Discrete-time model-based Iterative Learning Control: stability, monotonicity and robustness

Thomas James Harte

PhD in Automatic Control and Systems Engineering

Department of Automatic Control and Systems Engineering

University of Sheffield

January 2007

Abstract

In this thesis a new robustness analysis for model-based Iterative Learning Control (ILC) is presented. ILC is a method of control for systems that are required to track a reference signal in a repetitive manner. The repetitive nature of such a system allows for the use of past information such that the control system iteratively learns control signals that give high levels of tracking. ILC algorithms that learn in a monotonic fashion are desirable as it implies that tracking performance is improved at each iteration. A number of model-based ILC algorithms are known to result in a monotonically converging tracking error signal. However clear and meaningful robustness conditions for monotonic convergence in spite of model uncertainty are lacking.

This thesis gives new robustness conditions for monotonically converging tracking error for two-model based ILC algorithms: the inverse and adjoint algorithms. It is found that the two algorithms can always guarantee robust monotone convergence to zero error if the multiplicative plant uncertainty matrix satisfies a matrix positivity requirement. This result is extended to the frequency domain using a simple graphical test.

The analysis further extends to a Parameter Optimal control setting where optimisation is applied to the inverse and adjoint algorithms. The results show a increased degree in robust monotone convergence upon a previous attempt to apply optimisation to the two algorithms.

This thesis further considers the case where the multiplicative plant uncertainty fails to satisfy the positivity requirement. Robustness analysis shows that use of an appropriately designed filter with the inverse or adjoint algorithm allows a filtered error signal to monotonically converge to zero.

Acknowledgements

I would like to thank Professor David Owens and Dr Jari Hätönen for their patience and supervision during my period of study. Their insistance upon analytical rigour is hopefully evident in this thesis. I would like to thank the EPSRC for their funding. On a personal level I would like to thank my family and friends for their endless patience and support.

Sheffield, United Kingdom

January 2007 Thomas J Harte

Nomenclature

Uppercase Letters

A, B, C, D	system matrices for a discrete-time state-space representation
G	a plant matrix representation
I	identity operator
J	performance criterion
L	learning operator
T_f	trial length
N	number of samples in each trial

Lowercase Letters

$e(t)$	tracking error
k	trial index
l_2	space of square-summable sequences
$r(t)$	reference signal
t	time
$u(t)$	input function
$x(t)$	state
x_0	initial condition for a state-space description
$y(t)$	output

Miscellaneous

β	learning gain
$\lvert\cdot\rvert$	modulus
$\lVert\cdot\rVert$	norm
$\square$	indicates the end of a proof
$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the field of real numbers
$\mathbb{C}$	the field of complex numbers
M^{T}	the transpose of a matrix M
$\lambda(M)$	an eigenvalue of a matrix M
$\sigma(M)$	a singular-value of a matrix M

Abbreviations

2D	two-dimensional
ILC	Iterative Learning Control
NOILC	Norm-Optimal Iterative Learning Control
POILC	Parameter-Optimal Iterative Learning Control
SISO	single input, single output
MIMO	multiple input, multiple output
PI	proportional plus integral

Contents

Chapter 1

Introduction

1.1 Control and Iterative Systems

1.1.1 Feedback Control

Control of dynamical systems can be loosely termed as the manipulation of input signals to attain desirable system outputs. Control systems have been regularly employed to perform this task. One type of control system commonly found in control theory and practice is the feedback controller. A feedback controller assesses the error in the system output and accordingly alters the system input signal in order to improve the system's performance. Consider the following discrete-time, linear time-invariant (LTI), single-input single-output (SISO) system.

$$\begin{aligned} x(t+t_s) &= Ax(t) + Bu(t) \qquad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.1}$$

where A,B,C and D are matrices of appropriate dimensions, $u(\cdot)$ is an input variable, $x(\cdot)$ is a state variable, x_0 is its initial condition, $y(\cdot)$ is an output variable and t_s is the sampling interval. Throughout this thesis it shall be

assumed that $D = 0$ because it is rare that the output $y(t)$ for a physical system is directly effected by the input $u(t)$ without delay.

The role of the feedback controller typically falls into one of two modes: control for the regulation problem and control for the tracking problem. For the regulation problem a controller must be found that holds the output of (1.1), $y(t)$, at a set value despite being subjected to unknown disturbances and unknown system parameters. For the tracking problem the input signal $u(t)$ must be manipulated so that the output $y(t)$ follows a reference signal $r(t)$.

To tackle both problems the feedback controller measures the output error $e(t) = r(t) - y(t)$ and changes $u(t)$ according to some given rule that aims to reduce $e(t)$ until it is as small as possible. The design of this given rule, also known as the *control algorithm*, has been subject to vast quantities of research work resulting in a wide range of feedback laws and design methodologies. Some of these results include the classical PID controller and more modern feedback methods such as Adaptive and Robust control and have been implemented with success in manufacturing industries, chemical processing industries, food industries and automotive industries to name but a few.

1.1.2 Motivation for Iterative Learning

Suppose a system is run in an iterative manner where the system output is required to track a reference signal $r(t)$ over a fixed interval $t \in [0, T_f]$. At the end of each iteration the system is reset to an initial condition $x(0) = x_0$ and the system then tracks $r(t)$ once again. A system that operates in a such a repetitive manner may appear artificial but many industrial systems work in this way. In (Arimoto *et al.*, 1984), (Arimoto *et al.*, 1985), (Arimoto, 1991),

(Bondi *et al.*, 1988), (Casalino and Bartolini, 1984), (Mita and Kato, 1985) and (Zilouchian, 1994) the authors look at robotic control systems where the same tracking task is to be performed repeatedly. In (Lee *et al.*, 1996) the authors do the same for control in chemical batch processing and in (Lee and Lee, 1993) the authors look at control of servo systems that operate in an repetitive manner.

Should a feedback controller result in a non-zero tracking error over the fixed interval then this non-zero error is repeated for each and every iteration. This is because the feedback controller, and hence the control input signal $u(t)$, does not vary from iteration to iteration. This point is illustrated in the following example.

Example 1.1 *Consider the discrete-time, LTI SISO system*

$$(q^2 - 0.5q)y(t) = (0.5q + 0.25)u(t) \tag{1.2}$$

where q^{-1} is a delay operator of one sample. The sampling interval for the system is $t_s = 0.1$. The plant is required to track a reference signal $r(t) = sin\left(\frac{2\pi t}{T_f}\right)$ where $T_f = 10$ and $t \in [0, T_f]$.

A Proportional plus Integral (PI) control law is used to track $r(t)$ such that

$$u(t) = K_i t_s \sum_{i=0}^{t} e(i) + K_p e(t) \tag{1.3}$$

where $K_p = 0.4$ and $K_i = 1.5$ Figure 1.1 shows that the PI controller tracks $r(t)$ with a modicum of success. The inaccuracy in tracking is repeated for all iterations since the control parameters are fixed for each and every iteration.

The repeated tracking error of a feedback controller motivates the following question:

Can the control scheme be iteratively modified so as to improve the tracking performance from iteration to iteration?

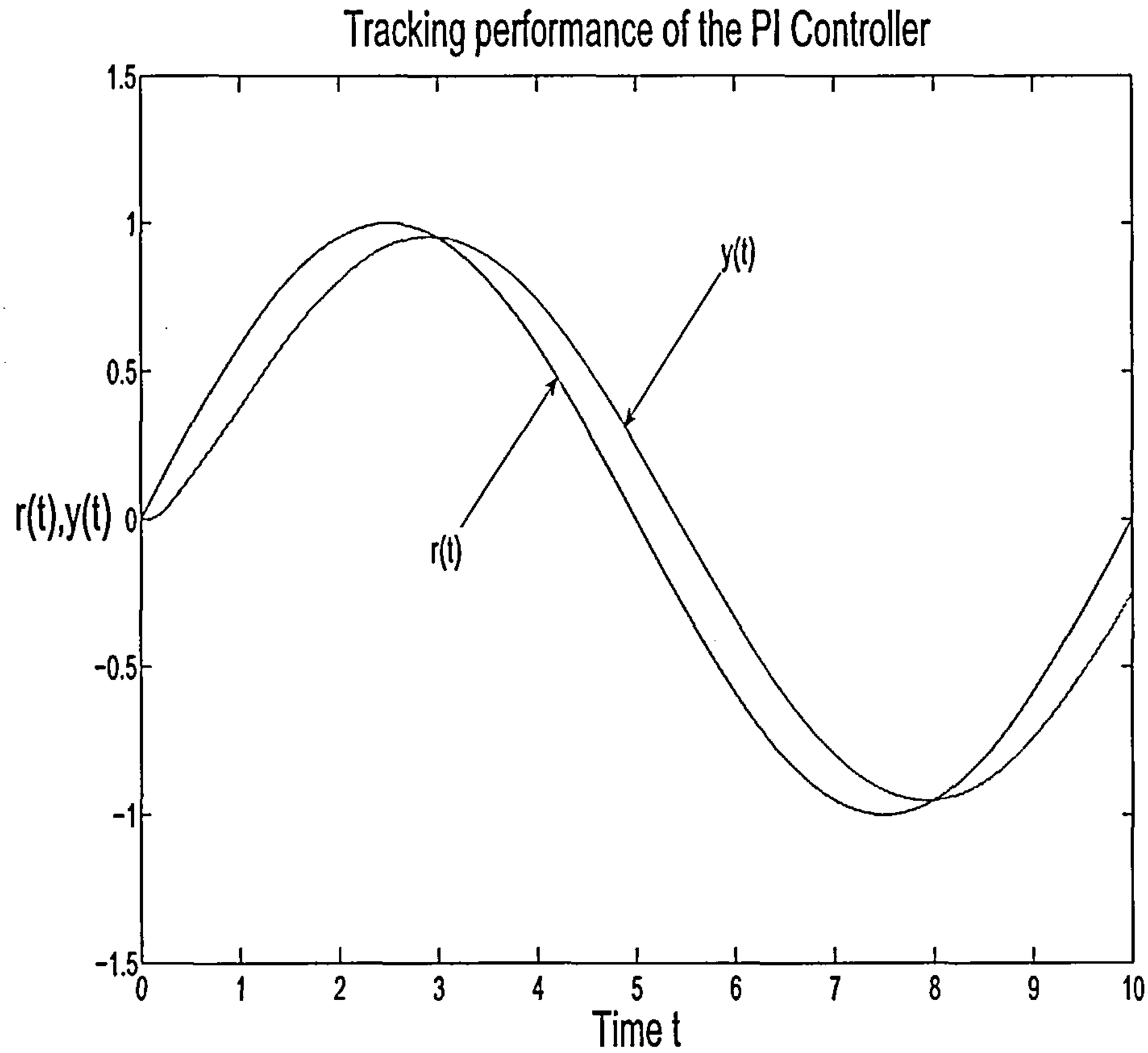

Figure 1.1: Tracking performance of the PI controller

One approach to this question is to apply *inverse dynamics* at each iteration, see (Spong and Vidyasagar, 1989) for an application of inverse dynamics for robot control. If a system representation can be found then the dynamics of the plant can be inverted to give an input signal that induces the desired output. Clearly for this approach to work the plant dynamics must be known exactly otherwise perfect tracking cannot be achieved. In the case where the plant is not perfectly known then system identification can be applied at each iteration. However this assumes that the structure of the system is known

and furthermore if the identification process is flawed to some degree then the inverse dynamics approach will still result in a repeated non-zero tracking error.

Another approach to the previous question is to ask a further question:

Can error signals from previous iterations be used to manipulate the control input so as to improve tracking performance for the current iteration?

The idea here is to iteratively learn the the correct input signal rather than learn the dynamics of the plant explicitly. This approach has come to be known as Iterative Learning Control (ILC).

1.1.3 A brief History of the origins of ILC

The first occurrence of ILC appears to be in an US patent application for *Learning control of actuators in control systems* (Garden, 1971). The idea in the patent is to store a control signal in a memory device and to iteratively update the control signal using the error between the desired output of the actuator.

The first academic appearance of ILC is in (Uchiyama, 1978). However since the publication was published in Japenese only it wasn't until (Arimoto *et al.*, 1984) that ILC became known to non-Japanese researchers. It was at this point that ILC became an research area in a *global* sense and as a result much of the ILC literature refers to (Arimoto *et al.*, 1984) as the starting point for ILC.

The origins of ILC research tended to be centred around work into robotics where repetitive tasks occur naturally. A typical task is a pick and place routine for a robotic arm acting upon a series of units on an assembly line. Examples of work on ILC for robotics are (Arimoto *et al.*, 1984), (Arimoto *et al.*, 1985), (Arimoto, 1991), (Bondi *et al.*, 1988), (Casalino and Bartolini,

1984), (Mita and Kato, 1985) and (Zilouchian, 1994). Surveys on ILC are present in (Moore, 1993), (Moore, 1999) and (Horowitz, 1993) and give a more detailed account on the origins of ILC.

1.1.4 Example implementation of ILC

Example 1.2 *Consider the dynamical system*

$$(q^2 - \alpha q)y(t) = (\beta q + \gamma)u(t) \tag{1.4}$$

where $\alpha = 0.5, \beta = 0.5$ *and* $\gamma = 0.25$. *The sampling interval is* $t_s = 0.1$. *The plant is required to track a reference signal* $r(t) = sin\left(\frac{2\pi t}{T_f}\right)$ *where* $T_f = 10$ *and* $t \in [0, T_f]$. *Note that this system, and its tracking requirement, is the same as that given in Example 1.1. The following iterative learning control law is implemented*

$$u_{k+1}(t) = u_k + he_k(t + t_s) \tag{1.5}$$

where h, *the learning gain, is given as* $h = 1.4$ *and* k *denotes the iteration round (or trial). This ILC law is often referred to as the Arimoto algorithm as it appeared in (Arimoto* et al., *1984) in a continuous-time context where* $u_{k+1}(t) = u_k(t) + h\dot{e}_k(t)$. *Note that the algorithm in (1.5) appears to be non-causal since* $u_{k+1}(t)$ *is a function of* $e_k(t+t_s)$. *However since the signal* $e_k(t)$ *for* $t \in [0, T_f]$ *is available for iteration* $k + 1$ *it is always possible to make* $u_{k+1}(t)$ *a function of* $e_k(t + p)$ *for some* $p > 0$.

Figure 1.2 shows that as $k \to \infty$ *the* l_2*-norm of the error signal* $e_k(t)$ $t \in [0, T_f]$ *converges towards zero. It can be shown that if, assuming* $\beta \neq 0$, *the learning gain* h *satisfies* $|1 - h\beta| < 1$ *then the error converges to zero as* $k \to \infty$ *(see the spectral radius condition in Chapter 2 for the origins of this result). The convergence of the* l_2*-norm is asymptotic meaning that for this example tracking performance gets worse before it gets better.*

Figure 1.3 shows the resulting tracking after 60 *iterations. The output* $y_{60}(t)$ *is visually indistinguishable from the reference* $r(t)$ *and marks a vast improvement upon the tracking of the PI controller used in Example 1.1.*

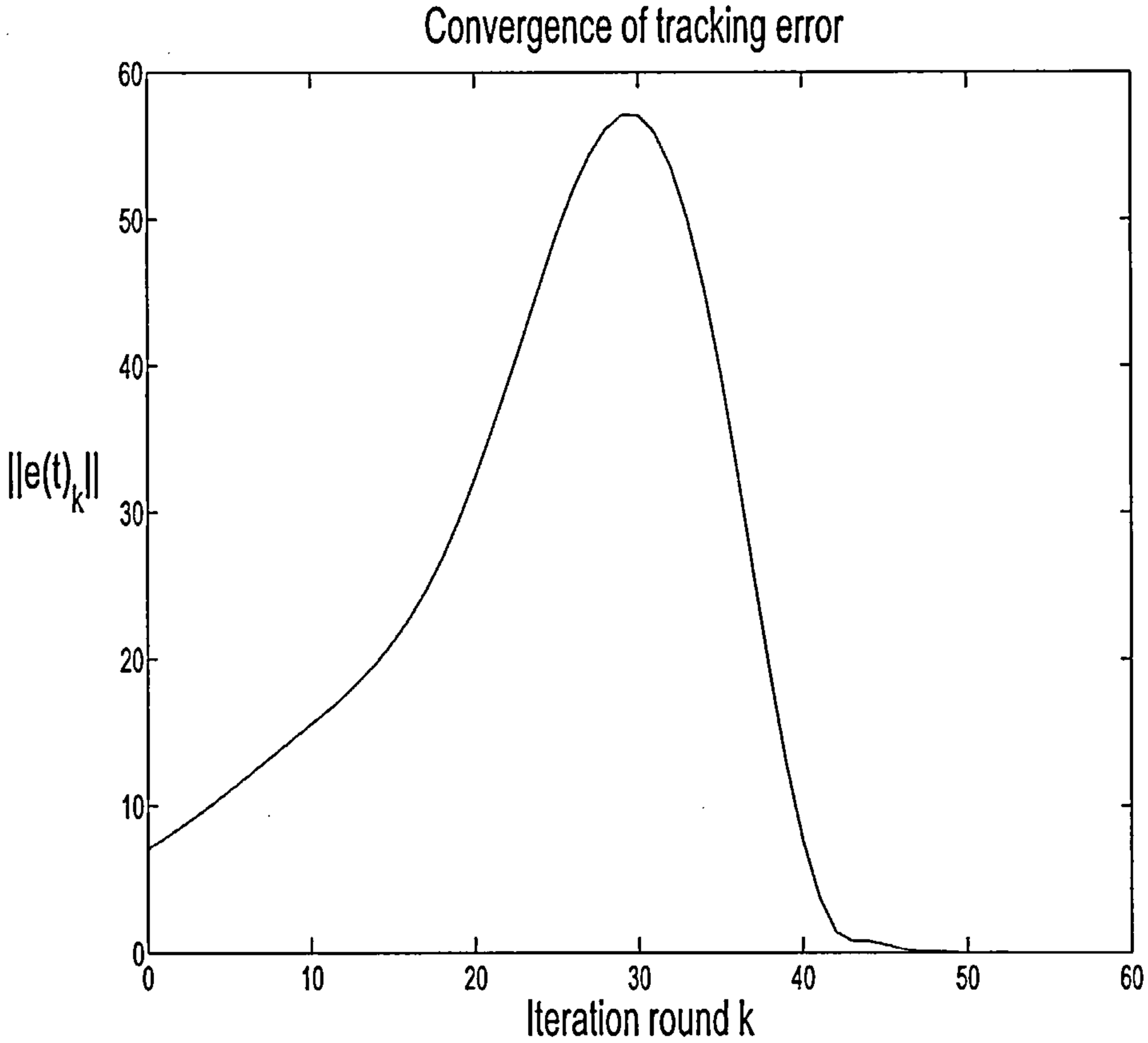

Figure 1.2: Convergence of l_2-norm of the tracking error using the Arimoto algorithm

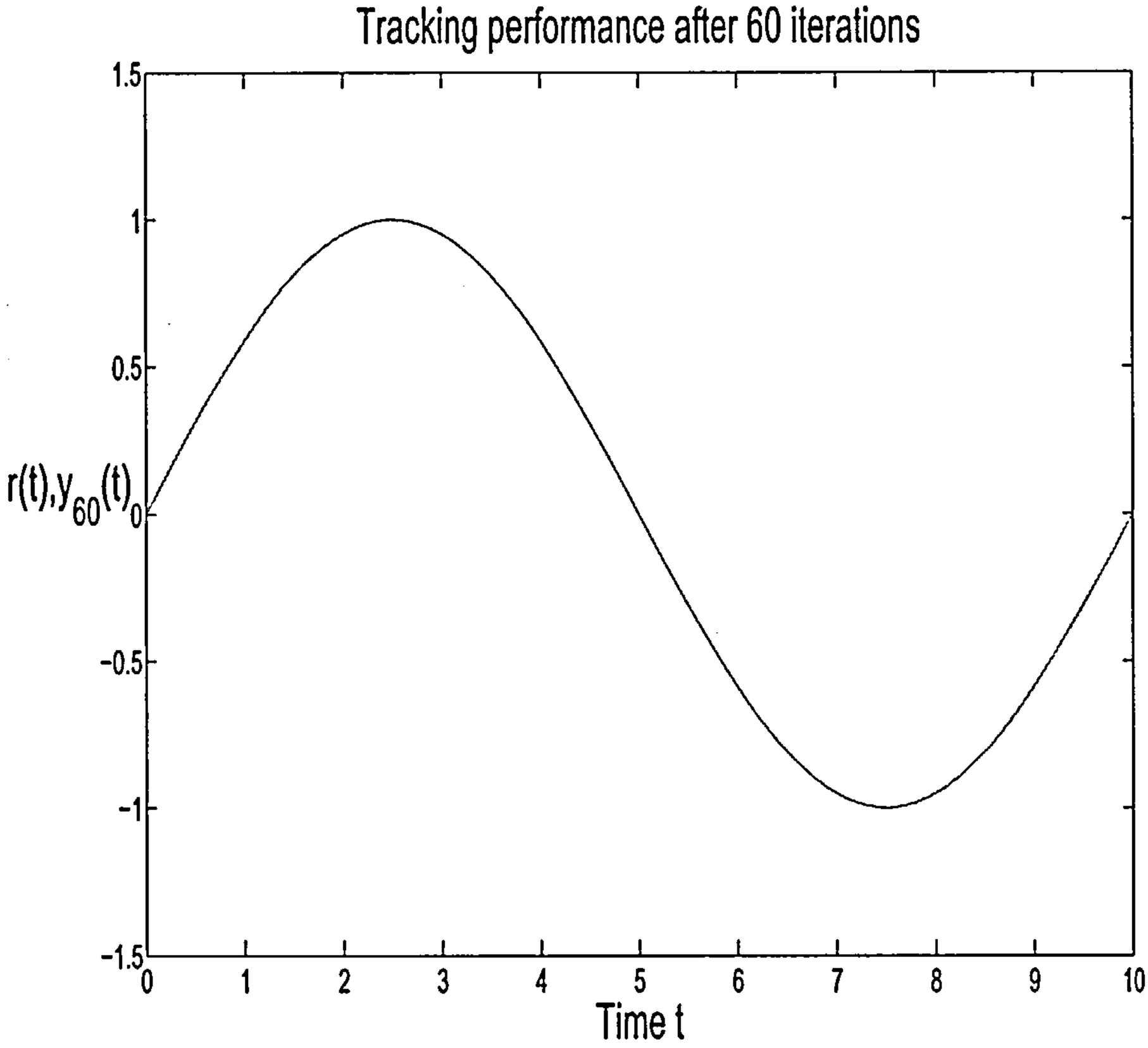

Figure 1.3: Tracking performance after 60 iterations

1.2 ILC Problem Definition

As a starting point consider a standard discrete-time, linear, time-invariant single-input, single-output state-space representation defined over a *finite*, *discrete* time interval, $t \in [0, N]$ (in order to simplify notation it is assumed that the sampling interval, t_s, is unity, hence the number of samples over time interval is $N = T_f$). The system is assumed to be operating in a repetitive mode where at the end of each repetition, the state is reset to a specified initial condition for the next operation during which a new control signal can

be used. A reference signal $r(t)$ is assumed to be specified and the ultimate control objective is to find an input function $u^*(t)$ so that the resultant output function $y(t)$ tracks this reference signal $r(t)$ *exactly* on $[0, N]$. The process model is written in the form:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1.6}$$

where the state $x(\cdot) \in \mathbb{R}^P$ (for some $P \geq 1$), output $y(\cdot) \in \mathbb{R}$, input $u(\cdot) \in \mathbb{R}$ and the initial state $x(0) = x_0$. The operators A, B and C are matrices of appropriate dimensions. From now on it will be assumed that $CA^jB \neq 0$ for some $j \geq 0$ (trivially satisfied in practice) and that the system (1.6) is both controllable and observable. Also, the notation $f_k(t)$ will denote the value of a signal at time t on iteration k.

The repetitive nature of the problem opens up the possibility of iteratively modifying the input function $u(t)$ so that, as the number of repetitions increases, the system asymptotically learns the input function that gives perfect tracking. To be more precise, the control objective is to find a recursive control law

$$u_{k+1} = f(u_k, u_{k-1}, \ldots u_{k-r}, e_{k+1}, e_k, \ldots, e_{k-s}) \tag{1.7}$$

with properties, independent of the control chosen for the first trial, such that:

$$\lim_{k\to\infty} \|e_k\| = 0 \quad \lim_{k\to\infty} \|u_k - u^*\| = 0 \tag{1.8}$$

Here the recursion is expressed in the *super-vector* form with $\|\cdot\|$ as a suitable norm and $u_k = [u_k(0)\ u_k(1)\ \ldots\ u_k(N)]^T$, $y_k = [y_k(0)\ y_k(1)\ \ldots\ y_k(N)]^T$, $e_k = [r(0) - y_k(0)\ r(1) - y_k(1)\ \ldots\ r(N) - y_k(N)]^T$. Furthermore, u^* is the input sequence (in time series or supervector form) that gives $r(t) = [\tilde{G}u^*](t)$ and $\tilde{G}$ is the convolution mapping corresponding to (1.6). Note that if the

mapping f in (1.7) is not a function of e_{k+1}, then it is typically said that the algorithm is of *feedforward* type, otherwise it is of *feedback plus feedforward* type unless it depends only on e_{k+1} when it is of *feedback* type.

For analysis it is important to notice that because the system (1.6) is defined over a finite time-interval, it can be represented equivalently as a matrix equation $y_k = Gu_k + d$, where G has the lower triangular band structure $(G)_{ij} = (G)_{(i+1)(j+1)}$ for $1 \leq i, j \leq N-1$ typical of linear time-invariant systems, i.e.

$$G = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \dots & \dots & 0 \end{bmatrix} \tag{1.9}$$

where $d = [Cx_0, CAx_0, ..., CA^N x_0]^T$ and the elements CA^jB of the matrix G are the Markov parameters of the plant (1.6). This matrix description of the plant dynamics is commonly used in discrete-time analysis of ILC (see (Moore, 1999) and (Norrlöf and Gunnarsson, 2002) for example). Assume from now on that the plant transfer function $G(z) = C(zI - A)^{-1}B$ has relative degree (pole-zero excess) p and hence that the Markov parameters satisfy the conditions $CA^{j-1}B = 0$ for $1 \leq j < p$ with $CA^{p-1}B \neq 0$. Assume also that the reference signal $r(t)$ satisfies $r(j) = CA^jx_0$ for $0 \leq j < p$ (or, alternatively, that tracking in this interval is not important). Then it can be shown (in a similar manner to (Hätönen *et al.*, 2003*a*)) that for analysis it is sufficient to analyse a *lifted* plant equation $y_{k,l} = G_l u_{k,l}$ where

$u_{k,l} = [u_k(0)\ u_k(1)\ \dots\ u_k(N-p)]^T$, $y_{k,l} = [y_k(p)\ y_k(2)\ \dots\ y_k(N)]^T$,

$$G_l = \begin{bmatrix} CA^{p-1}B & 0 & 0 & \dots & 0 \\ CA^{p}B & CA^{p-1}B & 0 & \dots & 0 \\ CA^{p+1}B & CApB & CA^{p-1}B & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-p}B & CA^{N-p-1}B & \dots & \dots & CA^{p-1}B \end{bmatrix} \tag{1.10}$$

and d is replaced by $d = [CA^p x_0, ..., CA^N x_0]^T$. Note that, because it was assumed that $CA^{p-1}B \neq 0$, the matrix G_l is invertible which confirms that, for an arbitrary reference r, there exists u^* such that $r = G_l u^* + d$. From now on this lifted plant model will be used as a starting point for analysis, and in order to simplify notation, the subscript l will be omitted, p will be assumed to be unity and the initial condition x_0 taken, without loss of generality, to be zero.

1.3 Publications Related to this Thesis

This thesis is based upon the following publications:

[P1] T.J. Harte, J. Hätönen and D.H. Owens, A new robust inverse-type ILC algorithm, in *Proc. of the IFAC Workshop on Periodic Control Systems (PSYCO04)*, Yokohama, Japan, 2004.

[P2] T.J. Harte, J. Hätönen and D.H. Owens, Discrete-time Inverse Model-based Iterative Learning Control: Stability, Monotonicity and Robustness, in *International Journal of Control* 78(8): pp. 577-586.

[P3] J. Hätönen, T. Harte, D.H. Owens, J. Ratcliffe and P. Lewin, A new robust Iterative Learning Control algorithm for application on a gantry

robot, in *Proc. of the IEEE conference on Emerging Technologies in Factory Automation*, Lisbon, Portugal, 2003

[P4] J. Hätönen, T. Harte, D.H. Owens, J. Ratcliffe, P. Lewin and E. Rogers, Iterative learning control - what is it all about? in *Proc. of the IFAC Workshop on Periodic Control Systems (PSYCO04)*, Yokohama, Japan, 2004.

Publication P1 presents a robustness analysis for an inverse model-based ILC algorithm. The paper is concerned with robust monotone convergence to zero tracking error when an inverse model contains uncertainty. Theoretical analysis shows it is necessary that the plant uncertainty must satisfy a positivity condition and that a tuning parameter must be made sufficiently small in magnitude. The result shows that an inverse type approach to ILC, contrary to past belief, offers a well defined degree of robustness. Chapter 4 is an extension of this paper where positivity conditions are developed for a generic feedforward type ILC algorithm and robustness conditions for model-based inverse and adjoint algorithms are presented.

Publication P2 again considers robustness issues of an inverse model-based algorithm and can be thought of as an extension of the analysis presented in P1. The major result of the paper is the extension of matrix conditions given in P1 into the frequency domain. In this paper the uncertainty in the plant is considered to be a transfer function in the frequency domain. It is shown that if the plant uncertainty satisfies a positive-real condition then the use of a sufficiently small learning gain guarantees robust monotone convergence to zero tracking error. The condition is given in the form of a relationship between the plant uncertainty and the learning gain which is interpreted in terms of a simple Nyquist plot. The result is of interest because it takes conditions usually described by matrices and represents them in a

domain commonly utilised in control systems theory. Chapter 5 presents this analysis and reports a similar result for an adjoint-type ILC algorithm.

Publication P3 presents a robustness analysis of an optimisation based adjoint ILC algorithm. In this paper analysis shows that the algorithm lacks a method of balancing convergence speed and robustness. A modification of the algorithm is proposed. Consequently a robustness analysis shows that if the plant uncertainty satisfies a positivity condition and a tuning parameter is made sufficiently large then the modified algorithm results in robust monotone convergence to zero tracking error. This marks a major improvement upon the standard algorithm. In the paper the algorithm is applied on an industrial scale multi-axis gantry robot. Experimental trials demonstrate that the algorithm results in near perfect tracking after 100 iterations where a poor guess is made for the initial input function. Chapter 6 is an extension of this publication and gives similar results for an inverse-model version of the algorithm.

Publication P4 discusses the importance of positivity conditions for robustly monotone converging ILC. The paper also proposes Discrete-Fourier Transform (DFT) versions of two ILC algorithms: the inverse and adjoint model based algorithms. A robustness analysis shows that the algorithms result in robust monotone convergence if the plant uncertainty satisfies a positivity condition and a learning gain is made sufficiently small. The paper highlights the potential of DFT based algorithms to relax the positivity conditions for robustness. Chapter 7 is an extension of this paper and presents results for algorithms using a generic set of basis functions (Discrete Fourier Transforms use a specified set of basis functions).

1.4 Overview of the Thesis

The remainder of this thesis is divided into seven chapters. Each chapter starts with a short introduction that motivates the work presented in that chapter. Each chapter concludes with a summary section that reviews the results developed in the chapter and puts them in context with previous chapters and ILC in general. As a result the reader should be able to attain an understanding of the main of results of this thesis without reading it in its entirety. Chapters 3 to 7 give simulations, performed in Matlab, that illustrate the key results presented in the chapters.

Chapter 2 introduces the reader to key mathematical techniques that are used to throughout the remainder of the thesis. The chapter concentrates on the convergence properties of an iterative process $v_{k+1} = Lv_k$. Contraction mapping and spectral radius conditions are given so that a sequence of vectors $\{v_k\}$ converges to a vector v_∞ as k approaches ∞.

Chapter 3 reviews optimisation based methods of ILC and more specifically Parameter Optimal methods. The chapter concentrates on such methods because the ILC literature has a vast and wide range of approaches, for example see the survey paper (Moore, 1999) that gives over 250 references, and optimisation based methods have been identified as having some desirable properties for Iterative Learning Control. These desirable properties are reviewed and some problems with existing optimisation based methods are highlighted. Consequently Chapter 3 can be seen as the motivation for the work presented in the remainder of the thesis.

Chapter 4 concentrates on the convergence and robustness properties of two model-based algorithms termed the inverse and adjoint algorithms. Initially a generically structured feedforward ILC algorithm is considered and conditions for monotone convergence to zero tracking error are developed.

Following this, the inverse and adjoint algorithms, which fit this generic feedforward structure, are then analysed. Necessary conditions for robust monotone convergence are then given in terms of the plant multiplicative uncertainty. The robustness analysis in this chapter is a marked improvement upon that of the parameter optimal methods reviewed in Chapter 3.

Chapter 5 again considers the robustness properties of the inverse and adjoint algorithms analysed in Chapter 4. The matrix-based conditions developed in Chapter 4 are extended into the frequency domain. The analysis provides a simple Nyquist plot test for robust monotone convergence, the test being that the Nyquist plot of the plant multiplicative uncertainty must lie in the right-half complex plane. The results in this chapter are a useful addition to those of Chapter 4 because it takes the abstract descriptions of systems over a finite-time interval, inherent with ILC, and translates them into the frequency domain context commonly used in control theory.

Chapter 6 takes the inverse and adjoint algorithms presented in Chapter 4 and puts them into the context of the POILC method reviewed in Chapter 3. The analysis is done in comparison with an existing gradient based optimisation method for the adjoint algorithm. The robustness analysis shows that use of POILC with the inverse and adjoint algorithms guarantees robust monotone convergence given that the plant uncertainty satisfies a positivity condition and a tuning parameter is made sufficiently large. The gradient based optimisation method however gives no guarantee of convergence even if the plant uncertainty meets the positivity condition.

Chapter 7 introduces a basis function approach to filtering the inverse and adjoint algorithms presented in Chapter 4. The results show that if the robustness conditions developed in Chapter 4 are not held then stability can be retained by use of appropriate filtering but at a loss of zero tracking error.

The loss of zero-tracking error need not be of significant detriment since it turns out that a filtered error signal can be made to monotonically converge to zero in spite of model uncertainty.

Chapter 8 draws conclusion upon the results presented in the thesis and suggests directions for future research work.

1.5 Reading Instructions

i) Chapters 2, 3 and 4 are self contained and can be read separately by those familiar with ILC. However the analysis presented in Chapters 5, 6 and 7 relates heavily to and builds upon the work presented in Chapters 3 and 4. Consequently the reader is advised to read Chapters 3 and 4 prior to Chapters 5, 6 and 7 in order to follow the proof of results in the latter chapters. If the reader is just interested in a statement of the results in Chapters 5, 6 and 7 then Chapters 3 and 4 need not necessarily be read.

ii) Where a norm is unspecified the reader should assume the l_2-norm is being considered. This is the case for Chapters 3 to 7.

Chapter 2

Convergence properties of iterative processes

In Chapter 1 the notion of ILC was introduced where a control signal over a fixed time interval is iteratively modified until the plant perfectly tracks a desired trajectory over the fixed time interval. Clearly ILC results in a two-dimensional control system where the two independent axes are the finite time axis $t \in [0, T_f]$ and the infinite iteration axis $k \in \mathbb{N}$.

The finite nature of the time axis has consequence (or lack of it) for the convergence properties of the 2-D system. The output of a finite-dimensional LTI system can never be become unbounded in finite time hence the stability of the ILC system does not play a significant role in convergence analysis.

However the iteration axis is infinite and as the number of iterations increases the output signal $y(t)$ where $t \in [0, T_f]$ either converges or diverges. Whether or not the system converges depends upon how the control signal is modified from iteration to iteration. The remainder of this chapter reviews some useful analysis tools for the convergence of ILC systems.

Example 2.1 *Consider the following ILC algorithm*

$$u_{k+1} = u_k + Ke_k \tag{2.1}$$

where u_{k+1},u_k and e_k are given in the super vector notation presented in section 1.2 and K is an $N \times N$ matrix.

The above ILC control law integrates along the iteration axis until (hopefully) $e_k = 0$. Such a structure allows for an iterative description of the error vector e_{k+1}. This description is obtained by noticing that $e_p = r - Gu_p$ for any iteration p. Applying this relation to both sides of (2.1) gives the following expression for e_{k+1}:

$$e_{k+1} = (I - GK)e_k = Le_k \tag{2.2}$$

From here in equation (2.2) shall be termed the error evolution equation and the $N \times N$ matrix L shall be termed the learning operator.

Clearly the convergence properties of the ILC system in Example 2.1 depend upon the matrix mapping L. The next two sections give two different conditions upon L for convergence.

The first condition is a contraction mapping condition which guarantees that the error signal e_k converges to zero as k reaches infinity for an arbitrary initial error. Furthermore should this condition be met then the tracking error monotonically converges (with respect to a relevant norm) to zero. This property is often desirable as it implies the tracking improves, in some manner, at each and every iteration until it is perfect.

The second condition is the spectral radius condition. If this condition holds then the tracking error converges to zero as k approaches infinity for an arbitrary initial error. However this condition gives no insight into the tracking during intermediate iterations and thus implies that even if the condition holds then tracking performance can get worse before it gets better.

2.1 Contraction Mappings

Consider the matrix mapping $L : \mathbb{R}^N \rightarrow \mathbb{R}^N$ and a norm on $\mathbb{R}^N$ such that $\|\cdot\| : \mathbb{R}^N \rightarrow \mathbb{R}^+$. The matrix mapping L is termed a contraction mapping if

$$\|Lx - Ly\| \leq \alpha\|x - y\| \tag{2.3}$$

for some $x, y \in \mathbb{R}^N$ and a scalar $|\alpha| < 1$. Applying the standard *Schwarz* inequality upon $\|Lx - Ly\|$ yields a sufficient condition for a contraction mapping in terms of the matrix L:

$$\|Lx - Ly\| = \|L(x - y)\| \leq \|L\|\|x - y\| \tag{2.4}$$

where the induced matrix norm of L, $\|L\|$, is further defined as follows:

$$\|L\| = \sup_{x \in \mathbb{R}^N, \|x\|=1} \|Lx\| \tag{2.5}$$

Clearly for the matrix mapping $L : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to be a contraction mapping then it is necessary and sufficient that $\|L\| < 1$.

Consider the ILC error evolution equation in (2.2) where $e_{k+1} = Le_k$. If the learning operator L is a contraction mapping then it implies the following:

$$\|e_{k+1}\| = \|Le_k\| \leq \|L\|\|e_k\| < \|e_k\| \tag{2.6}$$

for all $e_k \neq 0$.

Clearly if the learning operator is a contraction mapping the norm of the tracking error decreases at each iteration, unless of course it has already reached zero. This behaviour is termed monotonic convergence and is a very desirable property for an ILC algorithm since it implies that tracking is always improving rather than producing large and potentially damaging control signals.

The next example demonstrates that whether or not the learning operator is a contraction mapping is dependent upon the norm considered.

Example 2.2 *Consider the matrix mapping* $L : \mathbb{R}^N \to \mathbb{R}^N$. *Three commonly used norms on* $\mathbb{R}^N$ *are the* l_1, l_2 *and* l_∞ *norms defined as follows:*

$$\begin{aligned} \|v\|_1 &= \sum_{i=1}^{N} |v_i| \\ \|v\|_2 &= \sqrt{\sum_{i=1}^{N} |v_i|^2} \\ \|v\|_\infty &= \max_{i \in I} |v_i| \end{aligned} \tag{2.7}$$

where $I = \{1, 2, \ldots, N-1, N\}$, $v \in \mathbb{R}^N$ *and* $v = [v_1, v_2, \ldots, v_N]^T$. *The corresponding matrix norms for* L *become (see (Varga, 1962))*

$$\begin{aligned} \|L\|_1 &= \max_{j \in I} \sum_{j=1}^{N} |L_{ij}| \\ \|L\|_2 &= \overline{\sigma}(L) \\ \|L\|_\infty &= \max_{i \in I} \sum_{i=1}^{N} |L_{ij}| \end{aligned} \tag{2.8}$$

where $\overline{\sigma}(L)$ *is the largest singular value of* L. *Note that often* $\|L\|_1 \neq \|L\|_2 \neq \|L\|_\infty$, *although* $\|L\|_2^2 \leq \|L\|_1 \|L\|_\infty$, *and hence it should always be made clear which norm is being used to describe the convergence properties of the ILC algorithm.*

Note that in this thesis if no norm is specified then the reader should assume that the l_2-norm is being used where $\|v\|^2 = v^T v$ for an arbitrary vector $v \in \mathbb{R}^N$.

Even though it is often desirable for an ILC algorithm to be a contraction mapping it is not always possible to achieve it. In such cases it is still possible to test whether the algorithm converges asymptotically to a fixed point. This test is termed the spectral radius condition and is given in the following section.

2.2 Spectral radius condition

Consider again the matrix mapping $L : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with matrix norm $\|L\|$. The spectral radius of L is defined as $\rho(L)$ where:

$$\rho(L) = \lim_{k \rightarrow \infty} \|L^k\|^{\frac{1}{k}} \tag{2.9}$$

Note that $\rho(L)$ is the same regardless of choice of norm. It can be shown, see (Edwards and Owens, 1982) for example, that if $\rho(L) < 1$ then the ILC error evolution equation (2.2) converges to zero. This condition comes from the following:

$$\lim_{k \rightarrow \infty} \|e_k\| = \lim_{k \rightarrow \infty} \|L^k e_0\| \leq \lim_{k \rightarrow \infty} \|L^k\| \|e_0\| = \lim_{k \rightarrow \infty} \rho^k(L) \|e_0\| \tag{2.10}$$

Note, the last equality requires a rigourous proof. Clearly if $\rho(L) < 1$ then it follows that $\lim_{k \rightarrow \infty} e_k = 0$ for an arbitrary initial error e_0.

It is of worth noting that for an arbitrary $N \times N$ matrix L the spectral radius $\rho(L)$ is given by

$$\rho(L) = \max_{i \in I} |\lambda_i(L)| \tag{2.11}$$

where $I = \{1, 2, \ldots, N-1, N\}$ and $\lambda_i(L), i \in I$ is an eigenvalue of L. See (Varga, 1962) for full details of this property. A consequence of this result is that it is sufficient that the largest eigenvalue of L be inside the complex unit circle to guarantee convergence to zero tracking error.

The spectral radius condition turns out to be a tighter condition for convergence than the contraction mapping condition. In fact if the contraction mapping condition holds then it implies the spectral radius condition also holds, however the reverse is not true. This can best be seen by considering a matrix mapping L and its spectral radius $\rho(L) = |\lambda|$ and further considering the following equation:

$$\lambda v = Lv \tag{2.12}$$

where v is the corresponding eigenvector of λ. Taking the norm of both sides of the above equation and utilising the *Schwarz* inequality leads to the following relations:

$$\begin{aligned} |\lambda|\|v\| &\leq \|L\|\|v\| \\ \rho(L) &\leq \|L\| \end{aligned} \tag{2.13}$$

Clearly if $\|L\| < 1$ then it implies $\rho(L) < 1$ but the reverse is not true. As a consequence of this the spectral radius condition $\rho(L) < 1$ only guarantees asymptotic convergence to zero error. The next example illustrates that even if the initial error is small then a matrix L that satisfies $\rho(L) < 1$ can still generate large error signals during early iterations.

Example 2.3 *Consider the matrix mapping* $L : \mathbb{R}^2 \to \mathbb{R}^2$ *where*

$$L = \begin{bmatrix} \delta & 0 \\ \epsilon & \delta \end{bmatrix} \tag{2.14}$$

The matrix L *has a repeated eigenvalue of* $\lambda(L) = \delta$ *and two* Jordan *block eigenvectors* v_a *&* v_b *such that* $Lv_a = \delta v_a$ *and* $Lv_b = \delta v_b + v_a$. *The two eigenvectors are chosen to be:*

$$v_a = \begin{bmatrix} 0 \\ \epsilon \end{bmatrix} \quad v_b = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.15}$$

Suppose now that for the error evolution equation (2.2) e_0 *is chosen to be* $e_0 = v_b$. *The resulting error vectors* $e_1 = \delta v_b + v_a$ *and* $e_2 = \delta^2 v_b + 2\delta v_a$ *become:*

$$e_1 = \begin{bmatrix} \delta \\ \epsilon \end{bmatrix} \quad e_2 = \begin{bmatrix} \delta^2 \\ 2\delta\epsilon \end{bmatrix} \tag{2.16}$$

Note that if $\delta = 0$, *and hence* $\rho(L) = 0$, *then the process converges to zero in just two iterations. However by increasing* ϵ *the norm of* e_1 *can be made arbitrarily large regardless of* $\rho(L)$.

Note that Example 2.3 has parallels with the Arimoto algorithm presented in Example 1.2. The Arimoto learning operator also has N repeated eigenvalues of $\lambda(L) = 1 - hCB$. Whilst this algorithm has a simple method of satisfying the spectral radius condition (select h such that $0 < hCB < 2$) it can also suffer from large error signals during early iteration rounds.

2.3 Summary

In this chapter the convergence properties of an iterative process $e_{k+1} = Le_k$ were reviewed in the context of two conditions. The first was a contraction mapping that guaranteed that the norm of e_k decreases at each and every iteration. The second condition was a spectral radius condition that guarantees that $\lim_{k\to\infty} e_k = 0$, however it gives no guarantee that the norm of e_k will be suitably small during early iterations.

These two conditions are both of great importance and great use for the analysis of ILC since most algorithms result in an error evolution equation of the form $e_{k+1} = Le_k$. The remainder of this thesis concentrates on ILC algorithms of this form and applies the contraction mapping and spectral radius conditions where possible.

Chapter 3

Parameter Optimal ILC

3.1 Optimisation and ILC

Optimisation and control theory have been utilised together for many years and in numerous scenarios, see (Athans and Falb, 1966),(Anderson and Moore, 1989) and (Owens, 1981) for a formal background on the partnership. It is natural then that the ILC literature includes control schemes that employ optimisation. Some examples can be found in (Xu and Tan, 2002), (Owens and Hätönen, 2005) and (Furuta and Yamakita, 1987). Usually in such scenarios a performance index, or cost function, of a system is generated in terms of a set of parameters available for the controller to manipulate. These control parameters are then set, by some means, to minimise the value of this cost function and hence give optimal control performance. For example in (Furuta and Yamakita, 1987) the authors attempt to minimise the l_2 tracking error for the next trial. Such an optimisation appears an intuitive one for ILC however some different cost functions have been introduced with success. Take (Amann *et al.*, 1996) for example, where the following cost function is used.

$$\min_{u_{k+1}\in\mathbb{R}^N}\{J(u_{k+1}) = \beta\|e_{k+1}\|^2 + \|u_{k+1} - u_k\|^2\} \tag{3.1}$$

In (Amann *et al.*, 1996) the authors have been more concerned with transient performance of the control system, finding that whilst minimising $\|e_{k+1}\|^2$ directly may give fast convergence to zero error it can induce large error signals during the early trials. A trait that in a physical sense might make ILC implausible to implement. The cost function in (3.1) has an extra term that attempts to minimise the change in input from trial to trial as well the future error and hence gives a *cautious controller.*

Minimising (3.1) yields what is now termed the Norm Optimal ILC (NOILC) update law.

$$u_{k+1} = u_k + \beta G^T e_{k+1} \tag{3.2}$$

Convergence analysis of (3.2) gives the following inequality

$$\|e_{k+1}\|^2 \leq \frac{1}{1+\beta\underline{\sigma}(GG^T)}\|e_k\|^2 \tag{3.3}$$

where $\underline{\sigma}(GG^T)$ denotes the smallest singular value of GG^T. If β is chosen to be a positive scalar then convergence is monotonic, i.e. $\|e_{k+1}\|^2 \leq \|e_k\|^2$. This is a very desirable property for any ILC system since the performance is being bettered, in the l_2 sense, at each and every trial. Furthermore it converges to the limit $\|e_\infty\| = 0$.

However the algorithm is not without drawbacks. It is clear that (3.2) is non-causal in the iterative sense and must use state observation and prediction in order to be implemented. Even in the nominal case where perfect plant knowledge is assumed, the implementation has a high degree of complexity thus making any robustness analysis even more so complicated. These two drawbacks lead to a search for other ILC algorithms that were simpler and causal in implementation but maintained monotonic error convergence.

3.2 Derivation and Convergence of Parameter Optimal ILC

Consider the recently introduced simple form of Optimal ILC (Owens and Feng, 2003), termed Parameter Optimal ILC (POILC), where a scalar gain β is varied adaptively with iteration in an *Arimoto-type* algorithm of the following form

$$u_{k+1} = u_k + \beta_{k+1} e_k \tag{3.4}$$

In contrast with the NOILC input sequence (3.2) the input is causal in implementation and hence has a degree more of clarity in analysis.

The authors of (Owens and Feng, 2003) proposed to select β_{k+1} by minimising the following cost function

$$\min_{\beta_{k+1} \in \mathbb{R}} \{J(\beta_{k+1}) = \|e_{k+1}\|^2 + w\beta_{k+1}^2\} \tag{3.5}$$

The cost function (3.5) was motivated by that used in NOILC, see (Amann *et al.*, 1996), and also attempts to minimise the change between inputs from trial to trial whilst reducing the error. Note that $u_{k+1} - u_k = \beta e_k$ and hence minimising β^2 also minimises $\|u_{k+1} - u_k\|$. Minimising (3.5) gives β_{k+1} to be of the following form

$$\beta_{k+1} = \frac{e_k^T G e_k}{w + \|G e_k\|^2} \tag{3.6}$$

A review of the convergence analysis in (Owens and Feng, 2003) follows. Selecting a sub-optimal choice of $\beta_{k+1} = 0$ and noticing that $J(0) = \|e_k\|^2$ yields an inequality that proves monotonic-convergence of the algorithm.

$$\|e_{k+1}\|^2 + w\beta_{k+1}^2 \leq \|e_k\|^2 \tag{3.7}$$

Clearly if $w > 0$ then the error converges monotonically. However it converges to a limit e_∞ where $e_\infty^T G e_\infty = 0$. If $G + G^T$ is a positive-definite matrix then

the only solution for e_∞ is the zero vector. However in many applications such a condition on G cannot be held and so the error will converge to a non-zero solution, which may or may not be tolerable for the application at hand.

3.3 Multi-Parameter Optimal ILC

The discussion in section 3.2, and more formally the analysis in (Owens and Feng, 2003), highlighted that unless the plant matrix G is positive-definite then POILC can, and often in practice does, converge to a non zero error. This section introduces more than one adaptive scalar β_{k+1} and optimizes them all to force monotonic convergence to zero error.

Consider the algorithm:

$$u_{k+1} = u_k + \sum_{i=1}^{P} \beta_{i,k+1} \Lambda_i e_k \tag{3.8}$$

where

$$\Lambda_i = \begin{bmatrix} \lambda_i & 0 & 0 & \dots & 0 \\ \lambda_i^2 & \lambda_i & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_i^N & \lambda_i^{N-1} & \dots & \dots & \lambda_i \end{bmatrix} \tag{3.9}$$

The cost function put forward in (Owens and Feng, 2003) cannot be used directly but can be written in a similar form to optimize each $\beta_{i,k+1}$ and is given as follows

$$\min_{\Upsilon_{k+1} \in \mathbb{R}^N} J(\Upsilon_{k+1}) = \|e_{k+1}\|^2 + \Upsilon_{k+1}^T W \Upsilon_{k+1} \tag{3.10}$$

where W is a positive-definite, diagonal matrix and Υ_{k+1} is defined as below.

$$\Upsilon_{k+1} = \begin{bmatrix} \beta_{1,k+1} \\ \beta_{2,k+1} \\ \vdots \\ \beta_{P,k+1} \end{bmatrix} \tag{3.11}$$

Minimising (3.10) subject to (3.8) yields

$$\Upsilon_{k+1} = (W + H_k^T G^T G H_k)^{-1} H_k^T G^T e_k \tag{3.12}$$

where

$$H_k = [\Lambda_1 e_k, \Lambda_2 e_k, \ldots, \Lambda_P e_k] \tag{3.13}$$

The following proposition shows that Multi-Parameter Optimal ILC, like POILC, gives monotonic convergence but has a different condition for convergence to zero error.

Proposition 3.1 *The algorithm (3.8) where Υ_{k+1} is chosen using (3.12) results in monotonic error convergence. Furthermore if $P > N$ then $\lim_{k\to\infty} e_k = 0$.*

Proof.

Selecting a sub-optimal choice of $\Upsilon_{k+1} = \mathbf{0}$ in the cost function (3.10) yields $J(\mathbf{0}) = \|e_k\|^2$. Since this choice is sub-optimal it follows that

$$\|e_k\|^2 \geq \|e_{k+1}\|^2 + \Upsilon_{k+1}^T W \Upsilon_{k+1} \geq \|e_{k+1}\|^2 \tag{3.14}$$

Since W is a positive-definite matrix the inequality in (3.14) clearly demonstrates monotonic convergence. Reformulation of (3.14) gives

$$\|e_k\|^2 - \Upsilon_{k+1}^T W \Upsilon_{k+1} \geq \|e_{k+1}\|^2 \geq 0 \tag{3.15}$$

and applying induction further gives

$$\|e_0\|^2 - \sum_{j=1}^{k+1} \Upsilon_j^T W \Upsilon_j \geq 0 \tag{3.16}$$

and because k is arbitrary, $\lim_{k\to\infty} \Upsilon_k = 0$. This results in

$$\lim_{k\to\infty} \Upsilon_{k+1} = \lim_{k\to\infty} H_k^T G^T e_k = \lim_{k\to\infty} \begin{bmatrix} e_k^T G \Lambda_1 e_k \\ e_k^T G \Lambda_2 e_k \\ \vdots \\ e_k^T G \Lambda_N e_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.17}$$

Note that the term $e_k^T G \Lambda_i e_k$ can be written as a sum of terms which happen to be an Nth order polynomial of λ_i

$$e_k^T G \Lambda_i e_k = q_N \lambda_i^N + q_{N-1} \lambda_i^{N-1} + \cdots + q_1 \lambda_i^1 = 0 \tag{3.18}$$

where $\{q_1, q_2, \ldots, q_N\}$ is a set of scalars. There exists no more than $N-1$ non-zero solutions for λ_i, hence if $P \geq N$ and $\lambda_i \neq 0 \quad \forall i$ then at least one of the $e_k^T G \Lambda_i e_k$ terms is non-zero for any $e_k \neq 0$, hence $\lim_{k\to\infty} \Upsilon_{k+1}^* = 0$ if and only if $e_k = 0$. It follows that $\lim_{k\to\infty} e_k = 0$. □

Remark 3.1 *The application of POILC required the plant matrix be positive-definite to guarantee zero tracking error. In contrast the multi-parameter form has relaxed this restriction completely, however it is dependent upon the trial length, requiring at least N different λ_i to guarantee zero error. In many applications N is a large number meaning Multi-Parameter Optimal ILC can be computationally intensive.*

3.4 A Robustness Analysis of POILC

This section presents a robustness condition for the POILC algorithm discussed in section 3.2.

The optimal β_{k+1} given in (3.6) is dependent upon the plant matrix G. In most real situations an exact knowledge of this matrix is unavailable and is replaced with an approximate model G_o, where $G = G_o U$ and U represents the multiplicative uncertainty in the plant. The following proposition gives a condition for monotonic convergence when a sub-optimal β_{k+1} is used.

Proposition 3.2 *Suppose POILC is used with a nominal model G_o, then there exists a $w \in \mathbb{R}^+$ that guarantees monotonic error convergence if $sign\{v^T G_o v\} = sign\{v^T G v\}$ for an arbitrary $v \in \mathbb{R}^N$. Furthermore if the plant matrix $G + G^T$ is positive definite then the error converges to zero.*

Proof. Taking the inner product of e_{k+1} with itself yields the following equation:

$$\|e_{k+1}\|^2 = \|e_k\|^2 - 2\frac{e_k^T G_o e_k}{w + \|G_o e_k\|^2} e^T G e_k + \left(\frac{e_k^T G_o e_k}{w + \|G_o e_k\|^2}\right)^2 \|G e_k\|^2 \quad (3.19)$$

Note that the second term on the right hand side of (3.19) is of $O(\beta_{k+1})$ and that the third term is of $O(\beta_{k+1}^2)$. Hence if $sign\{v^T G_o v\} = sign\{v^T G v\} \quad \forall v$ and w is sufficiently large then $\|e_{k+1}\|^2 \leq \|e_k\|^2$. The limit e_∞, to which e_k converges to, is a solution of $e_\infty^T G_o e_\infty = 0$. If $G + G^T$ is a positive-definite matrix and $sign\{v^T G_o v\} = sign\{v^T G v\} \quad \forall\, v$ then the only solution for e_∞ is the zero vector. □

Remark 3.2 *The condition in Proposition 3.2 requires that $sign\{v^T G_o v\} = sign\{v^T G v\}$ for an arbitrary v and cannot be held with great confidence in many applications where $v^T G v$ is sign-indefinite, due to inaccuracies in the model. Proposition 3.2 implies that POILC can confidently be applied to positive-definite systems but for systems outside of this class robustness analysis is not clear.*

3.5 Simulation Examples

In this section simulation examples are given that demonstrate the results developed for Multi-POILC and the robustness of POILC.

3.5.1 Multi-Parameter Optimal ILC simulation example

Consider the case where a real plant $G(s)$ is given in the s-domain to be

$$G(s) = \frac{1}{s^2 + 2s + 1} \tag{3.20}$$

The system is sampled with a sample time of 0.1 seconds using zero-order hold. The trial length T_f is 0.8 seconds hence $N = 8$. It is easy to check numerically that the resulting 8×8 matrix $G + G^T$ is not positive-definite. The reference signal is chosen to be $r(t) = sin(\frac{2\pi}{T_f}t)$. The Multi-POILC algorithm described in section 3.3 is run for 18000 iterations where

$$\lambda_i = \lambda_{i-1} - \frac{\lambda_1}{P} \quad \forall \quad i \in \{2, 3, \ldots, P\} \tag{3.21}$$

where $\lambda_1 = 0.99$,$P = 32$ and $W = 0.001I$.

Figure 3.1 shows that the error monotonically converges. The analysis in section 3.3 suggests that since $P > N$ the error should monotonically converge to zero. Figure 3.1 shows that after 18000 iterations the error is still converging towards zero, however the rate of convergence is slow.

3.5.2 Robustness of POILC simulation example

Consider the case where a real plant $G(s)$ and model plant $G_o(s)$ are given in the s-domain to be

$$G(s) = \frac{2}{s + 2} \tag{3.22}$$

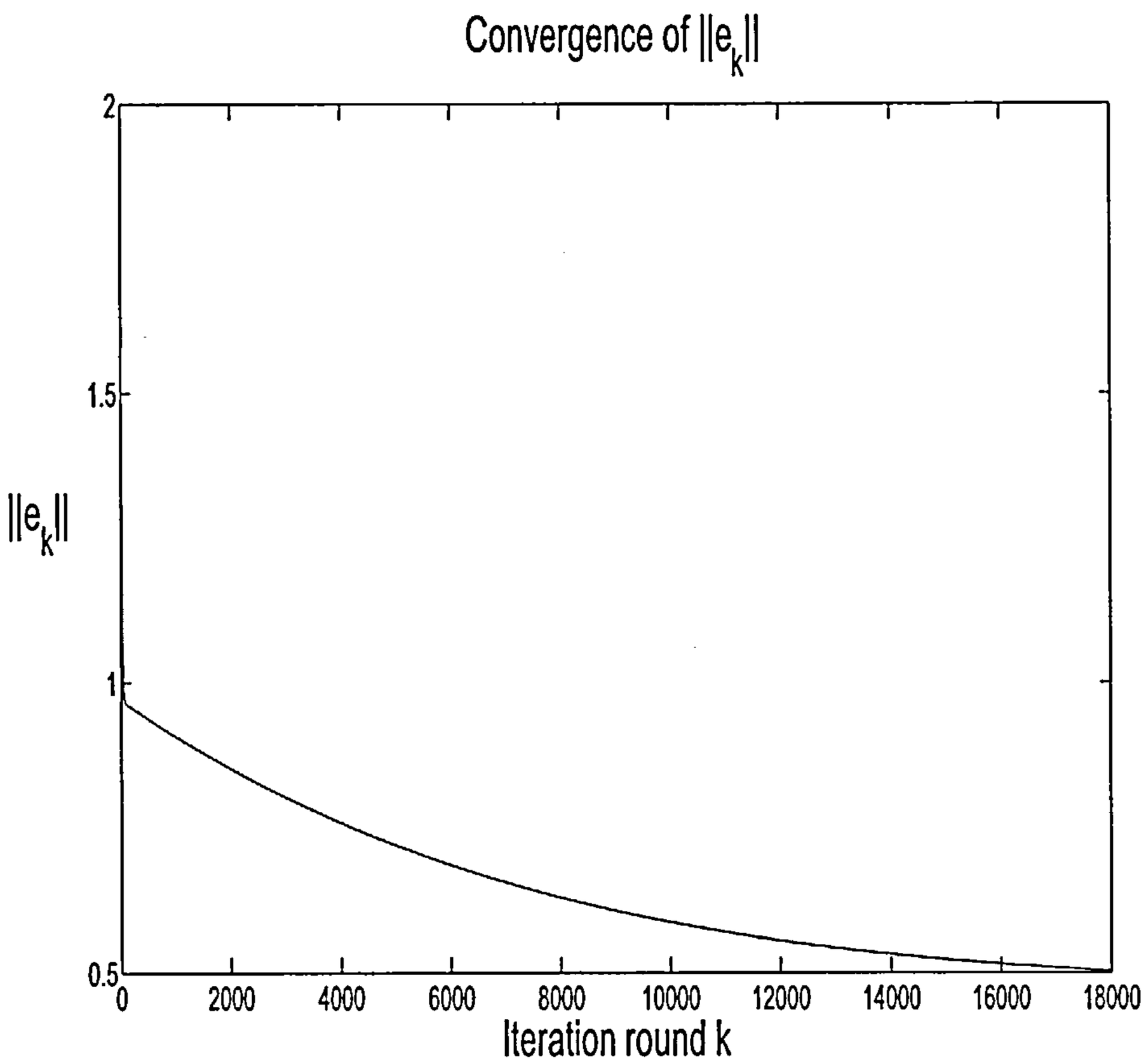

Figure 3.1: Convergence of error for Multi-Parameter Optimal ILC algorithm

$$G_o(s) = \frac{1}{s+1} \tag{3.23}$$

The two systems are sampled with a sample time of 0.1 seconds using zero-order hold. The trial length T_f is 25.5 seconds. It is easy to check numerically that the resulting matrices $G + G^T$ and $G_o + G_o^T$ are both positive-definite. The reference signal is chosen to be $r(t) = sin(\frac{2\pi}{T_f}t)$. The POILC algorithm described in section 3.2 is run for 50 iterations where

$$\beta_{k+1} = \frac{e_k^T G_o e_k}{w + \|G_o e_k\|^2} \tag{3.24}$$

and $w = 0.01$.

Figure 3.2 shows, as the analysis in section 3.4 suggests, that $\|e_k\|$ monotonically converges. It is worth noting that as $\|e_k\|$ gets smaller the rate of convergence slows. This is found to be a common occurrence when utilising POILC.

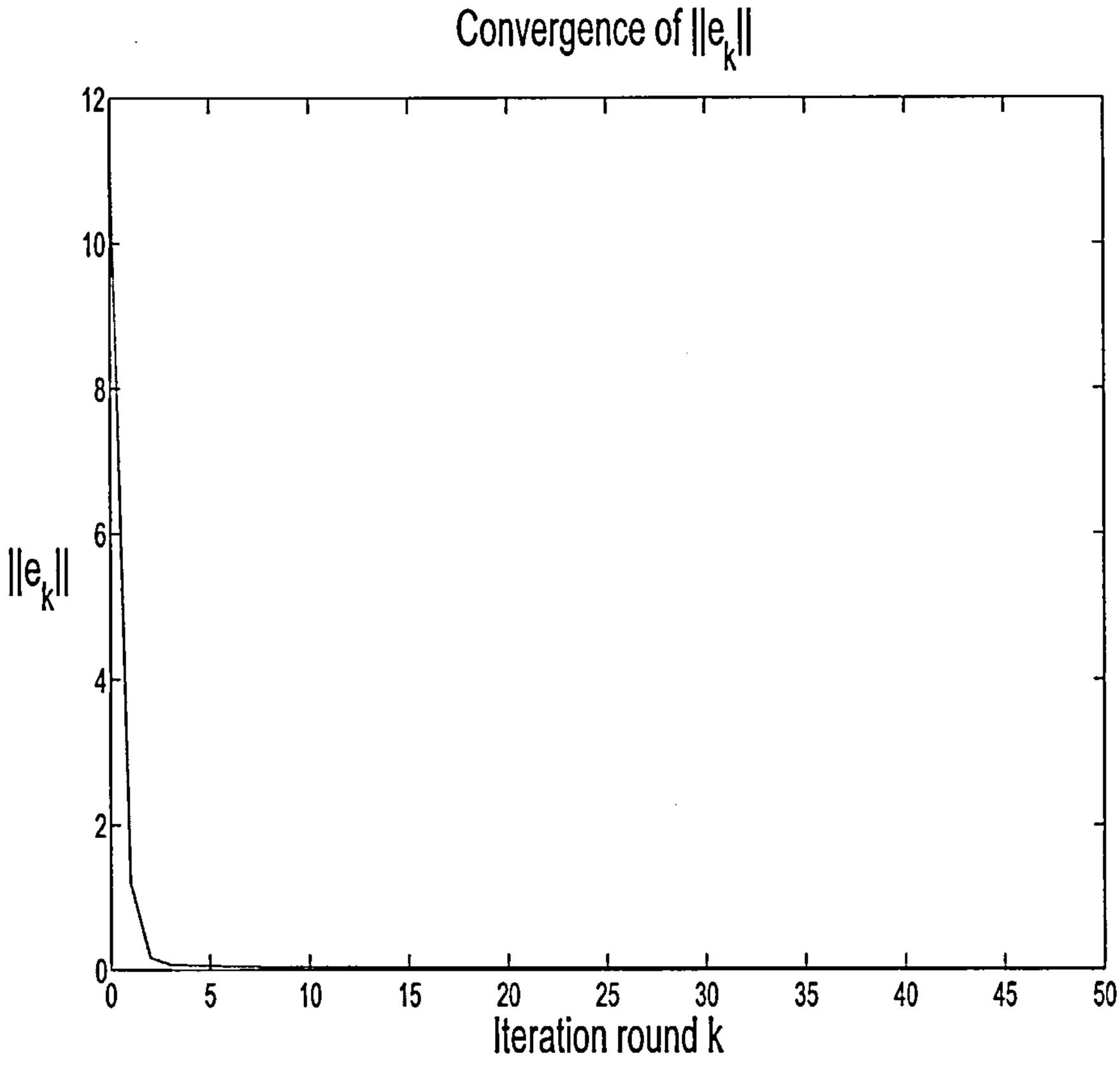

Figure 3.2: Convergence of error for POILC algorithm

3.6 Summary

In this chapter an existing POILC algorithm was reviewed. The algorithm achieves monotonic error convergence if a simple gain is optimised appropriately at each iteration. The algorithm gives convergence to zero error if and only if $G + G^T$ is a positive-definite matrix.

A multi-parameter version of the POILC algorithm was introduced which removes the condition upon $G + G^T$ in order to achieve monotonic error convergence to zero. However, as a simulation demonstrated, a method for selecting the necessary λ parameters is not clear and error convergence tends to be slow.

The previous work on the POILC algorithm lacked a robustness analysis. In this chapter a robustness condition was presented, namely if $sign\{v^T(G + G^T)v\} = sign\{v^T(Go + Go^T)v\}$ for all v then monotonic error convergence is retained. The condition was demonstrated via a simulation example. The condition presented is necessary and appears quite restrictive. It also fails to give a clear relation between the plant uncertainty and the stability of the algorithm. This result motivates the search for simple ILC algorithms that give clear robustness conditions for both stability and performance. The remainder of thesis tackles this subject.

Chapter 4

Time Domain conditions for convergence in ILC: Inverse and Adjoint algorithms

In Chapter 3 it was shown that Parameter Optimal ILC puts a highly restrictive condition upon the plant matrix in order to achieve monotonic convergence to perfect tracking. It was also shown that the robustness analysis of POILC does not provide meaningful conditions upon the plant uncertainty thus requiring the often unrealistic assumption that a perfect plant model is available.

This chapter analyses the stability and transient performance of a generically structured ILC control law, and then focuses on two particular algorithms that fit this structure, namely the *inverse* and *adjoint* algorithms. These two algorithms are model based algorithms and it is pertinent to consider their behaviour when a perfect plant model is unavailable. Therefore the stability and convergence conditions are interpreted in terms of model uncertainties. This robustness analysis forms the major contribution of this

chapter and is presented in publication P1 (Harte *et al.*, 2004). Section 1.3 gives a brief overview of this paper.

4.1 Stability of a feedforward Iterative Learning Controller

Consider a feedforward ILC input of the form

$$u_{k+1} = u_k + \beta K e_k \tag{4.1}$$

where K is an $N \times N$ matrix, β is a scalar and both are chosen by the designer. The error evolution equation corresponding to (4.1) is easily shown to be

$$e_{k+1} = (I - \beta GK)e_k \tag{4.2}$$

Convergence of the error depends upon the matrix βGK. The following proposition gives a condition for asymptotic convergence to zero error in terms of the spectral radius defined in Section 2.2

Proposition 4.1 *A necessary and sufficient condition for convergence of the above feedforward ILC algorithm is that the spectral radius of* $(I - \beta GK)$ *is strictly less than* 1. *An equivalent statement is that it is necessary and sufficient that* GK *only has eigenvalues in the open right half complex plane and that* $\beta > 0$ *has a sufficiently small magnitude.*

Proof. The proposition requires a rigorous proof which can done using matrix analysis or concepts from Multipass Processes and Repetitive Systems Theory and can be found in a similar form within (Edwards and Owens, 1982) and (Rogers and Owens, 1992). For a more intuitive explanation of the

spectral radius condition in (Edwards and Owens, 1982) and (Rogers and Owens, 1992) see Section 2.2 □

4.2 A Matrix Condition for Monotonic Convergence

As discussed in Chapter 2, Chapter 3, (Amann *et al.*, 1996) and (Owens and Feng, 2003) monotonically converging error signals are a desirable trait for any ILC algorithm. This section develops a condition for monotonic convergence of a feedforward type algorithm of the form in (4.1). The analysis is performed in terms of the l_2-norm of the output error e_k.

Proposition 4.2 *Suppose that the symmetric matrix $GK + (GK)^T$ is a positive-definite matrix. Then there exists a learning gain $\beta^* > 0$ such that $\beta^* > \beta > 0$ ensures that $\|e_{k+1}\|^2 < \|e_k\|^2$ for all $e_k \neq 0$. Furthermore the value of any such gain β can be obtained by satisfying the following matrix inequality*

$$(\frac{1}{\beta}I - GK)^T(\frac{1}{\beta}I - GK) < \frac{1}{\beta^2}I \tag{4.3}$$

(the inequality being expressed with respect to the partial ordering on symmetric matrices i.e. $M_1 < M_2$ if, and only if, $v^T M_1 v < v^T M_2 v$ whenever $v \neq 0$).

Proof. The use of equation (4.2) yields the difference $\|e_{k+1}\|^2 - \|e_k\|^2$ in the form

$$\|e_{k+1}\|^2 - \|e_k\|^2 = -\beta e_k^T(GK + (GK)^T)e_k + \beta^2 e_k^T(GK)^T GK e_k \tag{4.4}$$

Since $GK + (GK)^T$ is a positive-definite matrix and $\beta > 0$, then for an arbitrary nonzero e_k, the term $-\beta e_k^T(GK+(GK)^T)e_k$ is strictly negative and

the term $\beta^2 e_k^T (GK)^T GK e_k$ is strictly positive. To guarantee that $\|e_{k+1}\|^2 < \|e_k\|^2$ it is necessary and sufficient that the following inequality must be true:

$$\beta v^T (GK + (GK)^T) v > \beta^2 v^T (GK)^T GK v \qquad \forall \quad v \neq 0 \tag{4.5}$$

Since the left hand term of inequality (4.5) is of $O(\beta)$ and the right hand term is of $O(\beta^2)$ it follows that the inequality is met for all sufficiently small $\beta > 0$. This proves the existence of $\beta^* > 0$ (independent of e_k) that ensures monotonic convergence. A few algebraic manipulations of (4.5), namely completing the square, give the proposed inequality (4.3). □

Proposition 4.2 shows that if $GK + (GK)^T > 0$ and (4.3) holds true then error convergence is monotonic, the next proposition furthers shows that under this condition the error also converges to zero for an arbitrary initial error e_0.

Proposition 4.3 *Under the assumptions of Proposition 4.2 with* $0 < \beta < \beta^*$, *the algorithm ensures that the tracking error sequence* $\{e_k\}_{k \geq 0}$ *converges monotonically to zero in the* l_2 *norm.*

Proof. Because $\|e_k\| > \|e_{k+1}\| \geq 0$, $\lim_{k \to \infty} \|e_k\|$ exists, i.e. $\lim_{k \to \infty} \|e_k\| = E$, $E \geq 0$. If $E = 0$ then the proof is complete. Assume therefore that $E > 0$. Then there exists a subsequence $\{e_{k_j}\}$ of $\{e_k\}$ such that $\lim_{k_j \to \infty} e_{k_j} = e_\infty$, where e_∞ satisfies $\|e_\infty\| = E$. Consider the sequence $\{e_{k_j+1}\}$ generated by

$$e_{k_j+1} = (I - \beta GK) e_{k_j} \tag{4.6}$$

Based on Proposition 4.2, the sequence $\{e_{k_j+1}\}$ satisfies the inequality

$$\lim_{k_j \to \infty} \|e_{k_j+1}\| \leq \lim_{k_j \to \infty} \|e_{k_j}\| = E \tag{4.7}$$

However, the sequence $\{e_{k_j+1}\}$ is also a subsequence of $\{e_k\}$ (which can be assumed to have a subsequential limit $\hat{e}_\infty$ with norm $\|\hat{e}_\infty\| = E$), and therefore

it must hold that $\lim_{k_j \to \infty} \|e_{k_j+1}\| = \|\hat{e}_\infty\| = E = \|e_\infty\|$. A contradiction is obtained by noting the relationship $\hat{e}_\infty = (I - \beta GK)e_\infty$ and using Proposition 1 to show that $\|\hat{e}_\infty\| < \|e_\infty\|$, which concludes the proof. □

The next proposition refines this result to prove geometric convergence.

Proposition 4.4 *If $GK + (GK)^T$ is a positive-definite matrix, then there exists a real scalar $\alpha \in (0,1)$ (dependent on β but not e_0) such that $\|e_{k+1}\| \leq \alpha \|e_k\|$ for all $k \geq 0$.*

Proof. Using the error evolution equation to calculate the l_2-norm of e_{k+1} gives

$$\|e_{k+1}\|^2 = \|e_k\|^2 - \beta e_k^T (GK + (GK)^T) e_k + \beta^2 e_k^T (GK)^T GK e_k \qquad (4.8)$$

Since $GK + (GK)^T$ is a positive-definite matrix the following bound for $\|e_{k+1}\|$ is obtained:

$$\|e_{k+1}\|^2 \leq (1 - \beta \underline{\lambda} + \beta^2 \overline{\sigma}^2) \|e_k\|^2 \qquad (4.9)$$

where $\underline{\lambda}$ is the smallest eigenvalue of $GK + (GK)^T$ and $\overline{\sigma}$ is the largest singular value of GK. The choice of a sufficiently small $\beta > 0$ then indicates the existence of a real scalar $\alpha \in (0,1)$. □

Note: a simple consequence of the result is that $\|e_k\| \leq \alpha^k \|e_0\|$ for all $k \geq 0$ and hence the result can be regarded as an alternative proof of the previous proposition.

Remark 4.1 *Propositions 4.2, 4.3 and 4.4 advocate the use of a sufficiently small $\beta > 0$ since it induces monotonic error convergence when $GK + (GK)^T$ is a positive-definite matrix. It is worth noting however that as $\beta \to 0$ it follows that $u_{k+1} \to u_k$. This implies that an excessively small $\beta > 0$ will slow the rate of the convergence.*

The results in this section show that if $GK + (GK)^T$ is positive-definite then monotonic error convergence is achieved by selecting a simple learning gain, $\beta > 0$, to be sufficiently small. The remainder of this chapter focuses on two choices for the matrix K. The first choice is $K = G^{-1}$ and shall be referred to as the *inverse algorithm* and the second choice is $K = G^T$ and shall be referred to as the *adjoint algorithm.*

4.3 The Inverse Algorithm

ILC can be regarded as an iterative solution of an inverse problem (i.e. solve $r = Gu$ for the input time series u) in a manner that can be implemented both in the form of experimental procedures during plant operation or as a simulation process. As a consequence, the existence and properties of a plant inverse are crucial to theoretical developments. This section concentrates on the discrete-time case for plant inverses, an analysis for the continuous-time scenario is presented in (Furuta and Yamakita, 1987). In what follows, it is shown that inverses (or approximations to inverses) can generate algorithms that have useful theoretical properties. To underline the potential of iterative *inverse* procedures, consider the conceptual (benchmark) ILC algorithm:

$$u_{k+1} = u_k + \beta G^{-1} e_k \tag{4.10}$$

where β is a scalar. Assume that the procedure is initiated with the choice of an *arbitrary* initial control time series u_0 leading to an initial error e_0. If $\beta = 1$, a simple analysis of the corresponding error evolution equation shows the expected result that the error converges to zero in one iteration. More precisely, computation of e_1 gives

$$e_1 = r - y_1 = r - Gu_1 = r - G(u_0 - G^{-1}e_0) = 0 \tag{4.11}$$

In this case it would appear that the inverse model algorithm can be regarded as a theoretically *perfect* ILC algorithm. If $\beta \neq 1$, then a similar analysis indicates that $e_k = (1-\beta)^k e_0$ so that convergence to zero is assured in the range $0 < \beta < 2$.

Implementation of the above benchmark algorithm requires that an exact system model is available and can be implemented. Unfortunately an exact system model is invariably unavailable although a nominal model G_o may be available or be chosen deliberately to reduce the computational load on the control implementation.

4.4 Robustness of the Inverse algorithm

This section is interested in both robust stability and performance of this one-parameter inverse ILC algorithm. In the *nominal* case of $G = G_o$ it has been seen that the algorithm reduces the magnitude of the error at each iteration i.e. if $0 < \beta < 2$, then $||e_{k+1}|| = |1-\beta|||e_k|| < ||e_k||$ for all k where $e_k \neq 0$. It is easily seen that this monotonicity property is a very strong and valuable theoretical and practical property of ILC methods (see (Amann *et al.*, 1996) and (Owens and Feng, 2003)) as it indicates a guarantee of improved performance from trial to trial. In what follows the robustness of the inverse-model ILC algorithm is analysed under the constraint that control performance retains monotonic error convergence in the presence of model uncertainty.

The result takes the form of a relationship between the learning gain β and the multiplicative uncertainty representation $G = G_o U$, where U is a square matrix, and can be interpreted as defining the uncertainty that can be tolerated for a given β or as defining the range of β that can be

used for a given uncertainty. Taking U to represent a proper, causal, linear, time-invariant system then it must have a similar structure to G i.e. lower-triangular *Toeplitz*. The use of such multiplicative perturbations has been standard practice for years in robust control and is comprehensively covered in (Skogestad and Postlethwaite, 1996).

In general, robustness ideas reflect the need to retain a property (e.g. stability) despite the presence of modelling errors. In the context of this chapter the essential properties of an ILC algorithm will be

1. the convergence of the algorithm to a zero tracking error (independent of the initial trial control used) and

2. the monotonicity of the Euclidean norm (mean-square value) of the error time series.

Both properties are beneficial to ILC performance with convergence being essential. As a consequence it is natural to introduce the concept of *robust monotone convergence* by requiring both of these properties to be present despite the presence of modelling errors.

A crucial step is the characterization of the modelling errors that can then be tolerated.

A robust monotone convergence analysis for the inverse algorithm introduced above is given below. A main result expressing monotonicity of the inverse ILC algorithm now follows:

Proposition 4.5 *Suppose that the symmetric matrix $U + U^T$ is a positive-definite matrix. Then there exists a learning gain $\beta^* > 0$ such that $\beta^* > \beta > 0$ ensures that $\|e_{k+1}\|^2 < \|e_k\|^2$ for all $e_k \neq 0$. Furthermore the value of any such gain β can be obtained by satisfying the following matrix inequality*

$$(\frac{1}{\beta}I - U)^T(\frac{1}{\beta}I - U) < \frac{1}{\beta^2}I \tag{4.12}$$

Proof. The inverse algorithm is a special case of the generic feedforward ILC control law in (4.1), where $K = G_o^{-1}$. Substituting in the matrix $GK = GG_o^{-1} = U$ into the proof of proposition 4.2 completes this proof. □

Remark 4.2 *It is clear that the property of geometric convergence to zero error described in propositions 4.3 and 4.4 are true for the inverse algorithm if $U + U^T$ is a positive-definite matrix. The proof of such properties for the inverse algorithm is omitted to limit repetition.*

4.5 The Adjoint Algorithm

The analysis in section 4.2 shows that the key to obtaining monotone error convergence is choosing the matrix K such that $GK + (GK)^T$ is a positive-definite matrix. One obvious choice, other than the inverse, is to let $K = G^T$ so that the control law becomes:

$$u_{k+1} = u_k + \beta G^T e_k \tag{4.13}$$

This algorithm shall be termed as the *adjoint* algorithm throughout this thesis. The term adjoint is used because the relationship $(Gw)^T v = w^T(G^T v)$ is similar to that of an operator $\tilde{G}$ and its adjoint operator $\tilde{G}^*$ where the following inner product relationship holds:

$$\langle \tilde{G}\tilde{w}, \tilde{v} \rangle_{\mathcal{V}} = \langle \tilde{w}, \tilde{G}^* \tilde{v} \rangle_{\mathcal{W}} \quad \forall \quad \tilde{v} \in \mathcal{V}, \tilde{w} \in \mathcal{W} \tag{4.14}$$

For a formal description of adjoint operators see (Young, 1988), (Reddy, 1986), (Nering, 1963) or any other introductory text on functional analysis. An analysis of an adjoint algorithm in a continuous-time scenario can be found in (Furuta and Yamakita, 1987).

It is clear from the analysis in section 4.2 that the convergence properties of the adjoint algorithm depend upon the matrix GG_o^T. The following

proposition shows that robust monotone convergence for an arbitrary initial error requires that the multiplicative plant uncertainty $(U + U^T)$ to be a positive-definite matrix. Furthermore it shows the existence of a sufficiently small learning gain, β, that guarantees robust monotone convergence given that the necessary condition holds.

Proposition 4.6 *Suppose that the symmetric matrix $U + U^T$ is a positive-definite matrix. Then there exists a learning gain $\beta^* > 0$ such that $\beta^* > \beta > 0$ ensures that $\|e_{k+1}\|^2 < \|e_k\|^2$ wherever $e_k \neq 0$, $\forall k \geq 0$. Furthermore the value of any such gain β can be obtained by satisfying the following matrix inequality:*

$$(\frac{1}{\beta}I - G_o^T U G_o)^T(\frac{1}{\beta}I - G_o^T U G_o) < \frac{1}{\beta^2}I \tag{4.15}$$

Proof. The proof is similar to that of Proposition 4.2 with the most notable exception being the following necessary inequality for the existence of β^*.

$$\beta v^T G_o(U + U^T)G_o^T v > \beta^2 \|G_o U G_o v\|^2 \qquad \forall \quad v \neq 0 \tag{4.16}$$

The remainder of the proof is omitted for brevity. □

Remark 4.3 *It is clear that the property of geometric convergence to zero error described in propositions 4.3 and 4.4 are true for the adjoint algorithm if $U + U^T$ is a positive-definite matrix. The proof of such properties for the adjoint algorithm is omitted to limit repetition.*

4.6 Simulation Examples

Consider the case where a real plant $G(s)$ is approximated with a lower order system $G_o(s)$. In the s-domain, these are taken to be

$$G(s) = \frac{2}{s^2 + 2s} \tag{4.17}$$

$$G_o(s) = \frac{1}{s} \tag{4.18}$$

The two systems are sampled with a sample time of 0.1 seconds using zero-order hold. The trial length T_f is 25.5 seconds. It is easy to check numerically that the resulting multiplicative uncertainty matrix $U + U^T$ is positive-definite. The reference signal is chosen to be $r(t) = sin(\frac{2\pi}{T_f}t)$.

Figures 4.1 and 4.2 illustrate the existence of a sufficiently small $\beta > 0$ that induces robust monotone convergence. Figure 4.1 shows that if, for the inverse algorithm, $\beta = 2$ the convergence is monotonic however a small increase to $\beta = 2.05$ gives asymptotic convergence. A similar behaviour is illustrated in figure 4.2 for the adjoint algorithm. A small change from $\beta = 0.00765$ to $\beta = 0.00768$ sees monotonic error convergence change to divergence.

A comparison of the inverse and adjoint algorithms for this example shows that the inverse algorithm converges much faster than the adjoint algorithm. Experimental comparison, not presented here, for a range of LTI SISO systems and their reduced order models shows that in general the inverse algorithm converges faster than the adjoint. The experimental comparison adds one further observation. Whilst the adjoint algorithm is in general slower to converge it is also slower to diverge should the robustness conditions given in this chapter not be held.

Finally figure 4.3 shows the tracking performance after 120 iterations for the inverse algorithm where $\beta = 2$. The output $y_{120}(t)$ is visually indistinguishable from the reference $r(t)$ and demonstrates that the error signal converges towards zero.

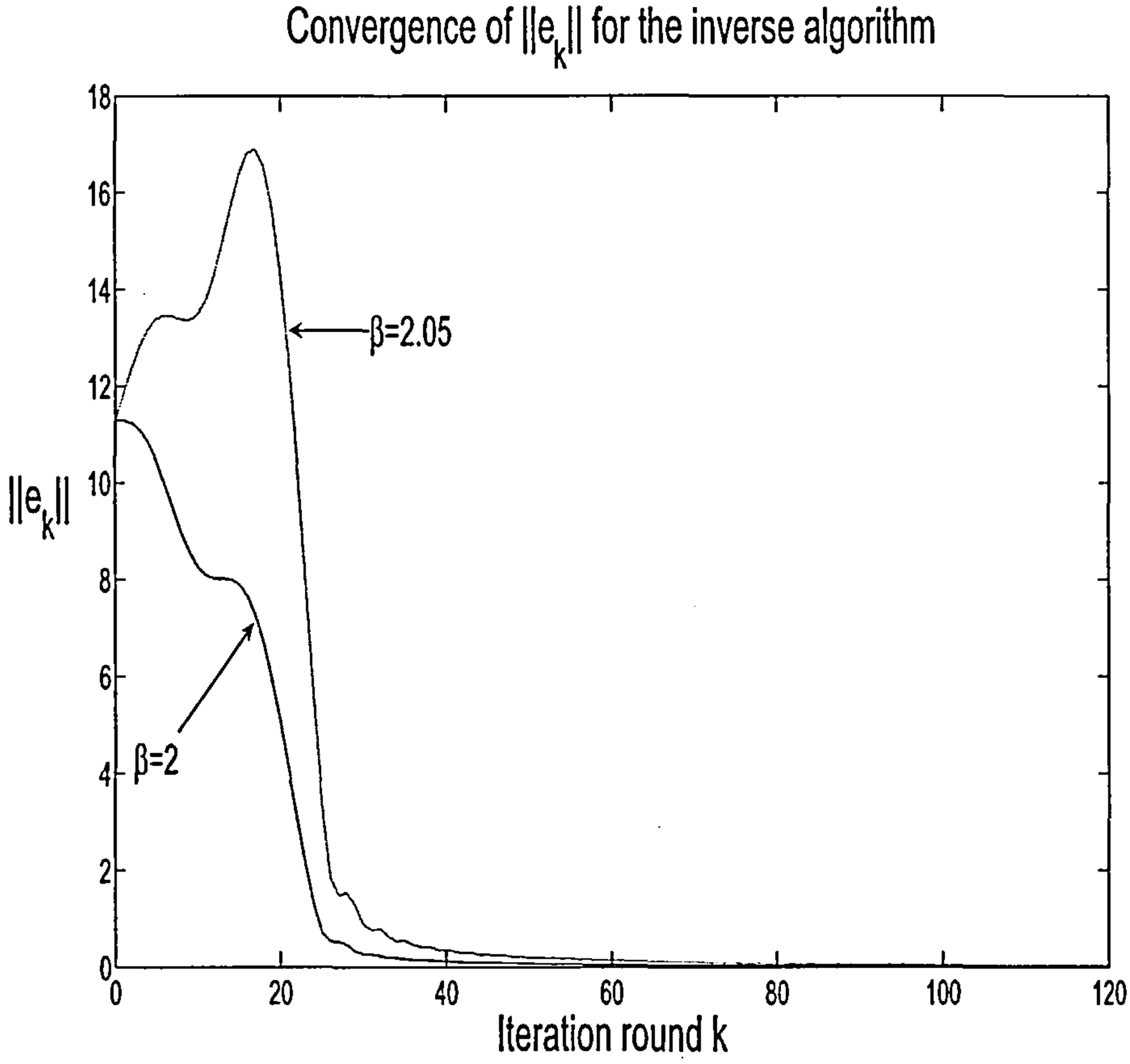

Figure 4.1: Convergence of error for the inverse ILC algorithm

4.7 Summary

In this chapter a generic feed-forward type ILC algorithm , $u_{k+1} = u_k + \beta K e_k$, was analysed. The algorithm is simple in structure, easy to implement and leaves a tuning variable, β and matrix K open for design and manipulation.

It was shown that if a matrix $GK + (GK)^T$ was positive-definite and if the simple scalar $\beta > 0$ is made small enough then the algorithm results in monotonic convergence to zero error. It is numerically simple to test the condition should the plant matrix G be available. Two design choices for

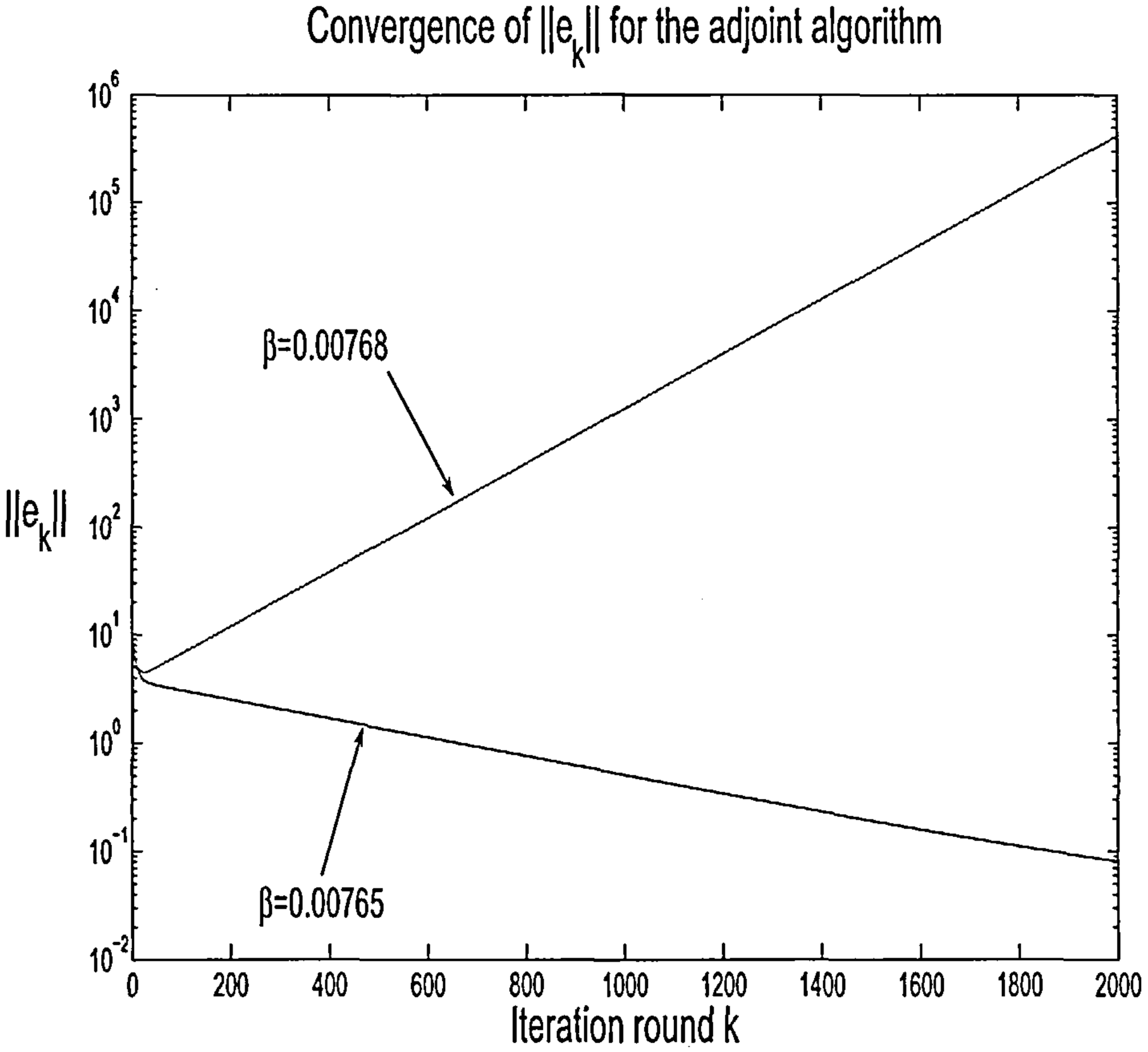

Figure 4.2: Convergence of error for the adjoint ILC algorithm

K that ensured the positive-definite condition was held were then presented. The first was $K = G^{-1}$ and was termed the inverse algorithm. The second was $K = G^T$ and was termed the adjoint algorithm.

Both the choices for K assume the controller has full knowledge of the plant matrix G. Accordingly a robustness analysis was performed where only a lower-order plant model G_o is available to the controller. The plant and its model are related by a multiplicative uncertainty U such that $G = UG_o$. It was shown that if $U + U^T$ is a positive-definite matrix and if the scalar $\beta > 0$

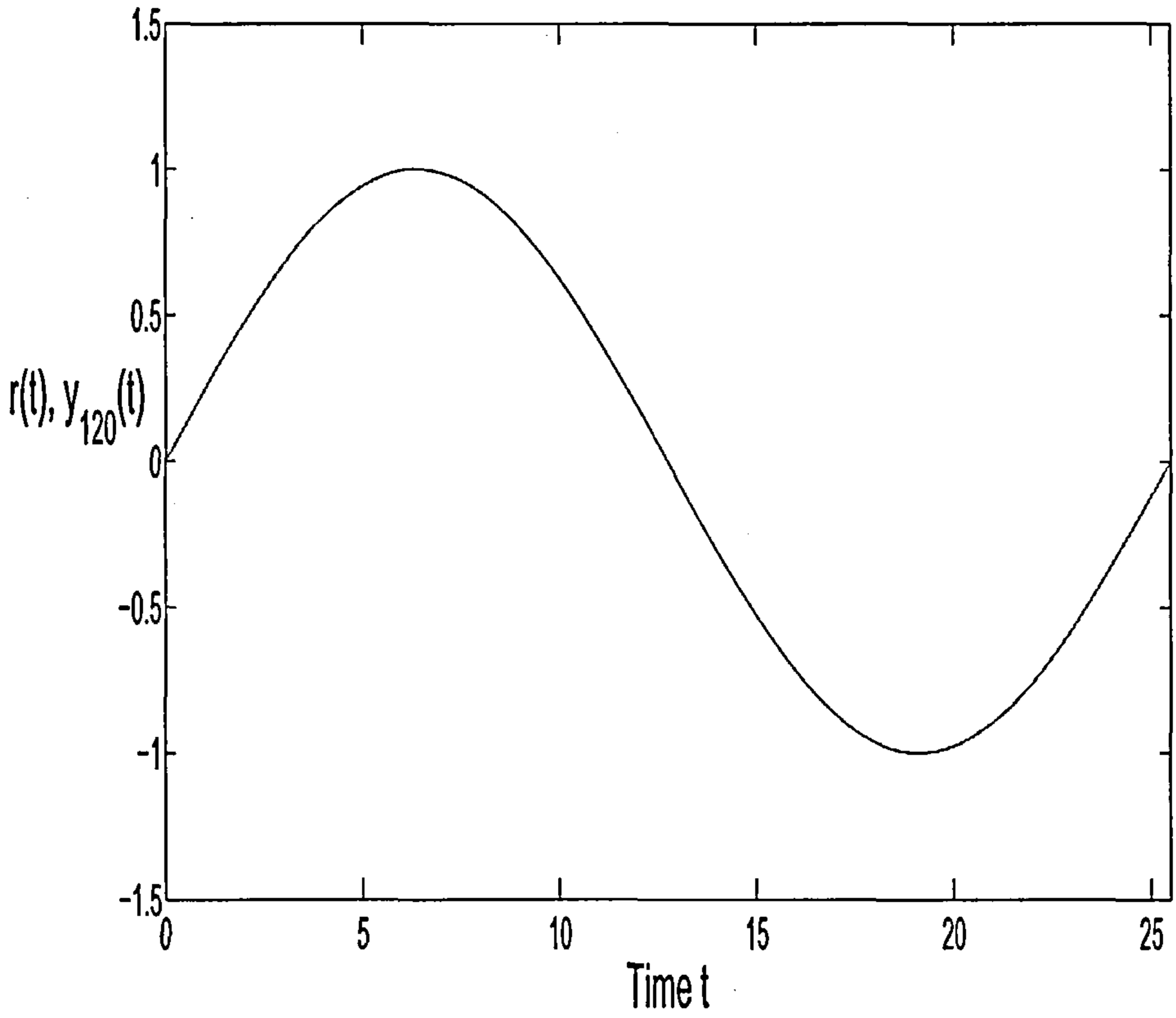

Figure 4.3: Tracking performance for inverse ILC algorithm with $\beta = 2$ after 120 iterations

is made small enough then the error will monotonically converge to zero in spite of the model uncertainty.

The results in this chapter confirm that feed-forward type ILC, and in particular the inverse and adjoint algorithms are powerful tools. The algorithms have a simple structure, a clear robustness analysis for both stability and performance and a simple design principle. The same statement cannot be made for POILC and its predecessors, reviewed in Chapter 3, hence the

results developed in this chapter are a significant development for ILC.

Chapter 5

Frequency Domain conditions for convergence using inverse and adjoint ILC

In Chapter 4 necessary conditions for robust monotone convergence using either the inverse or the adjoint algorithms were found in terms of the plant uncertainty matrix. Whilst these conditions give insight into the convergence properties of the inverse and adjoint algorithms they would be simpler to interpret if they could be extended into the *more natural* frequency domain commonly used in linear control theory and practice.

The major contribution of this chapter is the development of robustness conditions in the frequency domain and is presented, for the inverse algorithm, in publication P2 (Harte *et al.*, 2005).

5.1 System representations in the Frequency domain

Throughout this chapter a discrete-time plant shall be represented in the frequency domain by use of the z-transform and z-transfer functions.

A link from the work in Chapter 4 to z-transfer functions is obtained by noting that both matrices and z-transfer functions link the same input and output time series. It is therefore easily shown, see (Oppenheim and Schafer, 1989) and (Rosenbrock, 1970), that the following proposition holds true:

Proposition 5.1 *Suppose that $G(z)$ has relative degree greater than or equal to $G_o(z)$ and that $G(z) = U(z)G_o(z)$. If G, G_o and U are matrix representations of these systems then $G = UG_o$.*

Note 1: the relative degree assumption is needed to ensure that U is causal.

Note 2: the result is used later in this chapter to obtain frequency domain robustness conditions.

Note 3: in the case of the lifted plant where $zG(z) = U(z)zG_o(z)$, used throughout this chapter, the result holds without loss of generality.

5.2 A frequency domain condition for robust monotone convergence using the inverse algorithm

Consider the inverse algorithm presented in Chapter 4:

$$u_{k+1} = u_k + \beta G_o^{-1} e_k \tag{5.1}$$

The analysis in Chapter 4 showed that for robust monotone convergence using the inverse algorithm then it is necessary that the following matrix inequality be held.

$$(\frac{1}{\beta}I - U)^T(\frac{1}{\beta}I - U) < \frac{1}{\beta^2}I \tag{5.2}$$

The following proposition gives a sufficient condition for robust monotone convergence in terms of the uncertainty in the plant transfer function. The basis of the proposition is to consider the relationship between the matrix U and the underlying system with the transfer function $U(z)$.

Proposition 5.2 *Given the notation in Proposition 5.1, suppose that $U(z)$ is a stable system, then*

(i) $U + U^T > 0$ if $Re[U(z)] > 0$ at all points on the unit circle $|z| = 1$

(ii) a sufficient condition for monotonic convergence is that

$$\sup_{|z|=1} |\frac{1}{\beta} - U(z)| < \frac{1}{\beta} \tag{5.3}$$

Proof. For (i) note that, for any $v \in \mathbb{R}^N$ we have $v^T(U + U^T)v = 2v^TUv$. Note also that v^TUv is identical to the inner product in l_2 of the response of the system with transfer function $U(z)$ to the input whose first N values are those of the elements of v with all other values taken to be zero. Using standard contour integration then gives

$$2v^TUv = \frac{1}{2\pi i}\int_{|z|=1} v(z^{-1})U(z)v(z)\frac{dz}{z} = \frac{1}{2\pi}\int_0^{2\pi} |v(e^{i\theta})|^2 U(e^{i\theta})d\theta \tag{5.4}$$

where $v(z) = v_1 + z^{-1}v_2 + ... + z^{-(N-1)}v_N$. Part (i) follows as the left-hand-side is real which implies that the imaginary part of $U(e^{i0})$ can be deleted from the right-hand-side.

Moving to (ii), let $M_N = \frac{1}{\beta}I - U$.The maximal value for the left-hand side of (5.2) is given by $\|M_N\|^2\|v\|^2$ and therefore a sufficient condition for

monotonic convergence is that

$$\|M_N\| < \frac{1}{\beta} \tag{5.5}$$

Let M_∞ be the corresponding infinite matrix mapping when $N \to \infty$. Note that M_∞ is a lower-triangular Toeplitz matrix and therefore has a dynamical system interpretation, i.e. there exists a state-space realization (A_M, B_M, C_M, D_M) that generates M_∞. Due to the assumed stability of $U(z)$, M_∞ is a bounded mapping and therefore $\|M_\infty\|_2$ exists. This results in

$$\|M_\infty\| = \sup_{x \in S_\infty} \|M_\infty x\| \geq \sup_{x \in S_N} \|M_\infty x\| \geq \|M_N\| \tag{5.6}$$

where $S_\infty = \left\{ x \in l_2 \middle| \|x\| = 1 \right\}$ and $S_N = \left\{ x \in l_2 \middle| \|x\| = 1 \wedge x(t > N) = 0 \right\}$. The inequality in (5.6) follows as $S_N \subset S_\infty$. Hence $\|M_\infty\| < \frac{1}{\beta}$ implies that $\|M_N\| < \frac{1}{\beta}$. Furthermore it is a well known result, see (Grenander and Szegö, 1984) for example and further (Norrlöf and Gunnarsson, 2002) for its use in an ILC context, that $\|M_\infty\| = \sup_{|z|=1} |M(z)|$ where $M(z) := (\frac{1}{\beta} - U(z))$. The result follows trivially from this and (i) as (ii) implies that $Re[U(z)] > 0$ for all $|z| = 1$ and hence that $U + U^T > 0$. □

Remark 5.1 *The condition in (5.3) can be interpreted in terms of a Nyquist plot of $U(z)$ on the unit circle. For robust monotone convergence, it is sufficient that the Nyquist plot of $U(z)$ lies within a circle with radius $\frac{1}{\beta}$ centred about the point $(\frac{1}{\beta}, 0)$ on the Complex plane. Section 5.4 demonstrates this in the context of a simulation example.*

Remark 5.2 *As $\beta \to 0+$, the circle of centre $(\frac{1}{\beta}, 0)$ in the Complex plane and radius $\frac{1}{\beta}$ eventually fills the whole of the right Complex plane i.e. robust monotone convergence for any* strictly positive *uncertainty $U(z)$ can be achieved by using a sufficiently small learning gain $\beta > 0$.*

Remark 5.3 *A consequence of condition (5.3) is that the modelling error $U(z)$ must be a positive-real system. By definition, the phase shift of such a system lies within $\pm 90^o$ for all frequencies $|z| = 1$. The robustness result hence states that the algorithm can tolerate a plant uncertainty of $\pm 90^o$ phase shift for all frequencies but that the gain tolerance is phase dependent.*

5.3 Adjoint Algorithm - frequency domain condition

Consider the adjoint algorithm presented in Chapter 4:

$$u_{k+1} = u_k + \beta G_o^T e_k \tag{5.7}$$

It is easily shown that $\|e_{k+1}\|^2$ is given as follows:

$$\|e_{k+1}\|^2 = \|e_k\|^2 - \beta e_k^T G_o (U + U^T) G_o^T e_k + \beta^2 \|G_o U G_o^T e_k\|^2 \tag{5.8}$$

The analysis in Chapter 4 showed that it was necessary that $v^T(U+U^T)v > 0$ for all $v \neq 0$ and that the learning gain $\beta > 0$ be sufficiently small in order to induce robust monotone convergence. The following proposition extends this condition on $U + U^T$ to a sufficient condition in the z-domain and is given in terms of the plant uncertainty transfer function $U(z)$.

Proposition 5.3 *Given the notation in Proposition 5.1, suppose that $U(z)$ is a stable system, then*

(i) There exists a learning gain $\beta^* > 0$ such that $\beta^* > \beta > 0$ ensures that $\|e_{k+1}\| < \|e_k\|$ for all $e_k \neq 0$ if $v^T(U + U^T)v > 0$ for all $v \neq 0$.

(ii) $v^T(U + U^T)v > 0$ for all $v \neq 0$ if $Re[U(z)] > 0$ at all points on the unit circle $|z| = 1$

Proof. For (i) consider the difference $\|e_{k+1}\|^2 - \|e_k\|^2$ given as follows:

$$\|e_{k+1}\|^2 - \|e_k\|^2 = -\beta e_k^T G_o(U + U^T)G_o^T e_k + \beta^2 \|G_o U G_o^T e_k\|^2 \qquad (5.9)$$

Clearly if $v^T(U + U^T)v > 0$ for all $v \neq 0$ then there exists a sufficiently small β^* that ensures the right-hand side of (5.9) is strictly negative and hence $\|e_{k+1}\| < \|e_k\|$ for all $e_k \neq 0$.

For (ii) note that, for any $v \in \mathbb{R}^N$ we have $v^T(U + U^T)v = 2v^T Uv$. Note also that $v^T Uv$ is identical to the inner product in l_2 of the response of the system with transfer function $U(z)$ to the input whose first N values are those of the elements of v with all other values taken to be zero. Using standard contour integration then gives

$$2v^T Uv = \frac{1}{2\pi i}\int_{|z|=1} v(z^{-1})U(z)v(z)\frac{dz}{z} = \frac{1}{2\pi}\int_0^{2\pi} |v(e^{i0})|^2 U(e^{i0})d\theta \qquad (5.10)$$

where $v(z) = v_1 + z^{-1}v_2 + ... + z^{-(N-1)}v_N$. Part (ii) follows as the left-hand-side is real which implies that the imaginary part of $U(e^{i\theta})$ can be deleted from the right-hand-side. □

Remark 5.4 *The condition that $Re[U(z)] > 0$ for all $|z| = 1$ can be interpreted in terms of a Nyquist plot. The condition becomes that it is sufficient that the Nyquist plot of $U(z)$ lie in the right-hand side of the Complex plane.*

Remark 5.5 *A consequence of the condition $Re[U(z)] > 0$ for all $|z| = 1$ is that the modelling error $U(z)$ must be a positive-real system. By definition, the phase shift of such a system lies within $\pm 90^o$ for all frequencies $|z| = 1$. The robustness result hence states that the algorithm can tolerate a plant uncertainty of $\pm 90^o$ phase shift for all frequencies.*

5.4 Simulation Example

In this section a simulation example is given to illustrate the results presented in this chapter for the inverse algorithm.

Consider the case where a real plant $G(s)$is approximated with a lower order system $G_o(s)$. In the $s-$domain, these are taken to be

$$G(s) = \frac{1}{s(s+1)} \tag{5.11}$$

$$G_o(s) = \frac{1}{s} \tag{5.12}$$

The two systems are sampled with a sample time of $0.1s$ using zero-order hold and, transforming to the z-domain (without any change in notation for simplicity), a corresponding model uncertainty $U(z)$ is obtained using $U(z) = G_o^{-1}(z)G(z)$. Fig.5.1 shows the Nyquist plot $U(z)$ lies marginally within a circle of radius 0.5 about the point $(0.5, 0)$. The inverse of the matrix G_o generated by (5.12) is used to calculate the inputs for the update law (5.1).

The parameter β will be used to demonstrate the use of the theory and to test the conservatism of the sufficient condition (5.3) and Nyquist plot interpretation given in the section 5.2. An initial choice of $\beta = 1$ is made. Fig.5.2 then shows that the Nyquist plot of $U(z)$ lies in the open right-half complex plane (and hence it is positive) and also within the circle of radius 1 about the point $(1, 0)$ i.e. the sufficient condition (5.3) holds and robust monotone convergence should be expected. Fig.5.3 and Fig.5.4 indeed show that the l_2 error converges monotonically to zero. Increasing β to a value of $\beta = 2$ yields a circle that is indistinguishable visually from the Nyquist plot of $U(z)$ (see Fig.5.2) and once again the error is expected to converge monotonically to zero (see Fig.5.3 and Fig.5.4 for confirmation of

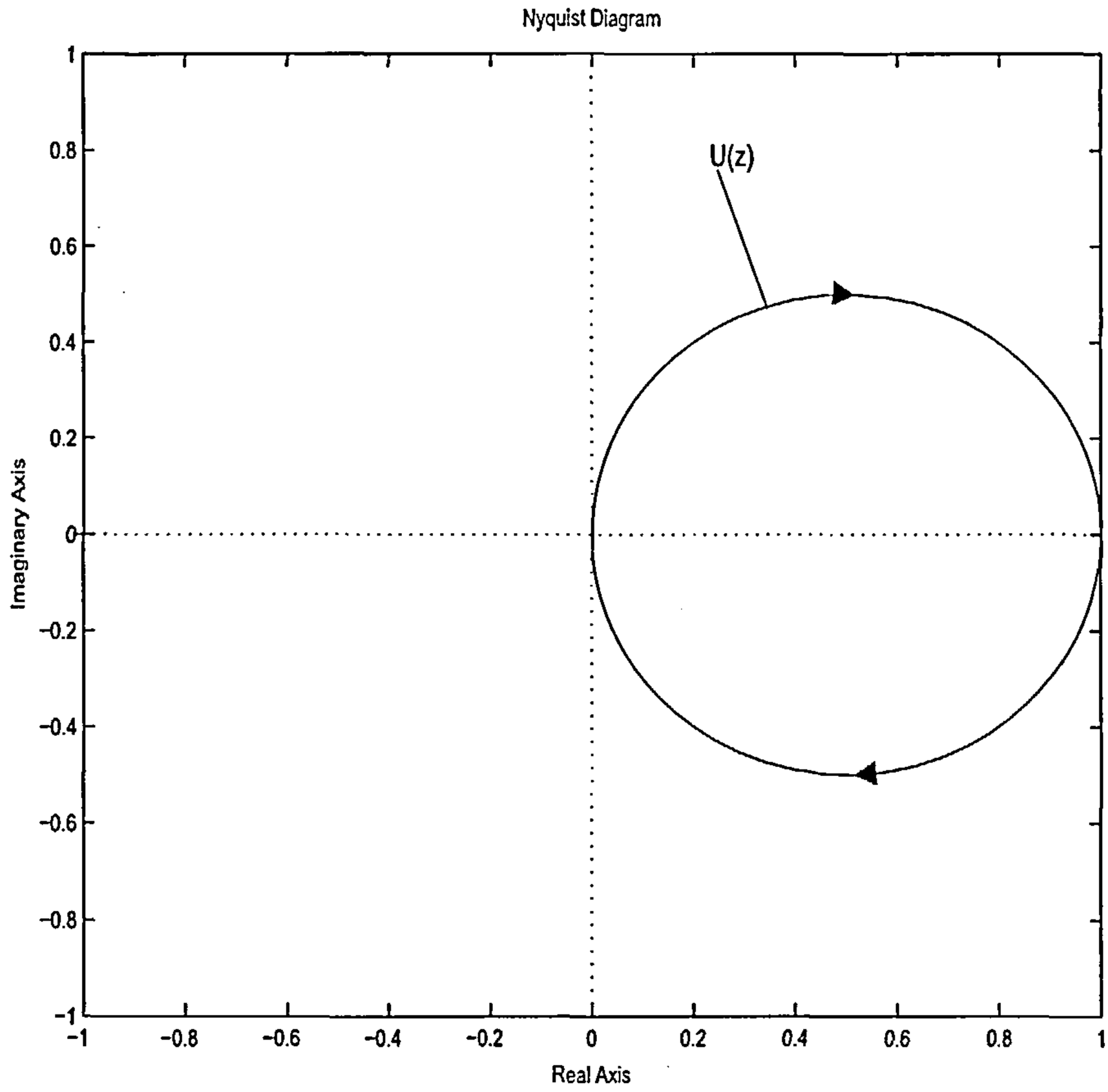

Figure 5.1: Nyquist plot of $U(z)$ on the unit circle

this property). A final choice of $\beta = 2.05$ (only a very small increase in β) shows (see Fig.5.2) that the Nyquist plot of $U(z)$ lies marginally outside of the $\frac{1}{\beta}$ circle and hence that (5.3) is broken. The very low conservatism of the robustness condition is seen from Fig.5.3 which shows that for this case the error convergence is no longer monotonic. However, the error does still converge asymptotically to zero (see Fig.5.4).

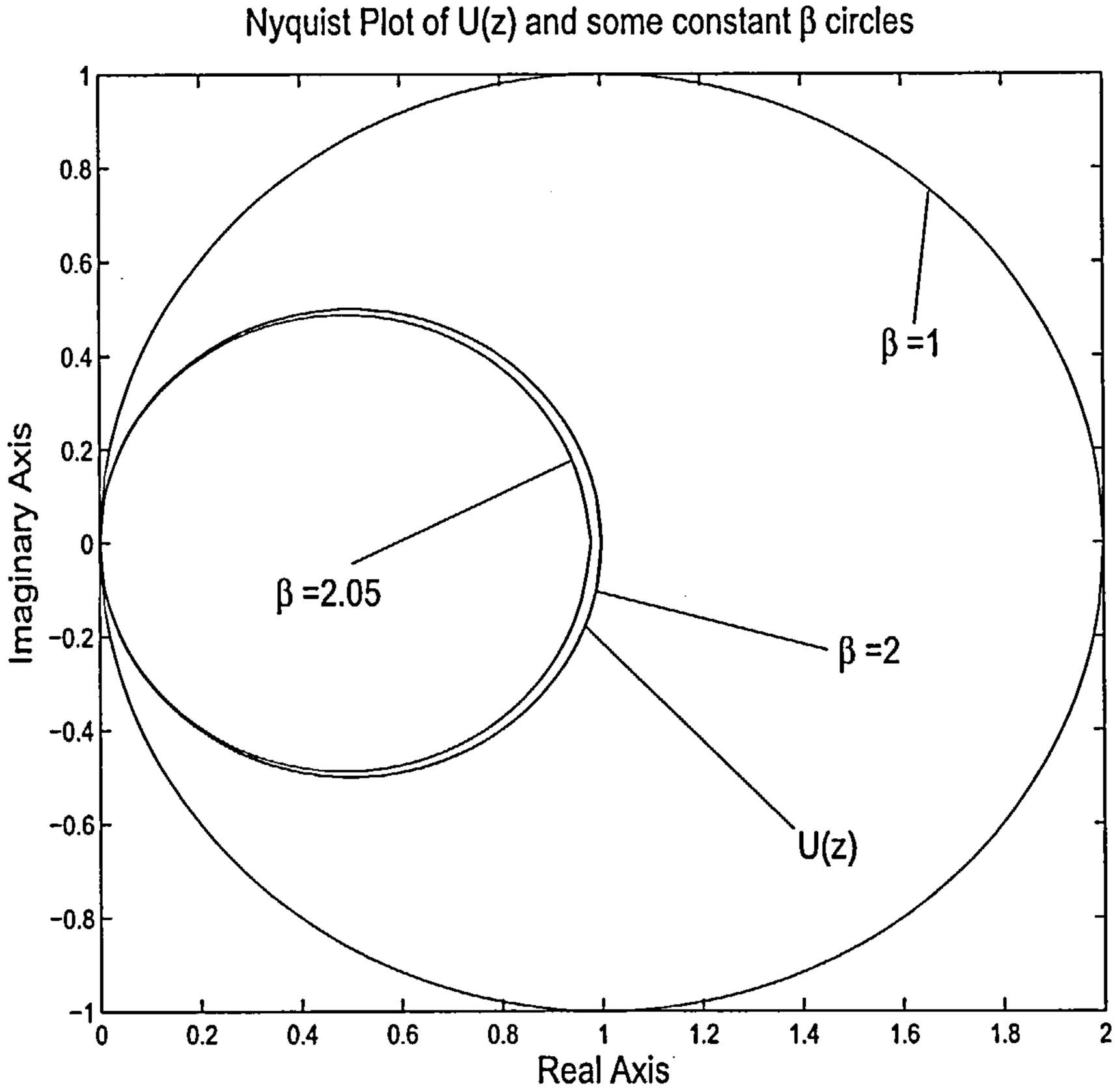

Figure 5.2: Nyquist Plot of Uncertainty with differing β circles

5.5 Summary

In this chapter the time-domain analysis of inverse and adjoint type ILC was extended to into the frequency-domain context. The Toeplitz structure of an uncertainty matrix, U, allows for a link between matrix and frequency domain descriptions of the plant uncertainty via the use of Parseval's relation. The results are typically of the form that robustness is retained if the multiplicative modelling error $U(z)$ has a positive real property and a simple learning gain is small enough. The results are expressed for the two model

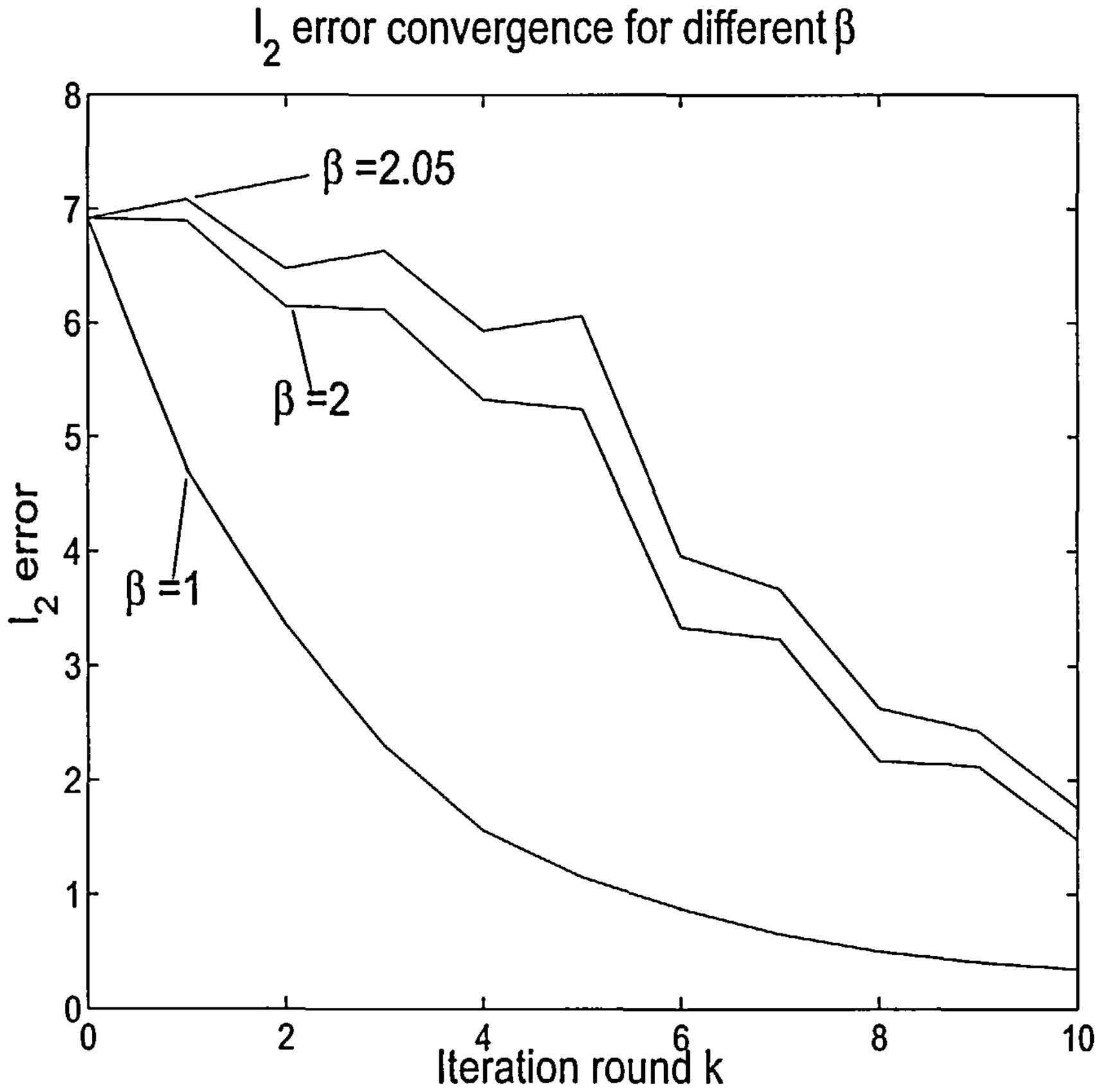

Figure 5.3: Error comparison for differing β, $k = 1, 2, .., 10$

based algorithms and provide a simple graphical Nyquist interpretation of the robustness analysis.

For the inverse algorithm the condition presented provides a direct link between modelling error and learning gain magnitude. For the adjoint algorithm it is not yet clear how the scaling properties of the matrix G_o^T extend into the frequency domain. The development of frequency domain conditions linking the model uncertainty, $U(z)$, the plant model, $G_o(z)$, and the magnitude of the learning gain, β, would be a significant contribution to the

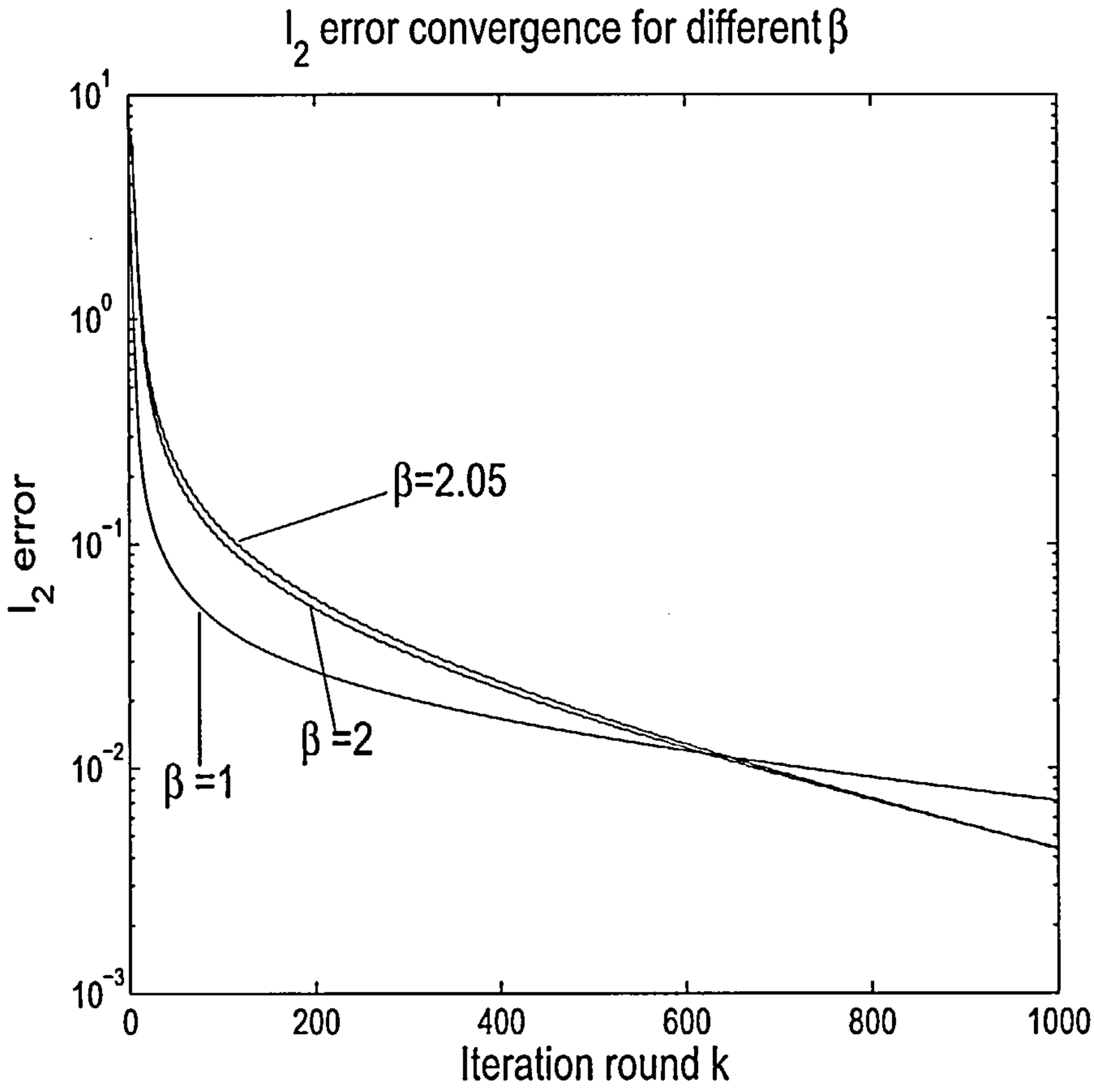

Figure 5.4: Error comparison for differing β, 1000 iterations

robustness analysis of adjoint ILC.

The robustness conditions developed in this chapter are only sufficient however numerical examples indicate that the degree of conservatism present may be very small in practice.

Chapter 6

Inverse and Adjoint type POILC

The POILC concepts used in (Owens and Feng, 2003) and reviewed in Chapter 3 can be applied to both the inverse and adjoint algorithms discussed in the previous two chapters. The main contribution in this chapter is the development of robustness conditions for both Inverse and Adjoint type POILC and is presented in publication P3 (Hätönen *et al.*, 2003*b*).

The analysis for both algorithms is similar and this chapter shall concentrate on analysis for the adjoint scenario. The results for the inverse algorithm shall be stated with brief analysis so as to avoid unnecessary repetition.

Parameter Optimisation was first applied to the adjoint algorithm in the context of continuous-time ILC in (Furuta and Yamakita, 1987) where the authors termed it the *steepest-descent* algorithm. The paper analyses the convergence properties of the algorithm when the plant model contains multiplicative uncertainty. The analysis is done under the assumption that the trial length is infinite. Furthermore, because the the steepest-descent algorithm contains a non-causal operator, the time axis is taken to

be $t \in (-\infty, \infty)$ resulting in a complicated analysis.

In this chapter a similar analysis is carried out in the discrete-time case assuming a finite trial length and results in a necessary and sufficient condition for monotonic convergence. The algorithm is found to be lacking a mechanism to ensure that the necessary and sufficient condition holds. A modified steepest-descent algorithm is then proposed that is less restrictive and provides a transparent mechanism for balancing the dual aims of convergence speed and robustness.

6.1 The Standard Steepest-Descent algorithm

In the standard steepest descent algorithm for discrete-time ILC the idea is to minimise the cost-function

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 \tag{6.1}$$

during each trial. Assume now that the input u_k is modified so that during iteration $k+1$ the input law $u_{k+1} = u_k + \beta_{k+1} G^T e_k$ is used where β_{k+1} is a scaling factor optimally updated at each trial. The cost index for iteration $k+1$ becomes

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 = \|e_k\|^2 - 2\beta_{k+1} e_k^T G G^T e_k + \beta_{k+1}^2 \|G G^T e_k\|^2 \tag{6.2}$$

It is straightforward to show that the optimal β_{k+1}^* is given as below.

$$\beta_{k+1}^* = \frac{\|G^T e_k\|^2}{\|G G^T e_k\|^2} \tag{6.3}$$

By optimality it can be seen the l_2 norm of the error converges monotonically.

$$\begin{aligned} J(\beta_{k+1}^*) &\leq J(0) \\ \|e_{k+1}\|^2 &\leq \|e_k\|^2 \end{aligned} \tag{6.4}$$

6.2 Robustness of the Standard Steepest-Descent algorithm

Assume now that only a nominal model of plant, G_o, is available and is related to the true plant by the equation

$$G = G_o U \tag{6.5}$$

where U is the multiplicative plant uncertainty. Furthermore, G_o is used in the update law so that $u_{k+1} = u_k + \beta_{k+1} G_o^T e_k$. The optimal β^*_{k+1} becomes

$$\beta^*_{k+1} = \frac{e_k^T G G_o^T e_k}{\|G G_o^T e_k\|^2} \tag{6.6}$$

however a nominal β_{k+1}

$$\beta_{k+1} = \frac{\|G_o^T e_k\|^2}{\|G_o G_o^T e_k\|^2} \tag{6.7}$$

is used instead. The following proposition gives a necessary and sufficient condition for monotonic error convergence in terms of the model uncertainty U.

Proposition 6.1 *Suppose $U+U^T$ is a positive-definite matrix, then if $\|e_k\| \neq 0$ and there exists a $\beta_{k+1} > 0$ such that $J(\beta_{k+1}) < J(0)$ then it implies e_{k+1} converges monotonically.*

Proof. If $J(\beta_{k+1}) < J(0)$ then from (6.1) the following inequality must be true

$$\beta_{k+1} e_k^T G_o (U + U^T) G_o^T e_k > \beta_{k+1}^2 \|G G_o^T e_k\|^2 \tag{6.8}$$

Note that both the terms in the above inequality are strictly positive but the left-hand term is of $O(\beta_{k+1})$, compared to $O(\beta^2_{k+1})$ on the right-hand side hence the inequality held for some sufficiently small β_{k+1}. □

Analysing β_{k+1}, in (6.7), there is no clear mechanism to ensure that β_{k+1} is small enough to meet the condition in Proposition 6.1. Consequently the next section introduces a new modified steepest-descent algorithm that results in monotonic convergence for plants with multiplicative uncertainty U such that $U + U^T$ is a positive-definite matrix.

6.3 A Modified Steepest-Descent algorithm

In order to enhance the robustness properties of the standard steepest-descent algorithm consider again the algorithm

$$u_{k+1} = u_k + \beta_{k+1} G^T e_k \tag{6.9}$$

where β_{k+1} is selected to minimise the optimisation problem

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 + w\beta_{k+1}^2 \tag{6.10}$$

where $w \in \mathbb{R}$, $w > 0$. The cost function $J(\beta_{k+1})$ in (6.10) reflects two design objectives. The first term in $J(\beta_{k+1})$ reflects the objective that the tracking error should be small during each iteration. The second term on the other hand tries to keep the magnitude of β_{k+1} small, possibly resulting in a more cautious and robust algorithm when compared to the standard steepest-descent algorithm.

The optimisation problem (6.10) can be solved in a straightforward manner where the optimal solution is as follows:

$$\beta_{k+1}^* = \frac{\|G^T e_k\|^2}{w + \|GG^T e_k\|^2} \tag{6.11}$$

A convergence analysis of this algorithm follows:

Proposition 6.2 *If* $w \in \mathbb{R}^+$ *then* $\|e_{k+1}\| < \|e_k\|$ *if* $e_k \neq 0$. *Furthermore,*

$$\lim_{k\to\infty} \|e_k\| \to 0 \quad \textit{and} \quad \lim_{k\to\infty} \beta^*_{k+1} \to 0 \tag{6.12}$$

demonstrating monotonic convergence to zero tracking error.

Proof. Selecting a sub-optimal choice $\beta_{z,k+1} = 0$ in the cost function (6.10) yields $J(0) = \|e_k\|^2$. Since this choice is sub-optimal it follows:

$$\|e_k\|^2 \geq \|e_{k+1}\|^2 + w\beta^{*2}_{k+1} \geq \|e_{k+1}\|^2 \tag{6.13}$$

which shows monotonic convergence. Reformulating (6.13) yields

$$\|e_k\|^2 - w\beta^{*2}_{k+1} \geq \|e_{k+1}\|^2 \geq 0 \tag{6.14}$$

and applying induction on $\|e_k\|^2$ gives

$$\|e_0\|^2 - w\sum_{i=1}^{k+1} \beta^{*2}_i \geq 0 \tag{6.15}$$

and because k is arbitrary, $\lim_{k\to\infty} \beta^*_{k+1} = 0$. This results in

$$\lim_{k\to\infty} \beta^*_{k+1} = \lim_{k\to\infty} \frac{\|G^T e_k\|^2}{w + \|GG^T e_k\|^2} = 0 \tag{6.16}$$

This is only possible if $\lim_{k\to\infty} e_k = 0$ because G is non-singular (this is the assumption made in Section 1.2. Furthermore, the interlacing result (6.13) implies that $\|e_k\| > \|e_{k+1}\|$ if $e_k \neq 0$ which completes the proof. □

6.4 Robustness analysis of the modified steepest-descent algorithm

Consider again the case where the true plant includes a multiplicative uncertainty, i.e. $G = G_o U$ where U is the uncertainty and G_o is the nominal plant model which is used so that

$$u_{k+1} = u_k + \beta_{k+1} G_o^T e_k \tag{6.17}$$

The optimal β^*_{k+1} is given by

$$\beta^*_{k+1} = \frac{e_k^T G G_o^T e_k}{w + \|G G_o^T e_k\|^2} \tag{6.18}$$

however since the true plant matrix is unavailable to the controller a sub-optimal β_{k+1} is used instead and is given as follows:

$$\beta_{k+1} = \frac{\|G_o^T e_k\|^2}{w_{k+1} + \|G_o G_o^T e_k\|^2} \tag{6.19}$$

Note that w_{k+1} is an iteration varying scalar and can be used to counter the effect of any plant uncertainties.

The following proposition shows that if $(U + U^T)$ is a positive-definite matrix then monotonic error convergence is attained if w_{k+1} is sufficiently large for all k.

Proposition 6.3 *If $U + U^T$ is a positive-definite matrix and $w_{k+1} > 0$ is selected to be sufficiently large then it implies that $\|e_k\| > \|e_{k+1}\|$ for all $e_k \neq 0$. Furthermore the sufficiently large w_{k+1} is characterised by the following inequality:*

$$w_{k+1} > \frac{w}{\underline{\lambda}} + \frac{\|G G_o^T\|^2}{\underline{\lambda}} \|e_k\|^2 \tag{6.20}$$

where $\underline{\lambda}$ is the smallest eigenvalue of $U + U^T$.

Proof. By the optimality of β^*_{k+1} it follows that

$$\|e_{k+1}\|^2 + w\beta^{*2}_{k+1} \leq J(\beta_{k+1}) \tag{6.21}$$

It further follows that if $J(\beta_{k+1}) < J(0)$ for all $e_k \neq 0$ then it implies that $\|e_{k+1}\| < \|e_k\|$ for all $e_k \neq 0$.

The proof continues by developing conditions for w_{k+1} and the matrix $U + U^T$ that ensure $J(\beta_{k+1}) < J(0)$ for all $e_k \neq 0$. Substituting (6.17) into (6.10) gives the following expression for $J(\beta_{k+1})$:

$$J(\beta_{k+1}) = J(0) - \beta_{k+1} e_k^T G_o (U + U^T) G_o^T e_k + \beta^2_{k+1}(w + \|G G_o^T e_k\|^2) \tag{6.22}$$

From (6.22) it can be seen that $J(\beta_{k+1}) < J(0)$ for all $e_k \neq 0$ if $U + U^T$ is a positive-definite matrix and the following inequality holds true for all $e_k \neq 0$:

$$\beta_{k+1} e_k^T G_o (U + U^T) G_o^T e_k > \beta_{k+1}^2 (w + \|GG_o^T e_k\|^2) \tag{6.23}$$

Substituting (6.19) into (6.23) followed by some algebraic manipulation yields the inequality in (6.20) and hence completes the proof. □

Remark 6.1 *Suppose w_1 is chosen to be large enough such that $\|e_0\| > \|e_1\|$ then the inequality in (6.20) will be held if $w_2 = w_1$. This can be seen by noticing that the right-hand side of (6.20) monotonically converges as $\|e_k\|$ monotonically converges, hence selecting $w_{k+1} = w_1$ for all $k > 1$ guarantees the condition holds for all $k > 1$. However selecting an excessively large and constant value for w_{k+1} can be quite conservative and can result in a small β_{k+1}, implying that $u_{k+1} \approx u_k$ in such a case. Consequently proposition 6.3 should be understood to be an existence result, and that in practice w_{k+1} can be selected by a trial and error approach.*

The following proposition shows that if the condition in proposition 6.3 holds then $\|e_k\|$ converges to zero.

Proposition 6.4 *Under the assumptions of proposition 6.3 then $\|e_k\|$ converges to zero.*

Proof. If the assumptions of proposition 6.3 hold then it follows that

$$\|e_{k+1}\|^2 + w\beta_{k+1}^2 \leq \|e_k\|^2 \tag{6.24}$$

The remainder of the proof is identical to that of proposition 6.2 where β_{k+1}^* is replaced with β_{k+1} and G is replaced with G_o. □

6.5 Inverse type POILC

The analysis techniques presented throughout the earlier sections of this chapter can be applied to inverse-type ILC. In fact replacing the matrix GG_o^T in the robustness analysis of adjoint-type POILC with GG_o^{-1} gives the corresponding analysis for inverse-type POILC. With this in mind this section shall refrain from indulging in unnecessary repetition by stating just the key results for inverse-type POILC.

The following update law is intended to be used

$$u_{k+1} = u_k + \beta_{k+1} G^{-1} e_k \tag{6.25}$$

where β_{k+1} is selected to minimise the following cost function

$$J(\beta_{k+1}) = \|e_{k+1}\|^2 + w\beta_{k+1}^2 \tag{6.26}$$

where the optimal β_{k+1}^* is given by

$$\beta_{k+1}^* = \frac{\|e_k\|^2}{w + \|e_k\|^2} \tag{6.27}$$

and w is chosen to be a positive scalar.

However the true plant includes multiplicative uncertainty, i.e. $G = G_o U$ where U is the uncertainty and G_o is the nominal plant model which is used in the update law (6.25) so that

$$u_{k+1} = u_k + \beta_{k+1} G_o^{-1} e_k \tag{6.28}$$

Note that since the the true plant is unknown, the optimal value for β_{k+1} is actually given by

$$\beta_{k+1}^* = \frac{e_k^T U e_k}{w + \|U e_k\|^2} \tag{6.29}$$

and is also unknown to the controller so a nominal estimate β_{k+1}, given below, must be used instead.

$$\beta_{k+1} = \frac{\|e_k\|^2}{w_{k+1} + \|e_k\|^2} \tag{6.30}$$

In (6.30) w_{k+1} is an iteration varying scalar that is used to counter the effect of any plant uncertainties.

The following propositions give sufficient conditions for the monotonic convergence of the error under the action of inverse-type POILC.

Proposition 6.5 *If $U+U^T$ is a positive-definite matrix and $w_{k+1} > 0$ is selected to be sufficiently large then $\|e_k\| > \|e_{k+1}\|$ for all $e_k \neq 0$. Furthermore the sufficiently large w_{k+1} is characterised by the following inequality:*

$$w_{k+1} > \frac{w}{\underline{\lambda}} + \frac{\|U\|^2}{\underline{\lambda}}\|e_k\|^2 \tag{6.31}$$

where $\underline{\lambda}$ is the smallest eigenvalue of $U+U^T$.

Proof. See the proof of proposition 6.3 replacing the matrix GG_o^T with U.□

Proposition 6.6 *Under the assumptions of proposition 6.5 then $\|e_k\|$ converges to zero.*

Proof. If the assumptions of proposition 6.5 hold then it follows that

$$\|e_{k+1}\|^2 + w\beta_{k+1}^2 \leq \|e_k\|^2 \tag{6.32}$$

The remainder of the proof is identical to that of proposition 6.2 where β_{k+1}^* is replaced with β_{k+1}, G is replaced with G_o and G^T is replaced with G_o^{-1}.□

6.6 Simulation Examples

Consider the case where a real plant $G(s)$ is approximated with a lower order system $G_o(s)$. In the s-domain, these are taken to be

$$G(s) = \frac{2.1}{s+1}\frac{20}{s+20} \tag{6.33}$$

$$G_o(s) = \frac{1}{s+1} \tag{6.34}$$

The two systems are sampled with a sampling interval of 0.1 seconds using zero-order hold. The trial length T_f is 25.5 seconds. It is easy to check numerically that the resulting multiplicative uncertainty matrix $U + U^T$ is positive-definite. The reference signal is chosen to be $r(t) = sin(\frac{2\pi}{T_f}t)$. An initial input $u_0 = 0$ is assumed.

Figure 6.1 shows that when the standard deepest descent algorithm is used that the error signal diverges. Figure 6.2 shows the error convergence for the modified algorithm with three different values of w_{k+1}. When w_{k+1} is chosen to be a constant equaling 5 for all iterations the error still diverges. A further increase to $w_{k+1} = 8$ shows stability is regained and the error monotonically converges. A final choice of $w_{k+1} = 0.001 + 0.7\|e_k\|^2$ results in a faster convergence. This final choice is adaptive and is based on the structure of (6.20). Such a structure attempts to manage the trade-off between robustness and monotone convergence speed by utilising the fact that the sufficient inequality for monotone convergence in (6.20) is heavily dependent upon $\|e_k\|^2$. The three values demonstrate the existence of a sufficiently large w_{k+1} that induces monotonic convergence, they further demonstrate that this value decreases as $\|e_k\|$ decreases.

Figure 6.3 shows the convergence of $w_{k+1} = 0.001 + 0.7\|e_k\|^2$. It is clear that during the early iterations w_{k+1} is sufficiently large, i.e $w_{k+1} > 8$ during the early iterations. Then as e_k approaches zero w_{k+1} converges towards 0.001 and hence the algorithm starts to converge towards the standard algorithm, i.e. $w_{k+1} \approx 0$ as $e_k \to 0$. This implies that, for this example, the standard algorithm can work for smaller error signals but fails for larger error signals.

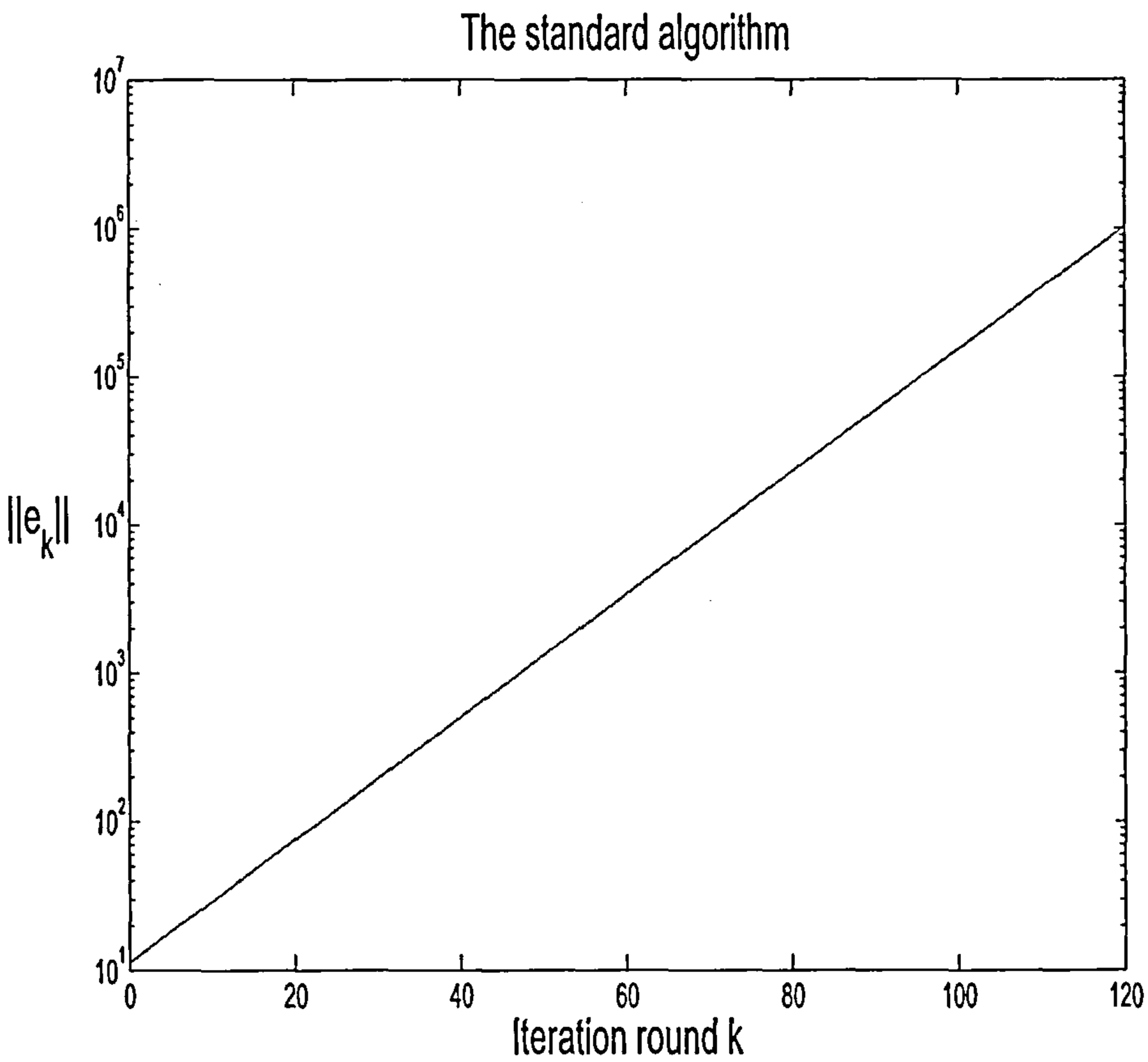

Figure 6.1: Convergence behaviour with the standard algorithm

6.7 Summary

In this chapter an algorithm $u_{k+1} = u_k + \beta_{k+1} K e_k$, $K = G^T$ or $K = G^{-1}$, was analysed where a scalar variable β_{k+1} is chosen at each iteration using parameter optimisation. In the nominal case where the plant does not have any uncertainty it was shown that the algorithm converges to zero error for an arbitrary discrete-time LTI SISO plant.

The scenario where only a plant model, G_o, is available to the controller was then considered. The model was taken to be related to the plant, G,

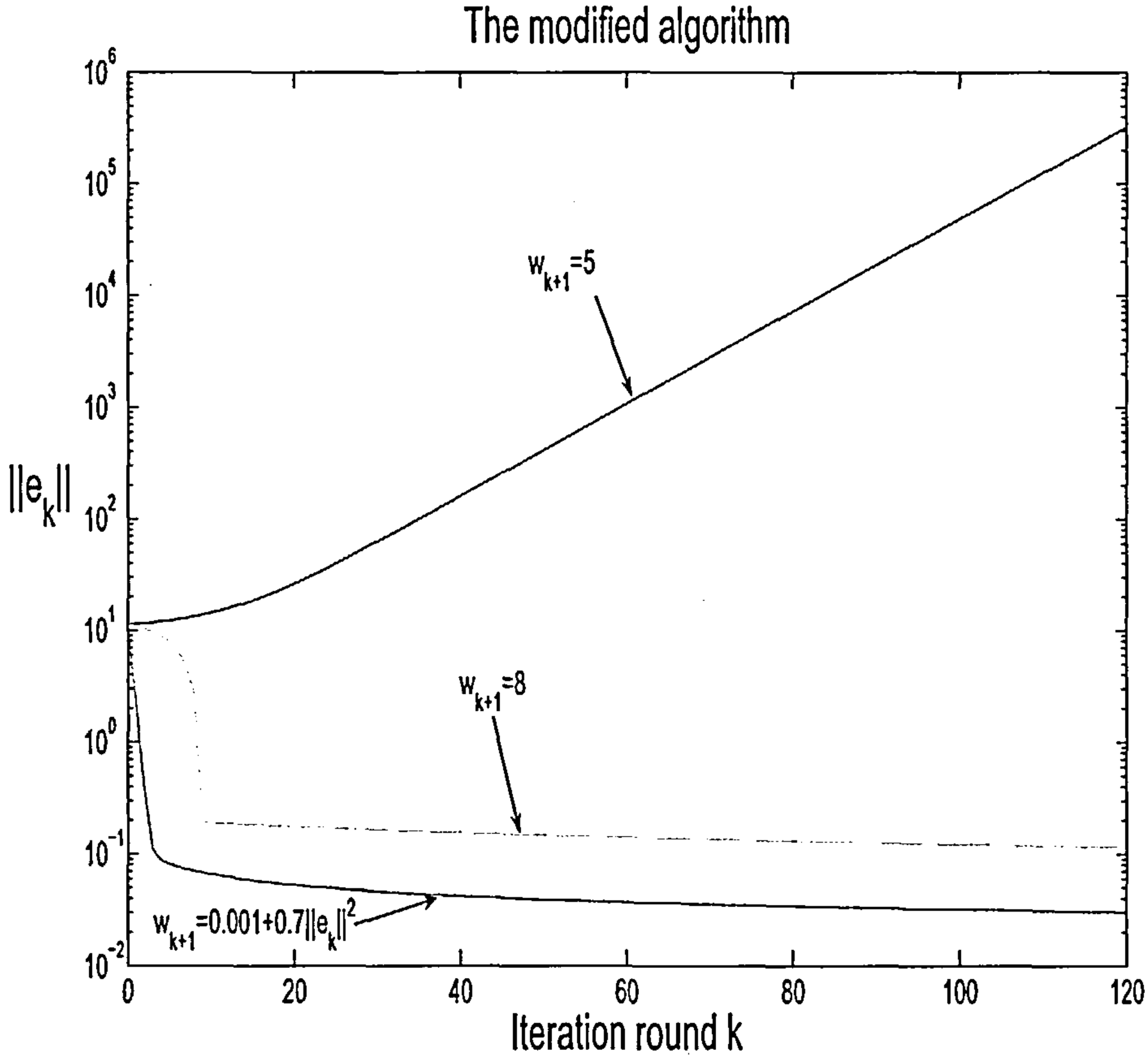

Figure 6.2: Convergence behaviour with the modified algorithm

by a multiplicative uncertainty U such that $G = UG_o$. It was shown in this case that the algorithm still converges monotonically to zero tracking error if $U + U^T$ is a positive-definite matrix and the scalar β_{k+1} is made sufficiently small.

Initially the case where β_{k+1} is optimised according to an index, termed the steepest descent index, was analysed. It was shown that for this algorithm that there is no clear mechanism to ensure β_{k+1} is sufficiently small.

The case where β_{k+1} was optimised according to the POILC index, see

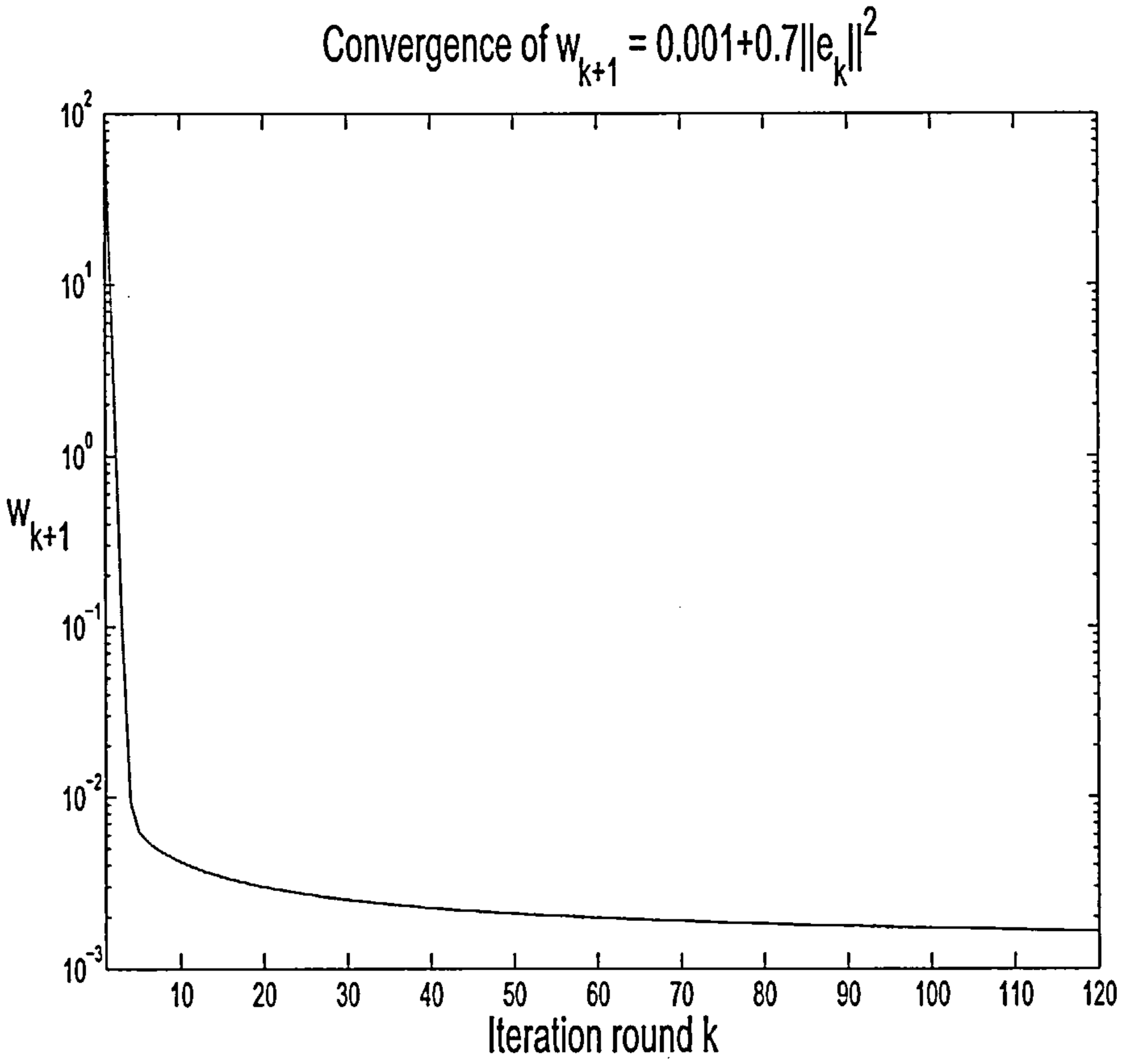

Figure 6.3: Convergence behaviour of iteration varying w_{k+1}

Chapter 3, was then analysed. The POILC index introduces an adaptive tuning parameter w_{k+1} into the algorithm. The tuning parameter provides a means of ensuring β_{k+1} is sufficiently small at each iteration and also gives a straightforward mechanism of finding a balance between convergence speed and robustness.

Note that the modified algorithm, where $K = G_o^T$, was applied on a industrial-scale gantry robot in P.3 (see section 1.3). Near perfect tracking was achieved in approximately 100 iterations. Not only does this result act

to validate the findings in this chapter it also shows further promise for the algorithm since the linear model, G_o, neglected a number of nonlinear elements in the gantry robot.

Chapter 7

Basis Functions and ILC

The analysis in Chapter 4 introduced the concept of robust monotone convergence and for the inverse and adjoint algorithms it was shown that in order to achieve robust monotone convergence that it is necessary for the plant multiplicative uncertainty matrix to be positive-definite. For some applications this condition may be too restrictive.

This chapter uses basis functions to look at the convergence properties of a *filtered* tracking error subject to plant uncertainty that does not meet the positive-definite requirement. A common set of basis functions, namely the singular values basis, is used as a filter to design a restricted set of input signals that induce monotonic convergence of a *filtered error* in spite of non-positive multiplicative model uncertainty. The chapter then gives a derivation of the results for a generic orthornormal basis. A simulation example then illustrates the robustness results and highlights directions for future work on the use of basis functions for ILC.

The main contribution of this chapter is the introduction of basis functions to retain the stability and monotonically converging properties of Inverse and Adjoint ILC algorithms in spite of non-positive multiplicative model

uncertainty. This concept was originally presented for a specific set of basis functions, the discrete fourier transform, in publication P4 (Hätönen *et al.*, 2004*b*) and this chapter can be seen as an extension of these results.

7.1 Basis Functions

This chapter heavily utilises basis functions and gives a brief review of the notion so as to avoid any notational confusion. For a more formal review see (Reddy, 1986), (Nering, 1963) or any other introductory text on linear analysis.

Throughout the chapter a set of vectors $\mathcal{V} = \{v_i\}$ is called a set of orthogonal basis functions if an arbitrary vector, $p \in \mathbb{R}^N$, can be written as

$$p = \sum_{i=1}^{N} \alpha_i v_i \tag{7.1}$$

for some $\alpha_i \in \mathbb{R}$ and where $\{v_i\}$ satisfies

$$v_i^T v_j = 0 \tag{7.2}$$

when $i \neq j$. Furthermore the basis shall be called an orthonormal basis if

$$v_i^T v_i = 1 \qquad \forall \quad i \in \{1, 2, \ldots, N\} \tag{7.3}$$

otherwise it shall be called an orthogonal basis.

Throughout this chapter a set of basis vectors $\mathcal{V}$ shall be written as a matrix mapping $V : \mathbb{R}^N \rightarrow \mathbb{R}^N$ where

$$V = [v_1, v_2, \ldots, v_N] \tag{7.4}$$

and can be used in equation (7.1) such that

$$p = V\alpha \tag{7.5}$$

where the coefficients α_i are written in the following *super-vector* notation:

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \tag{7.6}$$

The vector α is given as the solution to the minimisation problem

$$\min_{\alpha \in \mathbb{R}^N} J(\alpha) = \|p - V\alpha\|^2 \tag{7.7}$$

and is found to be

$$\alpha = (V^T V)^{-1} V^T p \tag{7.8}$$

The matrix $V^T V$ is a diagonal matrix with non-zero entries along the diagonal, this is due to the assumption that $\mathcal{V}$ spans $\mathbb{R}^N$ and the orthogonality assumption in (7.2). Hence $V^T V$ is non-singular and invertible. Furthermore it can used to scale any orthogonal basis so that it becomes orthonormal. The remainder of the chapter shall solely consider orthonormal bases.

7.2 Basis Function approach to ILC

The plant input u described in Chapter 1 can be written in terms of a basis set $\mathcal{V}$ as follows:

$$u = \sum_{i=1}^{N} \alpha_i v_i = V\alpha \tag{7.9}$$

The ILC problem defined in Chapter 1 can now become a question of learning the set of base coefficients $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$, such that $\|r - GV\alpha\|$ is minimised, rather than directly learning the perfect plant input u^*.

The following *integrating* update law is presented

$$\alpha_{i,k+1} = \alpha_{i,k} + \sum_{j=1}^{N} k_{ij} q_j^T e_k \tag{7.10}$$

where $q_i^T e_k$ is the i^{th} coefficient of e_k projected onto an orthonormal basis $\mathcal{Q} = \{q1, q2, \ldots, q_N\}$. The i^{th} input coefficient $\alpha_{i,k+1}$ is chosen to be the *integral* of a weighted sum of the error , e_k, projected onto the base $\mathcal{Q}$. The choice of k_{ij}, $\mathcal{V}$ and $\mathcal{Q}$ is left open for the designer to compensate for the dynamics of the plant G.

The update law in (7.10) can be rewritten in super-vector notation as follows:

$$\alpha_{k+1} = \alpha_k + KQe_k \tag{7.11}$$

The evolution of the error signal is then described as

$$e_{k+1} = (I - GVKQ)e_k \tag{7.12}$$

The analysis in Chapter 4 shows that $GVKQ+(GVKQ)^T$ must be a positive-definite matrix with a sufficiently small gain in order to achieve monotonic convergence to zero error.

7.3 Singular Value Decomposition and ILC

In the previous section two orthonormal matrices Q and V and a weighting matrix K were left as an open choice for the control designer to compensate the plant dynamics with. This section presents that a Singular Value Decomposition (SVD) of the plant G provides a natural choice for Q, V and K.

The singular values, σ_i, left and right singular vectors, v_i and q_i of a square matrix G are given by the eigenvalues and the left and right eigenvectors of GG^T. The following relation later proves to be a useful design and analysis tool:

$$Gv_i = \sigma_i q_i \tag{7.13}$$

Equation (7.13) describes the operation of a square matrix G upon a vector v_i as a rotation to another vector q_i and a scaling by a factor σ_i. From here on it shall be assumed that $\sigma_i > 0 \forall i \in \{1, 2, \ldots, N\}$.

The appropriateness of using SVD in a basis function approach to ILC comes from the fact that the set of left and right singular vectors, $\mathcal{V}$ and $\mathcal{Q}$, are both orthonormal and can be written such that

$$GV = QS \tag{7.14}$$

where

$$S = \begin{bmatrix} \sigma_1 & 0 & \ldots & \ldots & 0 \\ 0 & \sigma_1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \sigma_{N-1} & 0 \\ 0 & \ldots & \ldots & 0 & \sigma_N \end{bmatrix} \tag{7.15}$$

Using the SVD of G, the update law can be chosen in the following manner:

$$\begin{aligned} u_{k+1} &= \textstyle\sum_{i=1}^{N} \alpha_{i,k+1} v_i \\ \alpha_{i,k+1} &= \alpha_{i,k} + \beta \sigma_i^{-1} q_i^T e_k \end{aligned} \tag{7.16}$$

where $\beta \in \mathbb{R}$ is a positive scalar chosen to ensure stability and govern the learning rate.

Note that the term $\beta\sigma_i^{-1}$ in (7.16) gives the algorithm a similarity to the inverse ILC algorithm discussed in Chapter 4. The algorithm in (7.16) attempts to invert plant dynamics along each of the N directions as opposed to trying to directly invert the plant matrix. A similar analogy to the adjoint algorithm could be made by using $\beta\sigma_i$ in (7.16). This opens up possibility of using different update laws along different search directions, something that is not transparently done under the analysis of the inverse and adjoint algorithms in Chapter 4.

The following proposition gives a necessary and sufficient condition for monotonic error convergence for the algorithm in (7.16)

Proposition 7.1 *If* $0 < \beta < 2$ *then* $\|e_{k+1}\| < \|e_k\|$ *for all* $e_k \neq 0$.

Proof. The use of (7.16) and few algebraic manipulations yield an identity for e_{k+1}.

$$e_{k+1} = e_k - \sum_{j=1}^{N} G v_j \beta \sigma^{-1} q_j^T e_k \tag{7.17}$$

Then by use of (7.13) and the orthonormality property the following evolution equation is obtained

$$q_i^T e_{k+1} = (1 - \beta) q_i^T e_k \tag{7.18}$$

Clearly if $0 < \beta < 2$ then $(q_i^T e_{k+1})^2$ monotonically converges for $e_k \neq 0$. The proof concludes by noting that

$$\|e_{k+1}\|^2 = \sum_{i=1}^{N} (q_i^T e_{k+1})^2 \tag{7.19}$$

and hence if $(q_i^T e_{k+1})^2$ monotonically converges for all $i \in \{1, 2, \ldots, N\}$ then the same is true for $\|e_{k+1}\|^2$. □

7.4 Robustness of SVD for ILC

The previous section assumed that the singular value characteristics of the plant G are known precisely. In many scenarios this is not the case and only a nominal model G_o is available. If a multiplicative uncertainty U relates G and G_o(i.e. $G = UG_o$) then the SVD relation in (7.14) becomes:

$$G = UQSV^T \tag{7.20}$$

The following proposition puts a necessary condition upon this uncertainty for monotonic error convergence.

Proposition 7.2 *If $U + U^T$ is a positive-definite matrix, then there exists a sufficiently small $\bar{\beta}$ where $\overline{\beta} > \beta > 0$ such that $\|e_{k+1}\| < \|e_k\| \quad \forall \quad e_k \neq 0$.*

Proof. The proof is easiest done using the super vector notation where

$$u_{k+1} = u_k + \beta V S^{-1} Q^T e_k \tag{7.21}$$

The error evolution equation becomes:

$$e_{k+1} = (I - \beta G V S^{-1} Q^T) e_k = (I - \beta U) e_k \tag{7.22}$$

The error evolution equation is identical to that of the inverse algorithm detailed in Chapter 4, hence the existence of $\bar{\beta}$ such that $\|e_{k+1}\| < \|e_k\|$ for all $e_k \neq 0$ is clear. □

Remark 7.1 *The result concurs with thesufficient condition found in Chapter 4. However for some systems it is unrealistic to assume that the multiplicative uncertainty is positive-definite. Typically the condition is broken when high frequency modelling cannot be performed.*

The feedforward type algorithms presented in Chapter 4 have no obvious means of relaxing the positivity condition for monotonic convergence. In (Hara *et al.*, 1988) and (Tomizuka *et al.*, 1989) *Q-filtering* is applied to attain stability for repetitive control (a control problem with analogies to ILC). Inspired by this (Norrlöf and Gunnarsson, 2002) suggest the use of a *Q-filter* such that the plant input, u, becomes $u = Qu_{ILC}$ where u_{ILC} can be thought of as the input generated by an ILC algorithm. However the analysis presented in the paper is not clearly related to the positivity conditions for robust monotone convergence developed in Chapter 4 and also does not give any insight into the design of appropriate *Q-filters.*

The remainder of this chapter introduces basis functions as a means to design filters that relax the robustness conditions developed in Chapter 4.

7.5 Using Basis functions as filters

This section describes how a reduced set of orthonormal basis functions can be used as a filter.

7.5.1 Orthonormal filters

Consider the $N \times p$ matrix Q_p where

$$Q_p = [q_1, q_2, \ldots, q_{p-1}, q_p] \tag{7.23}$$

and the set $\mathcal{Q}_p$ is a subset of the orthonormal basis $\mathcal{Q}$. Note that the ordering of the vectors 1 to p, in (7.23), is done here purely for notational simplicity and that any subset of $\mathcal{Q}_p$ deemed appropriate can be chosen.

In the following equation,

$$x_f = Q_p Q_p^T x \tag{7.24}$$

the $N \times N$ matrix $Q_p Q_p^T$ acts on an arbitrary vector $x \in \mathbb{R}^N$ to produce a filtered vector $x_f \in \mathcal{W} \subseteq \mathbb{R}^N$. The filter $Q_p Q_p^T$ only allows components of x that are members of $\eta \mathcal{Q}_p$, where $\eta \in \mathbb{R}$, to pass through to x_f. Due to the orthonormality of $\mathcal{Q}$, these components pass through unchanged.

Example 7.1 *$\mathcal{Q}\{q_1, q_2, \ldots, q_{N-1}, q_N\}$ is an orthonormal basis. $\mathcal{Q}_p = \{q_1\}$ is a subset of $\mathcal{Q}$ and the vector x is chosen to be $x = \eta q_1 + q_z$ where $2 \leq z \leq N$. Due to the orthonormality of $\mathcal{Q}$ the filtered vector x_f becomes:*

$$x_f = Q_p Q_p^T x = Qp(\eta q_1^T q_1 + q_1^T q_z) = Q_p \eta = \eta q_1 \tag{7.25}$$

Figure 7.1 shows a plot of x and x_f. From this figure it is clear that q_z is not an element of $\mathcal{Q}_p$ and has been filtered out.

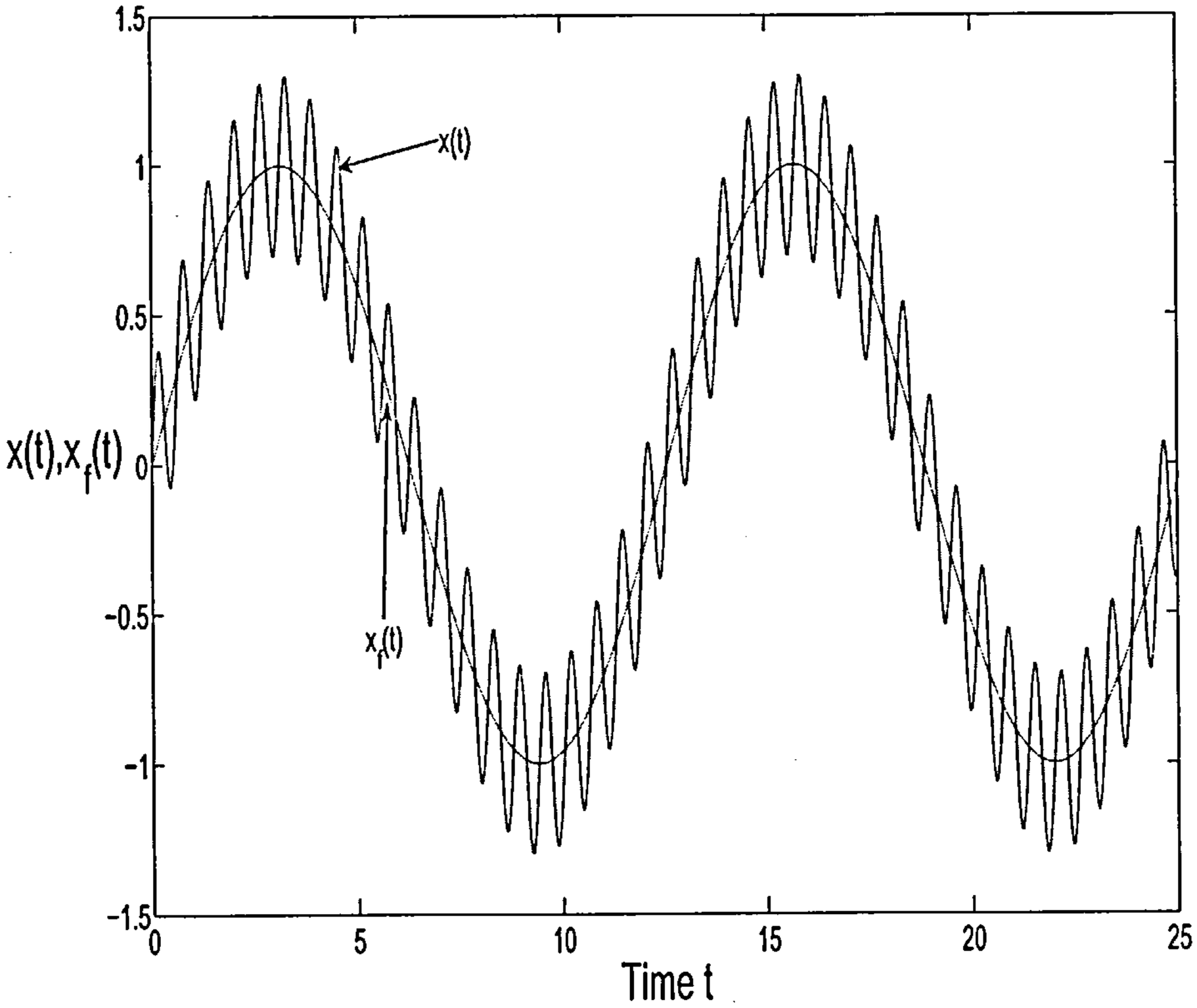

Figure 7.1: Comparison of filtered and unfiltered signals $x_f(t)$ and $x(t)$

7.5.2 Filtered Error and Component Error

At this juncture it is necessary to introduce two pieces of terminology, namely the *filtered error* and *component error*, both of which shall be used throughout the remainder of this chapter.

Filtered Error

From herein the $N \times 1$ vector $Q_p Q_p^T e_k$ shall be termed the *filtered error.*

It may be useful to consider the convergence properties of $\|Q_p Q_p^T e_k\|$

particularly if the filter $Q_p Q_p^T$ could be designed such that $GQ_pQ_p^Tu^* \approx Gu^*$ where u^* is the input that minimises $\|r - Gu\|$.

A point of caution should be observed however. Since $\mathcal{Q}_p \subset \mathcal{Q}$ the matrix $Q_pQ_p^T$ has p eigenvalues equaling 1 and $N - p$ eigenvalues equaling zero resulting in the following inequality:

$$\|Q_pQ_p^Te_k\| \leq \|e_k\| \tag{7.26}$$

The consequence of (7.26) is that convergence of $\|Q_pQ_p^Te_k\|$ does not immediately imply convergence of $\|e_k\|$.

Component error

From herein the $p \times 1$ vector $Q_p^Te_k$ shall be termed the *component error.* This vector can be thought of as the projection of the error onto each of the p basis vectors. It may also be useful to consider the convergence of $\|Q_p^Te_k\|$ since it has the following property:

$$\|Q_p^Te_k\| = \|Q_pQ_p^Te_k\| \tag{7.27}$$

The property can be obtained by observing that the $p \times p$ matrix $Q_p^TQ_p$ is in fact the identity matrix. This is due to the orthonormality of $\mathcal{Q}_p$.

7.6 Relaxed Basis SVD ILC

Suppose now that the plant multiplicative uncertainty $U + U^T$ is no longer positive-definite and is now characterised by three spaces $\mathcal{W}_{ps}$, $\mathcal{W}_z$ and $\mathcal{W}_n$ where

$$\begin{aligned}
\mathcal{W}_{ps} &= \{w_{ps}| \quad w_{ps}^T(U + U^T)w_{ps} > 0\} \\
\mathcal{W}_z &= \{w_z| \quad w_z^T(U + U^T)w_z = 0\} \\
\mathcal{W}_n &= \{w_n| \quad w_n^T(U + U^T)w_n < 0\}
\end{aligned} \tag{7.28}$$

and $\mathcal{W}_{ps} \bigoplus \mathcal{W}_z \bigoplus \mathcal{W}_n = \mathbb{R}^N$.

When using the algorithm in (7.16) if $QQ^T e_k = e_k \in \mathcal{W}_n$, for some arbitrary $e_k \in \mathbb{R}^N$, then the analysis in section 7.4 shows that the convergence of e_k would not be monotonic and there would be no guarantee of stability. This section presents a relaxed version of the algorithm in (7.16) which uses an orthonormal filter $Q_p Q_p^T : \mathbb{R}^N \rightarrow \mathcal{W}_{ps} \bigoplus \mathcal{W}_z$. The idea here is that if $QQ^T e_k = e_k \in \mathcal{W}_{ps}$, for some arbitrary $e_k \in \mathbb{R}^N$, then there exists a $\beta > 0$ such that the filtered error signal monotonically converges to zero. If $QQ^T e_k = e_k \in \mathcal{W}_z$, for some arbitrary $e_k \in \mathbb{R}^N$, then the algorithm will have converged to a fixed limit $E = e_k$.

The modified update law is given by

$$u_{k+1} = V_p \alpha_{k+1} \tag{7.29}$$

where $V_p = [v_1, v_2, \ldots, v_{p-1}, v_p]$, $p \leq N$ and

$$\alpha_{k+1} = \alpha_k + \beta S_p^{-1} Q_p^T e_k \tag{7.30}$$

Q_p and S_p are given as follows

$$Q_p = [q_1, q_2, \ldots, q_{p-1}, q_p] \tag{7.31}$$

and

$$S_p = \begin{bmatrix} \sigma_1 & 0 & \ldots & \ldots & 0 \\ 0 & \sigma_1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \sigma_{p-1} & 0 \\ 0 & \ldots & \ldots & 0 & \sigma_p \end{bmatrix} \tag{7.32}$$

It is worth noting that V_p and Q_p are reduced sets, of the type discussed in section 7.5.1, of the singular vectors defined in section (7.3) and are related by $GV_p = UQ_pS_p$. Note that just as in section 7.5.1 the ordering of the

vectors from 1 to p is done here purely for notational simplicity and that any set of p singular vectors deemed appropriate by the user can be chosen.

The error evolution equation for the algorithm in (7.29) becomes

$$e_{k+1} = e_k - \beta U Q_p Q_p^T e_k \tag{7.33}$$

The analysis of the method starts by considering the component error $Q_p^T e_k$ where

$$Q_p^T e_{k+1} = (I - \beta Q_p^T U Q_p) Q_p^T e_k \tag{7.34}$$

The following proposition gives a condition for monotonic convergence of the component error, $Q_p^T e_k$.

Proposition 7.3 *If $Q_p^T(U + U^T)Q_p$ is a positive-definite matrix then there exists a sufficiently small $\beta > 0$ such that $\|Q_p^T e_k\|$ converges to zero monotonically.*

Proof. The proof becomes identical to that of proposition 4.5 by making following substitutions, $e_k = Q_p^T e_k$ and $U = Q_p^T U Q_p$. □

The result in proposition 7.3 does not fully describe the mechanism within the relaxed algorithm since it is not clear when $Q_p^T(U + U^T)Q_p$ is a positive-definite matrix.

Proposition 7.4 *There exists a sufficiently small $\beta > 0$ such that if $Q_p Q_p^T e_k \in \mathcal{W}_{ps} \quad \forall \quad Q_p Q_p^T e_k \neq 0$ then $Q_p^T e_k$ converges to zero monotonically. If $Q_p Q_p^T e_k = 0$ then $Q_p^T e_k$ remains at zero.*

Proof Continuing on from the proof of proposition 7.3 it is necessary that

$$e_k^T Q_p Q_p^T (U + U^T) Q_p Q_p^T e_k > 0 \quad \forall \quad Q_p Q_p^T e_k \neq 0 \tag{7.35}$$

and hence it is necessary that $Q_p Q_p^T e_k$ be an element of $\mathcal{W}_{ps}$ for all $Q_p Q_p^T e_k \neq 0$. The existence of the sufficiently small $\beta > 0$ such that convergence is

monotonic to zero follows from the proofs of propositions 4.2 and 4.3. Finally it is trivial to see that if $Q_p Q_p^T e_k = 0$ then $\|Q_p^T e_k\| = 0$. □

Remark 7.2 *Proposition 7.4 shows that if the filter $Q_p Q_p^T$ can be selected such that it maps all error signals to the set $\mathcal{W}_{ps}$ where possible, and to $\mathcal{W}_z$ otherwise, i.e. $Q_p Q_p^T$ is a matrix mapping such that $Q_p Q_p^T : \mathbb{R}^N \rightarrow \mathcal{W}_{ps} \bigoplus \mathcal{W}_z$, then the filtered error converges to zero monotonically.*

It is insufficient to just consider the performance of $Q_p^T e_k$, the stability of e_k must be analysed to guarantee system integrity.

Proposition 7.5 *If $\lim_{k\rightarrow\infty} Q_p^T e_k = 0$ then e_k converges to a limit E such that $\|E\| < \infty$.*

Proof If $\lim_{k\rightarrow\infty} Q_p^T e_k = 0$ then it follows that $\alpha_{k+1} = \alpha_k$ and $u_{k+1} = u_k$ as $k \rightarrow \infty$ where u_k is bounded such that $\lim_{k\rightarrow\infty} \|u_k\| < \infty$. Hence the output error has converged to some limit E.

The value of E is given by $E = \lim_{k\rightarrow\infty} r - Gu_k$. Since both $\lim_{k\rightarrow\infty} u_k$ and r are bounded and G is BIBO stable by assumption then E must be bounded such that $\|E\| < \infty$. □

Remark 7.3 *The limiting error E is heavily dependent upon the relationship between r and V_p and Q_p. However the greater number of error signals e_k that are mapped to the space $\mathcal{W}_{ps}$, and hence the fewer mapped to $\mathcal{W}_z$, then the smaller E will be. This 'rule of thumb' is demonstrated by a simulation example in section 7.9 using an algorithm similar to that discussed in this section.*

7.7 Orthonormal Filters for Feedforward-type ILC

In this section an ILC algorithm is presented that attempts to map the error signal e_k to the space $\mathcal{W}_{ps}$, defined in (7.28), where possible and to $\mathcal{W}_z$ otherwise. Consider the following update law

$$u_{k+1} = u_k + \beta K V_p V_p^T e_k \tag{7.36}$$

where V_p is a reduced set of orthonormal basis vectors of the type discussed in section 7.5.1. The structure of the algorithm in (7.36) is the same as the one presented in section 4.2, except that e_k has been replaced with a filtered error, $V_p V_p^T e_k$.

This section proceeds by analysing the convergence properties of the component error, $V_p^T e_k$, when the matrix GK is characterised by three spaces $\tilde{\mathcal{W}}_{ps}$, $\tilde{\mathcal{W}}_z$ and $\tilde{\mathcal{W}}_n$:

$$\begin{aligned}
\tilde{\mathcal{W}}_{ps} &= \{\tilde{w}_{ps} | \quad w_{ps}^T \big(GK + (GK)^T\big) \tilde{w}_{ps} > 0\} \\
\tilde{\mathcal{W}}_z &= \{\tilde{w}_z | \quad w_{zz}^T \big(GK + (GK)^T\big) \tilde{w}_z = 0\} \\
\tilde{\mathcal{W}}_n &= \{\tilde{w}_n | \quad \tilde{w}_n^T \big(GK + (GK)^T\big) \tilde{w}_n < 0\}
\end{aligned} \tag{7.37}$$

where $\tilde{\mathcal{W}}_{ps} \bigoplus \tilde{\mathcal{W}}_z \bigoplus \tilde{\mathcal{W}}_n = \mathbb{R}^N$. The idea here is to design the matrix $V_p V_p^T$ such that it is a matrix mapping $V_p V_p^T : \mathbb{R}^N \to \tilde{\mathcal{W}}_{ps} \bigoplus \tilde{\mathcal{W}}_z$ so that if $V_p V_p^T e_k \in \tilde{\mathcal{W}}_{ps}$ then there exists a $\beta > 0$ such that $V_p^T V_p e_k$ monotonically converges to zero.

The following proposition shows that if $V_p V_p^T e_k \in \tilde{\mathcal{W}}_{ps}$ for all $V_p V_p^T e_k \neq 0$ then $\|V_p^T e_k\|$ converges monotonically to zero.

Proposition 7.6 *If $V_p V_p^T e_k \in \tilde{\mathcal{W}}_{ps}$ for all $V_p V_p^T e_k \neq 0$ then there exists a sufficiently small $\beta > 0$ such that $V_p^T e_k$ monotonically converges to zero. Furthermore if $V_p V_p^T e_k = 0$ then $\|V_p^T e_k\| = 0$.*

Proof The component error evolution equation for (7.36) becomes as follows:

$$V_p^T e_{k+1} = (I - \beta V_p^T G K V_p) V_p^T e_k \tag{7.38}$$

The analysis in Chapter 4 comprehensively shows that if the following inequality holds then a sufficiently small $\beta > 0$ exists.

$$e_k^T V_p V_p^T \left(GK + (GK)^T\right) V_p V_p^T e_k > 0 \quad \forall \quad V_p V_p^T e_k \neq 0 \tag{7.39}$$

Clearly (7.39) holds true if $V_p V_p^T e_k \in \tilde{\mathcal{W}}_{ps}$ for all $V_p V_p^T e_k \neq 0$.

The convergence of $\|V_p^T e_k\|$ to zero as $k \to \infty$ follows from the proof of proposition 4.3. Finally it is trivial to see that if $V_p V_p^T e_k = 0$ then $\|V_p^T e_k\| = 0$. □

It is insufficient to just consider the performance of $V_p^T e_k$, the stability of e_k must be analysed to guarantee system integrity.

Proposition 7.7 *If* $\lim_{k\to\infty} V_p^T e_k = 0$ *then* e_k *converges to a limit* E *such that* $\|E\| < \infty$.

Proof If $\lim_{k\to\infty} V_p^T e_k = 0$ then it follows that $u_{k+1} = u_k$ as $k \to \infty$ where u_k is bounded such that $\lim_{k\to\infty} \|u_k\| < \infty$. Hence the output error has converged to some limit E.

The value of E is given by $E = \lim_{k\to\infty} r - Gu_k$. Since both $\lim_{k\to\infty} u_k$ and r are bounded and G is BIBO stable by assumption then E must be bounded such that $\|E\| < \infty$. □

The following two subsections discuss the case where such filtering is used on the inverse and adjoint algorithms, presented in Chapter 4, where $K = G_o^{-1}$ or $K = G_o^T$. The discussion focuses on a relationship between the spaces $\mathcal{W}_{ps}$ and $\tilde{\mathcal{W}}_{ps}$, and hence the relationship between the two algorithms and the model uncertainty.

7.7.1 Filtered Inverse Algorithm

For the inverse algorithm $GK = U$, hence $\mathcal{W}_{ps} = \tilde{\mathcal{W}}_{ps}, \mathcal{W}_z = \tilde{\mathcal{W}}_z$ and $\mathcal{W}_n = \tilde{\mathcal{W}}_n$. It is therefore sufficient that $V_p V_p^T e_k \in \mathcal{W}_{ps}$ for all $V_p V_p^T e_k \neq 0$ for proposition 7.6 to be held true.

7.7.2 Filtered Adjoint Algorithm

For the adjoint algorithm $GK = G_o U G_o^T$, hence the following:

$$\begin{aligned} \tilde{\mathcal{W}}_{ps} &= \{\tilde{w}_{ps} | \quad w_{ps}^T G_o (U + U^T) G_o^T \tilde{w}_{ps} > 0\} \\ \tilde{\mathcal{W}}_z &= \{\tilde{w}_z | \quad w_z^T G_o (U + U^T) G_o^T \tilde{w}_z = 0\} \\ \tilde{\mathcal{W}}_n &= \{\tilde{w}_n | \quad w_n^T G_o (U + U^T) G_o^T \tilde{w}_n < 0\} \end{aligned} \tag{7.40}$$

For proposition 7.6 to be held true then $G_o^T V_p V_p^T e_k \in \mathcal{W}_{ps}$ for all $V_p V_p^T e_k \neq 0$. Should $G_o^T V_p V_p^T e_k = 0$ then $\|G_o^T V_p V_p^T e_k\| = 0$ which implies $\|V_p^T e_k\| = 0$ since $G_o G_o^T$ and $V_p V_p^T$ are positive-definite matrices.

7.8 Selection of V_p

Clearly the result obtained in proposition 7.6 does not suggest how to select V_p in order for the assumption $V_p V_p^T e_k \in \tilde{\mathcal{W}}_{ps}$ for all $V_p V_p^T e_k \neq 0$ to be held true. However if the reference signal is restricted to directions in which the designer has *good knowledge* of the plant dynamics then ILC can be restricted to those directions. In the case where r has *low frequency* content and the designer has poor plant knowledge at *high frequency* then the Fourier basis V_f, given below, provides a simple means of selecting V_p such that it rejects error signals of specified frequencies. Section 7.9 gives a simulation example that demonstrates this point.

Fourier basis V_f:

$$V_f = [cos(0), cos(\gamma t), sin(\gamma t), cos(2\gamma t), sin(2\gamma t), \ldots, \\ \ldots, cos((\tfrac{N}{2}-1)\gamma t), sin((\tfrac{N}{2}-1)\gamma t), cos(\tfrac{N}{2}\gamma t)] \tag{7.41}$$

where $\gamma = \frac{2\pi}{T_f}$, T_f is the length of the trial and N is the number of samples in the trial. Note that this basis is orthogonal but can be normalised as discussed in section 7.1.

It is also worth noting that this method of ILC also has the bonus of rejecting disturbances outside the range of the reference signal. The author of this thesis leaves a full and thorough investigation of this and the subject of selecting V_p in general as a subject of future work.

7.9 Simulation Example

Consider the case where a real plant $G(s)$ is approximated with a lower order system $G_o(s)$. In the s-domain, these are taken to be

$$G(s) = \frac{1}{s+1}\left(\frac{20}{s+20}\right)^2 \tag{7.42}$$

$$G_o(s) = \frac{1}{s+1} \tag{7.43}$$

The two systems are sampled with a sample time of 0.1 seconds using zero-order hold. The trial length T_f is 25.5 seconds. It is easy to check numerically that the resulting multiplicative uncertainty matrix $U + U^T$ is not positive-definite. The reference signal is chosen to be $r(t) = sin(\frac{2\pi}{T_f}t)$.

The standard inverse algorithm, given in section 4.3, is run with $\beta = 0.5$ and an initial guess of $u_0 = 0$. Figure 7.2 shows, as the theory in Chapter 4 suggests, that $\|e_k\|$ does not monotonically converge. In fact after an initial convergence the system becomes *unstable* and the output diverges. The result

implies the error, e_k, has components in $\mathcal{W}_n$ that, no matter how small they are initially, grow as k increases.

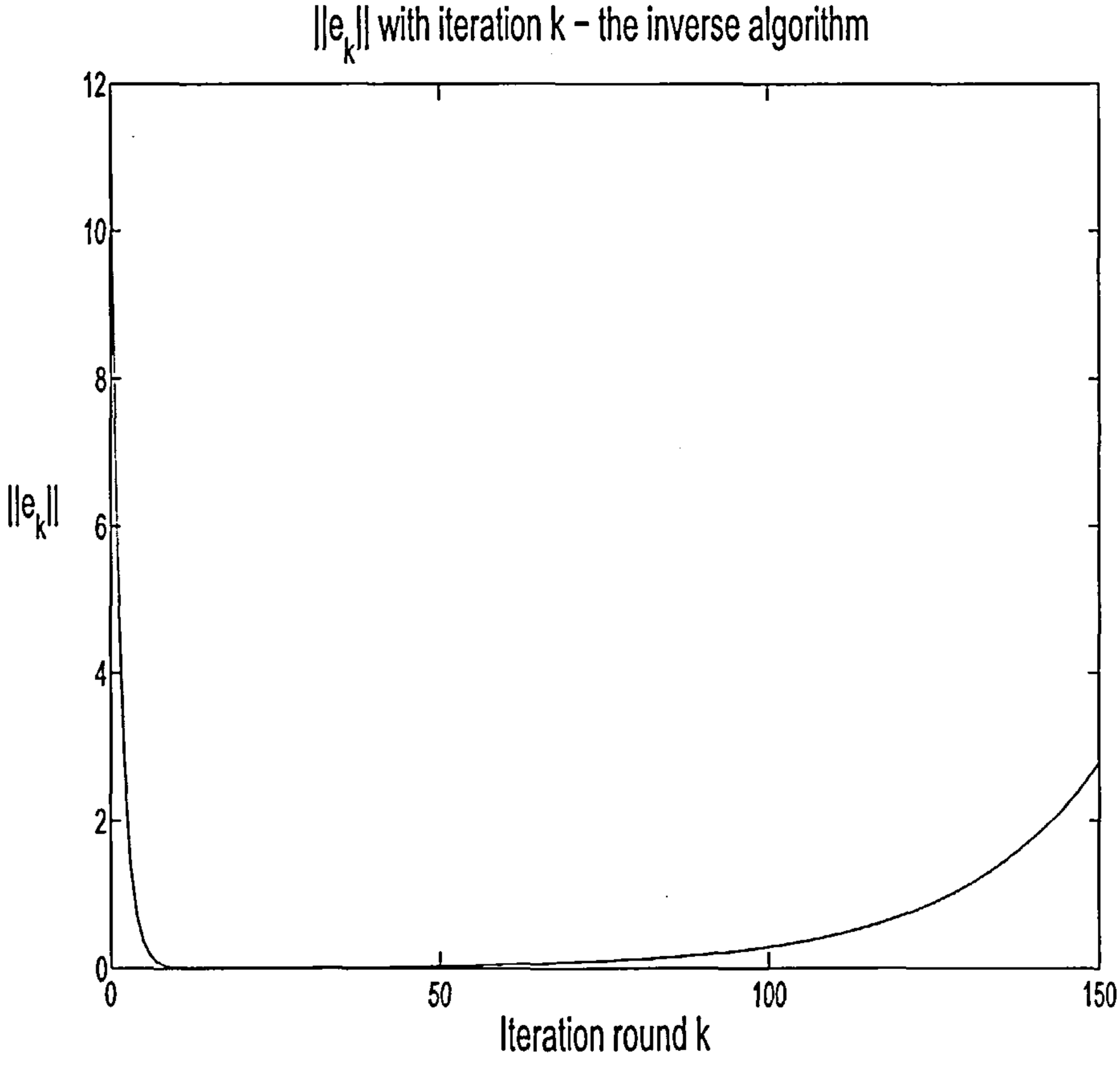

Figure 7.2: Divergence of error for inverse algorithm

The simulation is repeated but this time with the filtered inverse algorithm in (7.36) where $K = G_o^{-1}$ and the normalised Fourier basis in (7.41) is chosen where $p = 21$. Figure 7.3 shows that the filtered error monotonically converges to zero. The result implies that all non-zero filtered errors lie in $\mathcal{W}_p$.

Figure 7.4 shows a logarithmic plot of $\|e_k\|$ for both the inverse and

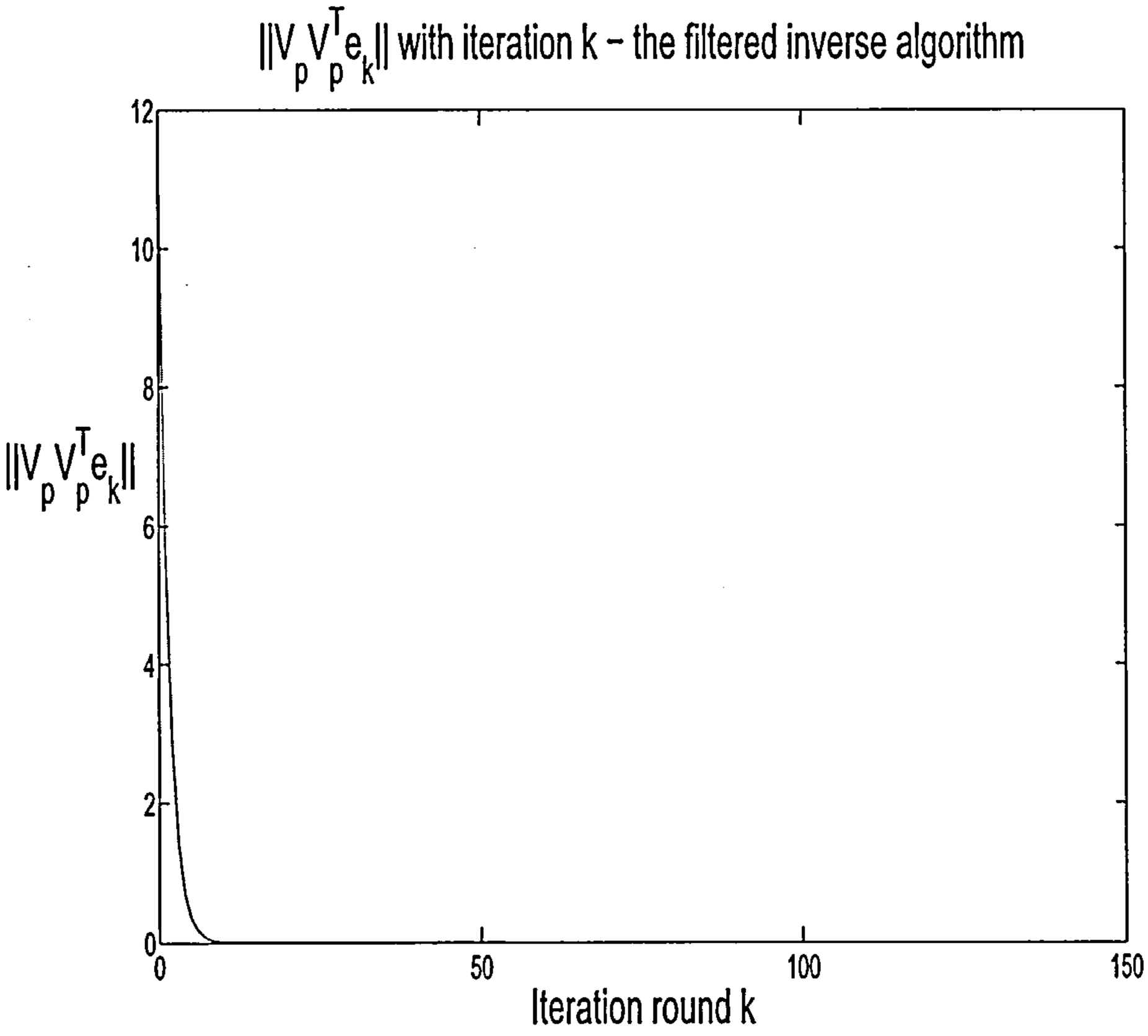

Figure 7.3: Convergence of the filtered error signal using the filtered inverse algorithm

filtered inverse algorithms. During the early iterations the two plots are visually indistinguishable. This concurs with the theory that if $e_k \in \mathcal{W}_{ps}$ the two algorithms are effectively the same. However from about iteration 12 onwards the inverse algorithm diverges whereas the filtered inverse algorithm converges to a fixed limit E. This demonstrates that the inverse algorithm feeds elements of $\mathcal{W}_n$ into the plant whereas the filtered algorithm rejects them.

Figure 7.5 shows the reference signal and the output, y_{150}, are visually indistinguishable. This implies that for this example the fixed error limit E is negligible.

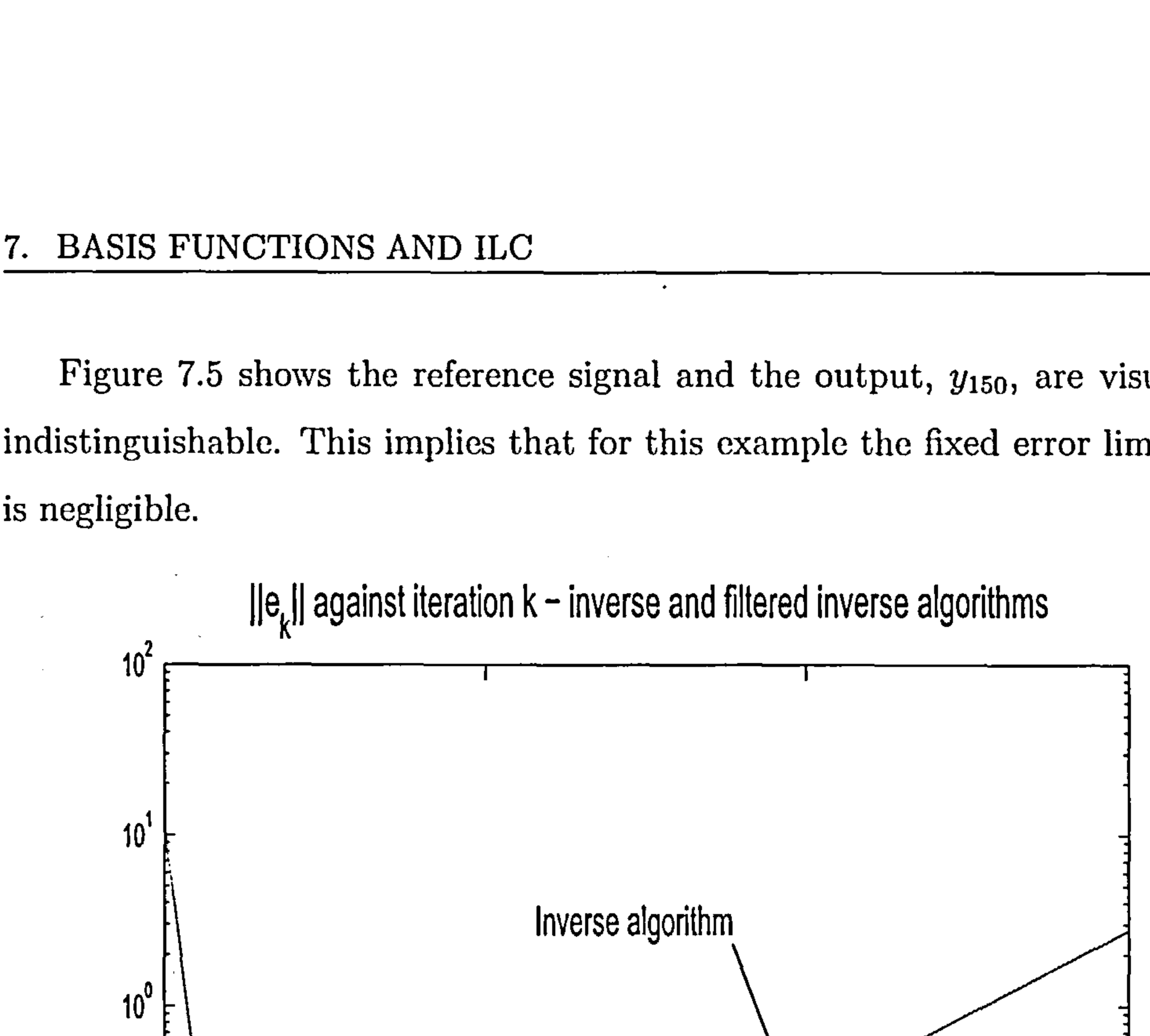

Figure 7.4: Comparison of error convergence for inverse and filtered inverse algorithms

Finally the case where a one-off additive measurement disturbance is simulated. It is assumed that perfect tracking has already been achieved, i.e. $e_0 = 0$, when the signal $d(t) = 10v_{255}$ is injected into the output measurement signal.

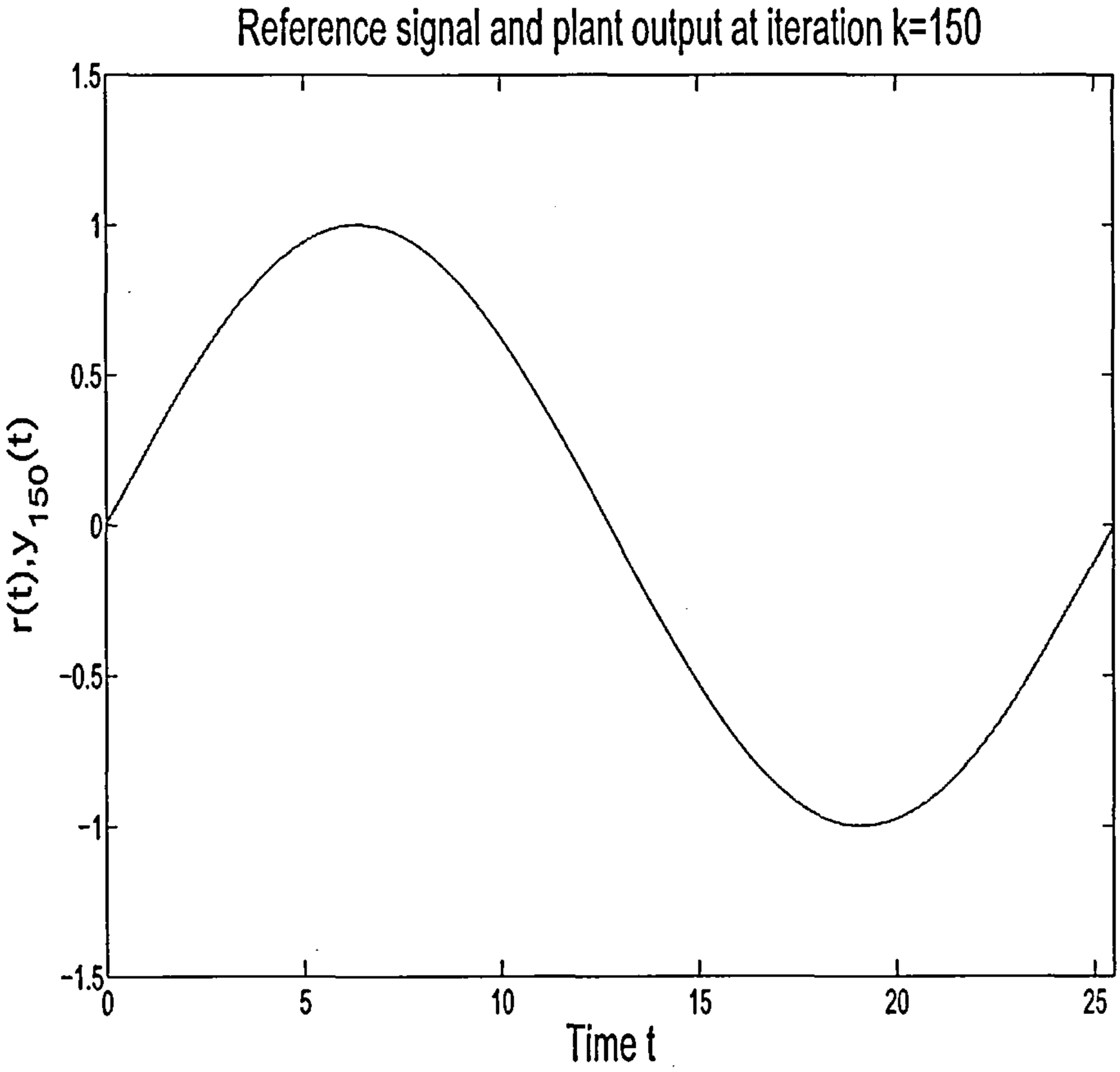

Figure 7.5: Comparison of reference and output signals after 150 iterations

Figure 7.6 shows a plot of $r(t) + \beta G G_o^{-1} d(t)$, the resulting output of the inverse algorithm, and figure 7.7 show a plot of $r(t) + \beta G G_o^{-1} V_p V_p^T d(t)$, the resulting output for the filtered algorithm. As the theory suggests the filtered algorithm rejects the disturbance. This demonstrates the potential benefits of the filtered algorithm when the plant is subject to output disturbances and measurement noise. This result motivates analysis of this rejection characteristic as a future piece of work.

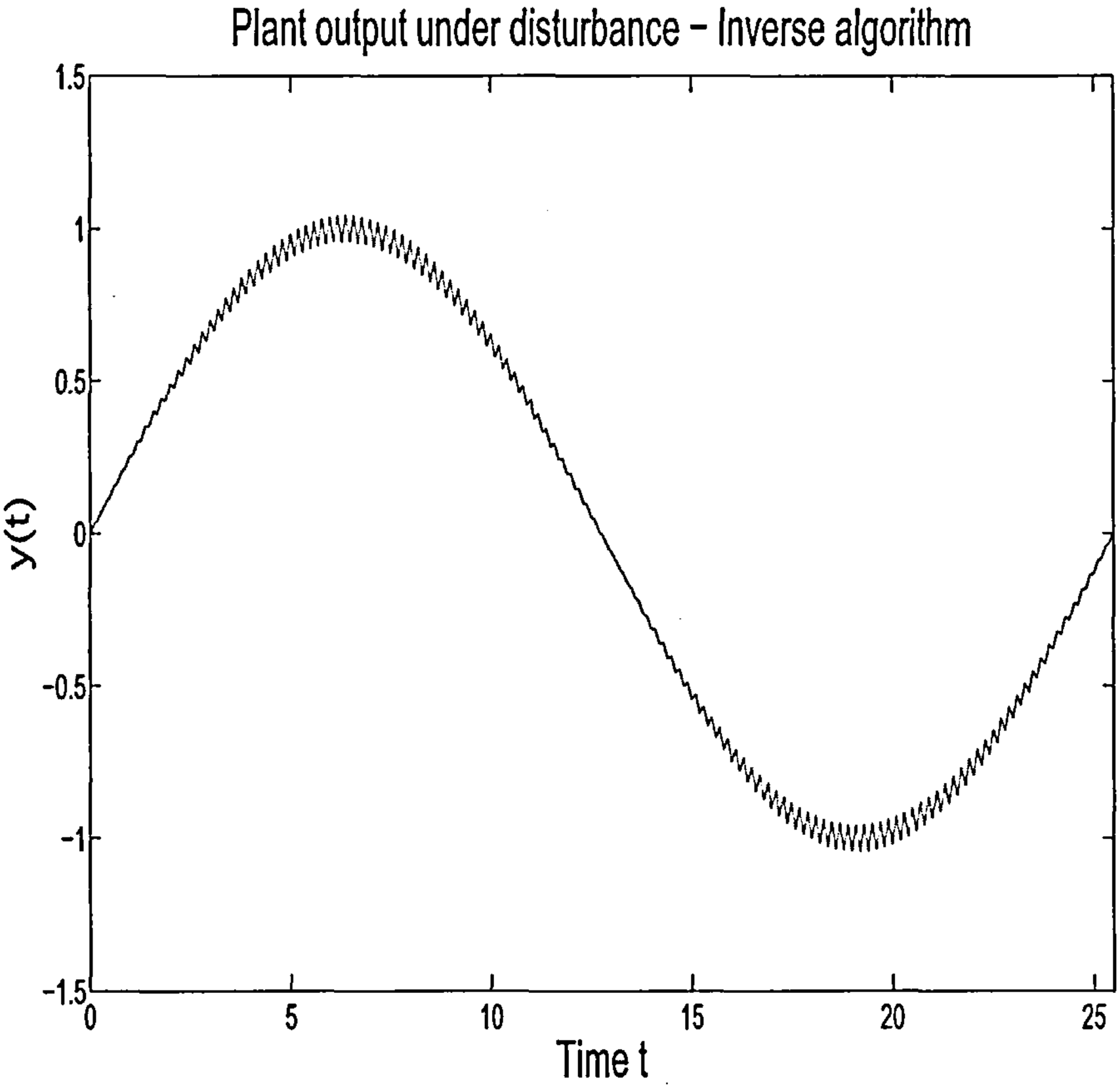

Figure 7.6: Plant output subject to disturbance using the inverse algorithm

7.10 Summary

In this chapter orthonormal basis functions were proposed as non-causal filters for use in feed-forward type ILC algorithms. Non-causal filters are applicable in ILC as signal processing can be applied off-line in between iterations. Orthonormal filters were shown to have the property of being able to either pass through signals unchanged or to entirely reject them. This is a property causal filters do not possess. Two sets of orthonormal basis functions that are commonly used in control theory and signal processing, namely the Singular

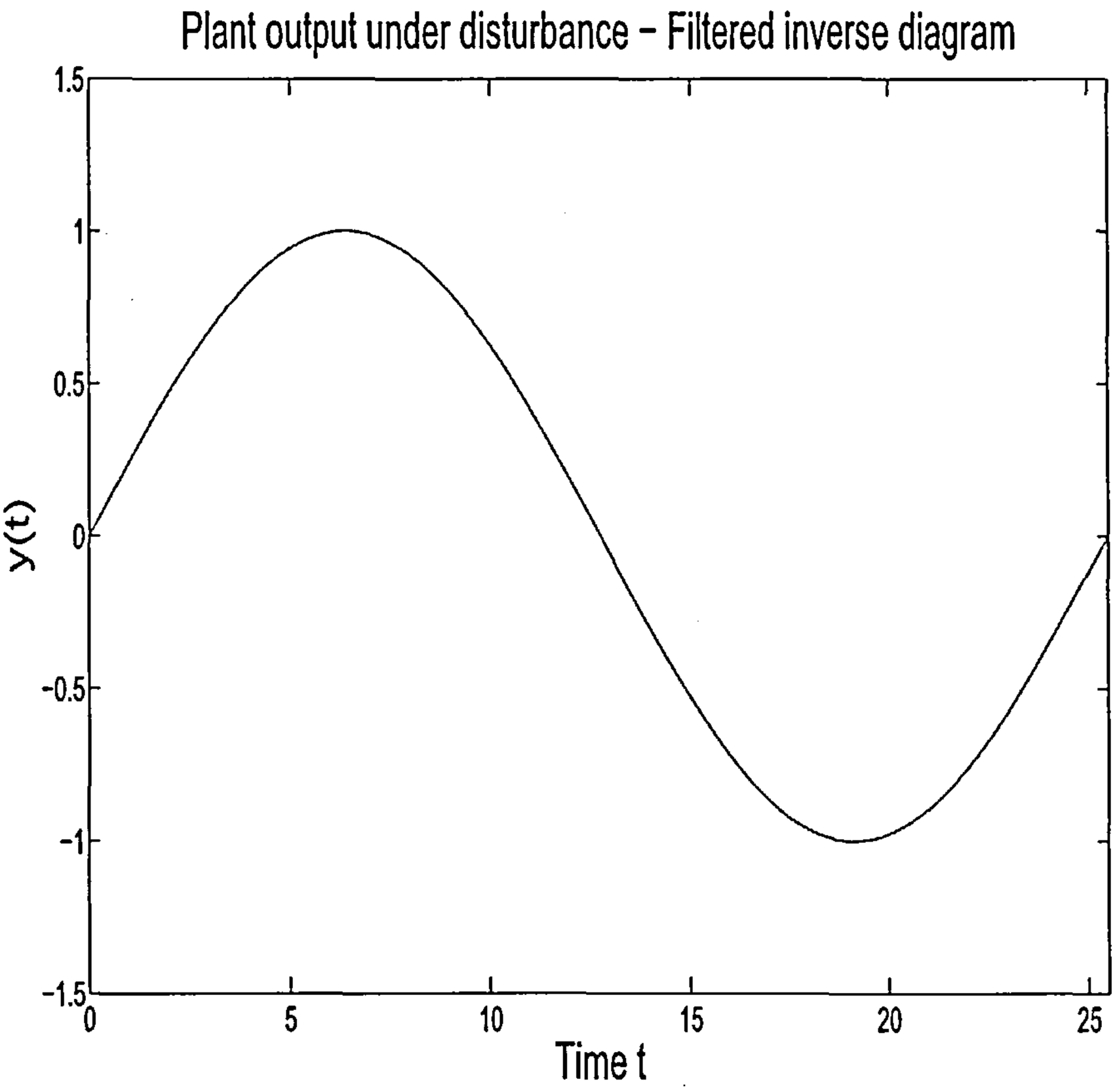

Figure 7.7: Plant output subject to disturbance using the filtered inverse algorithm

Value and Fourier bases, were put forward as suitable basis from which to design filters.

Multiplicative model uncertainty was categorised into three modes: a positive mode, a zero (or nullifying) mode and a negative mode. In the face of uncertainty with negative modes the conventional inverse and adjoint algorithms tend to suffer from long term instability. A filtered version of these two algorithms was proposed that tackled this stability problem by passing

through error signals that excite positive modes whilst rejecting error signals that excite negative (or non-monotonic) modes. It was shown that this addition of an orthonormal filter opened up a method by which to regain stability where the output error converges to a fixed limit. It was further shown that if the orthonormal filter is designed appropriately then the filtered error has robust monotone convergence in the presence of non-positive multiplicative uncertainty.

This result is a significant improvement for inverse and adjoint ILC and gives a method of regaining monotonic behaviour in the face of significant model uncertainty. This result also highlights two key issues for feedforward type ILC:

- Passing through (to the plant) error signals that excite positive modes of plant uncertainty allows for robust monotone convergence.

- Passing through error signals that excite negative modes of the plant uncertainty leads to a loss of monotonicity and possibly induces instability.

A simulation example demonstrated the ease of implementation of the filtered algorithm and further demonstrated that the limiting error need not be significant. The simulation example also highlighted the algorithm's potential ability to reject disturbances. A full and thorough investigation into the algorithm's ability to reject output disturbances and measurement noise would complement the results presented in this chapter and is an area suggested for further work. The exact choice of filter once given an orthonormal basis, i.e. how many and which basis vectors to use, is also a subject for future work as is the relationship between the limiting error and the model uncertainty.

Chapter 8

Conclusions and Future work

8.1 Overview

In this thesis new robustness analysis for model-based Iterative Learning Control (ILC) has been derived. In ILC a dynamical system is required to track a reference in a repetitive manner. The repetitive nature of the problem allows use of information from previous attempts to track the reference in order to improve future attempts. An ILC algorithm iteratively uses past tracking errors to modify input signals until perfect, or near perfect, tracking is achieved.

This iterative process creates a two-dimensional system where the time-axis is fixed and finite and the iteration axis is infinite. The output of an LTI system cannot become unbounded in finite time hence the stability of the 2-D process is typically characterised by its behaviour along the iteration axis. Stability analysis of ILC tends to concentrate on the learning operator L that maps the error signal from the current iteration to the next iteration. The stability of an ILC algorithm can be determined by considering the spectral radius $\rho(L)$ of the learning operator L and the induced norm $\|L\|$. If the

spectral radius is strictly less than unity then the algorithm asymptotically converges to zero error. If the induced norm $\|L\|$ is strictly less than unity then the algorithm monotonically converges to zero tracking error.

Asymptotic convergence is not necessarily a desirable trait for an ILC algorithm as it gives no guarantee of the input and output signals remaining within any required bounds. Monotonic convergence however implies tracking is improved, with respect to the norm, at each and every iteration. If the norm of the initial error is less than some required safety bound then all future errors will also be within this bound, hence monotonic convergence is a very desirable property for any ILC algorithm.

In this thesis monotonic convergence of ILC algorithms was considered. A number of ILC algorithms with monotonically converging characteristics were recognised to be model-based algorithms and accordingly robustness analyses were derived for these algorithms for the case when a perfect plant model is unavailable. More formally the concept of *robust monotone convergence* was introduced and considered, where a model-based ILC algorithm is required to give monotonic convergence in spite of plant uncertainties in the model.

8.2 Parameter Optimal ILC

A number of optimisation based ILC algorithms are known to result in monotonic convergence. However they tend not to be without drawbacks. Norm-Optimal ILC (NOILC) for example gives monotonic convergence but is non-causal in implementation and requires state-observation and prediction. The algorithm is heavily reliant upon a plant model and is complex in implementation. As a result of this there is no clear method of performing a robustness analysis.

Another optimisation based algorithm, termed Parameter Optimal ILC (POILC), also results in monotonic convergence but is simpler in form than NOILC. However the POILC algorithm struggles to guarantee convergence to zero error and is heavily reliant upon having a perfect knowledge of the plant dynamics. The simpler structure of POILC does allow for a robustness analysis however conditions for robust monotone convergence appear quite restrictive and fail to characterise the plant uncertainty tolerable for robust monotone behaviour. The analysis of POILC motivates the search for model-based ILC algorithms that are simple in structure and that allow conditions for robust monotone convergence to be derived in terms of plant uncertainty.

8.3 Time Domain conditions for convergence in ILC

In POILC and NOILC the optimal algorithms are model based but their structure appears to prevent conditions for robust monotone convergence to be developed in terms of plant uncertainties. A feedforward ILC algorithm can be used to tackle these issues. The feedforward ILC algorithm is of the form $u_{k+1} = u_k + \beta K e_k$ where β is a scalar, K is a square matrix and both are left open for design. The algorithm is simple in structure and allows the use of plant model information via the matrix K. Analysis of this algorithm derives a positivity condition for monotonic convergence in terms of β, K and the plant matrix G. Furthermore the condition guarantees convergence to zero tracking error.

If plant knowledge is utilised by selecting $K = G^{-1}$ or $K = G^T$ then the condition for monotonic convergence is immediately satisfied for some $\beta > 0$. The choices $K = G^{-1}$, termed the inverse algorithm, and $K = G^T$, termed

the adjoint algorithm, result in model-based algorithms. As a new result, also presented in publication P1 (Harte *et al.*, 2004), robustness analysis shows that if a multiplicative plant uncertainty U satisfies a positivity condition then both algorithms result in robust monotone convergence to zero error for some $\beta > 0$.

8.4 Frequency domain conditions for inverse and adjoint algorithms

The matrix based positivity conditions for robust monotone convergence provide insight into the type of model uncertainty tolerable for the inverse and adjoint algorithms. However it would be useful if the matrix based conditions could be translated into the frequency domain commonly used in control theory and practice. The Toeplitz nature of the uncertainty matrix U and the assumed stability of the underlying uncertainty transfer function $U(z)$ allows this to be done.

Through use of Parseval's relation, analysis shows that it is sufficient that $U(z)$ be positive real, for a sufficiently small $\beta > 0$, in order for the adjoint algorithm to give robust monotone convergence. A similar analysis yields a tighter condition for the inverse algorithm. Not only is it sufficient that $U(z)$ be positive real for robust monotone convergence but the condition also gives a relationship between $U(z)$ and the magnitude of β.

Both of these conditions result in useful and simple Nyquist plot interpretations. For the adjoint algorithm $U(z)$ must lie in the right-half complex plane and β must be made sufficiently small yet positive. For the inverse algorithm $U(z)$ must lie in the right-half complex plane but also within a circle of $\frac{1}{\beta}$ radius centred about the point $(0, \frac{1}{\beta})$ where $\beta > 0$. This analysis

was also presented, for the inverse algorithm, in publication P2 (Harte *et al.*, 2005).

8.5 Inverse and Adjoint POILC

The ideas in POILC can be applied to the inverse and adjoint ILC algorithms and provide a method of selecting the tuning parameter β so as to balance the dual aims of robust monotone convergence and convergence speed. The inverse and adjoint type POILC algorithms are both model-based and again yield positivity conditions upon the multiplicative plant uncertainty for robust monotone convergence. If the multiplicative plant uncertainty matrix, U, is positive-definite and a tuning parameter, w_{k+1}, is made sufficiently large then robust monotone convergence is guaranteed. This analysis was presented, for adjoint type POILC, in publication P3 (Hätönen *et al.*, 2003*b*).

The adjoint version of this algorithm has been applied on an industrial-scale gantry robot with good results. Despite the model neglecting inherent nonlinearities in the plant, the algorithm results in near perfect tracking after approximately 100 iterations.

8.6 Basis Functions and ILC

The iterative nature of ILC allows for offline signal processing to be performed on previous error signals in between trials. Offline signal processing has one major advantage over online signal processing in that it permits the use of non-causal (in the time-axis sense) filters.

The inverse and adjoint ILC algorithms require that a multiplicative plant uncertainty matrix be positive-definite in order to achieve robust monotone

convergence. As a new result it is shown that the addition of a non-causal filter to the inverse and adjoint algorithms allows for a relaxation of the positivity condition. More formally if the plant uncertainty matrix is not positive-definite then the tracking error can still converge to a fixed limit. This highlights a trade-off between the positivity of plant uncertainty and zero tracking error.

Analysis shows that the design of such a non-causal filter can be done using a set of orthonormal basis functions. Further analysis shows that a filtered error signal has robust monotone convergence to zero. Simulation studies demonstrate that the limiting tracking error need not be significantly large when using the filtered inverse and adjoint algorithms in scenarios where the standard inverse and adjoint algorithms suffer from long term instability. This analysis is an extension of the work presented in publication P4 (Hätönen *et al.*, 2004*b*).

8.7 Directions for future research

The work presented in this thesis opens up further directions for future research:

i) The analysis in Chapter 4 shows the inverse and adjoint algorithms to have a well defined tolerance to plant uncertainty for SISO systems. A similar analysis for multiple-input multiple-output (MIMO) implementations for both algorithms would complement the findings in this thesis. Potential issues could arise for the inverse MIMO algorithm if the MIMO plant matrix is not of full rank.

ii) The convergence analysis throughout this thesis has been performed using the l_2-norm. In some applications robust monotone convergence

with respect to some other norm may be desirable. Meaningful robustness conditions in terms of other norms would then be a useful contribution to the ILC literature. A starting point for such work could centre upon the rich structure of $\|U\|_\infty$ and $\|U\|_1$. Since U is lower-triangular and *Toeplitz* both norms are given by:

$$\|U\|_\infty = \|U\|_1 = \sum_{i=0}^{N-1} |m_i| \tag{8.1}$$

where $U(z) = \sum_{i=0}^{\infty} m_i z^{-i}$.

iii) The work in this thesis has assumed that at the end of each trial the system is perfectly reset such that $x_{k+1}(0) = x_0$. In a number of practical applications perfect resetting may not be possible so repeating the analysis presented in this thesis where imperfect resetting is assumed would be of value to existing ILC literature.

iv) The results derived in this thesis for the inverse and adjoint algorithms can be applied to repetitive control, a problem analogous to ILC. Premliminary works on inverse and adjoint repetitive control, by this author along with primarily J. Hätönen and D.H. Owens, can be found in (Hätönen *et al.*, 2004*a*) and (Freeman *et al.*, 2004). These works take a *sliding window* approach however one other repetitive control implementation can be considered. A batch-wise approach to repetitive control describes the output at the k^{th} batch as follows:

$$y_k = \sum_{i=0}^{\infty} G_i u_{k-i} \tag{8.2}$$

Such a description of repetitive control is highly analogous to *High-order* ILC, see (Norrlöf and Gunnarsson, 2006) for an example of High-order ILC, and provides a natural starting point for analysis of batch-wise inverse and adjoint repetitive control.

v) The analysis in Chapter 7 discusses the use of an appropriate subset of orthonormal basis vectors for filter design. Simulation studies infer methods for selecting the appropriate subset of basis vectors and also highlights the filter's ability to reject certain output disturbances. A full and thorough investigation of these issues would be a significant addition to the analysis presented in Chapter 7.

vi) ILC is an industrially inspired subject. Hence it is of great importance to experimentally verify the theoretical results presented in this thesis. This work has already commenced in publication P.3 where the adjoint algorithm was applied to a multi-axis robot and in (Ratcliffe *et al.*, 2004) where the inverse algorithm was applied to the same robot.

Bibliography

Amann, N., D.H. Owens and E. Rogers (1996). Iterative learning control using optimal feedback and feedforward actions. *International Journal of Control* **65**(2), 277–293.

Anderson, B.D.O and J.B. Moore (1989). *Optimal Control: linear quadratic methods.* Prentice Hall.

Arimoto, S. (1991). Passivity of robot dynamics implies capability of motor program learning. In: *Proceedings of the International Workshop on Nonlinear and Adaptive Control: Issues in Robotics.* Grenoble, France.

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

Arimoto, S., S. Kawamura, F. Miyazaki and S. Tamakie (1985). Learning control theory for dynamic systems. In: *Proceedings of the 24th IEEE Conference on Decsion and Control.* Florida, USA.

Athans, M. and P.L. Falb (1966). *Optimal Control: an introduction to the theory and its applications.* McGraw-Hill.

Bondi, P., G. Casalino and L. Gambardella (1988). On iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics and Automation* 4, 14–22.

Casalino, G. and G. Bartolini (1984). A learning procedure for the control of movements of robotic manipulators. In: *IASTED Symposium on Robotics and Automation.*

Edwards, J. B. and D.H. Owens (1982). *Analysis and Control of Multipass Processes.* Research Studies Press.

Freeman, C.T., P.L. Lewin, E. Rogers, J. Hätönen, T. Harte and D. H. Owens (2004). A novel repetitive control algorithm combining ilc and deadbeat control.. In: *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics.* Setubal, Portugal.

Furuta, K. and M. Yamakita (1987). The design of a learning control system for multivariable systems. In: *Preprints of the 1987 IEEE International Symposium on Intelligent Control.* Philadelphia, USA.

Garden, M. (1971). Learning control of actuators in control systems: Us patent us03555252. Technical report. Leeds & Northrup Company, Philadelphia, USA.

Grenander, U. and G. Szegö (1984). *Toeplitz Forms and Their Apllications.* Chelsea Publishing Company.

Hara, S., Y. Yamamoto, T. Omata and M. Nakano (1988). Repetitive control system: A new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control* **33**, 659–668.

Harte, T., J. Hätönen and D.H. Owens (2004). A new robust inverse-type ilc algorithm. In: *Proc. of the IFAC Workshop on Periodic Control Systems (PSYCO04).* Yokohama, Japan.

Harte, T.J., J. Hätönen and D.H. Owens (2005). Discrete-time inverse model-based iterative learning control: Stability, monotonicity and robustness. *International Journal of Control* **78(8)**, 577–586.

Horowitz, R. (1993). Learning control of robot manipulators. *Journal of Dynamic Systems, Measurement and Control* **115**, 402–411.

Hätönen, J., K. L. Moore and D.H. Owens (2003*a*). An algebraic approach to iterative learning control. *International Journal of Control* **77(1)**, 45–54.

Hätönen, J., T. Harte and D.H. Owens (2004*a*). Model-inverse based repetitive control. In: *Proceedings of UKACC International Conference on Control.* Bath, UK.

Hätönen, J., T. Harte, D.H. Owens, J. Ratcliffe, P. Lewin and E. Rogers (2003*b*). A new robust iterative learning control algorithm for application on a gantry robot. In: *Proc. of the IEEE conference on Emerging Technologies in Factory Automation (EFTA03).* Portugal.

Hätönen, J., T. Harte, D.H. Owens, J. Ratcliffe, P. Lewin and E. Rogers (2004*b*). Iterative learning control - what is it all about?. In: *Proc. of the IFAC Workshop on Periodic Control Systems (PSYCO04).* Yokohama, Japan.

Lee, J.J. and J.W. Lee (1993). Design of iterative learning controller with vcr servo system. *IEEE Transactions on Consumer Electronics* **39**, 13–24.

Lee, K.S., S.H. Bang, S. Yi, J.S. Son and S.C. Soon (1996). Iterative learning control of control of heat up phase for a batch polymerization reactor. *Journal of Process Control* **6**(4), 255–262.

Mita, T. and E. Kato (1985). Iterative learning control and its application to motion control of robot arm - a direct approach to servo-problems. In: *Proceedings of the 24th IEEE Conference on Decsion and Control.* Florida, USA.

Moore, K.L. (1993). *Iterative Learning Control for Deterministic Systems.* Springer-Verlag.

Moore, K.L. (1999). iterative learning control - an expository overivew. *Applied and Computational Controls, Signal Processing and Circuits* **1**, 151–214.

Nering, E.D. (1963). *Linear Algebra and Matrix Theory.* Wiley.

Norrlöf, M. and S. Gunnarsson (2002). Time and frequency domain properties in iterative learning control. *Internation Journal of control* **75**(14), 1114–1126.

Norrlöf, M. and S. Gunnarsson (2006). On the disturbance properties of high order iterative learning control algorithms. *Automatica* **42**, 2031–2034.

Oppenheim, A.V. and R.W. Schafer (1989). *Discrete-Time Signal Processing.* Prentice Hall.

Owens, D.H. (1981). *Multivariable and Optimal systems.* Academic Press.

Owens, D.H and J. Hätönen (2005). Iterative learning control - an optimization paradigm. *Annual Reviews in Control* **29**(1), 57–70.

Owens, D.H. and K. Feng (2003). Parameter optimisation in iterative learning control. *International Journal of Control* **76**(11), 1059–1069.

Ratcliffe, J., T. Harte, J. Hätönen, P. Lewin, E. Rogers and D.H. Owens (2004). Practical implementation of a model inverse optimal iterative learning controller on a gantry robot. In: *Proc. of the IFAC Workshop on Periodic Control Systems (PSYCO04)*. Yokohama, Japan.

Reddy, B.D. (1986). *Functional Analysis and Boundary Value Problems.* Wiley.

Rogers, E. and D.H. Owens (1992). *Stability Analysis for Linear Repetitive Processes.* Springer-Verlag.

Rosenbrock, H.H. (1970). *State-Space and Multivariable Theory.* Nelson.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable feedback control : analysis and design.* Wiley.

Spong, M.W. and M. Vidyasagar (1989). *Robot Dynamics and Control.* Wiley.

Tomizuka, M., T.C. Tsao and K.K.Chew (1989). Discrete-time domain analysis and synthesis of repetitive controllers. *Journal of Dynamic Systems, Measurement and Control* **111**, 353–358.

Uchiyama, M. (1978). Formulation of high-speed motion pattern of a mechanical arm by trial. *Transactions of the Society for Instrumentation and Control Engineers* **14**, 706–712.

Varga, R.S. (1962). *Matrix Iterative Analysis.* Prentice-Hall.

Xu, J. and Y. Tan (2002). Robust optimal design and convergence properties analysis of iterative learning control approaches. *Automatica* **38**(11), 1867–1880.

Young, N. (1988). *An Introduction to Hilbert Space.* Cambridge University Press.

Zilouchian, A. (1994). An iterative learning control technique for dual arm robotic system. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation.* San Diego, USA.