

# Application of optimal iterative learning control to the dynamic testing of mechanical structures

S Daley<sup>1\*</sup>, D H Owens<sup>1</sup>, and J Hätönen<sup>2,3</sup>

<sup>1</sup>Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK

<sup>2</sup>Borealis Polymers R&D, Porvoo, Finland

<sup>3</sup>Control Engineering Laboratory, Helsinki University of Technology, Helsinki, Finland

*The manuscript was received on 14 June 2006 and was accepted after revision for publication on 7 September 2006.*

DOI: 10.1243/09596518JSCE325

**Abstract:** Dynamic testing of mechanical components is carried out across a variety of industries to assess fatigue life or to facilitate optimal design. Such machines are predominantly hydraulically actuated and systems range from single-channel testing of small components to multichannel testing of motor vehicles and airframes. A typical requirement is that in-service loads or displacements are replicated as closely as possible on the tested structure. Such a control problem cannot normally be solved with standard feedback control and it is common to employ iterative methods whereby the actuator command signals are modified in a series of repetitive trials. The industry standard approach, the so-called inverse algorithm, does not always converge to a suitable solution. In this paper an alternative iterative approach, derived using optimization methods, is presented and shown to have superior robustness properties. This is demonstrated through the use of simulation studies and application to both laboratory-scale and full industrial-scale test rigs. Some key theoretical results are also presented that provide important support for the experimental results. In practice, the robustness properties of the new algorithm can enable the required accuracy to be obtained in a greatly reduced number of iterations in relation to the conventional approach, thereby significantly accelerating the test commissioning process.

**Keywords:** dynamic testing, iterative control, optimization methods, fluid power systems

## 1 INTRODUCTION

Servo-hydraulic component testing continues to be widely used across a variety of industries, not only for durability analysis but also to facilitate optimum system design prior to large-scale manufacture. Use of such systems is well established in the materials testing, aerospace, and automotive sectors, with applications ranging from small subassemblies through to road simulation with full vehicles and large-scale airframe tests.

A requirement for most systems is that the hydraulic actuators are controlled in such a way that desired reference profiles (of force, displacement, strain, etc.) are accurately replicated at specific locations on

the test specimen or structure. Depending on the application, the nature of the reference profile can vary and ranges from simple sinusoidal signals to complex waveforms obtained from in-service measurements. Typical of this latter requirement is road load simulation, shown in Fig. 1, where the actuators must be driven to replicate accelerations, on the vehicle body, that were previously measured for a specific road surface (see reference [1] for a detailed review).

Such waveform replication represents a very demanding problem that cannot, in most cases, be solved with standard three-term actuator controllers. As a result, an alternative methodology is widely utilized in practice that exploits the repetitive nature of the testing process and whereby the controller command signal is modified in an iterative series of trials until the measured signal on the specimen closely matches that required. Once the command

\* Corresponding author: Department of Automatic Control and Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK. email: steve.daley@shef.ac.uk



**Fig. 1** Road load simulation (courtesy of Jaguar Cars)

or drive signal is obtained in this way, the test proceeds, usually without further modification, for the required number of cycles.

The general problem of iteratively determining an input function  $u^*(t)$  so that the resultant output function  $y(t)$  tracks a reference signal  $r(t)$  *exactly* over a specified time interval  $t \in [0, N]$  has attracted a large academic research activity over the last two decades and has led to a class of algorithms known collectively as iterative learning control (ILC) (see reference [2] for a recent review). Academic research activity in ILC was originally motivated by the problems associated with the repetitive operation of robotic manipulators and began to flourish in 1984 following the key publication by Arimoto *et al.* [3], although the 1978 paper by Uchiyama [4] is now believed to be the first academic contribution to ILC. The industrial application of ILC-type algorithms, however, predates this intense academic activity, and development appears to have largely taken place independently. A US patent, 'Learning control of actuators in control systems' [5], was submitted as early as 1968, and many modern commercial systems have their foundation in the algorithm proposed by a team at General Motors in a technical report in 1976 [6]. This algorithm is commonly known as the 'inverse algorithm' since the control signal is updated on every iteration by a weighted sequence that is formed by filtering the previous iteration error using an inverse dynamic model of the plant. The inverse algorithm lies at the core of several commercial control software packages that are widely used in multichannel test facilities.

The inverse algorithm is most commonly implemented in the frequency domain where the system under test is assumed to be described in the frequency band of interest by the transfer function (presented

here for the single-input, single-output (SISO) case but also applicable for multiple-input, multiple-output (MIMO) systems)

$$H_{yu}(\omega_k) = \frac{S_{yu}(\omega_k)}{S_{uu}(\omega_k)} \quad (1)$$

where  $S_{yu}(\omega_k)$  is the cross-spectral density between the system response and the command signal,  $S_{uu}(\omega_k)$  is the autospectral density of the command signal, and  $\omega_k$  represents a discrete set of frequency values.

For a perfectly modelled linear system, the command needed to reproduce the desired response signal  $r(t)$  can be determined from

$$u^*(t) = F^{-1} \{H_{yu}^{-1}(\omega_k)F(r(t))\} \quad (2)$$

where  $F(\cdot)$  represents the Fourier transform operator and  $F^{-1}(\cdot)$  its inverse. In practice, however, such systems rarely approximate with sufficient accuracy to a linear system and so iterative methods are utilized as described above. The procedure is initiated by using expression (2) as a first estimate of the ideal command trajectory (usually weighted to provide a degree of caution). With this command sequence, the actual response sequence,  $y(t)$ , is used to determine the error

$$e(t) = r(t) - y(t) \quad (3)$$

The error is then used to calculate a correction sequence from

$$d_k(t) = F^{-1} \{H_{yu}^{-1}(\omega_k)F(e(t))\} \quad (4)$$

This in turn is utilized to provide an updated command sequence according to

$$u_{k+1}(t) = u_k(t) + \beta d_k(t) \quad (5)$$

where  $\beta \in [0, 1]$  is a learning gain used to introduce caution into the algorithm and as a mechanism for reducing the effect of modelling errors. The process continues with the steps in equations (3) to (5) being repeated until the desired accuracy is obtained.

Despite its popularity and the fact that the approach is frequently successful where conventional control methods fail, the inverse algorithm is known to have some limitations. In particular the convergence performance lacks robustness in the face of significant modelling error and is also sensitive to sensor noise. In industrial practice, such problems manifest themselves as difficulties in setting an appropriate learning gain with ultimately very slow convergence requiring a large number of iterations to reach an adequate solution; also, in the case of severe parasitic and resonant dynamics, there is error divergence at specific frequencies.

In the current paper, recent results [7, 8] that enable an assessment of the robustness of both the inverse algorithm and a class of ILC algorithms resulting from the use of optimization methods<sup>†</sup> (see reference [9]) are reviewed. The practical potential of this class of algorithm has previously been demonstrated for robotic applications [10]. Here, the significance of the robustness results are explored in the context of dynamic testing application experience and the paper also includes a comparison of the relative performance of the two approaches using both laboratory-scale and full-scale industrial test facilities.

## 2 MATRIX REPRESENTATION OF PLANT DYNAMICS

In order to provide a framework for stability and robustness analysis of the ILC algorithms, an alternative matrix representation of the controlled system is developed in this section. As a starting point, consider a standard discrete-time, linear, time-invariant state-space representation defined over a *finite, discrete* time interval,  $t \in [0, N]$  (in order to simplify notation it is assumed that the sampling interval,  $t_s$ , is unity and the plant is SISO; the results are readily extendible to the MIMO case). The system is assumed to be operating in a repetitive mode where, at the end of each repetition, the state is reset to a specified *repetition-independent* initial condition for the next operation during which a new control signal can be used. As noted above, a reference signal  $r^*(t)$  is assumed to be specified and the ultimate control objective is to find an input function  $u^*(t)$  so that the resultant output function  $y(t)$  tracks this reference signal  $r(t)$  *exactly* on  $[0, N]$ . The process model is written in the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) & x(0) &= x_0 \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{6}$$

where  $t$  is the sample number, the state  $x(\cdot) \in R^n$ , output  $y(\cdot) \in R$ , and input  $u(\cdot) \in R$ . The operators  $A$ ,  $B$ , and  $C$  are constant matrices of appropriate dimensions and  $D$  is a scalar. In the sequel it will be assumed that either  $D \neq 0$  or that  $CA^{j-1}B = 0$ ,  $1 \leq j < k^*$ , and  $CA^{k^*-1}B \neq 0$  for some  $k^* \geq 1$  (trivially satisfied in practice) and that the system (6) is both controllable and observable. If  $D \neq 0$ , then take  $k^* = 0$ . By construction,  $k^*$  is then the relative degree

<sup>†</sup>Note that the use of this general class of algorithms in dynamic testing is the subject of a number of patent applications.

of the transfer function  $G(z)$  of the system. Also, the notation  $f_k(t)$  will denote the value of a signal  $f$  at sample interval  $t$  on iteration  $k$ .

Although the state-space model is a natural description for the *dynamic* process, an equivalent ‘static’ matrix description is more amenable to analysis. As the linear system maps input time series into output time series, it follows that there exists a matrix relating these time series. This matrix is an entirely equivalent description of the system dynamics. To construct this matrix model in  $R^{N+1}$ , define the time series ‘super-vectors’ on the  $k$ th trial via

$$u_k = [u_k(0), u_k(1), \dots, u_k(N)]^T \tag{7}$$

$$y_k = [y_k(0), y_k(1), \dots, y_k(N)]^T \tag{8}$$

$$r = [r(0), r(1), \dots, r(N)]^T \tag{9}$$

$$e_k = [e_k(0), e_k(1), \dots, e_k(N)]^T = r - y_k \tag{10}$$

With the above definitions, the relevant formulae for the input–output response of the system can be written in the form,  $k \geq 0$

$$y_k = G_e u_k + d_0 \tag{11}$$

where  $G_e$  has dimension  $(N+1) \times (N+1)$  and the lower triangular band structure  $(G_e)_{ij} = (G_e)_{(i+1)(j+1)}$  that is required by causality and time invariance of linear time-invariant convolution systems, i.e.

$$G_e = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ \mathbf{CB} & D & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & D & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \dots & \dots & D \end{bmatrix} \tag{12}$$

Also  $d_0 = [Cx_0, CAx_0, \dots, CA^N x_0]^T$ .

The elements  $CA^j B$  of the matrix  $G_e$  are the Markov parameters of the plant (6), i.e. the values of the system impulse response. Suppose that the plant transfer function  $G(z) = C(zI - A)^{-1}B + D$  has relative degree (pole-zero excess)  $k^* \geq 0$ . Assume also that the reference signal  $r(t)$  satisfies  $r(j) = CA^j x_0$  for  $0 \leq j < k^*$  (or, alternatively, that tracking in this interval is not important). Then, for analysis, it is sufficient to analyse a ‘lifted’ plant equation that is just the above if  $k^* = 0$ , or if  $k^* \geq 1$

$$y_{k,l} = G_{e,l} u_{k,l} + d_1 \tag{13}$$

where the signals  $u$ ,  $y$ ,  $e$ ,  $r$ , etc., are modified to reflect these changes. For example

$$u_{k,l} = [u_k(0), u_k(1), \dots, u_k(N - k^*)]^T$$

$$y_{k,l} = [y_k(k^*), y_k(2), \dots, y_k(N)]^T, \text{ etc.}$$

and

$$\mathbf{G}_{e,l} = \begin{bmatrix} \mathbf{CA}^{k^*-1}\mathbf{B} & 0 & 0 & \dots & 0 \\ \mathbf{CA}^{k^*}\mathbf{B} & \mathbf{CA}^{k^*-1}\mathbf{B} & 0 & \dots & 0 \\ \mathbf{CA}^{k^*+1}\mathbf{B} & \mathbf{CA}^{k^*}\mathbf{B} & \mathbf{CA}^{k^*-1}\mathbf{B} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \dots & \dots & \mathbf{CA}^{k^*-1}\mathbf{B} \end{bmatrix} \quad (14)$$

with  $d_1 = [\mathbf{CA}^{k^*}x_0, \dots, \mathbf{CA}^{N-1}x_0]^T$ . For notational convenience, the subscripts  $e, l$  are dropped and the model is written in all cases  $k^* \geq 0$  in the simplified notational form

$$y_k = \mathbf{G}u_k + d \quad (15)$$

which has the structure of discrete dynamics in  $R^{N+1-k^*}$ .

With this representation the control problem is therefore transformed to one of finding a recursive control law

$$u_{k+1} = f(u_k, u_{k-1}, \dots, u_{k-r}, e_{k+1}, e_k, \dots, e_{k-s}) \quad (16)$$

with the properties that, independent of the control input time series chosen for the first trial, the resultant sequence of error and input signals satisfies

$$\lim_{k \rightarrow \infty} \|e_k\| = 0 \quad \lim_{k \rightarrow \infty} \|u_k - u^*\| = 0 \quad (17)$$

where  $\|\cdot\|$  denotes any norm for the time series. In what follows, this norm is taken to be the Euclidean norm  $\|f\| = \sqrt{f^T f}$  in  $R_p$  which is related to the mean square error of the time series by the multiplier  $\sqrt{p}$ .

### 3 ROBUST MONOTONIC CONVERGENCE

#### 3.1 Inverse ILC algorithm

With the notation introduced in the previous section, the inverse algorithm equation (5) can be restated in the time domain as

$$u_{k+1} = u_k + \beta \mathbf{G}^{-1} e_k \quad (18)$$

Combining this with plant description (15) it follows that

$$e_{k+1} = (1 - \beta) e_k, \quad k \geq 0 \quad (19)$$

and

$$e_{k+1} = (1 - \beta)^k e_0, \quad k \geq 0 \quad (20)$$

so that convergence to zero is assured if the learning gain is in the range  $0 < \beta < 2$ . Note as well from equation (19) that a learning gain in this range also assures that convergence is monotonic, i.e.

$\|e_{k+1}\| < \|e_k\|, \forall k \geq 0$  whenever  $\|e_k\| \neq 0$ . This property of monotonicity is very important in practice since it indicates a guarantee of improvement from one trial to the next. This is particularly important in dynamic testing applications owing to the potential for catastrophic mechanical damage as a result of an unsuitable trial control sequence.

This analysis, however, makes the assumption that there is no model uncertainty; in practice this is unlikely to be the case and algorithm (18) will necessarily be implemented with a nominal model  $\mathbf{G}_o$  with  $\mathbf{G}_o \neq \mathbf{G}$ . In this case the error evolution equation is [7]

$$e_{k+1} = (I - \beta \mathbf{G} \mathbf{G}_o^{-1}) e_k, \quad k \geq 0 \quad (21)$$

Robust convergence therefore is guaranteed if, and only if

$$|1 - \beta \lambda_i(\mathbf{G} \mathbf{G}_o^{-1})| < 1 \quad \forall i \quad (22)$$

and robust *monotonic* convergence is guaranteed if

$$\bar{\sigma}(I - \beta \mathbf{G} \mathbf{G}_o^{-1}) < 1 \quad (23)$$

where  $\bar{\sigma}(\cdot)$  denotes the maximum singular value and  $\lambda_i(\cdot)$  the  $i$ th eigenvalue. A simple geometric construction shows that equation (22) can always be satisfied with a suitably small learning gain if  $\text{Re}\{\lambda_i(\mathbf{G} \mathbf{G}_o^{-1})\} > 0 \forall i$ .

If the model uncertainty is assumed to have the multiplicative form

$$G(z) = U(z)G_o(z) \quad (24)$$

where  $U(z)$  is causal and stable, then, on the assumption that  $G(z)$  has relative degree greater than or equal to  $G_o(z)$ , equation (24) has the equivalent matrix form (see section 2)

$$\mathbf{G} = \mathbf{U} \mathbf{G}_o \quad (25)$$

and so convergence can be guaranteed for sufficiently small learning gain if  $\text{Re}\{\lambda_i(\mathbf{U})\} > 0 \forall i$ . Since in this expression  $\mathbf{U}$  is unlifted, then the condition can only be satisfied for multiplicative uncertainty when  $G(z)$  has relative degree equal to  $G_o(z)$ . It can be shown that an alternative result in the frequency domain (see reference [7]) is that robust monotonic convergence is guaranteed if

$$|1 - \beta U(z)| < 1, \quad \forall |z| = 1 \quad (26)$$

To satisfy this with a suitably small learning gain requires that  $U(z)$  is positive real, i.e.  $\text{Re}\{U(z)\} > 0, \forall |z| = 1$ . The significance of this result will be illustrated further in the following subsections.

### 3.2 Gradient-based ILC algorithm

Again using the notation of the previous sections, consider the matrix model  $\mathbf{y}_k = \mathbf{G}\mathbf{u}_k + \mathbf{d}$ ,  $k \geq 0$ , where  $\mathbf{r}$  is the desired reference time series vector,  $\mathbf{e}_k = \mathbf{r} - \mathbf{y}_k$  is the error time series on the  $k$ th trial, and the initial control input time series  $\mathbf{u}_0$  has been specified with  $e_0$  as the corresponding error. The resultant error is  $\mathbf{e}_k = \mathbf{r} - \mathbf{d} - \mathbf{G}\mathbf{u}_k$ . A simple analysis of  $\|\mathbf{e}_k\|^2 = \mathbf{e}_k^T \mathbf{e}_k$  indicates that the steepest descent direction for the error is just  $\mathbf{G}^T \mathbf{e}_k$  and hence that the alternative feedforward ILC algorithm (referred to as the gradient algorithm in the sequel)

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \beta \mathbf{G}^T \mathbf{e}_k \quad (27)$$

may be capable of ensuring a monotonic convergence of the Euclidean error norm sequence for suitable choice of  $\beta$ . With this algorithm it can be shown that the error evolution equation is

$$\mathbf{e}_{k+1} = (\mathbf{I} - \beta \mathbf{G}\mathbf{G}^T) \mathbf{e}_k, \quad k \geq 0 \quad (28)$$

and so *monotonic* convergence is guaranteed if

$$\bar{\sigma}(\mathbf{I} - \beta \mathbf{G}\mathbf{G}^T) < 1 \quad (29)$$

Convergence will be obtained if, and only if,  $|1 - \beta \lambda_i(\mathbf{G}\mathbf{G}^T)| < 1 \forall i$  or alternatively when

$$\beta < \frac{2}{\bar{\sigma}(\mathbf{G})^2} \quad (30)$$

It will be noted that, unlike the equivalent convergence condition for the inverse algorithm, this is dependent on the gain of the plant. So, for example, a change in units for a specific system model would necessitate a corresponding change in learning gain to ensure that the convergence properties for both definitions remain identical.

An analysis of the robustness can be developed in a similar way to the previous section by defining a multiplicative uncertainty as in equations (24) and (25). The equivalent condition to equation (26) for robust monotonic convergence of algorithm (27) is [8]

$$|1 - \beta |\mathbf{G}_o(z)|^2 \mathbf{U}(z)| < 1, \quad \forall |z| = 1 \quad (31)$$

From this it is clear that the gradient algorithm also requires the uncertainty to be positive real to ensure monotonic convergence. Note that, if the condition is violated, asymptotic convergence may still be retained but the error norm may increase from one trial to the next. The condition for the monotonic convergence of the inverse algorithm, equation (26), implies that large uncertainty associated with high-frequency parasitic dynamics, for example, necessitates a small learning gain and therefore a large number of trials to enable acceptable error. However, condition (31) shows that if  $|\mathbf{G}_o(z)|$  is small at the frequencies

associated with large uncertainty then, for monotonic convergence, the learning gain in algorithm (27) can be larger than that associated with the inverse algorithm. Although the converse is also true when the magnitude of  $|\mathbf{G}_o(z)|$  is greater than unity at frequencies corresponding to large uncertainty, this is frequently not the case in systems associated with dynamic test. This is because modelling problems are often associated with parasitic dynamics at high frequency when  $\mathbf{G}_o(z)$  has low-pass characteristics, with poor signal-to-noise ratios around severe zeros or at low frequency when  $\mathbf{G}_o(z)$  has high-pass characteristics. The contrast in performance is illustrated further using a representative example in the following subsection.

### 3.3 Illustrative example

In order to highlight the significance of the robustness results, a simple but representative example associated with high-frequency modelling error is simulated. The system considered is illustrated in Fig. 2 and comprises a two-degree-of-freedom structure with a well-damped low-frequency mode and a higher-frequency, highly resonant mode. The control problem considered is to determine a control force sequence  $f_c$  (applied at the top mass) such that a target force sequence (transmitted to the foundation) is accurately replicated. In practice, the damping levels in higher-frequency modes are often overestimated owing to the effects of static friction, which are more apparent at the excitation levels at which the identification is carried out (usually conservatively low to preserve the life of the test specimen). As a result the study evaluates the effect of modelling errors in the higher-frequency mode only.

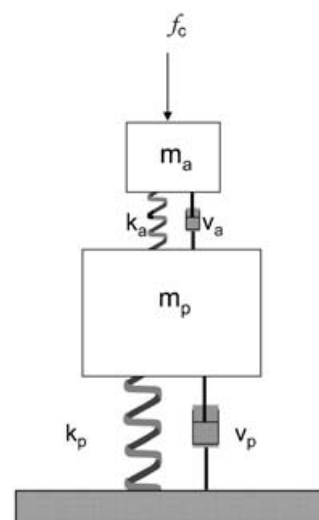


Fig. 2 Simple two-degree-of-freedom structure

The frequency response of the nominal plant, the true plant, and the uncertainty are shown in Fig. 3. It is clear that since the phase of the uncertainty lies within  $\pm 90^\circ$  this satisfies the requirement that  $\text{Re}\{U(z)\} > 0, \forall |z| = 1$  (note this will always be satisfied when the only error is associated with the level of damping).

With a reference signal of

$$r(j) = \left[ 1 + 0.1 \sin\left(\frac{20\pi j}{50}\right) \right] \cosh(j/50) \times \sin\left\{ 6\pi \left[ \frac{j}{50} \left( 2 - \frac{j}{50} \right) \right] \right\}, \quad 0 \leq j \leq 50$$

(chosen to excite a wide range of plant frequencies) and learning gain of 0.5 for both algorithms the evolution of the error norm (presented throughout the paper as  $20 \log_{10} \|e_k\|$ ) is as shown in Fig. 4.

An interesting feature of this result is that, although the error evolution of the inverse algorithm is clearly not monotonic, asymptotic convergence is displayed and the accuracy of the gradient algorithm is exceeded after 15 iterations. However, in practice the inverse algorithm progression would have been stopped after the third or fourth iteration to avoid subsequent excessive error amplitudes. The optimum gains for this example can be obtained by plotting the values of  $\beta$  that satisfy equations (26) and (31) as a function of frequency. This is shown in Fig. 5 where it can be seen that the model uncertainty associated with the parasitic resonance limits the inverse gain to 0.15 whereas the gradient gain need only be less than 2.

Figure 6 shows the error evolution for an inverse gain of 0.15 and a gradient gain of 1.9. Although the error evolution for both algorithms now displays

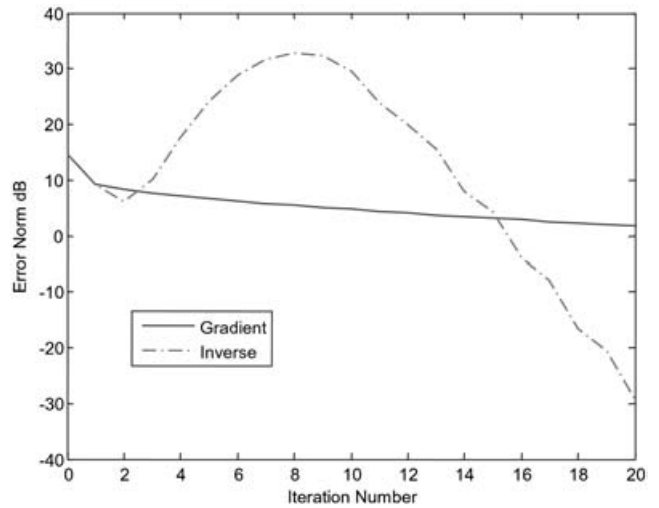


Fig. 4 Error norm evolution for 0.5 learning gain

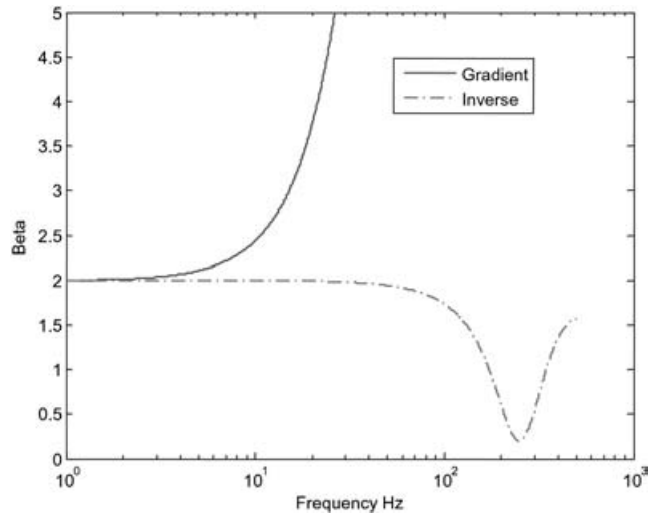


Fig. 5  $\beta$  as a function of frequency

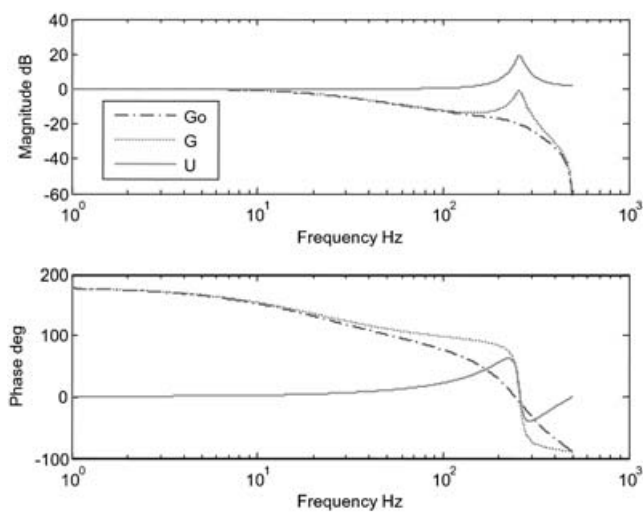


Fig. 3 Frequency response of nominal plant, actual plant, and uncertainty

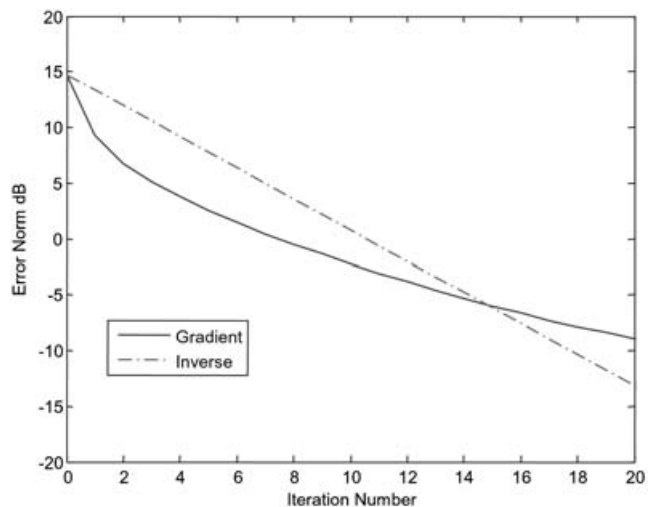


Fig. 6 Error evolution for optimal learning gains

monotonic convergence, it is clear that the initial convergence of the gradient algorithm is superior, leading to a sufficient accuracy in fewer iterations. Such rapid initial convergence is particularly important in test applications in order to reduce expensive set-up times and to avoid artificially reducing the life of the specimen.

## 4 APPLICATION STUDIES

Some key results that provide theoretical evidence of the enhanced robustness of the gradient algorithm for a specific class of uncertainty have been presented in the previous section. The authors have also been involved, however, in a number of application studies, both at laboratory scale and full industrial scale, which provide practical evidence that this class of algorithm has broad robustness properties that indicate generally improved performance over the industrial standard inverse algorithm. Results from two of these application studies are presented in this section.

### 4.1 Displacement control

The first system considered is the hydraulic position control test rig pictured in Fig. 7 and shown schematically in Fig. 8. This was designed and built by the ALSTOM Power Technology Centre in Whetstone, UK, for dynamic performance testing of autotuning position controllers [11] and has features that are common to many mechanical test rigs. The rig, consisting of two hydraulic cylinders (a main cylinder and a loading cylinder, which are coupled through a load cell), represents a demanding control problem owing to the presence of a number of non-linearities. The main cylinder is controlled by a three-way proportional valve that exhibits considerable non-linear behaviour, as it has significant overlap and magnetic hysteresis. The loading cylinder

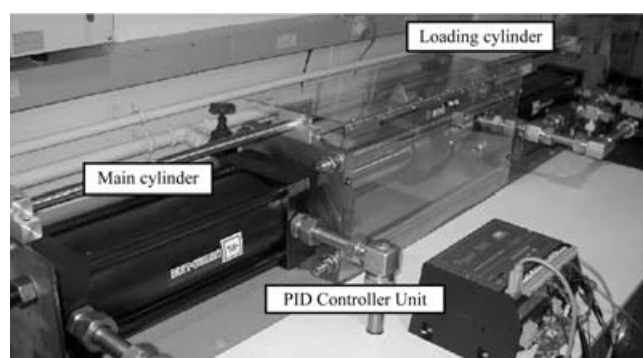


Fig. 7 Hydraulic position control test rig

is controlled by a servo valve that has been included to simulate complex variable loads. The main cylinder is also of the unequal area type, leading to a further non-linearity in the form of a directionally dependent time constant.

The flow of hydraulic fluid to the main cylinder is controlled by the three-way proportional valve using ALSTOM's digital proportional-integral-derivative (PID) speed control governor, the DIGIPID 1000. This can be seen in the foreground of Fig. 1. In the form used here the position of the main cylinder along its stroke is controlled and this is measured using a linear variable displacement transducer (LVDT) mounted above the cylinder. In order to gather performance information about the position controllers under test, other sensors monitor the following: the coil current of the valve, the force of the main cylinder, and the displacement of the loading cylinder.

The main parameters of the PID controller unit can be accessed via a serial link to data-processing software, which enables modification from a PC. This enables a position demand to be either set internally or directly from an external voltage. In the present study, the latter course is taken as the reference is generated from the ILC algorithm and implemented for convenience using a dSPACE system based around the TMS320 C40 DSP chip.

As a benchmark for the ILC controllers, their performance is compared with a PID controller that is conservatively tuned using the time domain autotuning method presented in reference [11]. The response of this system to a required target position sequence is shown in Fig. 9. The position measurement is normalized so that zero corresponds to the mid-position of the cylinder and  $\pm 5$  correspond to the extremes of travel. This clearly demonstrates the directional non-linearity that causes stability problems when trying to tighten the PID loop.

Here, the ILC approach is initiated with a command sequence that is calculated by inverting a model of the plant and applying this to the target signal. The model was derived using a frequency domain approach from input/output data collected with a small-amplitude band-limited random input. The resulting command signal from this model inversion process is shown in Fig. 10, together with the target reference.

The corresponding response is shown in Fig. 11 where it can be seen that, although the tracking in the rise period is good, the steady state error is large. With application of an ILC algorithm, the reference trajectory can be iteratively modified in an attempt to reduce this tracking error. Starting from the command sequence illustrated in Fig. 10,

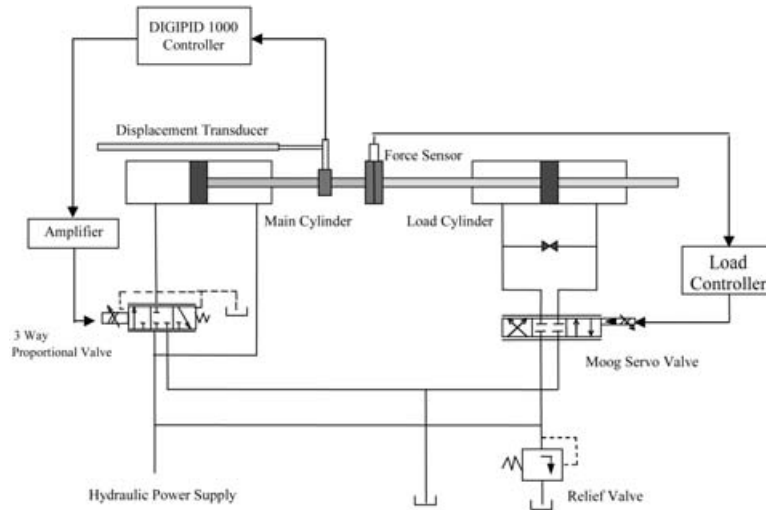


Fig. 8 Schematic diagram of the hydraulic position control test rig

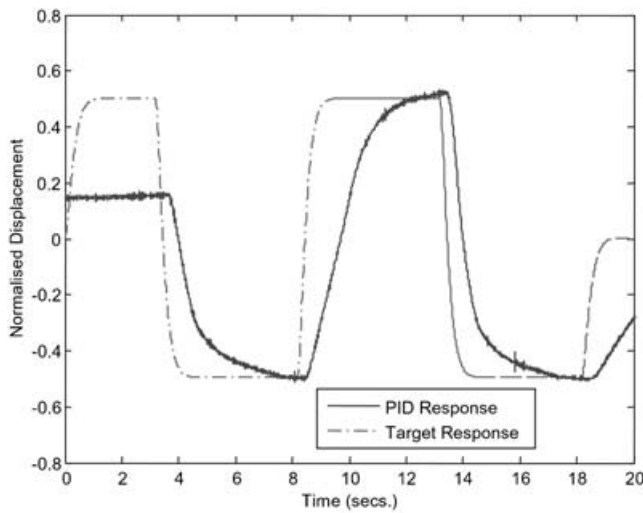


Fig. 9 Target trajectory with nominal PID controller response

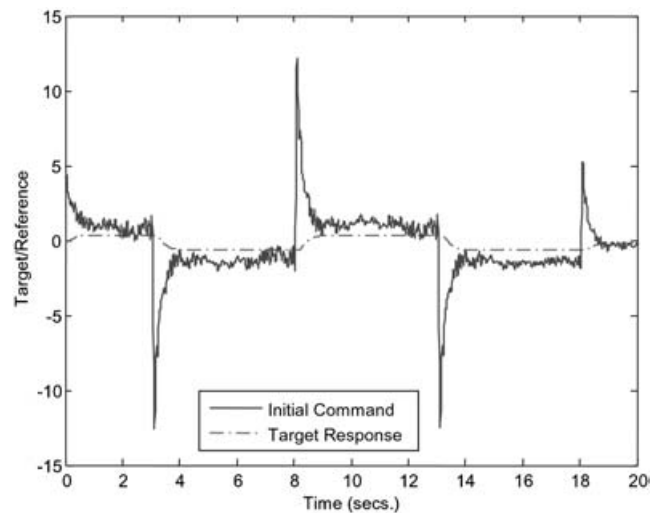


Fig. 10 Target trajectory with first iteration command from model inversion

Fig. 12 shows the evolution of the error norm with application of a gradient ILC algorithm<sup>†</sup>. It can be seen that monotonic convergence is achieved despite the clear inaccuracies in the model, thereby further demonstrating the algorithm robustness. It will also be noted that the convergence stagnates after eight iterations but is further improved from the ninth iteration by extending the bandwidth of the model.

<sup>†</sup>Although useful for analysing performance, the Markov parameter description introduced in section 2 is not required for implementation of the gradient algorithm and a frequency domain approach similar in form to that introduced in section 1 for the inverse algorithm is used in the application studies presented here.

Also shown in the figure is the error norm corresponding to the nominal PID controller; this is bettered by the ILC algorithm in one iteration.

The system response after the 12th iteration of the command sequence is shown in Fig. 13, together with the target and the response of both the nominal PID control and the initial command from the inverse model.

In order to assess the performance of the gradient algorithm relative to the inverse algorithm, the latter was also applied to calculate an iteratively updated command sequence. Starting from the same initial command (the result of which is shown in Fig. 11), the resulting error norm evolution for the inverse algorithm is also displayed in Fig. 12. It can be seen

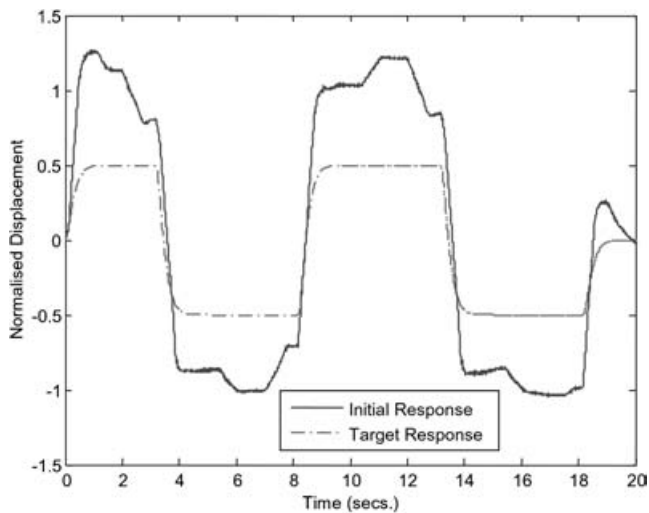


Fig. 11 Target and system response to first iteration using model inversion

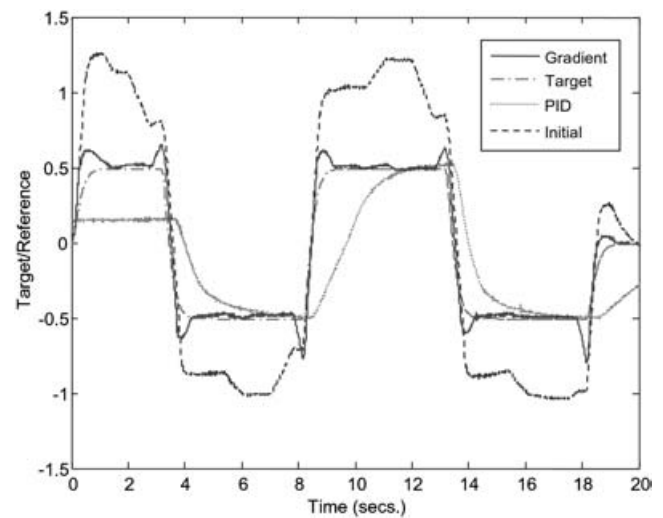


Fig. 13 Target trajectory with nominal PID response, response to command sequence from initial model inversion, and 12th gradient ILC iteration

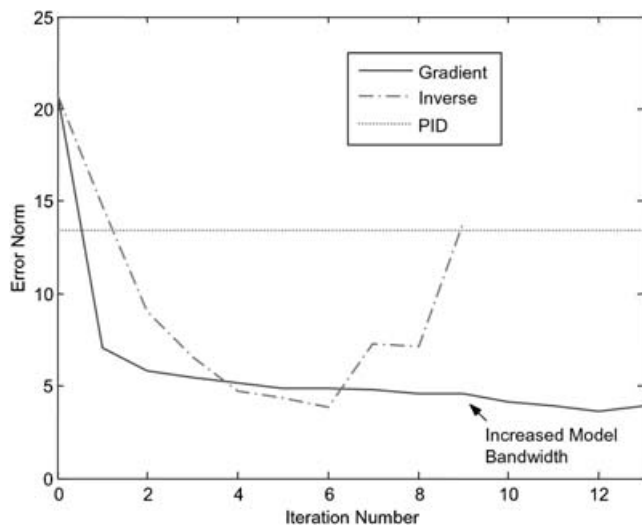


Fig. 12 Evolution of error norm with iteration

that this algorithm is slower to converge initially than the gradient ILC approach but also diverges after six iterations, providing further evidence of the latter's superior robustness properties.

#### 4.2 Automotive component test

The second application study presented uses a multichannel facility at the Jaguar Cars Whitley Engineering Centre. This is typical of the type of system used for subassembly durability testing in the automotive sector. The specific test set-up used is shown in Fig. 14 where the body component is loaded with five actuators, three of which are connected directly with the remaining two, applying bending loads through the use of a rocker arm arrangement. The substructure is mounted using a

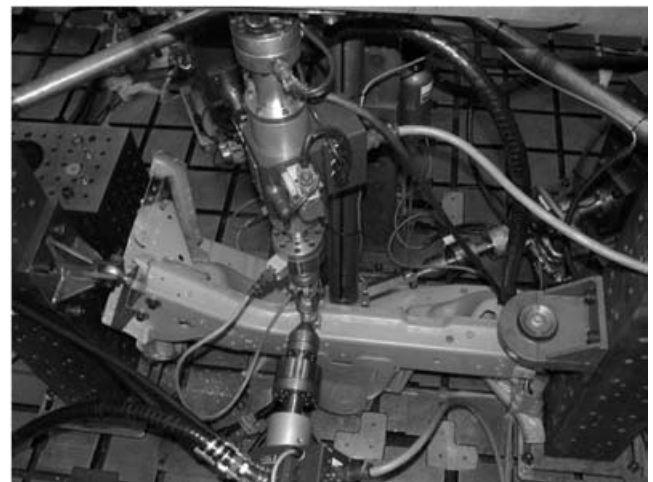


Fig. 14 Automotive substructure test

ball joint at one end and an elastomeric bush at the other, both of which add to the dynamic complexity of the response. The force applied by each actuator is measured with a load cell and each actuator is controlled with a commercial digital PID controller (each actuator control loop is identical in form to the load control shown schematically in Fig. 8). Desired forces for each actuator had previously been calculated on the basis of replicating the load on the structure that would be experienced when driving over a cobbled road surface.

A frequency response matrix model for the system was developed utilizing a small white noise perturbation on each channel. A typical magnitude response is shown in Fig. 15 where a significant element zero at 27 Hz is clearly visible. The multiple coherence function for channel 4 is shown in Fig. 16,

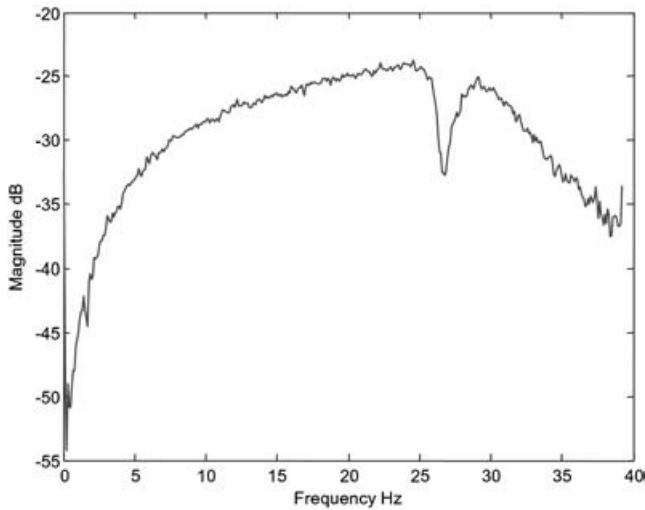


Fig. 15 Magnitude response for fourth output and third input

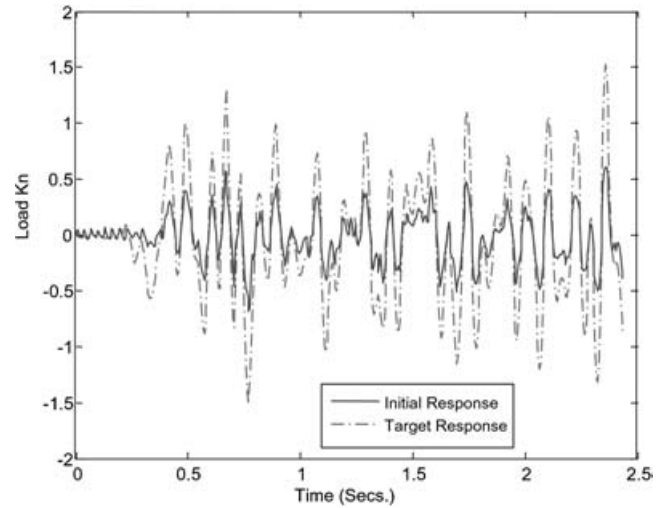


Fig. 17 Response to initialization command sequence

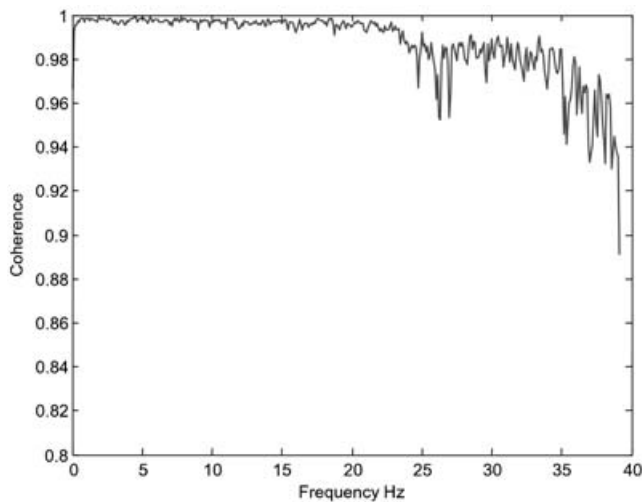


Fig. 16 Multiple coherence function for fourth channel

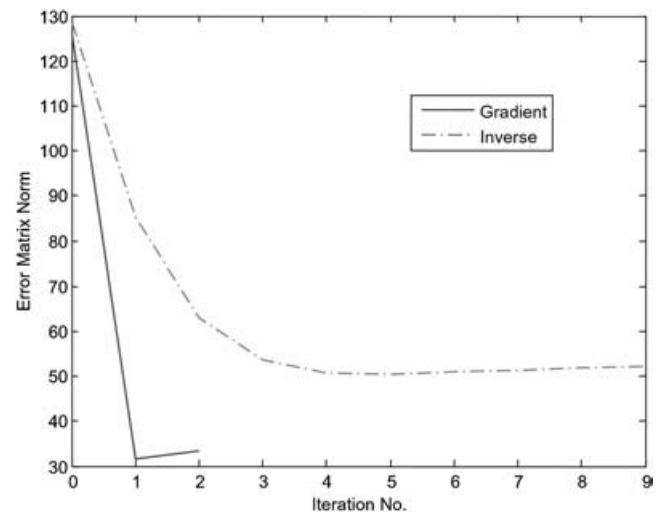


Fig. 18 Error evolution with iteration

indicating that even for this low level of excitation there is uncertainty associated with this zero and with increasing frequency.

As for the previous application example, the gradient and inverse ILC algorithms are both initiated with an identical command sequence that is formed by filtering the desired reference with the inverse of the plant model. Here, to ensure a conservative approach, this signal is scaled by 40 per cent; the resulting response in channel 1 is shown in Fig. 17.

The subsequent error evolution (taken as a norm across all channels) for both a gradient algorithm and the inverse algorithm is shown in Fig. 18. The inverse algorithm utilized a learning gain of 0.4 with increased values leading to error divergence; however, a learning gain approaching unity was possible

with the gradient algorithm, leading to the rapid convergence shown. Since this was a real industrial test specimen, the gradient algorithm was terminated on the second iteration as adequate accuracy was deemed to have been achieved (see Fig. 19).

These results again provide experimental support for the superior robustness properties of the gradient approach. A further explanation of the relative performance for this application can be obtained by evaluating the maximum singular values of the update operator for both algorithms as a function of frequency (simply the maximum singular value of the inverse of the system frequency response matrix in the case of the inverse algorithm). This provides a measure of the maximum possible update magnitude in each iteration and is shown in Fig. 20. It is clear

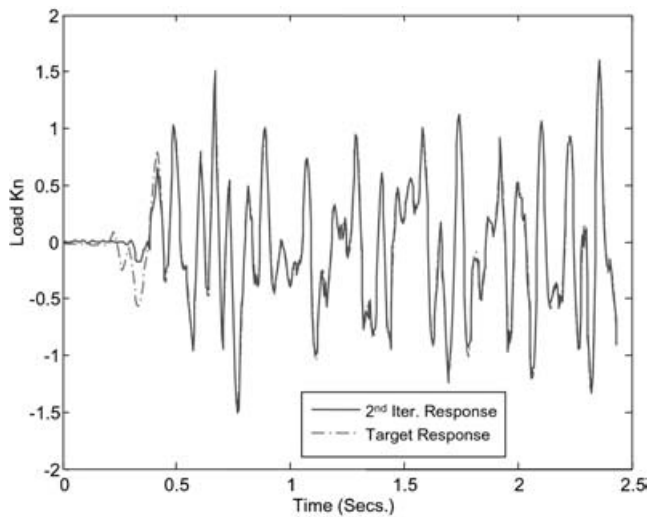


Fig. 19 Desired and actual response, second iteration

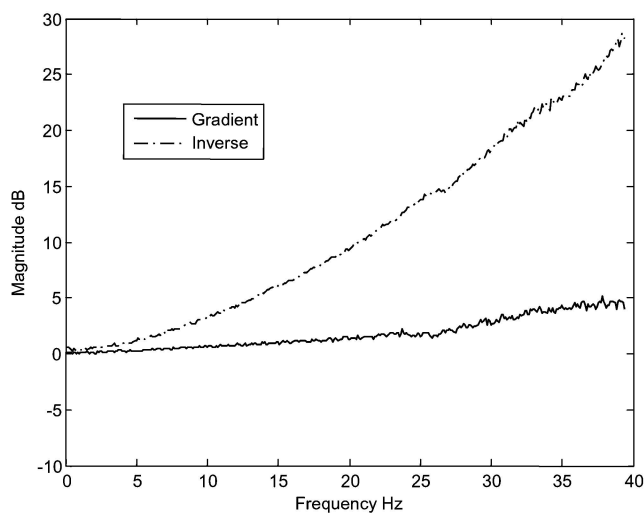


Fig. 20 Maximum singular values of update operator

that the inverse algorithm puts significantly higher weight at the higher frequencies where the model is least certain (see Fig. 16), a fact that will certainly lead to convergence problems in many applications.

## 5 CONCLUSIONS

Some recently derived robustness results for both the inverse and gradient iterative learning control algorithms have been presented. Robustness is defined in terms of the attainment of monotonic convergence of the error norm despite the presence of modelling uncertainty. It has been shown that for multiplicative uncertainty, monotonic convergence is possible for both algorithms with a suitably small

learning gain, provided the phase of the uncertainty remains within  $90^\circ$  of the nominal. The gradient algorithm, however, allows for a larger learning gain when the uncertainty is associated with frequencies where the plant magnitude is low. It has been argued that systems with such features commonly occur in the area of dynamic testing and that therefore a class of algorithms based on optimization methods can potentially outperform the industrial standard inverse algorithm. The paper has also presented some application studies on both laboratory-scale and full-industrial-scale test facilities that provide significant support for this argument.

## ACKNOWLEDGEMENTS

The authors are grateful to Peter Knight of ALSTOM, Steve Morris of Jaguar Cars, and Norman Thornton of nCode for their support during the application studies described in the paper.

## REFERENCES

- 1 **Dodds, C. J. and Plummer, A. R.** Laboratory road simulation for full vehicle testing – a review. In Proceedings of Symposium of *International automotive technology*, Pune, India, January 2001, SAE paper 2001-01-0047 (Society of Automotive Engineers).
- 2 **Longman, R.** Iterative learning control and repetitive control for engineering practice. *Int. J. Control*, 2000, **73**, 930–954.
- 3 **Arimoto, S., Kawamura, S., and Miyazaki, F.** Bettering operation of robots by learning. *J. Robotic Syst.*, 1984, **1**, 123–140.
- 4 **Uchiyama, M.** Formation of high speed motion pattern of mechanical arm by trial. *Trans. Soc. Instrumn Control Engrs*, 1978, **19**, 706–712.
- 5 **Garden, M.** (Leeds and Northrup). Learning control of actuators in control systems. US Pat. 03 555 252, January 1971.
- 6 **Cryer, B. W., Nawrocki, P. E., and Lund, R. A.** A road system simulation system for heavy duty vehicles. SAE technical report 760361, 1976.
- 7 **Harte, T. J., Hatonen, J., and Owens, D. H.** Discrete-time inverse model-based iterative learning control: Stability, monotonicity and robustness. *Int. J. Control*, 2005, **78**, 577–586.
- 8 **Owens, D. H., Hatonen, J., and Daley, S.** Robust nonlinear gradient-based discrete-time iterative learning control algorithms. Research Report No. 916, Department of Automatic Control and Systems Engineering, University of Sheffield, 2006 (also submitted to *Int. J. Robust and Nonlinear Control*).
- 9 **Owens, D. H. and Hatonen, J.** Iterative learning control – an optimization paradigm. *Annual Rev. Control*, 2005, **29**, 57–70.

- 10 Ratcliffe, J. D., van Duinkerken, L., Lewin, P. L., Rogers, E., Hatonen, J., Harte, T. J., and Owens, D. H. Fast norm-optimal iterative learning control for industrial applications. Proceedings of *American control conference*, Portland, 8–10 June 2005.
- 11 Liu, G. P. and Daley, S. Optimal-tuning PID control for industrial systems. *IFAC J. Control Engng Practice*, 2001, **9**, 1185–1194.

$\mathbf{G}_o$	plant nominal model Markov parameter matrix
$G(z)$	plant transfer function
$G_o(z)$	plant nominal model transfer function
$H_{yu}$	plant frequency response function
$\mathbf{r}$	target sequence vector
$r(t)$	target output at sample instant $t$
$S_{uu}$	autospectral density of command input
$S_{yu}$	cross-spectral density between command input and output
$\mathbf{u}_k$	input sequence vector at iteration $k$
$u(t)$	input at sample instant $t$
$u^*(t)$	optimal input sequence
$\mathbf{U}$	model uncertainty Markov parameter matrix
$U(z)$	model uncertainty transfer function
$x_o$	initial value of state vector
$\mathbf{x}(t)$	state vector at sample instant $t$
$\mathbf{y}_k$	output sequence vector at iteration $k$
$y(t)$	output at sample instant $t$
$z$	Z-transform complex variable
$\beta$	iteration algorithm learning gain
$\bar{\sigma}(\cdot)$	maximum singular value of matrix
$\lambda_i(\cdot)$	$i$ th eigenvalue of matrix
$\omega_k$	discrete frequency value

## APPENDIX

### Notation

$\mathbf{A}$	plant state transition matrix
$\mathbf{B}$	plant input matrix
$\mathbf{C}$	plant output matrix
$\mathbf{d}$	initial condition vector in Markov parameter model
$d_k(t)$	inverse operator update sequence at iteration $k$
$D$	plant direct transmission
$\mathbf{e}_k$	error sequence vector at iteration $k$
$e(t)$	error at sample instant $t$
$\mathbf{G}$	plant Markov parameter matrix