the ILC algorithm with modified update law (12) with $|\alpha| < 1$ converges in norm in \mathcal{U} to a control input that produces a non-zero limit error with norm that can be made arbitrarily small by choosing α arbitrarily close to unity.*

If $\alpha < 1$ is close to unity then the control input weighing in (17) is very close to zero. This is the essential link to the almost singular or cheap control problem [13, 16].

For applications, the results given above have to be converted into computational procedures. Central to this is that the resulting algorithms are causal in the ILC sense. The formal definition of this property is as follows.

Definition 2: *An ILC algorithm is causal if, and only if, the value of the input $u_{k+1}(t)$ at time t on trial $k + 1$ is computed only from data that are available from this trial in the time interval $[0, t]$ and from previous trials over the complete trial duration $[0, T]$.*

Note that this definition differs from the classical one as data from times $\hat{t} > t$ can be used, but only from previous trials. In the next section, we switch to the case of discrete linear time-invariant plant models as preparation for the application studies, which follow in the remainder of the article. An efficient implementation procedure maximizing computational efficiency of the resulting algorithm is also introduced as this has proved to be essential in the majority of the current applications where norm-optimal ILC has been experimentally applied.

A natural extension of norm-optimal ILC, termed predictive norm-optimal ILC, can also be derived. The intuition that motivated this work [4] is that predictive control contains the key to improved performance. Predictive norm-optimal ILC extends the performance criterion to take future predicted error signals into account. The extended criterion for computing the input u_{k+1} on trial $k + 1$ is

$$J_{k+1}(u_{k+1,N}) := \sum_{i=1}^N \lambda^{i-1} \left(\|e_{k+i}\|_{\mathcal{Y}}^2 + \|u_{k+i} - u_{k+i-1}\|_{\mathcal{U}}^2 \right)$$

This criterion includes the error not only of the next trial, but of the next N trials, together with the corresponding changes in input. The weight $\lambda > 0$ determines the importance of more distant (future) errors and incremental inputs compared to the present ones.

The actual ILC algorithm follows uniquely from minimization of the proposed cost criterion. It only remains to compute the minimizing input. This is done by dynamic programming. Once the input is found, a recursive formulation for the evolution of the error (and input) is computed. All interesting properties and characteristics of the norm-optimal predictive ILC algorithm can, as in the norm-optimal case, be obtained by examining the properties of the operators appearing in the recursive formulation.

3. Norm-Optimal ILC Computation

The process dynamics are assumed to be represented, after sampling if necessary, by the discrete linear time-invariant systems state-space model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \quad x(0) = x_0, \quad 0 \leq t \leq N_s \\ y(t) &= Cx(t) \end{aligned} \quad (18)$$

where N_s is the number of samples, $x(t)$ is the $n \times 1$ state vector, $y(t)$ is the $m \times 1$ output vector and $u(t)$ is

the $l \times 1$ control input vector. Also because N_s is finite, introduce the supervectors

$$y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N_s) \end{bmatrix}, \quad u = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N_s - 1) \end{bmatrix} \quad (19)$$

Then, using the transition matrix solution for $y(t)$ of (18), we can describe the process dynamics by

$$y = y_0 + Gu \quad (20)$$

where the $mN_s \times lN_s$ matrix G is given by

$$G = \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_s-1}B & CA^{N_s-2}B & \dots & CB \end{bmatrix} \quad (21)$$

and

$$y_0 = \begin{bmatrix} (CA)^T & (CA^2)^T & \dots & (CA^{N_s})^T \end{bmatrix}^T x_0$$

In the single-input single-output (SISO) case the Toeplitz matrix G is invertible if, and only if, $CB \neq 0$. This matrix could also be of very large dimensions but this is not a problem as it does not appear in the final calculations. If the system has a delay, and hence $CB = 0$, then it can be regularized as detailed in [5].

In the abstract setting, the spaces \mathcal{Y} and \mathcal{U} are taken as ℓ_2 spaces of $m \times 1$ and $l \times 1$ vectors on $[1, N_s]$ and $[0, N_s - 1]$, respectively. Writing the norms out as sums gives the performance index as

$$\begin{aligned} J_{k+1}(u_{k+1}) &= \frac{1}{2} \left(\sum_{t=1}^{N_s} (r(t) - y_{k+1}(t))^T Q (r(t) - y_{k+1}(t)) \right. \\ &\quad \left. + \sum_{t=0}^{N_s-1} (u_{k+1}(t) - u_k(t))^T R (u_{k+1}(t) - u_k(t)) \right) \end{aligned} \quad (22)$$

where the weighting matrices Q and R are of compatible dimensions, and symmetric positive definite. This is the familiar linear quadratic performance criterion from linear quadratic optimal control theory which is, in effect, a combination of the optimal tracking (tracking of $r(t)$) and the disturbance accommodation problem (regarding $u_k(t)$ as a known disturbance on trial $k + 1$).

Forming the block diagonal matrices \hat{Q} and \hat{R} with Q and R as the diagonal entries respectively, the definitions of the inner products in \mathcal{Y} and \mathcal{U} are

$$\langle y_1, y_2 \rangle_{\mathcal{Y}} = y_1^T \hat{Q} y_2 = \sum_{t=1}^{N_s} y_1(t)^T Q y_2(t) \quad (23)$$

$$\langle u_1, u_2 \rangle_{\mathcal{U}} = u_1^T \hat{R} u_2 = \sum_{t=0}^{N_s-1} u_1(t)^T R u_2(t) \quad (24)$$

The initial control $u_o \in \mathcal{U}$ can be arbitrary in theory but, in practice, it will be chosen to be a good first guess at the solution of the problem.

The control input on trial $k+1$ that minimizes the cost function here is obtained from the stationary condition

$$\frac{\partial J_{k+1}}{\partial u_{k+1}} = -G^T Q e_{k+1} + R(u_{k+1} - u_k) = 0 \quad (25)$$

or, since R is invertible,

$$u_{k+1} = u_k + R^{-1} G^T Q e_{k+1}, \quad \forall k \geq 0 \quad (26)$$

Here $R^{-1} G^T Q$ is equivalent to the adjoint operator G^* with respect to the weighted inner products (23) and (24) (recall (6) from the analysis of the previous section) and we can treat the former term here as an abbreviation for the latter. Moreover, the presence of the transpose of G in (26) implies that $u_{k+1}(t) - u_k(t)$ depends on values of $e_{k+l}(t)$ for $t < t \leq N$. Consequently this control law cannot be implemented but, as shown below, can be converted to one that is implementable.

The first part of Theorem 2 specialized to this case shows that the input sequence minimizes the error in a least squares sense, even if G is singular or non-square. Also if G is square and nonsingular, it is guaranteed that there exists a scalar $\sigma > 0$ such that $\|Gu\|_{\mathcal{U}}^2 \geq \sigma^2 \|u\|_{\mathcal{U}}^2$ holds, where σ is the smallest singular value of G .

If G has inverse with norm $\frac{1}{\sigma}$ then

$$\|e_{k+1}\|_{\mathcal{Y}} \leq \frac{1}{1 + \sigma^2} \|e_k\|_{\mathcal{Y}}$$

This is a corollary of the second part of Theorem 2 for this case and establishes that an exponential convergence rate is possible for 'regular' plants. Note also that σ , and hence the rate of convergence of $\|u_{k+1} - u_{\infty}\|_{\mathcal{U}}$ and $\|e_k\|_{\mathcal{Y}}$, can be changed arbitrarily by varying the weighting matrices Q and R in the cost function. This follows from

$$u^T G^T Q u \geq \sigma^2 u^T R u, \quad \forall u \in \mathcal{U}$$

and then if $R = pR_0$, where R_0 is fixed and the scalar p is a variable parameter, $\sigma = \frac{\sigma_0}{p}$ where σ_0 is the smallest singular value of R_0 . The parameter p thus provides

complete control over the convergence rate: the smaller p is, the faster the convergence rate of the input. For example, to obtain a guaranteed reduction of the error of about $\frac{1}{2}$ on each trial p should be chosen to be of the order of magnitude of σ_0 .

Simulation results suggest, however, that this may be overcautious as the smallest singular value stems from a worst-case consideration and the convergence for typical reference signals is (at least initially) much faster than is guaranteed by the bound here. This control over the convergence rate is one of the biggest advantages of this algorithm over alternatives where there is typically neither an exponential rate of convergence nor any possibility of increasing the speed of convergence.

To obtain an implementable form of this algorithm, first note that the adjoint (or transpose) for the class of plants considered here involves the operations of time reversal plus an appropriate change of the state-space parameters. Hence in

$$u_{k+1} - u_k = G^* e_{k+1} = R^{-1} G^T Q e_{k+1} \quad (27)$$

the adjoint operator G^* becomes the well known costate system [7]

$$\xi_{k+1}(t) = A^T \xi_{k+1}(t+1) + C^T Q e_{k+1}(t+1),$$

$$\xi_{k+1}(N_s) = 0$$

$$u_{k+1}(t) = u_k(t) + R^{-1} B^T \xi_{k+1}(t), \quad N_s > t \geq 0 \quad (28)$$

This system has a terminal condition at $t = N_s$ instead of an initial condition, marking it (as expected) as an anti-causal representation of the solution. It cannot therefore be implemented in this form, but a causal implementation can be found when assuming full state knowledge. The optimal control is transformed by writing for the costate

$$\xi_{k+1}(t) = [-K(t)(I + BR^{-1}B^TK(t))^{-1}A(x_{k+1}(t) - x_k(t))] + \zeta_{k+1}(t) \quad (29)$$

and then using these last two equations and standard techniques in optimal control theory [7, 2] to show that the matrix gain $K(t)$ is the solution of the familiar discrete matrix Riccati equation on the interval $[0, N_s - 1]$

$$K(t) = A^TK(t+1)A + C^TQC - [A^TK(t+1)B \times (B^TK(t+1)B + R)^{-1}B^TK(t+1)A] \quad (30)$$

with terminal condition $K(N_s) = 0$. Also the predictive or feedforward term is generated by

$$\begin{aligned} \xi_{k+1}(t) &= (I + K(t)BR^{-1}B^T)^{-1}(A^T\xi_{k+1}(t+1) \\ &\quad + C^TQe_k(t+1)) \\ \xi_{k+1}(N_s) &= 0 \end{aligned} \quad (31)$$

and the input update equation now is

$$u_{k+1}(t) = u_k(t) - \left[(B^T K(t) B + R)^{-1} \right. \\ \left. \times B^T K(t) A (x_{k+1}(t) - x_k(t)) \right] \\ + R^{-1} B^T \xi_{k+1}(t) \quad (32)$$

The norm-optimal ILC algorithm for the form of cost function used, consists of current trial full state feedback combined with feedforward from the previous trial output tracking error data. This representation of the solution is causal because (31) and (32) can be solved offline, between trials, by reverse time simulation using available previous trial data. For a time-invariant system, as here, the matrix $K(t)$ for $0 \leq t < N_s$, needs to be computed only once before the sequence of trials begins. If all entries in the plant state vector are not directly measurable then an observer based on the plant state-space model matrices can be used. The state vector entries on each trial must be stored until the next trial has been completed.

The main disadvantage limiting the practical application of the norm-optimal ILC algorithm is the large amount of computation which must be performed between each sample interval. To remedy this problem, a faster version of the algorithm can be used, which allows the majority of calculations to be performed during the design and commissioning of the control law. The remaining calculations are significantly reduced in number and consist solely of multiplications, additions, and subtractions.

Implementation of the algorithm is as follows. The matrix gain $K(t)$ defined by (30) can, as noted above, be calculated before the system operates, and hence, does not contribute to the real-time processing load. The predictive term (31) must be calculated between each trial. Note again that this equation has a terminal, as opposed to an initial, condition and must therefore be computed in descending sample order. The input update (32) must be calculated at each sample instant. It is therefore the input update equation which particularly contributes to the real-time processing load and has a significant influence on the minimum sample time that can be used in an application.

This improved implementation is derived by identifying simplifications which can be made to the computation of the original. Consider the predictive component (31); the only variables in this equation are the tracking error e_k and the predictive term itself ξ_{k+1} , and all of the other terms can be combined together to produce the matrices

$$\alpha(t) = \left(I + K(t) B R^{-1} B^T \right)^{-1}, \quad \beta(t) = \alpha(t) A^T \quad (33)$$

$$\gamma(t) = \alpha(t) C^T Q \quad (34)$$

leading to the computationally simpler predictive component equation

$$\xi_{k+1}(t) = \beta(t) \xi_{k+1}(t+1) + \gamma(t) e_k(t+1) \quad (35)$$

Exactly the same concept can be applied to the input update equation (32), resulting in the simplified input update equation:

$$u_{k+1}(t) = u_k(t) - \lambda(t) \{x_{k+1}(t) - x_k(t)\} + \omega \xi_{k+1}(t) \quad (36)$$

where

$$\omega = R^{-1} B^T, \quad \lambda(t) = (B^T K(t) B + R)^{-1} B^T K(t) A$$

The resulting implementation therefore requires seven matrices (A , B , C , β , γ , λ and ω) to be supplied to the real-time controller.

This reformulated algorithm uses significantly more memory than the original, because the memory allocation is static rather than dynamic, but norm-optimal ILC can recycle memory once calculations are complete. However it is worth observing that the process of recycling the memory takes time and decreases the amount of time available for computation of the algorithm. The approach here is preferable because it is relatively easier and cheaper to upgrade memory than to upgrade the processor.

In terms of improvement in computation speed, due to the reduced number of calculations, it is possible to calculate exactly the time required to perform each algebraic operation for both the basic algorithm and its reformulation, then find the total time for each variant. However, the results of this process still ultimately depend on the characteristics of the controller, the operating system and the efficiency of the program functions. In simulation comparisons there was a factor of three increase in available speed, when using an identical setup for both algorithms and running them at maximum simulation rate, without specifying the need for strict sample intervals. However, care must be exercised when transferring this result to experimental implementation. Although some increase in speed will be achieved by the reformulated algorithm, a quantifiable guarantee is beyond the scope of this article.

The computations required here are summarized in Table 1.

4. Measuring ILC performance

In norm-optimal ILC, the weighting matrices Q and R are used to adjust the balance between trial-to-trial error convergence speed and robustness, respectively. A critical task therefore is to investigate just how much the values

Table 1. Norm-optimal computation**First level (before operation):**

$$K(t) = A^T K(t+1)A + C^T Q C \\ - [A^T K(t+1)B (B^T K(t+1)B + R)^{-1} \\ B^T K(t+1)A]$$

$$\alpha(t) = (I + K(t)BR^{-1}B^T)^{-1}$$

$$\beta(t) = \alpha(t)A^T$$

$$\gamma(t) = \alpha(t)C^T Q$$

$$\omega = R^{-1}B^T$$

$$\lambda(t) = (B^T K(t)B + R)^{-1}B^T K(t)A$$

Second level (between trials):

$$\xi_{k+1}(t) = \beta(t)\xi_{k+1}(t+1) + \gamma(t)e_k(t+1)$$

Third level (between sampling instants):

$$u_{k+1}(t) = u_k(t) - \lambda(t)(x_{k+1}(t) - x_k(t)) + \omega(t)\xi_{k+1}(t)$$

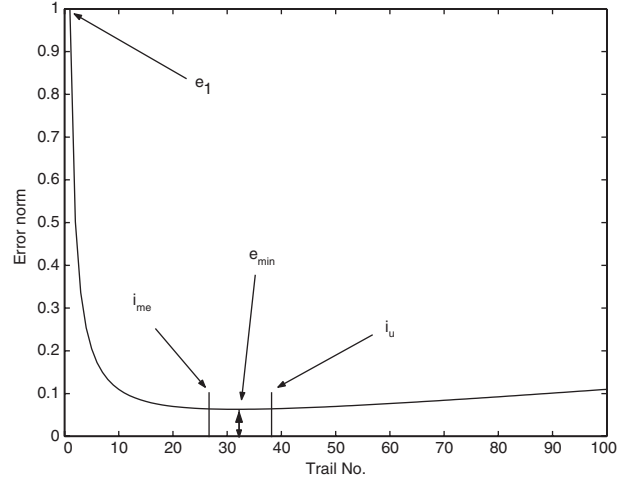
chosen affect algorithm performance. However, it is first necessary to discuss exactly what ILC performance is and how it can be measured.

It is generally recognized that there are three variables which are of particular importance when describing the performance of an ILC algorithm [45], these being

- convergence speed,
- minimum tracking error, and
- long-term performance.

Although the instantaneous data recorded during each trial (such as input voltage and output error) is useful for analyzing the learning process and its performance, it is clearly necessary to calculate some general measure of the tracking accuracy for each trial, and observe how this changes as the number of trials increases. This can specifically indicate minimum error, time to reach minimum error (convergence speed) and any sign of performance degradation. In implementation, it is possible that the error decreases from trial-to-trial and then, after a large number of trials have elapsed, start to grow again. This is sometimes termed long-term instability [32, 12] but is in fact a performance issue that is still not completely understood, and for the remainder of this article is referred to as long-term performance.

As for standard linear systems, popular measures of tracking accuracy are in terms of a suitable error norm. Fig. 1 shows the typical such plot for an ILC system with long-term performance degradation, where key parameters used to describe performance are also indicated. In particular, e_1 is the initial error norm value, i_{me} is the number of trials required to reach minimum error, e_{min} is the minimum error norm value and i_u is the number of trials until the error norm begins to increase. The typical

**Fig. 1.** Typical error norm curve for an ILC system with long-term performance degradation.

plot for a system without long-term performance degradation is similar, except that the i_u point is never reached and the error norm does not increase as the number of trials increases. Of these parameters, i_{me} and e_{min} are most commonly used to describe ILC performance.

The performance index that will be adopted involves simply integrating the area under the error norm curve for the first N trials, where N is selected appropriately for the system being considered. This results in the performance index for N trials, PI_N , given by

$$PI_N = \sum_{k=1}^N p(e)_k \quad (37)$$

where $p(e)_k$ denotes the error norm on trial k .

To allow a fair comparison of algorithm performance, test parameters such as the plant model parameters and reference trajectory must be held constant. In the case that the ILC algorithm does not involve parameter-dependent current trial feedback, the value of e_1 will be theoretically parameter-invariant, and the PI_N can be normalized by setting $e_1 = 1$. Because this does not hold for norm-optimal ILC, PI_N will instead be normalized using the error norm produced in the absence of ILC current trial feedback. Approximate upper and lower bounds on the value of PI_N are then, respectively, given by $PI_N(\max) = Ne_1$ (assuming that no improvement occurs for $k > 1$), and $PI_N(\min) = e_1$ (assuming that the error is zero for $k > 1$).

5. Application to a Gantry Robot Test Facility

The gantry robot, shown in Fig. 2, is a commercially available system found in a variety of industrial applications



Fig. 2. The gantry robot.

whose task is to place a sequence of objects onto a moving conveyor under synchronization. The sequence of operations is that the robot collects the object from a specified location, moves until it is synchronized (in terms of both position and speed) with the conveyor, places the object on the conveyor, and then returns to the same starting location to collect the next object and so on. This is sometimes referred to as ‘pick and place’ and is clearly suitable for the application of ILC.

The gantry robot can be treated as three SISO systems (one for each axis) which can operate simultaneously to locate the end effector anywhere within a cuboid work envelope. The lowest axis, X , moves in the horizontal plane, parallel to the conveyor beneath. The Y -axis is mounted on the X -axis and moves in the horizontal plane, but perpendicular to the conveyor. The Z -axis is the shorter vertical axis mounted on the Y -axis. The X and Y -axes consist of linear brushless DC motors, while the Z -axis is a linear ball-screw stage powered by a rotary brushless DC motor. All motors are energized by performance matched DC amplifiers. Axis position is measured by means of linear or rotary optical incremental encoders as appropriate.

To implement norm-optimal ILC, it is necessary to obtain a model for the plant which is to be controlled. Each axis of the gantry was modeled independently by means of sinusoidal frequency response tests. From these data it was possible to construct Bode plots for each axis and hence determine approximate transfer-functions. These were then refined, by means of a least mean squares optimization technique, to minimize the difference between the frequency response of the real plant and that of the model. The resulting X -axis Bode plot comparing the plant and the model is given in Fig. 3 (the remaining plots appear in [35]). From each Bode plot an approximate transfer-function was constructed and from that, a minimal order

state-space model. Here we only give detailed ILC design for the X -axis where a 7-order transfer-function was used in design, and the efficient implementation procedure [36] was used (with all axes) to obtain the results given below.

5.1. Test Parameters

With all axes operating simultaneously, the reference trajectories for the axes produce a three dimensional synchronizing ‘pick and place’ action, shown in Fig. 4. The trajectories generate a work rate of 30 units per minute which is equivalent to a trial time period of 2 seconds. Using a sampling frequency of 1 kHz, this generates 2000 samples per trial.

A 2-second stoppage time exists between each trial, which allows vibrations induced in the previous trial to die away and prevents their propagation between trials. Before each trial, the axes are homed to within $\pm 30 \mu\text{m}$ of a known starting location to minimize the effects of initial state error.

The plant input voltage for the first trial is zero. Therefore the algorithm must learn to track the reference in its entirety. There is no assistance from any other form of controller. In the practical implementation, the system states are estimated by means of a full-state Kalman estimator.

5.2. Experimental Results

As representative results from the experiments performed, Fig. 5 shows the error norm calculated for each axis during a 5000 trial test designed to investigate the long-term performance, where for the X and Y -axes $Q = 100$ and $R = 0.01$ and for the Z -axis $Q = 1000$ and $R = 0.1$. The mean square error (mse) has been used to enable comparison with other control methodologies that have been implemented on the gantry robot [35, 36], and the most important feature here is that there is no sign of an increasing error norm, indicating that the algorithm can achieve long-term performance compared to other algorithms implemented on the same plant which resulted in error divergence occurring after 100 or, in severe cases, just 3 trials.

To investigate the effect of varying Q and R , a batch of tests was performed using different combinations of these parameters. Table 2 displays the values of Q and R which were used to produce a total of 49 combinations. Each combination was implemented for 100 trials and the PI_{100} performance index described in Section 4 was calculated. Given that there are two tuning parameters, it is particularly suitable to plot the algorithm

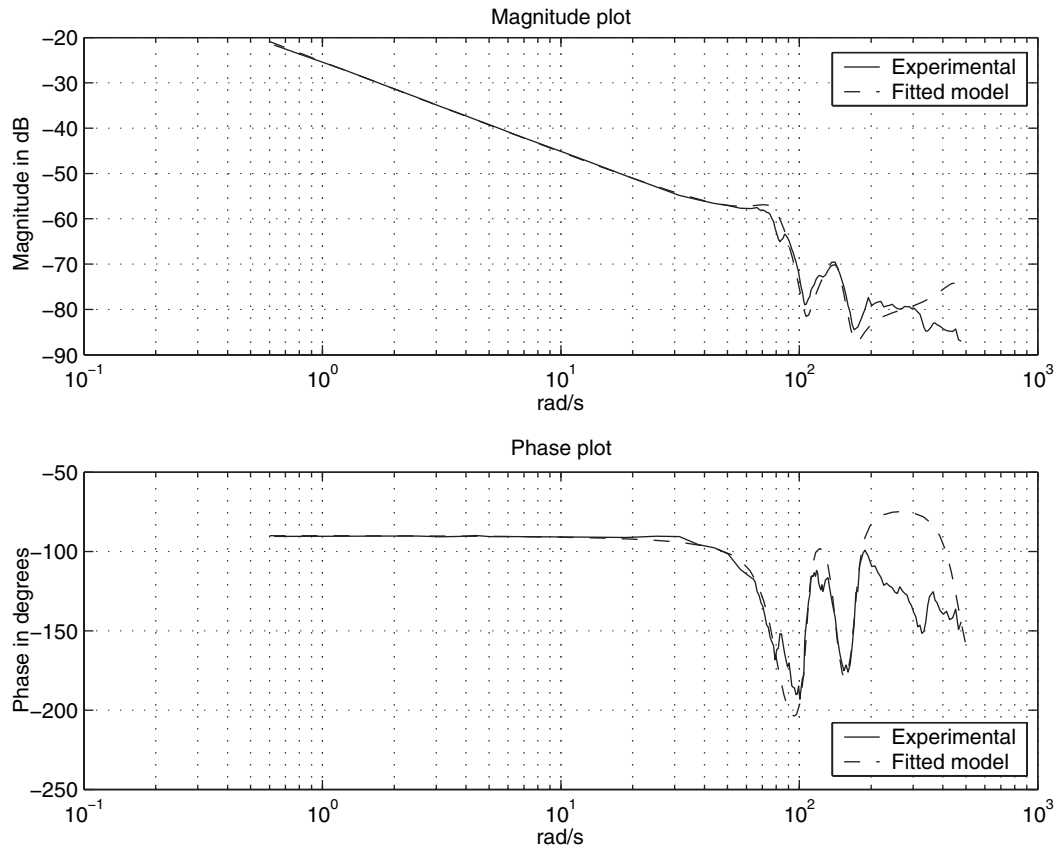


Fig. 3. Bode gain (top) and phase (bottom) plots for the X-axis.

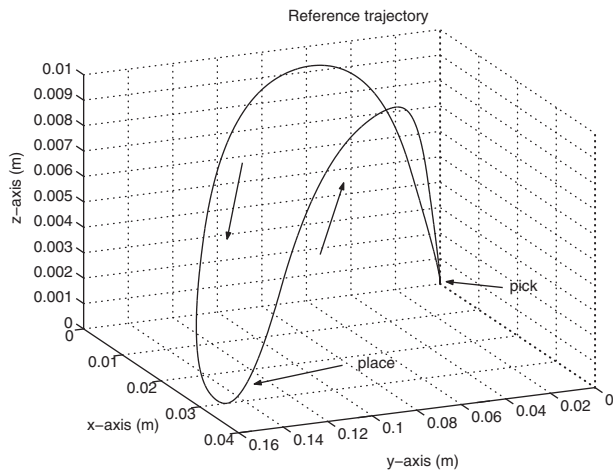


Fig. 4. Three dimensional reference trajectory.

performance on a three dimensional surface chart, as shown in Fig. 6 for the X-axis. The performance plots corresponding to the other two axes are very similar, particularly the Y-axis where the low frequency gain of the linear motor is practically identical to that of the X-axis.

Noting that Q affects the rate of error reduction and R limits the input change, interpreting the plots becomes a simple task. To the right of the chart is a region of poor tracking performance where the PI_{100} value is approximately 100 indicating that virtually nothing is learnt during the test. As could be expected, this corresponds to a small value for Q and a large value for R . With these settings, the algorithm is far too conservative. As the ratio of Q to R increases, gradually PI_{100} reduces, indicating that the performance is improving. This is represented by the slope to the right side of the chart. As the Q/R ratio continues to increase, PI_{100} is reduced to values very close to 1, indicating that the perfect trajectory is learnt in almost one trial. The balance of error reduction to input change is now approaching optimality. Temporarily increasing Q/R has little effect on the performance, until the system becomes unstable and PI_{100} jumps back to 100. This is represented by the channel and then the steep slope to the left of the chart. It is important to note that the ratio of Q to R is what determines the algorithm performance rather than the absolute values of each parameter.

The results that have been presented form part of an extensive programme of experiments [36], which have

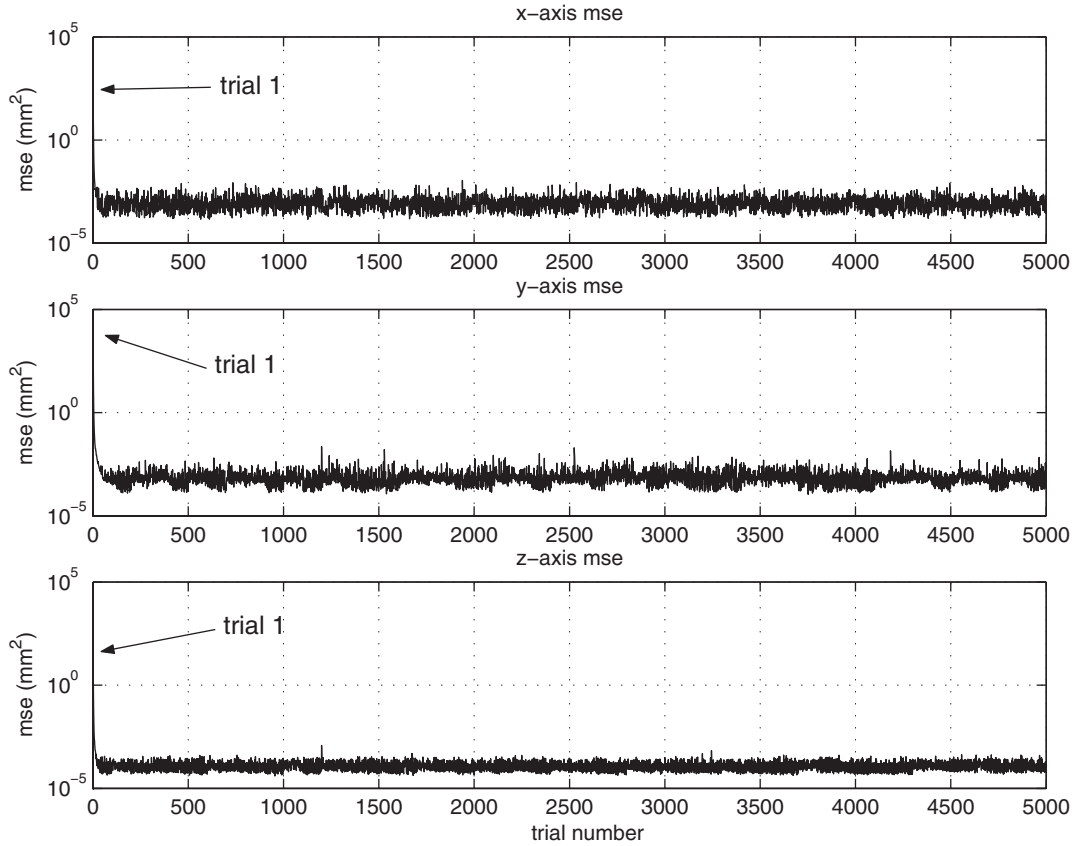


Fig. 5. Plot of the error norm over 5000 trials.

Table 2. Q and R values used in experiments

| Q | R |
|--------|--------|
| 0.1 | 0.0001 |
| 1 | 0.001 |
| 10 | 0.01 |
| 100 | 0.1 |
| 1000 | 1 |
| 10000 | 10 |
| 100000 | 100 |

confirmed the excellent performance of norm-optimal ILC in terms of convergence speed, minimum error, and long-term performance. The approach has proven well suited to the gantry robot due to the accuracy of the linear model of each axis, and the lack of interaction between axes. In the next section the algorithm is applied to an actual physical system associated with FELs. Then in the following section will be applied to control the movement of human subjects, this being a challenging application in which there exists strong nonlinear behavior, and significant difficulty in obtaining an accurate system model.

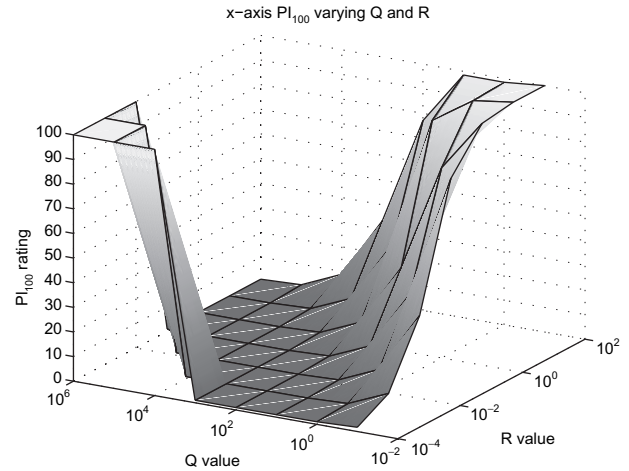


Fig. 6. X -axis PI_{100} for various Q and R .

6. Application to FELs

The results in this section are in the main from [28].

FELs use linear particle accelerators that increase the energy of the electrons by interaction with electromagnetic radio frequency (RF) fields [8]. They are operated in

pulsed mode, for example, every second there is a pulse for approximately 1 ms. This pulsed system has the following properties:

- the disturbances and uncertainties only show small changes from pulse to pulse,
- between pulses, of the order of several hundred milliseconds, could be used for computing optimal parameters and driving signals for the next pulse,
- the Field Programme Gate Array (FPGA) structure of the digital intrapulse controller allows arbitrary input signals at a frequency of 1 MHz,
- appropriate models could be identified by standard methods from measurement data.

Clearly the second of these properties, in particular, makes this an application for which ILC appears to be well suited. In this section, we describe the application of norm-optimal ILC using the same implementation as for the gantry robot and begin with some background leading on to the construction of a model for design.

6.1. Basic System Structure

In numerous research areas, a light source that is able to resolve objects on an atomic level is required, for

example, in molecular biology. Also laser light is used for a variety of experiments because it can be better focused compared to other light sources, is monochromatic, and very short pulses can be produced.

A FEL produces laser radiation with tunable wavelength. Here we focus on a research program that is aiming to build an FEL that can operate in the X-ray wavelength by the year 2014 [8]. The process uses a linear particle accelerator, which increases the energy of electrons by interaction with electromagnetic RF fields to a desired value. These fields are required to be very precise in amplitude and phase stability. Fig. 7 shows the structure of the Vacuum Ultraviolet FEL (FLASH) which is already working in the same establishment where this research program is based. The linear accelerator consists of resonators for the RF fields housed in shape cryomodules. The RF fields inside these superconducting resonator cavities are periodically supplied by an actuator system for a finite time interval and then turned off again.

In Fig. 8 the amplitude of the desired envelope of the RF field is shown for one RF pulse as a function of time. The field inside the accelerator cavities has to be kept constant once the required amplitude for the appropriate energy gain of the electrons has been reached at the end of the filling phase. During the flattop phase the electron beam is injected into the accelerator. When the electron beam has

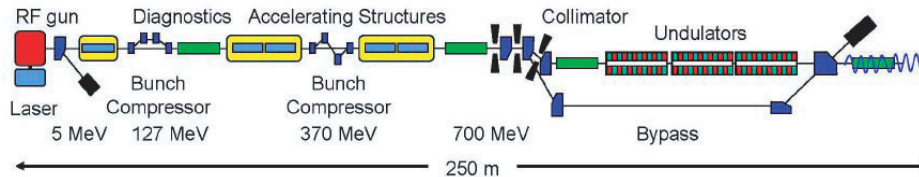


Fig. 7. An FEL with front end, accelerating structures and undulators.

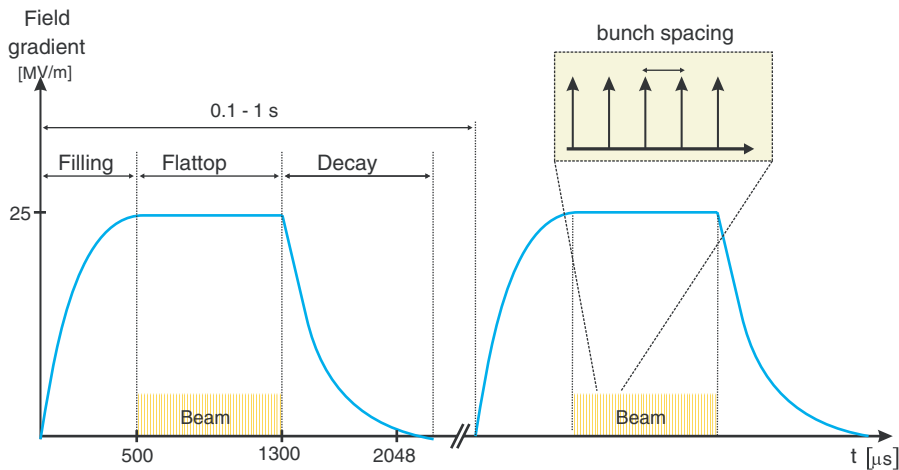


Fig. 8. One RF-pulse in superconducting cavities.

passed, the RF field is turned off and the field amplitude decays. The envelope of the RF field oscillation must be kept constant in amplitude and phase during the flattop time interval to transfer a precise amount of energy to the electrons.

Once the system is set up to a desired operating point, the pulse trajectory remains unchanged for a large number of pulses. Therefore repetitive disturbances can be suppressed by finding an optimal feedforward control signal to minimize the deflection from the reference. The driving signal is computed using norm-optimal ILC, for which a system model must be developed.

6.2. RF System of the Linear Accelerator

The acceleration takes place in the resonators where standing RF waves (modes) provide the energy, where the resonance frequency is determined by the geometry, and for the desired acceleration mode is 1.3 GHz. If the length of the cavity changes, the resonance frequency also changes. Due to the relatively thin walls, the resonators become susceptible to mechanical vibrations called microphonics which detune the resonance frequency. The high power RF fields in the cavities lead to deformation of the cavity walls and therefore also contribute to detuning. Induced currents cause Lorentz forces acting on the metal surroundings during the pulse sequence. Measurements have shown that detuning can occur up to $\Delta f \approx 500$ Hz. Because the Lorentz force is induced every time the electric field is generated, the resulting detuning is considered deterministic and repetitive.

Another source of disturbance is the electron beam itself. During its passage through the accelerating structure, the charged particles gain energy from the RF field, which leads to fluctuations in the current amount of energy stored in the system. The following bunches of charged particles will be influenced by these fluctuations, which have to be minimized by the control system. It can be assumed that the bunch arrival time will be constant from pulse-to-pulse, thus having the properties of a repetitive disturbance. Next we describe the general architecture and related aspects of the digital control scheme used in this application.

6.3. Digital Control System

The actuator system receives a precise RF signal of 1.3 GHz from the Master Oscillator (MO). This low power sinusoidal signal can be changed by the vector modulator in amplitude and phase. The output signal of the vector modulator is amplified by a klystron, which is an RF amplifier. The amplified RF waves are transferred from the klystron to the cavities inside the cryomodules via

a waveguide transmission and distribution system. For economical reasons, one high power klystron supplies all 8–32 cavities of an RF station, thus RF fields cannot be influenced in each cavity individually and the system is therefore underactuated.

The superconduction cavity simulator and controller (SIMCON) is based on FPGA structures that enables the implementation of fast algorithms. A block diagram representation of a Low Level RF (LLRF) control system is shown in Fig. 9, where the bottom part shows the digital FPGA controller. The LLRF control system has the task of keeping the pulsed RF fields in the superconducting cavities of the RF station at the reference value during the flattop phase of one RF pulse shown in Fig. 8.

After measuring the actual RF-field by pickup antennas, the signals are downconverted to an intermediate frequency of 250 kHz. The real (I) and imaginary (Q) field components are digitalized in analog-digital-converters (ADC) with a sampling frequency of 1 MHz. An overview of the signals shown in Fig. 9 in terms of I and Q is given next.

- The input signals u_I, u_Q are produced by actuator system and act directly on the vector modulator.
- The output signals y_I, y_Q are the real and imaginary parts respectively of the sum of the RF-field voltage vectors of eight cavities.
- The reference signals r_I, r_Q are the real and imaginary parts respectively of the vector sum of the RF-field's voltage vectors given by look-up tables for the specified field gradient.
- The feedforward signals f_I, f_Q are among the control signals determined by open-loop control.
- The control signals $u_{c,I}, u_{c,Q}$ are the ILC controller output signals, updating the previous, trial, or iteration, input signals.
- The control error signals e_I, e_Q are the deviations in real and imaginary parts respectively of the output signals from their reference signals.

Calibration of the measurement signals is done for compensation of effects resulting, for example, from different cable lengths. The control algorithm usually uses the vector sums of all calibrated measurement signals of the individual cavities as signals to be controlled, because of the lack of individual action for each cavity.

To reach the desired setpoint values even in open-loop adaptation of the feedforward signals must be sufficient to track the reference trajectory. The disturbance sources mainly arise from the strong effect of the Lorentz forces which are deterministic from pulse-to-pulse. It is possible to compensate the drift away from the operating point by a smooth increase of the driving signal over the flattop. Transients induced by the beam are predictable, and the arrival time is known. An increase in the driving power

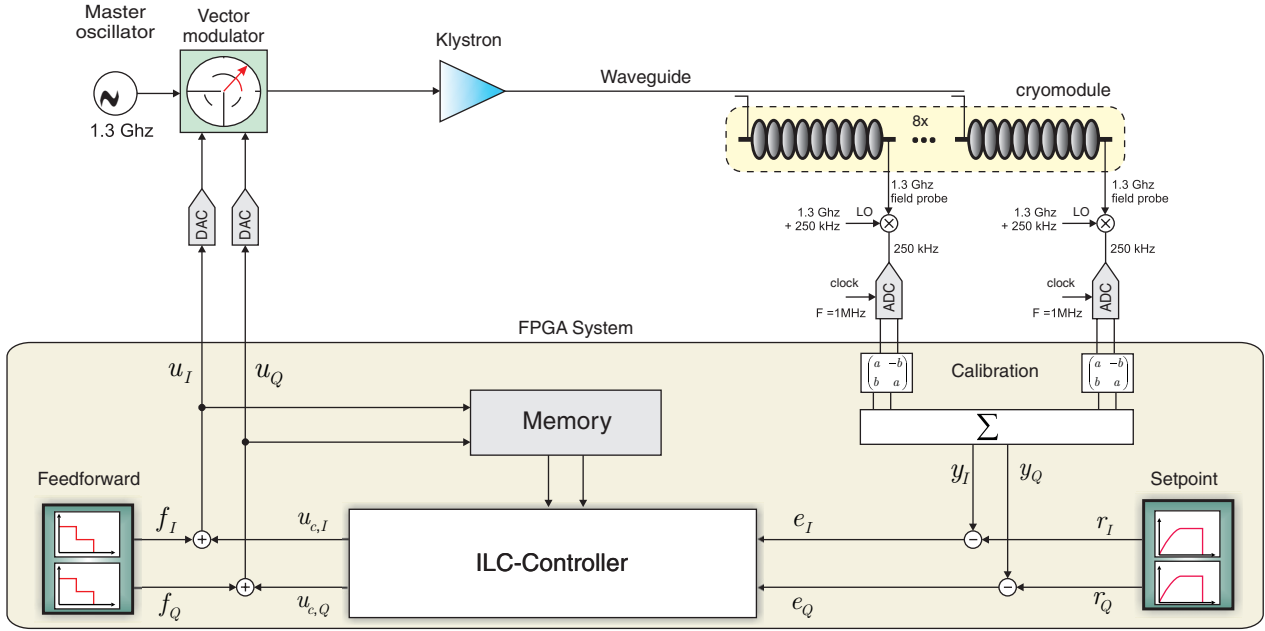


Fig. 9. Structure of the RF system with master oscillator, vector modulator, klystron, cryomodule, measurement, and calibration system and the FPGA implemented control system.

will keep this fluctuation low. Both compensation measures have a positive effect on control scheme performance while keeping the control error signals small. To predict the system behavior, it is essential to have an adequate model of the underlying system dynamics. Next we give an outline of the identification procedure used for model building.

6.4. System Modeling

Although additional external disturbances and a number of nonlinearities in the actuator system are known to be relevant for a broad range of operation setpoints, standard identification procedures for linear time invariant models can be used to estimate models that can be validated at specific setpoints in the manner outlined in, for example, [31]. The subspace identification method *N4SID* from Matlab's System Identification Toolbox was used to estimate the matrices A, B, C, D of the state-space model

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\quad (38)$$

where $u = (u_I \ u_Q)^T$ and $y = (y_I \ y_Q)^T$ denote the system input and output vectors respectively and x the state vector. The flattop phase of the pulse is the main interest for control and at the same time marks the systems operation point. Only measurements from this period are used for system identification. Persistent excitation signals can be

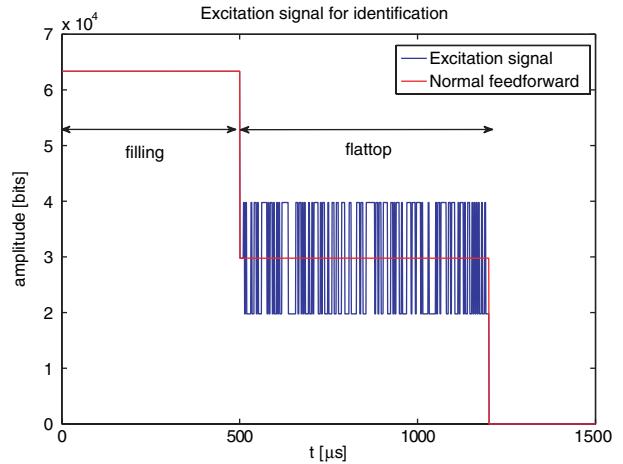


Fig. 10. Input disturbances on both channels at flattop.

injected into the accelerator system by superimposing random signals on standard feedforward tables with defined setpoints. A typical input sequence for the feedforward table is shown in Fig. 10.

In the first 500 μs (filling phase), the actuator system is operated at maximum power. The flattop phase starts from 500 μs , and during this phase the inputs are first reduced by a factor of 0.5 to reach the setpoint and soon after the excitation signal is added to both inputs, see Fig. 10. A high amplitude leads to a good signal to noise ratio. In Fig. 11 measured versus simulated signals are shown for an identified 4th order model.

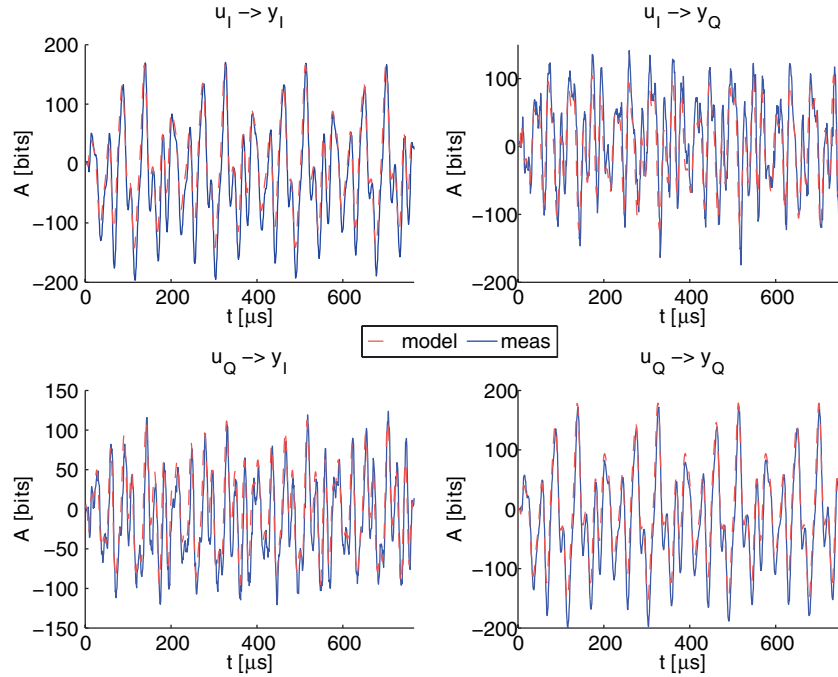


Fig. 11. Measured versus simulated vector sum signals (4rd order model).

Table 3. Available and required computation time for different levels

| | 1st level (before operation) | 2nd level (between trials) | 3rd level (between samples) |
|--------------------|------------------------------------|----------------------------------|-----------------------------------|
| Needed time [s] | 24.71 | 0.0147 | 0.0047 |
| Available time [s] | ∞ | 0.1 | 0.000001 |

6.5. Application of Norm-Optimal ILC

The system to be controlled is driven in pulsed mode. Moreover, the accelerating process is considered to be repetitive and the disturbances also show this behavior. Hence ILC is suitable for this application where $e_k = [e_I \ e_Q]^T$ and here the norm-optimal approach with cost function of the form (22) is employed. Fast and efficient computation is important when computing the updated input signal. The time interval between two consecutive RF pulses is approximately 0.1 s. Furthermore, a state feedback control input has to be computed in each sampling period of 1 μ s. Using a workstation computer with an *Intel Pentium 4* processor with a clock speed of 3 GHz, the computation time needed for the three different time levels is given in Table 3.

The time needed between samples exceeds the time that is available. Consequently, the third level computations are executed in the second level for the results below. The input signals are then computed for the whole trial at once, instead of for every sample separately. To prevent damage

to system components when the algorithm is implemented on the real plant, the input signals are limited as shown in Fig. 12. The limits are set to the maximum and minimum values of the input signals during the filling and the decay phase.

The following weighting matrices in the norm-optimal ILC cost function gave results which could be implemented on the real plant

$$Q = 100 \times I_{2 \times 2} \text{ and } R = I_{2 \times 2} \quad (39)$$

The state variables required for the state feedback component in the algorithm are obtained by simulation using the plant model. Including input and output disturbances in the simulation, results for the flattop phase are shown in Fig. 13, where only the signals of the first input and output are given. The shape of the disturbances can be computed as the deviation from the smooth trajectory of the first trial. As the number of trials increases, the output signals approach the desired setpoint (SP) trajectory. Rejection of the input and output disturbances can also be observed. Since the input signal reaches the given limits in the beginning of the phase, the output signal only approaches the setpoint slowly in the first 100 μ s.

6.6. Experimental Results

The norm-optimal ILC scheme has been successfully implemented on the real plant at the DESY test facility,

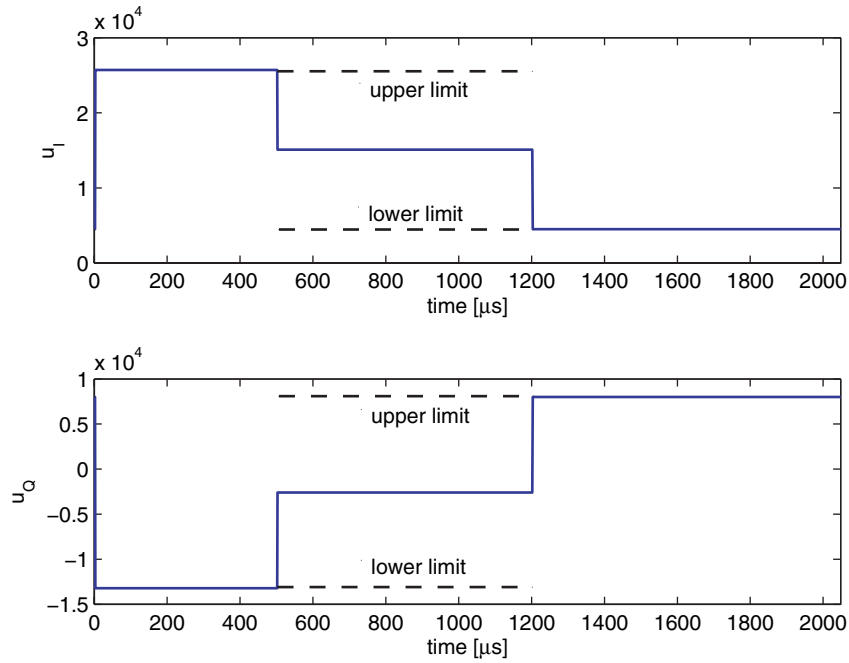


Fig. 12. Limits on input signals.

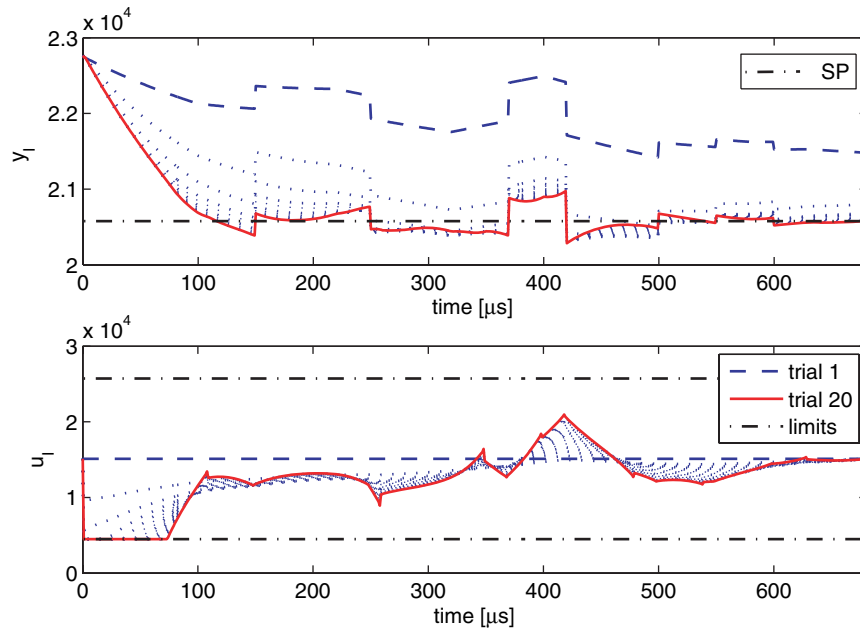


Fig. 13. Output and input signals after 20 trials.

with measurements shown in Figs. 14 and 15 for the cost function weighting matrices given in (39). The number of trials is initially set to 10.

Fig. 14 shows an increasing and decreasing trend of the first and the second output signal during the flattop phase, respectively. This is the general behavior caused by the detuning effects described previously. However,

on increasing the number of trials, both output signals approach the desired setpoint. After 10 trials, the output signals show only small deviations from the reference trajectory. Because only the signals during the flattop phase are controlled, the input signals of the filling and the decay phase are kept constant as illustrated in Fig. 15.

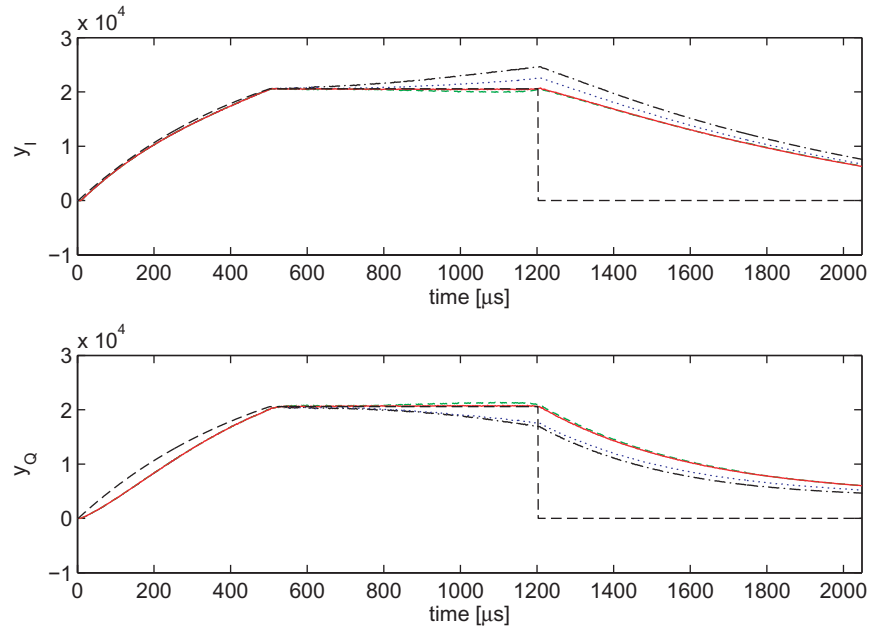


Fig. 14. Measured output signals with norm-optimal ILC applied.

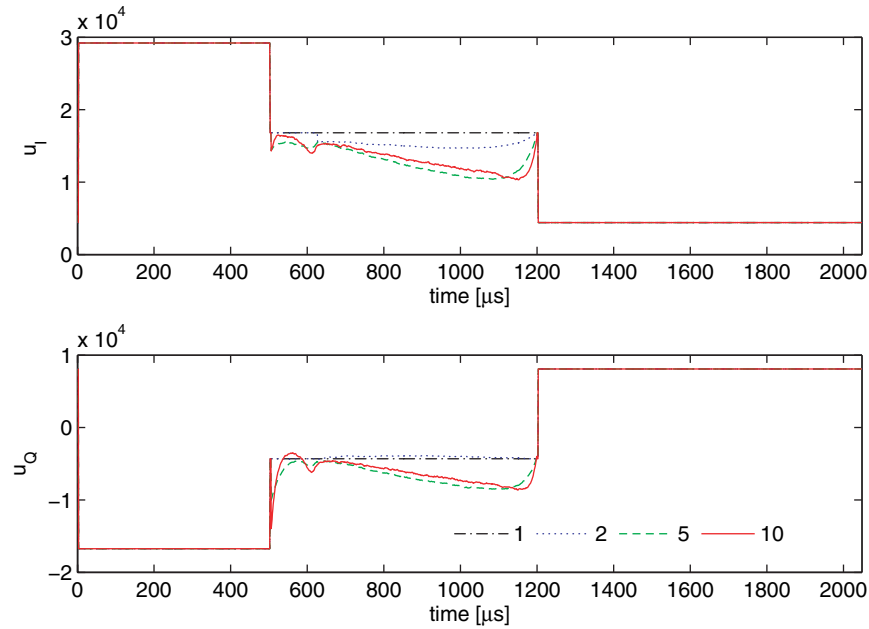


Fig. 15. Input signals from the norm-optimal ILC.

In order to emphasize the improvements in the field regulation gained by application of norm-optimal ILC, Fig. 16 gives a comparison to the decentralized proportional controller previously used for field regulation and a MIMO feedback controller.

Two major issues with the proportional controller arise from the results. The first is the high overshoot at the beginning of the flattop, resulting from the step given

in the feedforward tables. Second, the trend caused by the detuning can be seen and is especially visible for the phase. The MIMO feedback controller improves both effects but by itself is not able to fully compensate for them. Applying the norm-optimal ILC significantly reduces the control error to a level which is required to give an appropriate beam energy gain. The maximum allowed rms error is visualized in Fig. 16 by the black

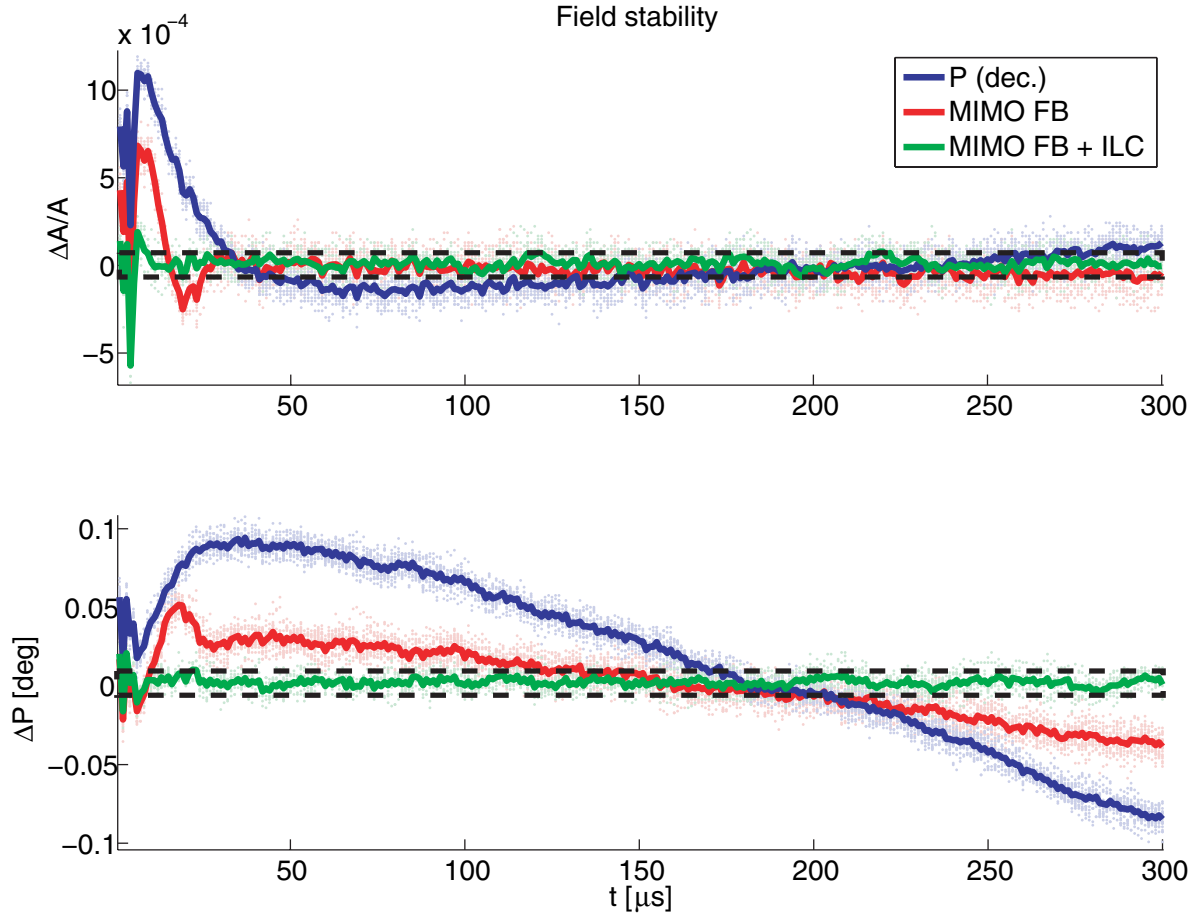


Fig. 16. Performance comparison between proportional feedback, MIMO feedback only, and the combination with norm-optimal ILC.

dashed lines. Next, the influence on the electron beam is discussed.

It is required to have a very stable field gradient in amplitude and phase during the flattop region, to guarantee a consistent acceleration energy for the electron beam passing the module. Due to the energy consumption of the beam, this can be treated as a disturbance. If a series of electron bunches, known as a bunch train, with a high repetition rate passes through the accelerator during one pulse, a beam induced transient can be observed, because the feedback controller is unable to provide fast enough compensation for this effect. Consequently an undesirable energy spread over the whole bunch train will arise.

In Fig. 17 the adaption of the RF-field amplitude during the flattop is shown. The solid lines give the mean values of the relative field amplitude variations $\Delta A/A$ for different numbers of trials. The dark dots show individual measurements from the last trial considered here, i.e. number 50. The measurements were undertaken in closed-loop operation with a decentralized proportional controller and a gain of 40. The strong field deviation at the beginning is caused by injection of a new electron bunch train after

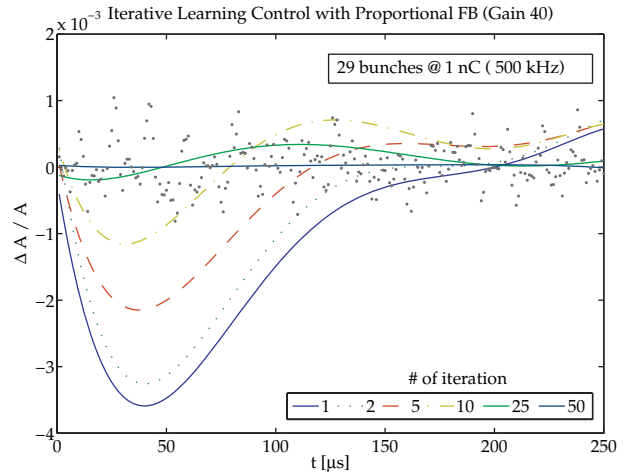


Fig. 17. Compensation for beam induced field deviations using norm-optimal ILC.

some time without beam as described above. It can be seen that within 50 trials the field amplitude is stabilized around the mean value, which means the field flatness is recovered.

Nevertheless, stochastic fluctuations cannot be fully compensated, which can be observed in Fig. 17 where the actual measurement points are given as dotted values. This is to be expected since the controller has been designed in a deterministic setting.

In summary, the experimental results show performance improvements are possible with norm-optimal ILC applied, even though only tests with a small number of trials could be executed. It has not yet been possible to investigate the performance of the controller when an electron beam is injected into the system, but experiments including the beam loading are planned. For further improvement of the performance, tests with an increased number of trials will be performed to confirm long-term performance. Moreover, implementing the control algorithm in FPGA will decrease the computational time needed so that input signals can be computed in every sampling period. A combination of the ILC with the existing proportional controller or the enhanced MIMO linear time-invariant controller, which is currently being developed [41], is expected to further increase the control performance.

7. Application to Stroke Rehabilitation

There are 300,000 people in the UK living with moderate to severe disabilities as a result of a stroke [37], of which 85% had an initial deficiency in the upper limb [33] and less than 50% have recovered useful upper limb function [33, 10]. These demographics reflect those across the European Union and, due to an aging population and better acute care, prevalence of stroke is likely to increase. Electrical Stimulation (ES) has been shown to improve motor control in a growing number of clinical applications [23, 24], and the approach is supported by neurophysiological [11] and motor learning research [40]. There is also clear evidence that recovery is enhanced when stimulation is applied coincidentally with a patient's own voluntary intention while performing a task [39].

The need to accurately apply ES to achieve a movement has motivated significant interest in the development and application of techniques that can provide a high level of precision. A range of model-based control schemes have been proposed, including optimal [25], H_∞ [26], and fuzzy [15] control of standing, sliding mode control of shank movement [27], and data-driven control [34] of the knee joint. Artificial neural networks are a popular approach, and have been applied to both the upper [43] and lower limbs [21] of paretic subjects.

However, these control laws have not transferred to clinical practice, a setting in which there is very limited time available for tuning parameters, and in which a high level of performance is required over a wide range of

participants and tasks. Only open-loop methods, or those triggered through voluntary activity (such as the myoelectric activity of the same muscle [44]) have been used in patient trials, but cannot provide the precise control over the applied stimulation necessary to fully exploit the association between the intended movement of the subject and the application of stimulation that allows them to achieve it. In order to maximize the potential of the treatment, it is necessary to apply simulation that results in accurate tracking in a small number of trials, but also exerts close control over the level of stimulation and error attained in order to promote the necessary sustained voluntary effort by the subject.

In an ongoing research programme [17, 18], a workstation has been designed and constructed in order to provide the controlled environment needed to assist patients in performing reaching tasks. ILC has enabled a high level of tracking performance to be achieved in this demanding setting, and recent clinical trials of 8 week duration have been conducted with five stroke participants. Following these, results have indicated statistically significant improvement in several areas, including patients' level of unassisted tracking and their ability to exert isometric force [22].

7.1. System Description

The task presented to the seated patient is to track reaching trajectories using their impaired arm. So that the objective is presented with maximum clarity, only trajectories in a fixed horizontal plane are used and the patient's hand is constrained to move in this plane by a custom-built robot. A data projector mounted above the subject shines an image of the trajectory path, as well as a moving spot to indicate the reference point, onto a target mounted above the subject's hand. In response to clinical need, ES is applied to the triceps muscle to provide forearm extension. A patient using the workstation is shown in Fig. 18. The stimulation comprises a sequence of bi-phasic pulses at 40 Hz with pulsewidth, u , in the range 0–350 μ s (resolution 1 μ s). The subject's arm is strapped to a five-bar robotic arm which is actuated using two dc brushless servomotors, and a force/torque sensor, situated between the penultimate and final links, measures the force applied by the subject.

A model of the combined human arm and robotic manipulator system is shown in Fig. 19, in which the fifth robotic link is strapped to the human arm. The robotic arm has a base co-ordinate system with components x_0, y_0 , and z_0 . Likewise the human arm has a base co-ordinate system with components x_0', y_0' , and z_0' .

The stimulated human arm model can be accurately represented by the expression

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(q, \dot{q}) = \tau + J^T(q)h \quad (40)$$

where

$$\mathbf{q} = \begin{bmatrix} \vartheta_u \\ \vartheta_f \end{bmatrix}^T, \mathbf{B}(\cdot) = \begin{bmatrix} b_1 & b_2 \\ b_2 & b_3 \end{bmatrix}, \mathbf{C}(\cdot) = \begin{bmatrix} -2c_1\dot{\vartheta}_f - c_1\dot{\vartheta}_f \\ c_1\dot{\vartheta}_u & c_2\dot{\vartheta}_f \end{bmatrix}$$

$$\mathbf{F}(\cdot) = \begin{bmatrix} F_u(\vartheta_u, \dot{\vartheta}_u) \\ F_f(\vartheta_f, \dot{\vartheta}_f) \end{bmatrix}, \boldsymbol{\tau} = \begin{bmatrix} 0 \\ -T\sigma(\vartheta_f) \end{bmatrix} \quad (41)$$

and $\mathbf{J}(\cdot)$ is the Jacobian [17]. Here $\boldsymbol{\tau}$ is the moment produced by the stimulated muscle, with $\sigma(\vartheta_f) = s_f c_\gamma / \sqrt{1 - c_f^2 c_\gamma^2}$ representing the $z_{0'}$ component of a unit vector parallel to the elbow axis. The variables ϑ_u and ϑ_f denote the upper arm angle and forearm angles respectively. The robotic arm has link lengths $l_1 = 0.45$ m,

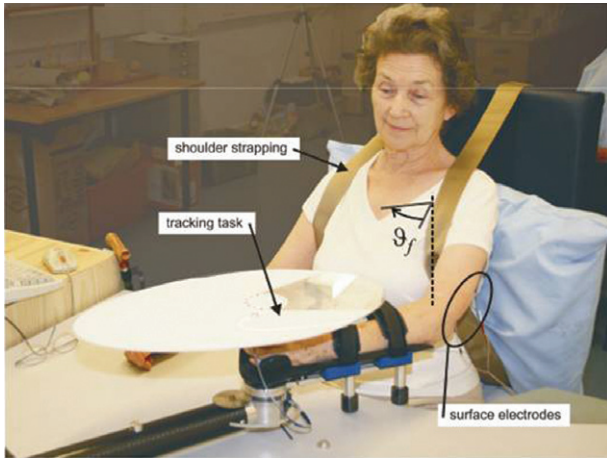


Fig. 18. Stroke participant using robotic workstation to perform a tracking task.

$l_2 = 0.2$ m and $l_3 = 0.66$ m, and the subject interacts with the robot by applying a force with components F_x and F_y , in the directions of \mathbf{x}_0 and \mathbf{y}_0 , respectively, at the point P which has a z_0 component of l_z in this system. A form of impedance control is used to govern the torque demand supplied to the motors and ensure safe interaction with the patient [14]. This produces the relationship

$$-\mathbf{h} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = \mathbf{K}_{K_x} \tilde{\mathbf{x}} + \mathbf{K}_{B_x} \dot{\tilde{\mathbf{x}}} + \mathbf{K}_{M_x} \ddot{\tilde{\mathbf{x}}} \quad (42)$$

at P , where $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$, $\hat{\mathbf{x}}$ is the robot reference, $\mathbf{x} = \mathbf{k}(\mathbf{q})$ is the direct kinematics equation for the human arm, and \mathbf{K}_{K_x} , \mathbf{K}_{B_x} and \mathbf{K}_{M_x} are gain matrices.

Control laws using ES for actuation of the forearm have been tested using unimpaired subjects who apply no voluntary effort, hence providing no means for control over upper arm movement. Rather than fixing the upper arm and restricting the reference trajectories used, the robotic arm was instead used to provide the necessary control over the upper arm angle. This was achieved through selection of variables in (42) to result in the relationship

$$\mathbf{h} = \mathbf{J}^{-T}(\mathbf{q}) \left(\underbrace{\begin{bmatrix} K_{K_1} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{K}_{K_x}} \tilde{\mathbf{q}} + \underbrace{\begin{bmatrix} K_{B_1} & 0 \\ 0 & K_{B_2} \end{bmatrix}}_{\mathbf{K}_{B_x}} \dot{\tilde{\mathbf{q}}} + \underbrace{\begin{bmatrix} K_{M_1} & 0 \\ 0 & K_{M_2} \end{bmatrix}}_{\mathbf{K}_{M_x}} \ddot{\tilde{\mathbf{q}}} \right) \quad (43)$$

Here $\hat{\mathbf{x}}$, K_{K_1} , K_{B_1} , and K_{M_1} are used to impose a moment about the upper arm to keep the target on the trajectory

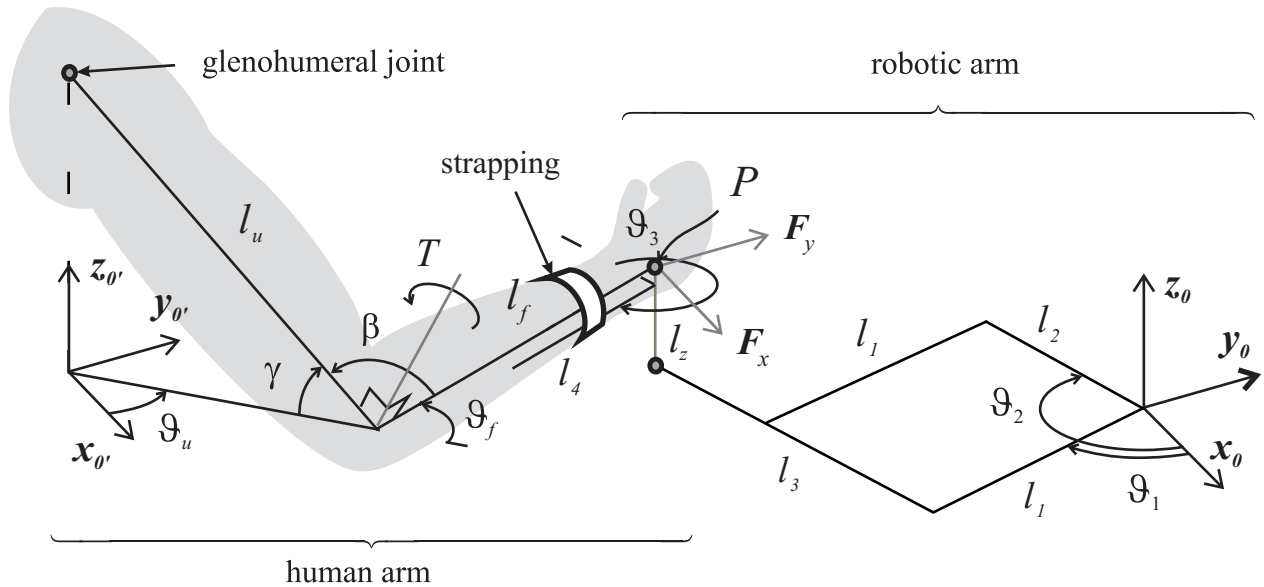


Fig. 19. Human and robotic arm systems.

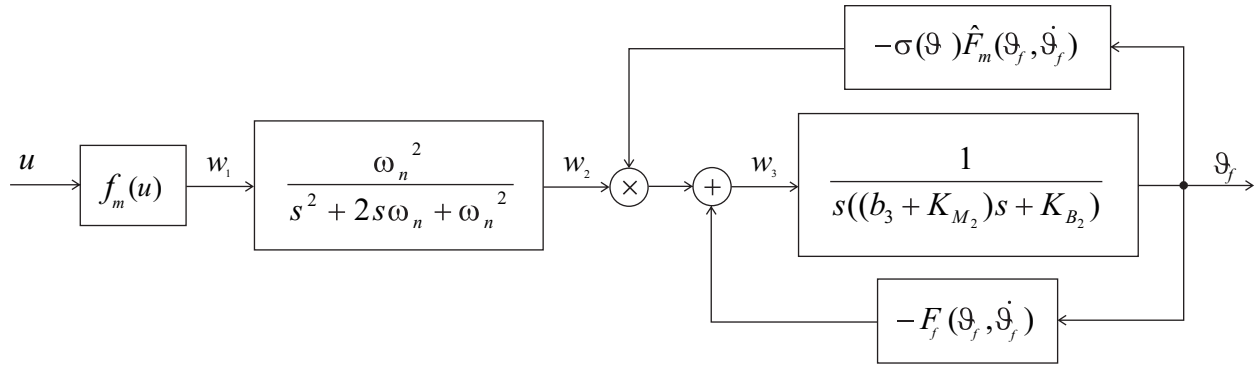


Fig. 20. Stimulated arm model.

path, while dynamics about the stimulated forearm are specified by K_{B_2} and K_{M_2} . If the reference path is then chosen to provide a smooth upper arm component variation with respect to that of the forearm, the joint coupling terms in (40) may be neglected and the forearm dynamics assumed independent of the upper arm. The resulting forearm dynamics are given by

$$b_3 \ddot{\vartheta}_f + F_f(\vartheta_f, \dot{\vartheta}_f) = -T\sigma(\vartheta_f) + K_{B_2} \dot{\vartheta}_f + K_{M_2} \ddot{\vartheta}_f \quad (44)$$

and a complete derivation appears in [19]. The moment, T , developed by the stimulated triceps muscle may be accurately modeled using

$$T(\beta, \dot{\beta}, u, t) = g_m(u, t) \times F_m(\beta, \dot{\beta}) \quad (45)$$

where $g_m(u, t)$ is a Hammerstein structure incorporating a static nonlinearity, $f_m(u)$, representing the isometric recruitment curve, cascaded with a critically damped second order system [42]. This is one of the few reported models that may be identified using excitation inputs suitable for application to stroke patients, although recent work has begun to develop alternative models for this application [29]. Since the elbow angle is given by $\beta(\vartheta_f) = \arccos(-c_f c_\gamma)$, the function $F_m(\beta, \dot{\beta})$ may be written as $\hat{F}_m(\vartheta_f, \dot{\vartheta}_f)$. Combining (44) and (45) results in the electrically stimulated arm system shown in Fig. 20. To identify the necessary parameters, the subject's arm is first moved about the workspace to yield the geometric relationships appearing in Fig. 19 via nonlinear optimization. The arm is then held stationary while ES is applied in order to identify parameters in the Hammerstein model, $g_m(u, t)$, using deconvolution and a nonlinear optimization procedure. Stimulation sequences and kinematic trajectories, imposed on the arm by the robot, are then applied, and an Least Mean Squares (LMS) optimization is used to provide all those parameters appearing in (40). Full details of the identification for both stroke and unimpaired subjects can be found in [17].

7.2. ILC Implementation

In previous work, simple structure ILC laws have been used to control the human arm [19], however, norm-optimal ILC offers the possibility of superior performance and greater control over the change in input which is of prime importance in this setting. Two approaches involving norm-optimal ILC will therefore be examined and experimentally verified. In the first of these the system is linearized through the introduction of a controller to compensate for the effect of the nonlinear terms before the application of norm-optimal ILC. The output of this controller, u , is the stimulation signal applied to the triceps and it comprises (i) the inverse of the recruitment curve, $f_m^{-1}(\cdot)$, (ii) the inverse of the muscle multiplier, $1/\hat{F}_m(\cdot)$, and (iii) the inverse of the elbow joint angle effect, $1/\sigma(\cdot)$. Because these functions vary only slowly when the trajectory is tracked [17], the control action is to cancel their effect while neglecting the intervening muscle dynamics. This approach has been supported both theoretically and experimentally when used with simple structure ILC laws [19], and results in the linear system approximation

$$G(s) = \frac{\omega_n^2}{s^2 + 2s\omega_n + \omega_n^2} \cdot \frac{1}{s((b_3 + K_{M_2})s + K_{B_2})} \quad (46)$$

A feedback controller is then implemented to act as a pre-stabilizer and provide satisfactory tracking over the initial trial, as shown in Fig. 21. Implementation of norm-optimal ILC is then achieved through discretization of the closed-loop system, and the states required in practice are provided by a Kalman filter-based state estimator. Experimental results confirming the efficacy of this approach are given in Section 7.3.

The second approach does not attempt to compensate for the plant nonlinearity. Newton method-based norm-optimal ILC [30] is a nonlinear method and will be applied directly to the system shown in Fig. 20. Here the starting

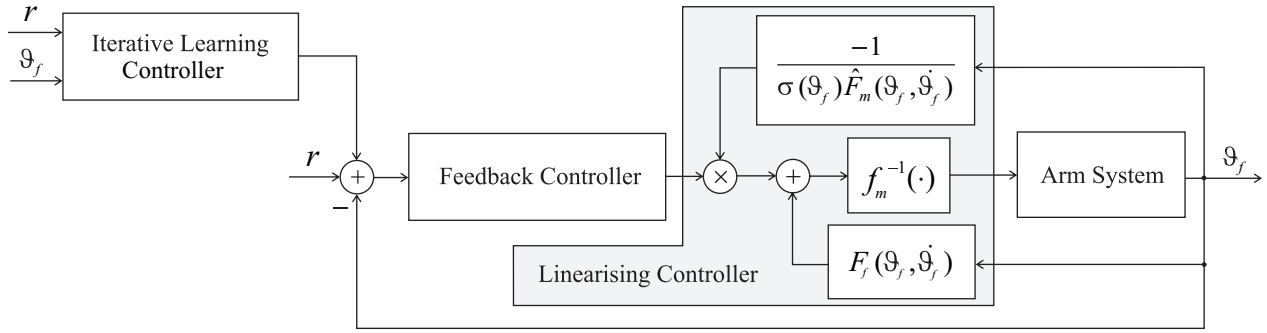


Fig. 21. Stimulated arm model with linearizing control.

point is a discrete-time system representation, and accordingly the linear activation dynamics are first represented by the state-space model $[\Phi_m, \Gamma_m, H_m]$. The relationship between w_1 and w_2 is then given by

$$\begin{aligned} \mathbf{x}_m(t+1) &= \Phi_m \mathbf{x}_m(t) + \Gamma_m w_1(t), \quad \mathbf{x}_m(0) = \mathbf{x}_{m0} \\ w_2(t) &= H_m \mathbf{x}_m(t) \end{aligned} \quad (47)$$

and similarly the arm dynamics are represented by the state-space model $[\Phi_p, \Gamma_p, H_p]$. Hence the relationship between w_3 , ϑ_f , and $\dot{\vartheta}_f$ is given by

$$\begin{aligned} \mathbf{x}_p(t+1) &= \Phi_p \mathbf{x}_p(t) + \Gamma_p w_3(t), \quad \mathbf{x}_p(0) = \mathbf{x}_{p0} \\ \begin{bmatrix} \vartheta_f(t) \\ \dot{\vartheta}_f(t) \end{bmatrix} &= \begin{bmatrix} H_{p1} \\ H_{p2} \end{bmatrix} \mathbf{x}_p(t) = H_p \mathbf{x}_p(t) \end{aligned} \quad (48)$$

The stimulated arm system on trial k is now described by

$$\begin{aligned} \mathbf{x}_k(t+1) &= \Phi \mathbf{x}_k(t) + \Gamma \begin{bmatrix} d(\mathbf{x}_k(t)) f_m(u_k(t)) \\ \end{bmatrix} \\ &= f(\mathbf{x}_k(t), u_k(t)) \end{aligned}$$

$$\begin{aligned} \vartheta_{f,k}(t) &= \bar{H}_{p1} \mathbf{x}_k(t) = h(\mathbf{x}_k(t)), \\ \mathbf{x}_k(0) &= \mathbf{x}_0, \quad t \in [0, N_s] \end{aligned} \quad (49)$$

where $\mathbf{x}(t) = [\mathbf{x}_p(t) \ \mathbf{x}_m(t)]^T$, $\mathbf{x}_0 = [\mathbf{x}_{p0} \ \mathbf{x}_{m0}]^T$, $\Phi = \text{diag}\{\Phi_p, \Phi_m\}$, $\Gamma = \text{diag}\{\Gamma_p, \Gamma_m\}$, $H_m = [\mathbf{0} \ H_m]$, $\bar{H}_{p1} = [H_{p1} \ \mathbf{0}]$ and $\bar{H}_{p2} = [H_{p2} \ \mathbf{0}]$. The integer N_s is equal to $\frac{T}{T_s} + 1$, where T_s is the sample time, and

$$\begin{aligned} d(\mathbf{x}_k) &= -\bar{H}_m \mathbf{x}_k \hat{F}_m(\bar{H}_{p1} \mathbf{x}_k, \bar{H}_{p2} \mathbf{x}_k) \sigma(\bar{H}_{p1} \mathbf{x}_k) \\ &\quad - F_f(\bar{H}_{p1} \mathbf{x}_k, \bar{H}_{p2} \mathbf{x}_k) \end{aligned} \quad (50)$$

in which the explicit time dependence of \mathbf{x}_k has been omitted. The next step is to replace (49) with a set of algebraic equations in \mathbb{R}^{N_s} by defining the shifted input and output vectors as

$$\begin{aligned} \mathbf{u}_k &= [u_k(0), u_k(1), \dots, u_k(N_s - 1)]^T \\ \vartheta_{f,k} &= [\vartheta_{f,k}(1), \vartheta_{f,k}(2), \dots, \vartheta_{f,k}(N_s)]^T \end{aligned} \quad (51)$$

and the relationship between the input and output time-series can be expressed by the algebraic functions

$$\begin{aligned} \vartheta_{f,k}(1) &= h(\mathbf{x}_k(1)) = h(f(\mathbf{x}_k(0), u_k(0))) \\ &= g_1(\mathbf{x}_k(0), u_k(0)) \\ \vartheta_{f,k}(2) &= h(\mathbf{x}_k(2)) = h(f(\mathbf{x}_k(1), u_k(1))) \\ &= g_2(\mathbf{x}_k(0), u_k(0), u_k(1)) \\ &\vdots \\ \vartheta_{f,k}(N_s) &= h(\mathbf{x}_k(N_s)) \\ &= h(f(\mathbf{x}_k(N_s - 1), u_k(N_s - 1))) \\ &= g_{N_s}(\mathbf{x}_k(0), u_k(0), u_k(1), \dots, u_k(N_s - 1)) \end{aligned} \quad (52)$$

and hence system (49) can be represented as

$$\vartheta_{f,k} = \mathbf{g}(\mathbf{u}_k), \quad \mathbf{g}(\cdot) = [g_1(\cdot), g_2(\cdot), \dots, g_{N_s}(\cdot)]^T \quad (53)$$

The ILC task of finding the input which drives the dynamic system (49) to track the desired output, equates to finding the solution that satisfies the nonlinear function (53) with $\vartheta_{f,k}$ substituted by $\vartheta_f^* = [\vartheta_f^*(1), \vartheta_f^*(2), \dots, \vartheta_f^*(N_s)]^T$. The Newton method is selected to solve this nonlinear equation, and is given in ILC notation as

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_{k+1} \mathbf{g}'(\mathbf{u}_k)^{-1} \mathbf{e}_k \quad (54)$$

where the scalar $\alpha_{k+1} \geq 0$ is a relaxation parameter, and $\mathbf{e}_k = \vartheta_f^* - \vartheta_{f,k}$. The derivative $\mathbf{g}'(\mathbf{u}_k)$ is equivalent to the linearization of (49), on trial k at $(\mathbf{u}_k, \mathbf{x}_k)$ which can be represented by the time-varying system

$$\tilde{\vartheta}_f = \mathbf{g}'(\mathbf{u}_k) \tilde{\mathbf{u}} \quad (55)$$

which equates to

$$\begin{aligned}\tilde{\mathbf{x}}(t+1) &= \mathbf{A}(t)\tilde{\mathbf{x}}(t) + \mathbf{B}(t)\tilde{\mathbf{u}}(t) \\ \tilde{\vartheta}_f(t) &= \mathbf{C}(t)\tilde{\mathbf{x}}(t)\end{aligned}\quad (56)$$

with

$$\begin{aligned}\mathbf{A}(t) &= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)} = \Phi + \Gamma \begin{bmatrix} \mathbf{p}(t) \\ \mathbf{0} \end{bmatrix} \\ \mathbf{B}(t) &= \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)} = \Gamma \begin{bmatrix} 0 \\ f'_m(\mathbf{u}_k(t)) \end{bmatrix} \\ \mathbf{C}(t) &= \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)_{\mathbf{u}_k(t), \mathbf{x}_k(t)} = \bar{\mathbf{H}}_{p1}\end{aligned}\quad (57)$$

where $\tilde{\mathbf{x}} = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\tilde{\mathbf{u}} = \mathbf{u}_{k+1} - \mathbf{u}_k$, $\tilde{\vartheta}_f = \vartheta_{f,k+1} - \vartheta_{f,k}$, $\tilde{\mathbf{x}}(0) = \mathbf{x}_{k+1}(0) - \mathbf{x}_k(0) = \mathbf{0}$ and

$$\begin{aligned}\mathbf{p}(t) &= -\bar{\mathbf{H}}_m \hat{\mathbf{F}}_m \sigma - \bar{\mathbf{H}}_m \mathbf{x} \bar{\mathbf{H}}_{p1} \hat{\mathbf{F}}'_m \sigma - \bar{\mathbf{H}}_m \mathbf{x} \bar{\mathbf{H}}_{p1} \sigma' \\ &\quad - \bar{\mathbf{H}}_{p1} \mathbf{F}'_f + \bar{\mathbf{H}}_m \mathbf{x} \bar{\mathbf{H}}_{p2} \hat{\mathbf{F}}^*_m \sigma - \bar{\mathbf{H}}_{p2} \mathbf{F}^*_f\end{aligned}\quad (58)$$

Here $*$ denotes differentiation with respect to the second variable, and the functional dependence has been omitted. If the system (55) can be forced to track \mathbf{e}_k , then the corresponding input is $\tilde{\mathbf{u}} = \mathbf{g}'(\mathbf{u}_k)^{-1} \mathbf{e}_k$ which can then be used in the update (54). Because this is itself an ILC problem, the norm-optimal approach is an obvious candidate for application to the time-varying system (56). Proceeding accordingly, the input to (55) on trial $m+1$ of norm-optimal ILC is chosen to minimize

$$\begin{aligned}J_{m+1} &= \frac{1}{2} \left(\sum_{t=1}^{N_s} \left[\mathbf{e}_k(t) - \tilde{\vartheta}_{f,m+1}(t) \right]^T \mathbf{Q} \left[\mathbf{e}_k(t) - \tilde{\vartheta}_{f,m+1}(t) \right] \right. \\ &\quad \left. + \sum_{t=0}^{N_s-1} \left[\tilde{\mathbf{u}}_{m+1}(t) - \tilde{\mathbf{u}}_m(t) \right]^T \mathbf{R} \left[\tilde{\mathbf{u}}_{m+1}(t) - \tilde{\mathbf{u}}_m(t) \right] \right)\end{aligned}$$

and M trials are performed in order to obtain the input $\tilde{\mathbf{u}}$. Its subsequent application in (54) again means the ability of norm-optimal ILC to transparently influence the error and stimulation levels, is able to directly translate into control over the stimulation input. Hence the potential for rehabilitation can be maximized.

7.3. Human Subject Results

The control laws were tested on a 60-year-old subject (and so age-matched with stroke participants). The reference path used comprised the extension component of elliptical reaching trajectories, as shown in Fig. 22, which were similar to those used in the clinical trials. The reference was followed at two constant speeds to produce 2.5- and 5-second duration movements, which were then preceded

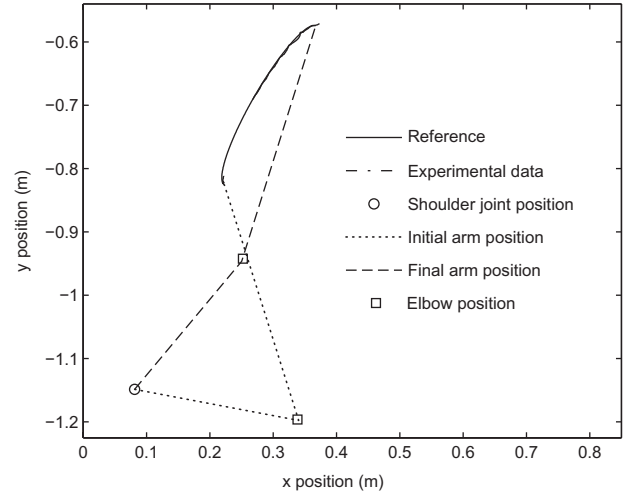


Fig. 22. Reference trajectory and position of subject tested.

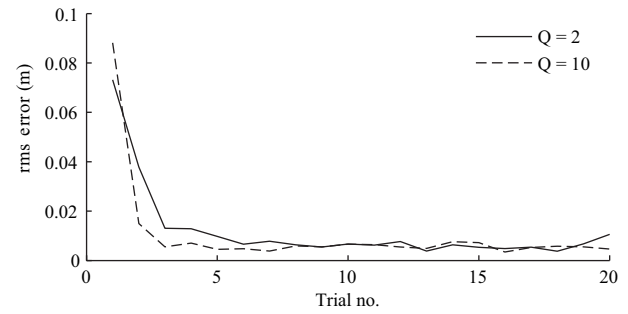


Fig. 23. Slow trajectory error results for linearized system.

by a 5-second stationary period to produce ‘slow’, and ‘fast’ trajectories. The goal of the two ILC approaches is for the forearm angle to track the forearm component, ϑ_f^* , of these trajectories. In both cases $N = 20$ trials were performed as this is the maximum number used in clinical practice. Due to the time required to identify the stimulated arm model, as well as the effect of muscle fatigue, it was not possible to perform sufficient tests on the subject to generate a PI_{20} plot similar to Fig. 5, and gains were instead selected heuristically. A sample frequency of 40 Hz was used in all tests undertaken.

Focusing initially on the linearized system implementation, a proportional plus derivative (PD) controller was first selected with values of $K_p = 6$, $K_d = 0.5$ chosen to affect a compromise between disturbance rejection and tracking performance. To provide the controlled system state vector elements, a Kalman estimator was implemented, using output and state covariance weights of 1 and 10, respectively.

Figs. 23 and 24 show error norm results for the slow and fast trajectories, respectively, each using Q values of 2 and 10, with R set at unity. The root mean square (rms) error is used to allow comparison with the simpler ILC

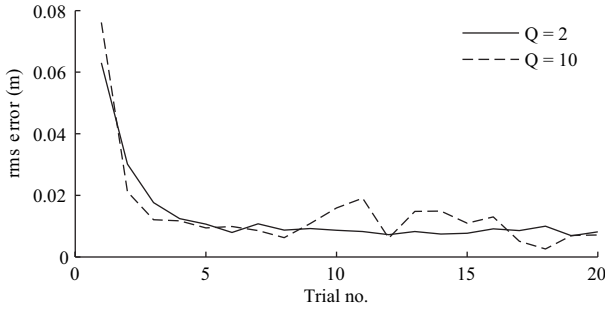


Fig. 24. Fast trajectory error results for linearized system.

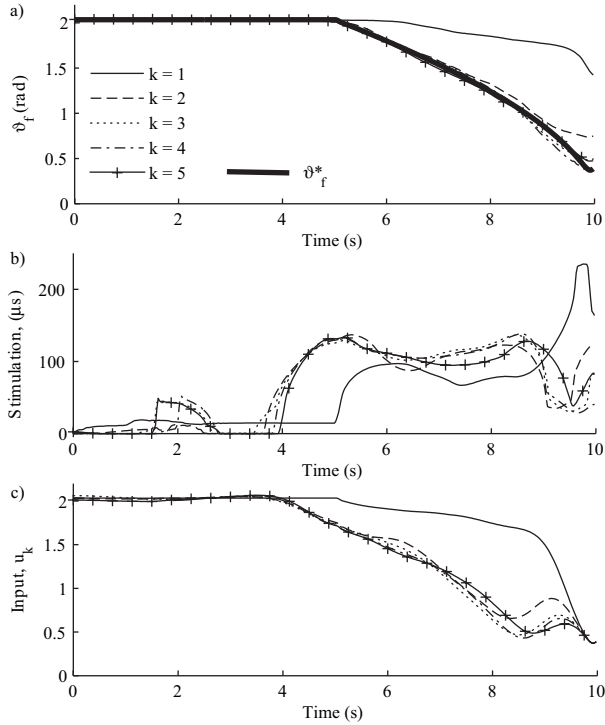


Fig. 25. (a) Tracking, (b) stimulation, and (c) input u_k signals using slow trajectory with $Q = 10$.

laws described in [19]. The larger Q value leads to faster convergence in each case, although increased error fluctuation is apparent in the latter. Although larger error values occur for the fast trajectory, accurate tracking is achieved in both cases. Fig. 25 shows the tracking, stimulation, and ILC input for the slow trajectory with $Q = 10$. It can be seen from (b) that norm-optimal ILC causes the maximum level of stimulation applied to decrease as the trials progress, and leads to its application during the initial 5 second waiting period. Fig. 25a,c illustrates the convergence of the forearm angle, ϑ_f , and norm-optimal ILC update, u_k , respectively.

Turning to the Newton method, the norm-optimal ILC parameters were selected as $M = 30$, $Q = 50\,000$ and

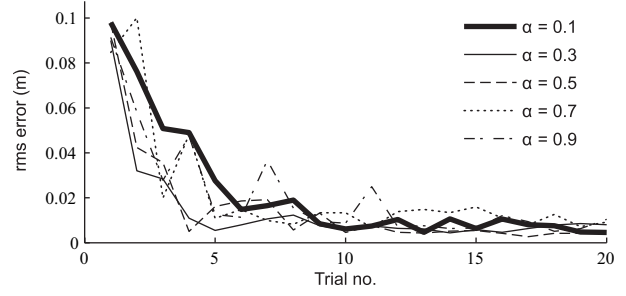


Fig. 26. Slow trajectory error results using various α .

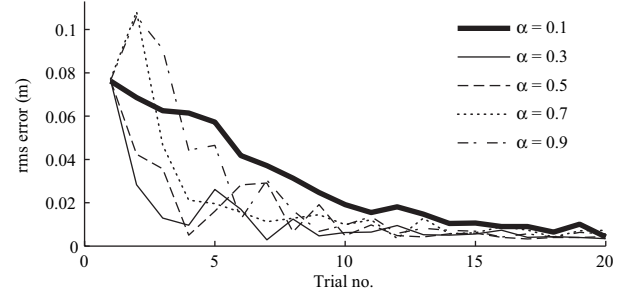


Fig. 27. Fast trajectory error results using various α .

$R = 1$ in order to produce an update which tracks the error well but does not produce an excessively large input for the subject. Figs. 26 and 27 show error results for slow and fast trajectories respectively, using various values of α . In both cases $\alpha = 0.3$ yields the most rapid convergence.

Fig. 28a shows tracking results using the fast trajectory with $\alpha = 0.3$. The reference is seen to be closely followed by the 5th trial, and the corresponding ILC update, u_k , (which corresponds to the stimulation pulsewidth in this open-loop system), is shown in Fig. 28b. Because it directly addresses system nonlinearity instead of using approximate cancellation, the Newton-based implementation generally leads to superior tracking when applied to more rapid trajectories, and results in a smoother stimulation signal which is more comfortable for the patient.

8. Conclusions and Further Work

This article has given an overview of norm-optimal ILC, starting with the theoretical foundations together with results from experimental benchmarking on a gantry robot system specially designed for this task where the pick and place operation used closely resembles task found in many industrial applications in, for example, the food processing industry. This has been followed by results from an application area involving a real process where even the first experimental test results show sufficient performance to merit further development work. The final section of

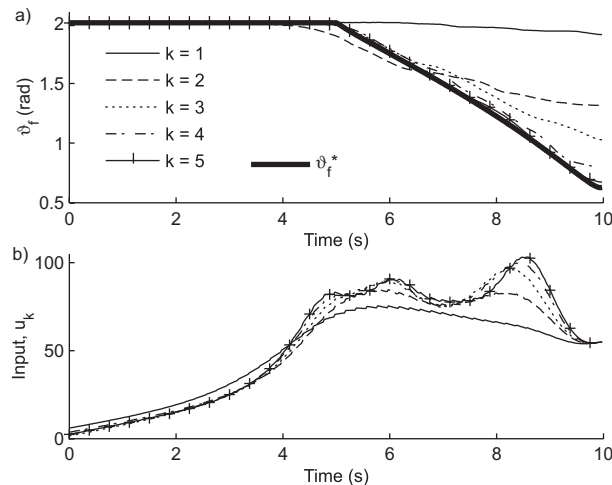


Fig. 28. (a) Tracking and (b) input signals using medium trajectory with $\alpha = 0.3$.

the article describes a recent application to stroke rehabilitation, involving ES applied to the human arm. The high levels of performance achieved in these contrasting applications demonstrate that the potential revealed by the theory of this form of ILC can be translated successfully to application domains. In doing so, norm-optimal ILC is able to provide a high level of performance that can be tuned in a transparent manner.

These results demonstrate that the ILC area in general has much potential for rapid transition from the basic theory through to applications take up. This is still a very active areas and areas to be addressed include the case when the signals measured are not noise free and also robust norm-optimal ILC. There has been work in these areas in the literature, again consult the survey articles [1, 9] as a starting point for the relevant literature but this needs to be developed and experimentally verified.

References

1. Ahn H, Chen Y, Moore KL. Iterative learning control: survey and categorization 1998–2004. *IEEE Trans Syst Man Cybern Part C Appl Rev* 2007; 37(6): 1099–1121
2. Anderson BDO, Moore JB. *Optimal Control—Linear Optimal Control*. Prentice-Hall, Englewood Cliffs, USA, 1989
3. Amann N, Owens DH, Rogers E. Iterative learning control using optimal feedback and feedforward actions. *Int J Control* 1996; 65(2): 277–293
4. Amann N, Owens DH, and Rogers E. Predictive optimal iterative learning control. *Int J Control* 1998; 69(2): 203–226
5. Amann N, Owens DH, Rogers E. Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proc Control Theory Appl* 1996; 143(2): 217–224
6. Arimoto S, Kawamura S, Miyazaki F. Bettering operations of robots by learning. *J Robot Syst* 1984; 1: 123–140
7. Athans MY, Falb PL. *Optimal Control*. McGraw-Hill, New York, USA, 1966
8. Brinkmann R et al. TESLA XFEL Technical Design Report. DESY, Hamburg, 2002
9. Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. *IEEE Control Syst Mag* 2006; 26(3): 2039–2114
10. Broeks JG, Lankhorst GJ, Rumping K, Prevo AJ. The long-term outcome of arm function after stroke: Results of a followup study. *Disability Rehabil* 1999; 21: 357–364
11. Burridge JH, Ladouceur M. Clinical and therapeutic applications of neuromuscular stimulation: A review of current use and speculation into future developments. *Neuromodulation* 2001; 4(4): 147–154
12. Chen K, Longman RW. Stability issues using FIR filtering in repetitive control. *Adv Astronaut Sci* 2002; 206: 1321–1339
13. Clements DJ, Anderson BDO. *Singular Optimal Control: The Linear-Quadratic Problem*. Lectures Notes in Control and Information Sciences, Vol. 5, Springer Verlag, Berlin, 1978
14. Colgate JE, Hogan N. Robust control of dynamically interacting systems. *Int J Control* 1988; 48(1): 65–88
15. Davoodi R, Andrews BJ. Computer simulation of FES standing up in paraplegia: a self-adaptive fuzzy controller with reinforcement learning. *IEEE Trans Rehabil Eng* 1998; 6(2): 151–161
16. Francis BA. The Optimal Linear-quadratic time-invariant regulator with heap control. *IEEE Trans Autom Control* 1979; 24: 616–621
17. Freeman CT, Hughes AM, Burridge JH, Chappell PH, Lewin PL, Rogers E. A model of the upper extremity using surface FES for stroke rehabilitation. *ASME J Biomech Eng* 2009; 131(3): 031011-1–031011-12
18. Freeman CT, Hughes AM, Burridge JH, Chappell PH, Lewin PL, Rogers E. A robotic workstation for stroke rehabilitation of the upper extremity using FES. *Med Eng Phys* 31(3): 364–373
19. Freeman CT, Hughes AM, Burridge JH, Chappell PH, Lewin PL, Rogers E. Iterative learning control of FES applied to the upper extremity for rehabilitation. *Control Eng Pract* 2009; 17(3): 368–381
20. Furuta K, Yamakita M. The design of a learning control system for multivariable systems. In: Proceedings of the IEEE International Symposium on Intelligent Control, Philadelphia, Pennsylvania, USA, 1987, 371–376
21. Graupe D, Kordylewski H. Neural network control of neuromuscular stimulation in paraplegics for independent ambulation. In: Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 1997, 1088–1091
22. Hughes AM, Freeman CT, Burridge JH, Chappell PH, Lewin PL, Rogers E. Feasibility of iterative learning control mediated by functional electrical stimulation for reaching after stroke. *J Neurorehabil Neural Repair* 2009; 23(6): 559–568
23. de Kroon JR, van der Lee JH, Ijzerman MJ, Lankhorst GJ. Therapeutic electrical stimulation to improve motor control and functional abilities of the upper extremity after stroke: a systematic review. *Clin Rehabil* 2002; 16: 350–360
24. de Kroon JR, Ijzerman MJ, Chae J, Lankhorst GJ, Zilvold G. Relation between stimulation characteristics and clinical outcome in studies using electrical stimulation to improve motor control of the upper extremity in stroke. *J Rehabil Med* 2005; 37: 65–74

25. Hunt KJ, Munih M, de Donaldson N. Feedback control of unsupported standing in paraplegia—part I: optimal control approach. *IEEE Trans Rehabil Eng* 1997; 5(4): 331–340
26. Hunt KJ, Jaime RP, Gollee H. Robust control of electrically stimulated muscle using polynomial H1 design. *Control Eng Pract* 2001; 9: 313–328
27. Jezernik S, Wassink RGV, Keller T. Sliding mode closed-loop control of FES: controlling the shank movement. *IEEE Trans Biomed Eng* 2004; 51(2): 263–272
28. Kichhoff S, Schmidt C, Lichtenberg G, Werner H. An iterative learning algorithm for control of an accelerator based free electron laser. In: Proceedings of the 47th IEEE Conference on Decision and Control, 2008, 3032–3037
29. Le F, Markovsky I, Freeman CT, Rogers E. Identification of electrically stimulated muscle after stroke. *Control Eng Pract* 2010; 18: 396–407
30. Lin T, Owens DH, Htunen JJ. Newton method based iterative learning control for discrete non-linear systems. *Int J Control* 2006; 79(10): 1263–1276
31. Ljung L. *System Identification—Theory for the User*. Prentice Hall, New Jersey, USA, 1999
32. Longman RW. Iterative learning control and repetitive control for engineering practice. *Int J Control* 2000; 73(10): 930–954
33. Parker VM, Wade DT, Langton-Hewer R. Loss of arm function after stroke: measurement, frequency and recovery. *Int Rehabil Med* 1986; 8(2): 69–73
34. Previdi F, Schauer T, Savaresi SM, Hunt KJ. Datadriven control design for neuroprostheses: a virtual reference feedback tuning (VRFT) approach. *IEEE Trans Control Syst Technol* 2004; 12(1): 176–182
35. Ratcliffe JD. Iterative learning control implemented on a multi-axis system, University of Southampton, PhD Thesis, 2005
36. Ratcliffe JD, Lewin PL, Rogers E, Hatonen JJ, Owens DH. Norm-optimal iterative learning control applied to a gantry robots for automation applications. *IEEE Trans Robot* 2006; 22(6): 103–107
37. Reducing brain damage: faster access to better stroke care, 2005, HC 452 2005–2006. [Online]. Available: www.nao.org.uk.
38. Rogers E, Galkowski K, Owens DH. *Control Systems Theory and Applications for Linear Repetitive Processes*. Lectures Notes in Control and Information Sciences, Vol. 349, Springer Verlag, Berlin, 2007
39. Rushton DN. Functional electrical stimulation and rehabilitation—an hypothesis. *Med Eng Phys* 2003; 25(1): 75–78
40. Schmidt RA, Lee TD. *Motor Control and Learning: A Behavioral Emphasis*, 3rd edn. Human Kinetics Publishers, part 3: Motor Learning, 1998, pp. 261–285
41. Schmidt C, Lichtenberg G, Werner H, Simrock S. Parameter estimation and tuning of a multivariable RF controller with FPGA technique for the free electron laser FLASH. In: Proceedings of the American Control Conference, 2008, 2516–2521
42. Shue G, Crago PE, Chizeck JH. Muscle-joint models incorporating activation dynamics, moment-angle, and moment-velocity properties. *IEEE Trans Biomed Eng* 1995; 42(2): 213–223
43. Tresadern P, Thies S, Kenney L, Howard D, Goulermas JY. Artificial neural network prediction using accelerometers to control upper limb FES during reaching and grasping following stroke. In: Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New York, USA, 2006, 2916–2919
44. Thorsen R, Spadone R, Ferrarin M. A pilot study of myoelectrically controlled FES of upper extremity. *IEEE Trans Neural Syst Rehabil Eng* 2001; 9(2): 161–168
45. Xu J-X, Tan Y. Robust optimal design and convergence properties analysis of iterative learning control approaches. *Automatica* 2002; 38(10): 1867–1880