University of Southampton Research Repository

Copyright © and Moral Rights for this thesis and, where applicable, any accompanying data are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis and the accompanying data cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content of the thesis and accompanying research data (where applicable) must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holder/s.

When referring to this thesis and any accompanying data, full bibliographic details must be given, e.g.

Thesis: Christopher Thomas Freeman (2004) "Experimental Evaluation of Iterative Learning Control Performance for Non-minimum Phase Plants", University of Southampton, School of Electronics and Computer Science, PhD Thesis, 209 pages.

UNIVERSITY OF SOUTHAMPTON

Experimental Evaluation of Iterative Learning Control Performance for Non-minimum Phase Plants

by

Christopher T. Freeman

A thesis submitted for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

March 2004

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

EXPERIMENTAL EVALUATION OF ITERATIVE LEARNING CONTROL PERFORMANCE FOR NON-MINIMUM PHASE PLANTS

by Christopher T. Freeman

This thesis describes the design and construction of a Single Input Single Output (SISO) non-minimum phase experimental test facility and the subsequent testing of a number of Iterative Learning Control (ILC) strategies. The system can be configured in three different ways in order to test the effect of increased plant complexity and non-linearity. The implementation of a number of both existing and new ILC strategies is detailed and results and analysis of their performance are presented. A principal objective has been to find the ILC controller that is most effective in forcing the output of the test-bed to follow a repetitive trajectory. The design and construction of the test-bed is explained in full and both linear and non-linear models of the system are produced. P-type, D-type and Delay-type ILC algorithms have been tested on the simplest form of the system. The phase-lead algorithm has been implemented and a method of establishing the optimum lead found, as well as a procedure to estimate unstable frequencies. Both causal and non-causal filters have been assessed for use with the algorithm. Phase-lead ILC has been implemented on the more complex plant and comparisons made with previous results. The use of a forgetting factor has been found to overcome the problem of instability, but at the expense of increased final error. The phase-lead algorithm has been vastly improved using additional phase-leads and this technique has been generalised to produce an novel optimisation routine which uses a large number of phase-leads. Its success has been confirmed with experimental results. A learning law utilising the plant adjoint fits naturally into this framework and practical results are presented. This method has been both reformulated into one which needs little plant knowledge, and also combined with deadbeat control to avoid truncation in the course of its implementation. Results are presented using these techniques and practical guidelines produced and tested. A simple method of increasing the learning at higher frequencies has been proposed and verified experimentally. An optimality based Repetitive Control algorithm has also been rigorously tested and the use of a relaxation parameter found to increase its robustness. Finally, a graphical method that represents both the robustness and the stability of an ILC algorithm applied to a known plant has been developed. This tool may find wide application when designing and developing future ILC strategies.

Contents

List of Figures

List of Tables

Acknowledgements

Thanks to my wonderful family and friends whose total disinterest in the subject of my studies did not diminish their support. Grateful acknowledgements must go to my tutors Paul Lewin and Eric Rogers for their help and guidance. Special thanks to James Ratcliffe and Jari Hätönen of Sheffield University for their continuing help and friendship.

Chapter 1

Introduction

Iterative Learning Control (ILC) is a control method that is applicable to systems which perform the same action repeatedly. Operating in this way it is able to use past control information such as input signals and tracking errors in the construction of the present control action. This sets ILC apart from most other control techniques and has allowed it to provide good performance with little knowledge of the process. Indeed, it may not even be possible to meet the control specifications involved using other control approaches. Because of this, ILC has been a popular branch of control theory since its formal conception over 20 years ago, applicable to most manufacturing and chemical processes, many robotic applications, and a wide range of other areas. Since ILC has brought about the creation of many new algorithms as well as being integrated with other control techniques, there is a large quantity of published work on the subject.

The object of this research is to address the lack of research carried out using ILC on experimental systems, and the almost complete absence of practical research performed on non-minimum phase systems. The specific aim is to select and implement as large a number of ILC algorithms as is feasible on a specially designed experimental apparatus. This necessitates the construction of a non-minimum phase test facility capable of providing a sufficient but manageable control challenge to the algorithms seen, together with a PC interface and programming environment. Because the experimental use of non-minimum phase systems is unprecedented, the natural starting place in terms of algorithms are the original P-type and D-type laws, and there exists a sufficient progression of additions and adaptations leading from them to produce an in-depth study that does not depart far from that starting point. However, to broaden the type of ILC studied, certain more complex strategies need to be examined. Each algorithm can be implemented on the experimental apparatus and its effectiveness examined using the data recorded. In order to conduct the objectives described, this thesis starts with a review of the ILC, presented in Chapter 2.

Chapter 3 describes the process of designing and building the non-minimum phase experimental test facility with special reference to ensuring that it is appropriate for use with the intended ILC algorithms. The design relates not only to the mechanical testbed, but to the interface between it and a PC, and to the software used to implement the control methods. Chapter 4 is concerned with producing models of the test facility for use with the design and simulation of more advanced controllers. Two separate models of each system configuration are produced. The first is a linear model which allows design of controllers using standard control techniques, but may sacrifice precision in the form of unmodelled dynamics and non-linearities. The second, a largely theoretical model consisting of each element of the system in turn, is therefore produced which will be used in order to verify the algorithms before they are tested on the actual experimental apparatus. The common three term controller is used in some implementations and details concerning its tuning are described. Due to the difficulties found in traditional tuning approaches, the controller's parameters are specially selected for the trajectories that will be used with the ILC algorithms. This involves ensuring that the output follows a time delayed copy, capturing its shape rather than its position. Chapter 5 details the implementation of some basic ILC algorithms. These are the P-type, D-type and Delay-type laws, the former two being well established in the control literature, the latter less well established. Their performance is assessed by analysis of a large amount of experimental data, and conclusions are made concerning their effectiveness and operation. The phase-lead algorithm is applied and assessed in Chapter 6 and a variety of both causal and non-causal filters are applied in order to increases the overall performance. This algorithm is then implemented on the more complex plant and comparisons are made with the previous results. The use of a forgetting factor is found to overcome the problem of instability, but at the expense of increased final error.

In Chapter 7 multiple phase-leads are used in an attempt to address the stability deficiencies of phase-lead ILC. An optimisation routine is derived which generalises the process. Its success is confirmed with experimental results. In Chapter 8 a learning law which uses the plant adjoint is found to relate closely with this method and practical results are found to offer increased stability but lack convergence speed. This algorithm is then reformulated into one which needs no plant knowledge. Results are presented using this technique and practical guidelines are produced and tested to improve its performance. A simple method of increasing the learning at higher frequencies is proposed and practical limitations are addressed and verified experimentally. Chapter 9 examines a recent optimality based Repetitive Control (RC) algorithm, proposed in (Hatonen et al., 2003c), and it is rigorously tested. Its performance is compared with the previous algorithms. The robustness of all the algorithms used is examined in Chapter 10 using a novel graphical analysis method. Conclusions and Further work are laid out in Chapter 11 where the relative merits of all the algorithms implemented are discussed.

Chapter 2

A Review of Iterative Learning Control

This section establishes the general setting of Iterative Learning Control. It will be seen that research conducted in the field of ILC has mainly concentrated on either developing and analysing new algorithms or with integrating an ILC approach with established control techniques in order to improve system performance. Practical applications of ILC have not yet been widely published. Papers that deal with the integration of Adaptive, Robust, Optimal and Neural techniques and ILC are summarised in later sections of this chapter, and it should be noted that several exist for Direct Learning Control but are not included as this approach does not use repeated learning.

2.1 Core ILC Algorithms

The concept of Iterative Learning Control was arguably first formally proposed by Arimoto, Miyazaki and Kawamura (Arimoto et al., 1984a). In it position feedback is used to update the present cycle to produce the so-called proportional (P-type) ILC algorithm,

$$u_{k+1}(t) = u_k(t) + \Gamma e_k(t) \tag{2.1}$$

in which

$$e_k(t) = y_d(t) - y_k(t) \tag{2.2}$$

and $u_k(t)$ is the control input on k^{th} trial, $e_k(t)$ is the error, $y_d(t)$ is the desired plant output, and $y_k(t)$ is the actual plant output. Convergence is proved for a class of both Linear Time-Invariant (LTI) and non-linear systems. A key observation is that the assumption that the output tracking task is realisable implies the plant is minimum phase. This presents problems in choosing the gain and perhaps precipitates the use of other design methods. The proof has been further extended to encompass a class of

Linear Time-Varying systems (Arimoto et al., 1984b) and manipulators (Arimoto et al., 1985c). A coefficient test has been used to provide linear systems with limits on the gain necessary for stability in (Judd et al., 1991). The derivative (D-type) algorithm (Arimoto et al., 1985a) uses the velocity error and is given by

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) \tag{2.3}$$

in which $\dot{e}_k(t)$ is the derivative of the error on the k^{th} trial with respect to t. This has been examined both on its own and also when combined with the P-type law in the same algorithm. Acceleration error is included along with velocity error in the update law (Arimoto et al., 1985b), which is proven for a class of Linear Time-Varying systems and manipulators. The problem that occurs when null 1^{st} Markov parameters ('irregular plants') make the P-type controller unusable since the positional error is not available at the first sampling instant has been addressed (Porter and Mohammed, 1990a,b). A continuous velocity and acceleration error algorithm with respective 'initial state shifting' and 'initial impulsive action' is able to compensate for the lack of information during the first sample. These assume the 2^{nd} Markov parameter is full rank (the plant has relative degree 2). The same algorithms have been applied to multivariable plants (Porter and Mohammed, 1991a,b) but extended to $l^{th} order$ rank defective Markov parameters.

Discrete versions of the algorithms have been designed directly (Porter and Mohammed, 1992a,c,b) for linear multivariable systems. The first is a simple discrete time law,

$$u_{k+1}(i) = u_k(i) + \Lambda\{e_k(i+1) - e_k(i)\} \tag{2.4}$$

where $u_k(i)$ is the control input on the k^{th} trial at the i^{th} sampling instant and $e_k(i)$ is the error on the k^th trial at i^{th} sampling instant, and Λ is a $l \times p$ constant matrix, where l and p are the number of plant inputs and outputs respectively. The algorithm can be thought of as the discrete conversion of the D-type law.The discrete version of the P-type law is

$$u_{k+1}(i) = u_k(i) + \Gamma e_k(i) \tag{2.5}$$

Consideration of a LTI plant shows that the optimum parameter can be obtained with just knowledge of the step response. The learning rates are found to decrease rapidly as the order of the irregularity of the plant increases. It is asserted that laws similar to Equations 2.1 and 2.3 should be applied to irregular plants by generalising them with compensators of the form $(Im+Ds)$ used as pre-filters. Therefore the algorithm has been modified with two extra recursive equations which involve error from the current and previous cycles at all sampling instants. Only first order linear systems are considered although the compensator can be modified for use with higher order systems. The pre-filter compensator is found to improve convergence rates.

Examination of how it is appropriate to use position error for systems with a direct transmission term ('d') has clarified the use of P-type and D-type ILC (Sugie and Ono,

1987, 1991). The need for velocity error for systems with relative degree of 1, the applicable classes, necessary conditions, and a class of nonlinear systems have been considered and it has been shown that the derivative of the error used in the update must equal the relative degree of the system in order to remove the delay. In the discrete case, a β step delay is required, using

$$e_k(i) = y_d(i+\beta) - y_k(i+\beta) \tag{2.6}$$

in Equation 2.5, where β is equal to the relative degree of the plant. This proof has been extended to single-input, single-output (SISO) non-linear systems of relative degree greater than one (with linear systems as a special case), and also multi-input, multi-output (MIMO) non-linear continuous time systems (Ahn and Choi, 1990; Ahn et al., 1993) with earlier results as a special case (Porter and Mohammed, 1991a). Expansion of this leads to its discrete-time implementation (Jang et al., 1994) in which a very general use of the relative degree is given for non-linear systems. The algorithm is made to be high order and a feedback controller is added, no assistance is provided for choosing the gains. (Note that the term 'high order' refers to the use of more than one previous trial, not to the order of error derivatives). A feedback controller has been added to significantly reduce transient initial errors (Jang et al., 1995), the analysis concentrates on non-linear systems with linear systems forming a subset. No restriction is made on the structure of the feedback controller.

The usefulness of current cycle information is encorporated in a discrete-time algorithm using D-type information and the present output at the present time in (Ma et al., 1993). An important note is that conversion from continuous-time to discrete-time, via the sampling theorem (to find a suitable sampling time), makes it difficult to still ensure the law's effectiveness. Current cycle information can again be used by the addition of a linear feedback term explicitly to the feedforward signal (Kuc et al., 1992). This scheme actually uses a cost function term added to the feedforward algorithm but produces an update consisting of a gain multiplied by the last plant input. Analysis is applied to non-linear continuous time systems with the possibility of varying parameters, and emphasis has been put on the need for a stable closed-loop system. The concept of high-gain feedback also uses current cycle information but with no errors from previous cycles (Owens, 1992). The algorithm is given by

$$u_{k+1}(t) = u_k(t) + (Ke_{k+1})(t) \tag{2.7}$$

This has been used with linear MIMO, relative degree one, minimum-phase systems, and the case where $(Ke_{k+1})(t) = Ke_{k+1}(t)$ has been investigated. A parameter estimator is used in the iterative sequence domain (Oh et al., 1988) for linear periodic systems. In this case a Recursive Least Means Squares (RLMS) method is applied, and the parameters

are used to produce an inverse model of the plant that is realised using the algorithm

$$u_{k+1}(t) = u_k(t) + \tilde{B}_k^+(t)[\dot{e}_k(t) - \tilde{A}_k(t)e_k(t)] \tag{2.8}$$

where $\tilde{A}$ is the estimated A matrix and $\tilde{B}^+$ is the generalised inverse of the estimated B matrix. Here u_k and e_k are vectors of length equal to the number of states. This system was found to be sensitive to perturbations in the plant model. The idea, however, is extended to discrete-time linear and non-linear systems (Oh et al., 1991) and the assumptions relaxed. The method again uses time-varying gains in what is essentially a combined P-type and D-type law. The controller gain term is determined adaptively at each iteration, and can deal with unknown system parameters. Emphasis has been placed on the need for the error instead of the error derivative for non-linear uncertain systems with a direct transmission term (Xu, 1997).

A high order equation has been proposed in the form of the P-type algorithm that can use data from all the preceding cycles (Bien and Huh, 1989). The second order case is given by

$$u_{k+1}(t) = P_1 u_k(t) + P_2 u_{k-1} + Q_1 e_k(t) + Q_2 e_{k-1}(t) \tag{2.9}$$

where P_1, P_2, Q_1, Q_2 are $l \times p$ constant matrices, where l and p are the number of plant inputs and outputs respectivey. This has been used on continuous LTI and non-linear systems and it is found to be faster than usual and certainly less prone to disturbances. The method has been extended to use N previous cycle values of both P-type and D-type update terms, encorporating a parameter to stabilise it at the beginning of each cycle (Chen et al., 1998a). Non-linear systems have been considered and it is found that a state delay affects convergence only slightly, and if initial state error and disturbances are bounded, then so too is the final error. The λ-norm is used in the proof of this method, a tool that has been proven to be unreliable with reference to continuous-time time-varying non-linear systems (Tayebi and Zaremba, 1999). A combined D-type and P-type law is addressed which has time-varying gains, and the infinity norm is used to produce a reliable convergence condition.

Use of data from more than one previous trial has been investigated, this time using all the previous inputs and P-type errors but with time-varying gains (Chien, 1998). A stabilising controller is used on discrete non-linear time-varying plants with disturbances, and its input is added to the feedforward (ILC) input in order to improve the convergence time. The stabiliser is found to make the overall controller more robust to plant uncertainty. This is an extension of another discrete P-type controller which uses all previous inputs and derivatives (Chien, 1996). In its derivation the same class of systems is considered, although MIMO systems are included and the same stabilising controller is used.

A second order updating formula has been produced (using data from the past two trials) that is capable of offering advantages over its first order equivalent (Norrlof and

Gunnarsson, 1999). The algorithm is given by

$$u_{k+1}(t) = Q(q)(u_k(t) + L_1(q)e_k(t) + L_2(q)e_{k-1}(t)) \tag{2.10}$$

Since 'q' is a delay operator such that $q^{-1}u(t) = u(t - t_s)$, where t_s is the sample time, the parameters Q, L_1 and L_2 are transfer functions operating on the error. Therefore this law is more general than the ones in which only static or time-varying gains were used in conjunction with the error at small number of points, but includes the static gain case as a subset. A very general representation of an iterative system is considered, and analysis is conducted in the frequency-domain. This uses the notion of iterative systems bounds, and stability bounds are given for the filters. An example shows how a second order ILC can stabilize an unstable first order ILC algorithm (in which $L_2 = 0$). This is applied to higher order systems and is expanded to address the transient behaviour and the selection of the second order filters (Norrlof, 2000). For linear iterative systems the first and second order algorithms are compared in terms of stability and convergence, and a simple design proposal is given. It is not possible to conclude that the higher order ILC is more successful than the first order. The disturbance aspects of the first order law have been considered using a linear framework in the frequency domain (Norrlof and Gunnarsson, 2001). The general linear system description is used, but including load and measurement disturbance. The choices of filters in the algorithm are analysed from this viewpoint.

The concept of Current Iteration Tracking Error (CITE) uses the algorithm

$$u_{k+1}(i) = u_k(i) + Q_0(i)e_{k+1}(i+1) + Q_1(i)e_k(i+1) \tag{2.11}$$

and has been formulated for discrete-time systems (Chen et al., 1996b). It is an extension of previous results which also use current cycle information (Kuc et al., 1992; Owens, 1992), and is analysed using non-linear uncertain systems with disturbances. In this law $Q_0(i)$, $Q_1(i)$ are time-varying gains for the i^{th} sampling instant. The term $e_{i+1}(i+1)$ is approximated using the error derivative, and it is found that $Q_0(i)$ influences the final error bound. The use of CITE can be extended to a class of uncertain non-linear systems (Chen et al., 1997). The existence of uncertainties, initialisation error, and disturbances have been considered. In this case the algorithm uses P-type errors from N previous trials and the present one. It is asserted that it is not sufficient to ensure only the boundedness of the final error. The current error has also been used for non-linear time-varying systems but with no previous errors, and for the current sampling instant (Chien and Liu, 1996). This produces a P-type law which includes the current error and uses a forgetting factor, $1 > \beta > 0$

$$u_{k+1}(t) = (1 - \beta)u_k(t) + K(e_k(t)) \tag{2.12}$$

where K is a $l \times p$ constant matrix, where l and p are the number of plant inputs and

outputs respectivey. A forgetting factor has also been used in a practical application and an inherent compromise between tracking ability and stability is confirmed in practice (Lewin, 1999). A novel way of increasing its stability is formulated and applied.

Convergence and robustness of discrete-time non-linear systems has been analysed in a framework which incorporates disturbances and uncertainties (Wang, 1998). A P-type discrete algorithm with time-varying gain is taken and it is found that with no disturbances the error tends to zero, and with bounded disturbances there are bounded error limits. Anticipatory ILC (Wang, 1999) is a formulation in which the error-prone D-type method is exchanged for a direct time-delay, that is

$$u_{k+1}(i) = u_k(i) + L(.)(y_d(t+\Delta) - y_k(t+\Delta)) \tag{2.13}$$

with $\Delta > 0$ being a small increment of time. The example given has $L(.)$ is equal to a constant. In the analysis continuous non-linear systems are taken and the algorithm is implemented in discrete-time. A similar method has been used which does not limit Δ to small values, finding the system's delay from shifting the system input until the best match with the output is reached (Barton et al., 2000) which also uses a constant gain. This is an important change to the previous formulations. A reproposal of Anticipatory ILC includes saturation of the control device (Wang, 2000). Here continuous-time non-linear systems are looked at, and the laws robustness is proven for bounded disturbances and measurement error. It is then shown how a zero-order hold can be used to discretise this for application in discrete-time (Wang and Sun, 2001b). The previous result can be expanded to an arbitrary relative degree by selection of the delay time, but it appears that this must be limited for satisfactory characteristics (Wang and Sun, 2001a).

An N-order PID-type algorithm has been taken and robustified against initial state error (Chen et al., 1992) for use with delayed non-linear time-varying MIMO systems. It is stated that disturbances can either be controlled or filtered, that the effect of the delay is very small, but that the initial state error is important. This method 'sweeps' each trial forwards and backwards at the end of each trial until the state error is found. he gain is chosen in order to satisfy an inequality.

The bound of non-zero initial error has been found to influence the final error bound (Lee and Bien, 1996). The combined P-type and D-type law is taken and it is found that the gain on the P-type term can asymptotically reduce it, and an optimal value is predicted. The results have been generalised to the PID-type ILC algorithm and showed that the performance can be increased by adding an integral term (Park and Bien, 1999). Examination of the same algorithm (Park and Bien, 2000) takes both constant and random initial error for both linear and non-linear systems. The algorithm

$$u_{k+1}(t) = u_k(t) + \Gamma(L[\dot{e}_k(t) + P(e_k(\cdot))(t))] \tag{2.14}$$

where P is an operator of the error function $e_k(t)$, is proposed for linear, finite-time convergence. A similar problem is detailed (Chen et al., 1996a,c) which combines 'Initial State Learning' (ISL) with CITE. ISL updates the initial state with a measure of the previous cycle's, the P-type error multiplied by a time-varying gain, and the desired output at that point. This algorithm is very similar the original CITE algorithm (Chen et al., 1996b). Using these laws on a non-linear time-varying system with uncertainty and disturbance, relationships for the gain matrices can be given and the desired response achieved. The system must be relative degree of one, but unknown state time-delays do not affect the results. This theme has been continued (Chen et al., 1999b), again dealing with both linear and non-linear time-varying uncertain systems. Slight changes are made to the previous CITE and ISL algorithms and gains chosen accordingly. Mention is made of initial state shifting and initial impulsive action (Porter and Mohammed, 1991a,b). It is noted that this method only results in the initial state reaching a desired region, not a state (Jiang and Unbehauen, 1999). In an attempt to correct this, a controller based on a modified equivalent error with dead zone can be introduced. The dead zone's width is then adjusted on-line, its initial width including the initial state and it is then decreased every cycle. This method basically removes the problem of having an initial error which tends to undermine the learning process.

A discrete-time controller for SISO plants is presented (Hillenbrand and Pandit, 2000) in which the sampling rate is first reduced to produce a relative degree of one system. A discrete version of the D-type algorithm with a constant gain is proposed, and the controller must find the gain and sampling rate necessary to fulfil a condition on the norm of the system matrix L, where $y = Lu$ and u and y are vectors consisting of the signals at all the time instants. To deal with initial state error it is found that increasing the sample rate and the gain for just the first sample improves matters.

It has been shown (Sogo and Adachi, 1996) that the general update algorithm

$$u_{k+1} = u_k + T(e_k) \tag{2.15}$$

where T is an operator and u_k and e_k are input and error functions, cannot provide the exponentially decreasing property necessary for robustness when applied to a certain class of linear systems. The algorithm is taken in its digital form, as a regularisation method and this is shown to work in exchange for the residual caused necessarily by the digital controller. The residual approaches zero as the sampling interval tends to zero.

2.1.1 Time Delay

State delays for continuous uncertain non-linear systems have been examined (Chen et al., 1998a) using the PID-type ILC algorithm. For bounded initial state error this law

is shown to converge to a residual ball centred on the origin. This paper is separated from other work on the subject by the explicit presence of a delay in its states.

ILC for linear systems with a time delay has been investigated (Park et al., 1998) since an erratic estimation of delay time may cause the control system to diverge. A holding mechanism is adopted in order to keep the control input at a constant value for the duration of the time delay uncertainty. Consequently the system output tracks a given desired trajectory at the discrete points, which are spaced by the size of the uncertainty. This technique is examined for the algorithm

$$u_{k+1}(i) = u_k(i) + \Lambda e_k(i + \xi) \tag{2.16}$$

but can be used on many others, with satisfactory results. An application that calculates the system delay online and uses it in place of ξ has been used on an industrial conveyor (Barton et al., 2000). It is concluded that resulting instability may possibly be due to inaccuracies in the delay time.

An application for batch processes has been formulated (Xu et al., 2001), using a Smith Predictor to remove the time delay for uncertain SISO processes. The plant must be represented by a transfer function and a deadtime and an ILC algorithm is used which includes terms for the actual and estimated process and deadtime. Three alternatives are investigated; the case where the transfer function is known, where it is unknown, and where the upper bound is known. Conditions are given for each.

2.1.2 Two Dimensional Modelling

A class of iterative learning controllers for discrete linear systems has been analysed and a 2-D model established in the form of 'Roessor's Model' (Geng et al., 1990). ILC is one of the few applications of 2-D system theory. The combined P-type and D-type controller is examined in the discrete-time domain, using 'i'to denote the sample number and 'j' the iteration number

$$u_(i, j + 1) = u(i, j) + K_1 e(i + 1, j) + K_0 e(i, j) \tag{2.17}$$

An estimator is used at the end of each cycle and a model of the plant produced in order to calculate at the two gains. This is different to adaptive control which usually works simultaneously with the algorithm. The intricacies of this are given separately, but the main focus really lies in the analysis afforded by the 2-D technique. 2-D analysis has been extended to cover n^{th} order linear discrete time systems, which can be multivariable as before (Kurek and Zaremba, 1993). The P-type law is examined and then an extension using both P-type and D-type updates, as well as estimated state information from the previous cycle. The system must have a relative degree of one, and the gains are given

assuming the explicit plant matrices are known. A delay can be included, and step-by-step application illustrates the way in which it is implementated. The original combined P-type and D-type law has also been extended with the addition of a present error term (Lee and Lee, 1993). This makes it less sensitive to disturbances, and its conditions are more easily satisfied. Like the original, an estimator is used, and the method is put into practice on a VCR servo system. 2-D system theory has again been used to derive a non-causal (in the time-domain) learning controller (Liang and Looze, 1993). A frequency domain approach is taken, but it is more analytic than practical.

The 2-D system has been extended to overcome the difficulty seen in (Kurek and Zaremba, 1993) of unknown convergence (Chow and Fang, 1998). Extension to discrete-time is achieved using 2-D continuous-discrete Roesser's linear model. The P-type algorithm, the P-type with present and past-cycle state information, and an algorithm similar to that of (Kurek and Zaremba, 1993) with a state feedback term are examined. Focus is dedicated specifically to this last algorithm (Fang and Chow, 1998), using the procedure,

$$u_{k+1}(t) = u_k(t) + K_1 e(t+1) + K_2(x(t) - \tilde{x}(t)) \tag{2.18}$$

where K_1 and K_2 are constant matrices of appropiate dimensions, $\tilde{x}(t)$ is the estimated plant state at time t. Note that $\tilde{x}$ is not obtained by using the next input, as it was previously (Kurek and Zaremba, 1993), but by using an estimator. Gains are found using given equations and the algorithm converges in 1 iteration. It is used here on linear discrete-time multivariable systems. A counterexample (Kurek, 2000) has been necessary to defend the original method against criticisms made against it.

2.1.3 Robotic Applications

A simple P-type algorithm using a filter has been applied to a robotic arm (Mita and Kato, 1985) and the conditions for convergence discussed. The effectiveness of the P-type law for the same application has been proved (Arimoto et al., 1988), but reaffirming that the inclusion of position error alone does not guarantee convergence. A non-linear analysis of ILC for robots has been presented, emphasising the need for high-gain feedback (Bondi et al., 1988).

A method specific to robotic manipulators, although not dissimilar to some other schemes, is proposed in (Kuc et al., 1991). A linear fixed-gain PD controller is used to provide stability via a feedback loop outside that of the ILC controller. The possibly unknown, time-varying, non-linear robotic equation is linearised along the desired joint trajectory, and the controller constructed using combined P-type and D-type error terms. An additional term is included which consists of previous-cycle error derivatives. This method has been found (Jiang et al., 1995b) to belong to a subset that could benefit from a scheme to linearise uncertain non-linear plants to which linearisation normally cannot

be applied. Lypunov's direct method is used to develop a sliding-mode which satisfies a linearisable control equivalent system. An optimal iterative law is then applied. An adaptive scheme has been formulated for use with the algorithm inorder to estimate the upper bound of the uncertainties when it is itself uncertain (Jiang et al., 1995a).

A set-point regulation problem has been introduced for manipulators with flexible links moving under gravity (Luca and Panzier, 1994). It acts by updating the torque input setpoint with a P-type term. Flexible joints have again been considered in another application which uses a combined P-type, D-type and acceleration error term controller with a forgetting factor to shift the weighting between the previous cycle input and a function of an initial guess (Wang, 1995). This theme has been expanded to cover the impedance control of robotic manipulators (Wang and Cheah, 1998). In this case only a target model is available, with no target trajectory defined. This results in force control being applied by the manipulator.

2.1.4 Auxiliary Methods

It has been noted that satisfying the commonly used λ-norm (Arimoto et al., 1984b), may leave room for the trajectory error to overshoot drastically, thereby making it an unreliable measure of convergence (Lee and Bien, 1997). This occurs because ILC is essentially open loop and it is possible to have a large error at the end of the cycle which then comprises a large portion of the updating input, which helps propogate the process. Although the error becomes larger at the end of the cycle, its λ-norm still decreases. This can be solved by making the gain exponentially decrease by defining the gain as

$$\Gamma(t) = e^{-\gamma t}\Gamma \tag{2.19}$$

A method of incorporating past experience has been presented which is applicable to most ILC algorithms (Arif et al., 2001). Here experience is used in the process of the selection of the initial control input for a new desired trajectory, the convergence can thus be improved without modifying the controller. Data about the system states, the output, and the corresponding control input for all iterations is stored in memory. The prediction of the control input for a new trajectory is achieved by dividing it into many query points, a linear model is fitted to each one to find a corresponding control input. A dense population of data will produce a good prediction. A detailed search procedure is included to help make this method effective.

2.1.5 Practical Applications

The control methods applied to systems in practice are generally applications of existing algorithms but include insight into the practical issues involved. A discrete PID-type

algorithm (in the iterative domain) has been applied with CNC machine tools (Kim and Kim, 1996) and is given by;

$$u_{k+1}(i) = u_k(i) + K_P e_k(i+1) + K_I \sum_{n=1}^{i+1} e_k(n) + K_D[e_k(i+1) - e_k(i)] \tag{2.20}$$

It is found that this works reasonably well despite uncertainty and disturbances. The summation includes all previous cycles as well as the present. Gains are chosen inorder to satisfy a given inequality. A far simpler scheme has been used with robotic systems (Poo et al., 1996). The hazardous conversion to discrete-time via the sampling theorem is brought to attention along with the fact that the determination of a gain matrix is difficult to achieve in practice. The method therefore uses a D-type term, a term comprising of the change in output from cycle to cycle, and a time-varying gain in its update law. The gains use approximations to the inertia matrix, and the system is tuned by reducing the sampling instant until the performance is satisfactory. Feedback-assisted learning control has been used for the heat-up phase of a reactor (Lee et al., 1996). This uses a standard PID controller command plus the error multiplied by the approximate plant inverse, that is $H(q^{-1})e(t)$, modelled as a first order lag. The presence of the 'q' operator means that the multiplication can be thought of as equivalent to convolution of the two signals. A filter is attached to remove high-frequency output noise. An interesting feature is that the reference trajectory is recalculated for each trial.

Ethanol concentration has been controlled in a batch process (Choi et al., 1996), using feedback-assisted learning control. The ILC part of the control is the general vector formulation

$$u_{k+1} = u_k + \frac{1}{G_p} e_k \tag{2.21}$$

with G_p the gain being an approximation of the objective system inverse. To model the non-linear time-delayed process all but the largest time constants are neglected in order to produce a first or second order system that also requires an extra deadtime. The First Order Plus Dead-Time (FOPDT) system parameters are found using Least Squares Estimation. (It is put in a certain form, the disturbances removed by linearisation and subtraction of successive sets, then converted into discrete form and the parameters selected to minimise the error). A standard feedback controller has been used in another application (Moon et al., 1996), again with its output fed to the ILC controller, the feedback controller getting no reference trajectory directly. The update is written in terms of the Laplace operator as

$$U_{k+1}(s) = U_k(s) + \Gamma(s) E_k(s) \tag{2.22}$$

which affords great flexibility in the form of the transfer function $\Gamma(s)$. One of the gain filters is chosen to equal zero above a certain frequency. The filters can be designed with only knowledge of the estimated plant upper and lower uncertainty bounds.

An application to coil-to-coil metal rolling machines is presented in (Garimella and Srinivasan, 1998), which uses a low-pass filter and approximate plant inverse. It is maintained that reduction of the plant's transfer function bandwidth improves robustness. A MIMO transfer function matrix is decoupled, each loop having its own PI controller. The ILC's gains are then selected with care. Control of human limbs has been achieved with use of a high order discrete D-type algorithm with time-varying gain whose control command is summed with a feedback controller's in order to produce the new plant input (Dou et al., 1999). Saturation is included for a realistic effect, although no help in gain selection is provided.

Control of a high-precision linear motor has been undertaken (Tan et al., 2001), again using a feedback controller whose output is the input to the ILC, a layout which has been used previously (Moon et al., 1996). This is the natural procedure when the ILC is located within the feedback mechanism. Alternatives layouts include both controllers being in parallel (each receiving the error signal) and their inputs summed, or the ILC located around both the plant and a feedback controller (indeed many papers assume that the 'plant' includes its own feedback controller). The ILC update used in this instance has the form

$$u_{k+1}(i) = u_k(i) + \gamma h'_M * u_k(i) \tag{2.23}$$

where $*$ denotes the discrete convolution. The notion of $\mathbb{Z}$-transform $Y(z) = H(z) \cdot X(z)$ where the upper case letters are the $\mathbb{Z}$-transforms of their lower case equivalents is used. It is shown that, if the filter is chosen to be zero-phase, then h'_M is simply a symmetric moving averager, and this is designed to produces an approximate plant inverse. The auto-tuning of the PID controller is also examined in the description of this application.

An overview of the practical issues involved with ILC implementation is given which goes some way to formulate a universal ILC controller (Longman, 2000). Linear, relative degree one systems, are examined and it is argued that all systems can be represented in this way with suitable choice of sampling time and linearisation applied if necessary. Only frequency response data is used and a selection of gain matrices, L, are considered for the vector update law

$$u_{k+1} = u_k + Le_k \tag{2.24}$$

An explanation is included of how to design cut-off filters for whichever structure of gain matrix is chosen. Each choice encountered in ILC design is examined to produce a thorough overview of the difficulties involved. The use of a simple algorithm is, however, a limiting factor so performance of a specific case is unavoidably compromised by the inclusion of a vast breadth of applicable classes. The close relationship between Repetitive Control (RC) and ILC is made clear.

2.2 Repetitive Control Approaches

Whilst in the framework of ILC a trial is defined to be of a set length, RC allows it to vary, its end occurring when a certain system state has been reached. This changes much of the analysis used with RC and seeks to divide the philosophies. It is also assumed that RC has no breaks betweens trials, while ILC can settle and perform calculations if necessary. RC was formulated using the Internal Model Principal (IMP) (Hara et al., 1985), which is a means of generating any repetitive signal, and is more often used in robust control to remove structured (periodic) uncertainties. An IMP model is located in the closed loop system and conditions presented for successful convergence of the linear and non-linear systems. Although there are too many papers on just ILC to contemplate a survey of repetitive control, there are many connections between the two approaches, and several methods exist which attempt to bridge the gap between the disciplines.

'No-reset ILC' is a mechanism allowing the implementation of RC methods for ILC by not ensuring the initial conditions are the same at the beginning of each cycle (Sison and Chong, 1996). The remaining ILC architecture is unchanged. A SISO LTI system has been examined (although MIMO systems are applicable), and a state equation produced in which the state includes both the state of the plant and the controller. The system 'A' matrix is found and decomposed in order to assume the same structure as an existing repetitive scheme that allows the placement of system eigenvalues. The gain matrix is set arbitrarily, a computationally extensive process since it has no structure.

It can be shown that for both ILC and RC process learnability is equivalent to output-dissipativity which is equivalent to strict positive realness, these definitions being defined in (Arimoto and Naniwa, 2000). The motivation behind this is to explain how a simple ILC scheme can work so effcetively with complicated non-linear systems.

ILC has been used to solve an repetitive problem in (Moore, 2000; Moore and Xu, 2000). RC 'trials' vary in length and there is a continuous horizon. Each trial is allowed to always return to its initial state by extending or reducing the reference, only allowing learning to occur where there are results in the trial before. The analysis concentrates on the high-order derivative algorithm and reformats it, and a simple and natural way to link the two techniques is established.

The two schemes are found not to be equivalent but are related in duality, a consequence of the difference in location of the Internal Model inside the compensator (Roover et al., 2000). A repetitive controller is realised by implementation of a robust servomechanism with appropriate internal model. ILC is implemented in the format of a disturbance observer/compensator - the dual of the repetitive controller. Therefore the design is generalised into a powerful analysis and design procedure of the internal model framework.

2.3 Robust Control Approaches

The extension of existing analysis techniques (i.e. (Liang and Looze, 1993)) for ILC in the H_∞ mathematical framework into the actual synthesis of an ILC controller is given in (Roover, 1996). It is shown how this process can be generalised to the synthesis of an H_∞ (sub)optimal controller. A feedback configuration with a general updating law (the P-type law) is used to generalise the process. To conduct the synthesis one of the filters is interpreted as a weighting function for learning performance, which can be simply a low pass filter, and the other is obtained by minimising a norm. This produces the highest L_2 convergence rate. To produce a workable solution it is reformulated into Standard Plant (SP) format, and the solution is obtained using two coupled Riccati equations. An important result is that uncertain system knowledge can be explicitly incorporated into the design procedure via weighting functions, turning it into a robust performance synthesis problem. These can be solved using the μ-synthesis (Structured Singular Value) approach, by performing D-K iteration. A practical design method for this past-error feedforward scheme has been given for the open-loop case (Roover and Bosgra, 2000).

A frequency-domain method for uncertain feedback control systems is proposed in (Moon et al., 1998). An LTI plant with multiplicative perturbations and an existing feedback controller is examined. It is shown that the design can be reformulated as a general Robust Control set-up. Here 'Linear Fractional Transformation' (LFT) is used to rewrite the plant transfer function. A simple ILC law is used to update the input,

$$U_{k+1}(s) = U_k(s)P(s) + Q(s)E_k(s) \tag{2.25}$$

the gain $P(s)$ is chosen as a low-pass filter, $Q(s)$ is found by solving a general robust control problem. This implies that the ILC law can be systematically designed by robust control techniques such as μ-synthesis. The lack of proper techniques to design an ILC for systems with model uncertainty is addressed in (Doh et al., 1999b,a). Many schemes assume the presence of an existing feedback controller, and this becomes more important if the model has uncertainty. A simple ILC algorithm with current feedback is given to illustrate this and the objective is not only to stabilise an uncertain system but also to guarantee the convergence of the ILC for model uncertainty. A sufficient condition for robust convergence and stability is found using μ-synthesis, and LFTs. A synthesis method is presented using D-K iteration (which reformulates it as an optimisation problem).

The combination of ILC and Internal Model Control (IMC) has been investigated for uncertain LTI systems (Tayebi and Zaremba, 2000) using the update given in Equation 2.25. The convergence is reformulated as a general robust problem. For a certain choice of IMC and ILC filters, the condition of convergence to zero is the robust performance condition of the IMC structure. A design procedure is proposed using the μ-synthesis

approach. For a simple ILC law applied to a feedback system with the internal model included, a design is proposed for the filters using the above method. Robust ILC (RILC) is presented in (Xu and Qu, 1998), and provides a general framework targeted at synthesizing learning and robust control methods using Lypunov's direct method, extending their use to more general classes of non-linear uncertain systems. It is used on systems with terms for known and unknown system parameters and structured and unstructured uncertainties. Robust control deals with the structured uncertainties and learning control handles the state-independent uncertainties, optimally eliminating them. Analysis focuses on a high-order MIMO non-linear system. Variable Structure Control (VSC) is again used as a robust control strategy in (Xu et al., 2000) which develops a scheme to address both periodic and non-periodic system uncertainties. Only the upper bounds of uncertainties are needed and the inverse model dynamics are used to separate the required efforts for the RC and ILC parts, partitioning the system uncertainties.

Adaptive Robust ILC (ARILC) has been formulated for non-linear uncertain systems (Xu and Viswanathan, 2000) and includes the presence of a dead-zone. The method ceases both learning and adaption when the error enters a specified bound. ARILC is the synthesis of both ILC and Adaptive Robust Control. ILC is applied to deal with the structured system uncertainties, and the adaptive robust strategy handles the non-periodic uncertainties associated with partially known bounding functions. The unknown upper parameters in these functions are estimated with adaption. Partitioning of the inverse system separates the type of uncertainty, and an update equation is given. The robust part is found by minimising the difference of a Lypunov function, the ILC part uses a high order P-type algorithm.

2.4 Optimal Approaches

System optimisation schemes have been applied to learning control (Togai and Yamano, 1985), and it is noticed that the P-type and D-type algorithms do not guarantee positional convergence. It is also noted that learning algorithms are in a sense asymptotically optimal since optimal values characterising the control actions are not obtained at once but after some time. A state observer and a reference model is used to obtain the error in the state variables, and this is used as the error in a D-type update. The gain term is chosen to minimise a cost function, and the Steepest Descent, Newton-Raphson and Gauss-Newton methods are all used to provide a solution.

A discrete ILC has been formulated directly (Ishihara et al., 1992), involving the Impulse Response (IR) of the system. A performance index is formulated where the gain minimises a quadratic performance index. To obtain a robust controller, a probablistic error model is icluded in the plant IR and an average quadratic performance criteria is minimised instead. It is suggested that this method requires a high order algorithm

in order to satisfy a commonly used convergence condition in (Hillenbrand and Pandit, 1999). Therefore it is shown that reducing the sampling rate can guarantee exponential convergence and robustness to uncertainties and initial state error for the same algorithm. It is reduced to give a relative degree of one, and for high convergence the system output must be sampled at a high frequency and system identification performed to ascertain the minimum sample time and the optimal gain.

A discrete-time algorithm has been proposed (Amann et al., 1995, 1996a) that is based on an optimisation principle for linear plants. For the continuous case, the minimisation of the linear quadratic performance criterion

$$J_{k+1} = \frac{1}{2}\int_0^T e_{k+1}^T(t)Qe_{k+1}(t) + [u_{k+1}^T(t) - u_k^T(t)]R[u_{k+1}(t) - u_k(t)]dt + \frac{1}{2}e_{k+1}^T(T)Fe_{k+1}(T) \tag{2.26}$$

with gain matrices Q, R and F. It has the non-causal solution

$$u_{k+1} = u_k + G^* e_{k+1} \tag{2.27}$$

This is transformed into a co-state system and this non-causal representation is in turn transformed into a causal algorithm, using a state-feedback system representation. The problem of finding an input to produce the desired output is a singular optimal control problem, which requies an iterative solution. The reduction of the error norm at each step, the automatic choice of step size, and use of current data are all desirable properties that are fulfilled. The solution to the Riccati equation is required and the use of current trial full state feedback along with feedforward of the previous trial tracking error. For a time-invariant system the Riccati equation needs only to be solved once. The rate of decrease in the error can be influenced in an intuitive way by several design parameters. This method has been reformulated in continuous time to include control and state relaxation parameters which of course change the update algorithm (Amann et al., 1996b). The update algorithms are presented in a slightly different form, and simulations are given for a linear time-varying plant and a non-linear pendulum to illustrate its performance and robustness. An extension of this method has been derived in order to address the problem of the performance depending on algorithmic parameters such as plant dynamic structure (Amann et al., 1998). For example, plant characteristics such as being non-minimum phase cause a slow rate of convergence. A predictive horizon is used in the computation of the control input (similar to Model/General Predictive Control) and a set of finite-time Linear Quadratic Regulator (LQR) problems is solved inbetween trials. The input increment at each trial must minimise a performance criterion consisting of the sum of future errors and control increments. It converges faster than previous algorithms with less foresight. It is shown that the convergent rate is independent of the plant dynamics. This predictive algorithm can achieve geometric convergence, the

infinite-dimensional limit operator being approximated by a finite-dimensional controller for practical purposes.

It has been shown that the method by (Amann et al., 1995, 1996a) could be implemented as an output feedback algorithm (Lee et al., 2000), thus improving the robustness. A framework (termed Quadratic-ILC (Q-ILC)) is developed to deal with constraints, non-linear dynamics, and disturbances. New algorithms are presented which consider these effects. An observer-based algorithm is given as an alternative the direct-error based algorithm to deal with disturbance. A cost function for constrained systems is given, apparently a standard quadratic programming problem, and also one that includes model uncertainty information which is a convex programming problem. The convergence is proved for these algorithms. The lack of guidelines for the tuning factors has been addressed (Kim et al., 2000), and the robustness is examined further. It is shown how Singular Value Decomposition (SVD) can be used to tune the gains, and a reduced-order algorithm is given which enhances the robustness and lessens the computational load. The robustness of norm optimal based ILC schemes in practical applications has been examined (Al-Towaim et al., 2001), concentating on the proposal of (Amann et al., 1996b) whilst describing a test facility used for experimental evaluation (as (Barton et al., 2000)).

Generalised Predictive Control (GPC) with learning is presented in (Bone, 1995). The method learns the repeatable portion of the disturbances, improving the accuracy of the predicted future response. It is based on Controlled Auto-Regressive and Integrated Moving Average (CARIMA) and the plant must be linear, or linearisable about a set point, as it uses a linear plant model. This method is found to significantly improve the performance of GPC.

An optimisation technique has been derived in the frequency domain (Gunnarsson and Norrlof, 1999) which is applicable to linear, SISO, discrete-time systems. It is given by

$$u_{k+1}(t) = Q(q)(u_k(t) + L(q)e_k(t)) \tag{2.28}$$

the filters $Q(q)$ and $L(q)$ are chosen by establishing and minimising a quadratic cost criterion.

Another technique (Gorinevsky, 1999) has been motivated by development of control methods for sampled, spatially invariant, distributed systems, to which an LTI plant can be said to belong. Development of a unified view of ILC that encompasses operator, linear-quadratic, model-predictive, regularisation, and frequency domain design is desired. A simple update is given, and a quadratic performance index is constructed. Bounds on the cost parameters are set out depending on whether robust convergence, quantified actuator magnitude, or nominal performance is deemed most important. Analysis is performed on a SISO linear system and the point is made that frequency

analysis performs design and analysis in a decoupled way since ILC control does not have to depend on the measurement in a causal way.

Linear time-varying systems have been examined in (Phan and Frueh, 1996), using basis functions to define an operating input-output space by utilising data gathered over repeated trials. Inverse control is employed to give the correct solution. By increasing the number of basis functions used, the strategy can take care of the gross motion first, and then the details at increasing levels of accuracy. The scheme can learn rapidly provided that the underlying input-output relationship is captured. Basis functions have again been used (Phan and Frueh, 1999), along with a reference model, to guide the learning process. The basis functions reduce the dimension of the system model. A discrete, linear time-varying system is examined, and input and output data is put in the reference model as it is found. An algorithm governs the update of the model, inputting data in the form of basis functions. Another algorithm minimises the output prediction error using this information. This is an ILC application of 'Model Reference Adaptive Control', and could also be classed as an adaptive design, but forms a link with previous research (Frueh and Phan, 2000), in which a solution to minimising a quadratic cost to achieve optimisation without requiring detailed knowledge of the system (assuming the plant is linear time-varying or LTI) is given. The optimal input is found for the cost function in terms of the plant, and then basis functions are used to reduce the number of experiments required. The optimal coefficients for the basis functions are found, and each newly added basis function does not alter the previously optimised ones. They are then used to modify the control action for the current trial. Implementation steps are laid out and a simulation is presented where the error cost matrix is altered so that only certain portions of the tracking error are weighted, illustrating its utility.

The use of basis functions is also examined in (Hamamoto and Sugie, 2001) and it is found that, since few systems have bi-properness and only a few have input passivity, most ILC methods adopt compensators for the system in the form of error derivatives, dual mapping (i.e. (Roover, 1996; Amann et al., 1995)), and low pass filters. Error derivatives are known to be error-prone, and the others require a large amount of system information. Linear continuous systems are examined, and the input space (usually infinite space L_2) is restricted and spanned with basis functions. A general update law is applied in this restricted space, using the difference between the optimal and previous parameter vectors as the error. The two filters are considered taking into account the uncertainty of the restricted plant. The input space is constructed to include the optimal input.

2.4.1 Applications

The application of ILC to an Optimal Tracking Problem has been conducted (Chen et al., 1998b,c) which requires the extraction of Aerobomb drag. A cost function is

derived, but the problem is not the usual one, as an unknown non-linear function must be extracted iteratively from the output data, which is taken as the desired trajectory. The system is non-linear and time-varying, and the D-type law with a time-varying gain is used to update the drag coefficient, the error found by integration. This method is extended through the use of a higher order ILC algorithm.

Iterative optimal control has been applied to the well-established problem of slosh in an industrial packaging machine (Grundelius, 2000; Grundelius and Bernhardsson, 2000). The problem is to find an open loop acceleration reference. A quadratic optimisation expression is given and the resulting procedure uses quadratic programming to find the update term, δU, only once. This is then used in the equation

$$U_{k+1} = U_k + \delta U_k \tag{2.29}$$

to update the acceleration reference. An extension of this again includes an optimisation problem, but it is discretised and the solution procedure uses quadratic programming continually. The previous cycle data updates terms in the minimisation problem.

An application of optimisation to the control of extruders has been detailed (Pandit and Buchheit, 1999) in which a performance index is presented which is required to be minimised every cycle. Calculus of variations is applied to this task, and either steepest descent, the Newton method, or another iterative algorithm is used to obtain the new control input. The procedure is complicated by the need for the plant operator to be calculated every cycle for use in the optimisation procedure.

2.5 Adaptive Techniques

Results have been presented on adaptive ILC for both linear and non-linear plants in (French et al., 1999). Universal Adaptive Stabilisation (UAS) is examined on a linear plant consisting of a proportional controller with adaptation on the gain. This is capable of stabilising a system from any state initial condition with very limited knowledge of the plant. It is converted to an ILC algorithm with several alternative laws, the linear system examined is restricted to minimum phase and not proved for a relative degree greater than one. These restrictions are dropped by considering a standard adaptive controller (tuning functions with K-filters), however the system must be LTI with known relative degree. The resulting ILC scheme learns along each pass length instead of relying on iterative updates at the end. The parameter estimates that the adaptive scheme produces are initialised at the start of each pass to the final value of the previous pass. The results form a theoretical base from which to develop implementable adaptive schemes.

An ILC algorithm for plants with measurement noise, based on an interpretation of the signal flow is presented in (Moore, 1999). This method for discrete-time LTI systems uses a deterministic, adaptive (in trial number) gain adjustment technique and is an extension of previous results (Moore, 1998), but with the inclusion of noise which makes the on-line estimation of Markov parameters necessary. A discrete D-type law is applied and the multi-loop perspective results from writing out the first few terms of the resulting control system. It is clear that the controller gain should be the inverse of the first Markov parameter, hence its estimation using a standard LMS procedure. Additional estimates can also be included in a feedforward mechanism.

Two previous formulations are brought together in (Avrachenkov et al., 1999). The first concerns learning using an approximation to the system performing multiple periods of a periodic task, and the second concerns a learning control scheme using definitions in Hilbert Space. This paper derives an ILC scheme for continuous systems using Broyden Space and the system is described by an operator in Hilbert Space. The update algorithm is given by

$$u_{k+1} = u_k - P_k^{-1} e_k \tag{2.30}$$

where P_k^{-1} is an approximation of the linear part of the system near the trajectory induced by the current control input as its gain. Broyden's update produces the next approximation operator if the system is not linear, if it is linear, a much simpler update can be used.

A stable-inversion method of learning controller is extended (Ghosh and Paden, 1999) to accommodate a class of non-linear, non-minimum phase plants with disturbance and initialisation errors. This extension requires the computation of the approximate inverse of the linearised plant rather than the exact inverse, thereby not requiring the output differential. The system is first linearised using a linear operator. An 'approximate inverse' learning operator is defined which is the pseudo inverse of the last operator, and is obtained using the stable-noncausal-solution approach. An update law is given using these two operators. With bounded disturbances and initiation error the error converges to a neighbourhood of zero, and the method can be extended to deal with uncertainties. Stable-inversion has been used (Ghosh and Paden, 2001) for the same type of plants but designed for relative degrees greater than one at the cost of differentiating the system output. Linear plants are first examined and the inverse is found easily if the system is minimum phase and 'stable inversion' is applied if it is not by decoupling it into a stable and a non-stable sub-system. This is then robustified by multiplying the pseudo-inverse by $e(1 + \frac{d}{dt})^r$ instead of by $e(\frac{d}{dt})^r$ in the update algorithm. This latter equation is also used for the non-linear plant having applied linearisation to the minimum phase case in the traditional way and to the non-minimum phase case by a given technique. It is noticed that no simple and direct iterative method exists for applying this stable inversion to non-minimum phase systems (Sogo et al., 2000). To this end the problem is formulised as a minimisation problem. The Hamiltonian function is given and also

the gradient function which has a gradient which satisfies a given adjoint system. An iterative method is proposed which involves integrating the system in the forward-time direction, integrating the adjoint system in the backward-time direction by using the output error and the state trajectory, and having an updating law of the state-space 'C' matrix multiplied by the gradient, multiplied by a gain. Convergence is established in the presence of uncertainties for linear systems.

An adaptive scheme is extended to encorporate an ILC update for non-linear systems in (Seo et al., 1999). The controller consists of three parts: (1) A feedback controller to stabilise the overall closed-loop system. (2) Uncertain non-linear compensation terms are approximated as linearly parameterised forms, and compensated for by using intelligent systems. It uses basis functions represented by either fuzzy logic or neural networks. The result is to keep the feedback gains small. (3) A Learning controller achieves precise tracking by updating the control input. The intelligent control parameters are also iteratively updated. The resulting controllers are unfortunately rather complex. Another scheme has been proposed (Choi and Lee, 2000) in which the parameter estimates are updated in the time domain instead of the iterative, and repetitive disturbances are identified and compensated for in the iterative domain. The controller consists of a feedback term to stabilise the system, a feedforward term which uses the estimated system parameters to compensate for the non-linear part of the system, and a simple learning law in the iterative domain. The parameter adaptation law is in the time domain. Existing adaptive algorithms can be used, and the learning gain can be adjusted independently of the parameter adaptation gain. The adaptive schemes proposed in (French and Rogers, 2000) also update the parameter estimates in the time-domain for uncertain non-linear systems (including non-minimum phase). It is shown how adaptive schemes can be modified to give a solution to either the feedback of feedforward ILC problem. Although concentration is primarily directed on theoretical issues, it is expected, with robust modifications, that the designs will work well in any iterative setting within domains where adaptive control has traditional success.

A 'universal' adaptive scheme (UAS), which involves an adaptive learning gain and a high gain feedback gain feedback result is proposed in (Owens and Munde, 2000). UAS is again used to produce a non-linear control algorithm consisting of a proportional controller with time-varying adaptation on the gain, capable of stabilising any system. It works from any initial choice of controller parameter and plant state. Algorithms for transferring it into the Adaptive ILC context are given; a simple CITE law for updating the control input and an iterative algorithm for updating the adaptive gain. This method is applied to a linear SISO plant with a relative degree of one, and the potential for achieving convergence under conditions of extreme uncertainty is demonstrated. The assumption of minimum phase systems is required to guarantee the feedback gain is able to be ultimately stabilising, and the error convergence to be monotonic.

Iterative adaptive learning has been used for injection moulding machines in (Havlicsek and Alleyne, 1999). A simple model is developed of the plant and a feedback controller is designed using standard linear techniques to stabilise the plant. A simple iterative learning law of the CITE type is used, the feedforward gain being a symmetric moving average filter operating every cycle.

2.6 Specialist System Types

Constrained non-linear systems are explicitly dealt with in (Chen et al., 1999a), which include manipulators as a subset. The learning law is a high updating scheme in the continuous-time domain, obtained by parametrising the control profile with a piecewise continuous functional basis. A sufficient condition for the convergence is given and the bound on the asymptotic error is due to the constraint condition and its Lie derivative.

Non-holonomic constraints for robotic systems have been examined (Oriolo et al., 2000), and a robust steering scheme proposed for systems that can be put in chained form, a canonical structure for non-holonomic systems. By over-parametrising the control law other goals can also be met. The system is put in chained form and a control law is given to steer the system between two points to produce 'natural' robot trajectories. To achieve robustness, however, its control parameters are computed through a learning scheme, improving at the same time a given performance criterion. The overparametrization provides space for the optimisation process.

Another special type of ILC problem has been examined (Xu et al., 1999) with particular reference to Rapid Thermal Processing. The problem is due to only the terminal tracking error, instead of the whole output trajectory tracking error, being available. A revised ILC method is proposed to address this; the control profile is parameterised with a piecewise continuous functional basis and the parameters are updated by the simple updating scheme which involves just the terminal error

$$\Xi_{k+1} = \Xi_k + L_1 e_k(N) \tag{2.31}$$

where L_1 is a learning matrix and Ξ is a vector of control parameters. the input is given by $u_k(t) = \Phi(t)\Xi_k$ A convergence condition is obtained for a class of uncertain discrete-time time-varying linear systems. The scheme involves linearisation about an equilibrium point.

Large scale linear systems are tackled in (Hwang et al., 1993) which take the form of connected subsystems. The system is decentralised by applying controllers on each one independently. The update is the same as that in Equation 2.8 and used with an estimator providing the states.

2.7 Neural-based ILC

Neural Networks have been used with ILC to produce a method for unknown non-linear systems (Chapter 12 (Bien et al., 1998)). It is noted that the human brain possesses long and short-term memory, and this provides a basis for a gain estimator and three controller scheme. (1) A feedback controller (FBC) stabilises the system at the initial stage before learning begins. (2) A short term memory-based ILC makes the system output converge exactly to the desired trajectory. (3) After iterative learning, the control information is accumulated by a trained Feedforward Neuro Controller (FNC) for generalisation to any trajectory tracking. The Learning Gain Estimator (LGE) estimates the learning gain of the ILC from input/output data and relative degree of the system. Both the FNC and the LGE use a piecewise linearly trained neuro network, referred to as SOLP (self-organising local perceptions). The procedure is as follows. (A) Design a feedback controller to stabilise the system. (B) Identify the system using a neural network by presenting persistently exciting inputs. Estimate the learning gain for the ILC. (C) Choose the desired trajectory. (D) Apply the summed FNC, FBC, and ILC outputs. (E) Transfer the ILC output to the FNC using a given training procedure. (F) Repeat steps (D) - (F). The ILC algorithm used is a high-derivative D-type law, and the scheme works for unknown (apart from relative degree) non-linear systems. It also exhibits rapid convergence and guarantees convergence to the desired trajectory unlike the pure 'neuro' control.

2.8 Summary

From early developments in the field of ILC, the technique has been extended to cover a great number of system classes and utilise a large number of control techniques. It can be seen that there exists a lack of experimentally-based research concerned with the testing and verification of algorithm performance. Not a single report detailing the comparison of different algorithms using the same experimental test-bed has been found. The lack of results for non-minimum phase plants, either theoretical or practical, is also evident in the literature.

Chapter 3

Experimental Apparatus

3.1 Introduction

Due to the lack of published work concerning ILC and non-minimum plants, the experimental test facility will be designed to belong to that class of system. Practical examples of this type of plant include the sway (sideways motion) control of ships using the rudder, and the heave (vertical motion) control of submarines by using the stern hydroplanes. Aircraft have equivalent types of motion to both of these examples. The experimental test-bed that is required does not actually need to perform a set task or represent an actual machine in any way. This therefore gives the design process a great deal of freedom and flexibility in meeting its goals. It has been decided that a sufficient control challenge exists in the form of a non-minimum phase plant without the extra difficulties associated with multivariable control. For this reason the plant will be SISO, a specification which also helps curtail production costs. The steps taken in the design of the apparatus are set out in this chapter, and reasons for the decisions made are expounded.

3.2 Mechanical System

In addition to being non-minimum phase, the only other criteria imposed on the experimental apparatus is that it be open-loop stable, possess sufficient relative degree and be of high enough order to test ILC schemes to their full potential. Electronic analogues have been consulted in the search for a non-minimum phase characteristic, themselves being a rich source of unusual input-output relationships. Although the Bridge Tee and Twin Tee circuits are good candidates for the purpose of design, their ungrounded capacitors are difficult to be physically realised by electrical-mechanical analogues. An 'inerter' (Smith, 2002) is capable of synthesising this element but it is an untested approach which may not offer reliability. Figure 3.1 shows an equivalent circuit which

does fulfil the non-minimum phase criteria whilst being implementable with reliable components. Its transfer function is given by,

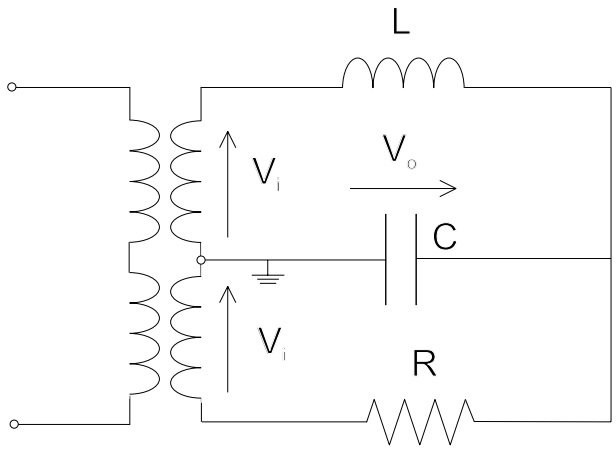


FIGURE 3.1: Electrical non-minimum phase circuit

$$\frac{V_o}{V_i} = \frac{\frac{R}{L} - s}{CRs^2 + s + \frac{R}{L}} \tag{3.1}$$

and its right half plane zero can be immediately seen.

$$J = C \qquad B = \frac{1}{R} \qquad K = \frac{1}{L} \tag{3.2}$$

The substitutions given in Equation 3.2 convert the electrical circuit to the mechanical realisation shown in Figure 3.2, where J represents an inertia, B a damper, K a torsional spring, G a gear ratio, and θ rotational position. The system is purely rotational with θ_i connected to the motor shaft, possibly by means of another gear, and θ_o constituting the output. The fact that the non-minimum phase characteristic can be achieved so

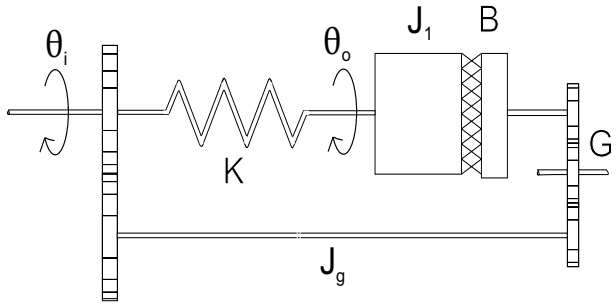


FIGURE 3.2: Mechanical realisation of stage 1

simply has led to the decision that the complete design be built up in three stages, each adding a section to the last. This results in three experimental test-beds on which ILC techniques can be applied, allowing complex algorithms to be implemented on the simpler configurations before they are tried on the complete system. This progression

also permits comparison of the results collected on each configuration, and modelling of the system to be built up additively, in the time-domain at least.

3.2.1 Stage 1

The simplest configuration of the system, 'stage 1', consists of just the mechanical section shown in Figure 3.2. Its state-space matrices, which neglect friction, are given in Appendix A. They show that it has a relative degree of one and 2 (of 4) unobservable states. This is unavoidable but may limit some of the algorithms that will later be applied. Simulation results of the system have been conducted for different values of G in order to utilise the full operational range of the damper but not damage it. From these it was decided to set the gear ratio to unity. This leads to the configuration shown in Figure 3.3 with a drive belt replacing the left-hand gears, and also connecting them to the drive shaft. The damper and spring are also split between the two drive trains in order to save space. Note that the motor and its timing pulley are located beneath the rest of the components, and that J_g includes the contribution of the left-hand timing pulley's mass, and J includes the right-hand gears' masses. The required values of the

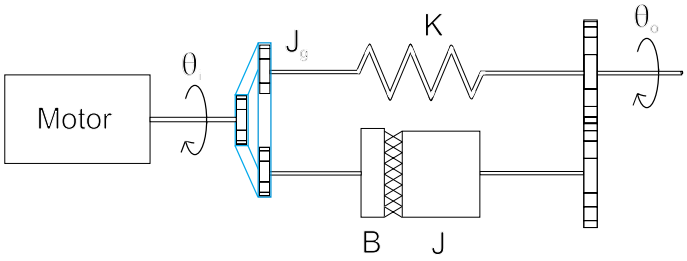


FIGURE 3.3: Stage 1 testbed schematic

parameters have been selected by conducting a number of simulations in order to obtain a response that meets the required needs in terms of stability, error, oscillation, and the non-minimum phase characteristic. This has involved linearising damper data-sheet information, calculating torsional spring equations, extrapolating from torsional spring data-sheets, and the summation of individual inertia values in order to calculate the values needed in the models. Chapter 4 discusses the modelling of all the system stages. The very approximate estimation of the stage 1 system's transfer function is given by,

$$\frac{\dot{\theta}_0}{\dot{\theta}_1} = \frac{-5.526s + 22.063}{s^2 + 5.5162s + 22.063} \tag{3.3}$$

and this allows a negative feedback gain K, such that $0 \leq \boldsymbol{K} \leq 1.0$, to be used on the system before instability occurs. This indicates that a feedback controller can be used effectively with the system. Even with the addition of the motor and inverter together with the simplifications made, this model should serve as a good approximation of the stage 1 plant dynamics.

3.2.2 Stage 2

The stage 2 plant consists of the non-minimum phase stage plus an additional spring-mass-damper system. To provide the extra complexity required, the sections shown in Figures 3.4 and 3.5 have been examined and simulated as possible additions between the motor and the non-minimum phase section. They differ in whether the damper is directly in the drive train, or connected 'to earth'. The gearing is intended to allow greater utilisation of the damper's full range of velocity.

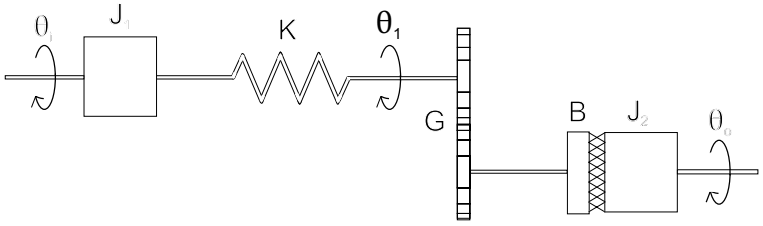


FIGURE 3.4: Drive train with an 'in line' damper

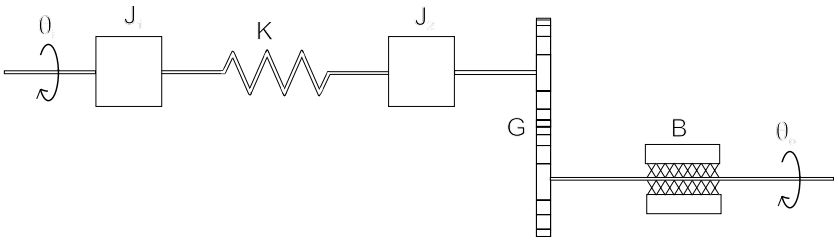


FIGURE 3.5: Drive train with a damper 'to earth'

The state-space matrices corresponding to Figures 3.4 and 3.5 are given in Appendix A, as well as those of the systems produces when the system shown in Figure 3.4 preceeds the non-minimum phase section and that of Figure 3.5 preceeds the non-minimum phase section. The results show that using the configuration shown in Figure 3.4 produces 1 (of 5) uncontrollable states, and 3 (of 5) unobservable states when combined with the non-minimum phase section. When Figure 3.5 is examined it is found that it only has 2 (of 4) unobservable states, and no uncontrollable states. This is the primary reason that the layout of Figure 3.5 has been chosen, but it has also been observed that locating the damper directly in the drive train is a much larger potential source of errors in the modelling stage, its characteristic being largely nonlinear. Since both alternatives have a relative degree of 3, there appears to be little compromise in the selection made. Figure 3.6 shows the overall schematic of the stage 2 test-bed. When data sheets on commercially available dampers are consulted, it can be seen that the gearing present in Figure 3.5 is unnecessary, especially since the non-minimum phase damper is required to operate at approximately twice the velocity of the other. The state-space matrices of the second stage system are given in Appendix A. Component values are chosen, as before, by examining simulated responses of the system and by observing component

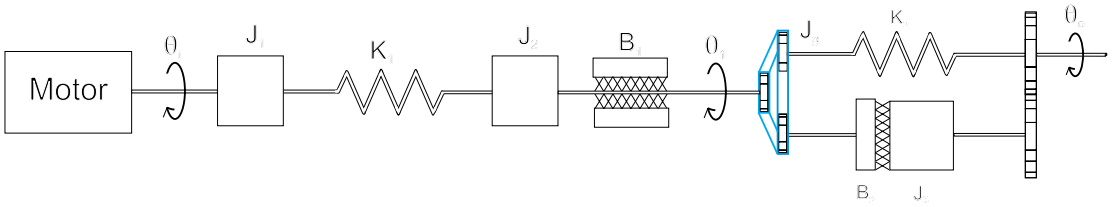


FIGURE 3.6: Stage two testbed schematic

limitations (velocity for dampers, maximum displacement for springs). The approximate estimation of the stage 2 system's transfer function is given by,

$$\frac{\dot{\theta}_0}{\dot{\theta}_1} = \frac{-0.647s + 2.588}{0.001s^4 + 0.014s^3 + 0.17s^2 + 1.229s + 2.588} \tag{3.4}$$

and this predicts that a negative feedback gain K, such that $0 \leq \boldsymbol{K} \leq 1.02$, can be applied before the onset of instability occurs.

3.2.3 Stage 3

The stage 3 plant consists of the stage 2 system plus an additional spring-mass-damper system. To further augment the stage 2 plant, a natural choice is to add another section of the form shown in Figure 3.5. This is preferable to the form of Figure 3.4 for the reasons given previously. The component values of the two sections are selected differently to add variety to the system's dynamics and separate possible oscillation modes. The complete system is shown in Figure 3.7. Note that 'X' and 'Y' specify the

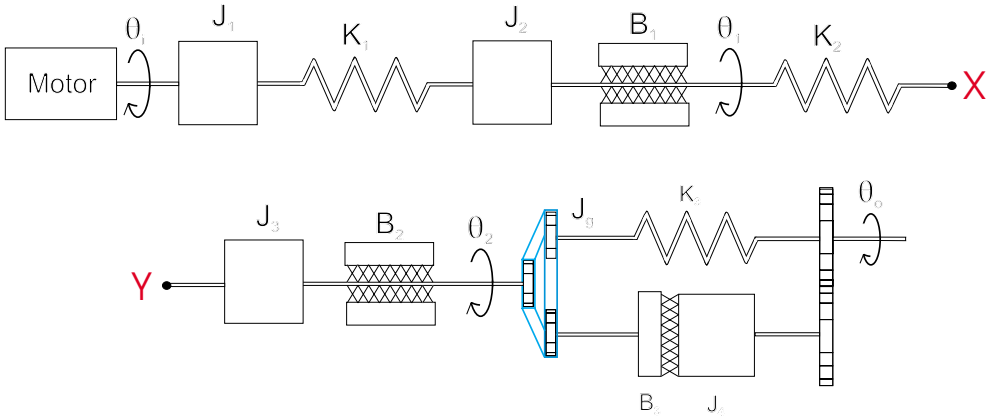


FIGURE 3.7: Stage three testbed schematic

same shaft and should follow on from one-another. The state space matrices of this system are contained in Appendix A. The stage 3 system has a relative degree of 5, 2 (of 8) unobservable states, and is fully controllable. The remaining component values (J_1, K_1, part of J_2) have been selected according to the guidelines already elucidated.

The resulting transfer function is given by,

$$\frac{\dot{\theta}_0}{\dot{\theta}_1} = \frac{-5.05s + 20.199}{0.0001s^6 + 0.0015s^5 + 0.034s^4 + 0.36s^3 + 2.185s^2 + 14.392s + 20.199} \tag{3.5}$$

and it should be again emphasised that this does not include friction, uses linearised damper information, unreliable spring equations, and approximate values of inertia. It is, however, accurate enough for the selection of parameters, and to show that the system is now of sufficient complexity to dispense with any further additions. The transfer function plant will become unstable with a feedback gain K, such that $0 \leq \boldsymbol{K} \leq 0.89$, can be applied which should allow an acceptable region of stability for ILC methods to operate in.

3.2.4 Physical Implementation

Figure 3.8 shows a schematic diagram of the test-bed with the structural elements omitted to leave just the component placement. The encoders are drawn in green, the dampers in dark blue, the timing belt in light blue, the motor in magenta, and the torsional springs in red. Note that the dampers all lie in the same horizontal plane as other elements in the system, and therefore overlap them. The non-minimum phase section is located directly above the rest of the drive train in order to conserve space. Whilst the two other spring-mass-damper sections can be removed and replaced with sections of shaft in order to make up stages 1 and 2, the non-minimum phase section is not altered.

A top-view of the non-minimum phase section is shown in Figure 3.9 with the principal

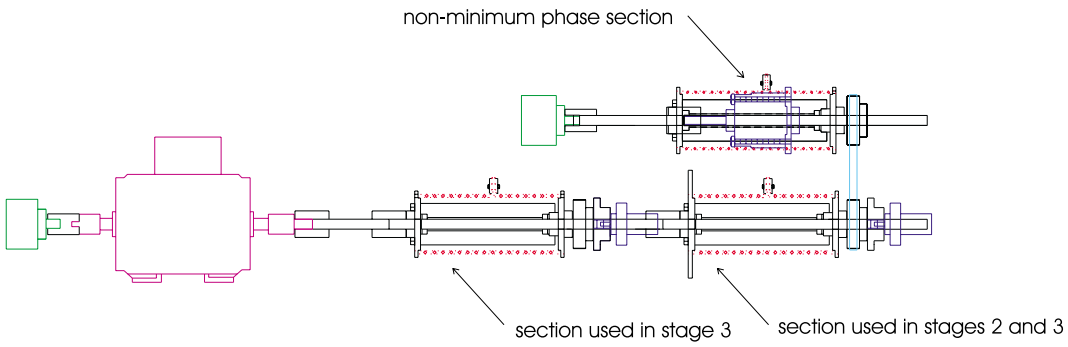


FIGURE 3.8: Stage three technical drawing

parts labelled. The torsional spring is situated around an aluminium mandrel which has a plate on either end. The spring is actually comprised of two springs connected together by means of the spring clamp, with their other ends secured to each end plate. The springs are wound in opposite directions to one another to make the torque/speed characteristic symmetrical. A roller bearing can rotate around the mandrel to reduce friction between the two, and whirling of the spring as velocity increases. The mandrel rotates about a shaft that reaches most of the way through, secured by two ball races, and terminated with a thrust bearing. The end plate nearest to this has another shaft

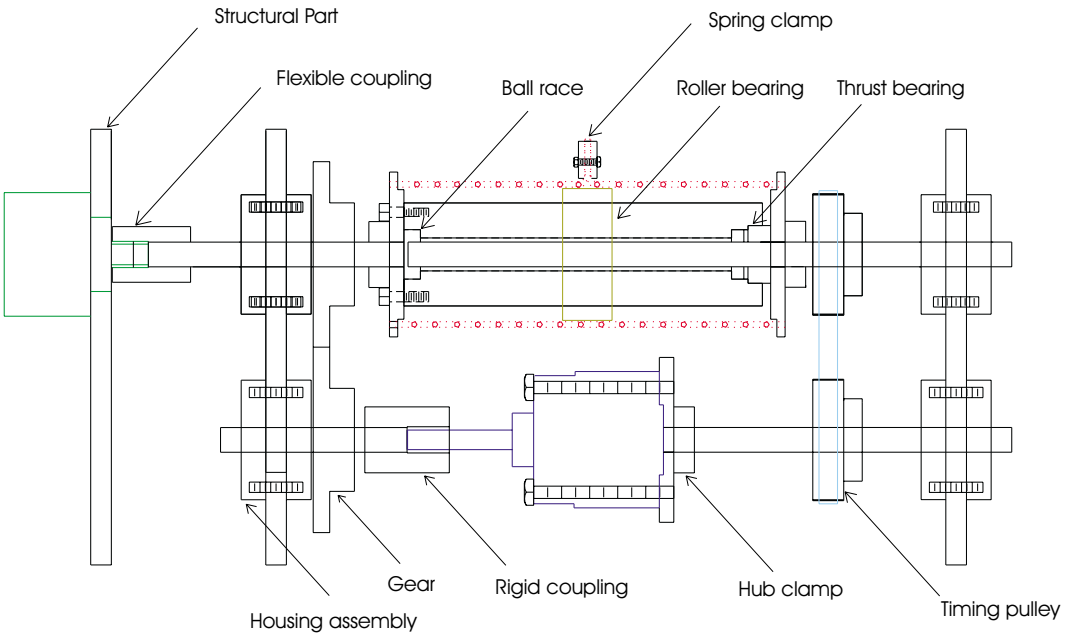


FIGURE 3.9: Non-minimum phase technical drawing

secured to it by means of a hub clamp. In this way one of the torsional springs is compressed and the other is expanded as one of the shafts is rotated axially with respect to the other. Inertia has been changed by means of the diameter of the endplates, the left hand endplate of section 2, for example, is constructed from stainless steel and has a larger diameter to add additional inertia at that juncture.

The structural parts of the apparatus can be seen in Figure 3.10 and consist of aluminium plates mounted on a test-bed by means of long aluminium sections running its full length. These long sections have grooves cut into them to accommodate the plates and ensure rigidity. Steel dowels are also used to add accuracy when reassembling the structure. A system comprising bearings, shafts and threaded bolts is used to tension the timing belt. The inverter and terminal box are mounted on either end of the apparatus to negate the need for further housing. Computer aided design (CAD) images of some of the principle parts of the system are shown in Appendix C.

FIGURE 3.10: Complete system in stage 3 form

3.3 Electrical System

Figure 3.11 shows the electrical connections in the system. A commercial ISA expansion card constitutes the interface between hardware and software. It can output two ±10V, 14-bit analogue signals, of which only one is necessary for this application, an opto-isolated digital output to enable the inverter, and two 32-bit counters for encoder input. Edge triggering is used by the encoder counters which produces a fourfold increase in their resolution. Information pertaining to the I/O card is given in Appendix B. All signal lines are screened and earthed to the PC case in order to minimise electromagnetic interference (EMI). The inverter provides three phase pulse width modulated (PWM)

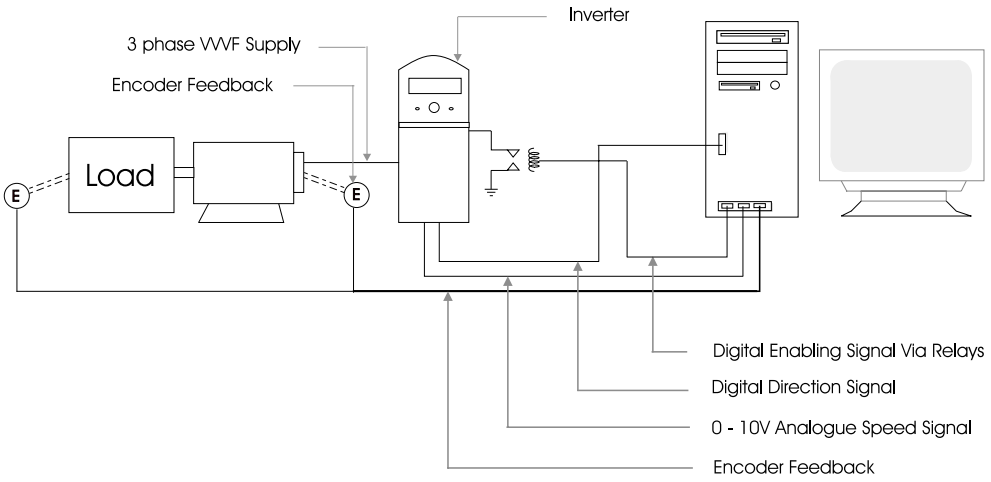


FIGURE 3.11: Experimental apparatus schematic

voltage to the motor, with a constant voltage to frequency ratio. Since the flux produced by the stator of the motor is proportional to this ratio, it means that torque can be kept constant throughout the speed range of the motor. As the inverter only accepts a unipolar speed demand, a direction signal is required from the computer. This is achieved by using a relay and a line from the PC's parallel port.

3.4 Software

A 1GHz Pentium 3 PC running Windows 98 is used to implement the control software. Since it is doubtful whether computer code running in a Dynamic Link Library (dll) under windows can deliver faster interrupts than 1 to 2 ms, DOS has been chosen as the operating system (OS) in which to implement the controllers. This also ensures the system resources are fully available to the task of inter-sample calculations. The DOS applications offer the operator menu-based run-time configuration of the system parameters. The library (lib) files that are provided with the I/O card allow functions in the C++ programming environment to communicate with it. The software has been

organised so that the implementation of all the controllers predominantly requires alteration of a single file. Controllers have been implemented as interrupt service routines (ISR), the interrupt being generated from a programmable interrupt controller on the I/O card.

3.5 Summary

A system has been designed that allows the implementation of a wide range of ILC algorithms on a SISO non-minimum phase plant. Particular attention has been paid to ensuring that it is appropriate for use with a wide range of those algorithms seen in Chapter 2, and for that reason it can be configured in three levels of mechanical complexity. This also allows the examination of how the system order and relative degree affect ILC control techniques.

Chapter 4

System Modelling and Validation

4.1 Introduction

Two separate approaches have been used to model the system. The first breaks the system down into separate components, or in as small groups as possible. Theoretical expressions for each one are produced and verified against experimental data. These simple components are then linked together to form a time-based model of the complete system. The second approach is conducted in the frequency domain and consists of a linear model that is fitted to experimentally obtained frequency response data. This approach is essential as a linear model is necessary for most model-based controllers and for many control analysis techniques. The time-based model is used to validate controllers that have been designed using the linear model. Both models have been created using a commercial, mathematical simulation software package.

4.2 Time Based Model

4.2.1 Overview

The components in the system are the I/O card, inverter, induction motor, encoder, and mechanical components consisting of inertias, springs, dampers, and a timing belt. Each of these components forms a self contained model, which interacts with the elements that come before and after it. Figure 4.1 shows this interaction.

4.2.2 I/O Card

The I/O card responds to a digital value representing the desired analogue output in volts. Measurements throughout the range of output voltage have shown an additional

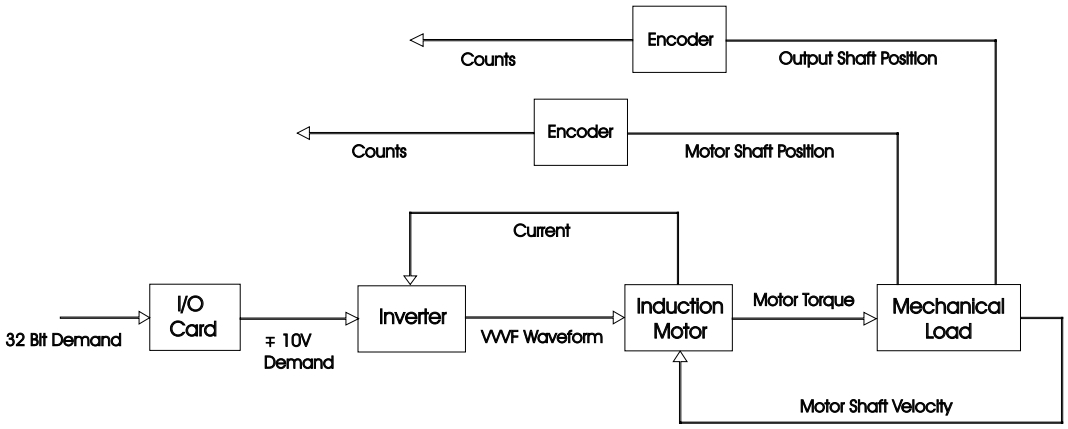


FIGURE 4.1: Time based model overview

attenuation of 0.9781, meaning that its overall gain is 0.97810^{-3}. The storage and outputting of 14-bit numbers results in a quantisation interval of $\frac{1}{2^{14}}$.

4.2.3 Induction Motor

A three phase induction motor can be represented directly by means of a phase variable model (Colak et al., 1995; Robertson and Hebbar, 1969). It is more convenient, however, to represent a.c motors using a two axis, d.q. approach (Jordan, 1965; Krause, 1965; Jones, 1967; Krause, 1968; DeSarkar and Berg, 1970; Hancock, 1974; Bose, 1987; Vas, 1990). Using this method requires less computational effort than the others (MacDonald and Sen, 1979), but has some inherent assumptions about the system that may not always produce acceptable results. If this is the case then it is possible to compensate for them, though this increases the complexity (Slemon, 1989). The equations for a two axis, d.q representation of an induction motor with N poles are given by

$$\begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{dr} \\ v_{qr} \end{bmatrix} = \begin{bmatrix} R_s + L_s p & 0 & M_{sr} p & 0 \\ 0 & R_s + L_s p & 0 & M_{sr} p \\ M_{sr} p & N\omega_r M_{sr} & R_r + L_r p & N\omega_r M_{sr} \\ -N\omega_r M_{sr} & M_{sr} p & -N\omega_r M_{sr} & R_r + L_r p \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} \tag{4.1}$$

$$T_r = J_{mot} p\omega_r + B_{mot}\omega_r - NM_{sr}(i_{ds} i_{dr} - i_{ds} i_{qr}) \tag{4.2}$$

where R_s and R_r are the stator and rotor resistance, L_s and L_r are the stator and rotor inductance, M_{sr} is the mutual inductance, w_r is the rotor mechanical angular velocity, and p is the differential operator. Parameter values have been obtained by no-load and locked rotor tests. Figure 4.2 shows experimental results which confirm the accuracy of this approach.

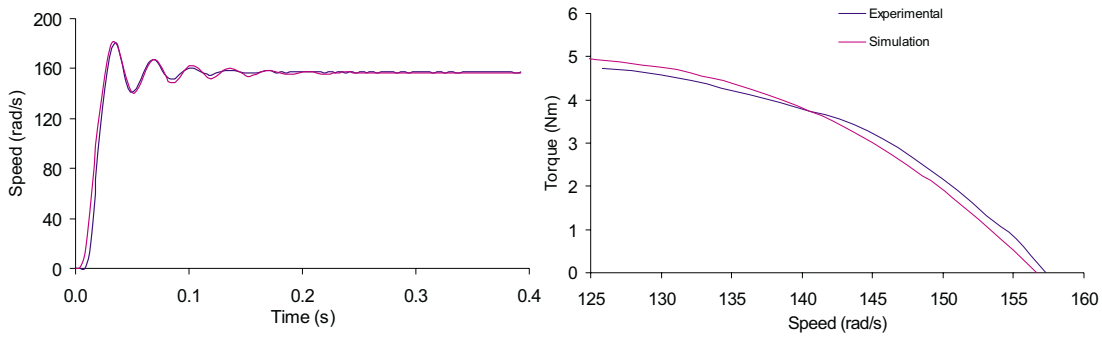


FIGURE 4.2: Induction motor on fixed supply

4.2.4 Inverter Model

The inverter utilises Sinusoidal Pulse Width Modulation (SPWM) which means that all harmonics below $2h-1$ are eliminated (Rashid, 1993), where h is the number of pulses per half cycle such that

$$h = \frac{f_c}{2f_0} \tag{4.3}$$

Here f_c is the carrier frequency and f_0 is the output frequency. The PWM switching frequency is 2.9KHz for the inverter used which means that all harmonics less than 1.45kHz are eliminated. The electrical circuit of the induction motor can be thought of as a low pass filter with a bandwidth of a few tens of hertz (Stefanovic and Barton, 1977), which implies that the inverter can be represented as an ideal variable frequency, variable amplitude sinusoid

$$v_0(t) = \sqrt{2}V_{rms}\alpha\sin(\beta t) \tag{4.4}$$

where β is the demanded output frequency in rads^{-1} and α is an amplitude scalar to give the correct V/f ratio. As Figure 4.3 shows, the inverter model comprises a control

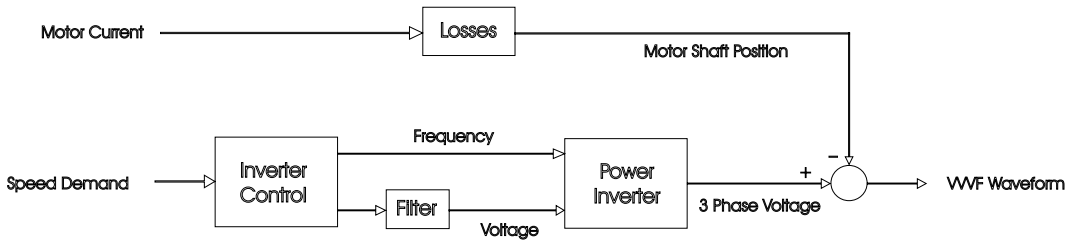


FIGURE 4.3: Overview of the inverter model

section to calculate the required voltage and frequency, and a power section to generate the Variable Voltage, Variable Frequency (VVVF) waveform based on Equation 4.4. A low pass filter on the voltage demand signal is used to approximate the resistance and capacitance of the d.c link. Load current determines inverter losses via a resistance term, this is then subtracted from the phase voltages. Information provided by the inverter

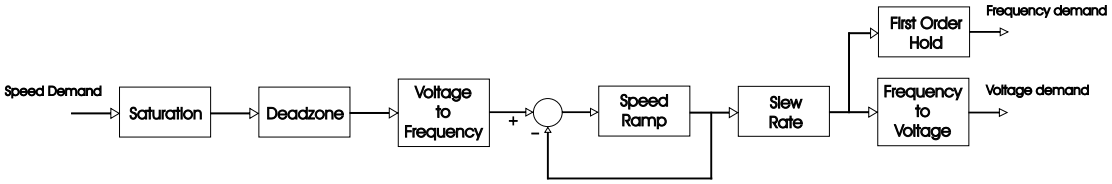


FIGURE 4.4: Inverter control block

manufacturer can be incorporated in the inverter control block that is shown in Figure 4.4. Expressions for the demand voltage to frequency and frequency to RMS line voltage have been found from the experimental results shown in Figure 4.5. Results confirming

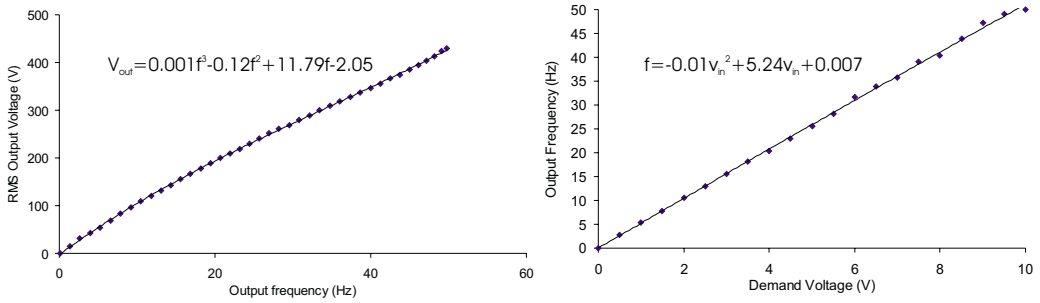


FIGURE 4.5: Frequency and voltage characteristics of the inverter

the accuracy of the overall motor and inverter simulation are shown in Figures 4.6 and Figure 4.7. The simulations do not match the experimental data exactly due to the simplifications and simplifications made in the modelling process and the presence of unmodelled high order dynamics.

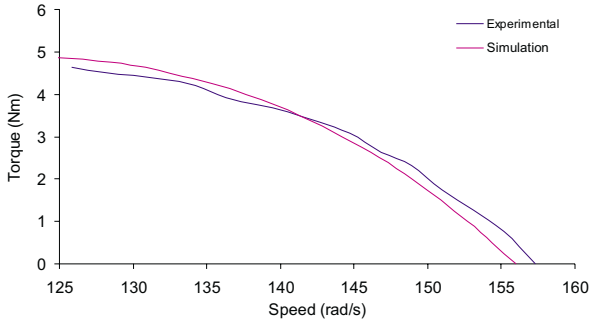


FIGURE 4.6: Induction motor speed test on inverter supply

4.2.5 Belt Model

In order to start extending the linear models of the three stages of the system seen in Sections 3.2.1, 3.2.2, and 3.2.3, the timing belt can be represented by the spring-mass-damper system shown in Figure 4.8. T_A represents the torque input to the belt and T_B and T_C are the load torques from the two other branches. Damping could be

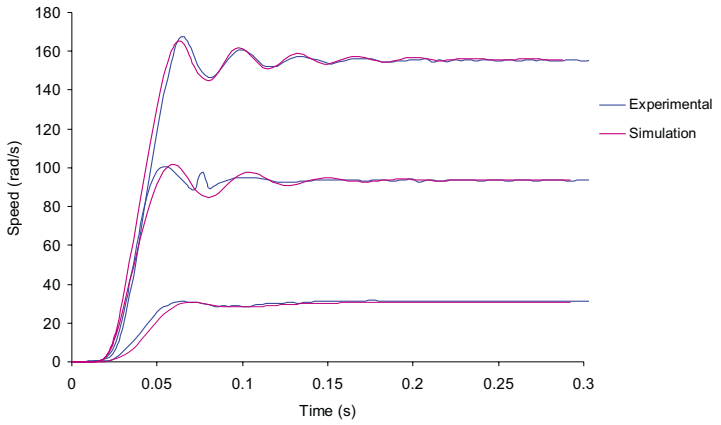


FIGURE 4.7: Step response of the induction motor on inverter supply

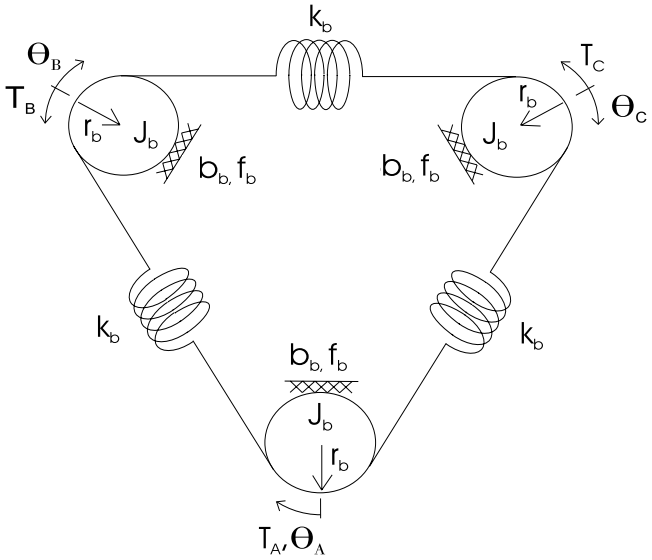


FIGURE 4.8: Timing belt model

included in parallel with each spring element k_b (Abrate, 1992), but this is negligible when compared to the other sources of damping in the system. The inertia, damping, and friction of those elements that are rigidly connected to each pulley should be added to the parameters J_b, b_b, f_b, or referred back as part of the load torque T_B or T_C. The resulting linear model of the belt uses small angle approximations and is given by

$$\begin{bmatrix} \ddot{\theta}_A \\ \ddot{\theta}_B \\ \ddot{\theta}_C \end{bmatrix} = \begin{bmatrix} \frac{-2r_b^2k_b}{J_b} & \frac{-b_b}{J_b} & \frac{r_b^2k_b}{J_b} & 0 & \frac{r_b^2k_b}{J_b} & 0 \\ \frac{r_b^2k_b}{J_b} & 0 & \frac{-2r_b^2k_b}{J_b} & \frac{-b_b}{J_b} & \frac{r_b^2}{J_b} & 0 \\ \frac{r_b^2k_b}{J_b} & 0 & \frac{r_b^2k_b}{J_b} & 0 & \frac{-2r_b^2k_b}{J_b} & \frac{-b_b}{J_b} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} \frac{1}{J_b} & 1 & 1 \\ 0 & \frac{-1}{J_b} & 0 \\ 0 & 0 & \frac{-1}{J_b} \end{bmatrix} \underline{u}(t) \quad (4.5)$$

where

$$\underline{u}(t) = \begin{bmatrix} T_A & T_B & T_C \end{bmatrix}^T \qquad \underline{y}(t) = \begin{bmatrix} \ddot{\theta}_A & \ddot{\theta}_B & \ddot{\theta}_C \end{bmatrix}^T$$
$$\underline{x}(t) = \begin{bmatrix} \theta_A & \dot{\theta}_A & \theta_B & \dot{\theta}_B & \theta_C & \dot{\theta}_C \end{bmatrix}^T \quad (4.6)$$

The coulomb friction parameters can only be incorporated into the system by the addition of an extra term on the right hand side of Equation 4.5:

$$-\begin{bmatrix} a_A & a_B & a_C \end{bmatrix}^T \tag{4.7}$$

where

$$sign(a_i) = \begin{cases} +ive \text{ if } \dot{\theta}_i > 0 \\ -ive \text{ if } \dot{\theta}_i < 0 \end{cases} \qquad |a_i| = \begin{cases} \frac{f_b}{J_b} \text{ if } \dot{\theta}_i \neq 0 \\ -\frac{\sum_i F_i}{J_b} \text{ otherwise} \end{cases} \qquad i\epsilon\{A, B, C\} \tag{4.8}$$

$\sum_i F_i$ is the sum of the forces acting on J_i and equals $J_i\ddot{\theta}_i$ provided that $\ddot{\theta}_i$ has been calculated at the same instant, ignoring friction. It is then superseded by the new value obtained using this term. If $\dot{\theta}_i = 0$ in the right hand side of Equation 4.7, the direction of $\sum_i F_i$ must be considered. The presence of coulomb friction means that the system is non-linear.

4.2.6 Stage 1 Mechanical System

The belt model must have the sections shown in Figure 4.9 added to it in order to represent the stage 1 system. The section shown in Figure 4.9 (a) models the interaction

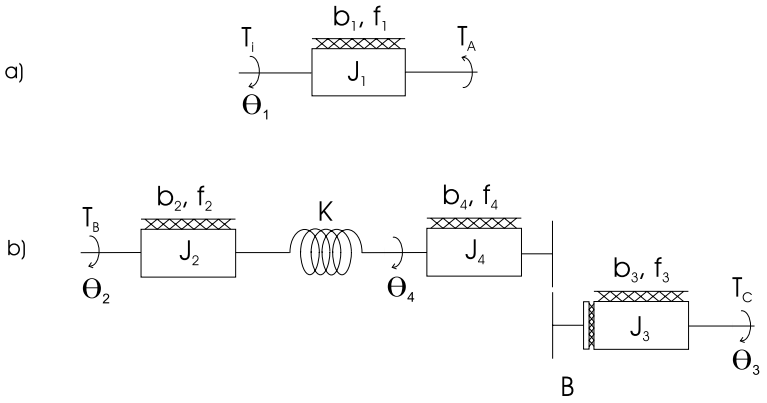


FIGURE 4.9: Stage 1 additions

between the torque supplied by the motor, T_i, and that driving the belt model, T_A. In fact the parameters b_1, f_1, J_1 can simply be added to the corresponding b_b, f_b, J_b and the position terms equated:

$$\theta_A = \theta_1 \tag{4.9}$$

Figure 4.9 (b) describes the interaction between T_B and T_C. The terms b_2, f_2, and J_2 could be added to the relevant J_b, b_b, and f_b of the belt model, and the equivalent done for b_3, f_3, and J_3. Before being combined with this model, the velocities of the belt

model must be equated to these new parameters by setting

$$\theta_B = \theta_2 \qquad \text{and} \qquad \theta_C = \theta_3 \tag{4.10}$$

The equations of this section are given by

$$\begin{bmatrix} \ddot{\theta}_4 \end{bmatrix} = \begin{bmatrix} \frac{K}{J_4} & 0 & 0 & \frac{B}{J_4} & \frac{-K}{J_4} & \frac{-(b_4+B)}{J_4} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0 & 0 \end{bmatrix} \underline{u}(t) \tag{4.11}$$

$$\underline{y}(T) = \begin{bmatrix} K & b_2 & 0 & 0 & -K & 0 \\ 0 & 0 & 0 & b_3+B & 0 & -B \\ \frac{K}{J_4} & 0 & 0 & \frac{B}{J_4} & \frac{-K}{J_4} & \frac{-(b_4+B)}{J_4} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} J_2 & 0 \\ 0 & J_3 \\ 0 & 0 \end{bmatrix} \underline{u}(t) \tag{4.12}$$

where

$$\begin{aligned} \underline{u}(t) &= \begin{bmatrix} \ddot{\theta}_2 & \ddot{\theta}_3 \end{bmatrix} \qquad \underline{y}(t) = \begin{bmatrix} T_B & T_C & \ddot{\theta}_4 \end{bmatrix} \\ \underline{x}(t) &= \begin{bmatrix} \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & \theta_4 & \dot{\theta}_4 \end{bmatrix}^T \end{aligned} \tag{4.13}$$

The right hand side of both the matrix state equation and Equation 4.12 requires an extra term to account for coulomb friction. These terms are respectively

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & a_4 \end{bmatrix}^T \qquad \text{and} \qquad -\begin{bmatrix} -J_2 a_2 & -J_3 a_3 & a_4 \end{bmatrix}^T \tag{4.14}$$

with

$$sign(a_i) = \begin{cases} +ive \ \text{ if } \ \dot{\theta}_i \geq 0 \\ -ive \ \text{ if } \ \dot{\theta}_i < 0 \end{cases} \qquad |a_i| = \begin{cases} \frac{f_b}{J_b} \ \text{ if } \ \dot{\theta}_i > 0 \\ -\frac{\sum_i F_i}{J_b} \ \text{ otherwise} \end{cases} \qquad i\epsilon\{2,3,4\} \tag{4.15}$$

The conditions used to calculate the friction in Equation 4.14 are the same as for the belt model. The overall, nonlinear, state-space representation of the system is obtained by combining the two models, this results in

$$\begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} = \begin{bmatrix} \frac{k_b r_b^2}{J_b+J_2} & 0 & \frac{-(K+2k_b r_b^2)}{J_b+J_2} & \frac{-(b_2+b_b)}{J_b+J_2} & \frac{k_b r_b^2}{J_b+J_2} & 0 & \frac{K}{J_b+J_2} & 0 \\ \frac{k_b r_b^2}{J_b+J_3} & 0 & \frac{k_b r_b^2}{J_b+J_3} & 0 & \frac{-2k_b r_b^2}{J_b+J_3} & \frac{-(B+b_3+b_b)}{J_b+J_3} & 0 & \frac{-B}{J_b+J_3} \\ 0 & 0 & \frac{K}{J_4} & 0 & 0 & \frac{B}{J_4} & \frac{-K}{J_4} & \frac{B-b_4}{J_4} \end{bmatrix} \underline{x}(t) \tag{4.16}$$

$$\begin{bmatrix} T_i \\ \ddot{\theta}_4 \end{bmatrix} = \begin{bmatrix} 2k_b r_b^2 & b_1+b_b & -k_b r_b^2 & 0 & -k_b r_b^2 & 0 & 0 & 0 \\ 0 & 0 & \frac{K}{J_4} & 0 & 0 & \frac{B}{J_4} & \frac{-K}{J_4} & \frac{B-b_4}{J_4} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} J_1+J_b \\ 0 \end{bmatrix} \underline{u}(t) \tag{4.17}$$

where

$$\begin{aligned} \underline{u}(t) &= \begin{bmatrix} \ddot{\theta}_1 \end{bmatrix} \qquad \underline{y}(t) = \begin{bmatrix} T_i & \ddot{\theta}_4 \end{bmatrix} \\ \underline{x}(t) &= \begin{bmatrix} \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & \theta_4 & \dot{\theta}_4 \end{bmatrix}^T \end{aligned} \tag{4.18}$$

The friction terms to be added to the right hand side of Equations 4.21 and 4.22 are respectively

$$-\begin{bmatrix} \frac{a_2}{J_2+J_b} & \frac{a_3}{J_3+J_b} & \frac{a_4}{J_4} \end{bmatrix}^T \qquad \text{and} \qquad -\begin{bmatrix} -a_1 & \frac{a_4}{J_4} \end{bmatrix}^T \tag{4.19}$$

with

$$sign(a_i) = \begin{cases} +ive \ \text{ if } \ \dot{\theta}_i \geq 0 \\ -ive \ \text{ if } \ \dot{\theta}_i \leq 0 \end{cases} \qquad |a_i| = \begin{cases} f_i, \ i = 4 \\ f_i + f_b \ \text{ if } \ \dot{\theta}_i \neq 0, \ i = 1,2,3 \\ \sum F_i \ \text{ otherwise} \end{cases} \qquad i\epsilon\{1,2,3,4\} \tag{4.20}$$

Again, the sum of all forces acting on J_i, $\sum F_i$, must be calculated neglecting the friction term, and if $\dot{\theta}_i = 0$, the direction of $\sum F_i$ must be used in its place.

4.2.7 Stage 2 Mechanical System

The belt model must have the sections shown in Figure 4.10 added to it in order to represent the stage 2 system. This produces the following relationships

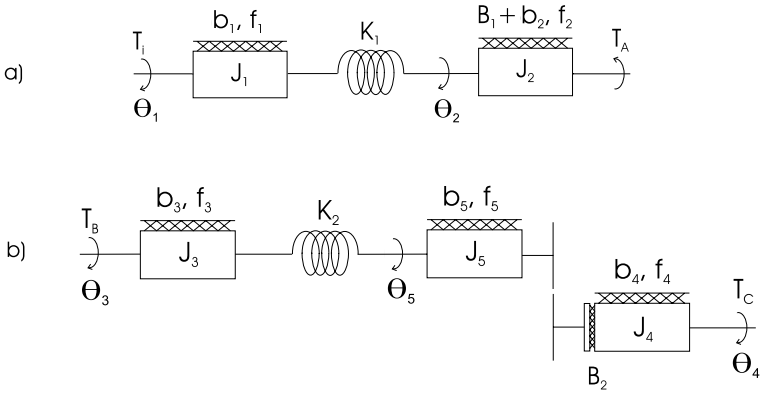


FIGURE 4.10: Stage 2 additions

$$\begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \end{bmatrix} = \begin{bmatrix} \frac{K_1-2k_br_b^2}{J_2+J_b} & 0 & \frac{-K_1}{J_2+J_b} & \frac{-(B_1+b_2+b_b)}{J_2+J_b} & \frac{k_br_b^2}{J_2+J_b} & 0 & \frac{k_br_b^2}{J_2+J_b} & 0 & 0 & 0 \\ \frac{k_br_b^2}{J_3+J_b} & 0 & 0 & 0 & \frac{-(K_2+2k_br_b^2}{J_3+J_b} & \frac{-(b_3+b_b)}{J_3+J_b} & \frac{k_br_b^2}{J_3+J_b} & 0 & \frac{K_2}{J_3+J_b} & 0 \\ \frac{k_br_b^2}{J_4+J_b} & 0 & 0 & 0 & \frac{k_br_b^2}{J_4+J_b} & 0 & \frac{-2k_br_b^2}{J_4+J_b} & \frac{-(B_2+b_4+b_b)}{J_4+J_b} & 0 & \frac{-B_2}{J_4+J_b} \\ 0 & 0 & 0 & 0 & \frac{K_2}{J_5} & 0 & 0 & \frac{B_2}{J_5} & \frac{-K_2}{J_5} & \frac{B_2-b_5}{J_5} \end{bmatrix} \underline{x}(t) \tag{4.21}$$

$$\begin{bmatrix} T_i \\ \ddot{\theta}_5 \end{bmatrix} = \begin{bmatrix} K+1 & b_1 & -K_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{K_2}{J_5} & 0 & 0 & \frac{B_2}{J_5} & \frac{-K_2}{J_5} & \frac{B_2-b_5}{J_5} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} J_1 \\ 0 \end{bmatrix} \tag{4.22}$$

where

$$\underline{u}(t) = \begin{bmatrix} \ddot{\theta}_1 \end{bmatrix} \qquad \underline{y}(t) = \begin{bmatrix} T_i & \ddot{\theta}_5 \end{bmatrix}^T$$
$$\underline{x}(t) = \begin{bmatrix} \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & \theta_4 & \dot{\theta}_4 & \theta_5 & \dot{\theta}_5 \end{bmatrix}^T \tag{4.23}$$

The friction terms to be added to Equations 4.21 and 4.22 are respectively

$$-\begin{bmatrix} \frac{a_2}{J_2+J_b} & \frac{a_3}{J_3+J_b} & \frac{a_4}{J_4+J_b} & \frac{a_5}{J_5} \end{bmatrix}^T \quad \text{and} \quad -\begin{bmatrix} -a_1 & \frac{a_5}{J_5} \end{bmatrix}^T \tag{4.24}$$

with

$$sign(a_i) = \begin{cases} +ive \text{ if } \dot{\theta}_i > 0 \\ -ive \text{ if } \dot{\theta}_i < 0 \end{cases} \qquad |a_i| = \begin{cases} f_i \text{ if } \dot{\theta}_i \neq 0,\ i = 1,5 \\ f_i + f_b,\ \ i = 2,3,4 \\ \sum_i F_i \text{ otherwise} \end{cases} \qquad \epsilon\{1,2,3,4,5\} \tag{4.25}$$

4.2.8 Stage 3 Mechanical System

The belt model must have the sections shown in Figure 4.11 added to it in order to represent the stage 3 system.

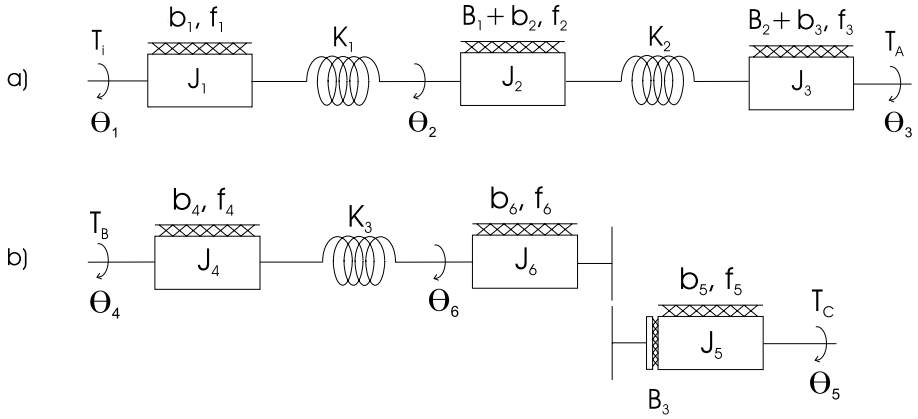


FIGURE 4.11: Stage 3 additions

$$\begin{bmatrix} \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \end{bmatrix} = \begin{bmatrix} \frac{K_1}{J_2} & 0 & \frac{-(K_1+K_2)}{J_2} & \frac{K_2}{J_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-2k_b r_b^2}{J_3+J_b} & 0 & \frac{K_2}{J_3+J_b} & 0 & \frac{-K_2}{J_3+J_b} & \frac{-(B_2+b_b+b_3)}{J_3+J_b} & \frac{k_b r_b^2}{J_3+J_b} & 0 & \frac{k_b r_b^2}{J_3+J_b} & 0 & 0 & 0 \\ \frac{k_b r_b^2}{J_4+J_b} & 0 & 0 & 0 & 0 & 0 & \frac{-2k_b r_b^2}{J_4+J_b} & \frac{-(b_4+b_b)}{J_4+J_b} & \frac{k_b r_b^2}{J_4+J_b} & 0 & \frac{K_3}{J_4+J_b} & 0 \\ \frac{k_b r_b^2}{J_5+J_b} & 0 & 0 & 0 & 0 & 0 & \frac{k_b r_b^2}{J_5+J_6} & 0 & \frac{-2k_b r_b^2}{J_5+J_b} & \frac{-(B_3+b_5+b_b)}{J_5+J_b} & 0 & \frac{-B_3}{J_5+J_b} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_3}{J_6} & 0 & 0 & \frac{B_3}{J_5} & \frac{-K_3}{J_5} & \frac{B_3-b_6}{J_5} \end{bmatrix} \underline{x}(t) \tag{4.26}$$

$$\begin{bmatrix} T_i \\ \ddot{\theta}_6 \end{bmatrix} = \begin{bmatrix} K_1 & b_1 & -K_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_3}{J_6} & 0 & 0 & \frac{B_3}{J_6} & \frac{-K_3}{J_6} & \frac{B_3-b_6}{J_6} \end{bmatrix} \underline{x}(t) + \begin{bmatrix} J_1 \\ 0 \end{bmatrix} \tag{4.27}$$

where

$$\underline{u}(t) = \begin{bmatrix} \ddot{\theta}_1 \end{bmatrix} \qquad \underline{y}(t) = \begin{bmatrix} T_i & \ddot{\theta}_6 \end{bmatrix}^T$$
$$\underline{x}(t) = \begin{bmatrix} \theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2 & \theta_3 & \dot{\theta}_3 & \theta_4 & \dot{\theta}_4 & \theta_5 & \dot{\theta}_5 & \theta_6 & \dot{\theta}_6 \end{bmatrix}^T \tag{4.28}$$

The friction terms to be added to the right hand side of Equations 4.26 and 4.27 are respectively

$$-\begin{bmatrix} \frac{a_2}{J_2} & \frac{a_3}{J_3+J_b} & \frac{a_4}{J_4+J_b} & \frac{a_5}{J_5+J_b} & \frac{a_6}{J_6} \end{bmatrix}^T \quad \text{and} \quad -\begin{bmatrix} -a_1 & \frac{a_6}{J_6} \end{bmatrix} \tag{4.29}$$

with

$$sign(a_i) = \begin{cases} +ive \ \text{ if } \ \dot{\theta}_i > 0 \\ -ive \ \text{ if } \ \dot{\theta}_i < 0 \end{cases} \qquad |a_i| = \begin{cases} f_i \ \text{ if } \ \dot{\theta}_i \neq 0, \ i = 1,2,6 \\ f_i + f_b, \ i = 3,4,5 \\ \sum_i F_i \ \text{ otherwise} \end{cases} \qquad i\epsilon\{1,2,3,4,5,6\} \tag{4.30}$$

Although derived from first principles, the stage 3 system was not employed because the first two stages were found to be more than adequate for experimentation.

4.2.9 Component Tests

The springs and dampers were individually tested to determine their characteristics.

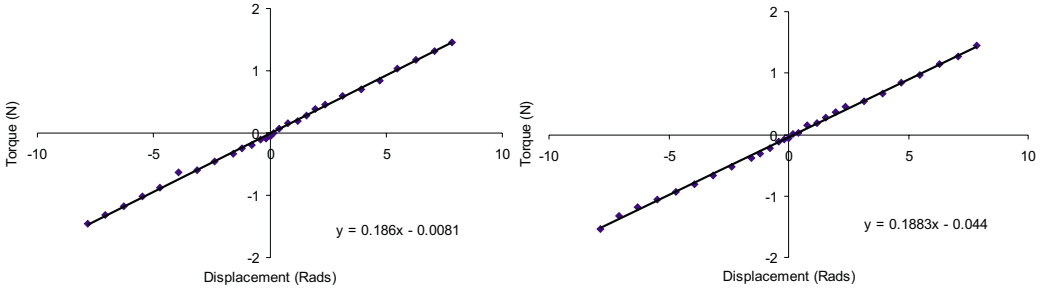


FIGURE 4.12: Results for springs 1 and 2

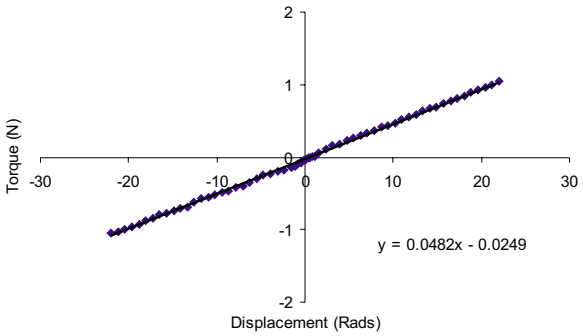


FIGURE 4.13: Results for spring 3

The two larger springs shown in Figure 4.12 have almost identical linear displacement to torque characteristics, and further experimentation revealeds negligible hysteresis. A polynomial expression has been fitted to each result. The less stiff non-minimum phase spring, shown in Figure 4.13, also has a characteristic that can be accurately modelled

as linear, and has a hysteresis which amounts to less than 0.03 Nrad at zero displacement. This hysteresis will only be considered if the model is unsatisfactory without it. Figure 4.14 shows that the damper selected to have a damping coefficient of 0.01

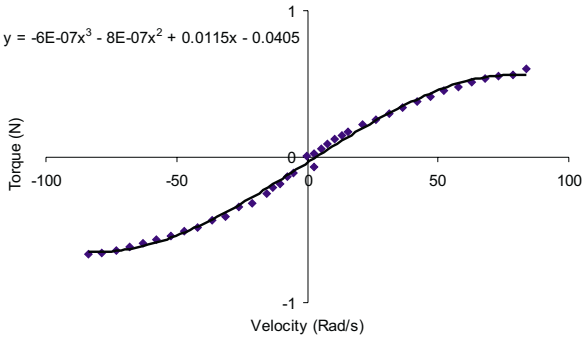


FIGURE 4.14: Results for non-minimum phase damper

Nsrad^{-1} actually has a non-linear characteristic modelled by the equation shown. The damper exhibits slight changes in characteristic due to thermal effects, but these have not been investigated to any great extent. Figure 4.15 shows the experimental results

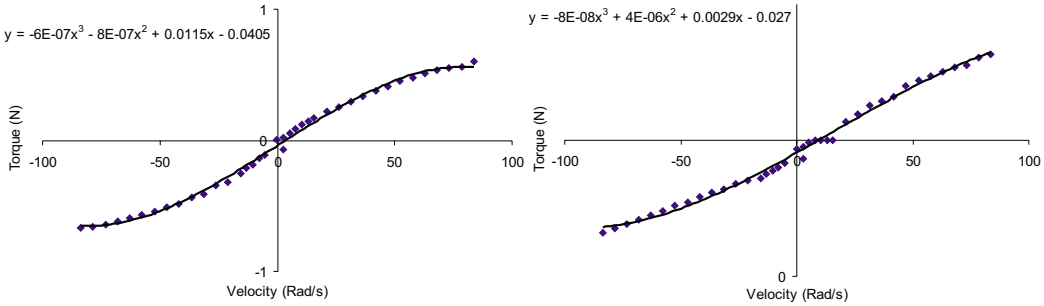


FIGURE 4.15: Results for small dampers

of tests conducted on the dampers used in the sections preceding the non-minimum phase section. They were selected to have coefficients of 0.0025 and 0.004 Nsrad^{-1} respectively, the results however show the true non-linear characteristic. In each case the fitted function is used in the simulation.

4.2.10 Stage 1 Model Validation

The simulations produced for the three stages of mechanical test-bed have been validated by comparing experimental and simulated responses to a variety of demands. Step response results are given in Figure 4.16, and Figure 4.17 shows responses to a repeating sequence input. The results show that the model is reasonably accurate but still fails to capture the high-order dynamics that can be seen in the experimental step response results.

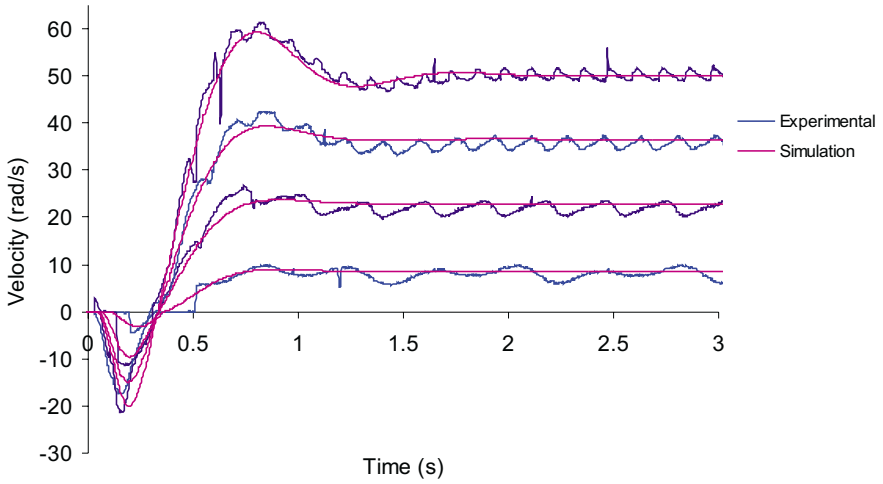


FIGURE 4.16: Stage 1 experimental and simulated step responses

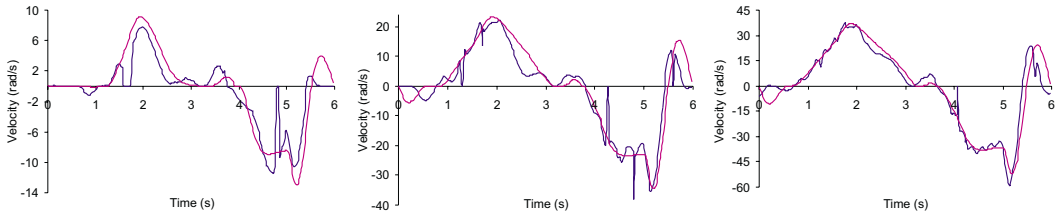


FIGURE 4.17: 10 UPM repeating sequence demands, 1, 2 and 3V amplitudes

Additional results for repeating sequence and sinewave inputs of varying amplitude and frequency are presented in Appendix E.

4.2.11 Stage 2 Model Validation

Step response results are given in Figure 4.18, and Figure 4.19 shows responses to a repeating sequence input. The results are less accurate than the corresponing test results for the stage 1 system since the addition of components directly into the drive-train has caused modelling errors to be magnified. Since the time-based models will be solely used to verify the performance of controllers derived using the frequency domain models, its accuracy appears to be satisfactory. Additional results for repeating sequence and sinewave inputs of varying amplitude and frequency are presented in E.

4.3 Frequency Based Model

Frequency response analysis is crucial in providing linear models suitable for use in model based controllers and general analysis. Frequency analysis also provides information not necessarily apparent from time response methods, such as the determination of pure

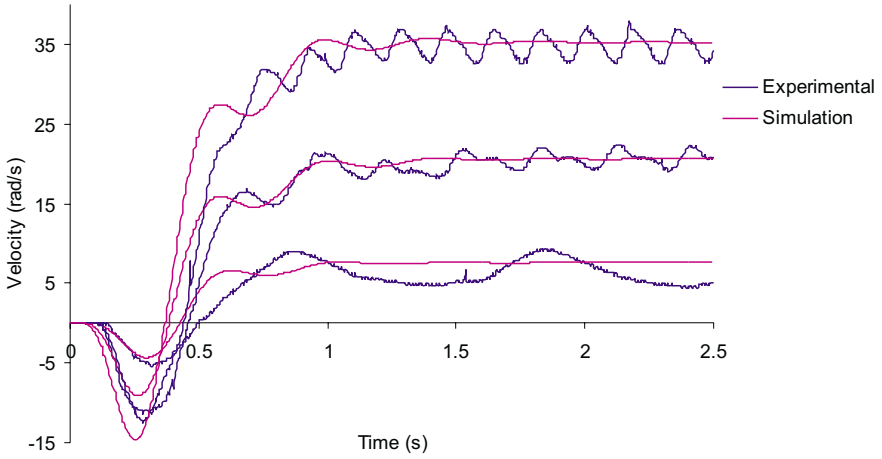


FIGURE 4.18: Stage 2 experimental and simulated step responses

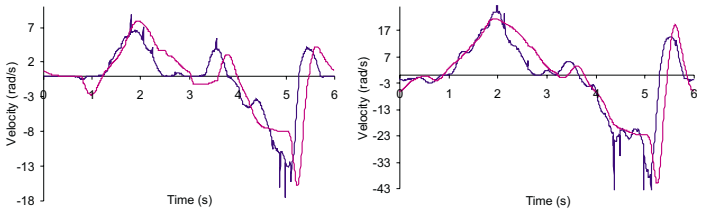


FIGURE 4.19: 10 UPM repeating sequence demands, 1 and 2V amplitudes

time delays. The system excitation is a sinusoidal voltage, and the position of the output shaft is the output.

4.3.1 Induction Motor and Drive Frequency Response

The drive system's response to sinusoidal demand signals of amplitude 1V and 3V, with a constant DC offset of 3V is shown in Figure 4.20. Frequency quenching can account for the presence of some of the low frequencies that gradually become more dominant than the high ones as the input frequency is increased. This is especially true when using the higher amplitude input signal. Examination of each frequency spectrum reveals no increasing low frequency component present that would be consistent with the 'sidebands' caused by an inverter sampling frequency within the 0-100Hz range used. Therefore it is unnecessary to include one in the model. Figure 4.21 a) shows the frequencies that a sinusoidal demand of 30Hz with an offset of 3V requires the inverter to provide. The 3V offset produces frequencies at 15Hz since the maximum output frequency of the inverter of 50Hz corresponds to a 10V demand. The case shown is a simple one as the difference in amplitudes (15Hz) also constitutes an output component which repeats at 30Hz intervals and exactly corresponds with the main frequency spectrum and no new frequencies are added. Figure 4.21 b) shows the usual complexity when the frequencies do not correspond. The 5V offset produces frequencies of 25Hz, and these repeat at

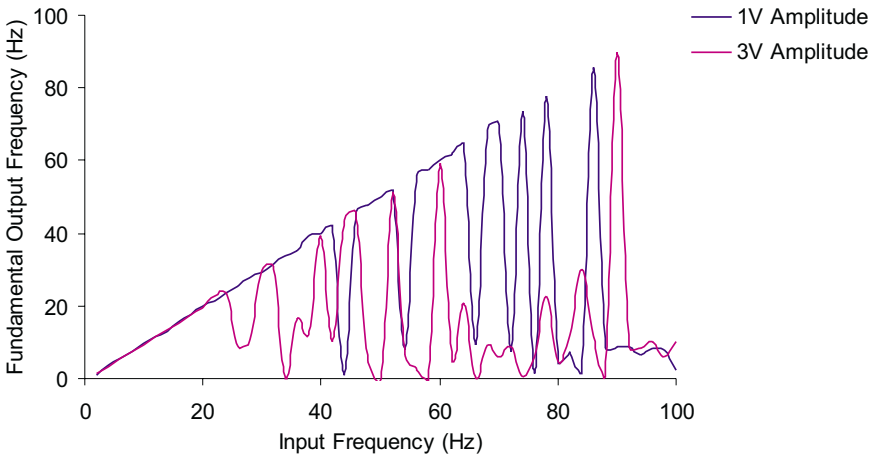


FIGURE 4.20: Drive input frequency to fundamental output frequency characteristic

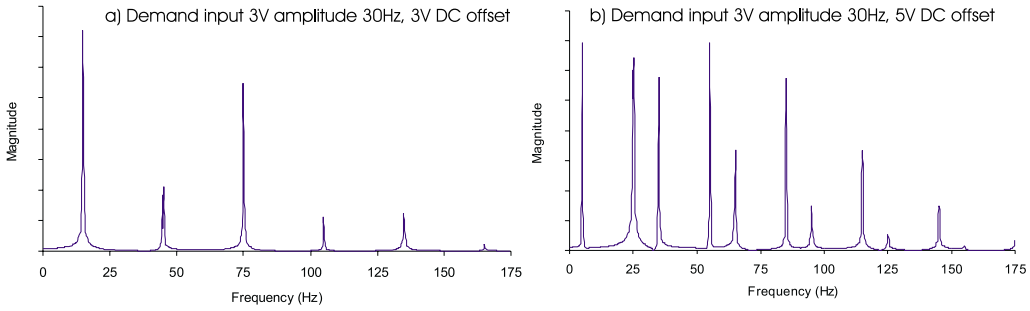


FIGURE 4.21: Simulated VVVF output spectrums of idealised inverter

30Hz intervals by virtue of the sinusoid superimposed on the demand. The difference in frequencies is 5Hz and so additional frequencies exist at 5Hz and at 30Hz intervals thereafter.

4.3.2 Stage 1 Linear Model

Linear models of the entire system in all three configurations have been calculated by means of experimentally produced bode plots. The frequency response of the system up to 30Hz was determined in each case. A fifth order approximation is sufficient to describe the bode plot data of the stage 1 system (Equation 4.31) its units are in radV^{-1}.

$$G_{stage1}(s) = \frac{123.853 \times 10^4 (3.5 - s)}{s(s^2 + 6.5s + 42.25)(s + 45)(s + 190)} \tag{4.31}$$

The bode plot of this model together with the experimental data used to obtain it is given in Figure 4.22.

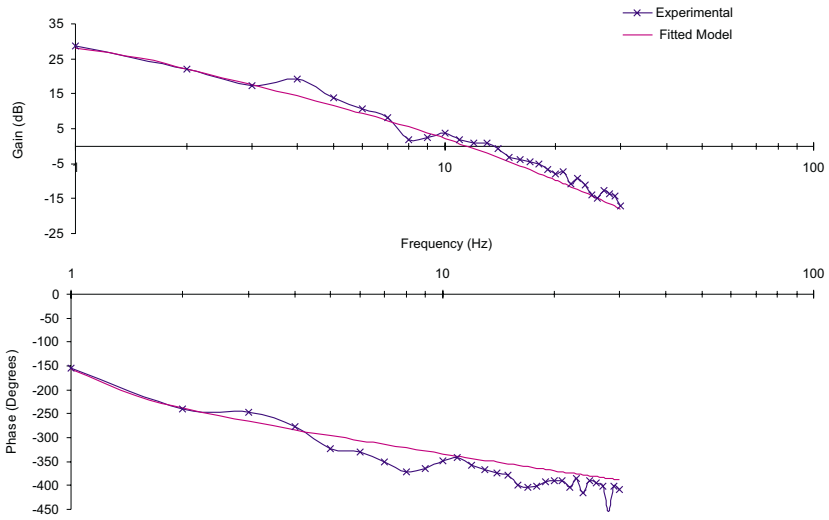


FIGURE 4.22: Experimental and modelled stage 1 frequency response

4.3.2.1 Model Validation

The simulations produced for the three stages of mechanical test-bed have been validated by comparing the actual response to a range of step responses with the simulations. Step response results for the stage 1 model are given in Figure 4.23, and Figure 4.24 shows responses to a repeating sequence input. Slight adjustments have been made to the model

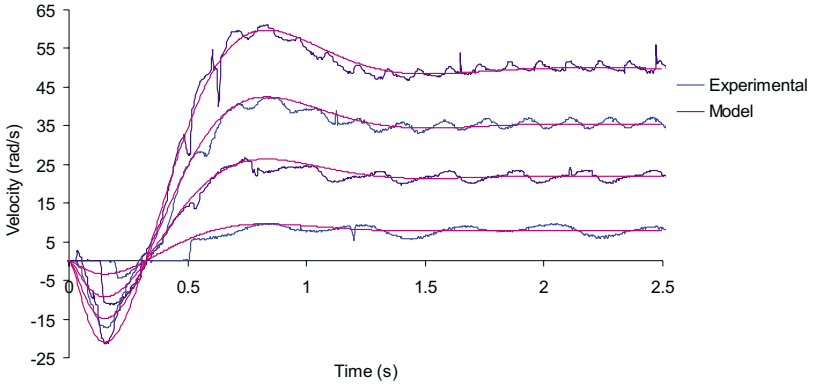


FIGURE 4.23: Step responses of the stage 1 model against experimental data

that best fits the bode plots in order to best fit the experimental data. These slightly differing models are given in Appendix F. As expected, when compared to the time-based simulation results (Figures 4.16 and 4.17), the results show that the frequency-based model is less well able to model the non-linear plant dynamics. Additional results of repeating sequence and sinewave inputs of varying amplitude and frequency are presented in Appendix E.

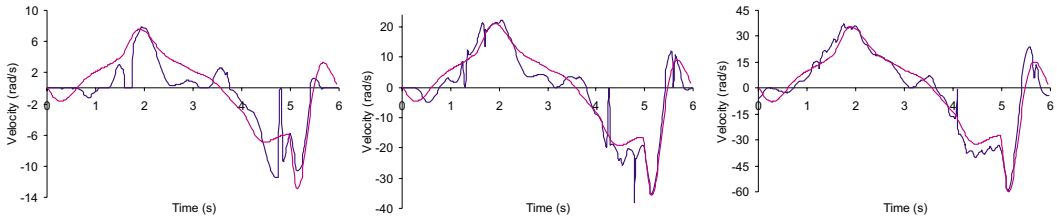


FIGURE 4.24: 10 UPM repeating sequence demands, 1, 2 and 3V amplitudes

4.3.3 Stage 2 Linear Model

A forth order approximation is sufficient to describe the bode plot data of the stage 2 system given in Equation 4.31, its units are in radV^{-1}.

$$G_{stage2}(s) = e^{-0.06s} \frac{1202(4-s)}{s(s+9)(s^2+12s+56.25)} \tag{4.32}$$

The bode plot of this system together with the corresponding experimental results is given in Figure 4.25.

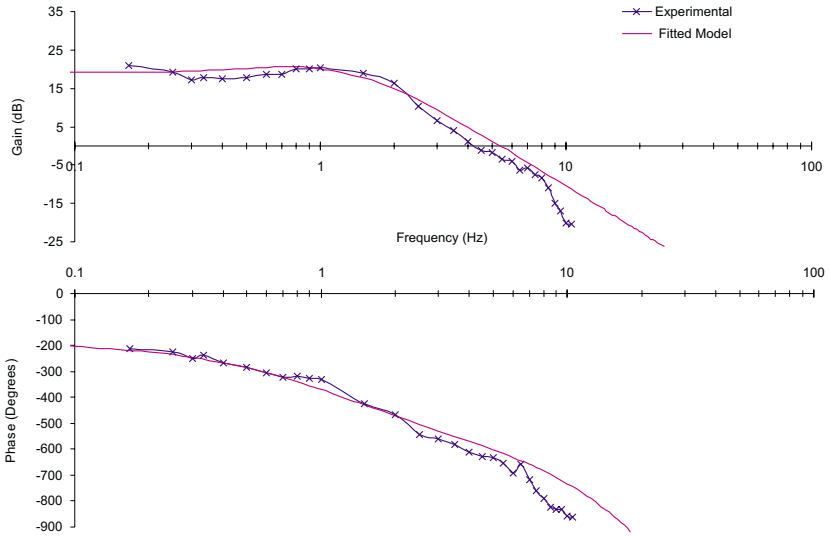


FIGURE 4.25: Experimental and modelled stage 2 frequency response

4.3.3.1 Model Validation

Step response results for the stage 2 model are given in Figure 4.26 and Figure 4.27 shows responses to a repeating sequence input. Again slight adjustments have been made to the model that best fits the bode plots in order to best fit the experimental data. These slightly differing models are given in Appendix F. The results again show that the linear model produces less accurate results than those of its time-based counterpart (Figures 4.18 and 4.19). When also compared to results obtained using frequency model of the

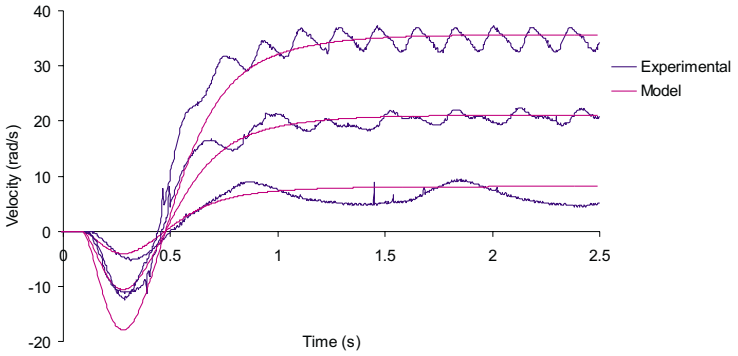


FIGURE 4.26: Step responses of the stage 2 model against experimental data

stage 1 system (Figures 4.23 and 4.24), it is clear that less accuracy has been achieved. This is due to the addition of non-linear components in the drive-train which causes modelling errors and non-linear characteristics to be both augmented and magnified. Additional results of repeating sequence and sinewave inputs of varying amplitude and

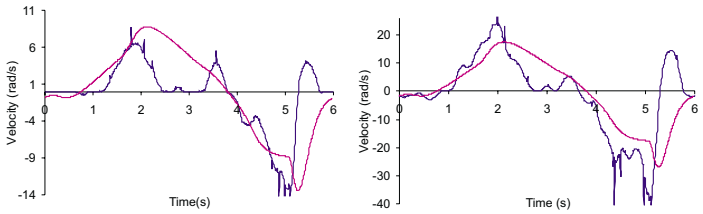


FIGURE 4.27: 10 UPM repeating sequence demands, 1 and 2V amplitudes

frequency are presented in Appendix E.

4.4 PID Control

The majority of ILC schemes are based on the assumption that the plant is stable and some explicitly propose the use of a feedback controller. This is most easily achieved with the use of a proportional, integral and differential (PID) controller, the control method most favoured in industry. This method is very simple to implement discretely, and selection of parameters can be achieved by using the plant models. The tuning of the controller gains using the plant models can therefore be seen as a form of model validation since the resulting gains can be verified against those found to be sucessful in practice.

Research into producing an auto-tuner to find optimal gains has been a very active area over the last few years, and this provides an inevitable link with the area of Iterative Identification and Control (II&C) where the problem of identifying the system model and choosing parameters for the control of the system are combined (i.e (Veres, 1999)).

Due to time constraints and the complexity of the II&C schemes, these methods have not been applied to the present system.

4.4.1 Classical PID Control

The traditional three term controller is given in the continuous domain by:

$$\begin{aligned} u(t) &= K_p e(t) + \frac{K_i}{T_i}\int e(t)dt + K_d T \frac{d}{dt}e(t) \\ &= u(t) + u_i(t) + u_d(t) \end{aligned} \tag{4.33}$$

The digital equivalent of this, given in its positional form is:

$$\begin{aligned} u_p(i) &= K_p e(i) \\ u_i(i) &= u(i-1) + \frac{K_i T_s}{2T_i}[e(i) + e(i-1)] \\ u_d(i) &= \frac{K_d T_d}{T_s}[e(i) - e(i-1)] \end{aligned} \tag{4.34}$$

where K_p, K_i and K_d are the proportional, integral and derivative gains and T_s, T_i and T_d are the sampling, integral and derivative periods.

The sampling frequency of 2.5kHz has been selected for the proportional and integral sections of the controller, making $T_s = 1/2500$ and $T_i = 1$. The frequency of the differential action has been reduced to 500Hz ($T_d = 5$) for several reasons. Due to the limited resolution of the encoders, the differential resolution is extremely limited. Increasing T_d allows more encoder pulses to be recorded between each sample time but of course reduces the time resolution. This compromise can be seen via analysis of the linear model in the frequency domain. Smaller differential frequencies increase the mid-to-low closed-loop gain at the cost of reduced phase and gain margins and system bandwidth. This is reinforced in the time-domain by the increasingly underdamped step responses seen at low differential frequencies. The full model was simulated with a variety of demands and the effect of increasing the differential frequency from 500 Hz to 2.5kHz saw less than 1% degradation in MSE performance. This indicates that the effect is quite small, especially using the small K_d values that the system requires of a three term controller. Perhaps the greatest motivation for the use of the smaller frequency is its consequence of reducing the effect of the noise present in the differential error signal.

4.4.2 Stage 1 PID Control

The Zeiger-Nichols method of PID tuning (Zeigler and Nichols, 1942) was found to produce unsatisfactory results. The values of K_i and K_d were excessively large and produced a wildly unstable system. Manual tuning of the parameters was therefore

undertaken, using root-locus design as a starting point. Once a range of suitable values was found, they were simulated using the time based model before being used on the actual stage 1 system. In both cases a cost comprising the total output error for the demand was calculated by the controller once the system had settled. This cost is given by

$$J_k = \sum_{n=1}^{N} |y_d(n) - y_k(n)| \tag{4.35}$$

where N is the number of samples in each trial, y_d the demand and y_k the output on the k^{th} trial. The PID values were chosen in order to minimise this cost. Unfortunately the natural time-delay of the output, caused by the system's lag at low frequencies, meant that the cost was only reduced by using such large controller values that the response was unstable. Figure 4.28 a) illustrates this effect and was produced using the time-based model. Figure 4.28 b) shows the advantage of tuning the PID gains in order to minimise the cost term given in Equation 4.36 where $y_d(i) = y_d(i+N)$. This means that the demand can be shifted back in time until a minimum error is reached. The output follows the shape of the demand closely, albeit a displaced copy. The error is 4.5 times less than that of the unshifted version, and the velocity no longer reaches unacceptable magnitudes

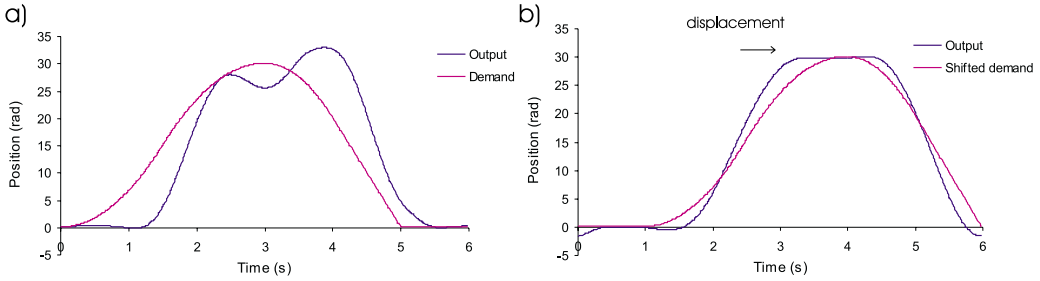


FIGURE 4.28: Simulations illustrating tuning of PID with and without a shift in demand

$$J_k = \sum_{i=1}^{N} |y_d(i+\gamma) - y_k(i)| \qquad \gamma = 1, 2, ...N - i \tag{4.36}$$

The final PID values for a for a subset of 18 demands are given in Appendix G. These encompass sinewaves and repeating sequences of 3 different unit rates, and 3 different amplitudes. The 20 UPM versions are shown in Figure 4.29. Each of these position demands has a maximum velocity of 10 rads^{-1}. Although 20 and 30 rads^{-1} versions have been tuned, it has been found that they are too large to be safely used with ILC. The demands shown will be refered to as the sinewave, R1 and R2 demands hereafter for conciseness. Figure 4.30 confirms the effectiveness of the tuning approach used, showing that the output resembles a shifted copy of the demand as closely as can be reasonably expected.

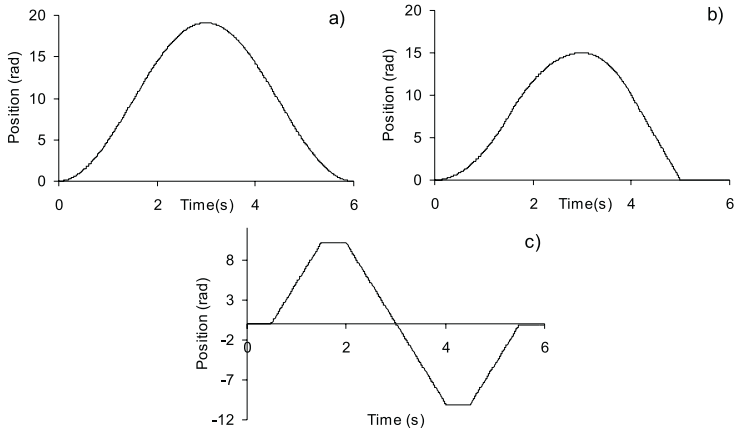


FIGURE 4.29: 10 rad/s amplitude 20 UPM demands

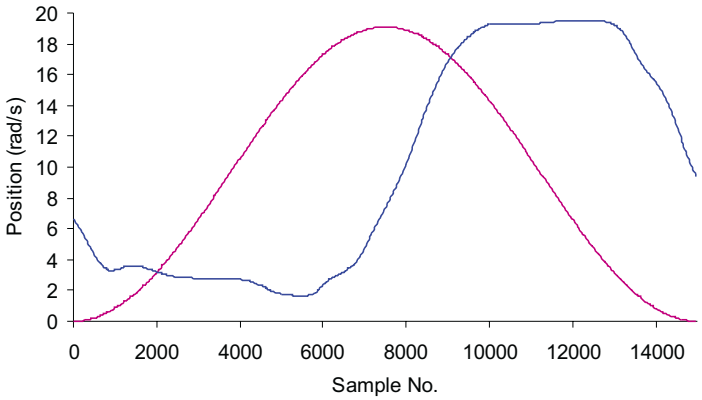


FIGURE 4.30: PID experimental data from sinewave demand at 10 UPM and 10 rads^{-1}

4.4.3 Stage 2 PID Control

PID controllers were tuned for the stage 2 plant in exactly the same manner as the stage 1 plant. The tuning parameters are shown in Appendix G.

4.5 Summary

A time-based model of both the stage 1 and stage 2 plant has been produced by decomposing the system into individual elements and producing mathematical expressions to capture the characteristics of each. The interaction between these components has been established and the resulting models have been and verified against experimental step response and demand response data. Linear models have been produced for the stage 1 and stage 2 plants using frequency response experimental data. The accuracy of these models has also been verified using experimental step response and demand response data. The derivation of the time-based model for the stage 3 system has also

been given. PID controllers have been tuned for the two plants that will be used to test ILC strategies. A cost function has been used which allows the shape of the demand to be followed at an arbitrary time-delay, thereby avoiding excessive gain values. The success of the PID gains found by simulation using the plant models has provided an alternative form of model verification.

Chapter 5

Basic ILC Algorithms

5.1 Introduction

Before considering more recent developments it is imporant to examine the performance of the basic algorithms that were first proposed by Arimoto, Miyazaki, Kawamura and Tamaki on the system. P-type, D-type and Delay-type algorithms are a natural starting point, and these can readily be extended to include higher orders, and also combined with several error cut-off techniques to aid convergence and stability. This section presents experimental results and investigates the effect of parameter variation. The algorithms have been implemented on the stage 1 system. The performance of the algorithms will establish a benchmark against which other learning stategies may be compared.

It is generally assumed that ILC trials are performed with intervals inbetween which allow the system states to return to some constant value. The main consequence of this action is to either necessitate a sudden control action at the beginning of each trial in order to arrive at the correct states, or to force the trajectory to coincide with the initial states. Since this has little impact on the ILC mechanism, in steady state at least, the resetting of the states has been omitted from all the experiments conducted unless expressly stated otherwise. This effectively means RC rather than ILC is being applied but the close relationship that exists between the two disciplines means results from one are relavant to the other and that the same convergence conditions can be applied (Longman, 2000). This also saves a large amount of time in performing a complete set of tests but forsakes the luxury of between-cycle calculation.

5.2 P-type ILC

P-type ILC, being perhaps the seminal ILC algorithm (Arimoto et al., 1984a), was examined first, the discrete law used is given by

$$\begin{aligned} u_{k+1}(i) &= u_k(i) + \phi e_k(i) \\ e_k(i) &= y_d(i) - y_k(i) \end{aligned} \tag{5.1}$$

where $u_k(i)$ is the control input on k^{th} trial at the i^{th} sample, $e_k(i)$ is the error, $y_d(i)$ is the desired plant output, $y_k(i)$ is the actual plant output and ϕ is a scalar gain. $e_k(i+1)$ has been used instead of $e_k(i+1)$ since a sample delay is required to counter the one time step delay through a differential equation when fed by a zero order hold. Since u_k is initially equal to the demand y_d, the P-type law can be interpreted as the demand for trial $k+1$ being made from the original demand plus the integral of all the errors up to and including trial k.

5.2.1 Results

The control algorithm was applied 'trial to trial' until instability made it impossible to continue without harming the test-bed. Figures 5.1, 5.2 and 5.3 show how the error evolves as the trial number increases using the 10, 15 and 20 UPM sinewave demands and a variety of gains. The error has been normalised to take into account the relative difficulty of each demand profile to produce the normalised error (NE), calculated using

$$\frac{\sum_{i=1}^{N} |y_d(i) - y_k(i)|}{\sum_{n=1}^{N} |y_d(i)|} \tag{5.2}$$

A NE of 1, for instance, would occur if the output was continually zero whatever the demand used. Although this expression helps counter the misleadingly large error of a long profile, it can do nothing to obviate the large error associated with a quickly changing demand, nor should it.

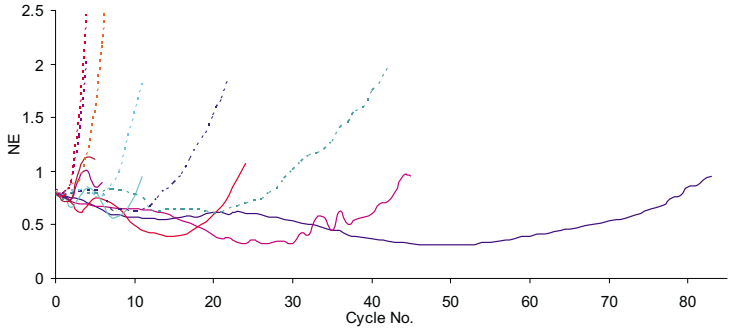


FIGURE 5.1: P-type error results for 10 UPM sinewave demand

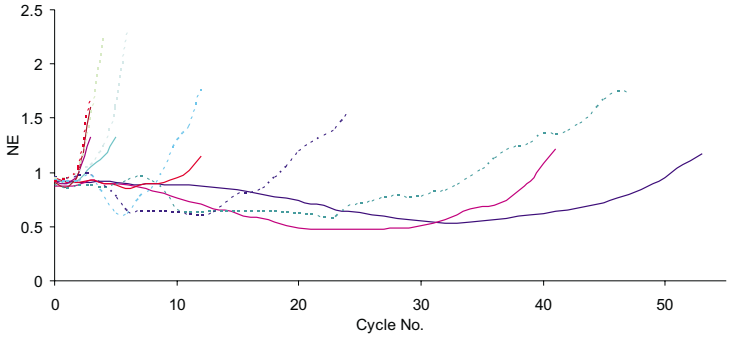


FIGURE 5.2: P-type error results for 15 UPM sinewave demand

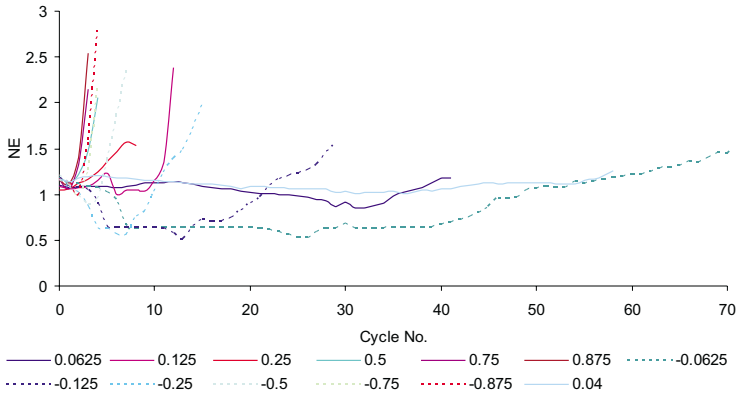


FIGURE 5.3: P-type error results for 20 UPM sinewave demand

The initial cycle (cycle 0) is not influenced by the ILC update but its effect appears during the next cycle. The gain values (ϕ) used are colour-coded and set out in the key shown in Figure 5.3. Changing the demand profile yields similar results and those obtained using the R1 demand are shown in Appendix H.1.

The increasing NE of the first cycle shows that the PID controller becomes less effective as the unit rate increases. Negative gains are investigated in accordance with the theory that the gain should have the same sign as the system matrix multiplication CB. Although they appear to produce superior results at higher unit rates, this is achieved only by the output assuming a fixed position and barely moving at all. The demand profile and corresponding output for a case using the negative gain is given in Figure 5.4 to show the general effect. Three important influences on performance can be observed from the results obtained;

- As the unit rate increases the number of trials until instability (T_{ins}) decreases and the minimum error increases.
- For repeating sequences and high unit rates T_{ins} decreases and the minimum error increases.

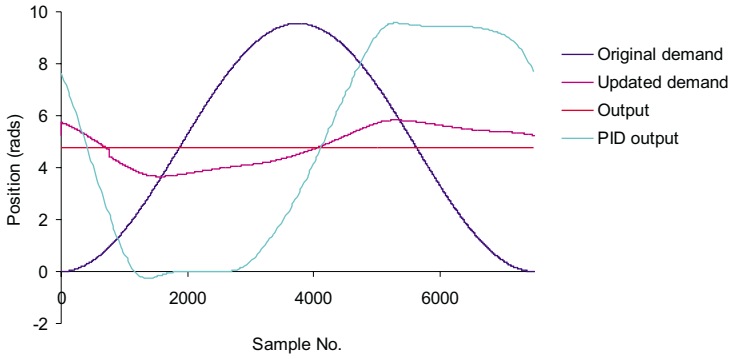


FIGURE 5.4: Data recorded during cycle 15 of 20 UPM sinewave demand with $\phi = -0.0625$

- After a certain value is reached, the effect of further decreasing the gain does not result in any further decrease in the minimum error.

Figure 5.5 shows results from the trial with the minimum cycle error. The output before learning, 'PID output', is included for comparison and is shown to significantly lag the demand. Use of P-type ILC removes the lag of the output but the original demand is not followed well. The updated demand is very oscillatory. In both cases the output

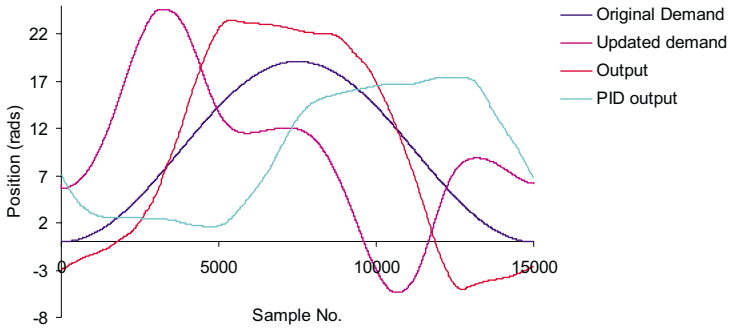


FIGURE 5.5: Data recorded during cycle 50 of 10 UPM sinewave demand with $\phi = 0.0625$

using just a PID controller is given for comparison. Figure 5.6 shows the change in the output as the trial number increases. The interval between trials shown is 10 cycles, and the development of instability is clearly observed.

5.3 D-type ILC

D-type ILC has close links with P-type ILC and was formulated very soon afterwards (Arimoto et al., 1985a). It is given in its discrete form by,

$$u_{k+1}(i) = u_k(i) + \phi(e_k(i+1) - e_k(i))/T \tag{5.3}$$

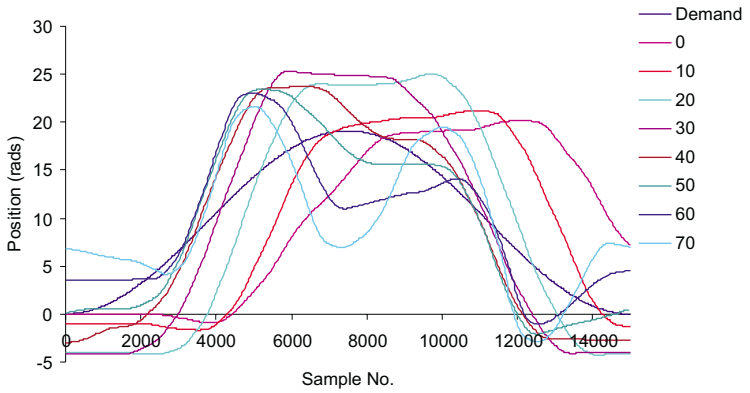


FIGURE 5.6: Output evolution of 10 UPM sinewave demand with $\phi = 0.0625$

where T is the sampling period, and it was the next algorithm to be tested. A single sample delay has again been used with both error terms for causality.

5.3.1 Results

Figures 5.7, 5.8 and 5.9 show results using 10, 15 and 20 UPM sinewave demands. Four gains are examined in this case, and results for the R1 demand are given in Appendix H.2. The results are an improvement over P-type, especially at higher unit rates. The

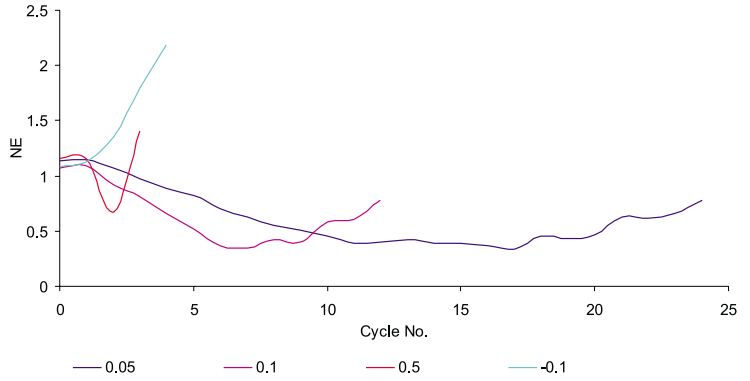


FIGURE 5.7: D-type error results for 10 UPM sinewave demand

drawback is seen in the form of more unstable behaviour and higher velocities which cause the tests to be terminated while still at a fairly low NE. The oscillatory nature caused by the D-type controller is shown in Figure 5.10. The same observations as noted for the P-type law, concerning changing the gain, unit rate and demand, are again relevant. Comparing D-type to P-type ILC also reveals some important features:

- T_{ins} is significantly reduced for all demands, especially those at high unit rates.
- The value of the minimum error is slightly reduced for all demands, more so for repeating sequences and higher unit rates.

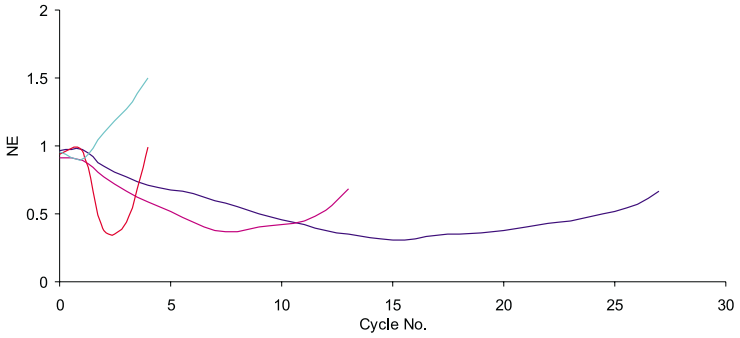


FIGURE 5.8: D-type error results for 15 UPM sinewave demand

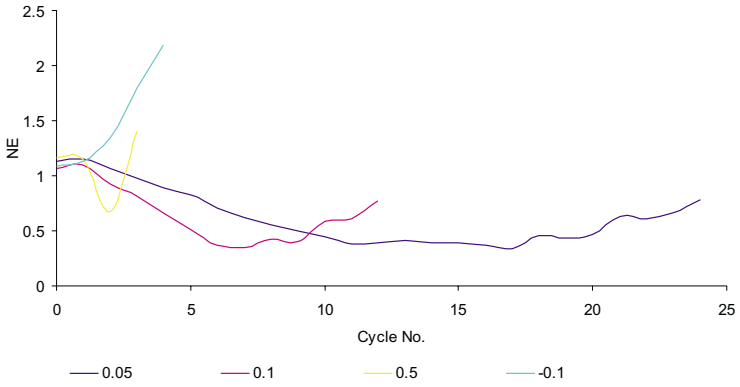


FIGURE 5.9: D-type error results for 20 UPM sinewave demand

Therefore D-type ILC is found to improve the error at the expense of the stability. Figure 5.10 shows the signals from the trial with the optimum performance, again with the PID output (equal to the output at trial 0) shown as a reference. The output oscillates

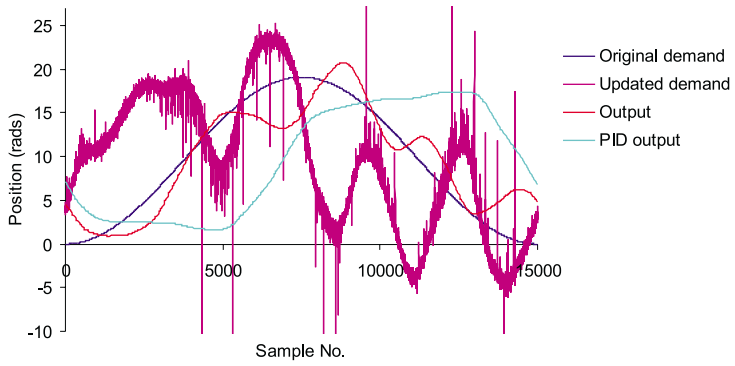


FIGURE 5.10: Data recorded during cycle 30 of 10 UPM sinewave demand with $\phi = 0.05$

around the demand, but at the expense of a highly oscillatory updated demand. The demand also suffers from a large amount of noise due to the differentiation of the error. Figure 5.11 shows how the output becomes unstable as the trial number increases due to increasingly high amplitude oscillations.

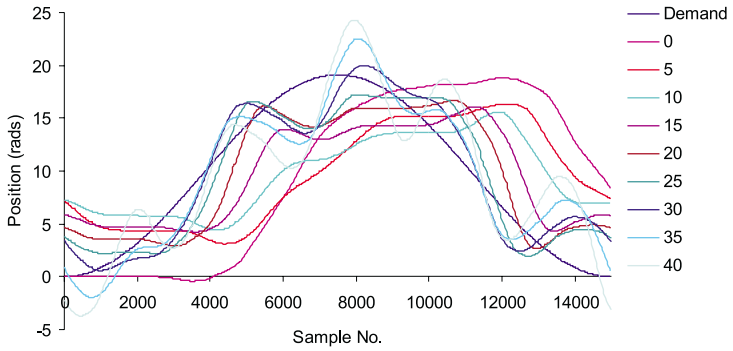


FIGURE 5.11: Output evolution for 10 UPM sinewave demand with $\phi = 0.05$

5.4 Delay-Type ILC

The use of error data from sampling instants ahead in time has gradually evolved, starting with 1 sample instant ahead in the case of D-type. It has been conjectured that the error differential that corresponded to the relative degree of the plant is needed in order to produce convergence. This makes sense when one remembers that the ideal update is the plant inverse multiplied by the error. In the discrete domain this leads to requiring samples n steps ahead, where n is the relative degree of the system. The notion of looking further ahead was only considered when examining systems with time-delays. The idea of finding the 'delay' of the plant by shifting an input signal until it most closely matched the output is relatively new and has little theoretical backup but has been found to produce good results (Barton et al., 2000). The technique will be used here. It is formally given as

$$u_{k+1}(i) = u_k(i) + \phi e_k(i + \tau) \tag{5.4}$$

where τ denotes the delay of the system in sampling instants. It has been refered to as 'Delay-type' due to its similarity is structure with the previous algorithms. In the following tests there is a preliminary cycle (cycle 0), input and output data is recorded during the next cycle, the optimum shift is then calculated during the third cycle and the ILC strategy then begins during the cycle after that. The results of the controller are first seen in cycle 4.

5.4.1 Results

Figures 5.12, 5.13 and 5.14 show results for the Sine demand and results for the R1 demand are given in Appendix H.3, the results are again plotted for a range of ILC gains. The results are a great improvement over the other strategies, drastically reducing the error to very low levels. Surprisingly the error of the higher unit rates is reduced to the level of, or even below that of the lower unit rates. Table 5.1 summarises the number of

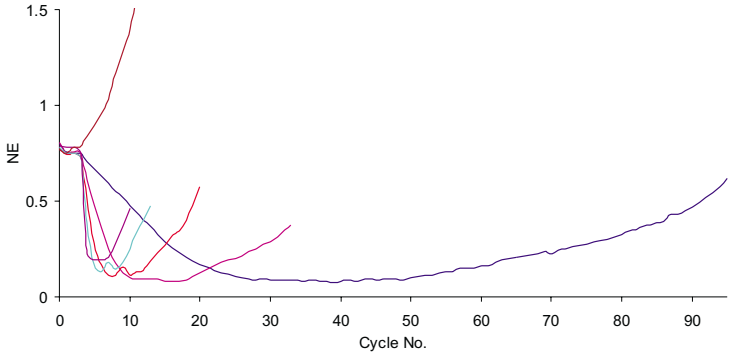


FIGURE 5.12: Delay-type error results for 10 UPM sinewave demand

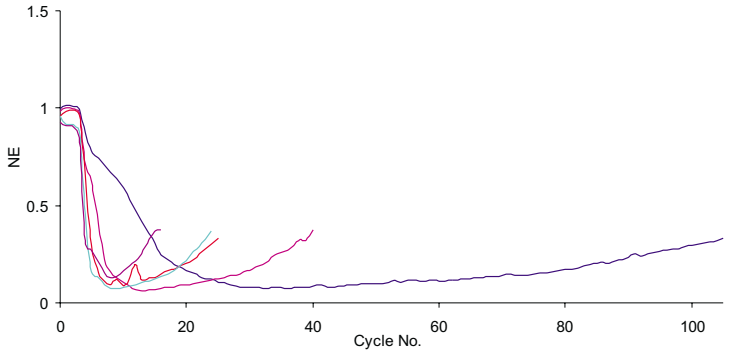


FIGURE 5.13: Delay-type error results for 15 UPM sinewave demand

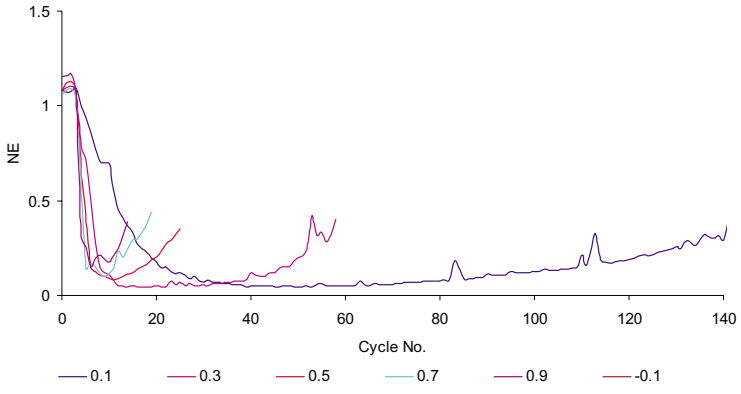


FIGURE 5.14: Delay-type error results for 20 UPM sinewave demand

sample instants delay that was calculated for each demand. The corresponding time in seconds is given in brackets. Figure 5.15 shows data from the best performing 10 UPM

	Time delay /samples		
Demand	10 UPM	20 UPM	30 UPM
1V Sine demand	3100 (1.24)	2700 (1.08)	2225 (0.89)
1V R1 demand	3625 (1.45)	2425 (0.97)	2575 (1.03)

TABLE 5.1: Delays between demand and system output as used in Delay-type ILC

trial, Figure 5.16 shows data from the best performing 20 UPM trial. Figures 5.17

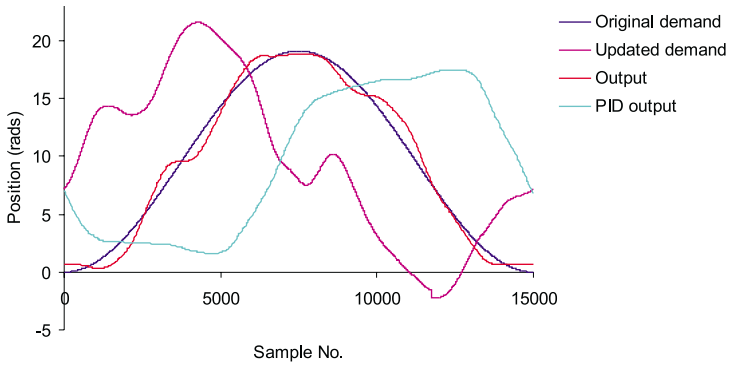


FIGURE 5.15: Data recorded during cycle 40 of a 10 UPM sinewave with $\phi = 0.1$

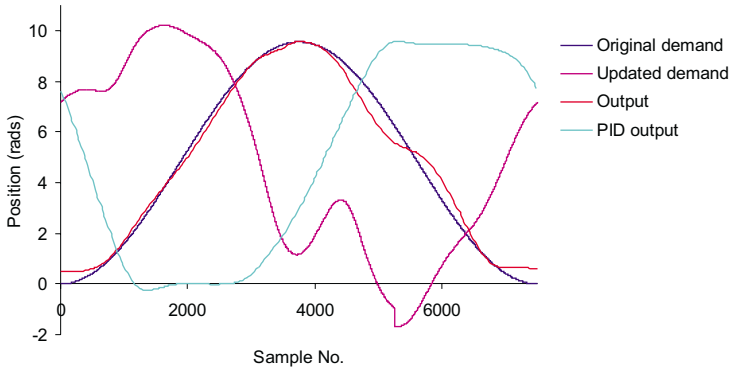


FIGURE 5.16: Data recorded during cycle 50 of a 20 UPM sinewave with $\phi = 0.1$

and 5.18 show the error evolution in each case, the former continues until the error is minimised, while the latter continues to show the process of instability.

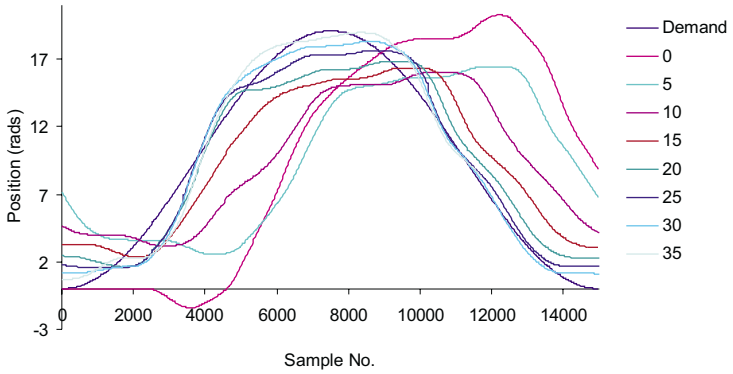


FIGURE 5.17: Output evolution for 10 UPM sinewave demand with $\phi = 0.1$

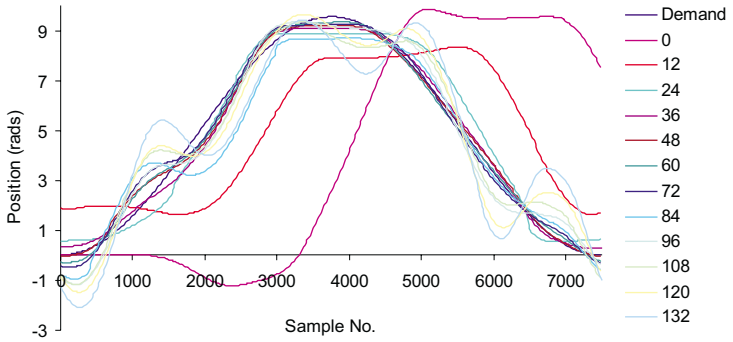


FIGURE 5.18: Output evolution for 20 UPM sinewave demand with $\phi = 0.1$

5.5 Summary

Simple ILC laws have been implemented on the stage 1 system. Whilst the P-type and D-type laws have proved largely ineffective, much greater success has been seen using the Delay-type algorithm. In this case the error has been reduced to much lower levels than the P-type and D-type for all the demands used and the number of trials until instability halts the test has been greatly increased, more than quadrupling when using higher unit rates. Certain trends with respect to the effect of unit rate and gain variation have been identified and discussed. Chapters 7 and 10 apply analytic methods in order to explain the performance of the algorithms that has been observed.

Chapter 6

Phase-lead Algorithm

6.1 Introduction

From initial experiments conducted on the stage 1 system, described in Chapter 5, it is apparent that the Delay-type algorithm outperformed the P-type and D-type laws. If the Delay-type algorithm given by Equation 5.4 is generalised to include all possible sample instances, it becomes

$$u_{k+1}(i) = u_k(i) + \phi e_k(i + \lambda) \qquad \lambda = 0, 1, ...N - 1 \tag{6.1}$$

where λ is the number of samples ahead, ϕ is the learning gain, $u_k(i)$ the control signal at the k^{th} iteration and N is the number of samples in the demand. This is sometimes referred to as phase-lead ILC and was introduced in (Wang and Longman, 1996). Equation 6.1 can be made continuous through the addition of the sample time T to give

$$u_{k+1}(Ti) = u_k(Ti) + \phi e_k(T(i + \lambda)) \tag{6.2}$$

As has been explained, λ cannot be regarded as being the total phase-lead since a single sample is required to make the update causal. In order to simplify further expressions it will be assumed that λ equals the phase-lead minus one. The phase-lead law was implemented on both the stage 1 and stage 2 systems in order to compare their relative performance. This is of interest since the stage 2 system is of higher order and has increased non-linearity, as demonstated in Chapter 4. It is also important to establish that the algorithm is not specific to a single plant and can be generally applied.

6.2 Stage 1 System

Figures 6.1, 6.2 and 6.3 show error results for 4 choices of the gain, ϕ, using a 10 UPM sinewave demand and the stage 1 system. Figure 6.1 corresponds to $\phi = 0.1$, Figure

6.2 to $\phi = 0.3$, Figure 6.3 to $\phi = 0.5$ and Figure 6.4 to $\phi = 0.7$. Error results for the R1 and R2 demands are given in Appendix I.1. Each experiment is stopped when the trial number reaches 400 or by the onset of instability, whichever occurs sooner. Since instability is associated with derivatives of the output signal, it can still occur at fairly low NE levels. Several points can be concluded from the collective results of these

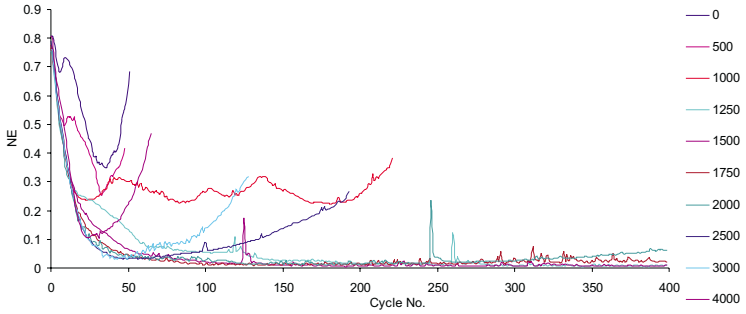


FIGURE 6.1: Error results for 10 UPM sinewave demand with variable λ and $\phi = 0.1$

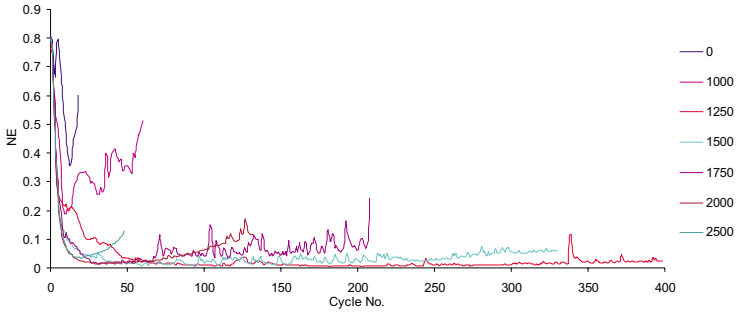


FIGURE 6.2: Phase-lead error results for 10 UPM sinewave demand with variable λ and $\phi = 0.3$

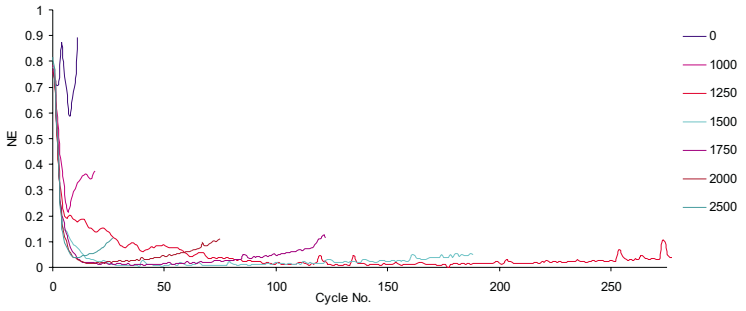


FIGURE 6.3: Phase-lead error results for 10 UPM sinewave demand with variable λ and $\phi = 0.5$

experiments. The first do not depend on the demand or unit rate:

- It is quickly seen that the λ values used by the Delay-type algorithm (and given in Section 5.4.1) are not the optimum values. This is surprising as these were seen

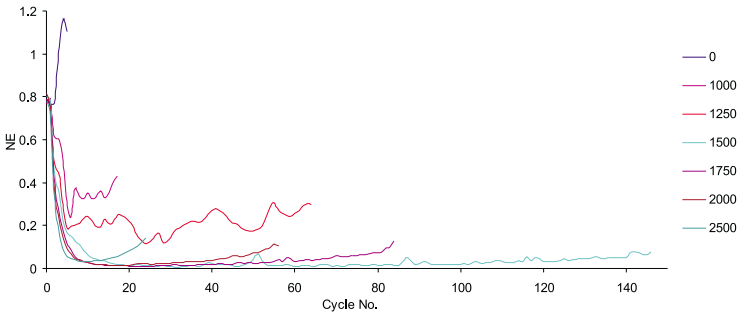


FIGURE 6.4: Phase-lead error results for 10 UPM sinewave demand with variable λ and $\phi = 0.7$

as the most intuitive values to take. The actual optimum phase-lead observed in each case (λ_{opt}) reduces from approximately 1250 to 1500 samples as ϕ increases. This appears to be independent of unit rate and demand profile. As the sample frequency is 2500, these correspond to 0.5 and 0.6 seconds.

- T_{ins} appears to be inversely proportional to the gain ϕ.
- As the unit rate increases the value of T_{ins} is reduced. There is a much greater difference between 10 and 15 UPM and 15 and 20 UPM in this respect.
- The repeating sequence has a slightly lower value of T_{ins} for a given ϕ, λ and unit rate. Also the values of λ_{opt} are all slightly lower than for the sinewave demand for given values of ϕ and unit rate.
- All the previous P-type and Delay-type results are of course included in the variable phase lead results, and they are also able to explain a peculiarity of the Delay-type results. It was observed that the minimum error actually reduces and T_{ins} increases as the unit rate increases under Delay-type control. This is explained since the Delay-type delay reduced as the unit rate increased, and approached the optimum number of samples that has been established. This effect overcomes the loss of performance at faster unit rates.

In an attempt to explain this lead, the impulse responses shown in Figure 6.5 have been examined. The first shows a generic first order response, the second a higher order response, and the third is the response of the stage 1 system. If the response were to be so simplified as to be itself a single impulse, they would occur at 0, m and n seconds for a), b) and c) respectively. Therefore the most accurate single impulse model of the inverse of these systems occurs at the times 0, m and n seconds before the output. Although this is an imprecise inversion, it approximates the method by which the simple structure algorithms function. P-type ILC works well on first order systems because they have the property that the error at sample i is most directly due to the input at the same instant. The success of phase-lead ILC is therefore evident; if the time taken for

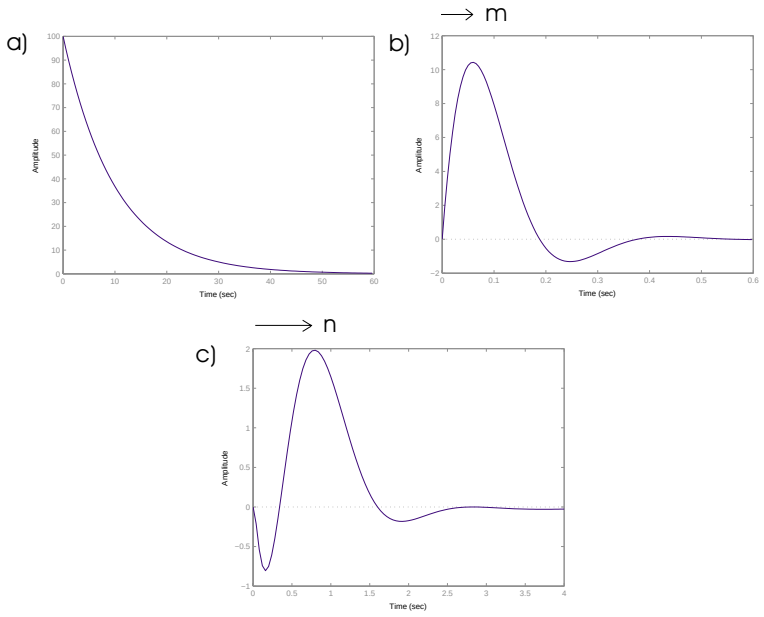


FIGURE 6.5: Impulse Responses of various systems

the maximum impulse response peak can be found and used as λ in the phase-lead law, then it should be as successful as P-type is for first order systems. Unfortunately this is not the case. The value n for the non-minimum phase system is found to be 1950 samples using the system model, well above the experimentally achieved optimum of 1500. When modeling phase-lead ILC on higher order systems, the optimum in terms of both convergence speed and minimum error has also consistently been found to be slightly below the value of m. The minimum error is also never zero, and divergence always occurs.

Choosing the phase lead in accordance with the maximum impulse response value is a simplification of a more general update: the case in which corrections are made at all sampling instants before the error, and the amount of correction dictated by the magnitude, at the error, of a system impulse response originating at the correction point. This means that corrections are made at every point that could possibly have been responsible for the error, and the amount of the correction is related to its 'measure of responsibility'. It will be seen that this actually equates to the G^T update, applied in Chapter 8.

Figure 6.6 shows data recorded during the best performing cycle of phase-lead ILC. The demand is followed closely, although the updated demand is quite oscillatory. Figure 6.7 shows how the output signal changes as the number of trials increases up until the lowest error of the test. Data from other experiments shows that oscillations go onto grow in the updated demand and in the output until their velocity becomes too great for the testbed. The failure of phase-lead ILC to converge to zero and remain there can, however, be explained. Figure 6.8 illustrates the failure mechanism that occurs when using phase-lead ILC; oscillations of a certain frequency grow gradually until they force

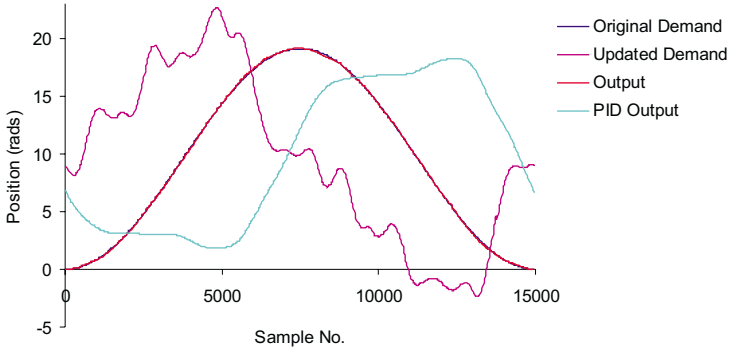


FIGURE 6.6: Data recorded during cycle 296 of 10 UPM sinewave demand with $\phi = 0.1$ and $\lambda = 1250$

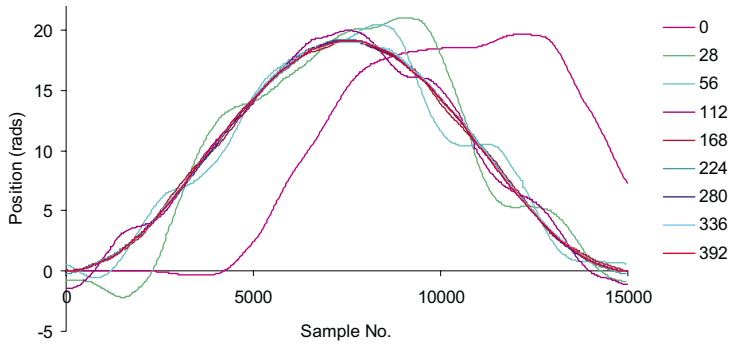


FIGURE 6.7: Output evolution of 10 UPM sinewave demand with $\phi = 0.1$ and $\lambda = 1250$

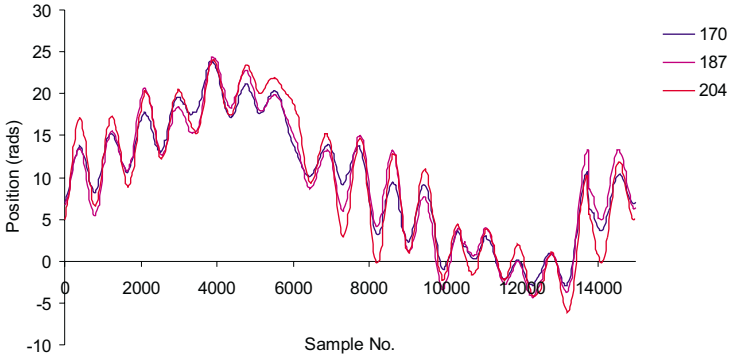


FIGURE 6.8: Output evolution of 10 UPM sinewave demand with $\phi = 0.5$ and $\lambda = 1250$

the output position, and hence velocity, to become unmanageable. Analysis of results using different gains and phase-leads yields the following conjecture:

- The frequency of the destabilizing oscillations (f) is only dependent on the phase-lead used, and can be estimated using Equation 6.3

$$\frac{1}{f}(\angle f - 180) = \frac{\lambda}{f_s} \tag{6.3}$$

where $\angle$f is the phase lag at f, and f_s the sampling frequency. This states that f is the lowest frequency that can be propagated by the phase-lead λ, and Figure 6.9 illustrates how this occurs. If q represents an instant of an oscillatory new demand input, it will

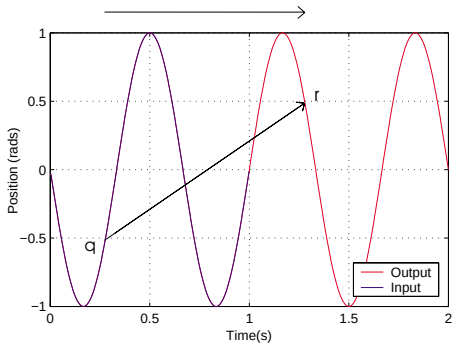


FIGURE 6.9: Propagation of an oscillation

directly affect the value r of the output (with some gain change). If r is larger than the originally specified demand then, by the nature of phase-lead ILC (with lead λ), q will be made increasingly negative. This only succeeds in increasing r in the next trial. The growth of the oscillations in the updated demand is a function of the gain ϕ and the magnitude of the gain at f. Figure 6.10 shows how the phase-lead oscillations can be predicted from the Bode plot of the system. Equation 6.3 is plotted for a range of phase-leads and their intersections with the phase plot show the frequencies of instability that would arise. Since, for the system considered here, instability is caused by the output velocity, as opposed to the actual positional output, the gain plot of $sG(s)$ should be examined instead of $G(s)$. This shows that as the phase-lead reduces from 2500 to 1250 samples, the gain of the velocity decreases from 1.5 to 0.63. This explains why the optimum lead is reduced from 1950 to 1500 samples; there is a compromise between the rate of learning and the rate of increase in the magnitude of oscillations caused by phase-lead ILC.

Figure 6.11 shows the inability of phase-lead ILC in coping with rapidly changing demands, a shortcoming which motivates the use of the filters in the following sections. Even with the optimum phase-lead, the R2 demand cannot be followed accurately for very many trials. Whilst instability can occur rapidly, Figure 6.12 shows how closely the output matches the 20 UPM R2 demand before the previously described oscillations cause instability.

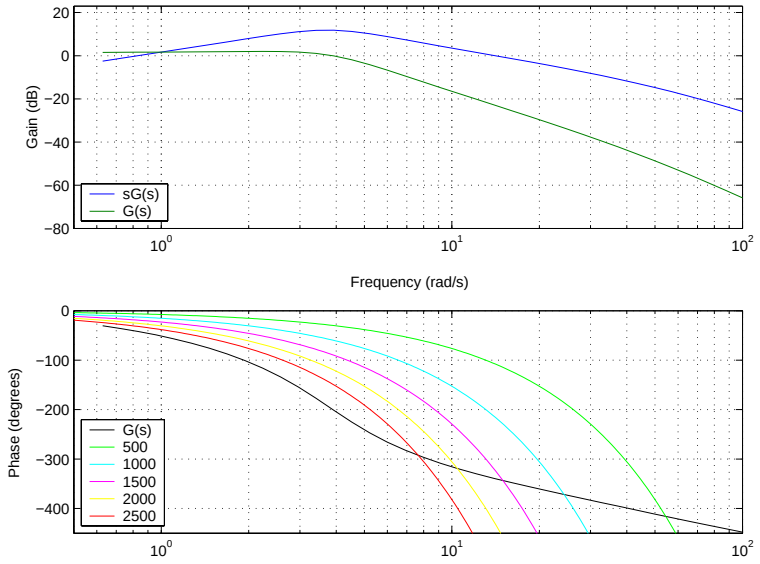


FIGURE 6.10: Bode plot showing intersections of phase-lead lines

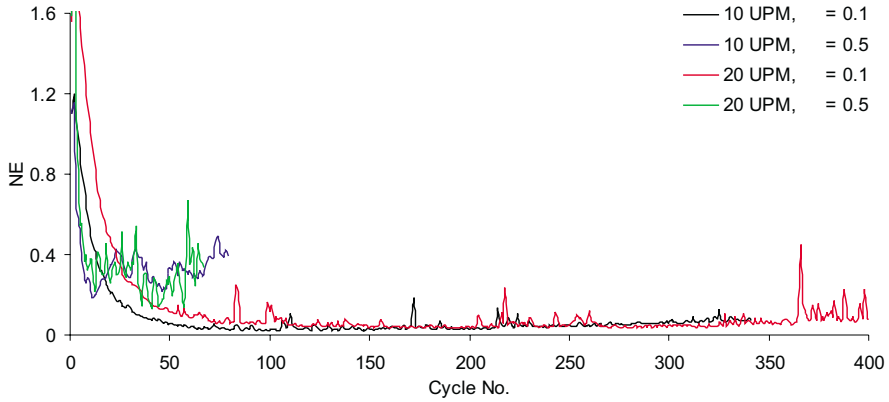


FIGURE 6.11: Phase-lead error results for R2 demand at two unit rates, both with $\phi = 0.1$ and $\lambda = 1500$

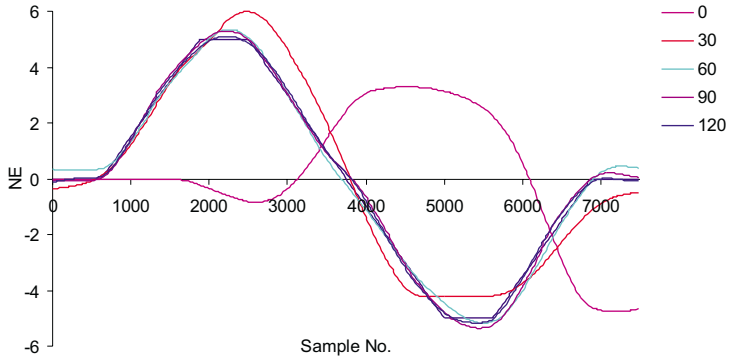


FIGURE 6.12: Output evolution of 20 UPM R2 demand with $\phi = 0.1$ and $\lambda = 1500$

6.2.1 Causal Filters

A filter can either be applied to the error, e_k, or to the input of the plant, u_{k+1}. The only difference is whether the demand itself is filtered, the effect of which will later be investigated. The open-loop system will therefore be considered to be the plant $G(s)$ in series with the filter, $F(s)$. The simplest way to reduce the destabilizing oscillations that have been observed is to use a causal low-pass filter to reduce the magnitude of the Bode plot of $F(s)G(s)$ at the frequency of oscillation. The act of adding a causal filter to the plant, however, changes the phase plot of the system and therefore the frequency at which a given phase-lead intersects with it. Furthermore, it is likely that the impulse response of the system will change also. It is therefore an iterative process to design a causal filter for use with phase lead ILC. Firstly a cut-off is selected below the frequency of unstable oscillations, and a class of filter to implement it. The usual criteria of a sharp cut-off and minimal phase-lag are favourable, although, as yet it is not clear as to their relative importance. Little emphasis has been placed on ripple in the stop-band. The impulse response of $F(s)G(s)$ is then obtained and the number of samples to its maximum determined. A Bode plot of $F(s)G(s)$ is drawn together with a line representing those frequencies that can be propagated by a phase lead of the number of samples calculated, in the same manner as that shown in Figure 6.10. The frequency of unstable oscillations is found by the intersection of this line with the phase plot. This frequency should correspond to the local minima seen on the magnitude plot caused by the filter. This ensures that no undue magnitude (and hence bandwidth) has been sacrificed below the unstable frequency. For a given filter it also ensures that no additional low frequency lag has been added other than what is necessary according to the filter chosen. If no such correspondence occurs, the cut-off must either be moved slightly, or the filter order changed in order to produce more lag and the design process repeated.

Three causal filters have been designed and tested on the system. The first is a 5^{th} order Chebychev lowpass filter with a cut-off of 2.5Hz, and 270° phase lag and 60dB attenuation at the unstable frequency. This filter is a compromise between sharpness of cut-off and lag. The second filter is more aggressive with an extra 90° lag but an additional 22dB attenuation centered on the unstable frequency. The third filter is a 4^{th} order Butterworth band-stop filter which has been selected for its high attenuation over a very small range of frequencies. The attenuation of 60dB is centered on the unstable frequency at a cost of just 90° lag beforehand. This filter was designed in order to maximize the system bandwidth whilst still reducing the effect of the unstable frequency. Experimental results using the causal filters are shown in Appendix I.2. Figure 6.13 shows an unexpected and illuminating effect observed when using the bandstop filter; there appears to be at least two higher frequencies than the unstable frequency which also progressively increase in magnitude as the cycle number increases. This suggests, and frequency analysis confirms it, that along with the unstable frequency identified using

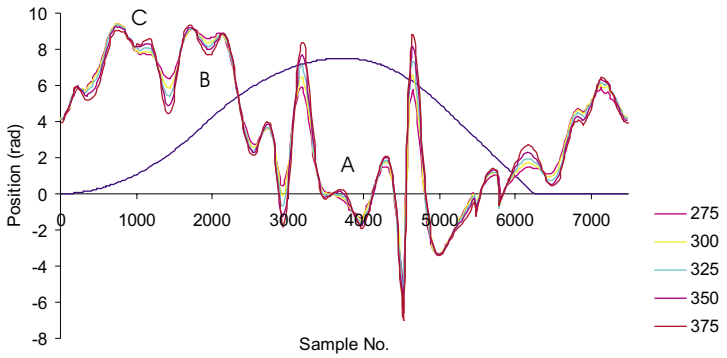


FIGURE 6.13: Output evolution of 20 UPM R1 demand using the bandstop filter with $\phi = 0.3$ and $\lambda = 1500$

Equation 6.3 and a Bode plot of $F(s)G(s)$, there are two other unstable frequencies. The unstable frequencies are 2.3, 2.65, and 5.2Hz. These can readily be explained if the cause of the original unstable oscillation (the largest frequency that can be propagated for a given phase-lead) is extended to include all frequencies that can be propagated. Equation 6.3 can then be updated as:

$$\frac{1}{f}(\angle f - 180(1+2i)) = \frac{\lambda}{f_s} \qquad i = 0, 1, 2, \ldots \tag{6.4}$$

where again f_i is the i^{th} frequency of oscillation, $\angle f_i$ is the phase lag at f_i, and f_s the sampling frequency. The first three instability frequencies, f_1, f_2 and f_3, will be referred to as the pimary, secondary, and tertiary frequencies. Figure 6.14 shows the Bode plot of $F(s)G(s)$ using the bandstop filter with the first three phase-lead lines, generated using Equation 6.4. The gains at the frequencies of intersection are highlighted on the magnitude plot for clarity. The primary, secondary and tertiary frequencies are found to be 2.3, 2.68, and 4.8Hz respectively, closely matching those experimental values observed. The reason for the prominence of these first three unstable frequencies can be seen from the gain plot; the magnitudes that correspond to these frequencies are all similar and close to -40dB. It is because the bandstop filter reduces the primary frequency alone to such a degree that the secondary and tertiary are so visible. Frequencies higher than the tertiary have been rarely observed due to their high attenuation. Further tests have shown that altering the demand profile used does not alter the findings by any great degree. If the demand contains a sizable component of one or more of the unstable frequencies then instability progresses sooner, the updated demand containing components that would have otherwise taken many cycles to build up. Looking at the frequency components present in those demand used, shown in Figure 6.14, it is clear that there are only very small quantities of these frequencies are present in the demands. Although only the 20 UPM demands are shown, the 15 and 10 UPM cases are obtained by multiplying the frequency scale by 0.75 and 0.5 respectively. Because these frequencies are close to the primary frequency, the filter is best located at the

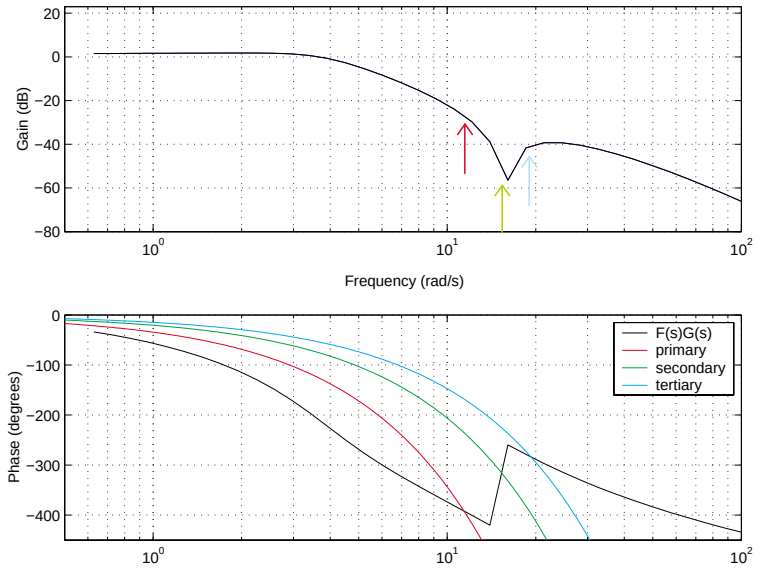


FIGURE 6.14: Bode plot showing intersections of $F(s)G(s)$ with primary, secondary and tertiary frequencies for $\lambda = 1750$

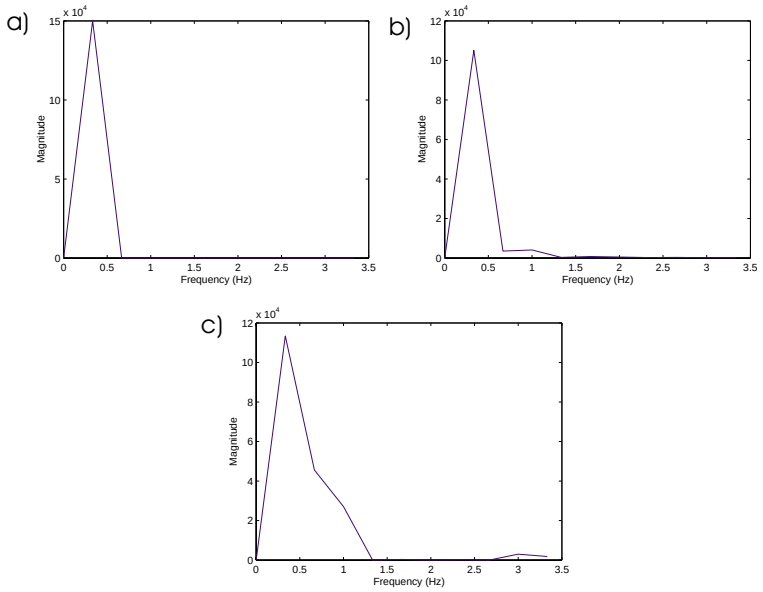


FIGURE 6.15: Power spectrums of the 20 UPM a) sinewave, b) R1 and c) R2 demands

input to the plant. Results have confirmed that performance is worse if the demand is left unfiltered. The exceptions to this rule occur when using non-causal FIR filtering and are discussed in Section 6.2.2.1.

The theoretically best lead using the bandstop filter (found from the impulse response) is 2250 samples. As with the unfiltered case, this is reduced when carried out in practice due to the higher attenuation of the unstable frequencies (the phase lead lines in Figure 6.14 move to the right), and becomes 1750. Two extra objectives can now be put forward in order to improve the design of future causal filters:

- To ensure that the best possible phase lead, derived from the impulse response, is as close as possible to that experimentally determined.
- To seek to move the intersection of the $F(s)G(s)$ phase plot and the optimum phase lead line further towards the right and thus at a higher frequency.

The former task involves reducing the magnitude at the fundamental frequency (and beyond) sufficiently to allow the convergence and stability advantages of using the most favorable phase lead to become more important than the extra attenuation gained by increasing it. Until this is true instability will always govern the process. The second task depends on the first; extra lag produced by a high-order causal filter causes a given phase lead to give rise to unstable frequencies which are slightly higher, and therefore more attenuated, than otherwise. Unfortunately a system with more low frequency lag will usually have an impulse response with a larger number of samples to its maximum value. The bandstop filter raised the fundamental frequency at optimum lead from 1.66Hz to 1.8Hz, and the two low pass filters both raise it to 2.3Hz, which helps to account for their success. It should also not be forgotten, however, that a surfeit of lag before the cut-off point will destabilize the system. The two lowpass filters were designed with these points in mind. The first has a magnitude plot very similar to the bandpass filter, enabling performance comparisons to be made in terms of lag and attenuation above the cut-off alone. The second low-pass filter is similar to the first but with more lag and more attenuation, enabling comparisons with the first to be made on that basis only. Figure 6.16 illustrates the shortcomings of the bandstop filter. Its lower attenuation

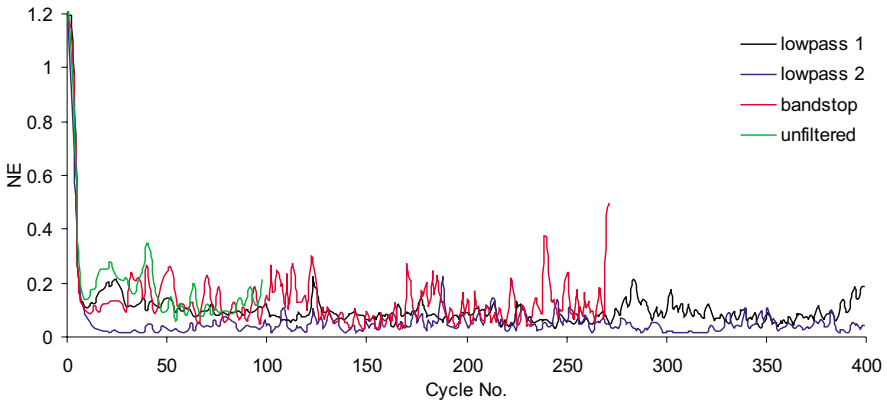


FIGURE 6.16: Phase-lead error results for 20 UPM R1 using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

of the instability frequencies, especially the secondary and tertiary, causes instability. As discussed, it also has lower instability frequencies than the other two filters. The superior performance of the second lowpass filter shows that frequency attenuation is more important than low frequency lag. Instability frequencies only account for a certain amount in explaining the lack of convergence, and it is has been found that the removal of frequencies below the primary improves convergence. This is due to two factors

- The influence of the primary frequency extends a certain amount below that frequency where it grows at a lesser rate or merely disrupts learning.
- High frequencies naturally destabilize the process of learning, more so if they are present in the demand. This makes intuitive sense since all the ILC algorithms are effectively built on the notion of a heavily simplified plant. Attenuating increasingly low frequencies in the plant is a method of simplifying it. The simplified plant then more closely matches that required by the ILC algorithm, and learning is improved.

Without being able to substantially change the unstable frequencies it is impossible to separate these two factors. The emphasis for the need of a precise cut-off is also diminished, the only certain requirement being a large amount of attenuation at the unstable frequencies. Figure 6.17 shows the output of the plant during a very unsatisfactory

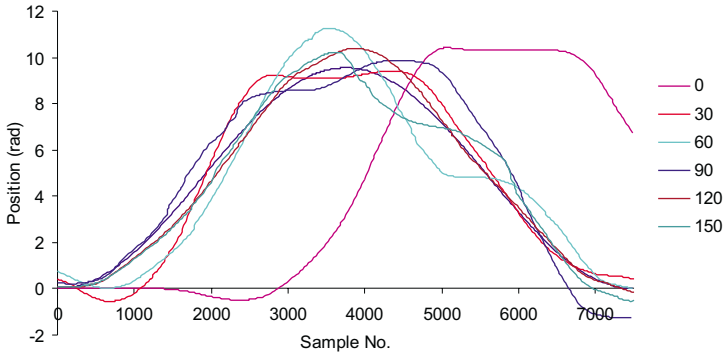


FIGURE 6.17: Output evolution of 20 UPM sinewave demand using the bandstop filter with λ_{opt} and $\phi = 0.1$

period of learning. No unstable frequencies are seen and the phase lead is the experimental optimum, although this has only been found to a resolution of 125 samples. From this and other similar cases it is clear that higher frequencies than those present in the demand disrupt the learning process, and it may be beneficial to select a cut-off frequency only marginally above the highest frequency present in the demand. Since the approximation to the actual plant that exists at the heart of these simple ILC schemes is most accurate at low frequencies, it is likely that these techniques are only capable of learning low frequencies. As the trial number grows, either the integration of the error at each sample caused by this ILC inaccuracy causes instability, or the instability frequencies overcome the attenuation which has been imposed on them to cause instability themselves. Having focused on the inadequacies of phase-lead ILC, Figure 6.18 shows the success of the causal filters that have been implemented. For the second lowpass filter, the learning process is nearly always stable over the 400 cycles that are undertaken with no divergence seen, and convergence is faster than in the non-filtered case. This is true for all the demands used.

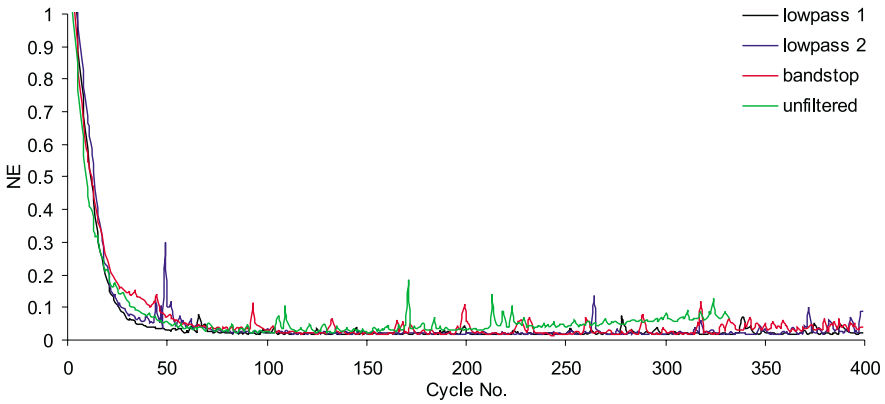


FIGURE 6.18: Phase-lead error results for 10 UPM R2 demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

6.2.2 Non-causal Filters

In order to assess whether additional lag in the system worsens the process of learning and subsequent stability, it is necessary to examine non-causal filters in place of the causal ones already tested. The filter design process is simplified as the oscillation frequencies are unchanged by the addition of the filter. The maximum impulse response is unlikely to be altered and so the design of the filter simply involves reducing the gain at these frequencies. Two classes of non-causal filter have been selected for use; one filter that can be implemented in batch mode, and one that has no such restriction. Although there are several techniques available for batch-mode filtering, the zero-phase IIR filter has been chosen for its simplicity and effectiveness.

6.2.2.1 Linear phase FIR filter with offset

A linear phase FIR filter is produced by creating a non-causal filter of order n, symmetrical about its mid-point(s), and then shifting it $\frac{n}{2}$ (or $\frac{(n+1)}{2}$) samples in order to make it causal. If this last stage is omitted then a zero-phase FIR filter is obtained which has no limitation on having to be performed in batches. This price of the non-recursive operation is a very large order compared with its IIR equivalent. Four filters of this type have been implemented, two lowpass filters and two bandstop. The first lowpass filter is of order $n = 2101$ and has a gain of -36dB at the primary frequency, its magnitude before the cut-off is extremely aggressive, taking a value of -18dB at 0.8Hz. The second lowpass filter has the same magnitude at the fundamental frequency but only -13dB at 0.8Hz, it also has greater attenuation at higher frequencies. The first bandstop filter is of order $n = 2325$ and has extremely high attenuation at low frequencies. At 1.6Hz this is -60dB which reduces slightly to -48dB at the primary frequency. The second bandstop filter has a higher cut-off point making it less aggressive at low frequencies. At 1.6Hz the attenuation is -13dB, increasing to -38dB at the primary frequency. Both bandstop

filters have similar characteristics above this frequency, the upper cut-off being 8Hz. The order of these filters approaches the maximum achievable with the hardware and sampling frequency used, therefore, although the attenuation is satisfactory, the cut-offs are not sharp. It is also advantageous that the filter should not be applied on data that is in the process of being updated, that is $n > 2\lambda$. Figure 6.19 shows how large

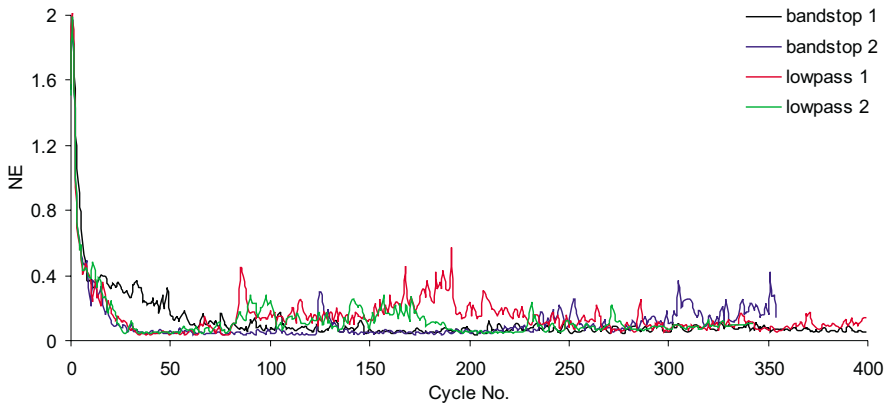


FIGURE 6.19: Phase-lead error results for 20 UPM R2 demand using a variety of non-causal filters, each with λ_{opt} and $\phi = 0.5$

attenuation causes slow convergence, the first bandstop filter taking double the number of cycles to converge in every test performed. Its low frequency attenuation effectively gives it a lower learning gain, ϕ, and increases T_{ins} at the cost of convergence. The effect of aggressive low frequency filtering extends beyond this, however; the large peaks that occur in the plots of NE against trial number are much reduced, even below the values seen in the less aggressive filters with much lower learning gains, ϕ. This means that greater low frequency attenuation produces less deviation in the cycle error. The first bandstop filter is more successful than the first lowpass filter, as, with a limited order, it is able to supply greater attenuation at low frequencies. The more aggressive filters are the only ones to allow the test to last 400 cycles in Figure 6.19. The performance of lowpass and bandstop filters on other demands is shown in Appendix I.3.

Without differences in the phase characteristic confusing the issue, lowpass and bandstop filters can also be compared. All the results obtained show that performance is determined by the amount of the low frequency attenuation. This suggests two points:

- Frequencies above 8Hz play an insignificant role in influencing the performance in the tests conducted
- Short term performance is mainly dictated by the magnitude plot of the system below the primary frequency

Short term performance is taken to include convergence rate and changes in the cycle error between trials. It differentiates between long term effects such as unstable frequencies and the effect of integrating high frequency error inherent in the simple ILC

laws. Long term performance is therefore mostly influenced by the attenuation at the primary frequency and above. Figure 6.20 illustrates these points, showing the most successful non-causal and causal filters of those tried. The non-causal bandstop 1 filter has less cycle error deviation due to its high frequency attenuation and lack of destabilizing phase-lag. This overcomes the advantage of the increased instability frequencies that occur when using causal filters. The less aggressive filters are seen to suffer from divergence during the test.

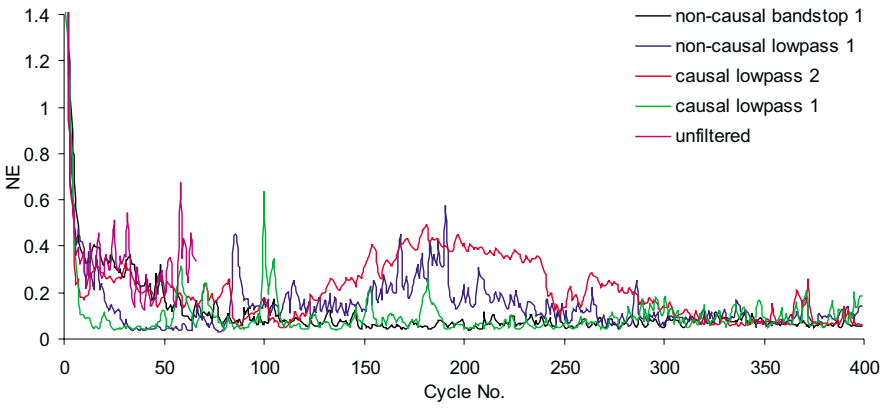


FIGURE 6.20: Phase-lead error results for 20 UPM R2 demand using the most successful filters, each with λ_{opt} and $\phi = 0.5$

6.2.2.2 Zero-phase IIR filter

A filter is designed in the normal way, but is run back and forth along a section of either the error or the ILC input to the plant. Unless the signal is divided into sections of less than $\frac{N-\lambda}{2}$ samples in length (where N is the samples per trial) and each one filtered separately, there will be insufficient time between the signal being recorded and the need for its use in the input to the plant. This length can be increased by using sections that overlap, and filtering them in parallel. This, however, only increases the allowable batch size to $N - \lambda$ samples. Because the ends of each batch are subject to error in the filtering process, which causes them to recombine imperfectly, a longer batch length is desirous. However, longer sized batches have the effect that there is at least a cycle's duration between the error being recorded and its use. It will be seen that this causes problems as well as very slow convergence. In order to increase convergence speed, it is tempting to allow learning to recommence whilst a section of the error is being filtered. This means that the error will then form the update that follows on from a different input to the one which caused it. Experiments have shown that this always leads to large oscillation of the cycle error and no further convergence. Therefore two methods have been used which keep the same input to the plant during the filtering process. The first filters a single cycle-length of error as it arrives, then repeats the input while it is filtered in reverse. The error is extended in both directions to avoid transients.

Convergence is twice as slow due to the cessation of learning. Learning must be halted during the reverse filtering stage as it would then form half of the next update and thus, in part, create the cycle error oscillations described. The second method takes lengths equal to three cycle-lengths and forward and reverse filters them in the same manner. The input is held for six cycles and only the middle cycle-length of error is used in the learning process. This helps reduce the filtering transients. The filter that has been used with these methods has a cut-off of 1.5Hz, following recommendations made in the last section. The small cut-off frequency/Nyquist frequency ratio has limited the filter order available for the class of filter chosen, and the cut-off is not ideal. The attenuation, however, will be double due to the dual filtering. Figure 6.21 shows results obtained

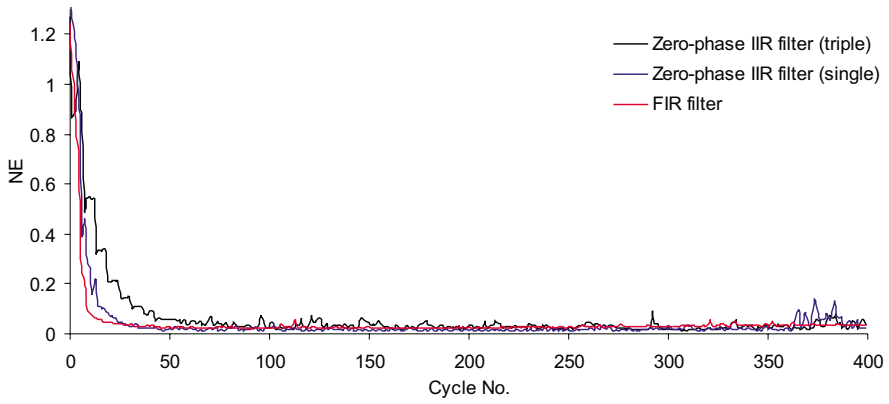


FIGURE 6.21: Error results for a 20 UPM R1 demand using non-causal filters, each with λ_{opt} and $\phi = 0.5$

using the two methods of non-causal filtering with a zero-phase IIR filter that have been described. The best result obtained with an FIR filter has been included for comparison. Figure 6.22 shows results obtained with the same filters, but using a different demand. The results show that batch mode filtering processes are ill-suited to ILC implemented

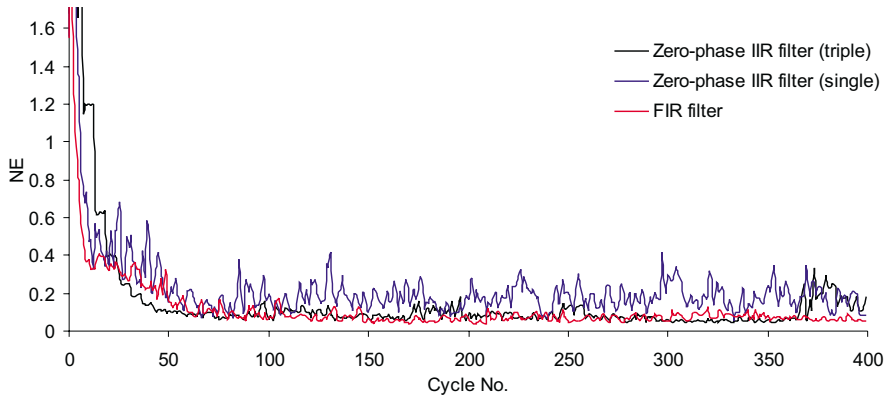


FIGURE 6.22: Phase-lead error results for 20 UPM R2 demand using non-causal filters, each with λ_{opt} and $\phi = 0.5$

in repetitive form (with no resetting of initial conditions). Despite large attenuation,

their performance leads to transitory cycle error, especially with high unit-rates and challenging demands. This is made more obvious when it is remembered that the triple segment IIR filter used only updates every 6 cycles, and the single segment, every two. Discrepancies that arise at the extremities of the system output for a fixed demand are the cause of these irregularities. The updates fit together imperfectly and cause oscillations. The value of the minimum error, however, is in some cases the lowest seen due to the choice of the cut-off frequency. It was found that the IIR filter's performance was improved in terms of less transient cycle error when the demand was not filtered. This differs from all the other tests performed in this respect, and is a consequence of both its increased attenuation, and ability to make corrections only every 6 cycles.

6.3 Stage 2 System

The phase-lead algorithm has been implemented on the stage 2 plant in order to compare its performance with the stage 1 case. The following figures show error results for 2 choices of the gain ϕ using a 10 UPM Sine demand. Figure 6.23 corresponds to

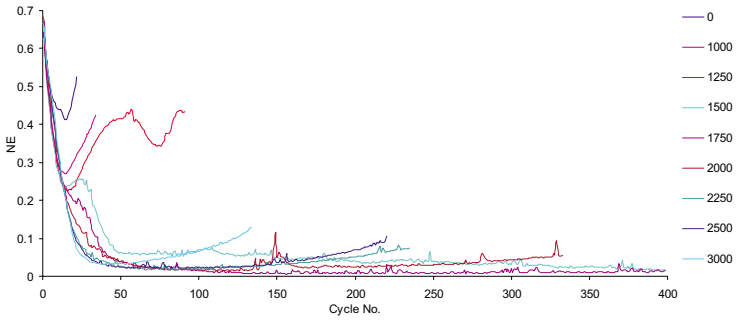


FIGURE 6.23: Phase-lead error results for 10 UPM sinewave demand with $\phi = 0.1$

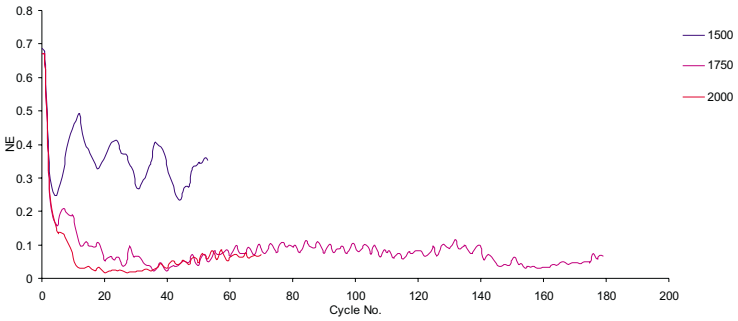


FIGURE 6.24: Phase-lead error results for 10 UPM sinewave demand with $\phi = 0.5$

$\phi = 0.1$ and Figure 6.24 to $\phi = 0.5$. Error results for the other demands are given in Appendix I.4. The stage 2 system's impulse response is shown in Figure 6.25, its maximum occuring at 0.98 seconds which corresponds to a phase-lead of 2450 samples.

The earlier discussion makes it unsurprising that the experimental optimum is somewhat lower, between 1750 and 2000. Comparing these results with those of the stage 1 system

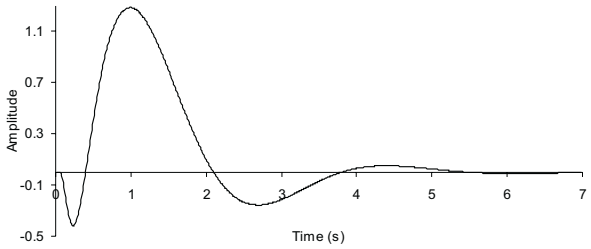


FIGURE 6.25: Impulse response of stage 2 plant

yields the following conclusions;

- T_{ins} is reduced and the stage 2 system's tests are shorter in duration.
- The minimum error is slightly larger in each case.
- A smaller range of phase-leads is able to produce good results.

These observations simply point to stage 2 being a higher order system with increased non-linearities, and phase-lead ILC being less capable of dealing with it. This also means that the plant model is likely to be less reliable which will effect those algorithms that directly use it.

6.3.1 Causal filters

Since causal filters helped produce the best results for the stage 1 system, they were also used on the stage 2 system. Six Chebychev filters, the parameters of which are given in Table 11.3, have therefore been designed for use with the phase-lead update according to the methods set out in Section 6.2.1. Figure 6.26 illustrates the general

Filter Name	Order	Cut-off /Hz	Impulse Max /samples	Attenuation (dB)		
				Primary	Secondary	Tertiary
TYPE-4-1.2	4^{th}	1.2	3600	50	68	77
TYPE-4-1.8	4^{th}	1.8	2975	68	66	86
TYPE-4-2.3	4^{th}	2.3	2700	56	65	109
TYPE-4-2.6	4^{th}	2.6	2650	63	64	90
TYPE-5-1.85	5^{th}	1.85	3750	87	86	93
TYPE-5-2.3	5^{th}	2.3	3150	77	89	101

TABLE 6.1: Stage 2 Phase-lead filter characteristics

effect of altering the filter cut-off point. As the cut-off frequency is reduced, the effect due to unmodelled dynamics is decreased. This produces less transients in the cycle

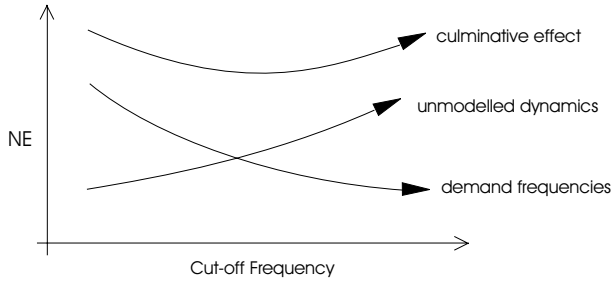


FIGURE 6.26: Factors effecting the NE

error and phase-lead ILC is more effective and the error reduced. Simultaneously, as the cut-off frequency is lowered, frequencies present in the demand that are required for learning are attenuated at the system input. This will reduce tracking of the demand at these frequencies and the error will consequently high. It is therefore important to choose a cut-off that is a compromise between these effects. Since causal filters are used, reducing the cut-off adds low frequency lag and therefore instability to the learning. All

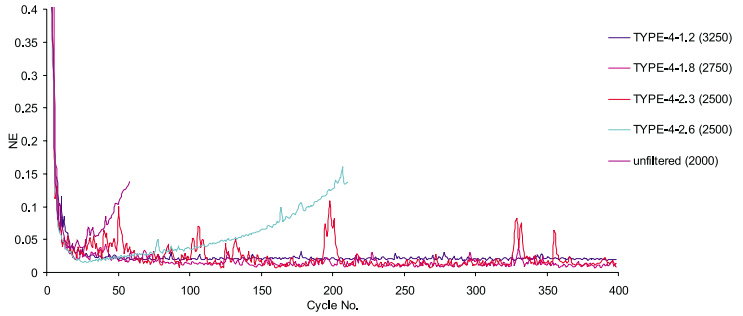


FIGURE 6.27: Phase-lead error results for 20 UPM sinewave demand using causal filters with λ_{opt} and $\phi = 0.5$

the filters described in Table 11.3 were used with the stage 2 system, and Figure 6.27 shows results using a selection for the 20 UPM sinewave demand. It can be concluded that the TYPE-4-1.8 filter maintains the best compromise between the effects discussed for this demand. Additional results are shown in Appendix I.4. The values of λ_{opt} are given in brackets. Figure 6.28, however, shows that stronger filtering leads to the best

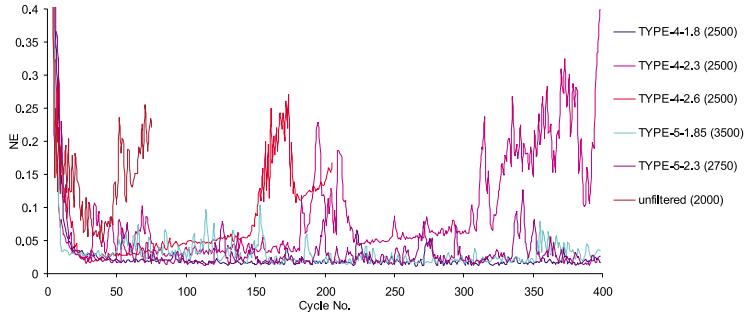


FIGURE 6.28: Phase-lead error results for 20 UPM R1 demand using causal filters with λ_{opt} and $\phi = 0.5$

results with the R1 demand. Although it has higher frequency demand components, the increased instability of the learning benefits more from stronger filtering than the detriment this causes to the tracking.

6.3.2 Forgetting Factor

The use of a 'forgetting factor', γ, in the phase-lead algorithm has been proposed by several authors (Lewin, 1999; Xu and Yan, 2003) as a means of removing the instability of the algorithm. The update of the phase-lead law is changed to;

$$u_{k+1}(i) = \gamma u_k(i) + \phi e_k(i + \lambda) \tag{6.5}$$

Figure 6.28 shows experimental results using the forgetting factor, additional results are shown in Appendix I.5. Stability is gained for the price of a greater residual error. Figure

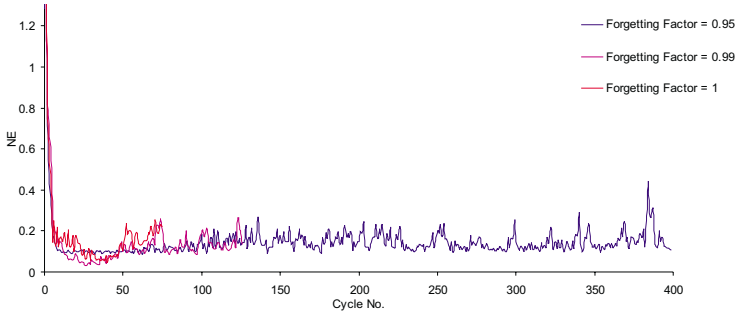


FIGURE 6.29: Phase-lead error results for 20 UPM R1 demand using forgetting factors with $\phi = 0.5$

6.29 shows forgetting factor results using the sinewave demand, this time applying the most sucessful filter from the previous section. The filter is able to provide the same long term stability but without the cost of increased residual error. The great advantage of the forgetting factor is its simplicity. The reason for increased stability but increased

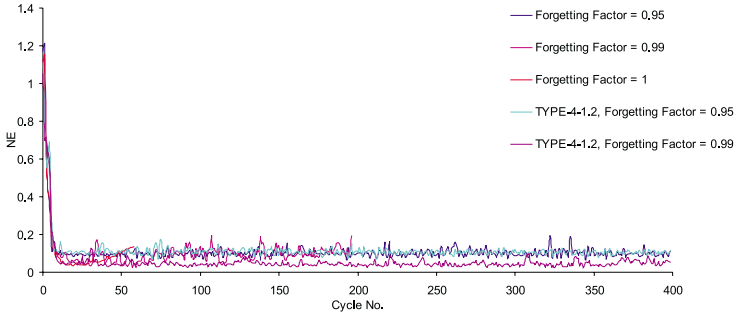


FIGURE 6.30: Phase-lead error for 20 UPM sinewave demand using forgetting factors and filtering with $\phi = 0.5$

minimum error is discussed in Chapter 7, where methods for improving phase-lead ILC's peformance are detailed.

6.4 Summary

Phase-lead ILC has been found to out-perform P-type, D-type and Delay-type ILC when applied to both the stage 1 and stage 2 systems. The technique has been examined and reasons for its success, and indeed failure, have been put forward. A method of arriving at the phase-lead that produces the best performance has been proposed, using knowledge of the plant model. The existence and effect of unstable frequencies caused by phase-lead ILC has been discussed and a method of predicting their value and harmfulness given. A number of both causal and non-causal filters have been tested and design procedures described in order to maximize performance. The role of attenuation of various frequencies has been discussed and results presented to illustrate the conclusions drawn. Results have indicated that demands can only be learnt up to a certain frequency. It has also been seen that, due to unstable frequencies, stability of phase-lead ILC cannot be assured as the cycle number progresses. A forgetting factor has been used with the stage 2 plant to reduce the instability, but has the disadvantage of increased minimum NE compared to filter-based approaches.

Chapter 7

Multiple Phase-lead Algorithms

7.1 Introduction

There are a number of methods that are related to the phase-lead ILC algorithm and are likely to offer improved performance. Those that also involve updates at specific phase-leads include the use of limit cycles to change the lead between trials of a single phase-lead update (Wirkander and Longman, 1999) and the decomposition of the error signal into its frequency components inbetween trials in order to use phase-leads to cancel out instabilities that would arise (Elci et al., 1994). However, it is possible to improve the properties of the phase-lead law more directly. Algorithms were implemented on the stage 2 plant. Conversion of the phase-lead law, given by Equation 6.2, to the discrete domain yields

$$U_{k+1}(z) = U_k(z) + \Phi(z)E_k(z) \tag{7.1}$$

where the phase-lead, λ, has been absorbed into a filter $\Phi(z)$, to give $\Phi(z) = \phi z^{\lambda}$. Substituting $E_k(z) = Y_d(z) - Y_k(z)$ and introducing $Y_k(z) = G(z)U_k(z)$, where $G(z)$ is the discrete plant model gives;

$$U_{k+1}(z) = U_k(z)(1 - \Phi(z)G(z)) + \Phi(z)Y_d(z) \tag{7.2}$$

The corresponding error evolution is then;

$$E_{k+1}(z) = E_k(z)(1 - \Phi(z)G(z)) \tag{7.3}$$

It would appear that the magnitude of the error at a frequecy w, is multiplied by $|1 - \Phi(e^{jwT})G(e^{jwT})|$ in sucessive trials. However, it can be shown (Longman, 2000), that this is only true for steady-state learning, when the effect of the initial conditions is zero. This only accurately models the situation if the system impulse response is much shorter than the trial length, a situation that is not applicable in the case of the stage 2 plant. The thinking behind this is related to how the 'wave' of learning that occurs

along a trial during learning operates. It can be further shown that the condition given by

$$|1 - \Phi(e^{jwT})G(e^{jwT})| < 1 \tag{7.4}$$

does guarantee monotonic convergence and is essentially the same as the true stability boundary in RC. Conversely, the true stability boundary in ILC is given by $0 < CB < 2$ and does not encompass the plant dynamics. It can again be shown that Equation 7.4 does still satisfy this and also that it should be considered a necessary condition for obtaining practical results (Longman, 2000). The true monotonic decay condition in ILC is given by

$$\|\underline{e}_j\| \leq \max_i(\sigma_i)\|\underline{e}_{j-1}\| \tag{7.5}$$

and can be approximated by Equation 7.4. Learning transients are caused by the initial conditions at the beginning of each trial, the effect of which lasts for approximately 4 times the largest time constant of the system. Satisfying the monotonic convergence condition implies the presence of good learning transients.

Equation 7.4 restrains the frequency plot of $\Phi(jw)G(jw)$ to lie in an open unit disc centered at 1. Inspecting Equation 6.4 of Chapter 6, and using the substitutions $f = w$, $\angle f = -\angle G(jw)$ and $\angle\Phi(jw) = T\lambda w$ for the unstable frequency, w, produces

$$\angle G(jw) + \angle\Phi(jw) = 180(1 + 2i) \qquad i = 0, 1, 2... \tag{7.6}$$

Figure 7.1 shows a Nyquist plot of $\Phi(jw)G(jw)$ for the stage 2 plant and the unfiltered phase-lead algorithm with $\lambda = 2500$. Equation 7.6 locates the points of intersection

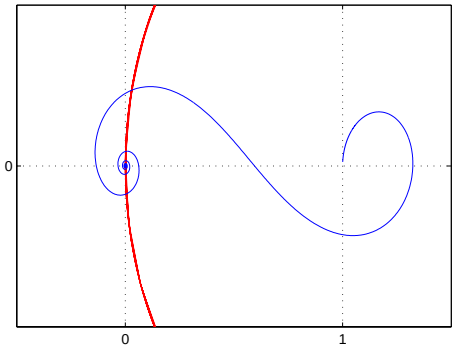


FIGURE 7.1: Nyquist plot of $\Phi(jw)G(jw)$ with unit circle centred on $+1$

with the imaginary axis to the left of the origin which are unstable frequencies and local magnitude maxima. It was their $-180°$ multiples of phase-lag that allowed them to be located easily by inspection of experimental data. A condition for stability (Wang and Longman, 1996) which is a direct consequence of Equation 7.4 is given by

$$\cos(\angle\Phi(jw) + \angle G(jw)) > \frac{1}{2}|\Phi(jw)G(jw)| \tag{7.7}$$

and, for phase-lead ILC is equal to

$$\cos(\lambda T w + \angle G(jw)) > \frac{\phi}{2}|G(jw)| \tag{7.8}$$

Since $|G(jw)|$ is usually greatest at lower frequencies, Equation 7.4 locates the frequencies where the left hand side is maximum (= 1) and therefore most likely to be unstable. This suggests that the design techniques of Chapter 6 have widespread application. In this chapter the stage 2 plant will be used and the use of additional phase-leads will be investigated in order to stabilise the phase-lead algorithm.

7.2 Verification of Phase-lead Conclusions

The system sample rate has been changed to 1000Hz in order to allow increased computation time between samples, making $T = 0.001$. According to the results of Chapter 6 the best phase-lead is now at $\frac{1000}{2500} \times 1750 = 700$ samples and this has been consistant with experiments. It has also been experimentally verified that no combination of strictly positive λ and ϕ satisfy the convergence criterion for all frequencies. The optimum lead was therefore a compromise between the amount of learning over the range of frequencies of the demand, and the magnitude of those unstable frequencies, w, which form a subset of those satisfying $|1-\Phi(e^{jwT}G(e^{jwT}| \geqslant 1$. The best solution is that which provides the greatest amount of learning per unit instability. The frequencies necessary for learning are of course dependent on the demand, whilst the instability frequencies are less so and will certainly be present in the system's transient response. A theoretical optimal value to the problem of selecting a single phase-lead, λ, is therefore given by Equation 7.9. The first term picks out the frequencies which should be reduced, that is $|1 - \Phi(e^{jwT})G(e^{jwT})| > 1$, and the second incorporates those frequencies that will be learnt.

$$\min_{\lambda,\phi} \int_{w=0}^{\infty} K \max\{0, |1-\phi e^{jwT\lambda}G(e^{jwT})|-1\} - |R(w)| max\{0, 1-|1-\phi e^{jwT\lambda}G(e^{jwT})|\} dw \tag{7.9}$$

$R(w)$ denotes the magnitude of the reference signal, emphasising those frequencies required to be learnt. K is a positive scalar which represents the relative importance between learning and instability. If $K = 0$ only learning of the demand is deemed important, as $K \to \infty$ only the lack of unstable frequencies (and the solution has $\Phi(jw) = 0$). A mathematical software package has been used to generate a vector for each case of $|1 - \phi e^{jwT\lambda}G(e^{jwT})|$ corresponding to as great a range of frequencies as necessary. A function is then applied to calculate the integral term in Equation 7.9 and this is repeated for a large range of amplitudes and phase-leads. A minimum is found by simply selecting the lowest of all those integrations performed. Figure 7.2 shows the graph of $|1 - \Phi e^{jwT}G(e^{jwT})|$ with solutions corresponding to $K = 0$, and $K = 34$ and that found to work best in practice. This latter takes into account the interaction

between learning and instability. Since $K = 34$ matches the best in practice, that is the compromise found between instability and learning. The result that provides maximum

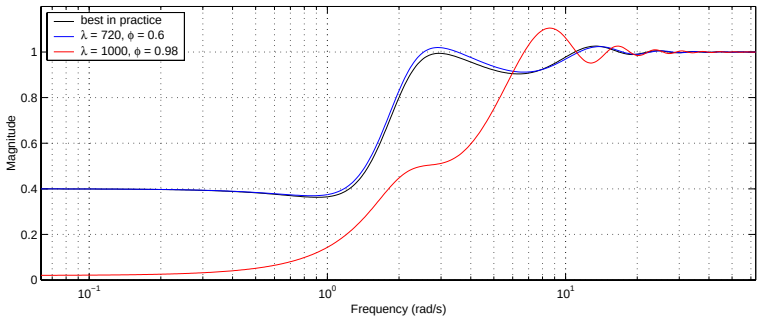


FIGURE 7.2: Phase-leads of $\lambda = 1000$, $\phi = 0.98$ ($K = 0$), $\lambda = 720$, $\phi = 0.6$ ($K = 34$) and the best in practice

learning ($K = 0$) does not prove most successful in practice. It does, however, closely correspond to the phase-lead of the maximum impulse response (980 samples at this sampling frequency), as noted in (Wang and Longman, 1996). Simulation with a variety of different plants confirms that the maximum impulse response usually approximates but does not correspond exactly to the maximum amount of learning.

7.3 Forgetting Factor Revisited

Because no single phase-lead can provide stability, the theoretical effect of the forgetting factor, seen in Section 6.3.2, has been investigated. This was the simplest method that suceeded in stabilsing the learning. The expression for the error propagation given in Equation 7.1 is therefore modified to

$$E_{k+1}(z) = E_k(z)(\gamma - \Phi(z)G(z)) + Y_d(1-\gamma) \tag{7.10}$$

The criteria for learning a zero demand when using a forgetting factor is adjusted to

$$|\gamma - \Phi(e^{jwT})G(e^{jwT})| < 1 \tag{7.11}$$

Expanding Equation 7.10 gives the n^{th} error as

$$\begin{aligned} E_n(z) \quad = \quad & E_0(z)(\gamma - \Phi(z)G(z))^n + (\gamma - \Phi(z)G(z))^{n-1}Y_d(1-\gamma) \\ & +(\gamma - \Phi(z)G(z))^{n-2}Y_d(1-\gamma) + \ldots + Y_d(1-\gamma) \end{aligned} \tag{7.12}$$

Therefore, even as $n \to \infty$, there will be a residue error of $Y_d(1-\gamma)$. It follows that values of $0.9 < \gamma < 1$ make it relatively simple to avoid the instability frequencies whilst still achieving a large degree of learning.

Figure 7.3 shows how the single phase-lead that provided most learning was made stable by use of a forgetting factor. Given a forgetting factor and a cost function, such as

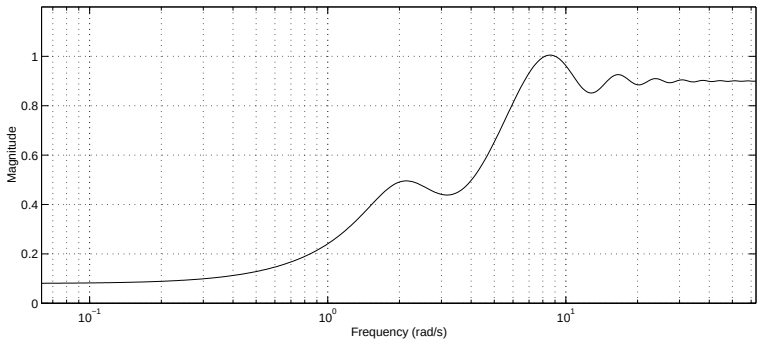


FIGURE 7.3: Forgetting factor = 0.9. Phase-lead of 1000, amplitude 0.98

Equation 7.9, to minimize, it is likely that a lower value can be obtained using a forgetting factor. Figure 7.4 shows the results obtained using $\gamma = 0.95$ and Equation 7.9 with $K = 0$ and $K \to \infty$. $R(w)$ is made equal to 1 to see what general results can be expected. If the forgetting factor is not set low enough to stabilize the system on its own, a high price is incurred on the learning by altering $\Phi(z)$ to satisfy it instead. It can be seen

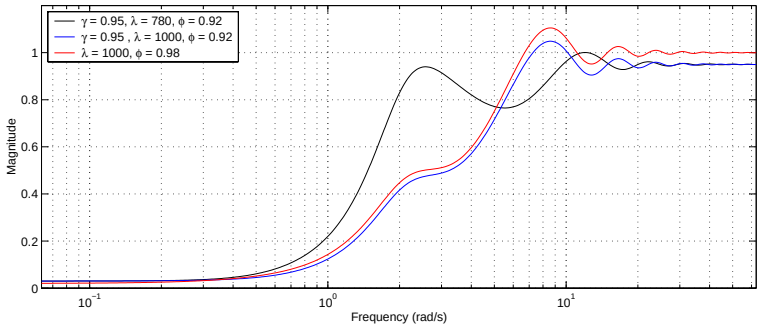


FIGURE 7.4: The single phase-lead with most learning ($\lambda = 1000$, $\phi = 0.98$). Forgetting factor 0.95 and 1)$K = 0$ ($\lambda = 1000$, $\phi = 0.92$), 2)$K \to \infty$ ($\lambda = 780$, $\phi = 0.92$)

that learning has been heavily compromised for stability. However, although the error converges, it does not converge to as low a value as previously due to the residue error.

7.4 Additional Phase-leads

Since a phase-lead naturally provides points of learning and points of instability, it is a natural progression to add another phase-lead in order to stabilize the learning. Figure 7.5 shows two instances of the single phase-lead that provided the most learning combined with another lead, λ_1, and its associated gain, ϕ_1, in the same algorithm, that is $\Phi(e^{jwT}) = 0.98e^{jw} + \phi_1 e^{jw\lambda_1 T}$. The first addition seeks to reduce the amount of instability using Equation 7.9 with $K \to \infty$, the second to decrease the criterion with $K = 34$. It has been found that no additional lead can further increase the total learning. The dual of Equation 7.6 predicts the frequencies which are local magnitude

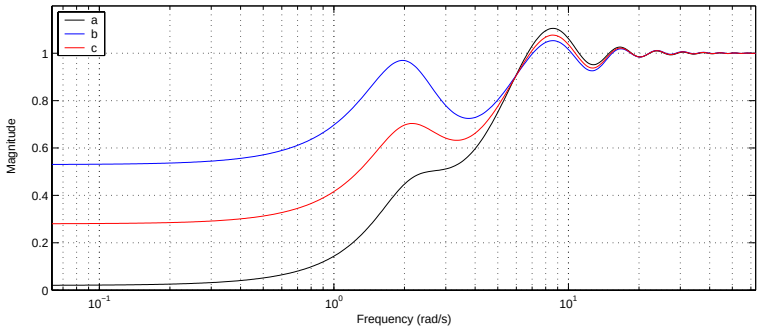
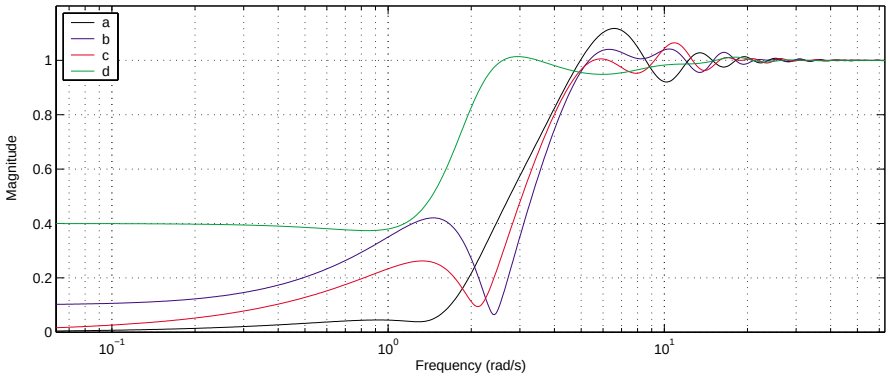


FIGURE 7.5: Phase-lead of 1000, amplitude 0.98, and with additions b)$K \to \infty$ ($\lambda = 0$, $\phi = 0.55$), c)$K = 34$ ($\lambda = 0$, $\phi = 0.3$)

minimisers of the the polar plot of $\Phi(jw)G(jw)$ and is given by

$$\angle G(jw) + \Phi(jw) = 360i \qquad i = 0, 1, 2... \tag{7.13}$$

and can be used as a simpler method to approximate the phase-leads that are likely to help stabilize the single phase-lead case. It is clear, however, that only marginal improvement has been achieved. Performing the optimization again, but with both phase-leads and corresponding amplitudes allowed to vary, yields improved results. The solution in each case was obtained by setting $\Phi(e^{jwT}) = \phi_1 e^{jw\lambda_1 T} + \phi_2 e^{jw\lambda_2 T}$ in Equation 7.9, and finding the cost for all combinations of amplitudes ϕ_1, ϕ_2 and leads λ_1, λ_2 within a certain bound and resolution. The minimum again was found by comparison of the costs. Figure 7.6 shows further improvements, in terms of solving Equation 7.9 with a)



	λ	ϕ
a	1400	0.5
	900	0.5
b	900	0.2
	600	0.4
c	1200	0.8
	0	0.2
d	1400	0.5
	1000	0.6

FIGURE 7.6: Plot of $|1-(\phi_1 e^{jwT\lambda_1} + \phi_2 e^{jwT\lambda_2})P(e^{jwT})|$ for variables given in the table

$K = 0$ and b) $K = 34$. It is clear that more learning has been achieved than previously. If $R(w)$ is set to 1 in order not to constrain the frequencies of learning, still better results are seen, Figure 7.6 shows this case with c) $K = 200$ and d) $K = 1000$. Figures 7.7 and 7.8 show experimental results using these optimizers. It is clear that great improvements over the single phase-lead case have been achieved in terms of stability with similar results in terms of minimum error and convergence. The success of an additional lead, and the intuitive idea of extra leads canceling the instability of present

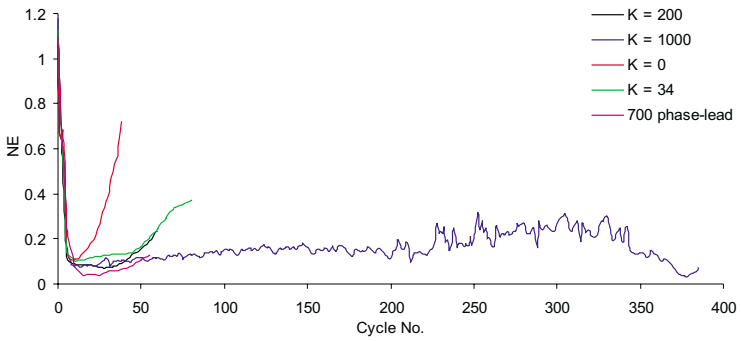


FIGURE 7.7: Phase-lead pairs for 20 UPM sinewave demand with single phase-lead comparison and phase-lead pairs as in Figure 7.6

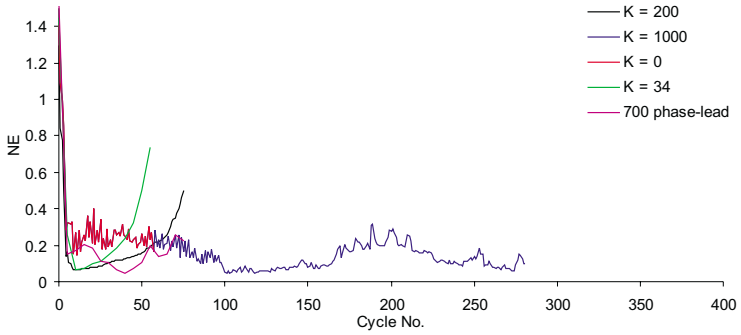


FIGURE 7.8: Phase-lead pairs for 20 UPM R1 demand with single phase-lead comparison and phase- lead pairs as in Figure 7.6

ones with no loss of learning, motivates its extension to the use of multiple leads. When the number of phase-leads and their respective amplitudes is increased, the approach of testing all possibilities, even with a modest resolution, becomes unfeasible and so alternative methods must be sought. The most successful iterative methods that have been tried attempt to approach a local optimum by continually adding small leads in positions where they reduce the cost most. Considerable care is required to try and avoid too premature a solution to this non-convex optimization problem. A technique that has been developed examines only the addition of small amplitudes, δ, at each operation and finds at each step

$$\min_{i,\epsilon} \int_{w=0}^{\infty} K \max\{0, |1 - [\epsilon e^{jwT\lambda_i} + \Phi(e^{jwT})]G(e^{jwT})| - 1\} \\ -max\{0, 1 - |1 - [\epsilon e^{jwT\lambda_i} + \Phi(e^{jwT})]G(e^{jwT})|\} dw \tag{7.14}$$

where $\epsilon \in \{+\delta, -\delta\}$. The compensator is then updated by taking

$$\Phi(e^{jwT}) = \Phi(e^{jwT}) + \epsilon e^{jwT\lambda_i} \tag{7.15}$$

If the cost has not decreased between cycles then a local minimum has been located. If K is initially set too high then a local minimum will already have been found since

any addition adds instability. It is therefore a problem of increasing the learning before it is allowed to become too large and future iterations effectively concentrate only on reducing instability. This is increasingly pertinent since it has been observed that any $\Phi(e^{jwT})$ can be 'stabilised', to an arbitrary value $\rho > 0$ exchanged for the right-hand side of Equation 7.4, by the addition of phase-leads, but this will usually produce a very unsatisfactory solution in terms of the total learning. Conversely, concentrating on learning and not instability brings about solutions which fall short of meeting the converence condition. It has been found that increasing K linearly as the iterations progress leads to a solution that incorporates a great deal of learning whilst satisfying the necessary stability requirements. Figure 7.9 shows both the phase-leads and the corresponding criteria $|1 - \Phi(e^{jwT})G(e^{jwT})|$, 41 phase-leads are shown and indicated by asterisks, and although values of $\lambda T > 2$ were feasible, no values $\phi > 0$ appeared in the solution. The algorithm uses $K = 100 + I$, where I is the cycle number. 1200 cycles

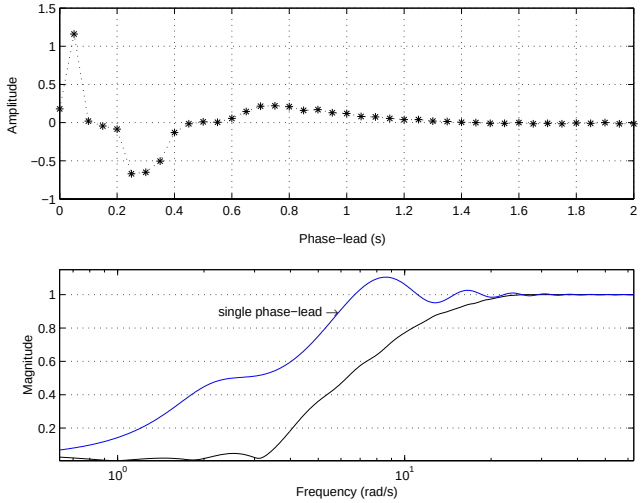


FIGURE 7.9: Multiple phase-lead update and comparison of its and the single phase-lead update's stability criterion

are required with $\delta = 0.005$, and a phase-lead resolution of 0.05. The optimization is intrinsically short-sighted in that it would rather add leads than remove them and approach the nearest local minimum possible. This is inevitable unless some other criterion is incorporated into the cost function which adds more structure. Without more knowledge of the global solution this is impractical. The solution achieved will not have sudden changes in amplitude characteristic of satisfying the criterion at high frequencies, but this should impart robustness for the same reason.

Figure 7.10 shows the minimiser calculated when the optimisation is repeated with an increased phase-lead resolution of 0.1. Instead of 41 phase-leads there are 19, and the calculation time has been reduced from 27 hours to $2\frac{1}{4}$ using a 1GHz pentium PC. 1100 cycles are rquired using the same update procedure. As before, the corresponding criteria $|1 - \Phi(e^{jwT})G(e^{jwT})|$ is shown below the phase-lead locations.

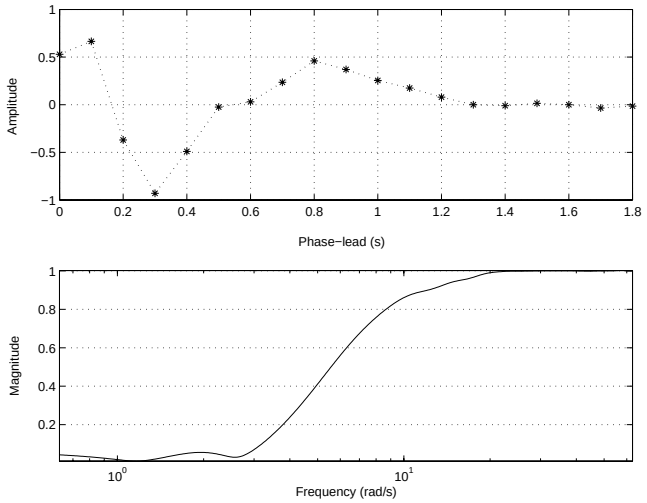


FIGURE 7.10: Reduced multiple phase-lead update and corresponding stability criterion

If phase-lags as well as phase-leads are allowed to be considered, the optimum is found to offer considerably less performance than when just using phase-leads, this is explained by the observations made regarding Equation 7.9. Figure 7.9 shows the best result found for both leads and lags. This optimization uses $\delta = 0.005$, $K = 100 + I$ and a phase-lead resolution of 0.05. Since lags have a greater impact on the cost criterion, several were added during early iterations to give large amounts of learning and relatively lightly penalized instability. In later iterations learning is sacrificed in order to cancel out the instability of these leads.

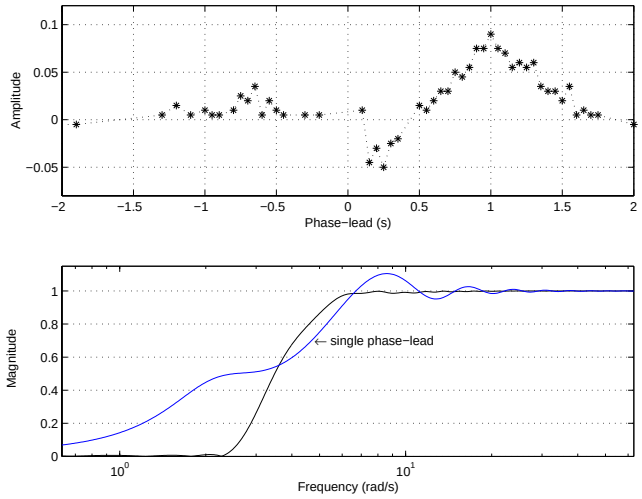


FIGURE 7.11: Multiple phase-lead and lag update and comparison of its and the single phase-lead update's stability criterion

Figure 7.12 shows the minimiser calculated when the optimisation is repeated with an increased phase-lead resolution of 0.1. 270 cycles are required in this case.

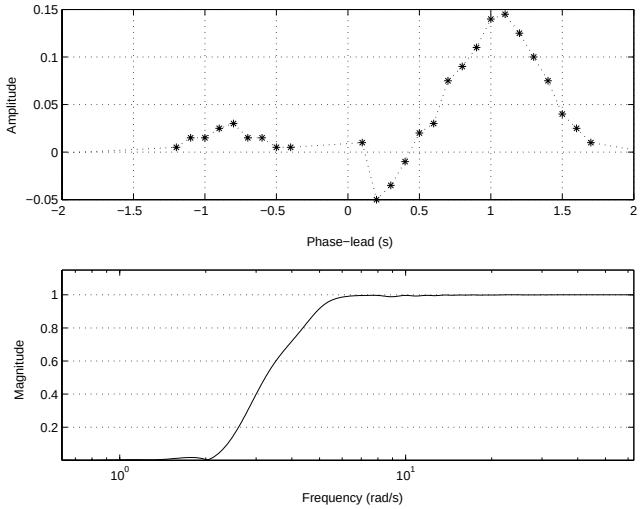


FIGURE 7.12: Reduced multiple phase-lead and lag update and corresponding stability criterion

Figure 7.13 shows the results of an optimisation involving just phase-lags where $\delta = 0.002$, and a phase-lead (= phase-lag) resolution of 0.1 have been used. The update procedure has been changed to $K = 1$ since increasing K as the cycle number increases terminates the optimisation prematurely and a large initial value causes the initial value of $\Phi(jw)$ to be the local minimiser. 2150 cycles are required but the results are poor.

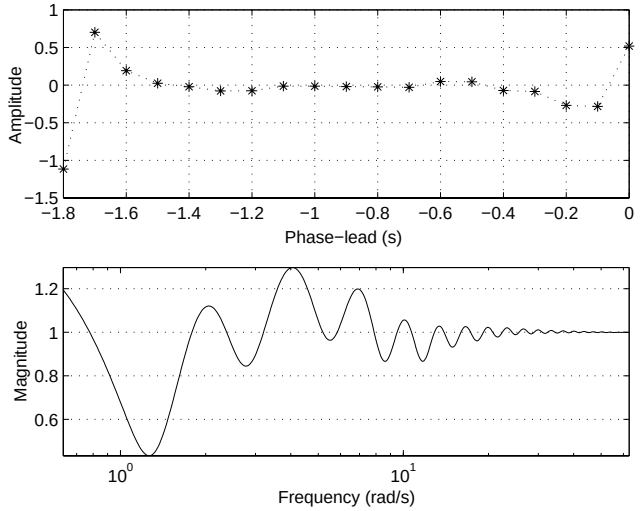


FIGURE 7.13: Reduced Phase-lag

The updates shown, each with p phase-lags and q phase-leads with associated amplitudes can be specified by

$$\Phi(e^{jwT}) = \phi_{-p}e^{-jwTp\Delta} + \phi_{p-1}e^{jwT(1-p)\Delta} + \cdots + \phi_{-1}e^{-jwT\Delta} + \phi_0 + \phi_1 e^{jwT\Delta} + \cdots + \phi_{q-1}e^{jwT(q-1)\Delta} + \phi_q e^{jwTq\Delta} \tag{7.16}$$

where the phase resolution is Δ. They can therefore be represented by the FIR transfer function

$$\Phi(z) = \phi_p z^{-p} + \phi_{p-1} z^{-(p-1)} \cdots + \phi_{-1} z^{-1} + \phi_0 + \phi_1 z + \phi_2 z^2 + \cdots + \phi_{q-1} z^{q-1} + \phi_q z^q \quad (7.17)$$

in which the sampling time is Δ seconds.

Promising general solutions have been found and it can be shown that it is their generality that makes them robust. The implementation of the phase-leads and their respective amplitudes shown in Figures 7.9 - 7.13 leads to large transients during the initial iterations. Large transients which do not lead to divergence of the cycle error are either due to very slow convergence of relatively low frequencies, or some practical limitation that means that the filter $\Phi(z)$ is implemented imperfectly. An instance of the latter in the present case arises because the phase-lead resolution of 0.05 corresponds to an update of the input only at every 50 samples at the sampling rate used. Spline interpolation has been used to provide a smooth update at every sample, but the algorithm is also unstable at high frequencies resulting in a gradual divergence.

Figures 7.14, 7.15 and 7.16 show experimental results obtained using the multiple phase-lead optimizer shown in Figure 7.9. The performance of the best performing single phase-lead case is also shown, its value given in brackets. The results show a great

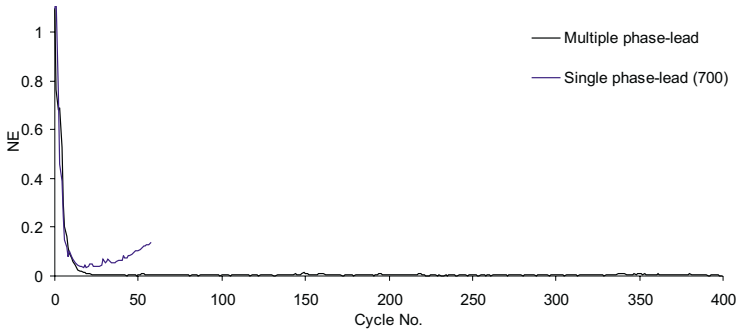


FIGURE 7.14: Single and multiple phase-lead error results for 20 UPM sinewave demand

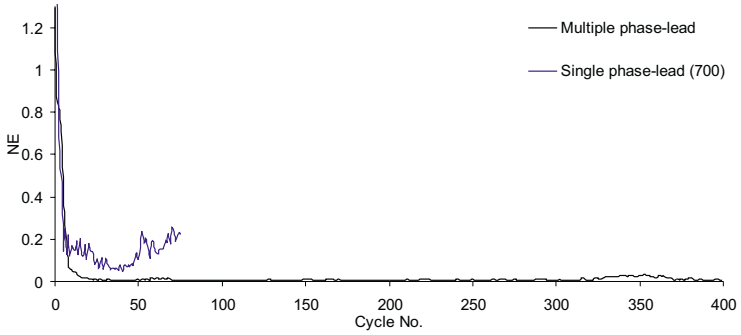


FIGURE 7.15: Single and multiple phase-lead error results for 20 UPM R1 demand

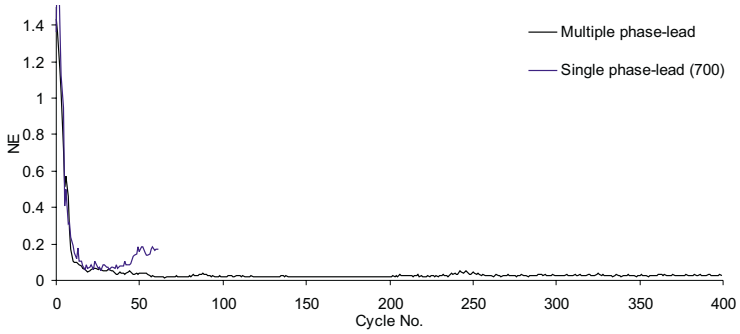


FIGURE 7.16: Single and multiple phase-lead error results for 20 UPM R2 demand

improvement in convergence and stability. Unless explicitly stated, every experiment has been performed with the updates scaled to have a total amplitude summation, $\sum_i \phi_i$, and hence steady-state gain, of 0.5. Figure 7.17 shows the effect of varying this value in the case of the optimal phase-lead update. Learning transients become very large when $\phi > 0.7$ and the system is unstable at low frequencies. Scaling the updates

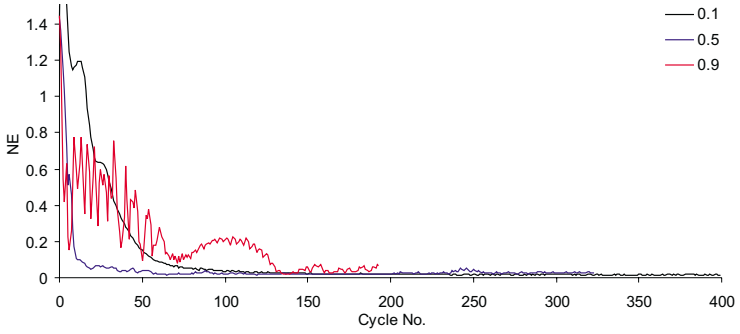


FIGURE 7.17: Multiple phase-lead error results for 20 UPM R2 demand with various ϕ

reduces the instability and reduces the learning transients. Any unstable frequencies diverge more slowly, but the rate of convergence is sacrificed. In order to compare different compensators, it is necessary to set the same scaling value. Since this also sets the steady-state value which is invariantly close to the frequency of the demand, it also ensures similar initial convergence rates. Each compensator's performance can be varied in relation to the other by changing ϕ, and increased convergence can be traded for reduced divergence and transients. It is therefore a sensible approach to compare controllers with the same scalar multiplier, whilst remembering the effect of varying each of these gains. The success of the multiple phase-lead update has been shown, but also its susceptibility to learning transients and high frequency instability. Because the update of Figure 7.9 has been the most successful, it will be referred to as the 'multiple phase-lead' update in the comparisons contained in the following chapters. The update was designed to satisfy the monotonic convergence criteria for the nominal plant model

over the system bandwidth, its failure to do so for all gains $0 \leq \phi \leq 1$ therefore indicates a lack of robustness in the algorithm.

7.5 Summary

In order to improve on the convergence and instability problems of the phase-lead law, the convergence conditions for RC and ILC have been investigated. This has allowed some of the results of the previous section to be explained and justified. The monotonic convergence condition has been used directly to formulate an optimisation routine that aims to satisfy by repeated changing amplitudes or adding terms to the controller. This leads to multiple phase-lead and phase-lag solutions which can be implemented as FIR filters. The results show little sign of instability and very fast convergence.

Chapter 8

Contraction Mapping Algorithms

8.1 Introduction

This algorithm has been briefly described in Chapter 6, and the plant adjoint, $G^*(z)$, has been used in ILC for some time (Fututa and Yamakita, 1987; Jang and Longman, 1994). The basic discrete formulation is

$$u_{k+1}(z) = u_k(z) + \phi G^*(z) e_k(z) \tag{8.1}$$

where $G^*(z) = G(-z)$. The implementation can be achieved using the vector equation $y_k = Gu_k$ and an equivalent matrix representation (Hatonen et al., 2003b), where

$$G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \dots & CB \end{bmatrix} \tag{8.2}$$

giving

$$G^T = \begin{bmatrix} CB & CAB & CA^2B & \dots & CA^{N-1}B \\ 0 & CB & CAB & \dots & CA^{N-2}B \\ 0 & 0 & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & CB \end{bmatrix} \tag{8.3}$$

where N is the number of sampling instants in the reference trajectory. The update is then given by

$$u_{k+1} = u_k + \phi G^T e_k \tag{8.4}$$

It can be shown (Furuta and Yamakita, 1986) that the error propagation is given by

$$e_{k+1} = (I - \phi GG^T)e_k \tag{8.5}$$

and so

$$\|e_{k+1}\|^2 - \|e_k\|^2 = -2\phi\|G^T e_k\|^2 + \phi^2\|GG^T e_k\|^2 \tag{8.6}$$

which means that with a small enough choice of ϕ, the right hand side can be made negative and monotonic convergence is assured. An equivalent condition (Hatonen et al., 2004) is given by

$$\sup_{\omega\in[0,2\pi]} |1 - \phi|G(e^{j\omega})|^2| < 1 \tag{8.7}$$

In the case of repetitive control the Markov parameters can be looped from the trial end to the trial beginning so that Equation 8.3 with each Markov parameter appearing once in each column, becomes

$$\begin{bmatrix} CB & CAB & CA^2B & \dots & CA^{N-1}B \\ CA^{N-1}B & CB & CAB & \dots & CA^{N-2}B \\ CA^{N-2}B & CA^{N-1}B & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CAB & CA^2B & CA^3B & \dots & CB \end{bmatrix} \tag{8.8}$$

where the equivalent matrix representation used in (Hatonen et al., 2003c) is applied. The number of looped entries depends on the number of non-zero Markov parameters, and this corresponds to more than one looped entry if this value is greater or equal to N. Since the Markov parameters equal the magnitudes of the system impulse response, it is possible to implement Equation 8.1 simply by applying the impulse response as a phase-lead update function, setting

$$\Phi(z) = CBz + CABz^2 + CA^2Bz^3 \cdots + CA^{M-1}Bz^M + CA^MBz^{M+1} \tag{8.9}$$

If the number of non-zero Markov parameters exceeds the trial length then they must either be removed (truncated) or looped so that all the entries in some diagonals of G^T have two terms. The looped entries really require data that hasn't yet been produced but are given data from the previous trial which causes transients in the learning. A discussion of the dangers of truncating the impulse response by too much and methods to address this, is given in (Chen and Longman, 2002).

8.2 Use of Pole-placement to avoid Truncation

One possible way to address the instability caused by truncating the plant's impulse response is to reduce its length by using a dead-beat controller which effectively places

all the poles at the origin of the z-plane. This results in the impulse response going to zero in n steps, where n is the number of states of the original system. Truncation would then be avoided provided the trial length exceeds n. The G^T algorithm is then applied to this dead-beat closed-loop system. The controller has been implemented using pole-placement by Ackermann's method to produce the state gain K. This is shown schematically in Figure 8.1 in which an estimator is required to implement the state feedback. It has been found to result in a very unstable system. Alternatively, if only a

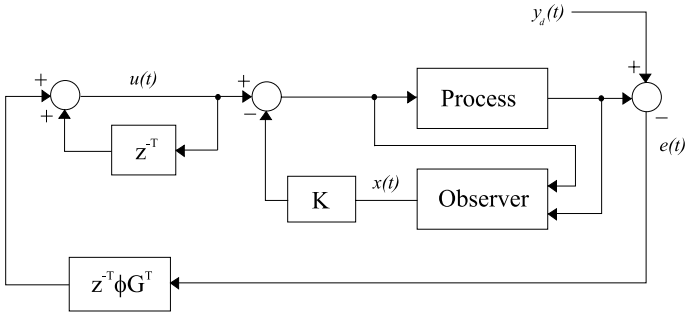


FIGURE 8.1: Flow diagram with closed-loop pole-placement plant

moderate adjustment is considered, the impulse response can be shortened significantly without noticeable loss of stability. Three systems have been simulated in order to illustrate the technique. The first is the stage 2, PID controlled system, G(z), which has dominant poles at $0.9894 \pm 0.0203j$ and $0.9061 \pm 0.0505i$. The second uses SVF to move these poles and create double poles at 0.7 and 0.94 to produce $G_1(z)$. The third again uses SVF to create double poles at 0.85 and 0.9, producing $G_2(z)$. The systems have been multipled by a scalar in order to have a steady-state step response (and hence impulse response integral) of unity. This makes their low frequency convergence rates of a similar magnitude and allows fairer comparisons to be made between them. Figure 8.2

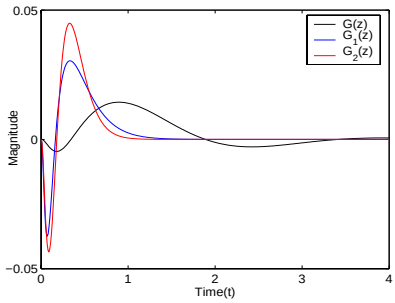


FIGURE 8.2: Impulse responses for pole-placed system simulations

shows the impulse responses of these systems and confirms that the impulse response has been shortened by use of pole-placement. Figure 8.3 shows plots of the stability criterion $|1 - \Phi(e^{jwT})G(e^{jwT})|$ for the resulting ILC schemes. Closer inspection of Figure 8.3 shows that the original system, G(z), whose impulse response has been concatenated to 6 seconds, reaches 1.0001 therefore failing to satisfy the criteria. However, the corresponding plots for the systems $G_1(z)$ and $G_2(z)$ do satisfy it over the range shown.

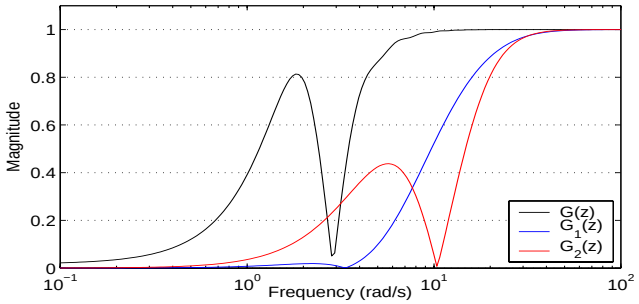


FIGURE 8.3: $|1 - \Phi(e^{jwT})G(e^{jwT})|$ for pole-placed system simulations

Figure 8.4 shows tracking and error simulation results for a 20 UPM R2 demand with

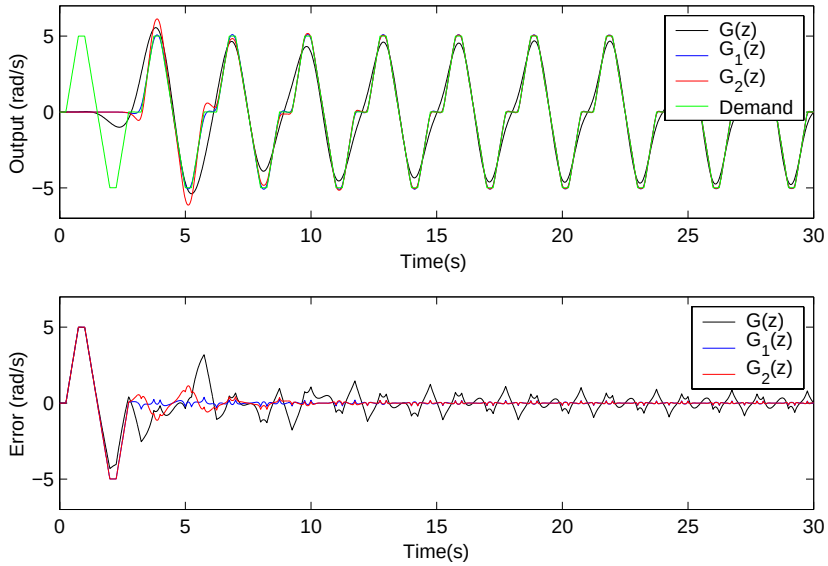


FIGURE 8.4: Output and error signals for pole-placed system simulations

$\phi = 1$. It is clear that, not only is the truncation instability removed by $G_1(z)$ and $G_2(z)$, but their convergence is far more rapid. There is possiblity of reduced robustness and increased transients using the method described, and for this reason it has not been implemented on the real stage 2 system.

8.3 Implementation of G^T

The impulse response is sampled and applied as a multiple phase-lead update in the manner of Chapter 7, using Equation 8.9. This seems an entirely natural proposition since the single phase-lead with greatest learning was found to closely correspond to the phase-lead update of just its maximum value, so here the remainder are inserted. Another intuitive reason for using G^T is that Equation 7.4 is simplified if $\Phi(z)G(z)$ has a zero phase-lead. In this case the plot of $\Phi(z)G(z)$ is constrained to lie in the unit circle centered at $+1$. This is much easier to satisfy than the original, and, for a closed-loop

system whose magnitude ≈ 1 over the system bandwidth, should be close to $+1$. If this is not the case then an additional scalar multiplier can be applied to satisfy the equation. Figure 8.5 shows the G^T update and plots of the stability criterion $|1 - \Phi(z)G(z)|$ in

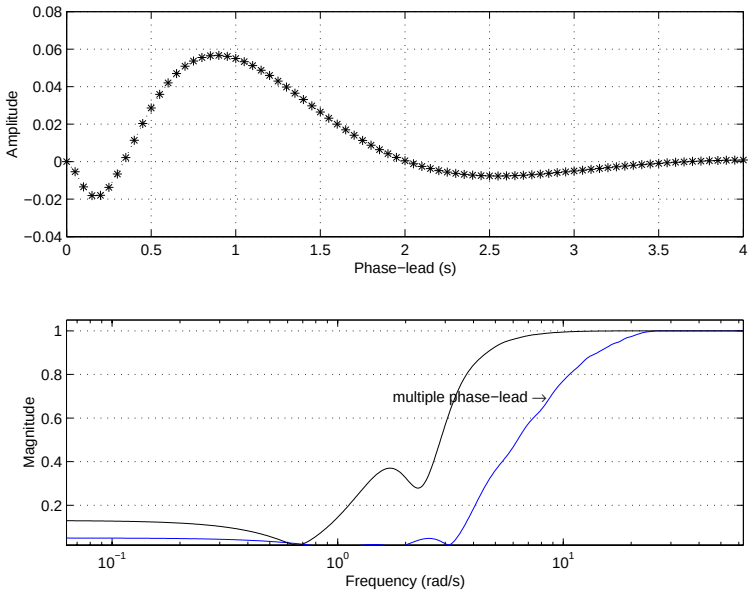


FIGURE 8.5: G^T formulation of phase-lead ILC with $\phi = 1.15$ (compared with multiple phase-lead)

comparison with that of the multiple phase-lead update.

It follows from Equation 8.7 that a sufficient condition for convergence is that, for $\omega \in [0, 2\pi]$

$$-1 < 1 - \phi|G(e^{j\omega T_s})|^2 < 1 \tag{8.10}$$

and since $|G(e^{jwT_s})| \geq 0$

$$0 < \phi < \frac{2}{\sup_{\omega\in[0,2\pi]} |G(e^{j\omega T_s})|^2} \tag{8.11}$$

A magnitude plot of the PID controlled stage 2 plant is shown in Figure 8.6. Since the maximum value of the magnitude is 1.326 (2.451 dB), in this case $0 < \phi < 1.137$ guarantees convergence for the nominal plant model. Provided Equation 8.7 is satisfied, the convergence at a frequency, w, is dictated by

$$|1 - \beta|G(jw)|^2| = 1 - \beta|G(jw)|^2 \tag{8.12}$$

(Longman, 2000), the smaller it is, the faster the convergence. It is desirable that $|G(jw)|$ equals unity at low frequencies and this is the reason the PID controller will again be used with the stage 2 system in all the tests conducted in this chapter, and the resulting system termed $G(j\omega)$.

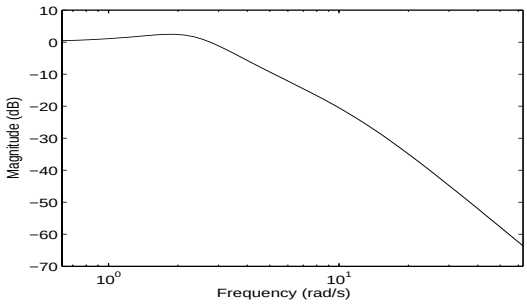


FIGURE 8.6: Bode plot of PID controlled stage 2 system

The success of G^T ILC is confirmed by the experimental results shown in Figures 8.7, 8.8 and 8.9.

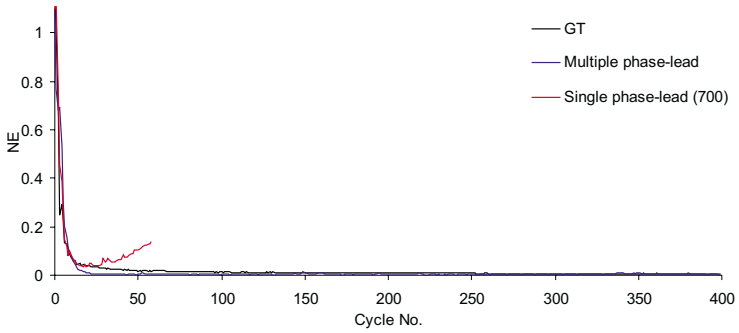


FIGURE 8.7: Error results for 20 UPM sinewave demand using single phase-lead and G^T algorithms with $\phi = 0.1$

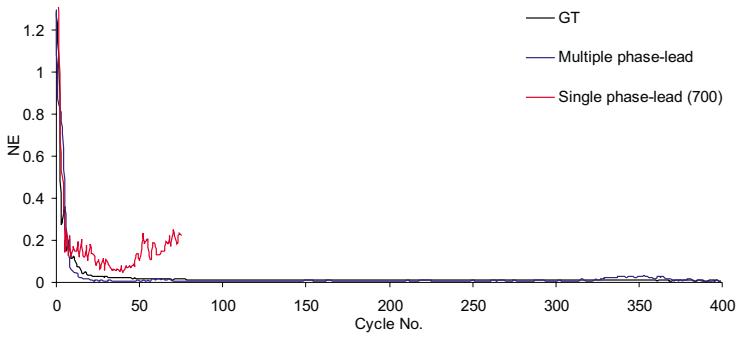


FIGURE 8.8: Error results for 20 UPM R1 demand using single phase-lead and G^T algorithms with $\phi = 0.1$

Because the length of the G^T update exceeds the length of the reference demand, updating every sample requires all the samples in the new input to be updated at every sample instant. In this case some updates will consist of amalgamates of more than one phase-lead. Since this is computationally intensive, the experiments have been implemented with updates made only every 16, 47 and 470 samples and their success indicates its

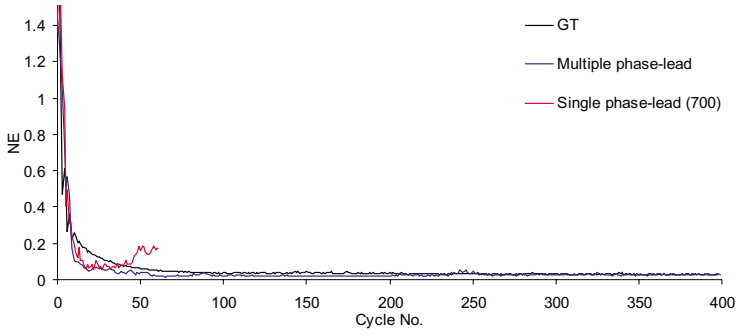


FIGURE 8.9: Error results for 20 UPM R2 demand using single phase-lead and G^T algorithms with $\phi = 0.1$

robustness to transients. It can be seen that G^T converges almost as fast as the single

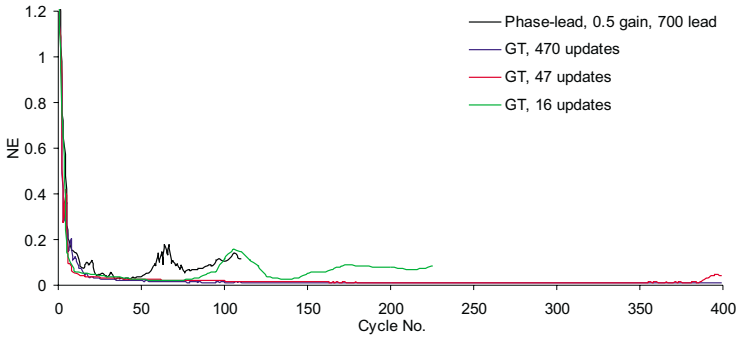


FIGURE 8.10: Error results for 20 UPM R1 demand using single phase-lead and G^T algorithms

phase-lead algorithm on a very demanding reference. It shows no sign, however, of instability. Appendix J shows results for other references. Because the 20 UPM R2 demand

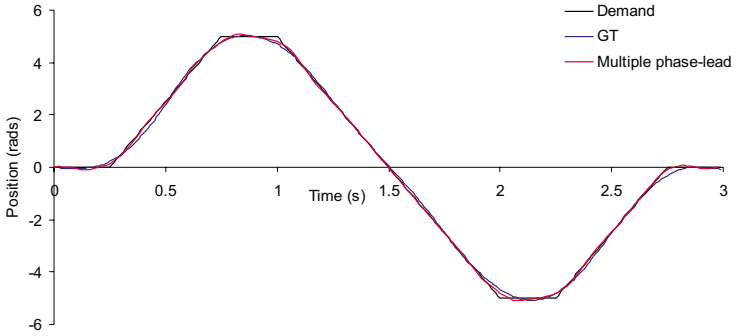


FIGURE 8.11: Comparison of G^T and the multiple phase-lead algorithms' best tracking for 20 UPM R2 demand

has higher frequency components, the superior learning of the multiple phase-lead ILC at these frequencies is evident in the rate of convergence and final error. Figure 8.11 shows its superior tracking. Figure 8.12 shows the effect of the variation of the gain multiplier, ϕ, and it is clear from comparison with Figure 7.17 of Section 7.4 that the

convergence and final error of G^T is always inferior to the multiple phase-lead update for a given ϕ. As expected, G^T produces less harmful learning transients. Alteration of ϕ can only supply certain freedoms since the learning at every frequency is affected, so the following statements are true

- The multiple-phase lead update always has more learning at high frequencies than G^T. If it was scaled to make the high frequency learning comparable then the low frequency learning would be too poor to be useful.
- Unfavourable learning transients and, to a degree, high frequency instability, are an inherent feature of the multiple phase-lead update. The extent of their effect depends on the demand, and if tolerable, the multiple phase-lead update should be used in preference to G^T and a low-pass filter applied at the highest frequency present in the demand.
- Since it was shown that the nominal plant model satisfies the monotonic convergence criteria for $0 < \phi < 1.137$, the instability seen in practice at lower values indicates a lack of robustness to plant uncertainty.

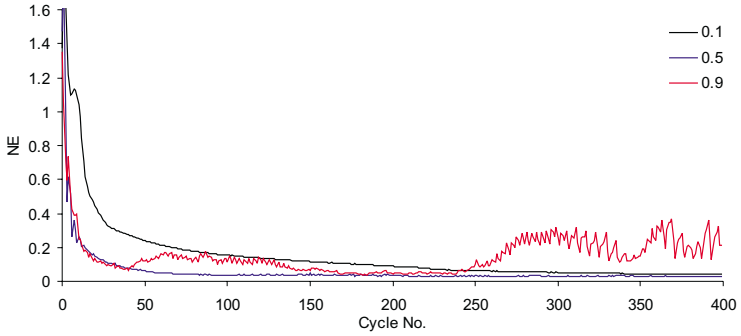


FIGURE 8.12: G^T algorithm error results for 20 UPM R2 demand with various ϕ

8.4 G^T with Reduced Plant Knowledge

Filtering has previously been applied to the error signal of a trial prior to its use in the next. If an IIR filter is used the filtering process is more restrictive in repetitive control mode as the entire error must be filtered in reverse in order to produce zero phase-lead, and this must be completed in a single sample instant. The solution that has been found to work well is to apply each input three times, recording the error, and then apply it once more during which time the error is filtered backwards in time. Three cycles of the input are used in order to minimize transients at the beginning and end, and the middle error is used as the next update. Using the impulse response of $G(s)$ has meant that G^T could always be implemented in FIR form and so the IIR implementation could be

avoided. In this section this will no longer be the case. If Equation 8.3 is written out in full as

$$\begin{bmatrix} u_k(0) \\ u_k(1) \\ u_k(2) \\ u_k(3) \\ u_k(4) \end{bmatrix} = \begin{bmatrix} CB & CAB & CA^2B & \dots & CA^{N-1}B \\ 0 & CB & CAB & \dots & CA^{N-2}B \\ 0 & 0 & CB & \dots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & CB \end{bmatrix} \begin{bmatrix} y_k(0) \\ y_k(1) \\ y_k(2) \\ y_k(3) \\ y_k(4) \end{bmatrix} \tag{8.13}$$

the rows can be interchanged in order to produce an expression involving G, as shown in Equation 8.14

$$\begin{bmatrix} u_k(4) \\ u_k(3) \\ u_k(2) \\ u_k(1) \\ u_k(0) \end{bmatrix} = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \dots & CB \end{bmatrix} \begin{bmatrix} y_k(4) \\ y_k(3) \\ y_k(2) \\ y_k(1) \\ y_k(0) \end{bmatrix} \tag{8.14}$$

If a reversal is henceforth considered to be with respect to time, then, if the reverse of an input $u(i)$ is applied to the system and the output recorded then a reversed copy of this signal is related to $u(i)$ by way of the plant transpose. This means that the update $G^T e_k$ can be generated by inputting a reversed copy of the error into the system and then reversing the corresponding output. It can only be implemented in IIR form. It is instructive to first verify the success of this technique in the ILC framework before extending it to the RC framework. As the plant provides its own transpose, this method has been termed 'automatic G^T'.

In the ILC problem we perform a trial of the input and store the error. Then the error is recorded and a reversed copy is fed into the system and the output reversed to become the new update addition, $G^T e$. Problems with this method are caused by non-linearities in the system, largely due to friction. To illustrate this, Figure 8.12 shows an idealised positional input that changes velocity direction once. The disparity

FIGURE 8.13: Slackness non-linearity

between input and output position, ζ, changes direction depending on the direction of the velocity. Assuming this positional slack has been 'taken up' at the start and that every change in velocity direction is continued until the positional slack has again been

taken up, then the value of $y(N)-u(N)$ will be $+\zeta$ for an odd number of velocity changes, and $-\zeta$ for an even number. However the output magnitude will be $< |\zeta|$ if the velocity change occurs close to the trial end. The error is shown in Figure 8.14 together with the reversed version. The velocity of both starts in one direction and then switches once. The response to this error depends on which direction the slack has been transferred

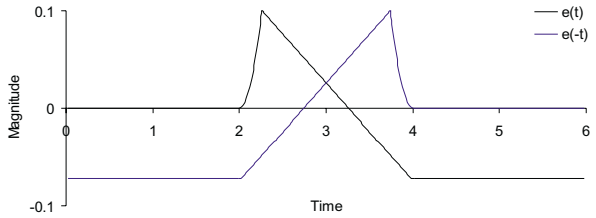


FIGURE 8.14: Error and reversed error

to. Since Figure 8.13 ended with negative velocity, if the reversed error followed directly on, then it would start with no slackness. It is an important point then, not only to ensure that reversed error follows on from the previous input, but also to seek to align the direction of slackness. These are necessary conditions in order to prevent transients which would certainly grow as there is a very sensitive learning feedback relationship centered at this point. The reversed error must therefore be offset to follow on or a ramp inserted in between. A possible output is shown in Figure 8.15. The final offset

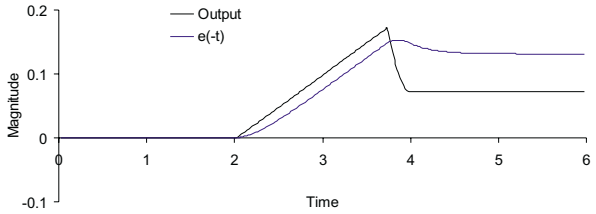


FIGURE 8.15: Response to offset reversed error

due to friction has clearly been increased since the direction of slackness of the reversed error was in the opposite direction to $e(-t)$. This is due to the error and the reversed error having an equal number of velocity changes and that the ramp, or offset required for continuity, does nothing to reset the slackness. Provided there is no instability, due to increasing transients which are caused by lack of continuity at the beginning of the reversed error or updated input, this slackness does not lead to instability. Instead it leads to a residual error in the tracking of the demand. There are, however, certain methods to reduce the impact of such non-linear characteristics:

- Insert a ramp from the end of one input to the beginning of the next to prevent transients in the output.

- To reduce the effect of the deadband, amplify the error and then scale down the resulting output by the same amount. The amplification is increased as the cycle number increases.

- To prevent a bias towards one side of the deadband, the error is flipped every alternate trial. It is flipped back to normal before being used in the update.

It is possible that the velocity crossing points could be found and the non-linearity compensated for. However the slackness of the system during input would have to be known also in order to update appropriately and the simplicity of ILC would be lost. Figure 8.16 illustrates the advantages of using these techniques, the results having been

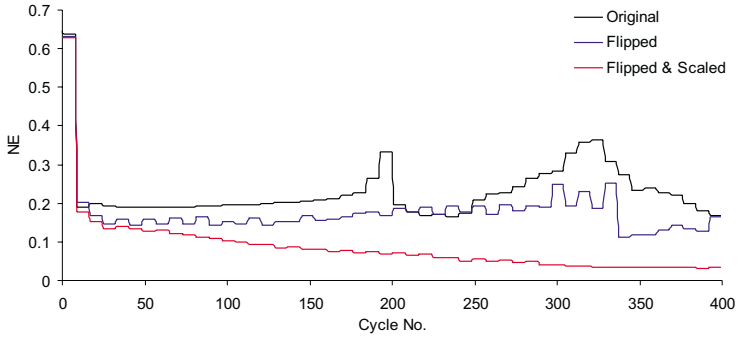


FIGURE 8.16: G^T algorithm error results for 10 UPM sinewave demand using different filtering techniques

produced within the ILC framework on the stage 2 plant. The demand was applied but with a cycle inserted either side. In the 4^{th} cycle a ramp was applied to link with the reverse error input comprising the next 3 cycles. Another ramp provided a link with the next cycle, making 8 cycles necessary for a single update.

The performance of the technique in ILC formulation suggests it to be suitable for Repetitive Control. In this case further copies of the demand were superimposed on every cycle, and superposition was used to remove the effects of the added inputs. As the cycles converge, the superpositions become smaller and the output of every trial converges to the reference. Updates are again only every 8 cycles but more frequent updates are found to come at the expense of large transients. Figure 8.17 shows that the performance of automatic G^T compares favourably with that of unfiltered single phase-lead ILC, but without its instability. Because of the slow update frequency, 1600 trials of automatic G^T have been performed which produces 200 updates. Since the phase-lead algorithm produces an update every cycle, the 200 updates shown in the graph correspond to 200 trials. Figure 8.18 shows similar results for a different demand, in this case 2000 trials of automatic G^T have been performed, and there are 250 updates. Additional results for both the RC and ILC formulations of the adjoint algorithm are presented in Appendix J.

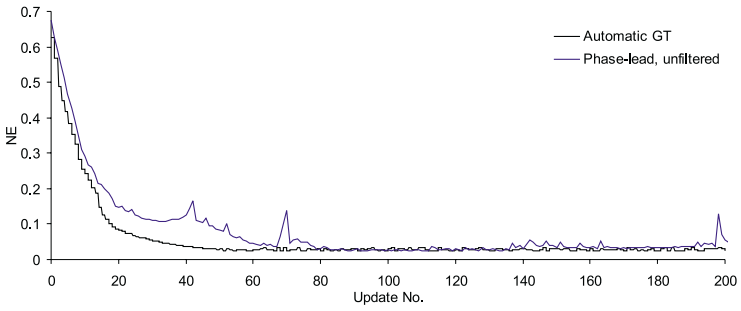


FIGURE 8.17: Error results for 10 UPM R2 demand using automatic G^T and the best single phase-lead

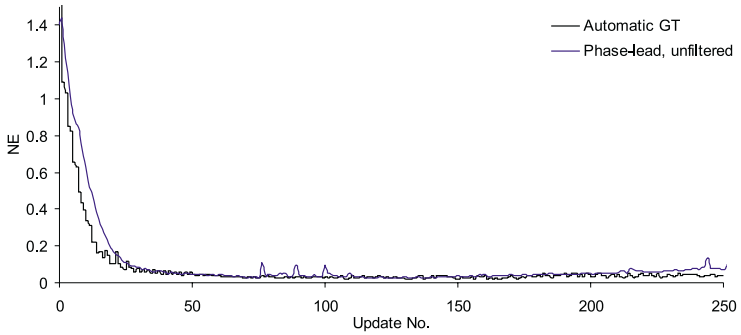


FIGURE 8.18: Error results for 20 UPM R1 demand using automatic G^T and the best single phase-lead

8.5 Extensions to G^T

Although additional compensators can always be added in series with any $\Phi(z)$, this is simplified in the G^T case because of the zero phase-lead of GG^T. If an additional compensator $F(z)$ is added then the monotonic convergence criterion is changed to;

$$|1 - FGG^T| = \sqrt{1 - 2|F||G|^2 \cos(\angle F) + 4|F|^2|G|^4} \tag{8.15}$$

Further simplicity arises if the phase-lead of $F(z)$ is zero. In this case the criterion becomes

$$|1 - FGG^T| = 1 - |F||G|^2 < 1 \tag{8.16}$$

Therefore the gain of F could be used to directly manipulate the convergence criterion. As has been discussed, a zero-phase filter can be implemented by applying an IIR filter forwards and backwards, and the way in which automatic G^T is carried out means there is time for an IIR filter to process the update in this way before the extra trial is concluded. However, in this section the original G^T algorithm is the only one considered for improvement, so a zero-phase FIR filter must be used instead. In this case a single filter $F(z)$ is again designed and implemented as a phase-lead update and then implemented directly afterwards as a phase-lag update. To obtain a single update, termed $F = QQ^T$,

it is simply concatenated with itself and scaled, a process described below. FF^T must then be concatenated with G^T in order to produce the final update, Φ.

Comparisons have been made between controllers with the same steady-state gain and it has been observed that the gain can then be used to affect a compromise between the initial convergence and divergence due to instabilities. Figure 8.19 shows a simple compensator designed to add amplification to the G^T law over certain regions. When this compensator is combined with GG^T a gain of 0.1 is used in order to ensure that the instabilty and therefore learning transients is not too great. The plots of the resulting $|1-0.1QQ^TGG^T|$are also shown. Figure 8.20 shows the experimental results achieved when

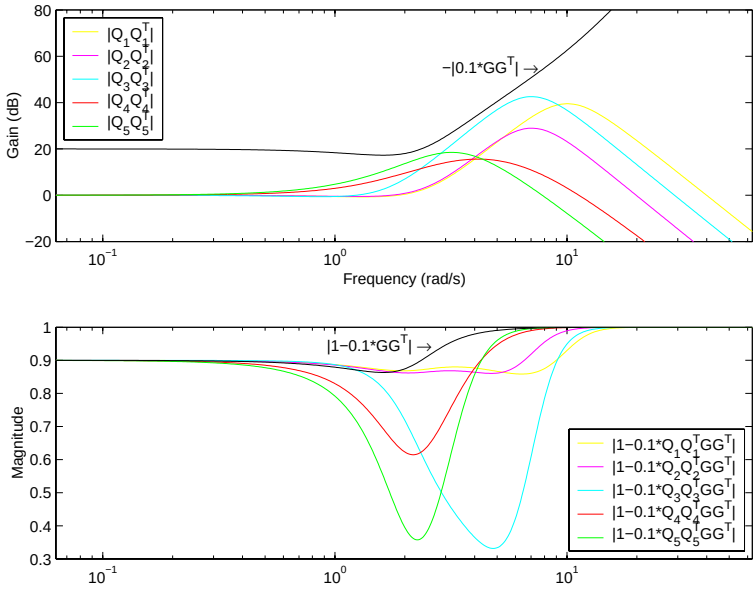


FIGURE 8.19: Design of QQ^T for additional learning to the impulse response

these compensators were tested using Equation 8.17 and a gain of 0.1. The transients were reduced by making the update resolution equal to 1. It can be seen that an improvement in convergence and final error is possible, but that too much high frequency amplification causes instability.

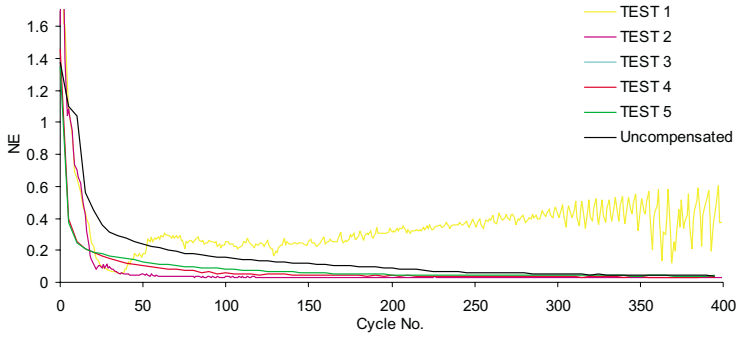


FIGURE 8.20: Success of compensators using 10 rads^{-1} 20 UPM R2 demand

It has been observed that QQ^T can be achieved by concatenating the sampled impulse responses. Care must be taken that the sample rate is sufficiently high, if Q has a low frequency cut-off then this can be achieved quite easily. The impulse response of G can then be concatenated with this to produce a single update. If $Q_i, (i = 1, 2, \ldots N)$ is the impulse response for Q, and $G_j, (j = 1, 2, \ldots M)$ is the impulse response for G then the resulting update is given by

$$\begin{aligned} QQ^T_{N+i-m} &= \sum_{i=0}^{N-1} \sum_{m=0}^{N-1} Q_{N+m} Q_{N+m-i} \\ G^T Q Q^T_{N+i-m+j} &= \sum_{j=0}^{M-1} \sum_{N+i-m=1}^{2N-1} Q^T Q_{N+m-i} G_j \end{aligned} \tag{8.17}$$

Figure 8.21 shows a different set of compensators, Q, using a gain of 0.5. The larger gain shows the effect of instability more clearly. Only tests 1 and 5 show divergence and the initial convergence and almost immediate divergence is caused by the learning transients. Only test 1 injects amplification which exceeds that of the multiple phase-lead update which suggests that that update was less prone to learning transients than the method implemented here. A partial cause of these transients can be found by

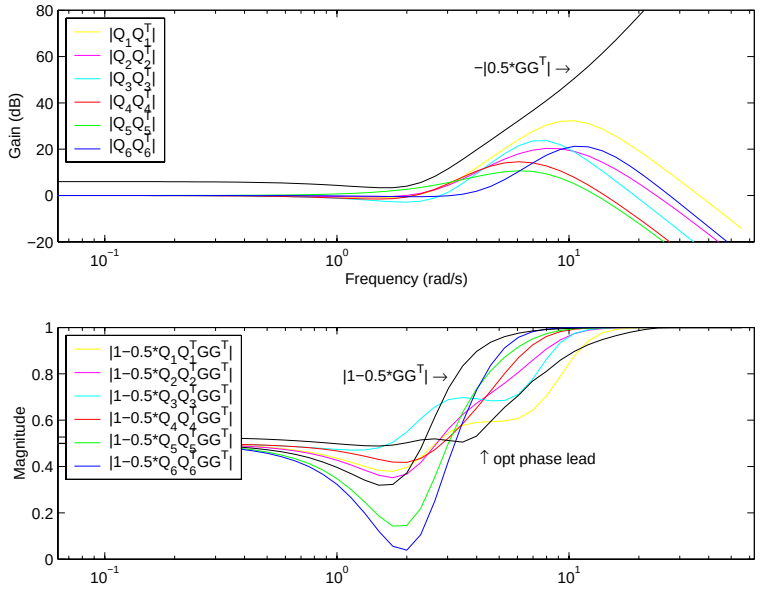


FIGURE 8.21: Design of QQ^T for additional learning to the impulse response

examining the update which causes them, such as that shown in Figure 8.22. Because the implementation of QQ^T necessarily incorporates phase-lag, the output $u(i)$ continues to be updated by errors occurring immediately after instant i. This means that, not only is the filtering process of the most recent learning update only half complete, but that the integral of all the future errors which contribute to $u(i)$ may make the outputted value of $u(i)$ quite erroneous. Because the response to this input determines the next error, the learning is very unstable. Neither G^T nor multi-phase-lead updates have

phase-lag which reduces their susceptibility to these learning transients. Figure 8.23

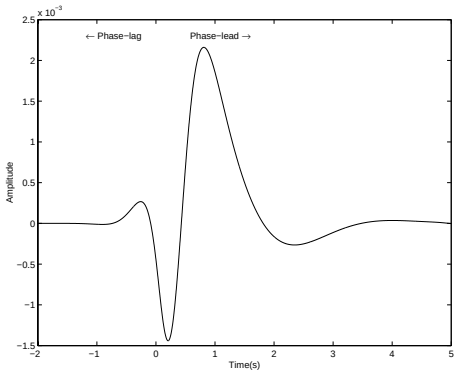


FIGURE 8.22: $|1 - 0.5 \times Q_5 Q_5^T G G^T|$ update

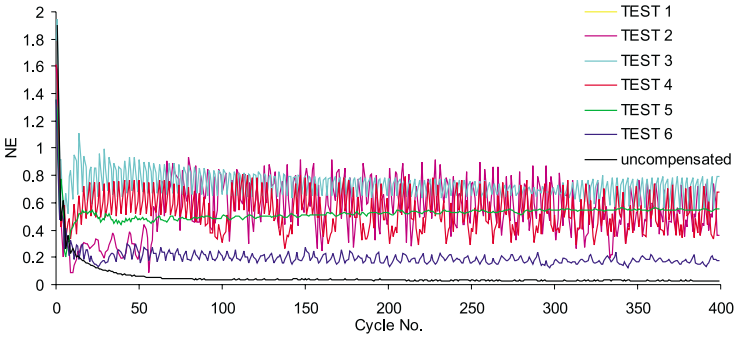


FIGURE 8.23: Success of compensators using 10 rads^{-1} 20 UPM R2 demand

shows the large sensitivity to transients. The oscillations are partly due to reasons given, and partly due to mid-range frequencies on the edge of stability whose phase-lead causes them to drift in and out of phase with their correct position, thereby causing oscillations in cycle error. The experiments have suggested that compensators of the type described produce far more transients than an equivalent multiple phase-lead update.

8.6 Summary

The learning law utilising the plant transpose has found to fit naturally into the framework introduced in Chapter 7. Practical results have been presented to assess its performance. This algorithm, which requires a model of the plant, is reformulated into one which needs little plant knowledge. Results have been presented using this technique and practical guidelines have been produced and tested to improve its performance. A simple method of increasing the learning at higher frequencies has been proposed and practical limitations have been addressed and verified experimentally.

The performance of the adjoint algorithm has been compared with that of the multiple phase-lead and best performing single phase-lead. It has been found to offer far greater

performance than the single phase-lead law and be less sensitive to high frequency instability than the multiple phase-laed algorithm. The adjoint algorithm, however, does have the disadvantage of slower convergence when compared to the multiple-lead law. Since both these laws have been shown to satisfy the monotonic convergence criteria for frequencies upto and exceeding the system bandwidth, it can be surmised that the adjoint algorithm has a greater robustness to plant uncertainty than the multiple phase-lead law.

Chapter 9

Optimality Based Algorithm

9.1 Introduction

All previously implemented algorithms have followed on from one another in an intuitive way and share a common framework. In order to broaden the type of methods considered, a law with a different structure will be considered in this chapter. Recently, a novel optimality based Repetitive Control algorithm was proposed in (Hatonen et al., 2003c). According to the convergence analysis, the algorithm will result in asymptotic convergence for an arbitrary discrete-time LTI plant and a T-periodic reference. However, the performance of the algorithm was tested only using simulation studies. In order to rigorously test how the algorithm performs with real systems has been implemented on the stage 2 system.

9.2 Algorithm Derivation

The derivation uses a combination of the polynomial systems approach presented in (Blomberg and Ylinen, 1983) and optimal control. The first creates a regulator system that ensures the output is tracked, and the second provides a feedback controller that drives the output $e(t)$ of this modified system to zero. Consider a process model defined for $t \in \mathbb{Z}$

$$A(z^{-1})y(t) = B(z^{-1})u(t) \tag{9.1}$$

where $A(z^{-1}), B(z^{-1}) \in \mathbb{C}[z^{-1}]$. It is assumed that this model is both controllable and observable and that a feedback controller is used to make the output track the reference. As a starting point note that because the reference signal is T-periodic (i.e. $y_d(t+T) = y_d(t)$), the polynomial $D(z^{-1}) = 1 - z^{-T}$ is a annihilator (or an internal

model) for $y_d(t)$, i.e.

$$D(z^{-1})y_d(t) = y_d(t) - z^{-T}y_d(t) = y_d(t) - y_d(t-T) = 0. \tag{9.2}$$

Next, both sides of Equation 9.1 are multiplied with $D(z^{-1})$ to give

$$\begin{aligned} &D(z^{-1})A(z^{-1})y(t) = D(z^{-1})B(z^{-1})u(t) \\ &= B(z^{-1})D(z^{-1})u(t) \\ &B(z^{-1})\tilde{u}(t) \end{aligned} \tag{9.3}$$

where $\tilde{u}(t) := u(t) - u(t-T)$. The left-hand side of Equation 9.3 can be written as

$$\begin{aligned} &D(z^{-1})A(z^{-1})y(t) = A(z^{-1})D(z^{-1})y(t) \\ &= A(z^{-1})\left(y(t) - y(t-T)\right) \\ &= A(z^{-1})\left(y(t) - y_d(t) + y_d(t-T) - y(t-T)\right) \\ &= A(z^{-1})\left(-e(t) + e(t-T)\right) \\ &= -D(z^{-1})A(z^{-1})e(t) = \tilde{A}(z^{-1})e(t). \end{aligned} \tag{9.4}$$

where $\tilde{A}(z^{-1}) := -D(z^{-1})A(z^{-1})$. Combining the last two equations produces

$$\tilde{A}(z^{-1})e(t) = B(z^{-1})\tilde{u}(t) \tag{9.5}$$

which is a controllable and observable dynamical regulation system, if $D(z^{-1})$ and $B(z^{-1})$ are coprime. Let this modified system have the state-space representation

$$\begin{aligned} x_m(t+1) &= A_m x_m(t) + B_m \tilde{u}(t) \\ e(t) &= C_m x_m(t) \end{aligned} \tag{9.6}$$

where the dimension of $x_m(\cdot)$ is $n+T$, and n is the order of the original process model. Consider the standard optimisation problem

$$\min_{\tilde{u} \in l_2} J(\tilde{u}, x_m(0)) \tag{9.7}$$

where

$$\begin{aligned} &J(\tilde{u}, x_m(0)) = \textstyle\sum_{i=1}^{\infty} e(i)^T Q e(i) + \tilde{u}^T(i) R \tilde{u}(i) \\ &= \textstyle\sum_{i=1}^{\infty} x_m(i)^T C_m^T Q C_m x_m(i) + \tilde{u}^T(i) R \tilde{u}(i) \end{aligned} \tag{9.8}$$

and Q and R are symmetric positive-definite weighting matrices. The well-known solution of the optimisation problem is given by the control law $\tilde{u}(t) = -Kx_m(t)$ or

$$u(t) = u(t-T) - Kx_m(t) \tag{9.9}$$

where K is given by the equation

$$K = (B_m^T S B_m + R)^{-1} B_m^T S A_m \tag{9.10}$$

and S is obtained from the algebraic Riccati equation

$$S = A_m^T[S - SB_m(B_m^T SB_m + R)^{-1}B_m^T S]A_m + Q \tag{9.11}$$

Since in practice it is not possible to measure the state $x_m(\cdot)$ directly, an observer will be used to estimate them using

$$\hat{x}_m(t+1) = A_m\hat{x}_m(t) + B_m\tilde{u}(t) + L(e(t) - C_m x_m(t)) \tag{9.12}$$

where L is the observer gain and the control law becomes

$$u(t) = u(t-T) - K\hat{x}_m(t) \tag{9.13}$$

If the modified system also has noise terms, it becomes,

$$\begin{aligned} x_m(t+1) &= A_m x_m(t) + B_m\tilde{u}(t) + Gw(t) \\ e(t) &= C_m x_m(t) + v(n) \end{aligned} \tag{9.14}$$

where $w(t)$ and $v(t)$ are zero mean Gaussian noise, $w(t)$ describing uncertainty in the state-space model and $v(t)$ describing uncertainty in the measurement process. If the covariance matrix Q_n of $v(t)$ and the covariance matrix R_n of $w(t)$ are known, it is possible to find an optimal observer gain L that minimises the variance of the estimation error. It is also a standard result in optimal control (Lewis and Syrmos, 1995) that by combining the optimal feedback controller and optimal observer the resulting closed-loop system is stable, and hence the expected value of $e(t)$ will go to zero as $t \to \infty$. The flow-diagram of the proposed algorithm is shown in Fig 9.1. It was also shown (Hatonen

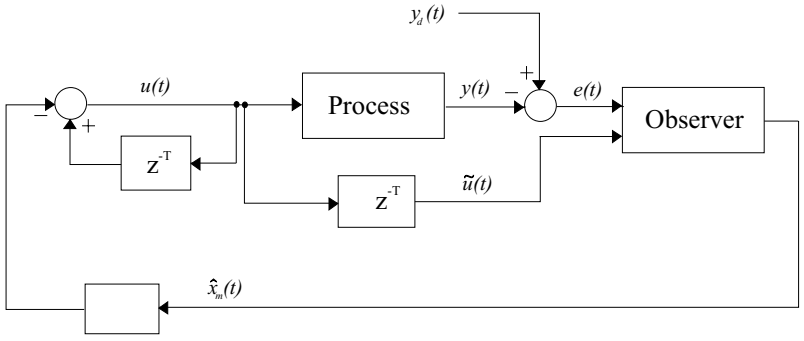


FIGURE 9.1: The flow diagram of the Repetitive Controller

et al., 2003c) that this approach also works for more complex reference signals, a typical example being multi-periodic reference signals of the form $y_d(t) = y_{d1}(t)+y_{d2}+\cdots+y_{dn}(t)$ where $y_{di}(t) = y_{di}(t+T_i)$. In this case the annihilator is the multiplication of all the separate annihilators.

9.3 Experimental Results

A PID loop around the plant is used in order to pre-stabilise it and provide greater stability. This approach has been found to afford greater success than when just using the open-loop plant. The values of the optimal weighting matrices, Q and R, were set at $10 \times C^T C$ and 1 and the Kalman covariance matrices, Q_n and R_n, were set at 10 and 1. A mathematical software package was used to calculate K and L, taking several hours and a large amount of memory, and thereby restricting the sampling frequency to 250Hz. Due to the sparseness of the resulting modified system given by Equation 9.5, the algorithm can be programmed very efficiently by partitioning the large $\tilde{A}$ matrix into a small non-sparse matrix and an off-diagonal matrix of ones. This results in two interacting state-space systems but only the one using the small matrix has to be calculated at each sampling instant. Instead of recalculating the larger system, its states can simply be renumbered at each sampling instant. The computational workload is therefore concentrated on the equations involving the K and L multiplications. Figures 9.2 and 9.3 show the initial cycles using the 20 UPM sinewave demand and the 20 UPM R1 demand respectively. In both cases the demand is followed well by the 3^{rd} trial,

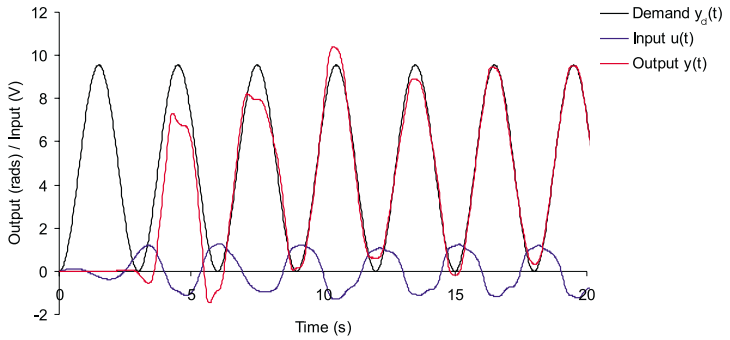


FIGURE 9.2: Optimal algorithm 20 UPM sinewave demand convergence

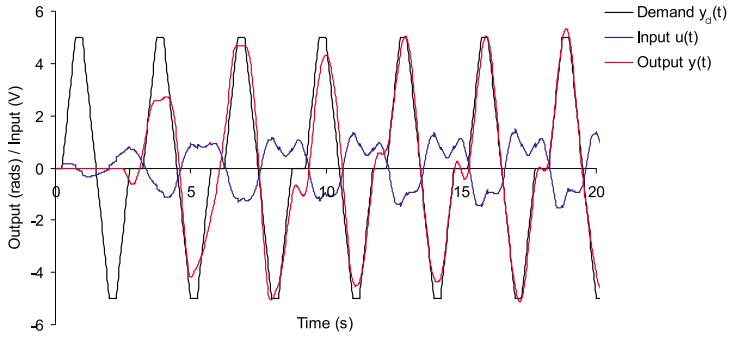


FIGURE 9.3: Optimal algorithm 20 UPM R1 demand convergence

and tracking is almost perfected by the 6^{th}. The corresponding input to the plant is also shown.

In order to increase robustness, a relaxation parameter, γ, was inserted in Equation 9.13 just as a forgetting factor was applied to the phase-lead algorithm in Chapter 6. This produces the control law

$$u(t) = \gamma u(t-T) - K\hat{x}_m(t) \tag{9.15}$$

where K is a function of γ since the system structure of Equation 9.5 has been altered. It is therefore necessary to recalculate it and the observer and kalman filter gains for different γ. Figures 9.4 and 9.5 compare results obtained using different values of γ. It

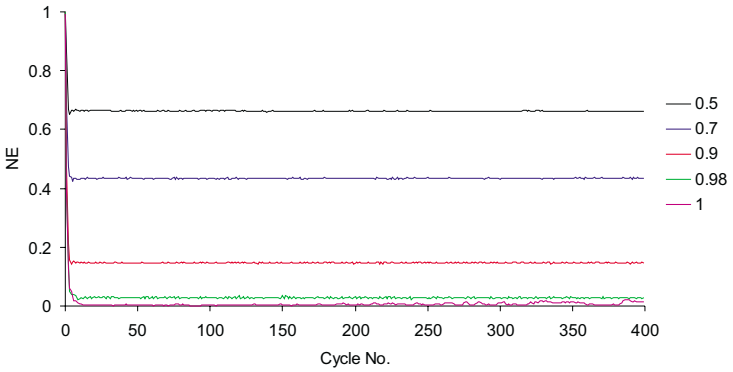


FIGURE 9.4: Optimal algorithm error results for 20 UPM sinewave demand with relaxation

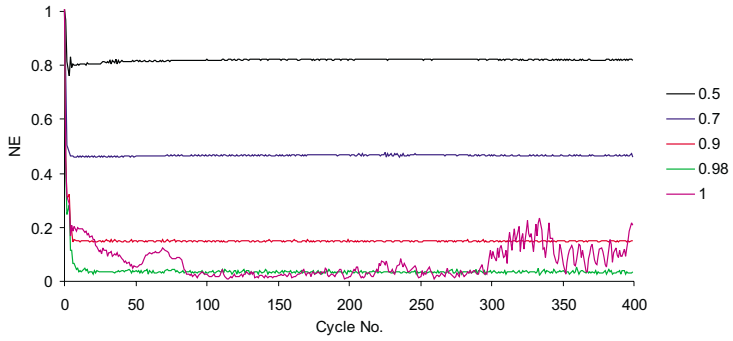


FIGURE 9.5: Optimal algorithm error results for 20 UPM R1 demand with relaxation

can be seen that for the 20 UPM R1 demand divergence occurs during later trials when using $\gamma = 1$. Divergence is prevented by instead using smaller values of γ, incurring a penalty to the final error bound reached depending on how close it is to 1. The relaxation parameter can be seen to sacrifice final error for stability which coincides with observations made concerning the use of the forgetting factor with the phase-lead algorithm. Figures 9.6 and 9.7 compare the best results produced with this scheme with the best of other algorithms. Gains of $\phi = 0.5$ have been used in each case. It

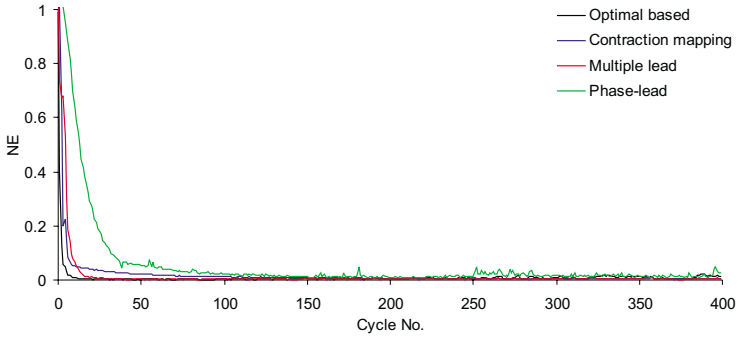


FIGURE 9.6: Error results for 20 UPM sinewave demand using different ILC schemes

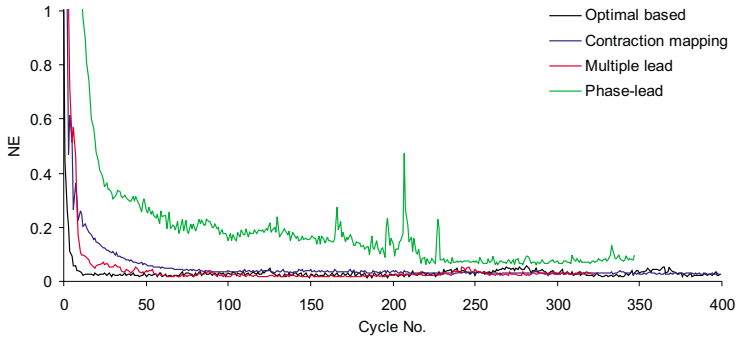


FIGURE 9.7: Error results for 20 UPM R1 demand using different ILC schemes

is clear that the optimality based algorithm has hugely improved convergence to these schemes, a similar final error bound and little sign of greater instability. To match its convergence rate, the gain of these other schemes can be increased but this always leads to the rapid onset of instability. The relationship between sucessive cycle errors for the update law given in Equation 9.13 is given by

$$e_{k+1} = \frac{\gamma}{HG+1} e_k(t) + \frac{1-\gamma}{HG+1} y_d(t) \tag{9.16}$$

where

$$H = \frac{-KL}{Iz - (A_m - LC_m - B_m K)} \tag{9.17}$$

A plot of the left hand side of the sufficient convergence condition (Owens, 1992)

$$\left| \frac{\gamma}{HG+1} \right| < 1 \tag{9.18}$$

is shown in Figure 9.8 for several values of γ. The second term on the right hand side of Equation 9.16 dictates the level of final error and is very small for γ close to 1. It is clear that a reduced γ helps satisfy the condition and suggests the replacement of γ with a filter. This could then be designed not only to satisfy the condition but also to add

learning at those frequencies present in the demand. The use of a filter would allow much greater control over the stability and convergence properties of the scheme. That fact that the convergence condition is clearly not satisfied for all frequencies upto the system bandwidth suggests that not satisfying the condition carries a lighter instability penalty than it did for the previous algorithms that have been implemented. The differences in the respective algorithm structures is investigated in the next chapter.

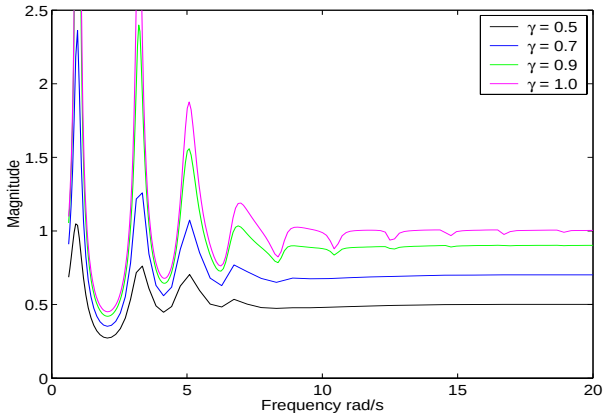


FIGURE 9.8: Convergence condition for different γ values

9.4 Summary

An optimality based RC scheme that has previously produced encouraging simulation studies has been implemented on the stage 2 plant. Experimental results using the technique on this facility have been presented and excellent performance has been observed in terms of convergence rate, stability and the final error bound. The results have been compared to those of other ILC schemes and its performance appears superior in terms of convergence rate and it is seen to provide a similar level of final error.

Chapter 10

Algorithm Robustness

10.1 Introduction

The P-type and phase-lead algorithms have been shown to have almost certain instability since they do not satisfy the sufficient monotonic convergence criterion, given by

$$|1 - \Phi(e^{jwT})G(e^{jwT})| < 1 \tag{10.1}$$

which was discussed in Chapter 7. The same criterion was found to ensure stability for the adjoint and multiple phase-lead algorithms for the stage 2 plant model over a large range of frequencies. Since the PID controller has been used as a pre-stabiliser for the plant in all the algorithms tested, $G(e^{jwT})$ represents this closed-loop system. Unfortunately this model does not represent the actual system perfectly, and the convergence properties observed using the model may only apply to systems that do. This section examines the changes in the convergence condition that occur when the actual plant equals the model with an multipicative uncertainty, $\Delta G_m(e^{jwT})$. In this case

$$G_p(e^{jwT}) = G(e^{jwT})(1 + \Delta G_m(e^{jwT})) \tag{10.2}$$

where $G_p(e^{jwT})$ is the actual plant. Here the robustness properties of the algorithms are not examined in a general setting, but specifically with the stage 2 system, on which all but the D-type law have been used. The findings can be applied to an additive uncertainty, $\Delta G_a(e^{jwT})$ with the substitution

$$\Delta G_m(e^{jwT}) = \frac{\Delta G_a(e^{jwT})}{G(e^{jwT})} \tag{10.3}$$

The notation $G(e^{jwT}) = G$, $\Delta G_m(e^{jwT}) = \Delta G_m$ has been adopted for conciseness in the remainder of this section.

10.2 Stability Criterion for the Uncertain System

The uncertainty changes the location of the open-loop learning system, ΦG, in the complex plane, ΦG becomes $\Phi G(1 + \Delta G_m)$. This is shown in Figure 10.1. It is clear that, by the application of either $\angle(1 + \Delta G_m)$ or $|1 + \Delta G_m|$, the point $\Phi G(1 + \Delta G_m)$ can be made to move out of the stability region, that is the unit circle, centred on $+1$, for every w. For a given value of $\angle(1 + \Delta G_m)$, the value of $|1 + \Delta G_m|$ can be calculated which causes $\Phi G(1 + \Delta G_m)$ to touch the unit circle. This creates a mapping of the stability boundary in terms of $\angle(1 + \Delta G_m)$ and $|1 + \Delta G_m|$. Alternatively $|1 + \Delta G_m|$ can be increased and the value of $\angle(1 + \Delta G_m)$ found, this produces the same result but is more difficult to ensure every value has been considered. The discussion will

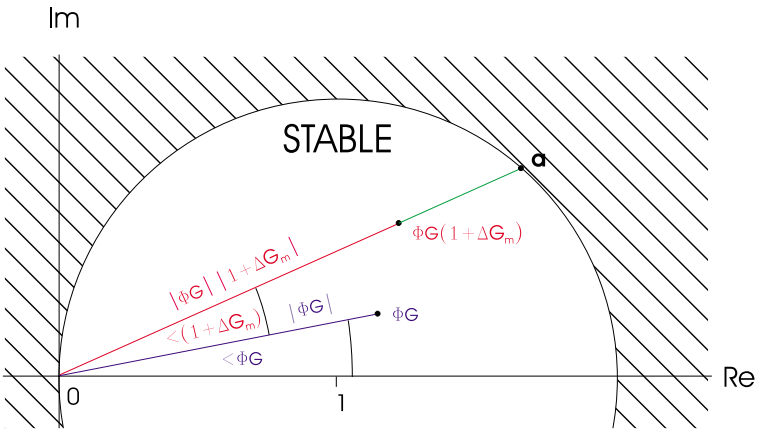


FIGURE 10.1: Graphical representation of convergence criterion

be restricted to consider only the principle argument of $\angle\Phi G + \angle(1 + \Delta G_m)$, that is $\angle\Phi G + \angle(1 + \Delta G_m) \in [-\pi\ \pi]$. In the diagram, '$a$' is the point at which the stability boundary is met. Its magnitude is given by

$$|a| = 2\cos(\angle\Phi G + \angle(1 + \Delta G_m)) \tag{10.4}$$

with the condition

$$\cos(\angle\Phi G + \angle(1 + \Delta G_m)) = 0\ \ \forall\ \ w\ :\ \frac{\pi}{2} < \angle\Phi G + \angle(1 + \Delta G_m) + 2\pi n < -\frac{\pi}{2} \tag{10.5}$$

where n is an integer. Substituting for 'a', the stability criterion (Equation 10.1) for the uncertain system is now,

$$|1 + \Delta G_m| < \frac{2\cos(\angle\Phi G + \angle(1 + \Delta G_m))}{|\Phi G|} \tag{10.6}$$

In terms of $|\Delta G_m|$ and $\angle G_m$ the necessary substitutions are

$$\angle(1 + \Delta G_m) = \tan\left(\frac{|\Delta G_m|\sin(\angle\Delta G_m)}{1 + |\Delta G_m|\cos(\angle\Delta G_m)}\right) \tag{10.7}$$

and

$$|1 + \Delta G_m| = \sqrt{1 + |\Delta G_m|^2 + 2|\Delta G_m| \cos(\angle \Delta G_m)} \tag{10.8}$$

Substitution of these into Equation 10.6 yields

$$\sqrt{1 + |\Delta G_m|^2 + 2|\Delta G_m| \cos(\angle \Delta G_m)} < \frac{2 \cos\left(\angle \Phi G + \tan\left(\frac{|\Delta G_m| \sin(\angle \Delta G_m)}{1 + |\Delta G_m| \cos(\angle \Delta G_m)}\right)\right)}{|\Phi G|} \tag{10.9}$$

This is an equation linking the uncertainty terms, solely as a function of the known system model. If the inequality is replaced with an equality then a mapping from the uncertainty terms to the stability boundary is described. However, solving this for $|\Delta G_m|$ analytically is a difficult task and so the uncertainty will be kept in the form of $1 + \Delta G_m$. This can be treated as a plant uncertainty in its own right. This uncertainty, which is in series with the plant, will be termed ΔG, and is given by

$$\Delta G = 1 + \Delta G_m \tag{10.10}$$

This uncertainty has a greater effect on the plant per unit variation than the additive and multiplicative uncertainties since $|\Delta G|$ scales the magnitude of the plant, and $\angle \Delta G$ adds directly to its phase. Replacing this in Equation 10.6 and setting it to an equality produces

$$|\Delta G| = \frac{2 \cos(\angle \Phi G + \angle \Delta G)}{|\Phi G|} \tag{10.11}$$

For each frequency, w, this expression can be used to locate the stability boundary in terms of the uncertainty ΔG. Since Equation 10.11 is a function of w, $|\Delta G|$ and $\angle G$, it can be plotted in 3 dimensions to show the location of the stability boundary for an uncertain system. This has been done for the stage 2 system in conjunction with a variety of algorithms.

Assuming ΦG has no unstable poles, an application of the Nyquist Stability Criterion to the geometry of Figure 10.1 shows that if the point +1 is not encircled, the system $1 - \Phi G$ has no unstable zeros. Note that the traditional point of interest, -1, has been reflected in the imaginary axis since ΦG has been plotted instead of $-\Phi G$.

10.3 Robustness Plots

In the following plots, for every frequency (plotted on the x-axis), the range of $\angle \Delta G$ is given on the y-axis for values between $\pm\pi$. For values $\angle \Delta G + n\pi$, where n is an integer, the values are the same as for the principle argument. The value of

$|\Delta G|$ (plotted on the z-axis) can then be examined to see if the system is stable. If $0 \leq |\Delta G| < 1$ then the actual system magnitude must be smaller than the model for stability by this magnitude and so the nominal system is unstable.

If $|\Delta G| \leq 1$ then Equation 10.1 is not satisfied and it is unlikely that the learning system is stable. For values $|\Delta G| > 1$, the magnitude shows by what factor $|G|$ can be multiplied and stability still ensured. The plots do not show values of the z-axis greater than 5 in order to focus on the areas close to the stability border. In every algorithm a gain of $\phi = 1$ has been used but reduction of ϕ simply elongates the plot in the z-axis direction. Another general point is that, for every plot, more than half the range of $\angle \Delta G$ must correspond to $|\Delta G| = 0$ for every frequency as $\Phi G \Delta G$ rotates into the left half plane. The plane $|\Delta G| = 1$, corresponding to zero phase uncertainty, has a displaced copy of the x-axis/y-axis grid overlaid on it. Therefore, for a given frequency and phase uncertainty, if this grid is broken by the plot then the system has a degree of robustness at those points.

Figure 10.2 shows the plot when using the P-type algorithm, that is $\Phi = 1$. With no phase uncertainty ($\angle \Delta G = 0$), the system is almost certainly unstable for frequencies in the range 1-7 rads^{-1} since the corresponding gain uncertainty, $|\Delta G|$, is approximately zero. Because the demands used have significant frequency content upto 10 rads^{-1}, it is these frequencies which consistitute the most important range where learning and robustness are most necessary. The P-type controller does not satisfy the criterion for the majority of this region and also for for a large section at higher frequencies. Figure 10.3 shows an 'overhead' view of the plot. The areas

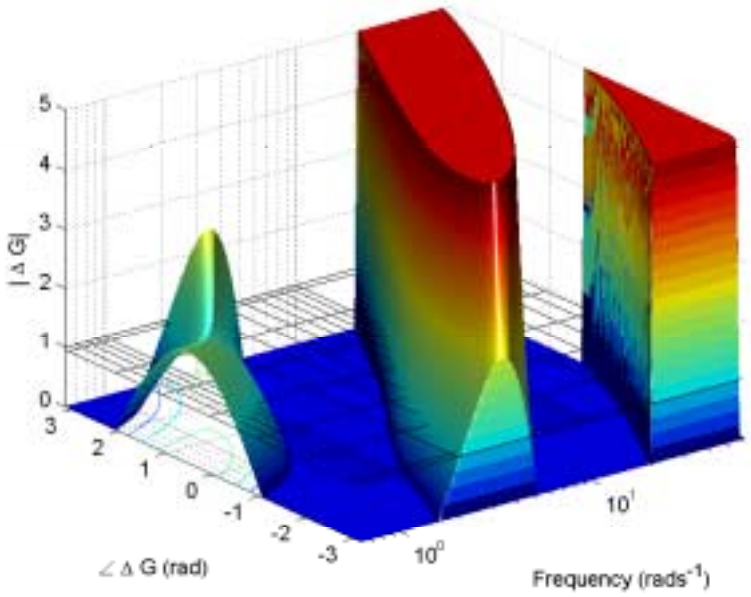


FIGURE 10.2: Robustness plot for the P-type algorithm

where the grid is broken are robustly stable, and the degree of this robustness is indicated by the warmth of their colour.

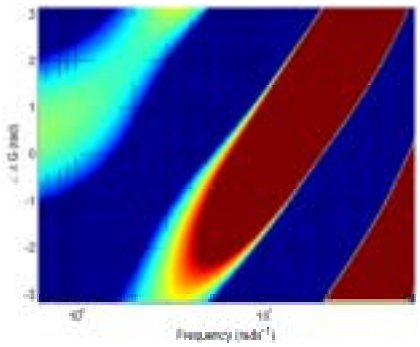


FIGURE 10.3: X-Y plane view of Figure 10.2

The phase-lead algorithm, shown in Figure 10.4, is given by $\Phi = e^{jwT\lambda_{opt}}$ (where λ_{opt} is the best performing phase-lead in practice) and has a far more robust plot than the P-type law since $|\Delta G|$ attains greater values for zero phase uncertainty. It should be expected that $|\Delta G| \approx 2$ at low frequencies and low values of phase uncertainty since this is consistant with it being an inverse of the model. (This is only the case when $|G| \approx 1$ at low frequencies which is true for the closed-loop stage 2 plant). This is the case for the phase-lead algorithm, but at high frequencies there is a period where $|\Delta G| < 1$ in the region of 2.5 rads^{-1}. The 'tunnel' of convergence around $\angle \Delta G = 0$ is constricted at this point showing not only a decrease in robustness, but instability for the nominal plant model at certain frequencies. The most prominent feature of the phase-lead algorithm is the diagonal regions of almost certain instability. Figure 10.5 shows the 'overhead'

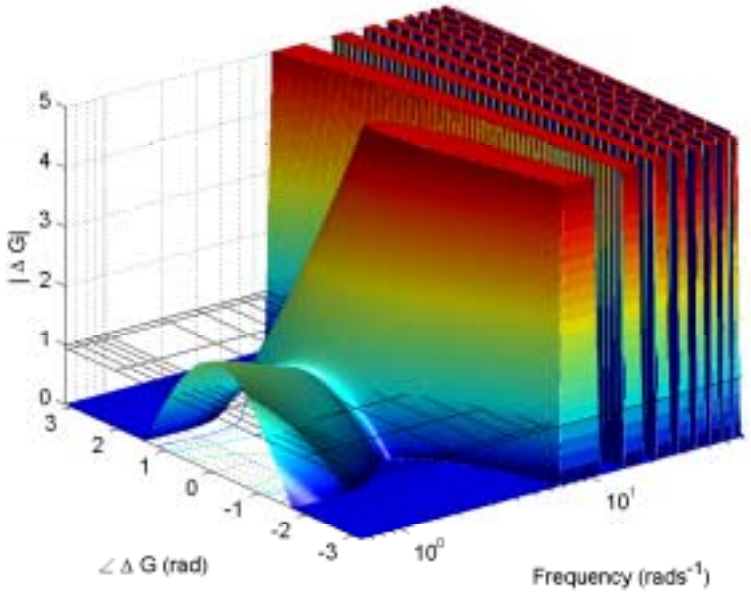


FIGURE 10.4: Robustness plot for the phase-lead algorithm with λ_{opt}

view of Figure 10.4.

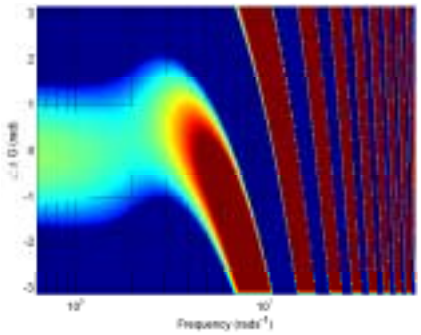


FIGURE 10.5: X-Y plane view of Figure 10.4

Figure 10.6 shows the robustness plot of the multiple phase-lead algorithm. The tunnel of convergence is free from constriction. Its magnitude is approximately 2 with no phase uncertainty up until about 5 rads^{-1} showing that the algorithm approximates the plant model inverse at these frequencies. The robustness properties of the algorithm slightly favour negative phase uncertainty in the region of 10 rads^{-1} but are hugely displaced towards positive phase uncertainty at high frequencies. This shows almost certain high frequency instability but would take a huge number of cycles to accumulate as $|G|$ is small. Figure 10.7 shows the

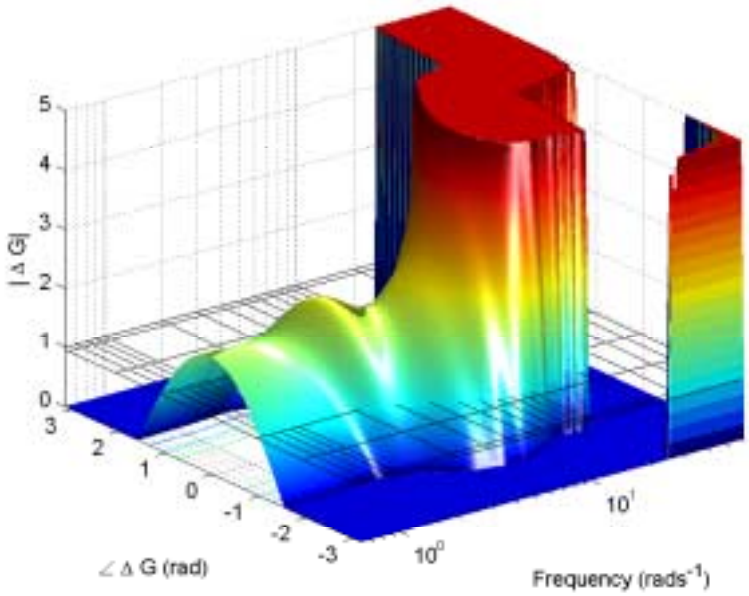


FIGURE 10.6: Robustness plot for the multiple phase-lead algorithm

'overhead' view of Figure 10.6.

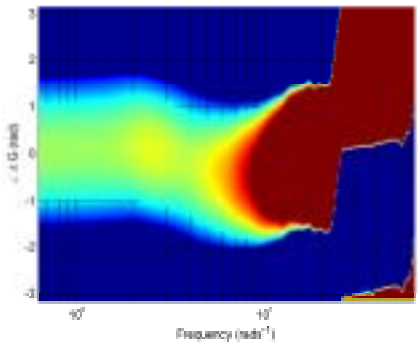


FIGURE 10.7: X-Y plane view of Figure 10.6

The robustness plot for the adjoint algorithm is shown in Figure 10.8. Almost definate instability is obvious at around 3 rads^{-1} since $|\Delta G| < 1$. This has been avoided in practice by selecting $0 \leq \phi < 1$. The failure of the algorithm as an approximate inverse at low frequencies is shown by the magnitude of $|\Delta G|$ which cannot be close to 2 for any great frequency range whatever value of ϕ is taken. This illustrates the compromise that is been experienced in the design of Φ: Robustness demands it to be large as soon as possible, but for rapid convergence it should also approximately equal 2. As well as being robust at low frequencies, the adjoint algorithm reaches very large values of $|\Delta G|$ and maintains them for $-\frac{\pi}{2} < \angle\Delta G < \frac{\pi}{2}$ (which is as much as can be possibly expected). Figure 10.9

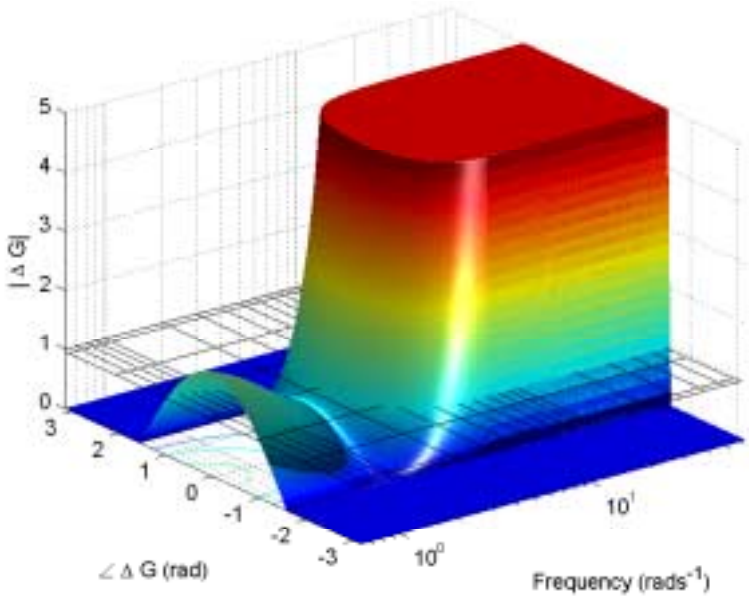


FIGURE 10.8: Robustness plot for the adjoint algorithm

shows the 'overhead' view of Figure 10.8.

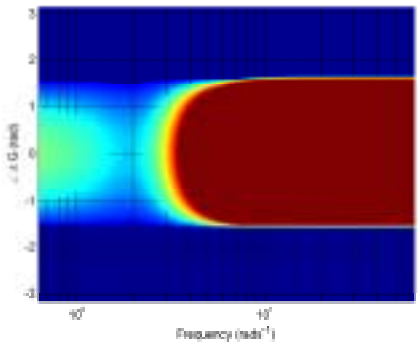


FIGURE 10.9: X-Y plane view of Figure 10.8

If the system inverse is plotted (i.e. setting $\Phi = G^{-1}$), its lack of robustness can be clearly seen. Even low frequencies suffer from a lack of robustness since a gain uncertainty greater than 1 is not present with any great degree of phase uncertainty present. It has been observed that the build-up of unstable high frequencies will be very slow, so the lack of robustness over the upper frequencies present in the demand is the principle concern. This is because a filter can always be used to simply remove the high frequencies. For $|\Delta G| = 1$, stability is only ensured with $\frac{\pi}{3} < \angle \Delta G < \frac{\pi}{3}$ at all frequencies, so the algorithm is not robust. Figure 10.11

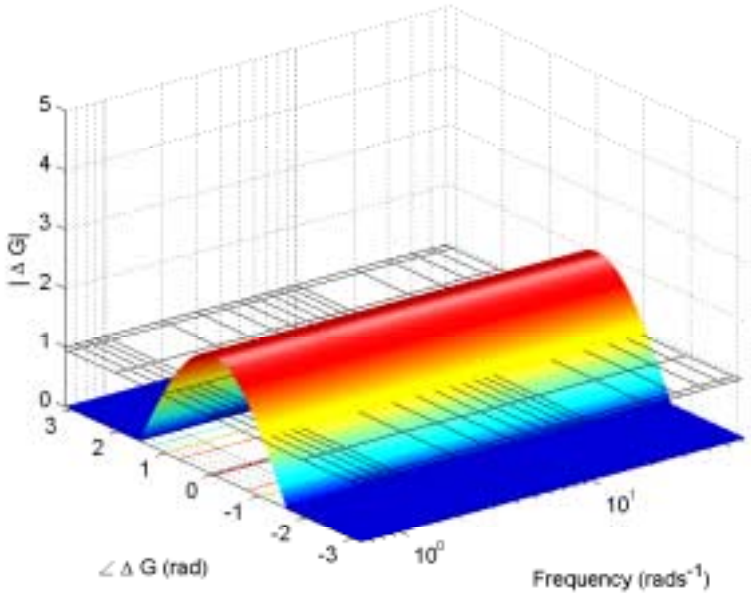


FIGURE 10.10: Robustness plot for $\Phi = G^{-1}$

shows the 'overhead' view of Figure 10.10.

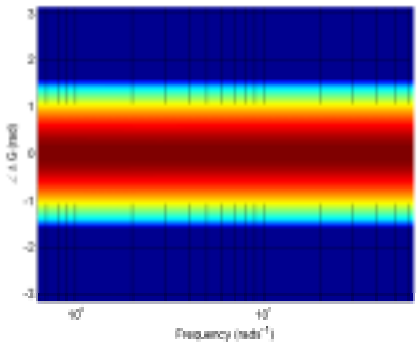


FIGURE 10.11: X-Y plane view of Figure 10.10

10.4 Alternative Algorithm Structure

The optimal algorithm cannot be represented in the previous form since its algorithm is a function of e_{k+1} instead of e_k, and its sufficient monotonic convergence criterion is consequently changed to

$$\left| \frac{1}{H(e^{jwT})G(e^{jwT}) + 1} \right| < 1 \tag{10.12}$$

where $H(e^{jwT})$ is defined in Chapter 9 and, once designed, does not depend on $G(e^{jwT})$. Adopting the notation of the last section, this dictates

$$|HG + 1| > 1 \tag{10.13}$$

so that $HG + 1$ must lie outside the unit circle centered on the origin. This is shown in Figure 10.12 with the same plant uncertainty description as previously. The value of $|\Delta G|$ that causes intersection with the stability boundary can again be calculated for each $\angle \Delta G$ over a range of frequencies.

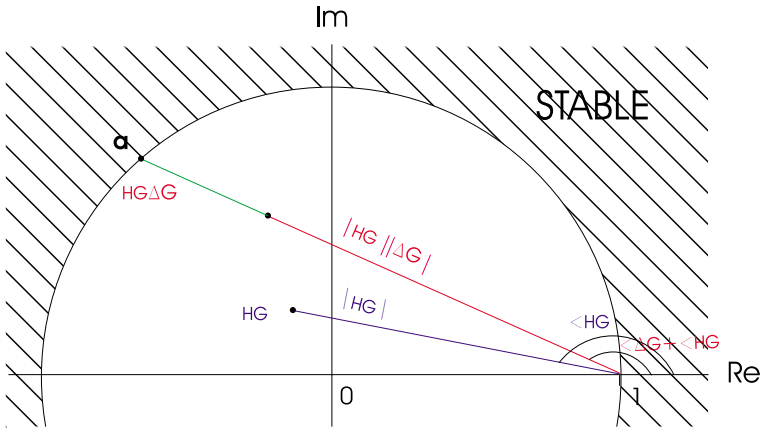


FIGURE 10.12: Graphical representation of convergence criterion

This produces

$$|\Delta G| = -\frac{2\cos(\angle HG + \angle \Delta G)}{|HG|} \tag{10.14}$$

which is the negative of Equation 10.11. The interpretation of the resulting plot is however extremely different to the previous case.

- If $|\Delta G| > \frac{2}{|HG|}$ the algorithm is always stable, and with further magnitude increase approaches an inverse and is stable for any plant uncertainty. In the previous section the inverse implied $|\Delta G| = 2$ for the case of zero phase uncertainty.
- Stability is assured for a range of $\angle \Delta G > 180°$ for all $|\Delta G|$, in the previous section the inequality was reversed.
- Values of $|\Delta G| < 1$ are now associated with stability, and $|\Delta G| > 1$ instability, a complete reversal of the previous case.

In order to reconcile the two plots, $\frac{1}{|\Delta G|}$ can be plotted instead of $|\Delta G|$. This means values of gain uncertainty below unity can still be interpreted as unstable and those above as stable. Figure 10.13 shows this plot of the optimal algorithm in this case. It clear that those almost certainly unstable regions where $|\Delta G| \approx 0$ are not present over the system bandwidth. The differences in the plot interpretation make robustness conclusions made on the basis of direct comparisons with earlier algorithms unreliable. Figure 10.14 shows the 'overhead' view of Figure 10.13.

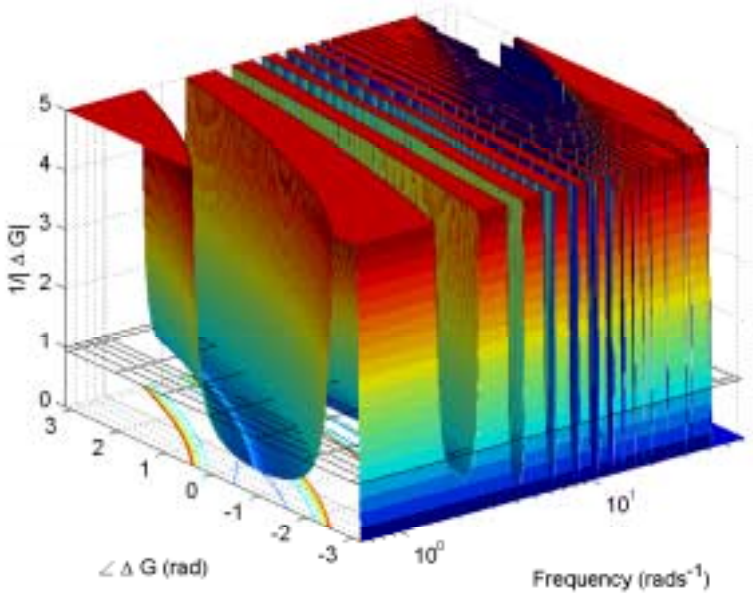


FIGURE 10.13: Robustness plot for the optimal algorithm

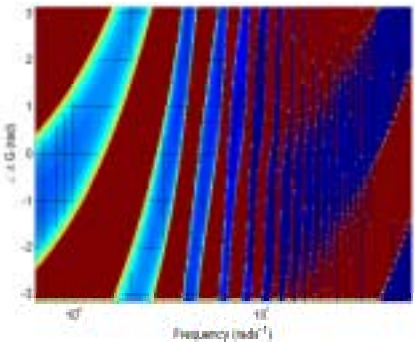


FIGURE 10.14: X-Y plane view of Figure 10.13

Assuming HG has no unstable poles, an application of the Nyquist Stability Criterion to the geometry of Figure 10.12 shows that if the point -1 is not encircled, then $1 + HG$ has no unstable zeros and so the system

$$\frac{e_{k+1}}{e_k} = \frac{1}{HG + 1} \tag{10.15}$$

is stable. This is automatically guaranteed if Equation 10.12 is satisfied.

10.5 Extended Algorithm Structure

The presence of a forgetting factor in either algorithm structure scales the circular stability boundary seen in Figures 10.1 and 10.12. In the former case it is enlarged by $\frac{1}{\gamma}$, and in the latter it is reduced by γ. The former case poses no problem as Equation 10.11 can simply be altered to

$$|\Delta G| = \frac{\cos(\angle \Phi G + \angle \Delta G) \pm \sqrt{\cos^2(\angle \Phi G + \angle \Delta G) + \left(\frac{1-\gamma}{\gamma}\right)}}{|\Phi G|} \tag{10.16}$$

and for every value of $\angle \Delta G$ there is a single $|\Delta G|$ which intersects the stability boundary. If a forgetting factor is used with the optimal algorithm, however, the criteria is changed to

$$\left| \frac{1}{HG + 1} \right| < \frac{1}{\gamma} \tag{10.17}$$

which has the graphical representation shown in Figure 10.15.

Therefore for $\arcsin(\gamma) < \angle HG + \angle \Delta G < \arcsin(\gamma)$ the circle is intersected twice and there are two values, a_1 and a_2, corresponding to the stability boundary. For values of HG within the circle, $|\Delta G|$ can be increased and reduced for stability. Therefore $|\Delta G| < 1$ can no longer be associated with instability in these cases.

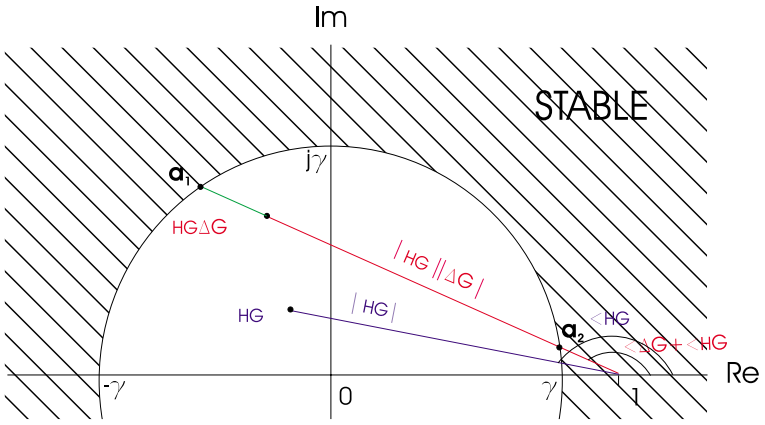


FIGURE 10.15: Graphical representation for forgetting factor

Extending this idea, the deadbeat scheme of Chapter 8 can be represented by the update

$$u_{k+1}(z) = u_k(z) + \phi R^*(z) e_k(z) \tag{10.18}$$

being applied as a feed-forward controller around the pole pole-placed plant, R. Adopting the notation of the previous section,

$$R = \frac{P}{G^{-1} + Q} \tag{10.19}$$

and P and Q are transfer fuctions that are independent of G. Since this is the same form of ILC update as those seen in Section 10.2, the stability criteria is again given by ΦR being in the unit circle centered on $+1$, where $\Phi = R^*$). However, in this case there is no simple relationship between the uncertainty of the plant G and the corresponding change in the pole-placed system R. In the previous cases the uncertainty could directly multiply $|\Phi G|$ and add to $\angle \Phi G$ which also allowed the number of intersections of the stability boundary, for a given uncertainty angle to be known. As the algorithm strucure increases in complication the analytic computation of the stability boundary mapping increases in difficulty.

10.6 Summary

A graphical method of examining the stability and robustness of ILC algorithms implemented on a known plant has been developed. Three dimensional plots have been drawn for each algorithm and the stability implications expounded. The adjoint algorithm has been seen to be the most robust but with the unavoidable compromise of lower convergence speed. The multiple lead law is less robust but learns higher frequencies more rapidly. The robustness plots of the optimal approach has a different interpretation due to its dissimilar structure compared

to other implemented algorithms. It is this structure that allows an increased robustness to uncertainty.

Chapter 11

Conclusions and Further Work

11.1 Discussion

This work has shown the failure of P-type and D-type ILC laws to control the output tracking of a non-minimum phase test facilty. In contrast, the simple phase-lead law has been shown to offer improved results. It has been demonstated how the best performing phase-lead in practice can be found theoretically and how the performance can be improved through the design and implementation of a variety of different filters and a forgetting factor.

A new multiple phase-lead algorithm has been developed and tested in order to expoit and extend the success of the phase-lead law. Its theoretical derivation has provided a framework in which algorithms can be produced which places emphasis on learning specific frequencies.

The adjoint method has been shown to share close links with both the phase-lead and multiple phase-lead laws. Practical design methods have been proposed and used for its implementation. Novel additions to this law include the use of a zero-phase filter to improve learning over certain frequency ranges and the combination of the adjoint algorithm with deadbeat control to avoid its possible truncation and increase its convergence. The formulation of automatic G^T was also developed in order to reduce the information that was necessary to apply the algorithm.

An optimal law has been implemented for the first time and found to offer excellent tracking and convergence properties.

Having implemented a variety of algoirithms, it is naturally desirous to rank them in order of performance and reach a definate conclusion as to which is best. The cycle error results of each algorithm have been compared throughout in order to guage the relative levels of convergence, final error and stability. It is difficult,

however, to compare several ILC laws over a number of criteria and give definative measures of their relative merits. The minimum level of error achieved using a given demand is, however, a solid means for comparison. Table 11.1 shows values for the stage 1 system of the final error, in terms of NE, the demands used are the 20 UPM versions and the tests have all been conducted over 400 cycles. Table 11.2 shows results for the stage 2 system. In these results, the PID controller has been tuned solely to reduce the NE and its parameters are not the same as those used in conjunction with applying ILC algorithms. A cost comprising of the

Filter Name	Sinewave demand	R1 demand	R2 demand	Cost
P-type	0.853457	1.190893	untested	1.7043
PID	0.537036	0.662516	0.930667	1.0000
D-type	0.333168	0.474994	untested	0.6737
Delay type	0.041439	0.104832	untested	0.1219
Phase-lead (unfiltered)	0.010063	0.025652	0.033925	0.0357
Phase-lead (filtered)	0.006722	0.014400	0.033495	0.0176

TABLE 11.1: Overview of minimum error values for the stage 1 plant

Filter Name	Sinewave demand	R1 demand	R2 demand	Cost
PID	0.476000	0.551000	1.014000	1.0000
Automatic G^T	0.010371	0.041431	0.094736	0.0718
Phase-lead (unfiltered)	0.009800	0.016500	0.057300	0.0410
Phase-lead (filtered)	0.006850	0.011361	0.034700	0.0259
G^T	0.005399	0.008410	0.026042	0.0195
Multi-lead	0.001585	0.003059	0.014896	0.0096
Optimal	0.000956	0.002854	0.009014	0.0064

TABLE 11.2: Overview of minimum error values a stage 2 plant

sum of the minimum errors for all the demands has been has been calculated to facuilitate a direct comparison. The resulting costs have been scaled so that the PID cost equals unity. Since the R2 demand was not used for all the stage 1 tests, it is not included in the cost function for the stage 1 plant.

It can be seen that for the stage 2 plant the minimum error achievable through use of a PID controller has been reduced by 498 times for the 20 UPM sinewave and by 193 and 122 times for the 20 UPM R1 and R2 demands respectively. Table 11.3 summarises the relative merits and drawbacks of each algorithm observed in practice.

This work has shown the simpleness with which ILC can be applied. For both the systems on which it was implemented, the phase-lead law, for example, was able to converge quickly and to a small error for most of the demands used. This was achieved with only 3 parameters to select (ϕ, λ, filter cut-off). Although it should be noted that filters have not been used with methods other than phase-lead ILC.

Filter Name	Advantages	Disadvantages
P-type	No model required 1 parameter to tune	Minimum error large
D-type	No model required 1 parameter to tune	Minimum error large
Delay type	No model required 2 parameters to tune	Minimum error quite large
Phase-lead	Can do without model 2 parameters to tune Simple filter design process effectively only adds 1 extra parameter Good convergence, minimum error	Poor transients
G^T	Good transients Fairly good convergence rate Fairly good minimum error	Plant model required
Automatic G^T	No model required Very Few parameters to tune Fairly good minimum error	Poor transients Low update frequency Slow convergence
Multi-lead	Good convergence, minimum error Quite good transients	Plant model required Calculation time large
Optimal	Excellent convergence Good minimum error Tuning difficult and there are several parameters	Plant model required Calculation time large Relaxation parameter may be required

TABLE 11.3: Comparison of algorithm attributes

The optimal algorithm also did not have the luxury of gain changing to reduce the convergence and increase the final error.

For the nominal plant case, it has been seen that all the algorithms except the G^T fail to satisfy the sufficint monotonic convergence criteria, whatever value of positive definate gain is used. The design process therefore consists of the selection of an algorithm which is stable over the system bandwidth, and the design of a filter which removes all frequencies above the bandwidth. A representation of robustness that has been formulated makes the design process a simple task as the stability and robustness of the algorithm can be ascertained from a single plot.

11.2 Conclusions

Being a practical assessment in an area lacking experimental results, this thesis has done much to reinforce ILC as a technique that can reliably produce superb peformance from demanding real systems repeating a set task. In addition, this can be achieved with the same small number of tunable parameters as the conventional PID controller.

The long-term stability of ILC laws can be assured by application of a filter and knowledge of bounds on the plant uncertainty. The filter is simply designed to remove those frequencies which would not satisfy the sufficient monotonic convergence criteria for all possible uncertainty. This can be done with reference to a suitable robustness plot.

The successful experimental control of a non-minimum phase plant offers encouraging signs that similar results can be achieved on a wide variety of plant types.

The comparison of a number of algorithms has enabled an algorithm to be chosen depending on practical considerations such as how well-modelled the plant is or the acceptable level of transients. Figure 11.1 shows a flow chart that may be used to select an ILC scheme for a general plant. It has been completed using observations made during the algorithms' implementation.

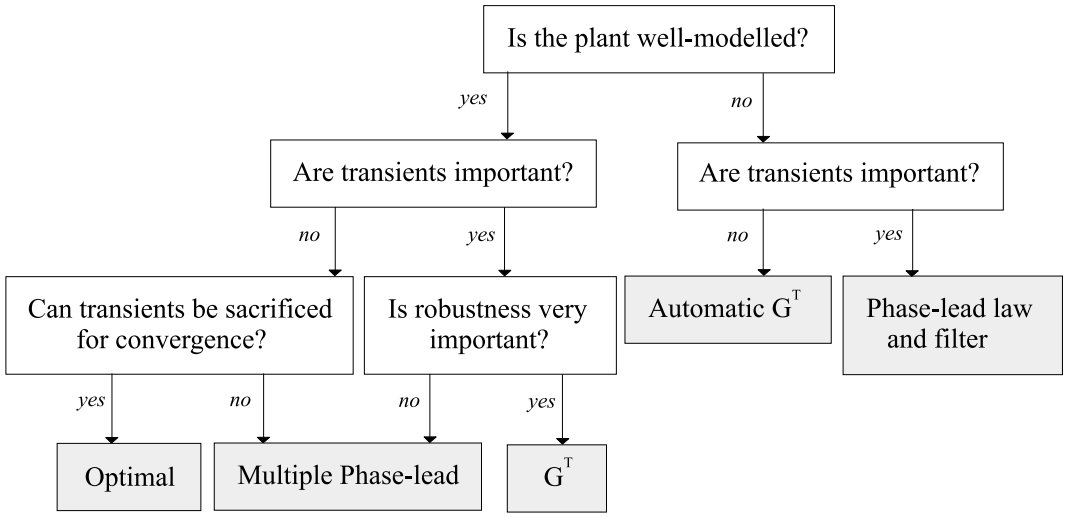


FIGURE 11.1: Design process flow chart

The close relationship between RC and ILC has been verified in practice and the same algorithms have been shown to work well in both frameworks. This suggests that switching between the two frameworks can produce similarly impressive results for each.

11.3 Further Work

The most obvious research directions are towards the continuation or direct addition to the work already completed.

It has been shown in Chapter 10 that all the algorithms considered in this thesis will eventually go unstable due to high frequency modelling error, even when using

traditional filters. FFT and inverse FFT filtering should therefore be applied to completely remove frequencies where the plant phase uncertainty is greater than 90°. This is especially important if these schemes were required to run indefinitely within an industrial environment.

The dead-beat controller and contraction mapping scheme of Chapter 8 has supplied some very encouraging results. Simulations have predicted a slighty faster convergence rate than the optimal algorithm tried, together with the good transients associated with the adjoint law. It will therefore be an ideal choice to be implemented on the system.

Schemes closely related to the contraction mapping scheme should be investigated. For example, the use of a scalar that is calculated before every cycle in order to increase its convergence (Hatonen et al., 2003a).

The effect of a disturbance on the plant which may permanently effect the plant dynamics should be considered. This would model the change of load in an industrial process for example.

Use the stage 3 plant to gain insight into the effect of further system complexity and non-linearity on the robustness of the updates seen.

Work that would naturally follow on:

The most accurate update method is of course to use the plant inverse, but this has been seen to be extremely ill-conditioned with respect to plant uncertainty. If this could be stabilised at low frequencies and then FFT filtered when the phase uncertainy $> 90°$ or at the value of the greatest frequency present in the demand (whichever is lower) then perhaps the best results possible could be achieved.

The good performance gained through use of the Kalman filter and optimal control naturally brings other optimal schemes to the forefront of those scemes to be implemented next. In particular the 'Norm Optimal' and 'Norm Predictive' methods (Amann et al., 1995, 1996a,b, 1998) will theoretically work on a non-minimum phase system and produce geometric convergence.

Perhaps the ultimate future goal would be to see a greater use made of ILC algorithms in industry as a simple way to drastically improve performance and efficiency.

Appendix A

Frictionless State Space Matrices

The following state-space systems descibe the mechanical systems shown in Chapter 3. It is to that chapter that the figure numbers refer.

A.1 Mechanical realisation of stage 1 (Figure 3.2)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K}{J} & \frac{-BG}{J} & \frac{-K}{J} & \frac{-B}{J} \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{K}{J} & \frac{-BG}{J} & \frac{-K}{J} & \frac{-B}{J} \\ -K & -BG^2 & K & -BG \end{bmatrix} \qquad d = \begin{bmatrix} -J_g \\ 0 \end{bmatrix} \tag{A.1}$$

where

$$\underline{\mathrm{u}}(t) = \begin{bmatrix} \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{in}\ddot{\theta}_i \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_i \dot{\theta}_i \theta_o \dot{\theta}_o \end{bmatrix}^T \tag{A.2}$$

Approximate theoretical values: $J = 0.001813$, $J_g = 0.001272$, $K = 0.04$, $B = 0.01$

A.2 Drive train with an 'in line' damper (Figure 3.4)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{-K}{J_2 G} & \frac{K}{J_2 G} & 0 \\ 0 & \frac{1}{G} & \frac{-K}{G^2 B} & \frac{K}{G^2 B} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 \\ \frac{1}{J_2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & K & -K & 0 \\ 0 & 0 & \frac{-K}{J_2 G} & \frac{K}{J_2 G} & 0 \end{bmatrix} \qquad d = \begin{bmatrix} 0 & -J_1 \\ \frac{1}{J_2} & 0 \end{bmatrix} \tag{A.3}$$

where

$$\underline{\mathrm{u}}(t) = \begin{bmatrix} T_{in} \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{out} \ddot{\theta}_o \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_o \dot{\theta}_o \theta_1 \theta_i \dot{\theta}_i \end{bmatrix}^T \tag{A.4}$$

A.3 Drive train with a damper 'to earth' (Figure 3.5)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{KG}{J_2} & 0 & \frac{-K}{J_2} & \frac{-BG^2}{J_2} \end{bmatrix} \qquad b = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ \frac{G^2}{J_2} & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -K & 0 & \frac{K}{G} & 0 \\ \frac{KG}{J_2} & 0 & \frac{-K}{J_2} & \frac{-BG^2}{J_2} \end{bmatrix} \qquad d = \begin{bmatrix} 0 & -J_1 \\ \frac{G^2}{J_2} & 0 \end{bmatrix} \tag{A.5}$$

where

$$\underline{\mathrm{u}}(t) = \begin{bmatrix} T_{in} \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{out} \ddot{\theta}_o \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_i \dot{\theta}_i \theta_1 \theta_o \dot{\theta}_o \end{bmatrix}^T \tag{A.6}$$

A.4 Drive train with an 'in line' damper preceding non-minimum phase section

$$A = \begin{bmatrix} \frac{K_1}{G_1^2 B_1} & 0 & \frac{1}{G_1} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{-K_1}{G_1(J_2+J_g)} & \frac{-K_2}{J_2+J_g} & \frac{-B_2 G_2^2}{J_2+J_g} & \frac{-K_2}{J_2+J_g} & \frac{-B_2 G_2}{J_2+J_g} \\ 0 & \frac{K_2}{J_3} & \frac{-B_2 G_2}{J_3} & \frac{-K_2}{J_3} & \frac{-B_2}{J_3} \end{bmatrix} \qquad b = \begin{bmatrix} \frac{K_1}{G_1^2 B_1} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{K_1}{G_1(J_2+J_g)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -K & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{K_2}{J_3} & \frac{-B_2 G_2}{J_3} & \frac{-K_2}{J_3} & \frac{-B_2}{J_3} \end{bmatrix} \qquad d = \begin{bmatrix} -K_1 & 0 & -J_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.7}$$

where

$$\underline{\mathrm{u}}(t) = \begin{bmatrix} T_{in} \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{in} \theta_o \dot{\theta}_o \ddot{\theta}_o \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_1 \theta_2 \dot{\theta}_2 \theta_o \dot{\theta}_o \end{bmatrix}^T \tag{A.8}$$

A.5 Stage two testbed schematic (Figure 3.6)

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(K_1+K_2 G_1^2)}{J_2+J_g G_1^2} & \frac{-(B_1 G_1^2+B_2 G_2^2 G_1^2)}{J_2+J_g G_1^2} & \frac{K_2 G_1^2}{J_2+J_g G_1^2} & \frac{-B_2 G_2 G_1^2}{J_2+J_g G_1^2} \\ 0 & 0 & 0 & 1 \\ \frac{K_2}{J_3} & \frac{-B_2 G_2}{J_3} & \frac{-K_2}{J_3} & \frac{-B_2}{J_3} \end{bmatrix} \quad b = \begin{bmatrix} 0 & 0 & 0 \\ \frac{K_1 G_1}{J_2+J_g G_1^2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{K_1}{G_1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{K_2}{J_3} & \frac{-B_2 G_2}{J_3} & \frac{-K_2}{J_3} & \frac{-B_2}{J_3} \end{bmatrix} \qquad d = \begin{bmatrix} -K_1 & 0 & -J_1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.9}$$

where

$$\underline{\mathrm{u}}(t) = \begin{bmatrix} \theta_i \dot{\theta}_i \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{in} \theta_o \dot{\theta}_o \ddot{\theta}_o \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_1 \dot{\theta}_1 \theta_o \dot{\theta}_o \end{bmatrix}^T \tag{A.10}$$

Approximate theoretical values: $G_1 = G_2 = 1$, $J_1 = 0.000837$, $J_g + J_2 = 0.001851$, $J_3 = 0.00167$, $K_1 = 0.2$, $K_2 = 0.04$, $B_1 = 0.01$, $B_2 = 0.01$

A.6 Stage three testbed schematic (Figure 3.7)

$$
\begin{aligned}
A &= \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\frac{K_1}{J_2} & 0 & \frac{-(K_1+K_2)}{J_2} & \frac{-B_1}{J_2} & \frac{K_2}{J_2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & \frac{K_2}{J_3+J_g} & 0 & \frac{-(K_2+K_3)}{J_3+J_g} & \frac{-(B_2+B_3G_2^2)}{J_3+J_g} & \frac{K_3}{J_3+J_g} & \frac{-B_3G_2}{J_3+J_g} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & \frac{K_3}{J_4} & \frac{-B_3G_2)}{J_4} & \frac{-K_3}{J_4} & \frac{-B_3}{J_4}
\end{bmatrix} \\
b &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} K_1 & 0 & -K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0\frac{K_3}{J_4} & \frac{-B_3G_2}{J_4} & \frac{-K_3}{J_4} & \frac{-B_3}{J_4} & \end{bmatrix} \qquad d = \begin{bmatrix} -J_1 & 0 \end{bmatrix}
\end{aligned}
\tag{A.11}
$$

where

$$
\underline{\mathrm{u}}(t) = \begin{bmatrix} \ddot{\theta}_i \end{bmatrix} \qquad \underline{\mathrm{y}}(t) = \begin{bmatrix} T_{in}\ddot{\theta}_o \end{bmatrix}^T \qquad \underline{\mathrm{x}}(t) = \begin{bmatrix} \theta_i\dot{\theta}_i\theta_1\dot{\theta}_1\theta_2\dot{\theta}_2\theta_o\dot{\theta}_o \end{bmatrix}^T \tag{A.12}
$$

Approximate theoretical values: $G_1 = G_2 = 1$, $J_1 = 0.000837$, $J_2 = 0.002575$, $J_g + J_3 = 0.001842$, $J_4 = 0.00167$, $K_1 = 0.2$, $K_2 = 0.2$, $K_3 = 0.04$, $B_1 = 0.0025$, $B_2 = 0.005$, $B_3 = 0.01$.

Appendix B

Motion Control Card Specifications

The motion control card is a two axis interface card that resides in a single ISA slot within the PC. The hardware consists of a programmable interrupt controller that is capable of interrupting the host PC at periods of 0.1ms increments. Two 32-bit counters provide an interface to the encoder feedback signal. Software allows configuration of the counters so that receipt of a marker pulse will generate an interrupt to latch the value currently in the counter. Two 12-bit DACs provide a ±10V analogue demand signal. Digital I/O on the card allows the implementation of enabling signals. The pin designations are provided in the following table.

	Encoder		Output	
Pin No.	Signal	Function	Signal	Function
1	Ai	A phase input	+15V	+15V supply
2	Bi	B phase input	0V	0V common
3		Reserved	-15V	-15V supply
4	Mi	Marker input	enC1	Motor enable 1(coll)
5	Prb	Probe input	enE1	Motor enable 1(emit)
6	nAi	Inverse A input	An1	Analogue output 1
7	nBi	Inverse B input	An2	Analogue output 2
8		Reserved	enC2	Motor enable 2 (coll)
9	nMi	Inverse marker i/p	enE2	Motor enable 2 (emit)
10	nPrb	Inverse probe i/p		
11	+12V	+12V supply		
12	+5V	+5V supply		
13	0V	0V common		
14	-5V	-5V supply		
15	-12V	-12V supply		

TABLE B.1: Interface card connections

The software consists of a TSR function that is loaded at boot up. A C function, *open_motion*, opens the card and provides access from within a C program. The card is closed by use of function *close_motion*.

Communication with the card at run time is through two functions:

long *read_motion* (long *command*, long *channel*)

and

void *write_motion* (long *command*, long *channel*)

Channel is an integer that selects the channel upon which the command is to be performed. *Write_motion* adjusts the card as required, whereas *read_motion* returns a long representing the desired parameter. *Command* is one of a number of defined strings that allow access to the various parts of the card such as the interrupt controller, the encoder counters and the DACs. Using this structure the command to read the 32-bit value on the encoder counter of axis 0 is:

count = read_motion(AXIS_32,0L)

and to write 2000mV to the DAC for axis 1 is:

write_motion(DAC_MV,1L,2000L)

Appendix C

CAD Designs

The following drawings were required in order to realise the experimental test facility designs shown in Chapter 3.

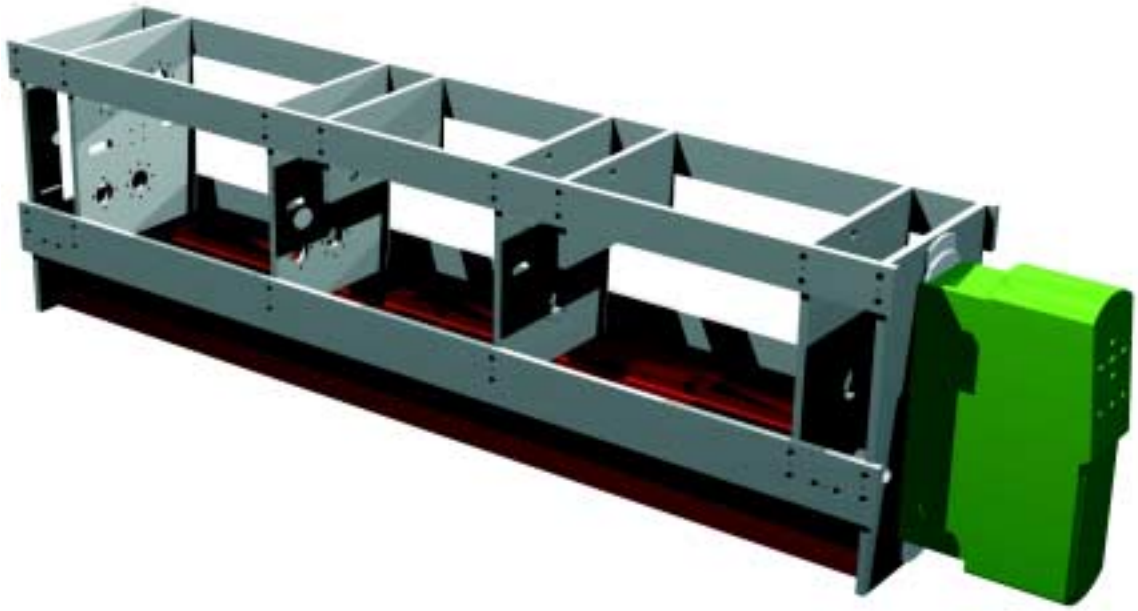


Figure C.1: CAD design of principle test-bed elements

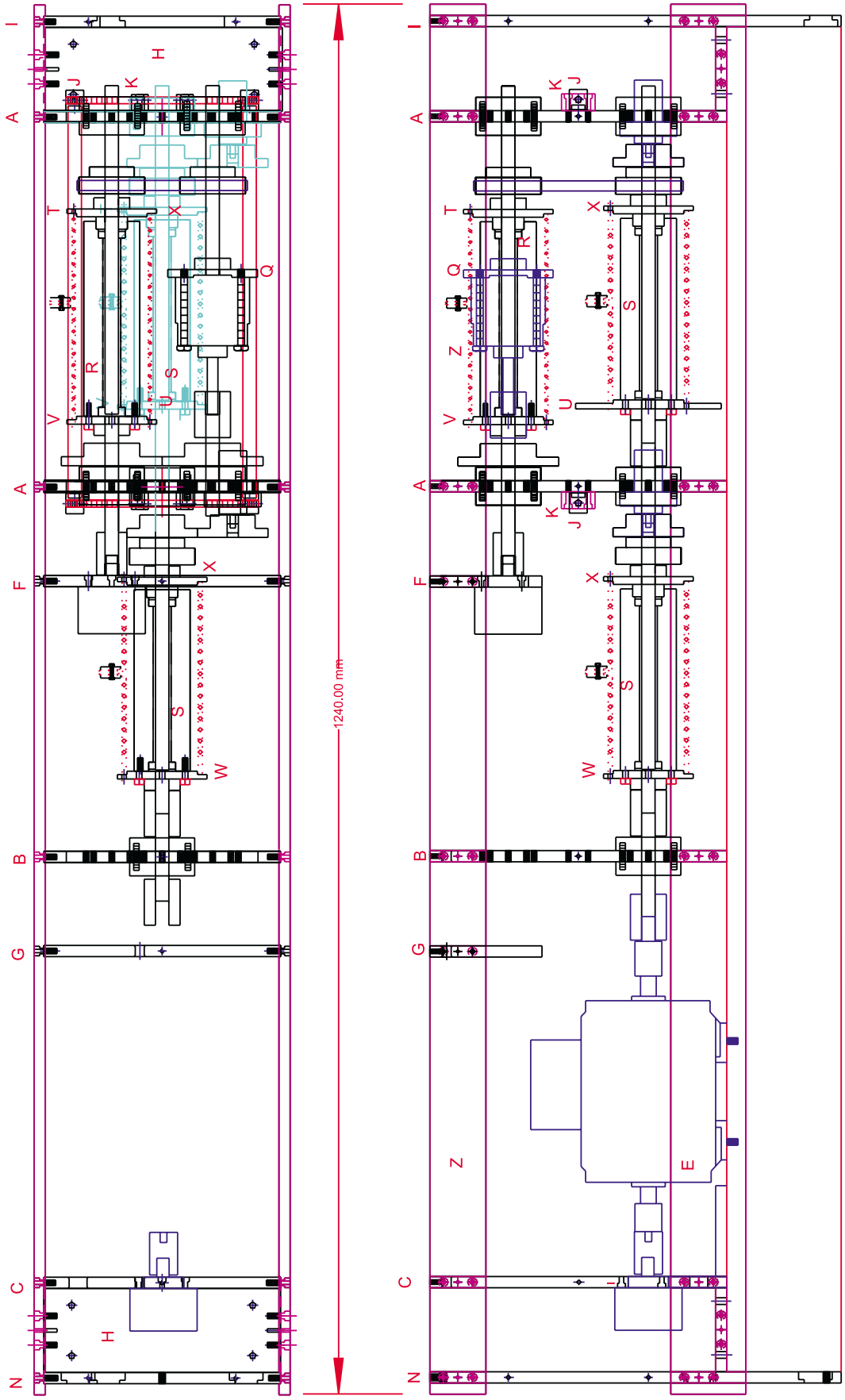


FIGURE C.2: CAD designs showing top and side views of the experimental test-bed

Appendix D

Model Parameters

The following parameters have been used within the time-based plant models.

D.1 General Parameters

INDUCTION MOTOR			
Supply Voltage	V_s	=	415 V
Supply frequency	f	=	50 Hz
Stator resistance	R_s	=	30.6 Ω no-load (28.5 cold)
Rotor resistance	R_r	=	27.756 Ω
Stator self inductance	L_{sl}	=	0.135 H
Rotor self inductance	L_{rl}	=	0.135 H
Mutual inductance	L_m	=	0.745 H
	L_s	=	$L_{sl} + L_m$
	L_r	=	$L_{rl} + L_m$
	M_{sr}	=	L_m
Number of Poles	P	=	2
Inertia	J_m	=	0.0007 kgm^2
Viscous friction	B_m	=	0.0012 $Nsrad^{-1}$
Coloumb friction	C_m	=	0.07 N
INVERTER			
Deadzone		=	0.028 s
Resistance		=	2.1 Ω
Slewrate		=	0.0504 Vs^{-1}
Speed Integral		=	59
Filter pole		=	130
I/O CARD PARAMETERS			
Maximum DAC Voltage		=	10 V
DAC Resolution		=	14-bit

STAGE 1

$J_1 = 0.00018$ Kgm^2
$J_2 = 0.0007$ Kgm^2
$J_3 = 0.0007$ Kgm^2
$J_4 = 0.0026$ Kgm^2

$b_1 = 0.00015$ Nsrad^{-1}
$b_2 = 0.00015$ Nsrad^{-1}
$b_3 = 0.00015$ Nsrad^{-1}
$b_4 = 0.00015$ Nsrad^{-1}

$J_b = 0.000069$ Kgm^2
$k_b = 80000$ Nrad$^-1$
$r_b = 0.0004$ m
$b_b = 0.04$ Nsrad^{-1}
$f_b = 0.002$ N

$f_1 = 0.025$ N
$f_2 = 0.013$ N
$f_3 = 0.013$ N
$f_4 = 0.04$ N

K $= 0.1157$ Nrad$^-1$
B(x) $= -5.775 \times 10^{-7}x^3 - 1.4 \times 10^{-6}x^2 + 0.0275x$ Nsrad^{-1}

STAGE 2

$J_1 = 0.00018$ Kgm^2
$J_2 = 0.0014$ Kgm^2
$J_3 = 0.0007$ Kgm^2
$J_4 = 0.0007$ Kgm^2
$J_5 = 0.0026$ Kgm^2

$J_b = 0.000069$ Kgm^2
$k_b = 80000$ Nrad$^-1$
$r_b = 0.029$ m
$b_b = 0.0004$ Nsrad^{-1}
$f_b = 0.05$ N

$b_1 = 0.0015$ Nsrad^{-1}
$b_2 = 0.004$ Nsrad^{-1}
$b_3 = 0.0015$ Nsrad^{-1}
$b_4 = 0.0015$ Nsrad^{-1}
$b_5 = 0.0057$ Nsrad^{-1}

$f_1 = 0.025$ N
$f_2 = 0.02$ N
$f_3 = 0.013$ N
$f_4 = 0.013$ N
$f_5 = 0.04$ N

$K_1 = 0.6$ Nrad$^-1$
$B_1(x) = -2.4 \times 10^{-7}x^3 + 1.2 \times 10^{-5}x^2 + 8.7 \times 10^{-3}x - 0.081$ Nsrad^{-1}
$K_2 = 0.1157$ Nrad$^-1$
B_2(x) $= -5.61 \times 10^{-7}x^3 - 1.36 \times 10^{-6}x^2 + 0.02499x$ Nsrad^{-1}

Appendix E

Simulation Results

The results shown were used to verify the accuracy of the system models.

E.1 Stage 1, Time Based model, Output shaft

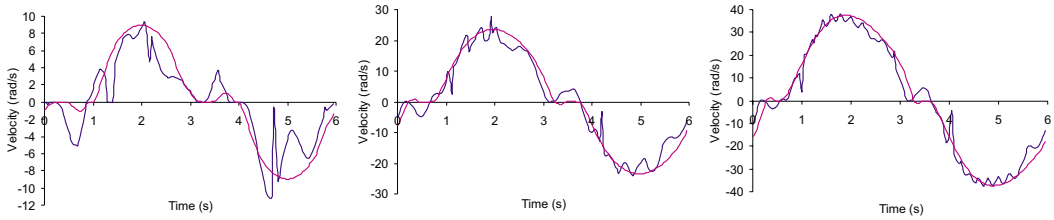


FIGURE E.1: 10 UPM sinewave demands, 1, 2 and 3V amplitudes

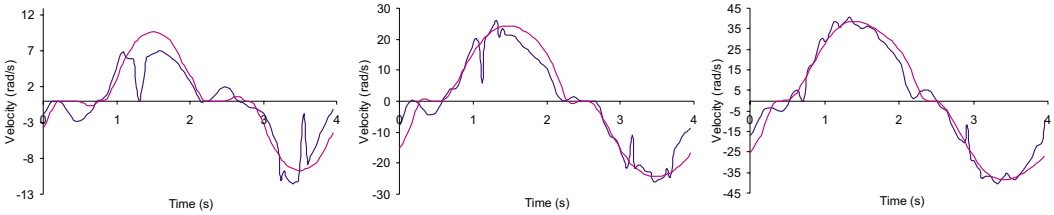


FIGURE E.2: 15 UPM sinewave demands, 1, 2 and 3V amplitudes

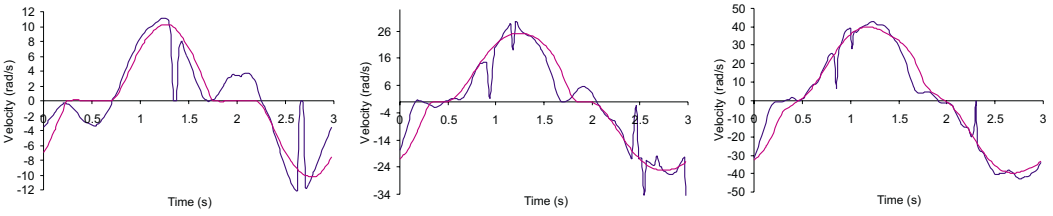


FIGURE E.3: 20 UPM sinewave demands, 1, 2 and 3V amplitudes

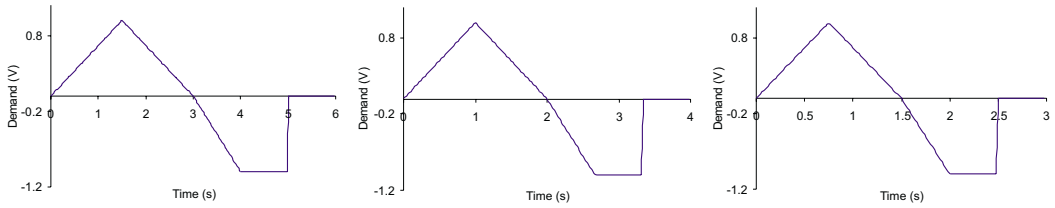


FIGURE E.4: Repeating sequence demands of 10, 15 and 20 UPM for the 1V case

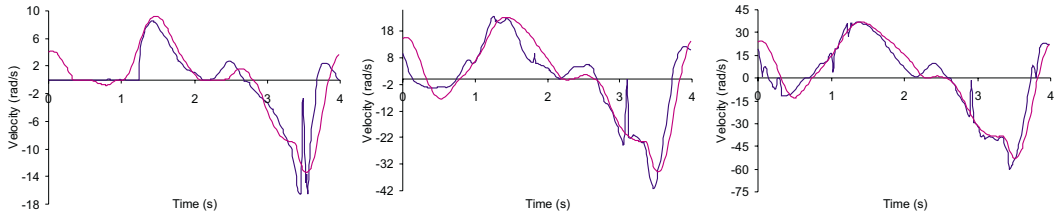


FIGURE E.5: 15 UPM repeating sequence demands, 1, 2 and 3V amplitudes

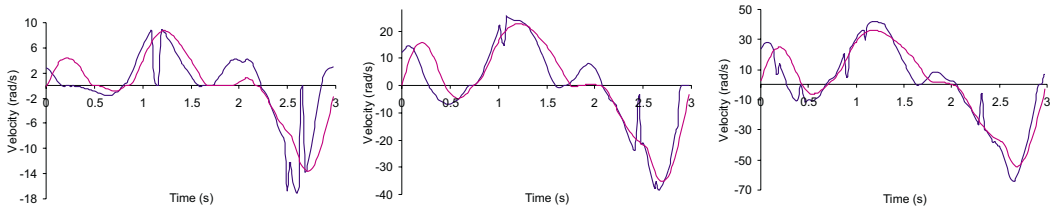


FIGURE E.6: 20 UPM repeating sequence demands, 1, 2 and 3V amplitudes

E.2 Stage 1, Frequency model, Output shaft

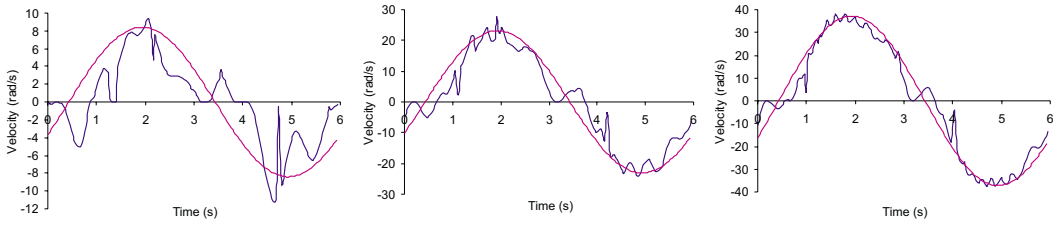


FIGURE E.7: 10 UPM sinewave demands, 1, 2 and 3V amplitudes

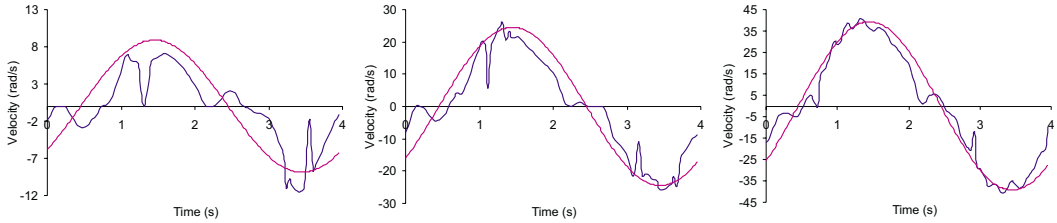


FIGURE E.8: 15 UPM sinewave demands, 1, 2 and 3V amplitudes

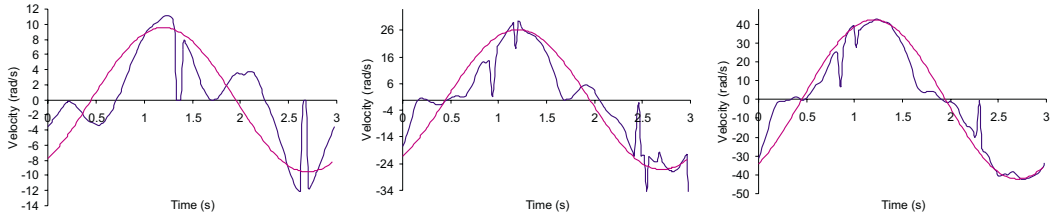


FIGURE E.9: 20 UPM sinewave demands, 1, 2 and 3V amplitudes

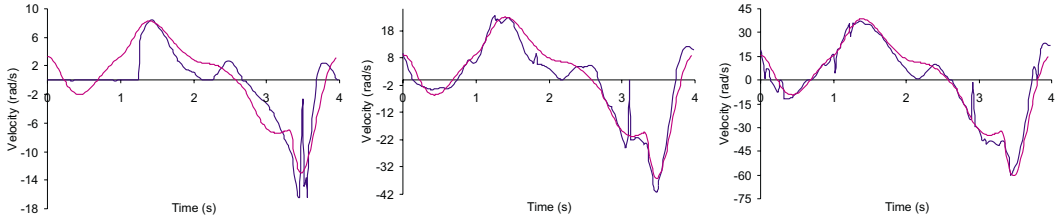


FIGURE E.10: 15 UPM repeating sequence demands, 1, 2 and 3V amplitudes

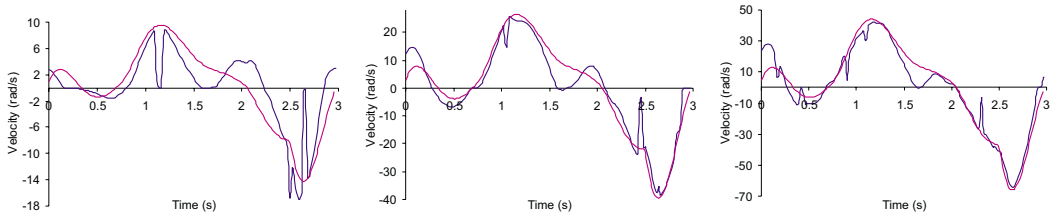


FIGURE E.11: 20 UPM repeating sequence demands, 1, 2 and 3V amplitudes

E.3 Stage 2, Time based model, Output shaft

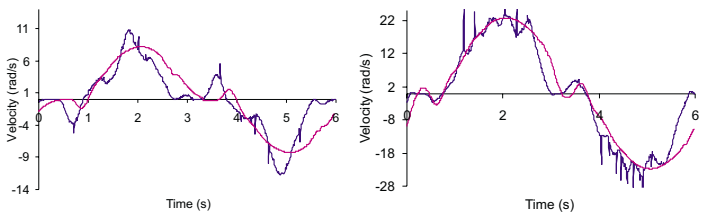


FIGURE E.12: 10 UPM sinewave demands, 1 and 2V amplitudes

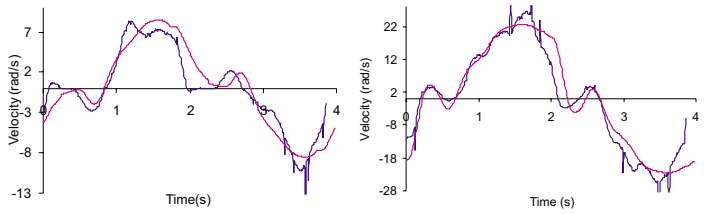


FIGURE E.13: 15 UPM sinewave demands, 1 and 2V amplitudes

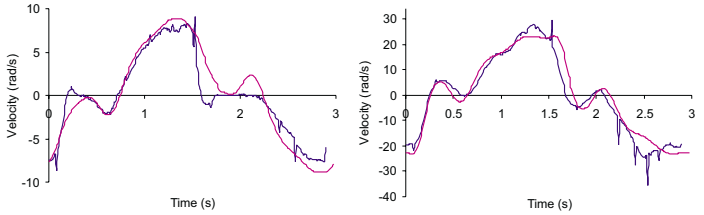


FIGURE E.14: 20 UPM sinewave demands, 1 and 2V amplitudes

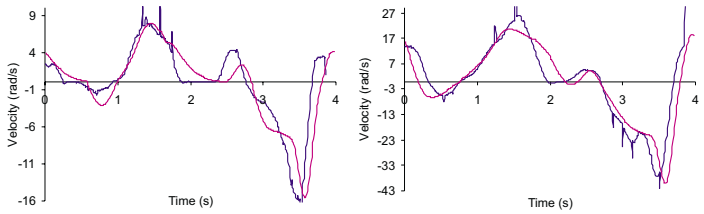


FIGURE E.15: 15 UPM repeating sequence demands, 1 and 2V amplitudes

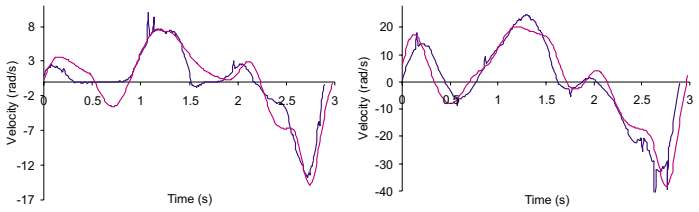


FIGURE E.16: 20 UPM repeating sequence demands, 1 and 2V amplitudes

E.4 Stage 2, Frequency model, Output shaft

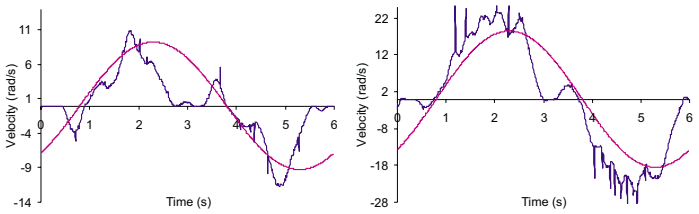


FIGURE E.17: 10 UPM sinewave demands, 1 and 2V amplitudes

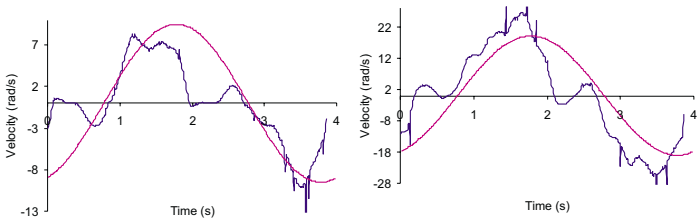


FIGURE E.18: 15 UPM sinewave demands, 1 and 2V amplitudes

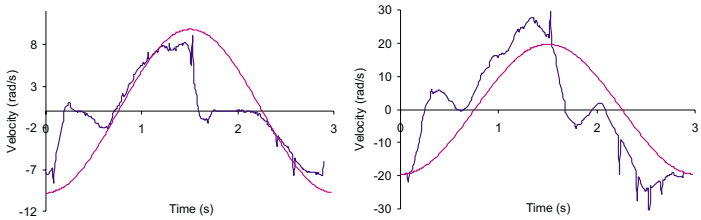


FIGURE E.19: 20 UPM sinewave demands, 1 and 2V amplitudes

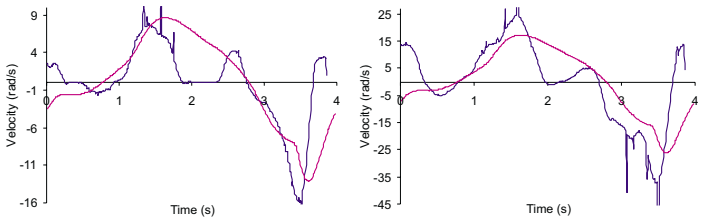


FIGURE E.20: 15 UPM repeating sequence demands, 1 and 2V amplitudes

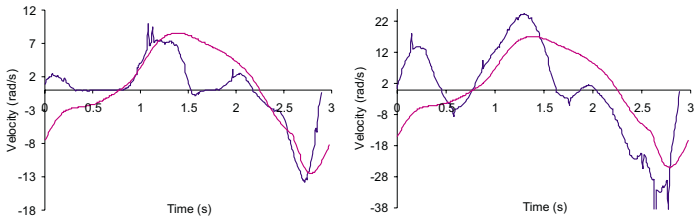


FIGURE E.21: 20 UPM repeating sequence demands, 1 and 2V amplitudes

Appendix F

Frequency Models

The following linear plant models were reached by Bode plot fitting and the fitting of experimental results.

F.1 Stage 1

1V Step and Sinewave: $G_{stage1}(s) = \frac{50.791\times10^4(4-s)}{s(s^2+5.78s+29.7)(s+45)(s+190)}$

2V Step and Sinewave: $G_{stage1}(s) = \frac{69.838\times10^4(4-s)}{s(s^2+5.78s+29.7)(s+45)(s+190)}$

3V Step and Sinewave: $G_{stage1}(s) = \frac{74.917\times10^4(4-s)}{s(s^2+5.78s+29.7)(s+45)(s+190)}$

4V Step and Sinewave: $G_{stage1}(s) = \frac{79.043\times10^4(4-s)}{s(s^2+5.78s+29.7)(s+45)(s+190)}$

1V Repeating Sequence: $G_{stage1}(s) = \frac{88.142\times10^4(2.5-s)}{s(s^2+5.2s+42.25)(s+45)(s+190)}$

2V Repeating Sequence: $G_{stage1}(s) = \frac{122.821\times10^4(2.5-s)}{s(s^2+5.2s+42.25)(s+45)(s+190)}$

3V Repeating Sequence: $G_{stage1}(s) = \frac{137.270\times10^4(2.5-s)}{s(s^2+5.2s+42.25)(s+45)(s+190)}$

F.2 Stage 2

Step and Reference: $G_{stage2}(s) = e^{-0.06s}\frac{1202(4-s)}{s(s+9)(s^2+12s+56.25)}$

1V Reference: $G_{stage2}(s) = e^{-0.06s}\frac{781.3(4-s)}{s(s+9)(s^2+12s+56.25)}$

2V Reference: $G_{stage2}(s) = e^{-0.06s}\frac{991.65(4-s)}{s(s+9)(s^2+12s+56.25)}$

3V Reference: $G_{stage2}(s) = e^{-0.06s}\frac{1141.9(4-s)}{s(s+9)(s^2+12s+56.25)}$

Fitted Bode Plot: $G_{stage2}(s) = e^{-0.01s}\frac{756.16(2.9-s)(s^2+1.184s+0.64)}{s(s+5.2)(s^2+9.3+86.49)(s^2+0.474s+0.6241)}$

$G_{stage2}(s) = e^{-0.055s}\frac{146\times10^4(4.5-s)}{s(s+5)(s^2+7s+100)(s^2+27s+2025)}$

$G_{stage2}(s) = e^{-0.09s}\frac{933.33(6-s)}{s(s+7)(s^2+10s+100)}$

$G_{stage2}(s) = e^{-0.06s}\frac{137\times10^4(6.5-s)}{s(s+4.5)(s^2+7.6s+90.25)(s^2+30s+2500)}$

Appendix G

Controller Gains

The following PID parameters were reached as a result of the tuning method described in Chapter 4.

G.1 Stage 1

	10 UPM			20 UPM			30 UPM		
Demand	K_p	K_i	K_d	K_p	K_i	K_d	K_p	K_i	K_d
1V Sinewave	97	18	0	125 97	15 50	0 0	148	15	3.5
2V Sinewave	92	11	0	110 97	0 18	0 0	115	1.5	4
3V Sinewave	97	7.5	0	110 97	0 16	0 0	113	4	0
1V Repeating Sequence	97	37	1	146 97	2.5 72	0 0	145	17	2.5
2V Repeating Sequence	97	27	7.5	120 97	2.5 28	0 0	125	5	0
3V Repeating Sequence	97	15	2.5	110 97	1 18	0 0	115	5	0

Table G.1: PID Gains

G.2 Stage 2

values in mV/count

	10 UPM			20 UPM			30 UPM		
Demand	K_p	K_i	K_d	K_p	K_i	K_d	K_p	K_i	K_d
1V Sinewave	125	0	0	123	0	2.5	138	0	0
1V Repeating Sequence 1	128	3	0	137	5	0	153	2	2
1V Repeating Sequence 2	137	5	3	145	4	5	147	10	0

TABLE G.2: PID Gains

Appendix H

Basic ILC Algorithm Results

The following experimental results use the P-type, D-type and Delay-type ILC algorithms which have been discussed in Chapter 5.

H.1 P-type Results

H.1.1 Stage 1

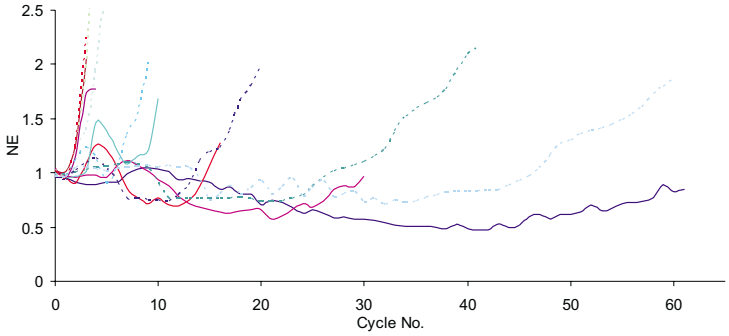


FIGURE H.1: Error results for 10 UPM R1 demand

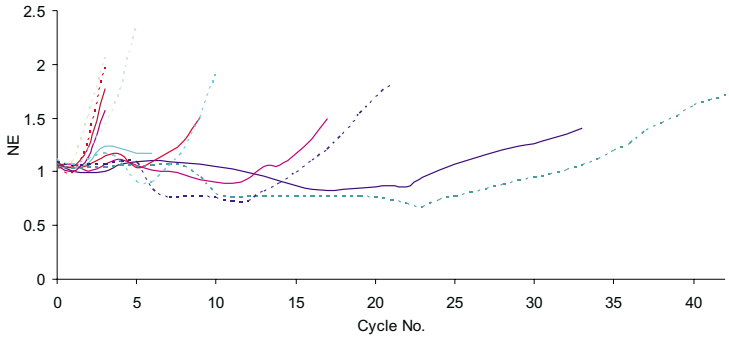


FIGURE H.2: Error results for 15 UPM R1 demand

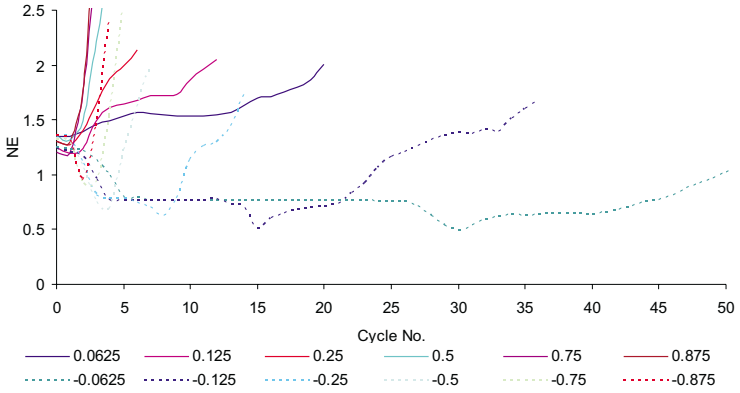


FIGURE H.3: Error results for 20 UPM R1 demand

H.2 D-type Results

H.2.1 Stage 1

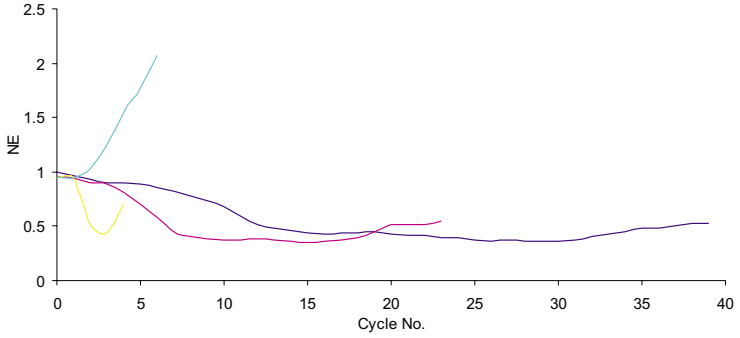


FIGURE H.4: D-type error results for 10 UPM R1 demand

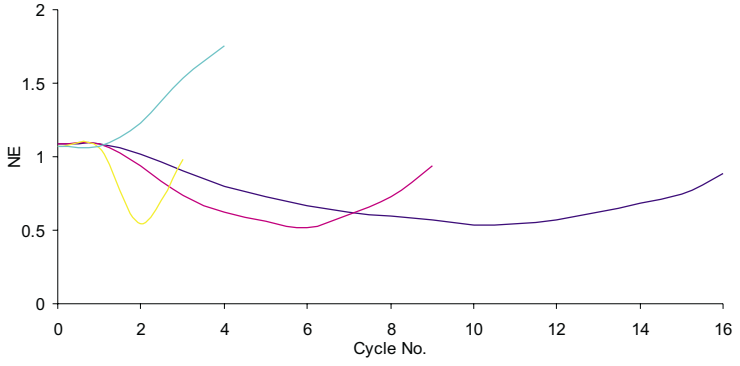


FIGURE H.5: D-type error results for 15 UPM R1 demand

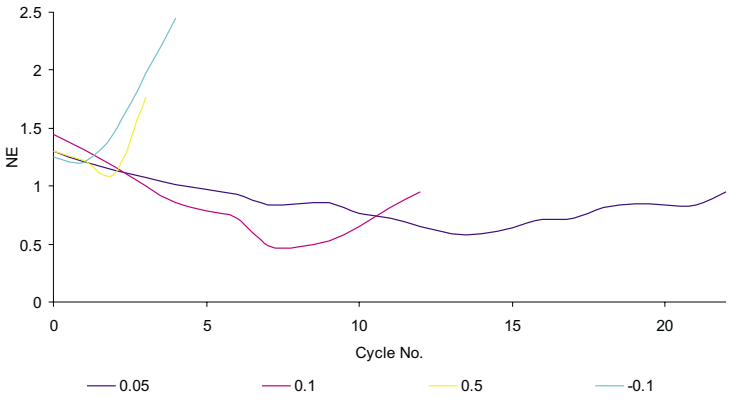


FIGURE H.6: D-type error results for 20 UPM R1 demand

H.3 Delay-type Results

H.3.1 Stage 1 R1 Demand

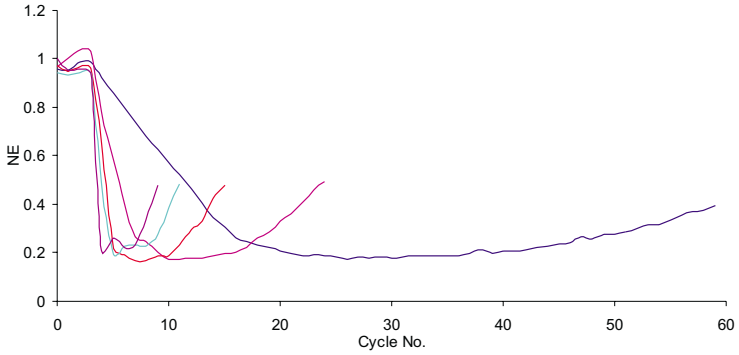


FIGURE H.7: Delay-type error results for 10 UPM R1 demand

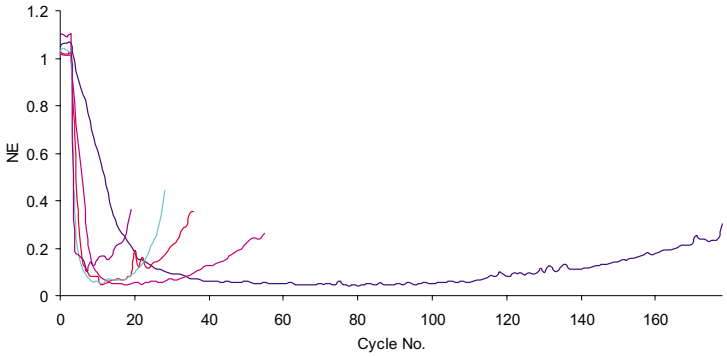


FIGURE H.8: Delay-type error results for 15 UPM R1 demand

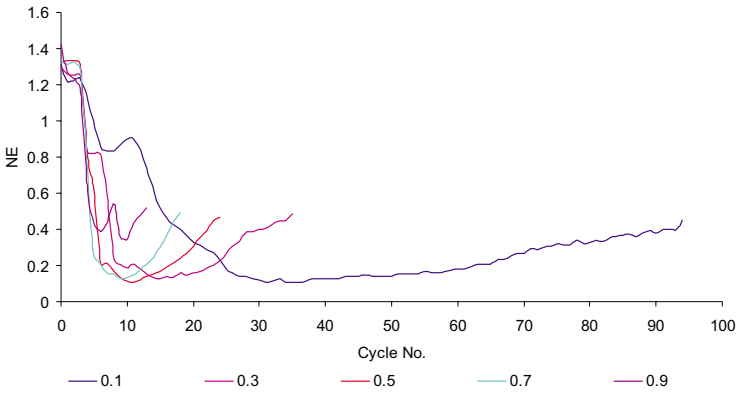


FIGURE H.9: Delay-type error results for 20 UPM R1 demand

Appendix I

Phase-lead Results

Experimental results obtained using the phase-lead algorithm are contained in this section. The results refer to both the stage 1 and stage 2 plants in conjunction with stabilising methods described in Chapter 6. The results are laid out as follows:

I.1 Stage 1 Unfiltered Phase-lead Results

I.1.1 15 UPM Sinewave Demand

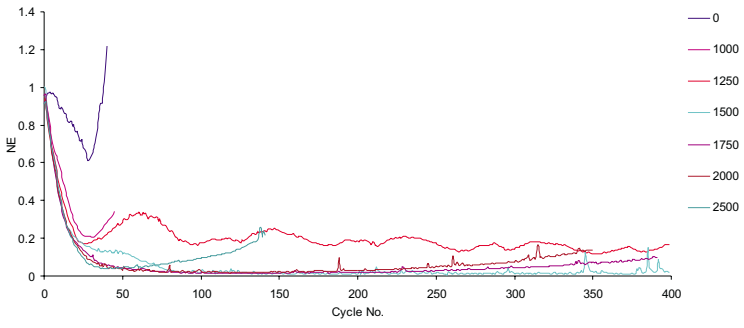


FIGURE I.1: Phase-lead error results with variable λ and $\phi = 0.1$

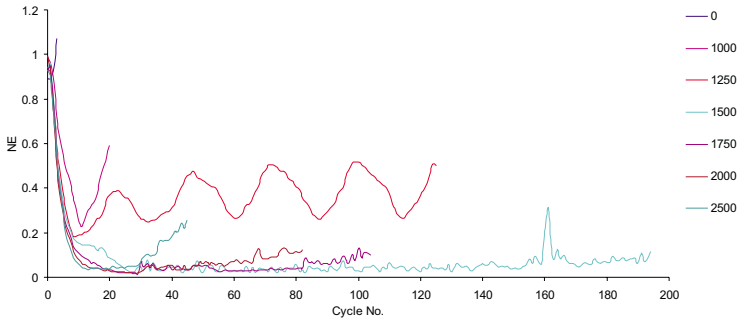


FIGURE I.2: Phase-lead error results with variable λ and $\phi = 0.3$

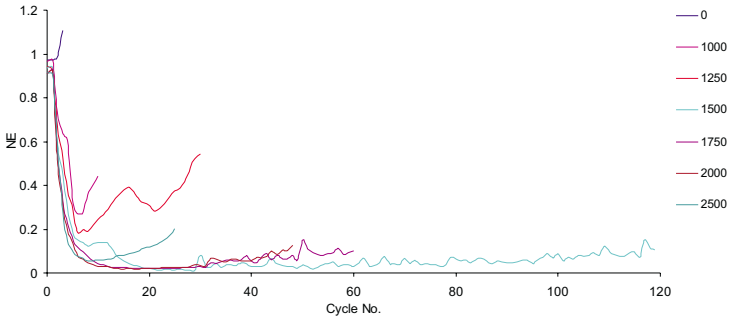


FIGURE I.3: Phase-lead error results with variable λ and $\phi = 0.5$

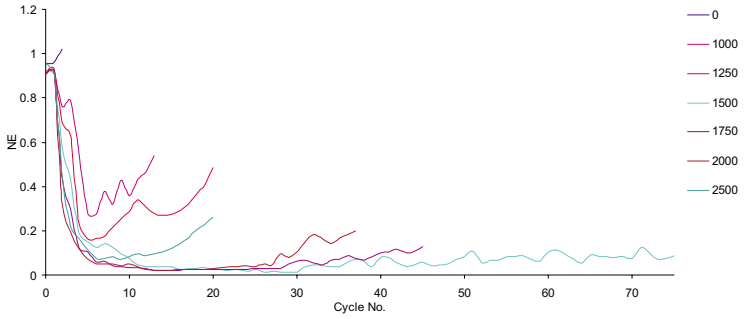


FIGURE I.4: Phase-lead error results with variable λ and $\phi = 0.7$

I.1.2 20 UPM Sinewave Demand

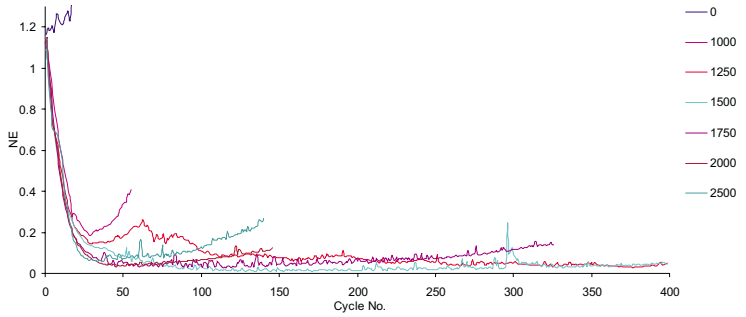


FIGURE I.5: Phase-lead error results with variable λ and $\phi = 0.1$

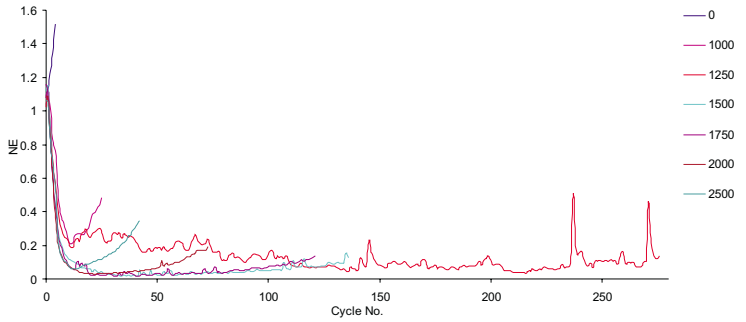


FIGURE I.6: Phase-lead error results with variable λ and $\phi = 0.3$

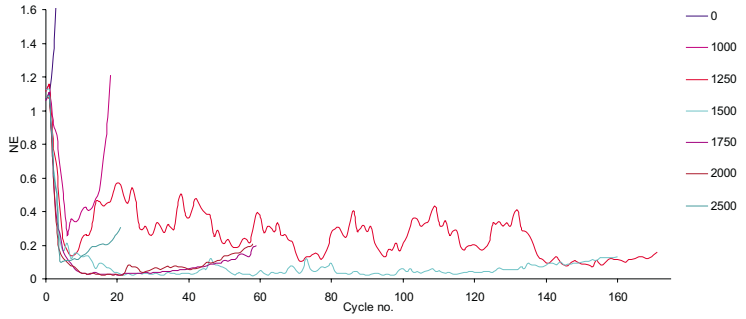


FIGURE I.7: Phase-lead error results with variable λ and $\phi = 0.5$

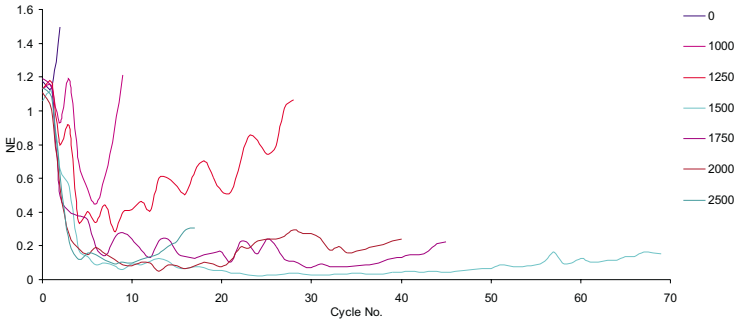


FIGURE I.8: Phase-lead error results with variable λ and $\phi = 0.7$

I.1.3 10 UPM R1 Demand

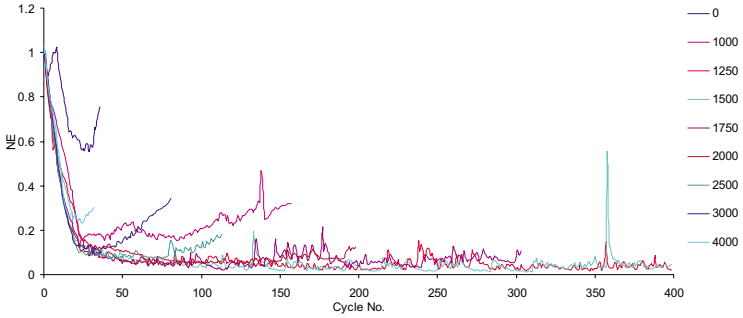


FIGURE I.9: Phase-lead error results with variable λ and $\phi = 0.1$

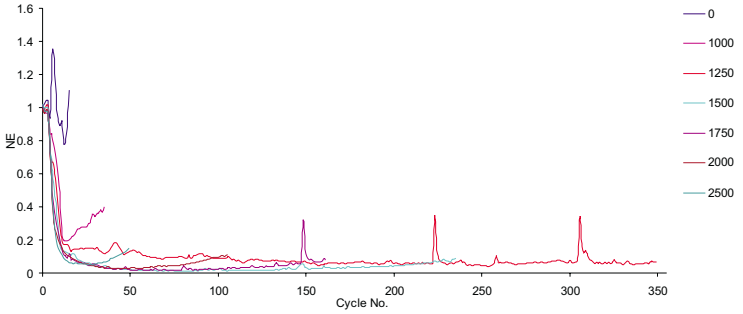


FIGURE I.10: Phase-lead error results with variable λ and $\phi = 0.3$

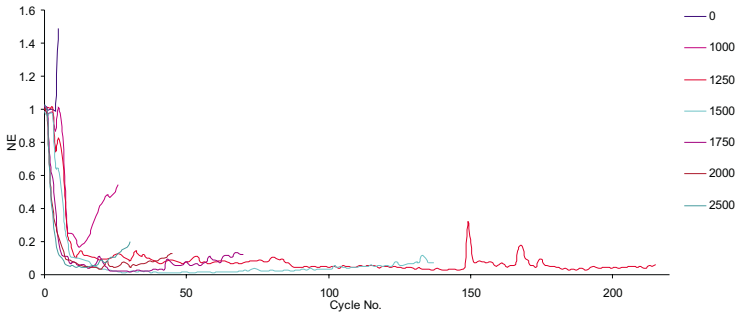


FIGURE I.11: Phase-lead error results with variable λ and $\phi = 0.5$

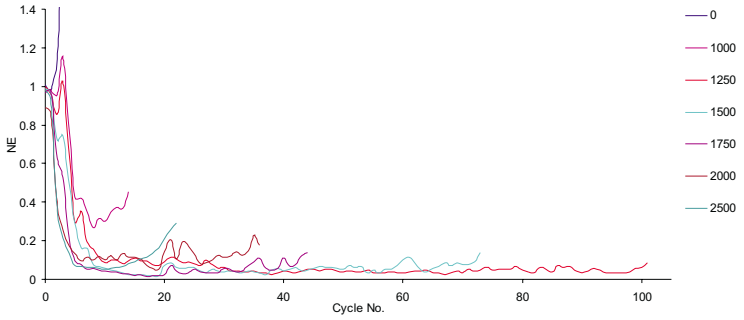


FIGURE I.12: Phase-lead error results with variable λ and $\phi = 0.7$

I.1.4 15 UPM R1 Demand

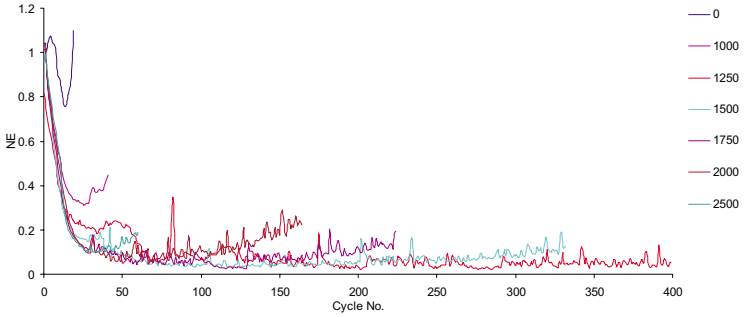


FIGURE I.13: Phase-lead error results with variable λ and $\phi = 0.1$

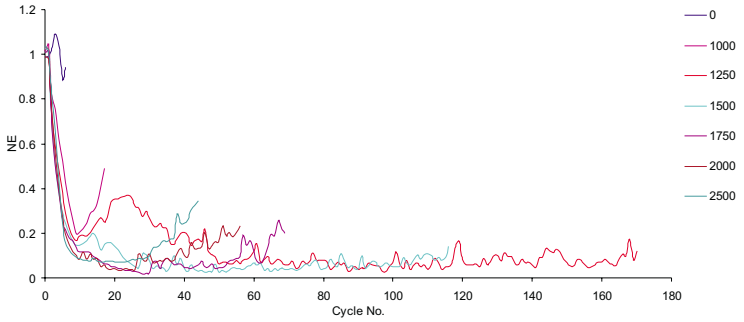


FIGURE I.14: Phase-lead error results with variable λ and $\phi = 0.3$

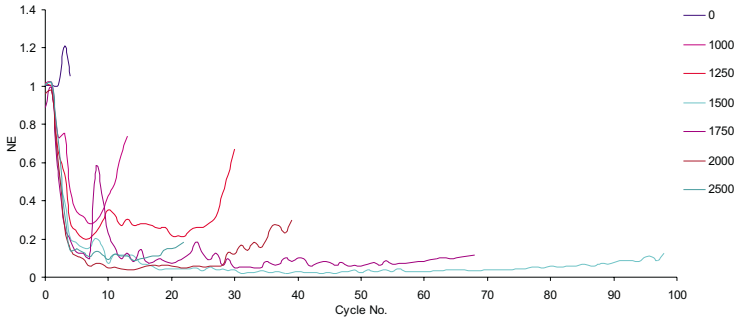


FIGURE I.15: Phase-lead error results with variable λ and $\phi = 0.5$

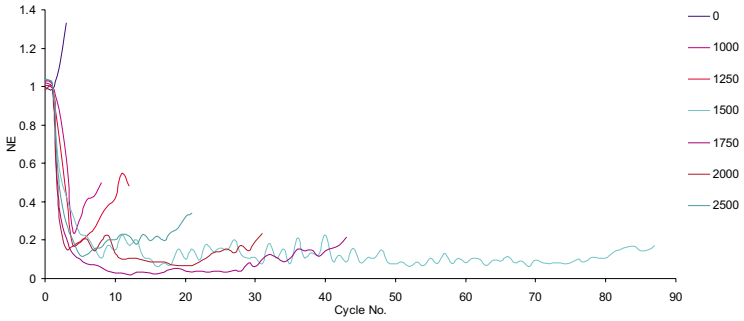


FIGURE I.16: Phase-lead error results with variable λ and $\phi = 0.7$

I.1.5 20 UPM R1 Demand

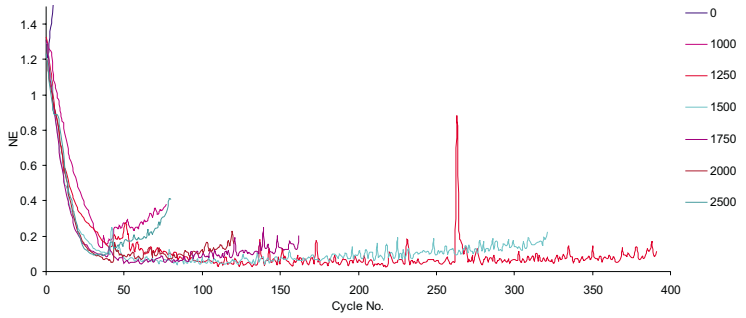


FIGURE I.17: Phase-lead error results with variable λ and $\phi = 0.1$

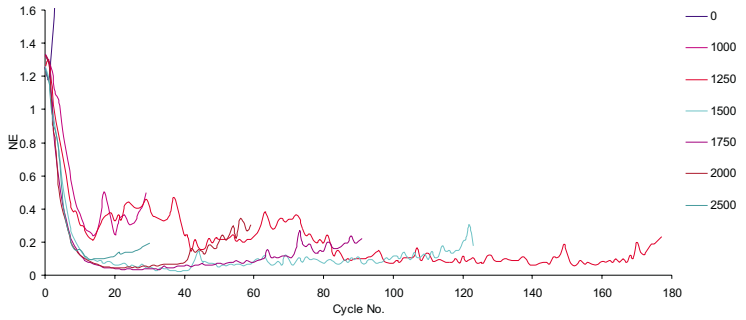


FIGURE I.18: Phase-lead error results with variable λ and $\phi = 0.3$

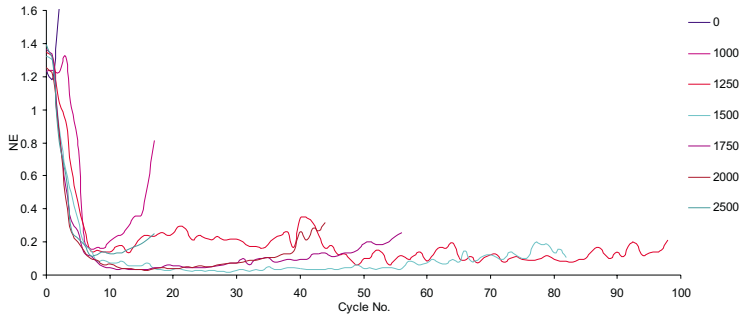


FIGURE I.19: Phase-lead error results with variable λ and $\phi = 0.5$

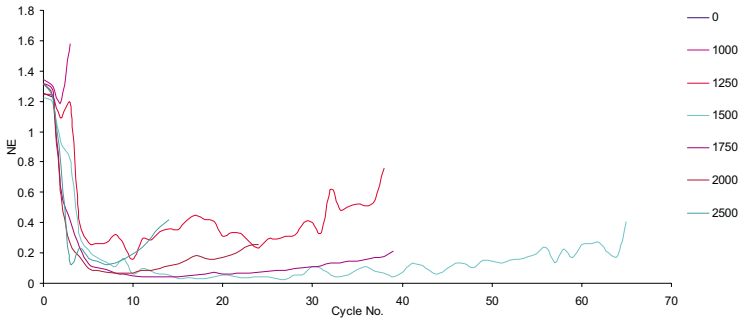


FIGURE I.20: Phase-lead error results with variable λ and $\phi = 0.7$

I.1.6 10 UPM R2 Demand

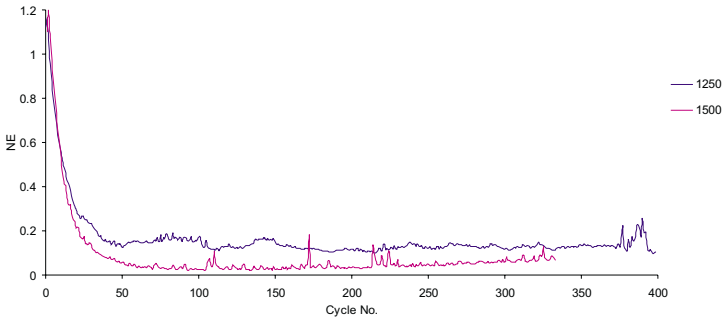


FIGURE I.21: Phase-lead error results with variable λ and $\phi = 0.1$

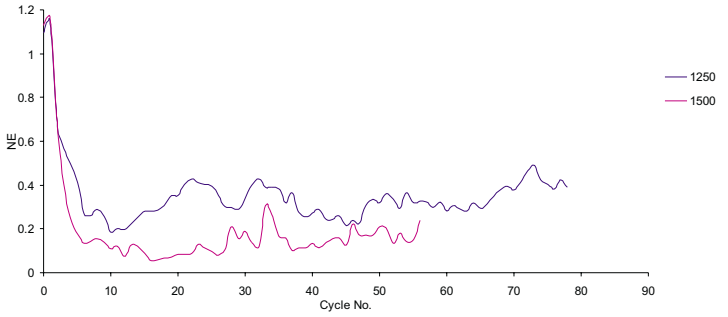


FIGURE I.22: Phase-lead error results with variable λ and $\phi = 0.5$

I.1.7 20 UPM R2 Demand

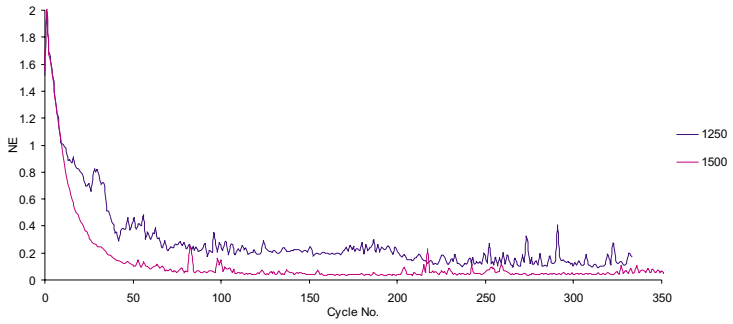


FIGURE I.23: Phase-lead error results with variable λ and $\phi = 0.1$

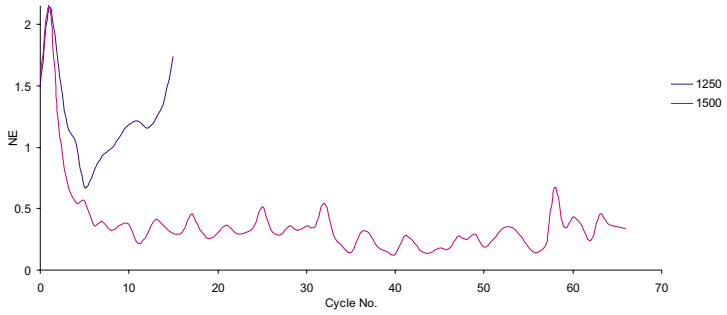


FIGURE I.24: Phase-lead error results with variable λ and $\phi = 0.5$

I.2 Stage 1 Phase-lead Results using Causal Filters

I.2.1 Sinewave Demand

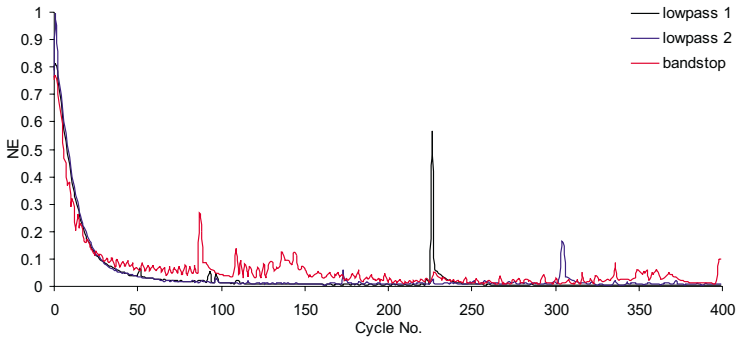


FIGURE I.25: Phase-lead error results for 10 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

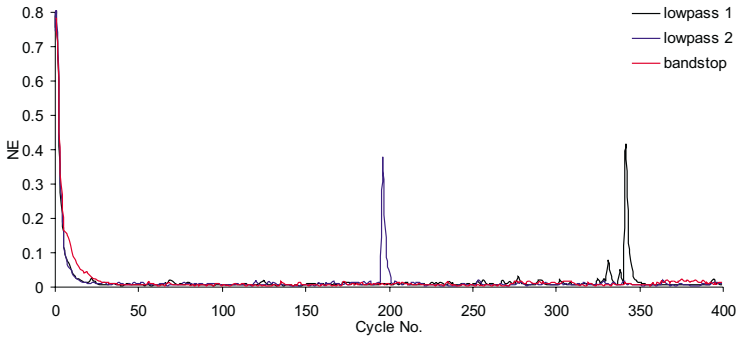


FIGURE I.26: Phase-lead error results for 10 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

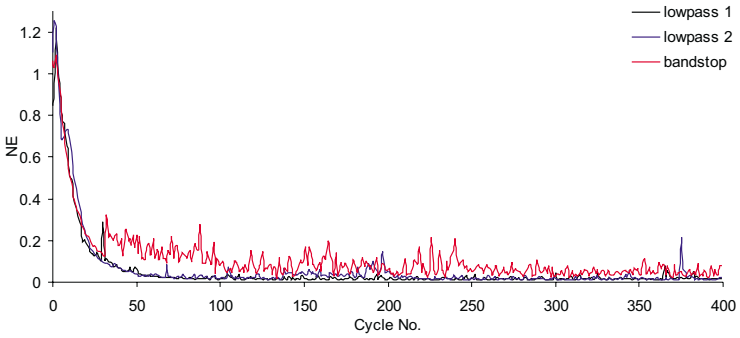


FIGURE I.27: Phase-lead error results for 20 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

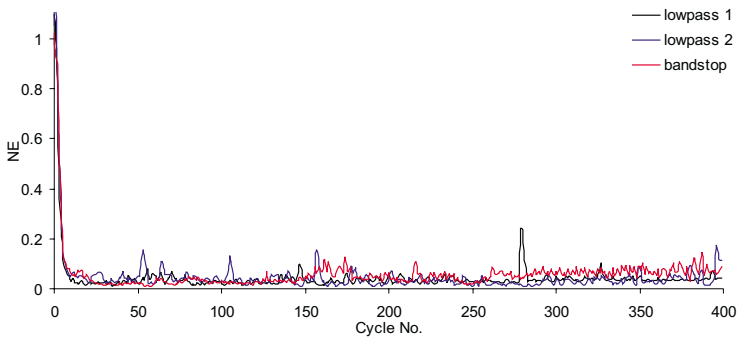


FIGURE I.28: Phase-lead error results for 20 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

I.2.2 R1 Demand

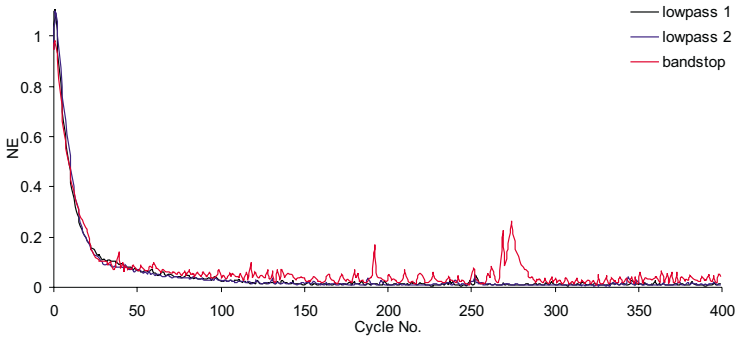


FIGURE I.29: Phase-lead error results for 10 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

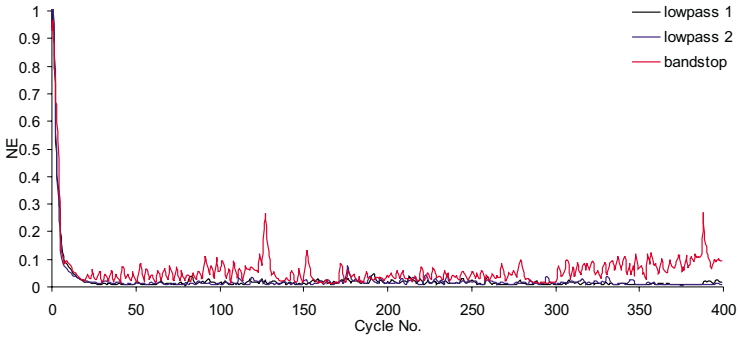


FIGURE I.30: Phase-lead error results for 10 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

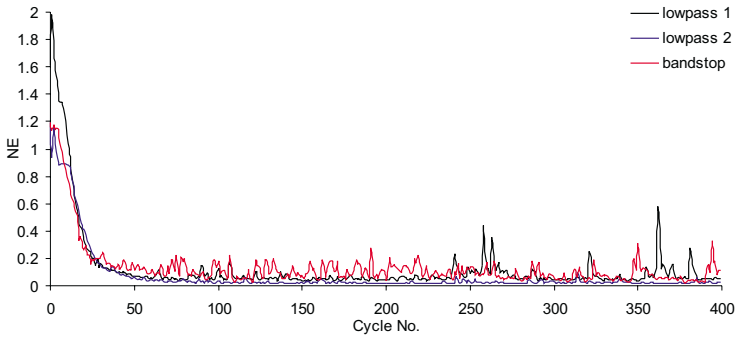


FIGURE I.31: Phase-lead error results for 20 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

I.2.3 R2 Demand

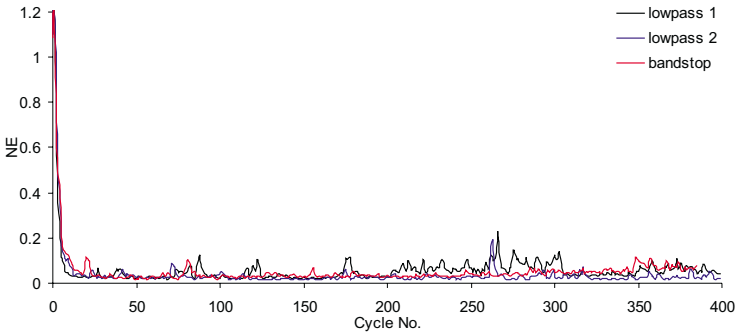


FIGURE I.32: Phase-lead error results for 10 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

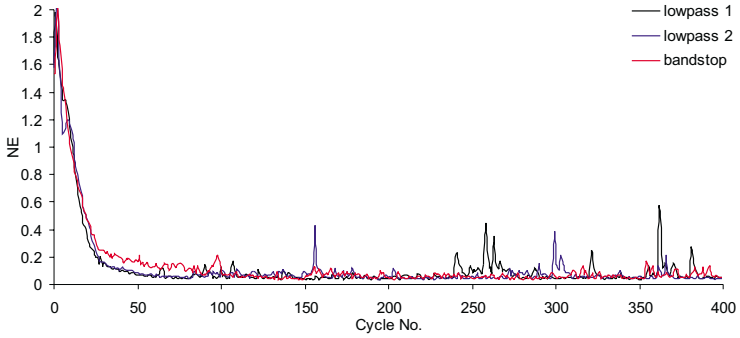


FIGURE I.33: Phase-lead error results for 20 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.1$

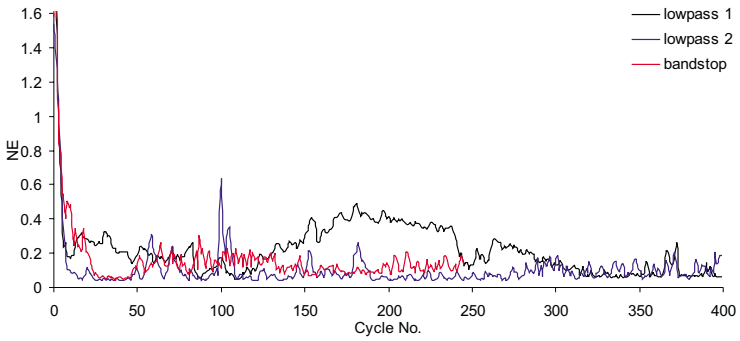


FIGURE I.34: Phase-lead error results for 20 UPM demand using a variety of causal filters, each with λ_{opt} and $\phi = 0.5$

I.3 Stage 1 Phase-lead Results with Non-causal Linear Phase FIR Filters

I.3.1 Sinewave Demand

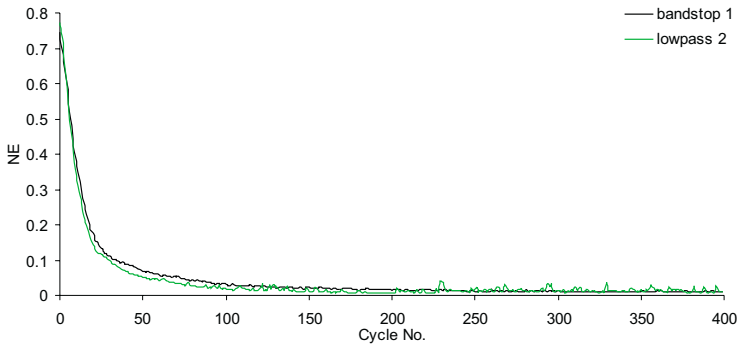


FIGURE I.35: Phase-lead error results for 10 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.1$

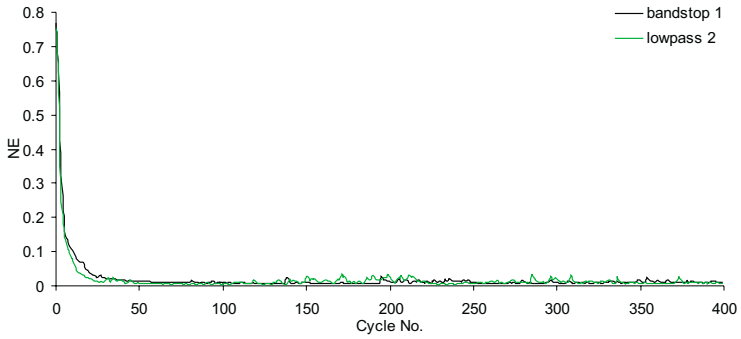


FIGURE I.36: Phase-lead error results for 10 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.5$

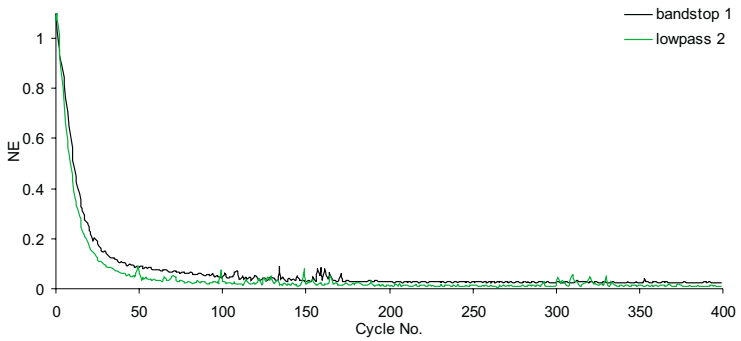


FIGURE I.37: Phase-lead error results for 20 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.1$

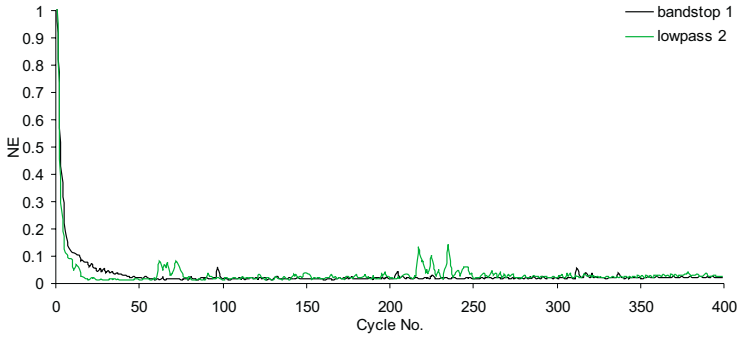


FIGURE I.38: Phase-lead error results for 20 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.5$

I.3.2 R1 Demand

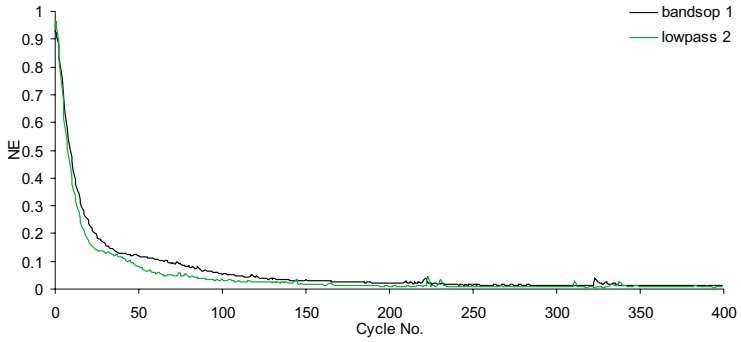


FIGURE I.39: Phase-lead error results for 10 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.1$

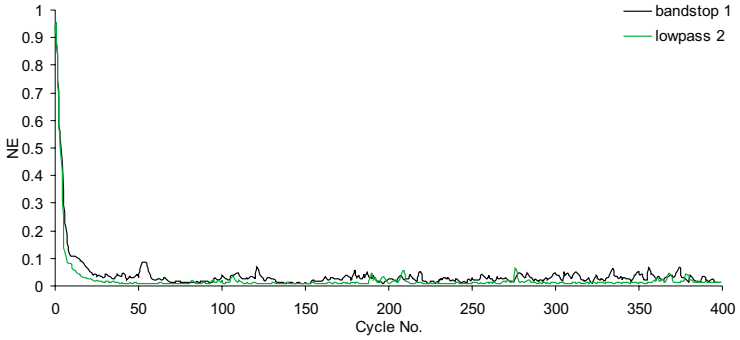


FIGURE I.40: Phase-lead error results for 10 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.5$

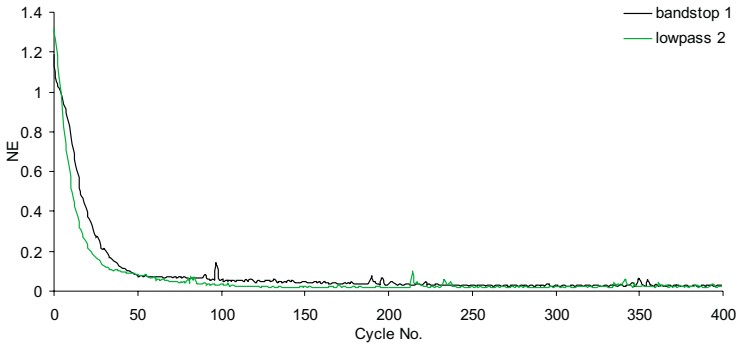


FIGURE I.41: Phase-lead error results for 20 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.1$

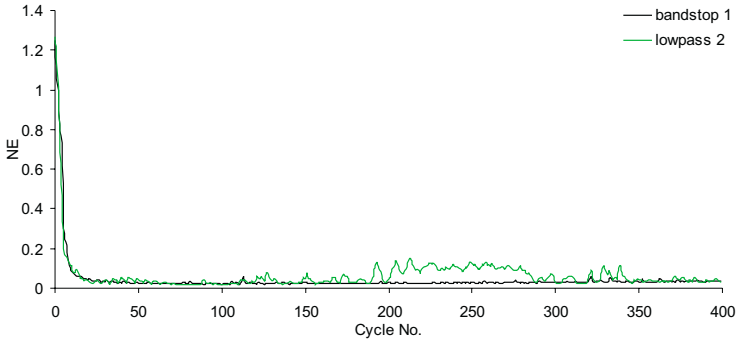


FIGURE I.42: Phase-lead error results for 20 UPM demand using two non-causal filters, both with λ_{opt} and $\phi = 0.5$

I.3.3 R2 Demand

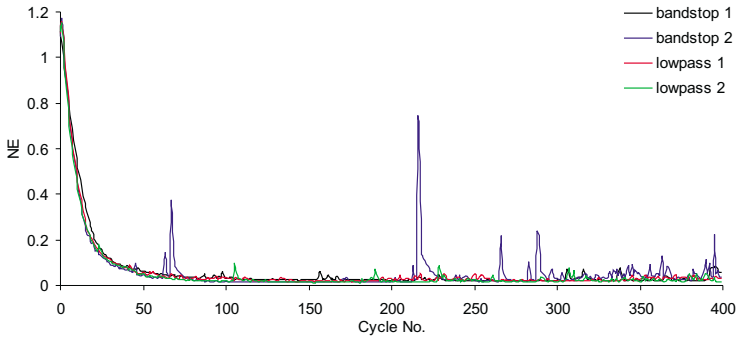


FIGURE I.43: Phase-lead error results for 10 UPM demand using a variety of non-causal filters, each with λ_{opt} and $\phi = 0.1$

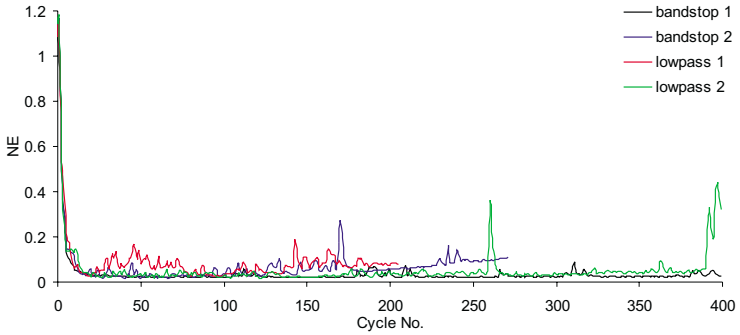


FIGURE I.44: Phase-lead error results for 10 UPM demand using a variety of non-causal filters, each with λ_{opt} and $\phi = 0.5$

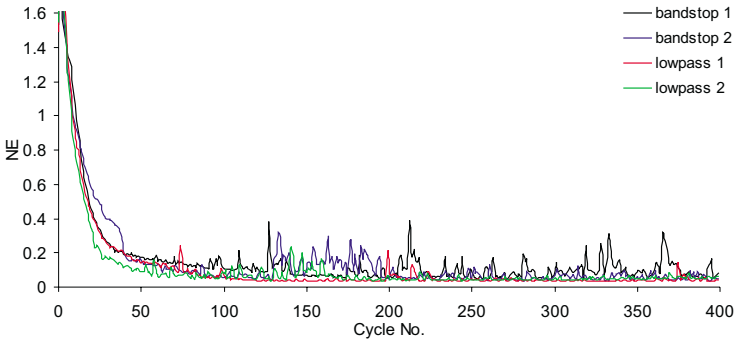


FIGURE I.45: Phase-lead error results for 20 UPM demand using a variety of non-causal filters, each with λ_{opt} and $\phi = 0.1$

I.4 Stage 2 Phase-lead Results

I.4.1 Unfiltered

I.4.1.1 Sinewave Demand

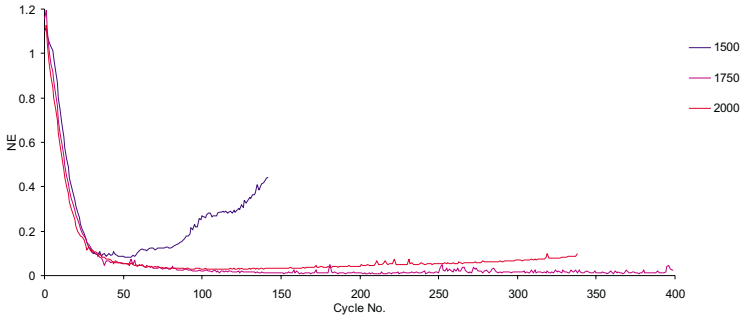


FIGURE I.46: Phase-lead error results for 20 UPM demand with $\phi = 0.1$

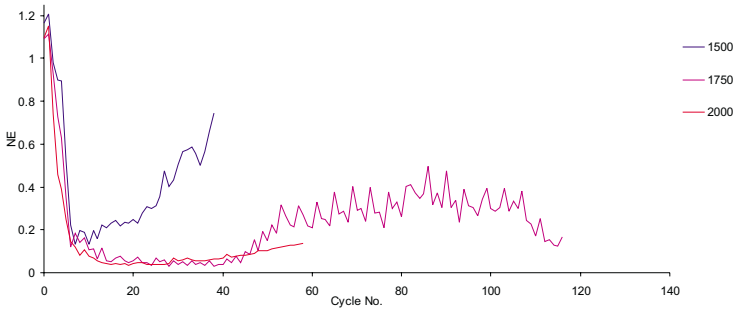


FIGURE I.47: Phase-lead error results for 20 UPM demand with $\phi = 0.5$

I.4.1.2 R1 Demand

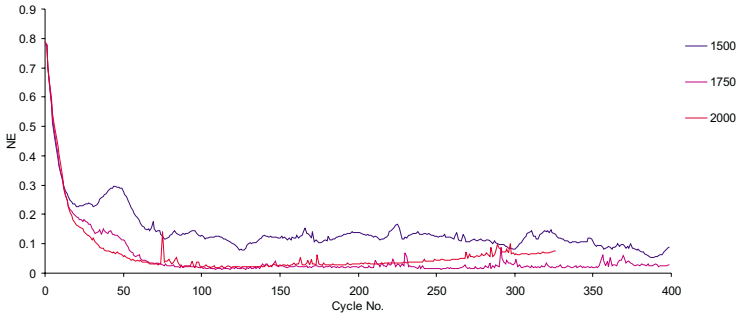


FIGURE I.48: Phase-lead error results for 10 UPM demand with $\phi = 0.1$

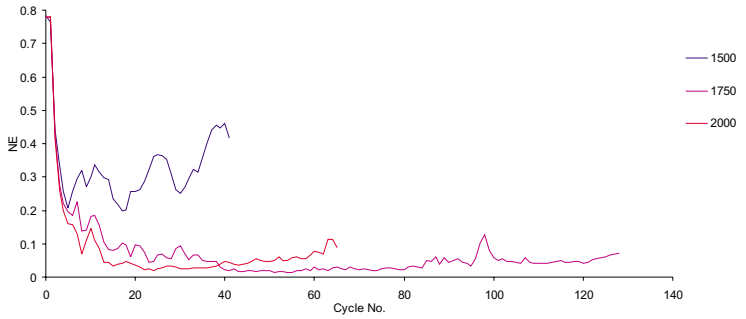


FIGURE I.49: Phase-lead error results for 10 UPM demand with $\phi = 0.5$

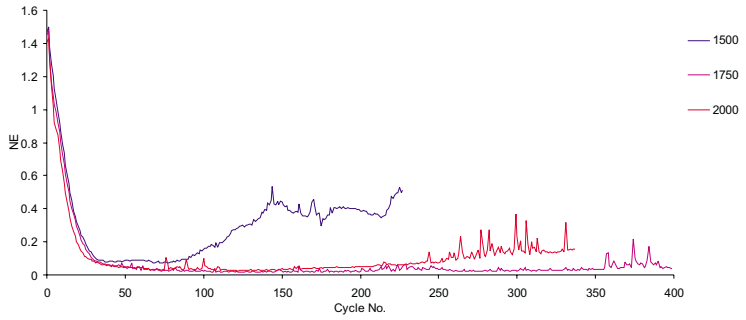


FIGURE I.50: Phase-lead error results for 20 UPM demand with $\phi = 0.1$

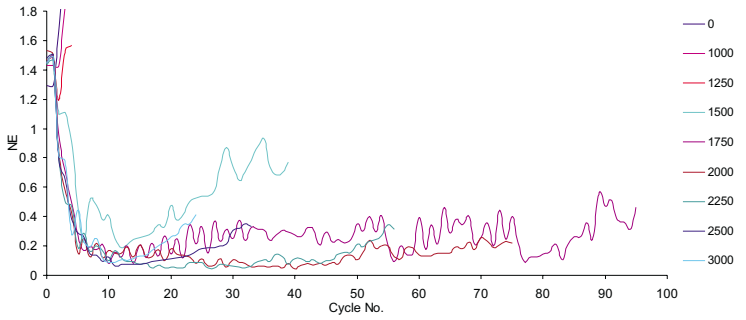


FIGURE I.51: Phase-lead error results for 20 UPM demand with $\phi = 0.5$

I.4.1.3 R2 Demand

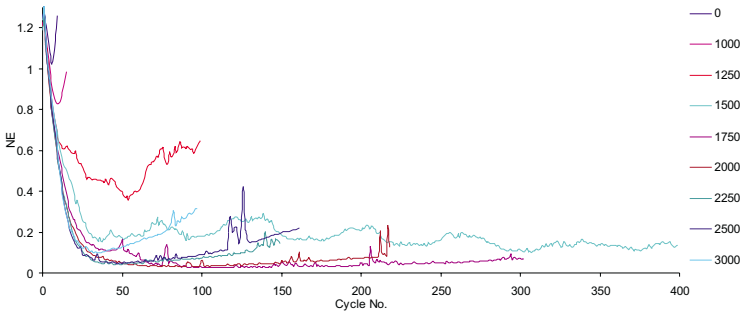


FIGURE I.52: Phase-lead error results for 10 UPM demand with $\phi = 0.1$

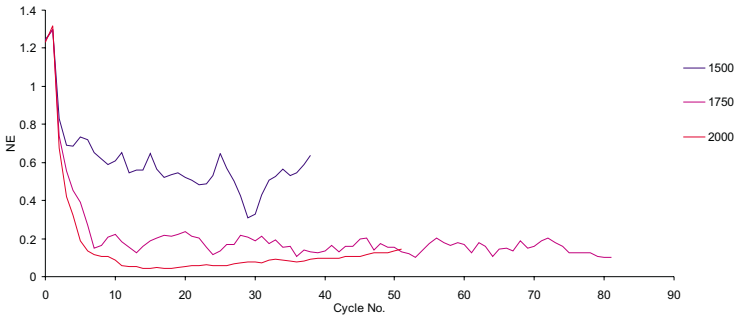


FIGURE I.53: Phase-lead error results for 10 UPM demand with $\phi = 0.5$

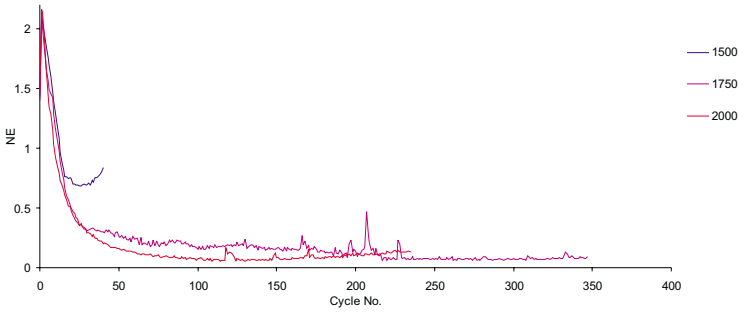


FIGURE I.54: Phase-lead error results for 20 UPM demand with $\phi = 0.1$

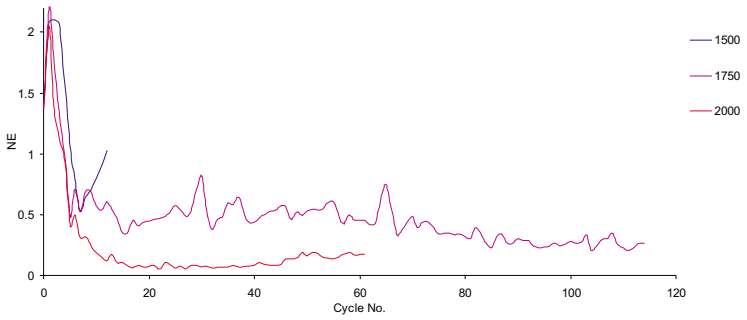


FIGURE I.55: Phase-lead error results for 20 UPM demand with $\phi = 0.5$

I.4.2 Filtered

I.4.2.1 TYPE-4-1.2

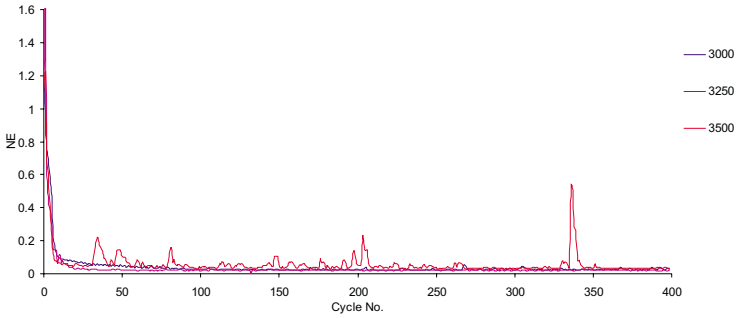


FIGURE I.56: Phase-lead error results for 20 UPM sinewave demand with various λ and $\phi = 0.5$

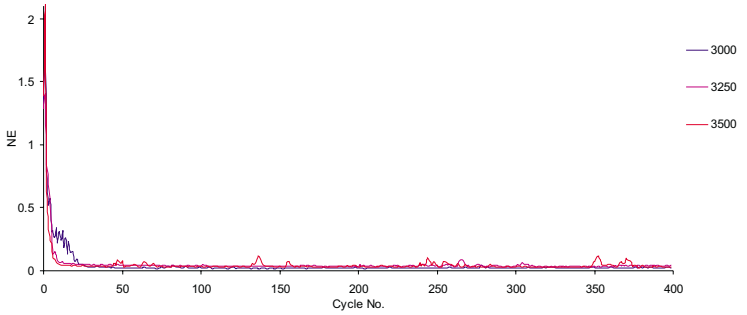


FIGURE I.57: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

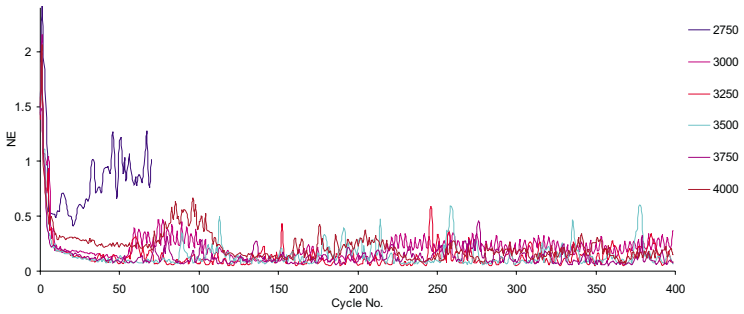


FIGURE I.58: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.4.2.2 TYPE-4-1.8

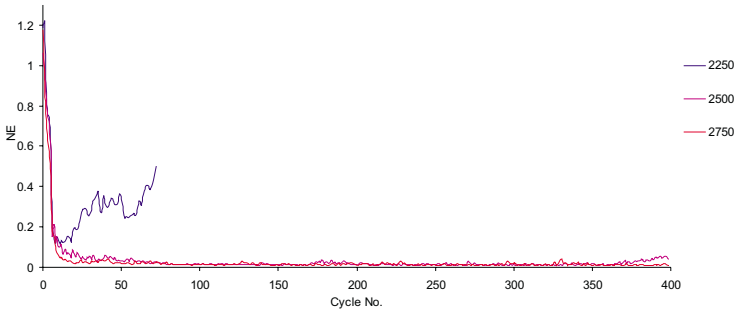


FIGURE I.59: Phase-lead error results for 20 UPM sinewave demand with various λ and $\phi = 0.5$

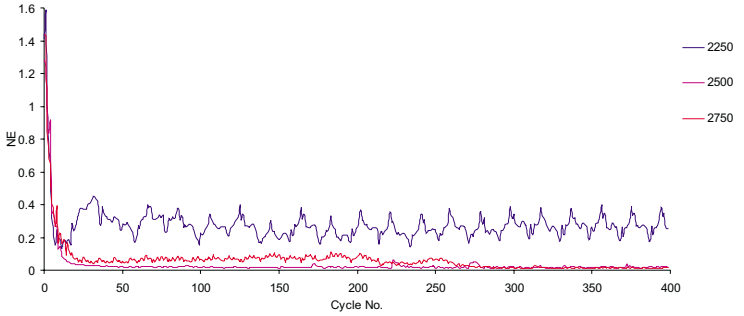


FIGURE I.60: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

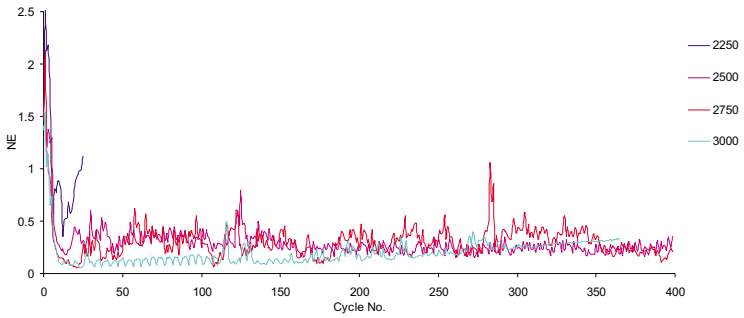


FIGURE I.61: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.4.2.3 TYPE-4-2.3

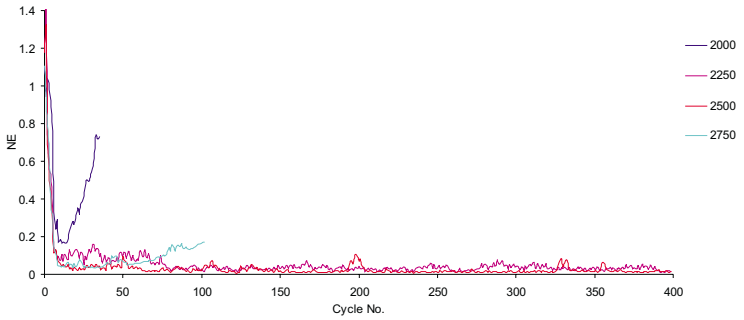


FIGURE I.62: Phase-lead error results for 20 UPM sinewave with various λ and $\phi = 0.5$

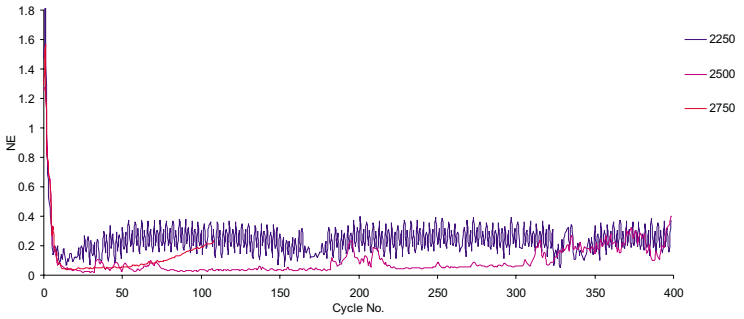


FIGURE I.63: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

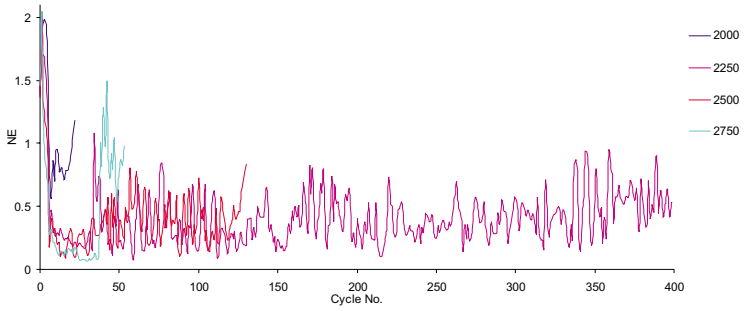


FIGURE I.64: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.4.2.4 TYPE-4-2.6

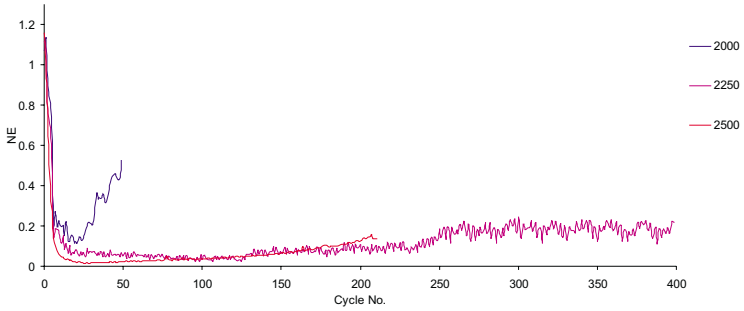


FIGURE I.65: Phase-lead error results for 20 UPM sinewave demand with various λ and $\phi = 0.5$

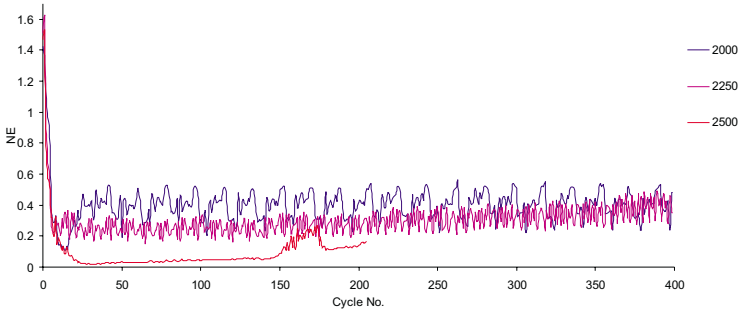


FIGURE I.66: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

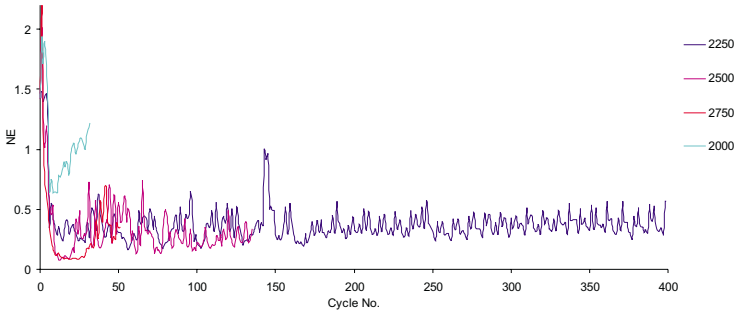


FIGURE I.67: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.4.2.5 TYPE-5-1.85

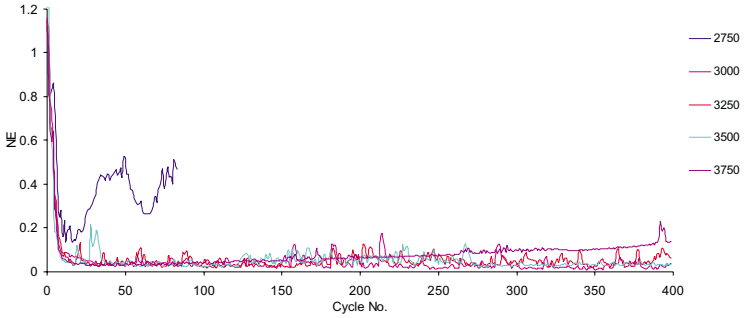


FIGURE I.68: Phase-lead error results for 20 UPM sinewave demand with various λ and $\phi = 0.5$

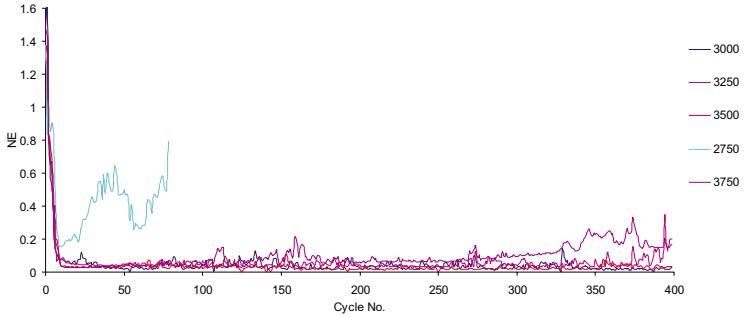


FIGURE I.69: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

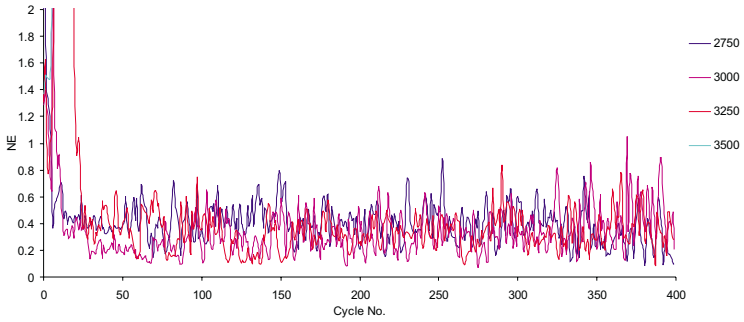


FIGURE I.70: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.4.2.6 TYPE-5-2.3

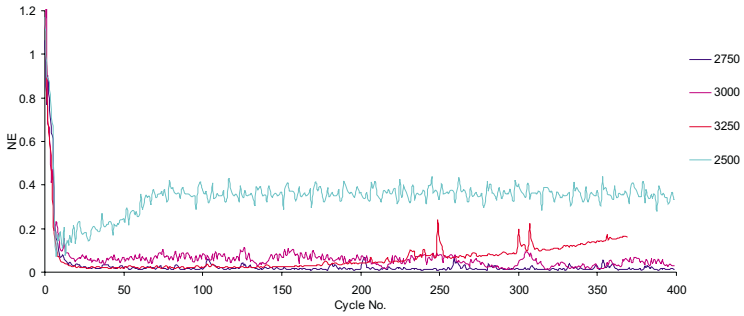


FIGURE I.71: Phase-lead error results for 20 UPM sinewave demand with various λ and $\phi = 0.5$

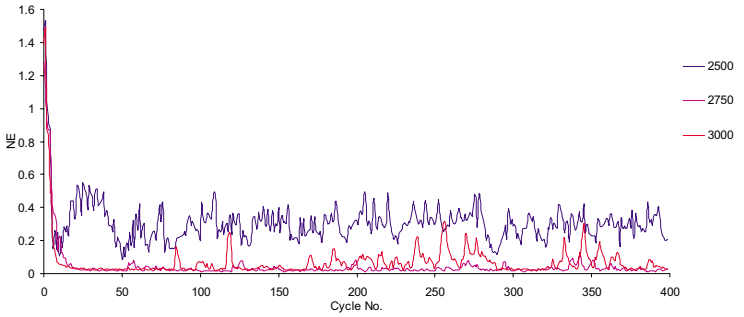


FIGURE I.72: Phase-lead error results for 20 UPM R1 demand with various λ and $\phi = 0.5$

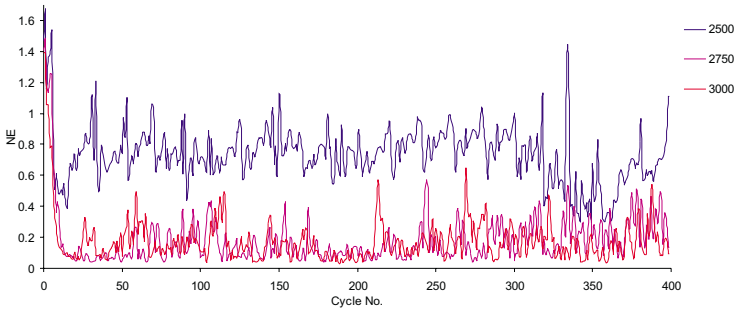


FIGURE I.73: Phase-lead error results for 20 UPM R2 demand with various λ and $\phi = 0.5$

I.5 Stage 2 Forgetting Factor

I.5.1 Unfiltered

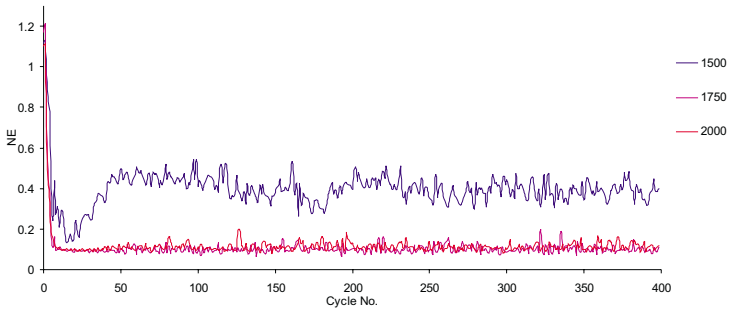


FIGURE I.74: Error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

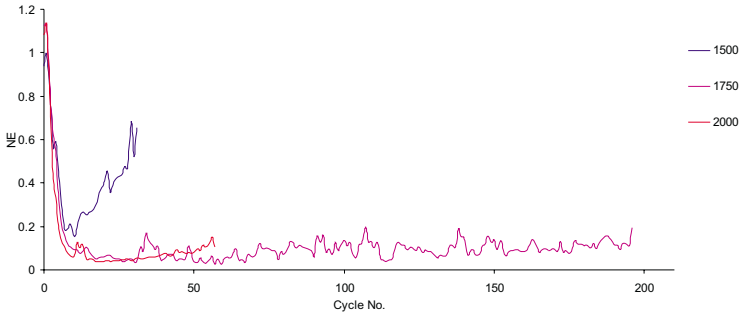


FIGURE I.75: Error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

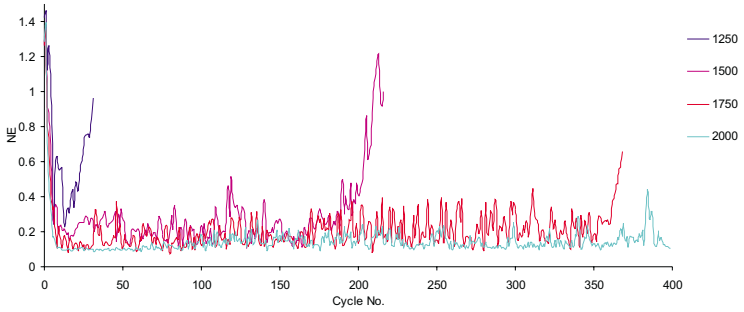


FIGURE I.76: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

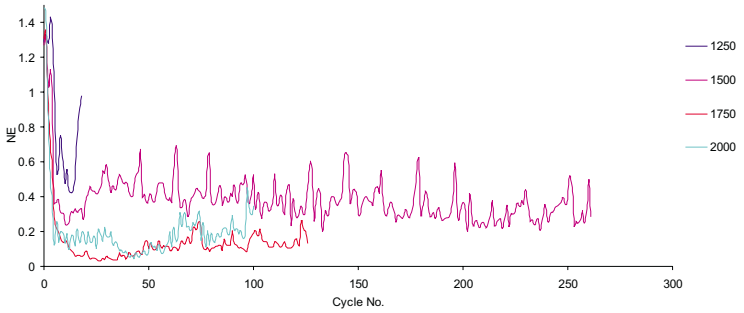


FIGURE I.77: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

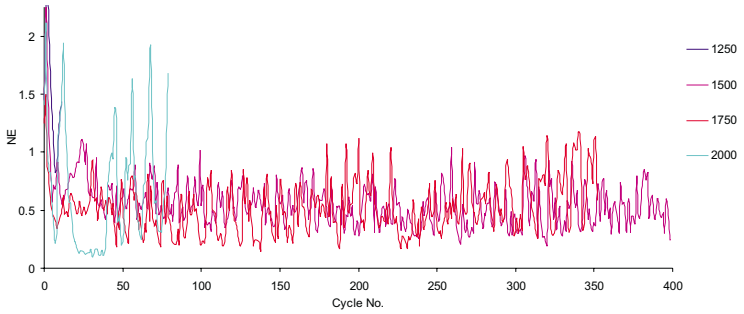


FIGURE I.78: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

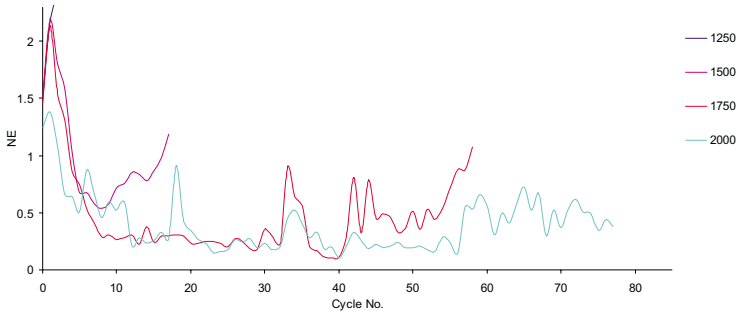


FIGURE I.79: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

I.5.2 Filtered

I.5.2.1 TYPE-4-1.2

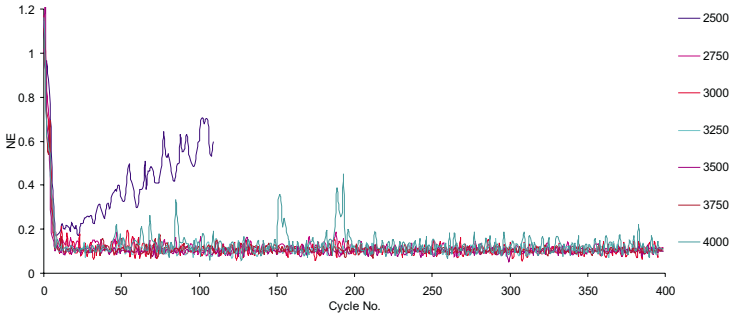


FIGURE I.80: Error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

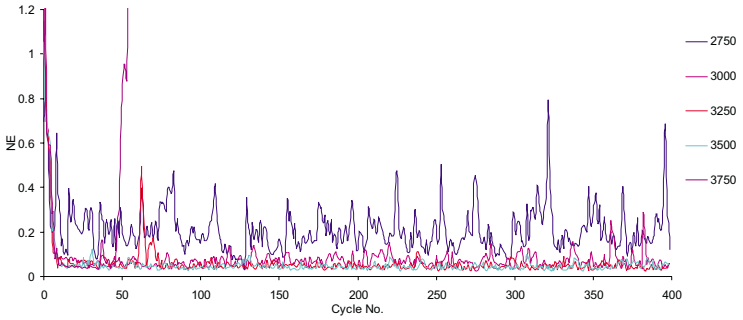


FIGURE I.81: Error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

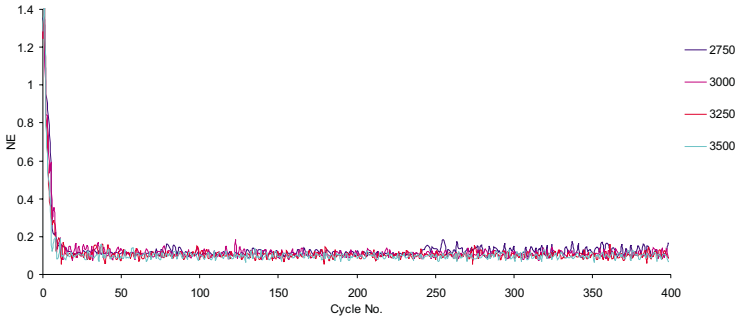


FIGURE I.82: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

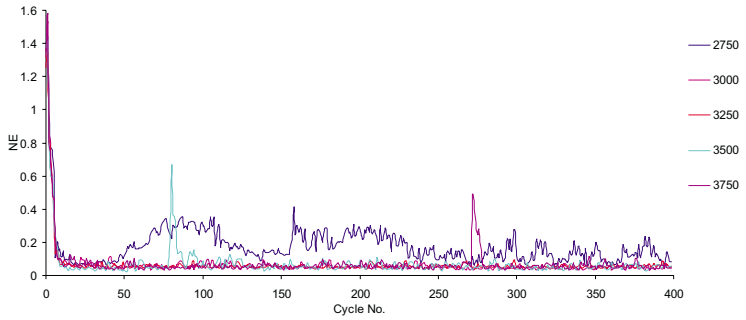


FIGURE I.83: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

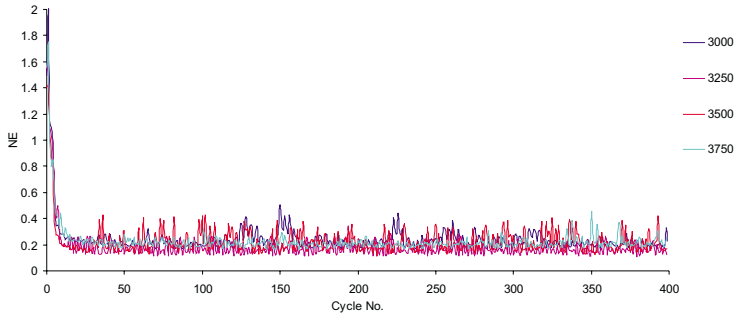


FIGURE I.84: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

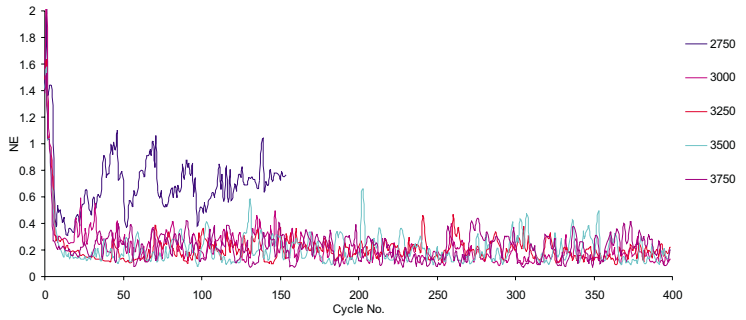


FIGURE I.85: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

I.5.2.2 TYPE-4-2.3

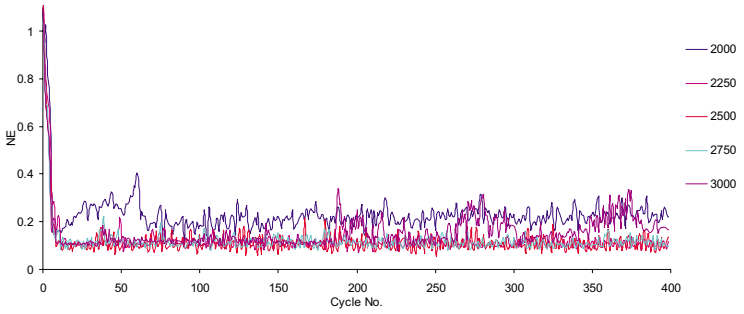


FIGURE I.86: Phase-lead error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

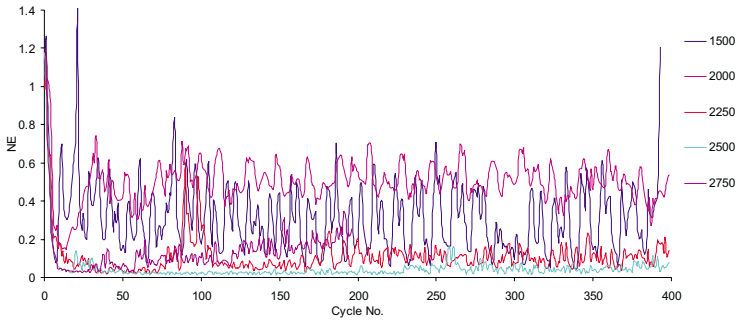


FIGURE I.87: Error results for 20 UPM sinewave demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

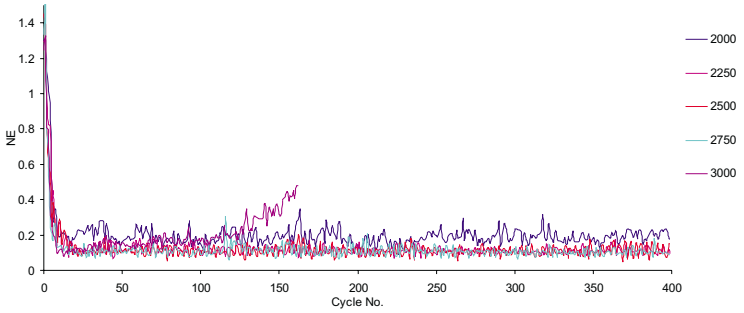


FIGURE I.88: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

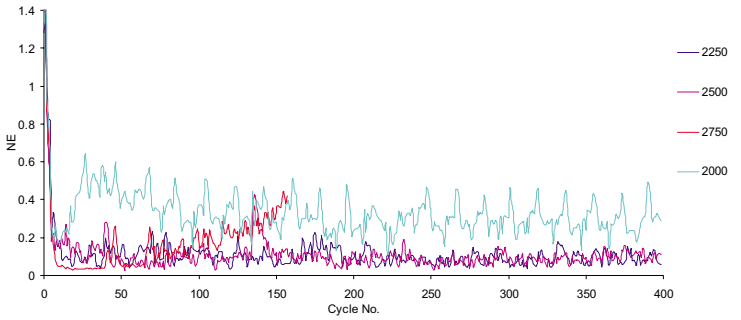


FIGURE I.89: Error results for 20 UPM R1 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

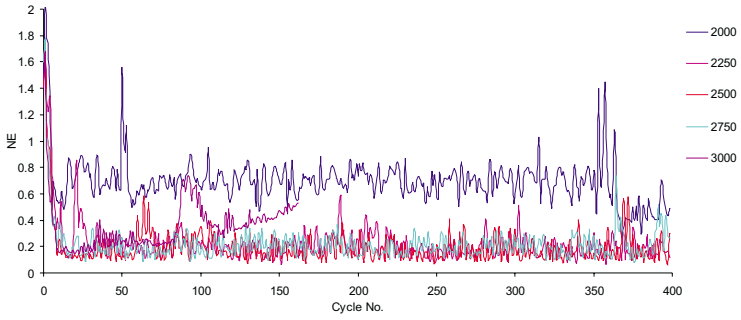


FIGURE I.90: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.95$

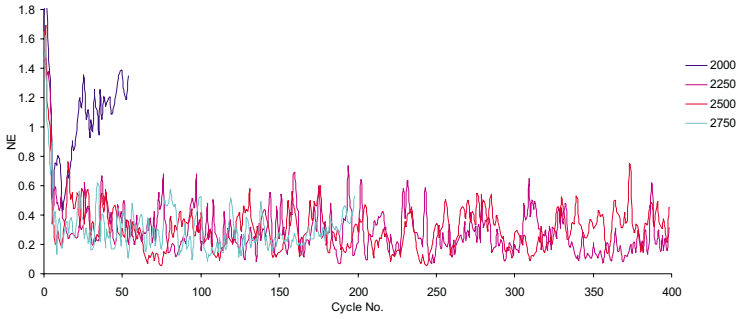


FIGURE I.91: Error results for 20 UPM R2 demand with various λ, $\phi = 0.5$ and $\gamma = 0.99$

Appendix J

Contraction Mapping Results

The results presented in here relate to various formulations of the adjoint algorithm which are described in Chapter 8.

J.1 Stage 2

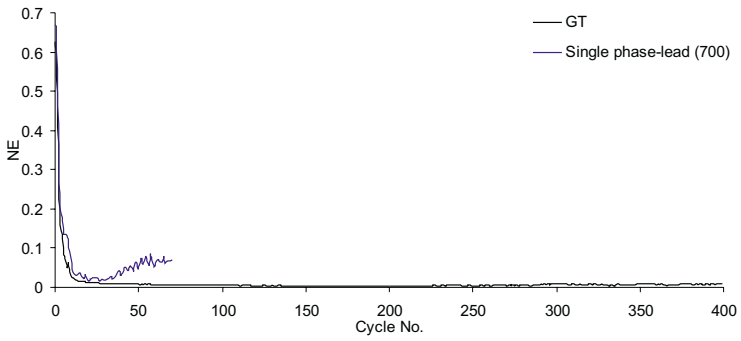


FIGURE J.1: Error results for 10 UPM sinewave demand using single phase-lead and G^T algorithms

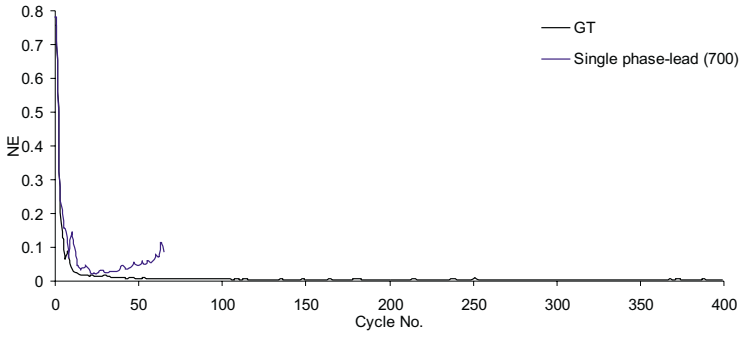


FIGURE J.2: Error results for 10 UPM R1 demand, using single phase-lead and G^T algorithms

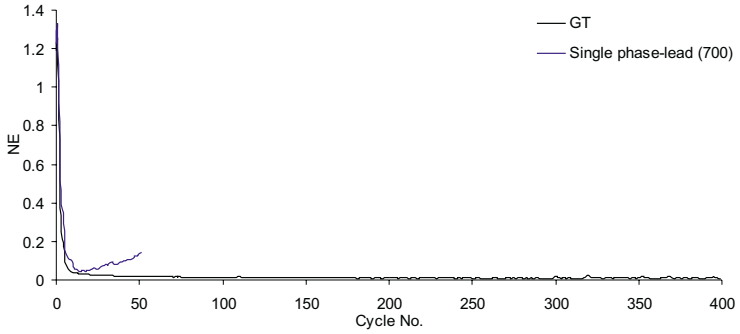


FIGURE J.3: Error results for 10 UPM R2 demand, using single phase-lead and G^T algorithms

J.2 Stage 2 Resolution Comparison

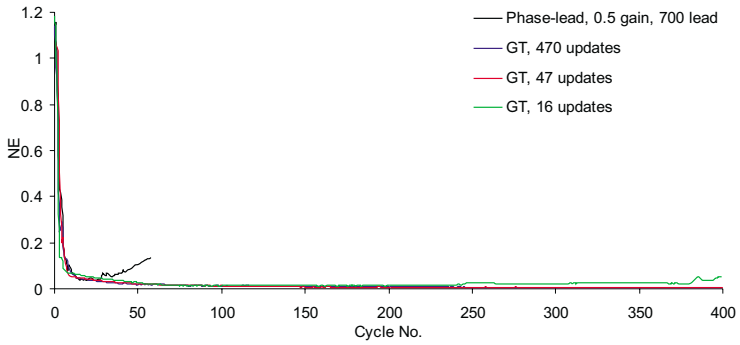


FIGURE J.4: Error results for 20 UPM sinewave demand, using of single phase-lead and G^T algorithms

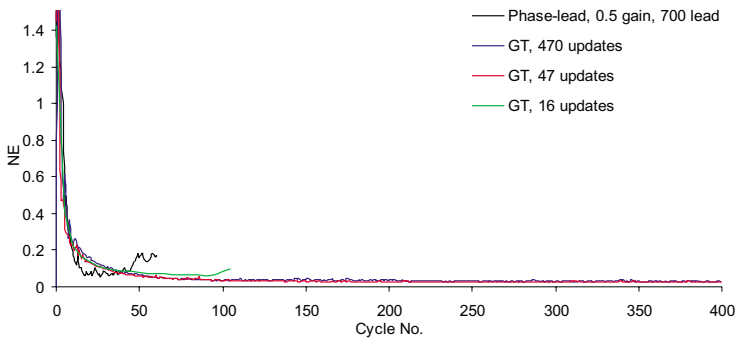


FIGURE J.5: Error results for 20 UPM R2 demand, using single phase-lead and G^T algorithms

J.3 Stage 2 Automatic G^T ILC Formulation

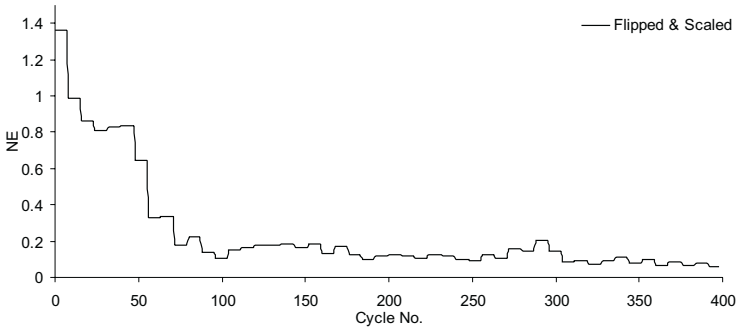


FIGURE J.6: Error results for 10 UPM sinewave demand using Automatic G^T

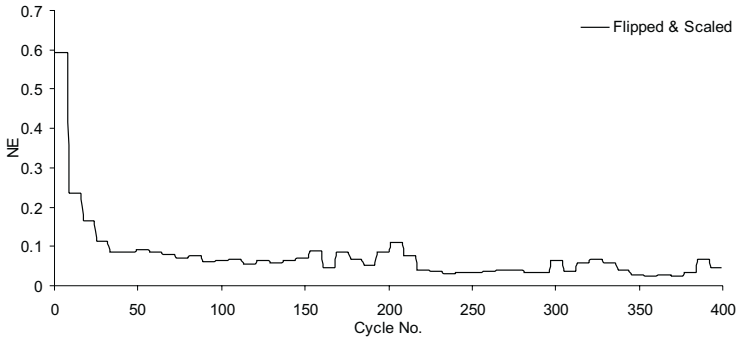


FIGURE J.7: Error results for 10 UPM R2 demand using Automatic G^T

J.4 Stage 2 Automatic G^T RC Formulation

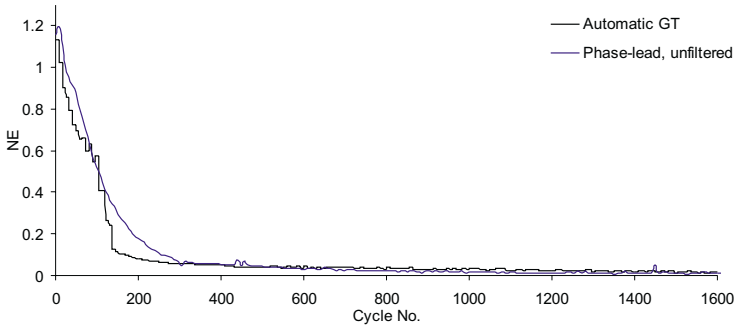


FIGURE J.8: Error results for 20 UPM sinewave demand using single phase-lead and G^T algorithms with $\phi = 0.1$

Bibliography

S. Abrate. Vibrations of belts and belt drives. *Mechanisms and Machine Theory*, 6(27):645–659, 1992.

H. Ahn, C. Choi, and K. Kim. Iterative learning control for a class of nonlinear systems. *Automatica*, 29(6), 1993.

H. Ahn and C. Choi. Iterative learning controller for linear systems with a periodic disturbance. *Electronics Letters*, 26(18), August 1990.

T. Al-Towaim, P. Lewin, and E. Rogers. Norm optimal iterative learning control applied to chain conveyor systems. In *IFAC ALCOSP*, Italy, 2001.

N. Amann, D. Owens, and E. Rogers. Iterative learning control for discrete-time systems using optimal feedback and feedforward actions. In *Proceedings of the 34th Conference on Decision and Control*, December 1995.

N. Amann, D. Owens, and E. Rogers. Iterative learning control for discrete-time systems with exponential rate of convergence. In *IEE Proceedings - Control Theory Applications*, volume 143, March 1996a.

N. Amann, D. Owens, and E. Rogers. Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control*, 65(2), 1996b.

N. Amann, D. Owens, and E. Rogers. Predictive optimal iterative learning control. *International Journal of Control*, 69(2), 1998.

M. Arif, T. Ishihara, and H. Inooka. Incorporation of experience in iterative learning controllers using locally weighted learning. *Automatica*, 37, 2001.

S. Arimoto, F. Miyazaki, S. Kawamura, and S. Tamaki. Learning control theory for dynamical systems. In *Proceedings of the 24th Conferenece on Decision and Control*, December 1985a.

S. Arimoto, F. Miyazaki, and S. Kawamura. Bettering operation of dynamical systems by learning: A new control theory for servomechanism or mechatronics systems. In *Proceedigs of the 23rd Conference on Decision and Control*, December 1984a.

S. Arimoto, F. Miyazaki, and S. Kawamura. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984b.

S. Arimoto, F. Miyazaki, and S. Kawamura. Applications of learning method for dynamic control of robot manipulators. In *Proceedings of the 24th Conference on Decision and Control*, December 1985b.

S. Arimoto, F. Miyazaki, and S. Kawamura. Can robots learn by themselves? In *2nd ISRR MIT Press*, pages 127–134, Cambridge, Massachusetts, 1985c.

S. Arimoto, F. Miyazaki, and S. Kawamura. Realisation of robot motion based on a learning method. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), Jan/Feb 1988.

S. Arimoto and T. Naniwa. Equivalence relations between learnability, output-dissipativity and strict positive realness. *International Journal of Control*, 73(10), 2000.

K. Avrachenkov, H. Beigi, and R. Longman. Updating procedures for iterative learning control in hilbert space. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

A. Barton, P. Lewin, and D. Brown. Practical implementation of a real-time iterative learning position controller. *International Journal of Control*, 73(10), 2000.

Z. Bien and K. Huh. Higher order iterative learning control algorithm. In *IEE Proceedings*, volume 136, May 1989.

Z. Bien, Xu, and J. Kluwer. *Iterative Learning Control, Analysis, Design, Integration and Applications*. Academic Publishers, 1998.

H. Blomberg and R. Ylinen. *Algebraic Theory for Multivariable Linear Systems*. Academic Press, 1983.

P. Bondi, G. Casalino, and L. Gambardella. On the iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics and Automation*, 4(1), Feb 1988.

G. Bone. A novel iterative learning control formulation of generalized predictive control. *Automatica*, 31(10), 1995.

B. K. Bose. *Power Electronics and AC Drives*. Prentice Hall, 1987.

K. Chen and R. W. Longman. Stability issues using fir filtering in repetitive control. *Advances in the Astronautical Sciences*, 206:1321–1339, 2002.

W. Chen, Y. Soh, and C. Yin. Technical note: Iterative learning control for constrained nonlinear systems. *International Journal of Systems Science*, 30(6), 1999a.

Y. Chen, W. Gong, and M. Sun. An iterative learning controller with initial state learning. *IEEE Transactions on Automatic Control*, 44(2), Feb 1999b.

Y. Chen, Z. Gong, and C. Wen. Analysis of a high-order iterative learning algorithm for uncertain nonlinear systems with state delays. *Automatica*, 34(3), 1998a.

Y. Chen, M. Sun, B. Huang, and H. Dau. Robust higher order repetitive learning control algorithm for tracking control of delayed repetitive systems. In *Proceedings of the 31st Conference on Decision and Control*, December 1992.

Y. Chen, C. Wen, and M. Sun. An initial state learning method for iterative learning control of uncertain time-varying systems. In *Proceedings of the 35th Conference on Decision and Control*, December 1996a.

Y. Chen, C. Wen, and M. Sun. A robust high-order iterative learning controller using current iteration tracking error. *International Journal of Control*, 68(2), 1997.

Y. Chen, C. Wen, J. Xu, and M. Sun. High-order iterative learning identification of projectile's aerodynamic drag coefficient curve from radar measured velocity data. *IEEE Transactions on Control Systems Technology*, 6(4), July 1998b.

Y. Chen, J. Xu, and T. Lee. Current iteration tracking error assisted iterative learning control of uncertain nonlinear discrete-time systems. In *Proceedings of the 35th Conference on Decision and Control*, December 1996b.

Y. Chen, J. Xu, and T. Lee. An iterative learning controller using current iteration tracking error information and initial state learning. In *Proceedings of the 35th Conference on Decision and Control*, December 1996c.

Y. Chen, J. Xu, and C. Wen. Iterative learning-based extraction of aerobomb drag. *Journal of Spacecraft*, 35(2), 1998c.

C. Chien and J. Liu. A p-type iterative learning controller for robust output tracking of nonlinear time-varying systems. *International Journal of Control*, 64(2), 1996.

C. Chien. A discrete iterative learning control of nonlinear time-varying systems. In *Proceedings of the 35th Conference on Decision and Control*, December 1996.

C. Chien. A discrete iterative learning control for a class of nonlinear time-varying systems. *IEEE Transactions on Automatic Control*, 43(5), May 1998.

J. Choi, H. Choi, K. Lee, and W. Lee. Control of ethanol concentration in a fed-batch cultivation of acinetobacter calcaceticus rag-1 using a feedback-assisted iterative learning algorithm. *Journal of Biotechnology*, 49, 1996.

J. Choi and S. Lee. Adaptive iterative learning control of uncertain robotic systems. In *IEE Proceedings - Control Theory Applications*, volume 147, March 2000.

T. Chow and Y. Fang. An iterative learning control method for continuous-time systems based on 2-d system theory. *IEEE Transactions on Circuits and Systems- 1: Fundamental Theory and Applications*, 45(4), April 1998.

I. Colak, S. D. Garvey, and M. T. Wright. Simulation of induction machines using phase variables and inverse inductance matrix. *International Journal of Electrical Engineering Education*, 32:354–365, 1995.

A. K. DeSarkar and G. J. Berg. Digital simulation of three-phase induction motors. *IEEE Transactions on Power Apparatus and Systems*, 6(PAS-89):1031–1037, 1970.

T. Doh, J. Moon, and M. Chung. An iterative learning control for uncertain systems using structured singular value. *Transactions of the ASME*, 121, December 1999a.

T. Doh, J. Moon, K. Jin, and M. Chung. Robust iterative learning control with current feedback for uncertain linear systems. *International Journal of Systems Science*, 30(1), 1999b.

H. Dou, K. Tan, T. Lee, and Zhou. Iterative learning feedback control of human limbs via functional electrical stimulation. *Control Engineering Practice*, 7, 1999.

H. Elci, R. W. Longman, M. Phan, J. N. Juang, and R. Ugoletti. Automated learning control through model updating for precision motion control. *ASME Adaptive Structures and Composite Materials: Analysis and Applications*, AD-Vol. 45/MD-Vol. 54:299–314, 1994.

Y. Fang and T. Chow. Iterative learning control of linear discrete time multivariable systems. *Automatica*, 34 (11), 1998.

M. French, G. Munde, E. Rogers, and D. Owens. Recent developments in adaptive iterative learning control. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

M. French and E. Rogers. Non-linear iterative learning by an adaptive lyapunov technique. *International Journal of Control*, 73(10), 2000.

J. Frueh and M. Phan. Linear quadratic optimal learning control (lql). *International Journal of Control*, 73(10), 2000.

K. Furuta and M. Yamakita. Iterative generation of optimal input of a manipulator. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 579 – 584, April 1986.

K. Fututa and M. Yamakita. The design of learning control systems for multivariable systems. In *Proceedings of the IEEE International Symposium on Intelligent Control*, pages 371–376, Philadelphia, Pennsylvania, 1987.

S. Garimella and K. Srinivasan. Application of iterative learning control to coil-to-coil control in rolling. *IEEE Transactions on Control Systems Technology*, 6(2), March 1998.

Z. Geng, R. Carroll, and J. Xies. Two-dimensional model and algorithm analysis for a class of iterative learning control systems. *International Journal of Control*, 52(4):833–862, 1990.

J. Ghosh and B. Paden. Pseudo-inverse based iterative learning control for nonlinear plants with disturbances. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

J. Ghosh and B. Paden. Iterative learning control for non-linear non-minimum phase plants. *Journal of Dynamic Systems, Measurement, and Control*, 123(21), March 2001.

D. Gorinevsky. Distributed system loopshaping design of iterative control for batch processes. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

M. Grundelius and B. Bernhardsson. Constrained iterative learning control of liquid slosh in an industrial packaging machine. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

M. Grundelius. Iterative optimal control of liquid slosh in an industrial packaging machine. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

S. Gunnarsson and M. Norrlof. Some aspects of an optimization approach to iterative learning control. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

K. Hamamoto and T. Sugie. An iterative learning control algorithm within prescribed input-output subspace. *Automatica*, 37, 2001.

N. N. Hancock. *Matrix Analysis of Electrical Machinery*. Pergamon Press, Second Edition, 1974.

S. Hara, T. Omata, and M. Nakano. Synthesis of repetitive control systems and its applications. In *Proceedings of the 24th Conference on Decision and Control*, December 1985.

J. Hatonen, C. Freeman, D. Owens, P. Lewin, and E. Rogers. Robustness analysis of a gradient-based repetitive algorithm for discrete-time systems. December 2004.

J. Hatonen, T. J. Harte, D. H. Owens, J. D. Ratcliffe, and P. L. Lewin. A new robust iterative learning control algorithm for application on a gantry robot. In *Proceedings of 9th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 305–312, Lisbon, Portugal, 2003a.

J. Hatonen, D. H. Owens, and K. L. Moore. An algebraic approach to iterative learning control. *International Journal of Control*, 77(1):45–54, 2003b.

J. Hatonen, D. H. Owens, and R. Ylinen. A new optimality based repetitive control algorithm for discrete-time systems. In *Proceedings of the European Control Conference (ECC03)*, Cambridge, UK, September 2003c.

H. Havlicsek and A. Alleyne. Nonlinear control of an electrohydraulic injection molding machine via iterative adaptive learning. *IEEE/ASME Transactions on Mechatronics*, 4(3), Sept 1999.

S. Hillenbrand and M. Pandit. A discrete-time iterative learning control law with exponential rate of convergence. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

S. Hillenbrand and M. Pandit. An iterative learning controller with reduced sampling rate for plants with variations of initial states. *International Journal of Control*, 73(10), 2000.

D. Hwang, Z. Bien, and B. Kim. Decentralized iterative learning control methods for large scale linear dynamic systems. *International Journal of Systems Science*, 24(12), 1993.

T. Ishihara, K. Abe, and H. Takeda. A discrete-time design of robust iterative learning controllers. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(1), January/February 1992.

H. S. Jang and R. W. Longman. A new learning control law with monotonic decay of the tracking error norm. In *Preceedings of the 32nd Annual Allerton Conference on Communication, Control, and Computing*, pages 314–323, Monticello, Illinois, 1994.

T. Jang, H. Ahn, and C. Choi. Iterative learning control discrete-time nonlinear systems. *International Journal of Systems Science*, 25(7), 1994.

T. Jang, H. Ahn, and C. Choi. Iterative learning control in feedback systems. *Automatica*, 31(2), 1995.

P. Jiang and R. Unbehauen. An iterative learning control scheme with dead-zone. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

Y. Jiang, D. Clements, and T. Hesketh. Adaptive repetitive control of nonlinear systems. In *Proceedings of the 34th Conference on Decision and Control*, December 1995a.

Y. Jiang, D. Clements, and T. Hesketh. Betterment learning control of nonlinear systems. In *Proceedings of the 34th Conference on Decision and Control*, December 1995b.

C. V Jones. *Unified Theory of Electrical Machines*. Butterworth and Co., 1967.

H. E. Jordan. Analysis of induction machines in dynamic systems. *IEEE Transactions on Power Apparatus and Systems*, 11(PAS-84):1080–1088, 1965.

P. J. Judd, L. Hideg, and R. P. Van Til. Coefficient test for stability analysis of iterative learning control systems. In *Proceedings of the 30th Conference on Decision and Control*, December 1991.

D. Kim and S. Kim. An iterative learning control method with application for cnc machine tools. *IEEE Transactions on Industry Applications*, 32(1), Jan/Feb 1996.

W. C. Kim, I. S. Chin, K. S. Lee, and J. Choi. Analysis and reduced-order design of quadratic criterion-based iterative learning control using singular value decomposition. *Computers and Chemical Engineering*, 24, 2000.

P. C. Krause. Simulation of symmetrical induction machinery. *IEEE Transactions on Power Apparatus and Systems*, 11(PAS-84):1038–1053, 1965.

P. C. Krause. The method of multiple reference frames applied to the analysis of symmetrical induction machinery. *IEEE Transactions on Power Apparatus and Systems*, 1(PAS-87):218–227, 1968.

T. Kuc, J. Lee, and K. Nam. An iterative learning control theory for a class of nonlinear dynamic systems. *Automatica*, 28(6), 1992.

T. Kuc, K. Nam, and J. Lee. An iterative learning control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 7(6), December 1991.

J. Kurek and M. Zaremba. Iterative learning control synthesis based on 2-d system theory. *IEEE Transactions on Automatic Control*, 38(1), January 1993.

J. Kurek. Correspondence: Counterexample to iterative learning control of linear discrete -time multivariable systems. *Automatica*, 36, 2000.

H. Lee and Z. Bien. Study on robustness of iterative learning control with non-zero initial error. *International Journal of Control*, 64(3), 1996.

H. Lee and Z. Bien. A note on convergence property of iterative learning controller with respect to sup norm. *Automatica*, 33(8), 1997.

J. Lee and J. Lee. Design of iterative learning controller with vcr servo system. *IEEE Transactions on Consumer Electronics*, 39(1), February 1993.

J. Lee, K. Lee, and W. Kim. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36, 2000.

K. S. Lee, S. H. Bang, S. Yi, J. S. Son, and S. C. Yoon. Iterative learning control of heat-up phase for a batch polymerization reactor. *Journal of Process Control*, 6(4), 1996.

P. Lewin. Iterative learning control of repetitive processes. In *Proceedings of the 2nd International Conference on Walking and Climbing Robots*, pages 295–303, Portsmouth, UK, 1999.

F. L. Lewis and V. L. Syrmos. *Optimal Control*. John Wiley & Sons, Inc., 1995.

Y. Liang and D. Looze. Performance and robustness issues in iterative learning control. In *Proceedings of the 32nd Conference on Decision and Control*, December 1993.

R. W. Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10), 2000.

A. De Luca and S. Panzier. An iterative scheme for learning gravity compensation in flexible robot arms. *Automatica*, 30(6), 1994.

M. L. MacDonald and P. C. Sen. Control loop study of induction motor drives using dq model. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, 4(26):237–243, Nov 1979.

L. Ma, T. Low, and S. Tso. Discrete iterative learning controller. *Electronics Letters*, 29(12), June 1993.

T. Mita and E. Kato. Iterative control and its application to motion control of robotic arm - a direct approach to servo problems. In *Proceedings of the 24th Conference on Decision and Control*, December 1985.

J. Moon, T. Doh, and M. Chung. A robust approach to iterative learning control design for uncertain systems. *Automatica*, 34(8), 1998.

J. Moon, M. Lee, and M. Chung. Track-following control for optical disk drives using an iterative learning scheme. *IEEE Transactions on Consumer Electronics*, 42(2), May 1996.

K. Moore and J. Xu. Editorial: Special issue on iterative learning control. *International Journal of Control*, 73 (10), 2000.

K. Moore. Multi-loop control approach to designing iterative learning controllers. In *Proceedings of the 37th Conference on Decision and Control*, pages 666–671, December 1998.

K. Moore. An iterative learning control algorithm for systems with measurement noise. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

K. Moore. A non-standard iterative learning control approach to tracking periodic signals in discrete-time non-linear systems. *International Journal of Control*, 73(10), 2000.

M. Norrlof and S. Gunnarsson. A frequency domain analysis of a second order iterative learning control algorithm. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

M. Norrlof and S. Gunnarsson. Disturbance aspects of iterative learning control. *Engineering Applications of Artificial Intelligence*, 14, 2001.

M. Norrlof. Comparative study on first and second order ilc- frequency domain analysis and experiments. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

S. Oh, Z. Bien, and D. Hwang. Iterative learning control method for discrete-time dynamic systems. In *IEE Proceedings-D*, volume 138, March 1991.

S. Oh, Z. Bien, and I. Suh. An iterative learning control method with application for the robot manipulator. *IEEE Journal of Robotics and Automation*, 4(5), 1988.

G. Oriolo, S. Panzier, and G. Ulivi. Learning optimal trajectories for non-holonomic systems. *International Journal of Control*, 73(10), 2000.

D. Owens and G. Munde. Error convergence in an adaptive iterative learning controller. *International Journal of Control*, 73(10), 2000.

D. Owens. Iterative learning control - convergence using high gain feedback. In *Proceedings of the 31st Conference on Decision and Control*, December 1992.

M. Pandit and K. Buchheit. Optimization iterative learning control of cyclic production processes with application to extruders. *IEEE Transactions on Control Systems Technology*, 7(3), May 1999.

K-H. Park and Z. Bien. A study on the robustness of a pid-type iterative learning controller against initial state error. *International Journal of Systems Science*, 30:49–59, 1999.

K. Park, Z. Bien, and D. Hwang. Design of an iterative learning controller for a class of linear dynamic systems with time delay. In *IEE Proceedings - Control Theory Applications*, volume 145, Nov 1998.

K. Park and Z. Bien. A generalized iterative learning controller against initial state. *International Journal of Control*, 73(10), 2000.

M. Phan and J. Frueh. Learning control for trajectory tracking using basis functions. In *Proceedings of the 35th Conference Decision and Control*, December 1996.

M. Phan and J. Frueh. Model reference adaptive learning control with basis functions. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

A. Poo, K. Lim, and Y. Ma. Application of discrete learning control to a robotic manipulator. *Robotics and Computer-Integrated Manufacturing*, 12(1), 1996.

B. Porter and S. Mohammed. Iterative learning control of completely irregular plants with initial state shifting. *Electronics Letters*, 26(2), January 1990a.

B. Porter and S. Mohammed. Iterative learning control of completely irregular plants with initial impulsive action. *Electronics Letters*, 26(13), June 1990b.

B. Porter and S. Mohammed. Iterative learning control of partially irregular multivariable plants with initial state shifting. *International Journal of Systems Science*, 22(2), 1991a.

B. Porter and S. Mohammed. Iterative learning control of partially irregular multivariable plants with initial impulsive action. *International Journal of Systems Science*, 22(3), 1991b.

B. Porter and S. Mohammed. Digital iterative learning control of compensated linear multivariable plants. *International Journal of Systems Science*, 23(11), 1992b.

B. Porter and S. Mohammed. Digital iterative learning control of linear multivariable plants. *International Journal of Systems Science*, 23(9), 1992a.

B. Porter and S. Mohammed. Learning rates in the digital iterative learning control of linear multivariable plants. *International Journal of Systems Science*, 23(10), 1992c.

M. H. Rashid. *Power Electronics - Circuits, Devices and Applications, Second Edition*. Prentice Hall, 1993.

S. D. T. Robertson and K. M. Hebbar. A digital model for three-phase induction machines. *IEEE Transactions on Power Apparatus and Systems*, 11(PAS-88):1624–1634, 1969.

D. De Roover, O. Bosgra, and M. Steinbuch. Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control*, 73(10), 2000.

D. De Roover and O. Bosgra. Synthesis of robust iterative learning controllers with application to a wafer stage motion system. *International Journal of Control*, 73(10), 2000.

D. De Roover. Synthesis of a robust iterative learning controller using an h approach. In *Proceedings of the 35th Conference on Decision and Control*, December 1996.

W. Seo, B. Park, and J. Lee. Intelligent learning control for a class of nonlinear dynamic systems. In *IEE Proceedings - Control Theory Applications*, volume 146, March 1999.

L. Sison and E. Chong. No reset iterative learning control. In *Proceedings of the 35th Conference on Decision and Control*, December 1996.

G. R. Slemon. Modelling of induction machines for electric drives. *IEEE Transactions on Industry Applications*, 6(25):1126–1131, 1989.

M. C. Smith. Synthesis of mechanical networks: The inerter. *IEEE Transactions on Automatic Control*, 47(10), October 2002.

T. Sogo and N. Adachi. Convergence rates and robustness of iterative learning control. In *Proceedings of the 35th Conference on Decision and Control*, December 1996.

T. Sogo, K. Kinoshita, and N. Adachi. Iterative learning control using adjoint systems for nonlinear non-minimum phase systems. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

V. R. Stefanovic and T. H. Barton. The speed-torque transfer function of electric drives. *IEEE Transactions on Industry Applications*, 5(IA-12):428–436, 1977.

T. Sugie and T. Ono. On an iterative learning control law for dynamical systems. In *IFAC 10th Triennial World Congress*, Munich, 1987.

T. Sugie and T. Ono. An iterative learning control law for dynamical systems. *Automatica*, 27(4), 1991.

K. Tan, H. Dou, Y. Chen, and T. Lee. High precision linear motor control via relay-tuning and iterative learning based on zero-phase filtering. *IEEE Transactions on Control Systems Technology*, 9(2), March 2001.

A. Tayebi and M. Zaremba. Exponential convergence of an iterative learning controller for time-varying nonlinear systems. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

A. Tayebi and M. Zaremba. Internal model-based robust iterative learning control for uncertain lti systems. In *Proceedings of the 39th Conference on Decision and Control*, December 2000.

M. Togai and O. Yamano. Analysis and design of an optimal learning control scheme for industrial robots: A discrete system approach. In *Proceedings of the 24th Conference on Decision and Control*, December 1985.

P. Vas. *Vector Control of AC Machines*. Oxford University Press, 1990.

S. M. Veres. Iterative identification and control redesign via model un-falsification: a basic scheme. *International Journal of Control*, 72:887–903, 1999.

D. Wang and C. Cheah. An iterative learning control scheme for impedance control of robotic manipulators. *The International Journal of Robotics Research*, 17(10), Oct 1998.

D. Wang and M. Sun. Anticipatory iterative learning control for non-linear systems with arbitrary relative degree. *IEEE Transactions on Automatic Control*, 46(5), May 2001a.

D. Wang and M. Sun. Sampled-data iterative learning control for nonlinear systems with arbitrary relative degree. *Automatica*, 37, 2001b.

D. Wang. A simple iterative learning controller for manipulators with flexible joints. *Automatica*, 31(9), 1995.

D. Wang. Convergence and robustness of discrete time nonlinear systems with iterative learning control. *Automatica*, 34(11), 1998.

D. Wang. On anticipatory iterative learning control designs for continuous time nonlinear dynamic systems. In *Proceedings of the 38th Conference on Decision and Control*, December 1999.

D. Wang. On d-type and p-type ilc designs and anticipatory approach. *International Journal of Control*, 73(10), 2000.

Y. Wang and R. W. Longman. Use of non-causal digital signal processing in learning and repetitive control. *Advances in the Astronautical Sciences*, 90:649–668, 1996.

S-L. Wirkander and R. W. Longman. Limit cycles for improved performance in self-tuning learning control. *Advances in the Astronautical Sciences*, 154:763–773, 1999.

J-X. Xu and R. Yan. Fixed point theorem based iterative learning control for ltv systems with input singularity. In *Proceedings of the American Control Conference*, Denver, Colorado, June 2003.

J. Xu, Y. Chen, T. Lee, and S. Yamamoto. Terminal iterative learning control with an application to rtpcvd thickness control. *Automatica*, 35, 1999.

J. Xu, Q. Hu, T. Lee, and S. Yamamoto. Iterative learning control with smith time delay compensator for batch processes. *Journal of Process Control*, 11, 2001.

J. Xu and Z. Qu. Robust iterative learning control for a class of nonlinear systems. *Automatica*, 34(8), 1998.

J. Xu, B. Viswanathan, and Z. Qu. Robust learning control for robotic manipulators with an extension to a class of non-linear systems. *International Journal of Control*, 73(10), 2000.

J. Xu and B. Viswanathan. Adaptive robust iterative learning control with dead-zone scheme. *Automatica*, 36, 2000.

J. Xu. Analysis of iterative learning control for a class of nonlinear discrete-time systems. *Automatica*, 33(10), 1997.

J. G. Zeigler and N. B. Nichols. Optimum settings for automatic controllers. *Transactions of the ASME*, pages 759–768, November 1942.