Advances in Iterative Learning Control with Application to Structural Dynamic Response Reconstruction

Johannes Jacobus Arnoldi Eksteen

Submitted in partial fulfillment of the requirements for the degree
Philosophiae Doctor
in the Faculty of Engineering, Built Environment and Information Technology,
University of Pretoria, Pretoria

July, 2014

Contents

Abstract

Title: Advances in Iterative Learning Control with Application to Structural Dynamic Response Reconstruction

Author: Johannes Jacobus Arnoldi Eksteen

Supervisor: Prof. P.S. Heyns

Department of Mechanical and Aeronautical Engineering, University of Pretoria

Degree: Philosophiae Doctor

Iterative learning control (ILC) is a repetitive control scheme that uses a learning capability to improve the tracking accuracy of a desired test system output over repeated test trials. ILC is sometimes used in response reconstruction on complex engineering structures, such as ground vehicles, for purposes of fatigue testing. The compensator that is employed in ILC in such cases is traditionally an approximate, linear inverse model of the closed-loop test system.

This research presents advances in ILC, particularly with respect to its application in response reconstruction for fatigue testing purposes. The contribution of this research focuses on three aspects: the use of a nonlinear inverse model in the ILC compensator instead of a linear inverse model; the use of multiple inverse models, each one defined over a different part of the test frequency band, instead of one model that covers the entire test frequency band; and the development and use of a new type of ILC algorithm. The contributions are implemented and demonstrated on a quarter vehicle road simulator, with favorable results for the use of nonlinear inverse models and multiple inverse models. The new ILC algorithm is shown to be competitive with the conventional inverse model-based algorithm, giving comparable to slightly worse results than the conventional ILC algorithm. In order to invert the nonlinear inverse models this research also presents advances in the stable inversion method that is used to invert such models.

Keywords: Iterative learning control, response reconstruction, fatigue testing, NARX models, Kolmogorov-Gabor polynomials, system identification, stable inversion, nonlinear, discrete time, Picard iteration, Mann iteration, quarter vehicle road simulator.

Acknowledgements

My thanks to everyone who served as study leaders at different times of this research, namely Prof. P.S. Heyns in recent years, Dr. A.D. Raath in the beginning, Prof. J.L. van Niekerk and Dr. M. Heyns. Thanks also to former colleagues, in particular C.R. Cater.

The financial assistance of the National Research Foundation and Investmech, Pty. Ltd. is gratefully acknowledged.

The patience of all involved, including the study leaders, my university, and my family is gratefully acknowledged.

Jan Eksteen, July 2014

.

And they shall see His face; and His name shall be in their foreheads.

Nomenclature

a_i	Model parameters in linear system identification
$A(q)$,$B(q)$,$F(q)$	Polynomials obtained in linear parametric system identification
b_i	Model parameters in linear system identification
c	Scalar scale factor in ILC
C	Linear operator representing a zero-phase filter in ILC
$e(k)$	White-noise input in discrete time linear models
E	Expectation
$F(.)$	NARX model
$f(.)$	Nonlinear function in the dynamical equation of the nonlinear state space formulation
$\bar{f}(.)$	Inverse of $f(.)$
$G(q)$	Generally an infinite series representing the discrete time impulse response of a linear discrete time system
$\hat{G}(q)$	Estimate of $G(q)$
G	Nonlinear operator representing the open loop plant
$\hat{G}$	Approximate model of G
$\bar{G}$	Inverse model of G
$h(.)$	Nonlinear function in the output equation of the nonlinear state space formulation
$\bar{h}(.)$	Inverse of $h(.)$
$H(q)$	Generally an infinite series representing the discrete time impulse response of the noise model of a linear discrete time system
$\hat{H}(q)$	Estimate of $H(q)$
i	Iteration index in ILC
K	Lipschitz constants
L	Nonlinear operator representing the ILC compensator
$\bar{L}$	Nonlinear operator representing the model-based component of the ILC compensator, usually representing an approximate inverse of T.
m	Iteration index in stable inversion
n	Model order; number of parameters in system identification models
N	Signal length in system identification

q	Backward shift operator as used in discrete time system models
Q	Linear operator representing a zero-phase low pass filter in ILC
r	Relative degree of a nonlinear system
s	Lipschitz constants
$T(q)$	Generally an infinite series representing the discrete time impulse response of a linear discrete time system in closed-loop form
T	Nonlinear operator representing the closed loop control system
$\hat{T}$	Approximate model of T
$\bar{T}$	Inverse model of T
$\mathcal{T}_{11}$	Nonlinear operator representing the system formulation of the conventional ILC algorithm
$\bar{\mathcal{T}}_{11}^{0}$	Nonlinear operator representing the system formulation of the alternative ILC algorithm
$\bar{\mathcal{T}}_{11}$	Nonlinear operator representing the system formulation of the modified alternative ILC algorithm
$u(k)$	Input signal of system (discrete time)
$u_d(k)$	Desired input signal of system (discrete time)
$u^{(i)}(k)$	i-th iteration of $u(k)$
$x(k)$	State vector
$y(k)$	Output signal of system (discrete time)
$\hat{y}(k\|k-1)$	One-step-ahead prediction of $y(k)$
$y_d(k)$	Desired output signal of system (discrete time)
$y^{(i)}(k)$	i-th iteration of $y(k)$
$\tilde{y}(k)$	Iteration variable of the alternative ILC algorithm
$\tilde{y}^{(i)}(k)$	i-th iteration of $\bar{y}(k)$
$x(k)$	State vector in transformed state space system (discrete time)
α_m	Scale factor of Mann and Ishikawa iteration for the m-th iteration
$\alpha(.)$	Nonlinear function in the dynamical equation of the nonlinear state space formulation in normal form
$\beta(.)$	Nonlinear function in the dynamical equation of the nonlinear state space formulation in normal form
β_m	Scale factor of Ishikawa iteration for the m-th iteration
$\eta(k)$	State vector in normal form and inverse state space system (discrete time)
$\eta^{(m)}(k)$	m-th iteration of η in stable inversion fixed point iteration (discrete time)
$\varepsilon(k)$	Prediction error in linear system identification
θ	Parameter vector in system identification
$\xi(k)$	State vector in normal form and inverse state space system (discrete time)
Ψ	Nonlinear function representing a coordinate transformation

Abbreviations

AD	Analogue to digital conversion
ARX	Autoregressive with exogenous input
CT	Continuous time
DA	Digital to analogue conversion
DT	Discrete time
FRF	Frequency response function
ID	Identification
ILC	Iterative learning control
LTI	Linear time invariant
MIMO	Multi input-multi output
MISO	Multi input-single output
NARX	Nonlinear ARX
NVH	Noise, vibration and harshness
OE	Output error
OLS	Orthogonal least squares
PSD	Power spectral density
SID	System Identification
SI	Stable inversion
SISO	Single input-single output
ZPF	Zero phase filter

Chapter 1

Introduction

1.1 Background

This research deals with adaptations and improvements in a number of control related fields in the context of response reconstruction, namely stable inversion of nonlinear systems, iterative learning control, and nonlinear system identification. Response reconstruction is essentially the approach of reconstructing structural responses in a test specimen in the laboratory for purposes of structural integrity validation testing (e.g. fatigue testing). The adaptations and improvements in this research will sometimes relate specifically to the way the above mentioned methods are usually implemented in response reconstruction.

For fatigue testing by means of response reconstruction to be reliable, the responses that are desired to be reconstructed need to be realistic, which means that they are representative of normal usage of the test specimen in fatigue terms (dynamic representativity is implied by this). The underlying loading/excitation that occurs in service may be either random or deterministic. In the automotive environment random loading is common. In the aeronautical environment (e.g. wing loading) the loading often contains both random and deterministic components. The loading may also have discernable sequence effects (also common in the aeronautical environment), which can be fatigue relevant (Bannantine *et al.* 1990). Fatigue testing can be done using either sinusoidal loading, block loading, stationary random loading, or actual service loading, with the latter being clearly the most representative in that it gives account of low frequency mean variations, statistical properties as well as sequence effects (Wright, 1999:61-74), which is important for the loading to be realistic in fatigue terms. While load history characterization in fatigue terms in more reduced terms is possible, e.g. stress-level crossings, time between levels and power spectral density (PSD) graphs, unless either of the former two are not combined with PSDs, they are considered inferior to basic range and mean counting with cumulative damage calculation. While the latter representation of fatigue content

using the Palmgren-Miner rule is still subject to inaccuracy, this is largely attributed to inaccurate accounting of load sequence effects. Thus, for fatigue testing to give proper account of normal service loading in fatigue terms is best achieved by reproducing the mean-and-ranges information of the service loading data, as well as the sequence information in the service loading data. This clearly points to the accurate reconstruction of actual service loading without much simplification as the most reliable approach in fatigue testing, which can be achieved by the reconstruction of the actual, in-service structural responses.

The reconstruction of service responses in fatigue testing often requires the desired responses to be logged during field measurements on the test specimen prior to the test, after which the responses are reconstructed in the test specimen in the laboratory. The actual reconstruction is accomplished using an iterative control algorithm called *iterative learning control* (ILC), which originated in the early 1970s in the robotics and automation field. Iterative learning control is essentially an iterative control scheme during which the test inputs are refined over successive iterations until the achieved test outputs/responses match the desired outputs/responses as closely as possible.

ILC is capable under favourable conditions (low levels of nonlinearity and low frequency content of signals) of achieving the reconstruction with remarkably simple compensators employed in the algorithm. However, to be truly reliable and safe the algorithm needs to ideally use an inverse dynamic model of the closed-loop test system in the compensator. (With "safe" is meant that the achieved test responses converge in well-behaved manner to the desired values without, for example, large overshoot prior to convergence.) Such models are usually obtained by system identification of black box models (Ljung (1999), Nelles (2001) and Billings (2013)). This means the models are not necessarily based on actual known system dynamics, but are calculated by optimizing the model's parameters until the model is able to predict the output from an arbitrary input with sufficient accuracy.

A good example of when we want to perform ILC with safe, reliable and not too rapid convergence of the achieved responses to the desired responses is when performing response reconstruction for purposes of full-scale automotive fatigue/durability testing in the laboratory (Raath (1993a, 1993b and 1997), Weal *et al.* (1997), Mianzo *et al.* (1998), De Cuyper *et al.* (1999), and French (2000)). In this approach either the vehicle's wheel hubs are fixed to electro-hydraulic actuators, or the tyres may rest on pans on top of the actuators, with the actuators under PID feedback control. Full-scale automotive fatigue tests are usually dynamic in nature. (The type of fatigue testing usually performed in aeronautical applications is typically quasi-static in nature, and the responses that are reconstructed are synthetic, often generated from load spectra.) Other uses of response reconstruction are also possible (in automotive and other industries), for example noise-vibration-harshness (NVH) testing, functional vibration testing, and shock testing. The response reconstruction procedure using the ILC algorithm is sometimes also referred to as *dynamic service load simulation*, *remote parameter control* (RPC), *time-waveform replication* (TWR), or *iterative transfer function control* (ITFC).

Response reconstruction for full-scale automotive fatigue testing in the laboratory also happens to be one of the first industry applications of the ILC procedure (Cryer *et al.* 1976). In fact, in as much as being applied to a discrete-time nonlinear system (as most physical test systems usually are to some degree), using an approximate linear inverse model in the ILC compensator that is stably solved in the frequency domain (i.e. non-causally), and using a scale factor in the ILC compensator, it was surprisingly advanced for such an early implementation. In recent decades a variety of analytical methods have also been developed to better evaluate the fatigue strength of dynamically loaded, complex structures analytically/computationally, including dynamic and nonlinear finite element analysis and better modelling of fatigue damage mechanisms. Though these methods are becoming more and more accurate, experimental validation of structural integrity is still the most reliable approach for such structures, especially in safety critical situations or situations with high cost implications of failures.

The University of Pretoria has been active in the development of response reconstruction methods since the late 1980s. Initially the focus had been on frequency domain methods for obtaining the inverse model (via system identification of linear models in the frequency domain), but the focus soon shifted to linear parametric models obtained by system identification in the time domain (Raath, 1993a). With the shift to the time domain came the difficulty of stably inverting models that, in inverse form, often tended to be unstable. Early methods to cope with this involved solving the inverse models anti-causally (i.e. in reverse time). Later, soon after the theory of stable inverse of linear systems was first published (Chen & Paden, 1992), the fundamentals of the stable inversion approach was independently developed and implemented in the inversion of time domain models in response reconstruction at the university with contributions from industry partners.

The mid 1990s saw early investigations at the university into the possible use of nonlinear polynomial models in response reconstruction (Cater, 1997), with the early conclusions being that the potential gains in accuracy are limited in case of automotive fatigue testing. The research reported in this thesis is based on the supposition that potential applications of response reconstruction are widely varying and, with the ever increasing need for test accuracy and reliability, in the long run the investment in the extra complexity and computational burden will be justified. The initial impetus for this research was therefore the implementation of nonlinear polynomial models in response reconstruction, and the exploitation of the theory of stable inversion of nonlinear models that has since been published (Devasia *et al.* 1996 and Zeng & Hunt, 2000) to invert the models. Another major early impetus was basing the response reconstruction algorithm on the solid theoretical footing that ILC already enjoyed in the literature (the ILC algorithm had been developed and implemented in response reconstruction for fatigue testing purposes independently of the formal, publicized development of ILC in the robotics and automization field).

Over the course of this research and during the gradual development of software to serve as both

Figure 1.1: Six-axis durability test on Class 8 truck.

a research vehicle for this research and a tool with which to serve the needs of industry (with respect to fatigue, shock and vibration testing), a number of tests were conducted for industry, sometimes employing some of the elements of this research (Eksteen *et al.* 2012). See Fig. 1.1 - Fig. 1.3. These were early implementations, done under strict budget constraints using available jigging hardware. (Future upgrades at the university will see, for example, the develoment of bell-crank systems for load application in automotive test rigs.)

In dealing with the topics of real-time control systems, stable inversion, and iterative learning control this research is very focussed on the topic of inversion of dynamic system models. For this reason this chapter will seek to first establish the inversion underpinnings of real-time control systems, from which the focus shifts slightly to the inversion of the model of a closed-loop control system. From the inversion of the model of a closed-loop control system, another not-too-big step can be taken to present the iterative learning control procedure, with all the above aimed at achieving exact, or as accurately as possible, tracking of a given desired output for a test sytem/plant. With these building blocks firmly in place the discussion turns to their use in an engineering application, namely response reconstruction, which is usually done for purposes of fatigue testing in the laboratory, but also for other purposes. At this point a brief introduction will be given to response reconstruction and the methodology will be explained. Novel contributions of the research will be highlighted where relevant in the discussion in this chapter.

Figure 1.2: Five-axis durability test on a Class 7 truck cab.

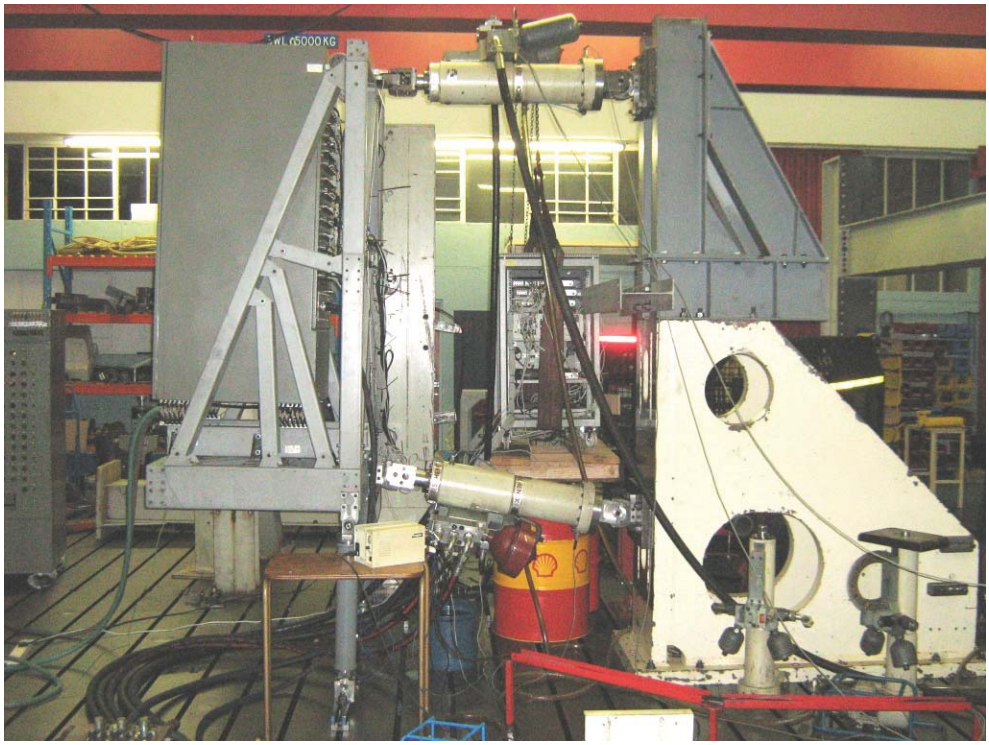

Figure 1.3: Two-axis shock and vibration test on defence industry electronics.

1.2 Asymptotic Tracking

Consider a dynamical system represented by an operator G. Assume the existence of a desired plant output, which we would like the plant to track as closely as possible. Normally a tracking control system would be used to accomplish asymptotic tracking of the desired plant output. In this study, however, we are concerned with achieving *exact* tracking of the desired output by the plant. This, of course, requires the determination of the input signal exactly corresponding to the desired output, implying that exact tracking is tantamount to exact inversion of the plant. It can be shown, in fact, that system inversion lies at the core of the concept of asymptotic tracking control systems (Goodwin, Graebe and Salgado, 2001:24-37).

One conceptual possibility for using the inverse model of a system in the control of the system to achieve a desired output (also called a *desired response* or *reference output*) is the open-loop (feed-forward) control architecture, shown in Fig. 1.4, in which $\bar{G}$ is an inverse model of the plant. (The hat symbol is widely used to indicate an approximate version or a model of a system, as result of which we will rather use an overbar than a hat symbol to indicate the inverse of a system, and occasionally the $^{-1}$ exponent.) For this approach to be truly accurate requires an exact inverse model to be used, the physical system (plant) and the inverse model to be stable, and the absence of disturbances. Any disturbances or model error will generally directly translate into a loss of tracking accuracy. Obtaining an accurate inverse model can also be challenging as the exact inverse model is frequently unstable due to the physical system often being non-minimum phase.

Towards finding a more robust concept for achieving good tracking control, the scheme shown in Fig. 1.5 may be proposed, in which $\hat{G}$ is a model of the plant. If the plant input signal is designated u, the output y, and the plant by the operator G, then

$$y = Gu \ ,$$

(in functional analysis operators are used without parenthesis) and the plant inverse is given as

$$u = G^{-1}y \ .$$

Let the desired plant output be r. In Fig. 1.5 the inverse model of Fig. 1.4 is now replaced by a feedback controller in series with the plant, with a compensator compensator H and $\hat{G}$ a model of the plant. The output of the controller is:

$$u = H(r - \hat{G}u)$$

Inverting H yields

$$r - \hat{G}u = H^{-1}u \ ,$$

from which

$$u = \hat{G}^{-1}(r - H^{-1}u) \ .$$

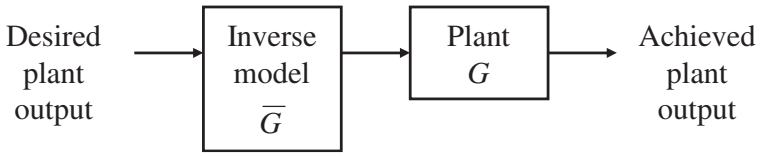

Figure 1.4: An open-loop (feed-forward) control strategy employing an inverse model of the plant.

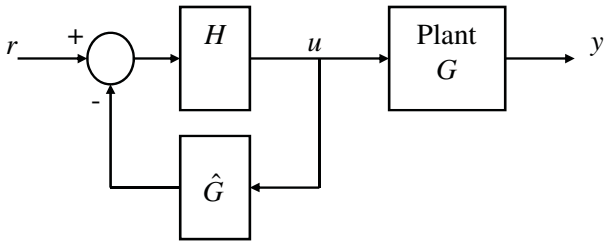

Figure 1.5: An open-loop control strategy employing a feedback controller with a plant model $\hat{G}$ in series with the plant.

It follows that if H has a high gain, then H^{-1} is small and $r - H^{-1}u \approx r$, causing this equation - and thus the feedback controller - to represent an approximate inverse of the plant. Thus, the high-gain feedback controller in Fig. 1.5 is still an open-loop control system but approximates the plant inverse without explicitly using it.

Due to the controller in Fig. 1.5 still being an open-loop controller, it still suffers the weaknesses of open-loop control, namely the effect of model inaccuracy and disturbances. However, if we replace the model $\hat{G}$ of the plant in Fig. 1.5 with the actual plant, G, as in Fig. 1.6, then we are no longer dependent on the plant model having to be accurate, but use the plant itself in the control system. The closed-loop control system in Fig. 1.6 achieves the same goal as the use of an inverse model as feedforward controller, only more robustly, and is therefore at its core a robust form of approximate system inversion. This type of feedback control system is called an *asymptotic tracking control system* because it causes the plant to attempt to track the desired plant output that is fed to the input of the control system, with tracking of a fixed value (a set point) being theoretically only achieved asymptotically (i.e. in infinite time). Note that other functions of the control system may be to stabilize the plant if needed, shape the dynamic response of the plant, and reduce the sensitivity of the plant to disturbances. Asymptotic tracking control systems are frequently required in industry to track randomly-varying desired plant outputs; However, the tracking of more broad-band signals is generally only approximate due to the fundamental limitations of real-time feedback control systems.

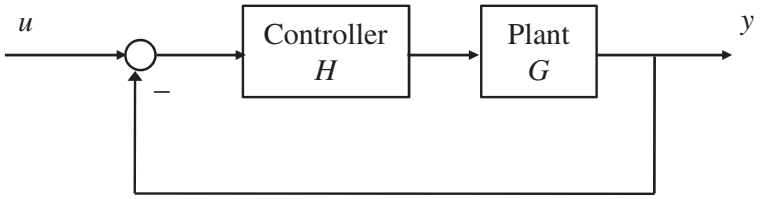

Figure 1.6: A closed-loop (tracking) control system

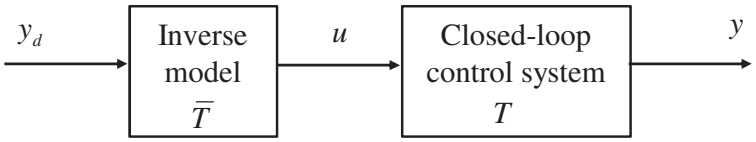

Figure 1.7: Inversion of a closed-loop system T via an off-line, open-loop compensator $\bar{T}$.

1.3 Exact Tracking and System Inversion

One of the principal limitations of a real-time feedback control system, such as the asymptotic tracking control system, is its inherently causal nature. The problem with this is that plants are frequently non-minimum phase (especially sampled test systems), and therefore tend to have unstable zeros, which results in the inverses of the plant having unstable poles. Inverse models with unstable poles can only be solved for bounded solutions non-causally. Since exact tracking of a desired plant output is effectively the exact inversion of the plant for that output, we need to do non-causal calculations or risk losing tracking accuracy. Clearly exact tracking therefore cannot generally be achieved with real-time control systems (which are causally constrained). Generally speaking, to achieve exact tracking, we need to have a feed-forward controller that is serially-linked to the real-time control system and represents the inverse of the closed-loop control system itself. This is illustrated in Fig. 1.7 for a closed-loop control system T.

In this configuration the desired output is first fed to the inverse of the closed loop control system, which calculates the required input signal and feeds it to the real-time control system. Because the inverse is not in a real-time feedback loop, it may be implemented off-line, which allows it to be implemented non-causally. To calculate bounded solutions for inverse models using non-causal methods, we have to use a method called *stable inversion*, particularly when the model is nonlinear.

1.4 Stable Inversion

During inversion of a system T, in the simplest case of a single-axis linear system, the zeros of T become the poles of the inverse T^{-1}, and *vice versa*. When T has any stable zeros, the inverse system will have unstable poles, making it unstable. This phenomenon can be genaralized to nonlinear and multi input-multi output (MIMO) systems. The stable inversion method firstly constructs the inverse model $\bar{T}$, and secondly solves this inverse model for the bounded solution of the input for a given output. When the system is nonlinear, the bounded solution of the input is calculated iteratively. For the continuous-time case see Chen and Paden (1992) and Devasia *et al.* (1996). For the discrete-time case see Hunt and Meyer (1997) and Zeng and Hunt (2000).

The iterative solution of the inverse model for a given desired output is in fact a fixed-point problem. A standard iteration scheme in fixed point problems, which is also the only one reported on in stable inversion literature, is the Picard iteration scheme. However, fixed-point iteration methods also feature other iteration schemes that have not yet been reported in stable inversion literature to the author's knowledge, namely Mann iteration and Ishikawa iteration. In one of the novel contributions of this research we will implement these iteration schemes in stable inversion of nonlinear models and exploit their advantages to achieve better results than with Picard iteration.

Another contribution of this research to stable inversion is the incorporation of a low-pass filter in the algorithm. The filter eliminates divergence of the algorithm in inverse models at high frequencies, which frequently result from complications in the system identification methods sometimes used to obtain the models in the first place.

The non-causal, inverse model-based control scheme shown in Fig. 1.7, while potentially being capable of exact inversion, still suffers from the problem of sensitivity to model errors and disturbances due to still being an open-loop controller. Having at our disposal a potentially "quite" accurate non-causal inverse model that is solved off-line, we have the tools needed to compose an iterative control scheme that involves the inverse model and uses iterative test trials to compensate for whatever inaccuracy may still reside in the inverse model. Such a scheme is precisely what is achieved with iterative learning control, which is discussed next.

1.5 Iterative Learning Control

1.5.1 Concept

Consider a closed-loop control system represented by an operator T, which we would like to track the desired plant output signal y_d as closely as possible. Iterative learning control (ILC) is a repetitive control scheme that uses a learning capability to improve the tracking accuracy over repeated test trials

of the closed-loop control system (Ahn *et al.*, 2007). The learning mechanism involves calculating the tracking error of the previous iteration, and then calculating a corrective action from the tracking error and the inverse model, which is added to the previous input signal to obtain an "updated" input signal. The updated input signal is intended to improve the tracking accuracy of the next test trial, following which the tracking error may be calculated again and the process repeated. Under sufficient conditions for convergence of the algorithm the desired output can be exactly tracked, possibly over a reduced test bandwidth as may be required to ensure convergence.

ILC is probably best explained by presenting the actual algorithm mathematically: Let the system input for the i^{th} system trial be designated $u^{(i)}$, the corresponding plant output $y^{(i)}$, thus

$$y^{(i)} = Tu^{(i)} \ ,$$

and let the tracking error of the i^{th} iteration be designated $e^{(i)}$, thus

$$e^{(i)} = y_d - y^{(i)} \ .$$

The most basic form of the learning formula of ILC is then given as

$$u^{(i+1)} = u^{(i)} + Le^{(i)} \ ,$$

where L is a (usually linear) operator that is called the *ILC compensator*. The updated input signal $u^{(i+1)}$ is thus calculated by the update formula *between* test trials, i.e. off-line (or in "batch" form). This also allows the calculations to be done non-causally or in the frequency domain. Specifying the ILC compensator L is the main aim of the *design problem* of ILC.

ILC is often presented in the context of systems that have to repeatedly track the same desired output signal, for example industrial robots working on serial production lines that repeatedly do the same welding or spray painting job. In this case the real-time control system is typically some type of asymptotic tracking control system, and ILC then uses the results of the repeated tracking exercises to further improve the tracking accuracy (for early references see, e.g., Arimoto *et al.* (1984) and Craig (1984)). This is done on an ongoing basis, and therefore ILC not only optimizes tracking accuracy for a given desired output and one specific set of system dynamics, but can also be used to *maintain* tracking accuracy by responding to gradual changes in the system dynamics through its "learning" capability.

1.5.2 The Use of ILC in Response Reconstruction and for Model Inversion

When ILC is implemented purely to obtain the system input resulting in the exact tracking of the desired output and then terminated once this has been achieved, it implies ILC is being used merely to *reconstruct* a desired system response (i.e. output) rather than to *maintain* tracking accuracy

over repeated test trials. When used in this way the ILC procedure is referred to here as *response reconstruction*. We will also say ILC is used *in* response reconstruction, partly because

- Response reconstruction for purposes of, e.g., fatigue testing, typically involves more steps than the reconstruction itself; and
- There are other ways of achieving a response reconstruction, e.g. the above mentioned feed-forward approach using inverse models of the closed loop test system.

When using ILC purely for response reconstruction, it is essentially used as a once-off inversion procedure because it calculates the input corresponding to the given desired system output. In this case it is the physical test system being "inverted".

We can also replace the physical test system with a system model and perform the test trials "mathematically", in which case ILC is being used as an inversion procedure for the system model. The fact that ILC is capable of exact tracking (subject to sufficient conditions) implies that the inversion of the model will be exact. It is noteworthy that when we use ILC in this way to invert a model, we do not calculate the inverse model, but only use the model in the normal sense as if it is a physical test system. Using ILC in this way is an *indirect* way of inverting a model because the exact inverse system model is never calculated, as opposed to the *direct* approach followed in stable inversion. The use of ILC to accomplish inversion of nonlinear models was first proposed in Markusson (2002).

1.5.3 Inverse Model-Based ILC Compensators

While ILC is capable of achieving convergence with even very simple forms of the ILC compensator L, such as a constant gains matrix (the so-called P-type ILC), the rate of convergence achieved with these compensators is not necessarily monotone. On the contrary, the convergence error (and achieved test outputs) may grow very large before finally decaying to zero. Achieving monotone convergence over relatively wide frequency bands generally requires the use of an approximate system inverse model in L. This is referred to as inverse model-based ILC. The rate of convergence may, additionally, be slowed down by scaling down L using a scalar scale factor. These aspects have already been explored in ILC; see, for example, Ghosh and Paden (2001).

When using ILC to invert a nonlinear system model, an approximate linear inverse model may be calculated for use in the ILC compensator. Such a model may be obtained by linear system identification, and as such represents a linearized version of the nonlinear test system around the average operating point represented by the identification data. The linear inverse model may be improved as an approximation of the actual nonlinear inverse model by generating it as the inverse of a linearization of the nonlinear model around the average operating point represented by the input signal of the previous ILC iteration. The linearization and inverse may furthermore be recalculated at

every time point for the state of the input at that time (rendering it time varying). Modifying the ILC-based inversion of a nonlinear model in this way results in the Newton method of inverting a system (with appropriate reformulation of the system), which has recently been presented by Smolders *et al.* (2008) for continuous-time systems. For a brief introduction to the Newton method of solving nonlinear equations, see Burden *et al.* (1993). This shows the ILC-based inversion of a nonlinear model using an approximate linear time-invariant (LTI) model to be a special case of the novel Newton inversion method. As a purely ILC approach to be executed on physical test systems this approach is referred to as Newton method based ILC, and was first presented for nonlinear discrete time systems by Lin *et al.* (2006). Promising to realize the advantages of rapid convergence of the Newton method in ILC, it is worthy of future investigation as an alternative to more complex methods utilizing a nonlinear inverse model in the ILC compensator (discussed next). Lin *et al.* (2006) also examine convenient ways of numerically implementing the algorithm, as does Smolders *et al.* (2008).

When the physical test system is highly non-linear and/or non-smooth, a good nonlinear system model may be readily identified, but how to invert the model may still be an open problem. In such cases, when doing ILC on the model in order to invert it without calculating its actual inverse, the use of an approximate *nonlinear* inverse model (e.g. a smooth model with a lower degree of nonlinearity than the actual model) in the ILC compensator may give better results than an approximate linear inverse model. The ability to accurately "invert" an accurate but complex model in this way (using ILC) may then be employed in an ILC compensator to do ILC on such physical test systems. Using ILC in this way could be a fruitful field of future research. Markusson (2002), however, prefers a linear ILC compensator based on an approximate linear inverse model, and considers a nonlinear ILC compensator based on an approximate nonlinear inverse model to be rather unjustified given the complexity of solving it.

1.5.4 Novel Contributions

In this study as a novel approach we not only implement *nonlinear*, inverse model-based ILC compensators in ILC of nonlinear test systems, as opposed to using linear inverse-based compensators, but do so in the response reconstruction context for fatigue and other tests. The use of nonlinear inverses in the ILC compensator L necessarily requires the use of stable inversion to generally obtain a bounded solution for the model (unless ILC is used for the model inversion as explained in Section 1.5.2). Note that in case of divergence of ILC with a linear L the use of a nonlinear (and potentially more accurate) L may result in convergence, may widen the frequency range over which convergence occurs, or may increase the accuracy of the best results achieved prior to divergence. We will also show how an accurate (nonlinear) inverse model in L simplifies the ILC dynamics (also shown in Markusson (2002)), resulting in monotone convergence and allowing better control of the rate of convergence.

We also develop an altogether new ILC algorithm, and show that the alternative algorithm

demonstrates the same properties as the conventional algorithm with regard to convergence and control of the rate of convergence. The alternative algorithm is relevant only when using a nonlinear L since, when using a linear L the alternative algorithm is equivalent to the conventional ILC algorithm.

Finally, we also note that the use of Mann and Ishikawa iteration in stable inversion has an analogy in ILC in that the conventional and alternative ILC algorithms developed here both have parallels in the Picard and Mann iteration schemes. It is furthermore shown that the application of Ishikawa iteration to ILC result in novel ILC iteration schemes for both the conventional and alternative ILC algorithms.

1.6 Response Reconstruction

1.6.1 Methodology

A good example of when we want monotone convergence that is not too rapid is the use of ILC in response reconstruction for purposes of full-scale fatigue/durability testing in the laboratory of complex structures subjected to variable-amplitude dynamic loading (e.g. ground vehicles). Full-scale durability testing in the laboratory of such structures is done by simulating on a representative test specimen in the laboratory the same structural loading the structure would experience in normal service, usually at an accelerated rate of damage accumulation in order to expedite the development cycle of new designs and save test costs. The more representative the desired response is of in-service loading, and the more accurate the reconstruction of the desired response is, the more reliable the test is. This implies that the overall process involves various elements, including field measurements to gather in-service responses, the reconstruction of the desired responses (which involves both a system identification phase and an ILC phase), evaluation of the achieved test acceleration, followed by actual fatigue testing. The overall methodology for response reconstruction for fatigue testing is shown in Fig. 1.8. The ILC algorithm shown in Fig. 1.8 is a generalization of the one described in Section 1.5.1 and is taken from Section 3.3. The methodologies of response reconstruction for other purposes, such as shock or vibration testing, will be simple variations of Fig. 1.8.

During field measurements (step A in Fig. 1.8) for purposes of fatigue testing it is preferable to measure the responses of the structure to the service conditions rather than the applied loads, mostly because the responses are usually easier to measure than the applied loads. Responses such as strains also allow explicit calculation of cumulative fatigue damage at the sensor locations. The field measurements are done for two purposes:

- To gather data to represent the normal usage of the structure with. This data, typically the strains, can then be subjected to cumulative fatigue damage calculations for quantifying the characteristic average fatigue damage per hour of normal usage for the structure. Other ways

of characterizing normal-usage service loading from such measurements are also possible, e.g. PSDs of acceleration data (see Section 1.1).

- To gather response data from which a desired response history, y_d, for the anticipated fatigue test may be selected that is at the same time as representative of actual usage as possible but severe enough to result in a sufficiently accelerated fatigue test when reconstructed in the laboratory. For response reconstruction, where the need is for accurate tracking of actual service responses for the (accelerated) test, the approach used in editing the service response data to obtain the desired responses is similar to the "time correlated fatigue analysis with damage editing method" in Halfpenny (2006). See also Shafiullah and Wu (2013) for a helpfull overview of methods aimed more at component level, and references therein. Cumulative fatigue damage calculations on y_d and comparison with results of fatigue damage calculations on the normal usage measurements allows projections of test duration and thus test cost to be made in advance.

The specimen is fitted in the laboratory with actuators that are carefully placed so as to be able to mimic the loads applied to the structure in normal service as closely as possible and with as little as possible mass-loading of the test specimen. When using electro-hydraulic actuators there is a real-time control system, typically PID feedback control, for every actuator that utilizes either actuator load or displacement as feedback. The real-time control system is employed to ensure stability of the actuators and appropriately shape the dynamic response. (The excitation frequencies that are normally involved are of such a nature that the specimen is frequently dynamically excited, loosely meaning the excitation of the structure's resonant frequencies.) The test system inputs are represented by the reference input signals of the real-time control systems, namely the control system command signals for either actuator load or displacement.

Once installed in the laboratory test rig the entire closed-loop control system is dynamically modelled (step B in Fig. 1.8). This is achieved by calculating a black box model of the system using system identification. The more common approach in system identification for purposes of response reconstruction for fatigue testing is to calculate linear models in the frequency domain (so-called *frequency response functions* (FRFs)), which can be readily inverted and solved in the frequency domain. An alternative approach, which is the one employed at the University of Pretoria, is to do the system identification in the time-domain, resulting in parametric time-domain models, which have traditionally been chosen to be in linear format (Raath, 1993a).

Having obtained a model $\hat{T}$ of the closed-loop test system, the model is inverted, giving $\tilde{L} := \bar{\hat{T}}$, and used in the ILC compensator, L, together with a iteration gain c chosen to determine the rate of convergence of ILC. Traditionally, in response reconstruction, L consists of the product of the gain and the inverse model, i.e.

$$L = c\tilde{L} \ .$$

ILC is then performed on the test system (step C in Fig. 1.8) to calculate the test system drive signals that best reconstruct the desired responses. (Smolders *et al.* (2008) generalises c to a constant matrix.)

Achieving an accurate simulation of the service loading requires exact tracking to be achieved of the desired output/response signals in the sensors in the laboratory specimen. Such desired responses have typically been measured at the same sensor locations (on a representative test specimen) under the normal in-service conditions. By exact tracking of the desired responses the structural stressing and excitation can be exactly recreated at the sensor locations on the structure. If the actuators are applied in a realistic manner and the specimen is structurally representative, this ensures that the stressing and vibration patterns *throughout* the structure will also be accurately recreated, which in turn implies accurate reconstruction of the metal fatigue and many other failure mechanisms the structure would experience in service. Full equivalence may require additions of some environmental factors such as corrosive environments, temperature, humidity, etc., which are sometimes incorporated in the laboratory test (Dobson & Schwab, 2006).

Having accomplished the exact or nearly exact tracking, the final input signals to the test system and the final achieved responses in the test specimen are then saved. The achieved strain responses in the test specimen may be subjected to cumulative fatigue damage calculations and the results compared to the normal in-service rates of damage accumulation in order to determine the test acceleration over normal usage and the required duration of the fatigue test to simulate a given service life for the structure (step D in Fig 1.8).

Finally, the final input signals may then be repeatedly sent to the system to simulate the structure's operational life in the laboratory (step E in Fig 1.8). The input signals are not changed in order to maintain accurate tracking of the desired responses even if the structure begins to deteriorate due to fatigue. This is because in normal operation the effect of structural degradation will be exactly the changes in structural responses experienced in the test specimen, while the applied loading generally stays the same, which is achieved by not changing the test input signals.

1.6.2 New Developments and Contributions

The literature of response reconstruction (with which is meant basically the literature of simulation testing of ground vehicles for durability/fatigue evaluation purposes) does not refer to the convergence theory of ILC, alternative formulations and variations in the ILC algorithm, or even use the ILC terminology, except for a few very recent exceptions (Smolders *et al.* (2008) and Deckers *et al.* (2012)). Not even the use of stable inversion in either linear or nonlinear ILC compensators have been reported in either ILC or response reconstruction literature until Markusson (2002). ILC theory contributes to response reconstruction a good understanding of the principles for achieving convergence, controlling the rate of convergence, increasing the robustness of convergence against model inaccuracy, and han-

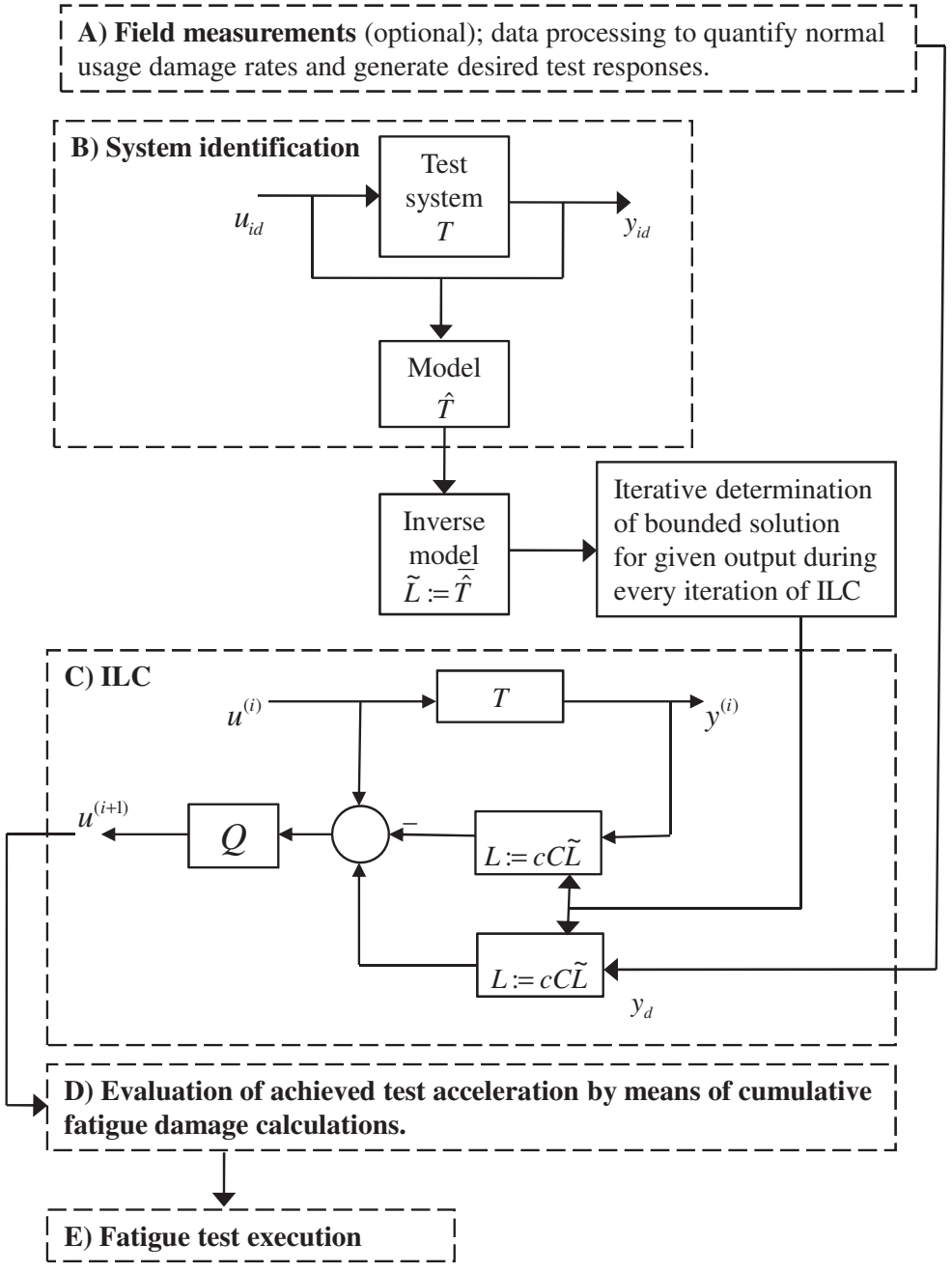

Figure 1.8: Methodology used in response reconstruction for purposes of fatigue testing. Here Q and C are zero phase filters (see Section 3.3.4).

dling disturbances. While this study does not focus on the handling of disturbances, it does present from the field of ILC literature in the robotics and automation context guidelines for the design of the compensator and choice of filters for the inverse-based compensator presented in this research.

The use of a nonlinear inverse model in the ILC compensator instead of a linear model was already mentioned in Section 1.5.4 as a novel approach not just for ILC, but for response reconstruction in particular. Smolders *et al.* (2008) has since reported on similar efforts in the fatigue testing setting. The advantages of using a nonlinear, inverse model-based ILC compensator instead of a linear inverse model-based ILC compensator will be evaluated on an experimental quarter vehicle road simulator. The nonlinear inverse model used in the inverse-based ILC compensator will be obtained from stable inversion performed on a model obtained by nonlinear parametric system identification. The model formulation that will be used is the polynomial nonlinear ARX (NARX) model structure, whose parameters may be identified using linear least squares methods (Cater (1997), Nelles (2001)). The approach used in Smolders *et al.* (2008) is to model the linear and nonlinear dynamics separately, the former using a state space model, and the latter using a sigmoidal neural network. The purpose with modelling the linear dynamics separately was to be able to more easily incorporate known properties from the lumped mechanical systems they focussed on (resulting in a so-called *grey-box* model). Furthermore, instead of stable inversion they used the Newton's method to invert the model, which is also capable of handling non-minimum phase systems and was a novel contribution of their research.

Another contribution of this research is the conversion of the NARX model formulation to a form amenable for stable inversion, and the bounds placed on the structure of the model during system identification to facilitate stable inversion of the model.

The alternative ILC algorithm proposed in Section 1.5.4 will also be evaluated on the quarter vehicle road simulator.

Finally, in this research a multiple model approach is proposed to deal with inaccuracies arising from a number of complications of the system identification process of such models. The use of this multiple model approach to improve the accuracy of response reconstruction will also be evaluated on the quarter vehicle road simulator.

1.7 Document Overview

In Chapter 2 the conventional stable inversion method for inverting a square, nonlinear discrete-time system is presented. The Mann and Ishikawa iteration modifications of the stable inversion algorithm is presented. In a series of examples the stable inversion method is demonstrated, and the advantages of the use of low pass filters in the algorithm as well as the use of the Mann and Ishikawa iteration schemes are presented.

In Chapter 3 an overview of the theory of ILC of discrete-time linear systems is presented, with special focus on model-based or inverse-based ILC compensators L, for the general deterministic square, MIMO case, which is applicable to many laboratory situations. A *general inverse based compensator*, which explicitly uses the approximate system inverse model, is proposed. An alternative ILC algorithm is also presented together with a modification that enables it to fully match the various properties of the conventional algorithm. An example is presented of ILC on a nonlinear system where in a series of case studies, the ability of ILC to achieve exact tracking (i.e. converge) is demonstrated, the ability of the alternative algorithm to sometimes converge when the conventional algorithm diverges is demonstrated, and the conventional and alternative algorithms are compared in a case where both diverge.

Chapter 4 presents an overview of the broader response reconstruction procedure, with a focus on the contributions of ILC theory to the design guidelines of the ILC compensator and choice of filters as used in response reconstruction. An overview of linear parametric system identification is presented, followed by a presentation of nonlinear parametric system identification, with a focus on the formulation and identification of NARX models. The procedure for the stable inversion of the NARX model is presented. Some mechanisms of inaccuracy in the identification of NARX models are presented, and the multiple model approach as a solution that can lead to improved model accuracy is presented. Finally, in a series of tests on the experimental quarter vehicle road simulator the use of nonlinear inverse-based ILC compensators is evaluated in response reconstruction, the use of the multiple model approach is evaluated, and the use of the alternative ILC algorithm is evaluated.

Chapter 5 presents the main results and conclusions, primarily regarding the use of nonlinear inverse models in the ILC compensator, the use of the alternative ILC algorithm, and the use of multiple models to cover different parts of the test spectrum, and makes recommendations for future research.

Chapter 2

Stable Inversion of Nonlinear Systems

2.1 Introduction

This chapter presents the conventional stable inversion method for inverting a square, nonlinear, analytical discrete-time system, of which the NARX models that are used in response reconstruction in this research is a special case (though NARX models may be non-square). The reformulation of the system model in a suitable form (called the *normal form*), the inversion of the reformulated model, and the iterative solution of the inverse model for the bounded solution are presented separately in sections 2.3, 2.4 and 2.5 respectively. "Conventional" refers to the fact that iterative solution of the inverse model is usually accomplished with Picard iteration (cf. sections 2.5.1 and 2.5.2). The employment of Mann and Ishikawa iteration for the iterative solution of the inverse, discussed in sections 2.5.3 and 2.5.1 respectively, is a novel contribution of this research, with Theorems 2.2 and 2.3 being new. Another novel contribution is the employment of a low pass filter in both Picard and Mann iteration in Section 2.5.4.

In a series of examples the stable inversion method is demonstrated, and the advantages of using Mann iteration over Picard iteration and the use of low pass filters in the algorithm are demonstrated. Example 1 demonstrates the advantages of using Mann iteration and a low pass filter for a short deterministic signal. Example 2 does the same for a random signal. Example 3 demonstrates the use of Ishikawa iteration, then again demonstrates the advantages of using Mann iteration and low pass filtering, and finally evaluates various strategies for the gain used in Mann iteration (including iteration dependent and time dependent strategies).

As an introduction we first present in this section the case where the system model may be directly inverted without requiring any system reformulation, which due to its simplicity is presented directly in square, MIMO form.

Consider a MIMO nonlinear, discrete time state space system of the form

$$x(k+1) = f(x(k), u(k)) \tag{2.1}$$

$$y(k) = h(x(k), u(k)) \ , \tag{2.2}$$

$u(k), y(k) \in \Re^m$ and $x(k) \in \Re^n$, with $(x^\circ, u^\circ) = ([0], [0])$ an equilibrium pair of the system, i.e. $f(x^\circ, u^\circ) = x^\circ$ and $h(x^\circ) = [0]$. f and h are analytic in their domains, i.e. $f, h \in C^\omega$. (C^k designates the function space of k-times continuous differentiable functions, while C^∞ designates the space of infinitely continuously differentiable functions, i.e. smooth functions. A function is in C^ω if it is in C^∞ (i.e. smooth) and the Taylor series expansion of the function converges to the function over its whole domain (Isidori, 1995, p. 471).) The presence of $u(k)$ in h represents a direct transmission from the input to the output. Assume the Jacobian matrix

$$\left.\frac{\partial y(k)}{\partial u(k)}\right|_{x^\circ, u^\circ} \tag{2.3}$$

is non-singular at (x°, u°). By the inverse function theorem (Isidori, 1995, p. 471) this implies $y(k) = h(x(k), u(k))$ is a diffeomorphism from around the point (x°, u°) (i.e. is bijective (i.e. invertible) and both h and $\bar{h} := h^{-1}$ are in C^∞ (i.e. smooth) in a neighbourhood of x°). Therefore h may be inverted for the unique $u(k)$ corresponding to a given $y(k)$, giving the smooth mapping $u(k) = h^{-1}(x(k), y(k))$, which, when substituted into Eq. 2.1, yields the inverse system as

$$x(k+1) = f(x(k), h^{-1}(x(k), y(k))) \tag{2.4}$$

$$:= \bar{f}(x(k), y(k)) \tag{2.5}$$

$$u(k) = \bar{h}(x(k), y(k)) \ . \tag{2.6}$$

The formulation of the inverse of MIMO nonlinear discrete-time systems when the Jacobian (Eq. 2.3) is nonsingular is thus straight-forward. When the Jacobian of Eq. 2.3 is singular it may be possible to convert the system to a special form, called the *normal form*, from which the inverse system may still be derived. The derivation of the normal form that is presented here follows somewhat the presentation in Castillo *et al.* (1991) and Monaco *et al.* (1987). See also Nijmeijer *et al.* (1990). For the equivalent continuous time analysis see Isidori (1995). We first present the derivation of the normal form for the single input-single output (SISO) case, which is useful for research purposes without the clutter and extra complexity of MIMO notation. Thereafter the conversion to normal form and subsequent inversion is presented for MIMO nonlinear discrete-time systems.

After having formulated the inverse system the process of obtaining the bounded solution for a given known output signal is generally an iterative procedure which is essentially a fixed point problem. The conventional approach achieves this using *Picard iteration*, but this algorithm can give poor results (i.e. diverge or diverge strongly) where other iteration schemes may succeed (converge or diverge less strongly). Here we present two alternative iteration schemes from the theory of fixed point

determination, namely *Mann iteration* and *Ishikawa iteration*, both of which are known to have better convergence properties than Picard iteration. These iteration schemes have not been discussed before in the literature in connection with stable inversion. Here we show that they give better results for stable inversion than Picard iteration. The focus in this study will be on Mann iteration in particular. We also further refine these iteration methods by introducing the use of low pass filter in the algorithm aimed at improving the convergence properties, and allowing time-varying scale/gain factors in the algorithms.

2.2 The Normal Form for SISO Systems

Consider the SISO, nonlinear discrete-time system

$$x(k+1) = f(x(k), u(k)) \tag{2.7}$$

$$y(k) = h(x(k)) \ , \tag{2.8}$$

with $u(k), y(k) \in \Re$ and $x(k) \in \Re^n$. The functions f and h are analytic in their domains (and therefore smooth). Let $(x^\circ, u^\circ) = ([0], 0)$ be an equilibrium pair of the system. Let the undriven state dynamics be denoted by $f_0(\cdot) = f(\cdot, 0)$ and the l-times iterated composition of f_0 by $f_0^l(\cdot)$. The system has *local relative degree* r around the point (x°, u°) if (Castillo *et al.* 1991)

$$\frac{\partial h \circ f_0^l \circ f(x(k), u(k))}{\partial u(k)} = 0 \tag{2.9}$$

for $l = 0, ..., r-2$ and all x in a neighbourhood of x°, and

$$\left.\frac{\partial h \circ f_0^{r-1} \circ f(x(k), u(k))}{\partial u(k)}\right|_{x^\circ, u^\circ} \neq 0 \ . \tag{2.10}$$

Assume

$$0 \in \mathrm{Im}(h \circ f_0^{r-1} \circ f(x(k), u(k))) \ , \tag{2.11}$$

where Im denotes the image of a mapping. From the above definition of local relative degree it follows that

$$\begin{aligned} y(k+i) &= h \circ f_0^i(x(k)) \ , & i = 1, \ldots, r-1 \\ y(k+r) &= h \circ f_0^{r-1} \circ f(x(k), u(k)) \\ &= h \circ f_0^r(x(k)) + S(x(k), u(k)) \ , \end{aligned} \tag{2.12}$$

around (x°, u°) with $S(\cdot, 0) = 0$ and, from the definition of local relative degree:

$$\left.\frac{\partial y(k+r)}{\partial u(k)}\right|_{x^\circ, u^\circ} = \left.\frac{\partial S(x(k), u(k))}{\partial u(k)}\right|_{x^\circ, u^\circ} \neq 0 \ . \tag{2.13}$$

This implies that $y(r)$ is the first instant of the output affected by $u(0)$. From the analicity of f and h it follows that either $r \leq n$ or $r = \infty$, the latter case in which the input never affects the output.

We now develop a coordinate change with which to transform the above SISO system with local relative degree r to the normal form. To this end note that, by the definition of r, the functions

$$\begin{aligned}\psi_1(x(k)) &= h(x(k)) \\ \psi_2(x(k)) &= h \circ f_0(x(k)) \\ &\dots \\ \psi_r(x(k)) &= h \circ f_0^{r-1}(x(k))\end{aligned}$$

are linearly independent at the point x°. In order to complete a coordinate transformation when $r < n$, an arbitrary set of $\hat{n} = n - r$ additional smooth functions $\psi_{r+1}(x(k)), \dots, \psi_n(x(k))$ may always be found such that the set of functions

$$z(k) = \Psi(x(k)) = (\psi_1(x(k)), \dots, \psi_n(x(k)))^{\mathrm{T}}$$

has a non-singular matrix at x°. This implies $\Psi(x(k))$ is a local diffeomorphism around x° (i.e. is bijective (i.e. invertible) and both Ψ and Ψ^{-1} are in C^∞ (i.e. smooth) in a neighbourhood of x° (Isidori, 1995:11)), and therefore qualifies as coordinate transformation in a neighborhood of x° for the system. Using the definition of local relative degree the system in the new coordinates becomes:

$$\begin{aligned}z_i(k+1) &= \psi_i(x(k+1)) \\ &= h \circ f_0^{i-1}(x(k+1)) \\ &= h \circ f_0^{i-1} \circ f(x(k), u(k)) \\ &= h \circ f_0^{i}(x(k)) \\ &= z_{i+1}(k) \end{aligned} \tag{2.14}$$

for $i = 1, \dots, r-1$, and

$$\begin{aligned}z_r(k+1) &= \psi_r(x(k+1)) \\ &= h \circ f_0^{r-1}(x(k+1)) \\ &= h \circ f_0^{r-1} \circ f(x(k), u(k)) \\ &= h \circ f_0^{r}(x(k)) + S(x(k), u(k)) \\ &= h \circ f_0^{r}(\Psi^{-1}(z(k))) + S(\Psi^{-1}(z(k)), u(k)) \\ &:= \alpha(z(k)) + \beta(z(k), u(k)) \, . \end{aligned} \tag{2.15}$$

For $i = r+1, \dots, n$ it follows that

$$\begin{aligned}z_i(k+1) &= \psi_i(x(k+1)) \\ &= \psi_i(f(x(k), u(k))) \\ &= \psi_i(f(\Psi^{-1}(z(k)), u(k))) \\ &:= q_{i-r}(z(k), u(k)) \, . \end{aligned} \tag{2.16}$$

Finally, for the output we have $y(k) = h(x(k)) = \psi_1(x(k)) = z_1(k)$. The system in the new coordinates around (x°, u°) is in the so-called normal form:

$$\begin{aligned}
z_1(k+1) &= z_2(k) \\
&\vdots \\
z_{r-1}(k+1) &= z_r(k) \\
z_r(k+1) &= \alpha(z(k)) + \beta(z(k), u(k)) && (2.17)\\
z_{r+1}(k+1) &= q_1(z(k), u(k)) && (2.18)\\
&\vdots \\
z_n(k+1) &= q_{n-r}(z(k), u(k)) && (2.19)\\
y(k) &= z_1(k) && (2.20)
\end{aligned}$$

Partitioning the coordinate transformation as $z = (\xi^{\mathrm{T}}, \eta^{\mathrm{T}})^{\mathrm{T}} = (\xi_1, \ldots, \xi_r, \eta_1, \ldots, \eta_{n-r})^{\mathrm{T}}$, the normal form may be restated as

$$\begin{aligned}
\xi_1(k+1) &= \xi_2(k) \\
&\vdots \\
\xi_{r-1}(k+1) &= \xi_r(k) \\
\xi_r(k+1) &= \alpha(\xi(k), \eta(k)) + \beta(\xi(k), \eta(k), u(k)) && (2.21)\\
\eta(k+1) &= q(\xi(k), \eta(k), u(k)) && (2.22)\\
y(k) &= \xi_1(k) && (2.23)
\end{aligned}$$

By the definition of local relative degree, resulting in Eq. 2.13, it follows that S, and in turn β, are diffeomorphisms and therefore may be inverted for the unique u corresponding to a given $\xi_r(k+1)$, giving the smooth function (from Eq. 2.21):

$$\begin{aligned}
u(k) &:= \gamma(\xi(k), \eta(k), \xi_r(k+1)) && (2.24)\\
&= \beta^{-1}(\xi(k), \eta(k), \xi_r(k+1) - \alpha(\xi(k), \eta(k))) \,. && (2.25)
\end{aligned}$$

Note that $u(k) = \gamma(\xi(k), \eta(k), v(k))$, with $(\xi(k)^{\mathrm{T}}, \eta(k)^{\mathrm{T}})^{\mathrm{T}} = \psi(x)$ and $v(k)$ an external control signal, defines a non-singular, nonlinear static state feedback. Substitution into Eq. 2.21 and Eq. 2.21 gives:

$$\begin{aligned}
\xi_1(k+1) &= \xi_2(k) \\
&\vdots \\
\xi_{r-1}(k+1) &= \xi_r(k) \\
\xi_r(k+1) &= v(k) \\
\eta(k+1) &= q(\xi(k), \eta(k), \gamma(\xi(k), \eta(k), v(k)))
\end{aligned}$$

$$:= \bar{f}(\xi(k), \eta(k), v(k))$$
$$y(k) = \xi_1(k) \tag{2.26}$$

Clearly this feedback renders the $\eta(k)$ states unobservable and linearizes the observable part of the system (the ξ states). Now, for initial condition $\xi(0) = [0]$ and external control signal $v(k) = 0$ for all k, clearly all state evolutions in Eq. 2.26 remain in $\mathcal{H}_0 = \{(\xi(k)^{\mathrm{T}}, \eta(k)^{\mathrm{T}})^{\mathrm{T}} : \xi(k) = [0]\}$, implying $\xi(k) = [0]$ for all k. Such evolutions result in zero output, $y(k) = \xi_1(k) = 0$, and are characterized by

$$\eta(k+1) = \bar{f}([0], \eta(k), 0) \ . \tag{2.27}$$

The system dynamics constrained to $\mathcal{H}_0$ is called the *zero output constrained dynamics*, or the *zero dynamics* in short, and is given by Eq. 2.27 (Monaco *et al.* 1987). From $\xi_r(k+1) = y(k+r) = v(k)$ in Eq. 2.26 it is clear that the feedback $u(k) = \gamma(\xi(k), \eta(k), v(k))$ can be used to force the output to $v(k) = 0$ in r samples. In view of this the zero dynamics (in the case of discrete time systems) can be considered as the residual dynamics of the closed loop system when static state feedback is used to force the output to zero. Its stability thus determines the ability to reach an equilibrium point $x^\circ = [0]$ since, while the output (and thus ξ) is brought to zero, η will eventually decay to zero too due to the asymptotic stability of the zero dynamics, implying x will eventually reach $x^\circ = [0]$. A nonlinear system is said to be *minimum phase* if its zero dynamics is asymptotically stable.

2.3 The Normal Form for MIMO Systems

Consider the square, MIMO nonlinear discrete-time system

$$x(k+1) = f(x(k), u(k)) \tag{2.28}$$
$$y(k) = h(x(k)) \ , \tag{2.29}$$

with $u(k) \in \Re^m$, $x(k) \in \Re^n$, $y(k) \in \Re^m$ and f and h analytic in their domains. Let $(x^\circ, u^\circ) = ([0], [0])$ be an equilibrium pair of the system. The output y_i has local relative degree r_i around the point (x°, u°) if

$$\frac{\partial h_i \circ f_0^l \circ f(x(k), u(k))}{\partial u_j(k)} = 0 \tag{2.30}$$

for $l = 0, \ldots, r_i - 2$, $j = 1, \ldots, m$, and all x in a neighbourhood of x°, and there exists a $j_i \in \{1, \ldots, m\}$ such that

$$\left.\frac{\partial h_i \circ f_0^{r_i - 1} \circ f(x(k), u(k))}{\partial u_{j_i}(k)}\right|_{x^\circ, u^\circ} \neq 0 \ . \tag{2.31}$$

Define the *input-output decoupling matrix* as

$$A(x(k), u(k)) := \left(\frac{\partial y_i(k + r_i)}{\partial u_j(k)}\right)_{i,j} = \left(\frac{\partial h_i \circ f_0^{r_i - 1} \circ f(x(k), u(k))}{\partial u_j(k)}\right)_{i,j} \ . \tag{2.32}$$

Assume that

$$\text{rank } A(x(k), u(k))|_{x^\circ, u^\circ} = m \ , \tag{2.33}$$

in which case the system is said to have *vector relative degree* (or characteristic number) $(r_1, r_2, ..., r_m)$. Assume

$$0 \in \text{Im}(h_i \circ f_0^{r_i - 1} \circ f(x(k), u(k))) \ . \tag{2.34}$$

In order to derive a coordinate transformation with which to transform the system to the normal form we select the vector

$$\begin{aligned}
\xi(k) &= (\xi^{(1)}(k)^{\mathrm{T}}, \ldots, \xi^{(m)}(k)^{\mathrm{T}})^{\mathrm{T}} && (2.35)\\
\xi^{(i)}(k) &= (\xi_1^{(i)}(k), \ldots, \xi_{r_i}^{(i)}(k))^{\mathrm{T}} && (2.36)\\
&= (h_i(x(k)), h_i \circ f_0(x(k)), \ldots, h_i \circ f_0^{r_i - 1}(x(k)))^{\mathrm{T}} && (2.37)\\
&= (y_i(k), \ldots, y_i(k + r_i - 1))^{\mathrm{T}} \ . && (2.38)
\end{aligned}$$

The function vector $\xi(k)$ has $|r| := \sum_{i=1}^m r_i$ functions. When $|r| < n$, an arbitrary set of $\hat{n} = n - |r|$ additional smooth functions

$$\eta(k) = (\eta_1(k), \ldots, \eta_{n-|r|}(k))^{\mathrm{T}} \tag{2.39}$$

may always be found such that the function vector

$$\psi(x(k)) = (\xi^{\mathrm{T}}(k), \eta^{\mathrm{T}}(k))^{\mathrm{T}} \tag{2.40}$$

has a non-singular matrix at x°, and therefore defines a local coordinate transformation in a neighborhood of x° for the system. This coordinate transformation results in the following system in the new coordinates:

$$\begin{aligned}
\xi_1^{(i)}(k+1) &= \xi_2^{(i)}(k)\\
&\vdots\\
\xi_{r_i - 1}^{(i)}(k+1) &= \xi_{r_i}^{(i)}(k)\\
\xi_{r_i}^{(i)}(k+1) &= \alpha_i(\xi(k), \eta(k)) + \beta_i(\xi(k), \eta(k), u(k))\\
&\quad i = 1, \ldots, m\\
\eta(k+1) &= q(\xi(k), \eta(k), u(k))\\
y_1(k) &= \xi_1^{(1)}(k)\\
&\vdots\\
y_m(k) &= \xi_1^{(m)}(k) && (2.41)
\end{aligned}$$

This is the normal form for the MIMO system. By setting

$$\xi_{r_i}^{[1,m]} := (\xi_{r_1}^{(1)}, \ldots, \xi_{r_m}^{(m)})^{\mathrm{T}} \tag{2.42}$$

$$\alpha := (\alpha_1, \ldots, \alpha_m)^{\mathrm{T}} \tag{2.43}$$

$$\beta := (\beta_1, \ldots, \beta_m)^{\mathrm{T}} \tag{2.44}$$

$$\xi_1^{[1,m]} := (\xi_1^{(1)}, \ldots, \xi_1^{(m)})^{\mathrm{T}} \tag{2.45}$$

the MIMO normal form may be written more compactly as

$$\xi_{r_i}^{[1,m]}(k+1) = \alpha(\xi(k), \eta(k)) + \beta(\xi(k), \eta(k), u(k)) \tag{2.46}$$

$$\eta(k+1) = q(\xi(k), \eta(k), u(k)) \tag{2.47}$$

$$y(k) = \xi_1^{[1,m]}(k) \tag{2.48}$$

By the definition of vector relative degree (Eq. 2.33) it follows that β (Eq. 2.46) is a diffeomorphism and may be inverted for the unique u, giving the smooth mapping

$$u(k) = \gamma(\xi(k), \eta(k), \xi_{r_i}^{[1,m]}(k+1)) \tag{2.49}$$

$$= \beta^{-1}(\xi(k), \eta(k), \xi_{r_i}^{[1,m]}(k+1) - \alpha(\xi(k), \eta(k))) \; . \tag{2.50}$$

The non-singular, nonlinear static state feedback $u(k) = \gamma(\xi(k), \eta(k), v(k))$, with $(\xi(k)^{\mathrm{T}}, \eta(k)^{\mathrm{T}})^{\mathrm{T}} = \psi(x)$ and $v(k)$ an external control signal, results in

$$\begin{aligned}
\xi_1^{(i)}(k+1) &= \xi_2^{(i)}(k) \\
&\vdots \\
\xi_{r_i-1}^{(i)}(k+1) &= \xi_{r_i}^{(i)}(k) \\
\xi_{r_i}^{(i)}(k+1) &= v_i(k) \\
&\quad i = 1, \ldots, m \\
\eta(k+1) &= q(\xi(k), \eta(k), \gamma(\xi(k), \eta(k), v(k))) \\
&:= \bar{f}(\xi(k), \eta(k), v(k)) \qquad (2.51) \\
y(k) &= \xi_1^{[1,m]}(k) \qquad (2.52)
\end{aligned}$$

Proceeding exactly as in the case of SISO systems we observe that the feedback renders the $\eta(k)$ states unobservable and linearizes the observable part of the system, namely the ξ states. Furthermore, we may define the *zero output constrained dynamics* as evolutions with $\xi(k) = [0]$ for all k (and thus $v(k) = \xi_{r_i}^{[1,m]}(k+1) = [0]$). The zero dynamics results in zero output $y(k) = \xi_1^{[1,m]}(k) = [0]$, and is characterized by (from Eq. 2.51)

$$\eta(k+1) = \bar{f}([0], \eta(k), [0]) \; . \tag{2.53}$$

The other concepts associated with the zero dynamics, namely the meaning of the zero dynamics and the definition of minimum phase, carry over unchanged from the SISO case.

Note that when the above assumption regarding the input-output decoupling matrix fails, namely when (cf. Eq. 2.33) rank $A(x(k), u(k))|_{x^\circ, u^\circ} \neq m$, an algorithm is available for determining the

zero dynamics (Monaco *et al.* 1987), however in which the analysis is not generally local in nature as is the case here.

2.4 The Inverse System

Given a bounded desired output $y_d(k)$, $k \in \mathcal{Z}$, the *stable inversion problem* is to find a *desired state trajectory* $x_d(k)$ and *control input* $u_d(k)$ satisfying Eq. 2.28, that are in the sequence space l^∞ (and thus bounded on $k \in \mathcal{Z}$), and that by Eq. 2.29 results in $y(k) = y_d(k)$ for the given system. (l^∞ is a complete, normed vector space; see Kreyszig (1978:33, 61). A complete, normed vector space is also called a Banach space, and therefore l^∞ is a Banach space). In this study u_d is also called the *desired input*. The desired state trajectory and input is found as the bounded solution of the inverse system for the given desired output. We now focus on obtaining the inverse system of the system in the normal form (Eq. 2.46 and Eq. 2.47). The output equation of the inverse system is already available as the function $\bar{h} = \gamma$, γ as defined in Eq. 2.50. With $y(k) = y_d(k)$ known, the vectors $\xi(k)$ and $\xi_{r_i}^{[1,m]}(k+1)$ follow directly from their definitions. However, to determine $u(k)$ we still need to determine $\eta(k)$. To this end we substitute $\bar{h} = \gamma$ into Eq. 2.47, giving the $\bar{n}$ dimensional system

$$\begin{aligned} \eta(k+1) &= q(\xi(k), \eta(k), \bar{h}(\xi(k), \eta(k), \xi_{r_i}^{[1,m]}(k+1))) && (2.54)\\ &:= \bar{f}(\xi(k), \eta(k), \xi_{r_i}^{[1,m]}(k+1)) && (2.55)\\ &:= \bar{f}(\eta(k), \Xi(k)) \ , && (2.56) \end{aligned}$$

in which $\Xi(k)$ represents $\xi(k)$ and $\xi_{r_i}^{[1,m]}(k+1)$. The smoothness $\bar{h}$ and q implies the smoothness of $\bar{f}$. The inverse system is thus given by

$$\begin{aligned} \eta(k+1) &= \bar{f}(\eta(k), \Xi(k)) && (2.57)\\ u(k) &= \bar{h}(\eta(k), \Xi(k)) \ . && (2.58) \end{aligned}$$

with $\bar{h} = \gamma$ as in Eq. 2.50. Now define

$$\mathcal{U}(\eta(k), \Xi(k)) := \bar{f}(\eta(k), \Xi(k)) - A\eta(k) \tag{2.59}$$

with

$$A := \left. \frac{\partial \bar{f}(\eta(k), \Xi(k))}{\partial \eta(k)} \right|_{[0],[0]} \tag{2.60}$$

A has l eigenvalues inside the unit circle and $\hat{n} - l$ eigenvalues outside the unit circle. The state dynamics, Eq. 2.57, may be restated as

$$\eta(k+1) = A\eta(k) + \mathcal{U}(\eta(k), \Xi(k)) \ , \tag{2.61}$$

which is structured like a linear system - a fact that is subsequently utilized in constructing a bounded solution for the system. Noting the existence of a similarity transformation that transforms A into

the Jordan form, in the sequel we assume without loss in generality that Eq. 2.61 is already in this form, for which

$$A = \begin{bmatrix} A_s & [0] \\ [0] & A_u \end{bmatrix} , \tag{2.62}$$

with A_s consisting of Jordan blocks representing the l eigenvalues inside the unit circle, and A_u consisting of Jordan blocks representing the $\hat{n} - l$ eigenvalues outside the unit circle. The bounded state transition matrix for the linear matrix difference equation $\eta(k+1) = A\eta(k)$ is the $\hat{n} \times \hat{n}$ matrix $\phi(k)$, $k \in \mathcal{Z}$, given as

$$\phi(k) = \begin{cases} \begin{bmatrix} A_s^k & [0] \\ [0] & [0] \end{bmatrix} , & k > 0 \\ \\ \begin{bmatrix} [0] & [0] \\ [0] & -A_u^k \end{bmatrix} , & k < 0 , \end{cases} \tag{2.63}$$

with

$$\begin{aligned} \phi(0^+) &= \begin{bmatrix} I_s & [0] \\ [0] & [0] \end{bmatrix} \\ \phi(0^-) &= \begin{bmatrix} [0] & [0] \\ [0] & -I_u \end{bmatrix} , \end{aligned}$$

and I_s and I_u the $l \times l$ and $(\hat{n} - l) \times (\hat{n} - l)$ identity matrices respectively. Assuming that $y_d(k)$, and thus $\Xi(k)$, $k \in \mathcal{Z}$, is bounded, and that $\bar{f}(\eta(k), \Xi(k))$ is bounded if $\eta(k)$ and $\Xi(k)$ are bounded, it can be shown (Zeng *et al.* 2000) that the bounded solution $\eta(k)$ of Eq. 2.61 (and thus Eq. 2.57) is equivalent to the bounded solution $\eta(k)$, $k \in \mathcal{Z}$, of

$$\begin{aligned} \eta(k) &= \sum_{i=-\infty}^{k-1} \phi(k-i)\mathcal{U}(\eta(i-1), \Xi(i-1)) \\ &+ \phi(0^+)\mathcal{U}(\eta(k-1), \Xi(k-1)) \\ &+ \sum_{i=k+1}^{\infty} \phi(k-i)\mathcal{U}(\eta(i-1), \Xi(i-1)) . \end{aligned} \tag{2.64}$$

For convenience we will designate this solution of Eq. 2.61 more compactly as

$$\eta(k) = \sum_{i=-\infty}^{\infty} \phi(k-i)\mathcal{U}(\eta(i-1), \Xi(i-1)) . \tag{2.65}$$

Define a non-causal linear operator G (Kreyszig, 1978:49) to represent the linearly-structured system in Eq. 2.61 as

$$\eta = G\mathcal{U}(\eta, \Xi) . \tag{2.66}$$

The procedure of obtaining the bounded solution, $\eta(k)$, for Eq. 2.61, and thus Eq. 2.57, by using the stable solution of $\phi(k)$ in Eq. 2.64 lies at the heart of stable inversion. Before discussing the calculation of the bounded $\eta(k)$, some norm definitions are in order. Proceeding as in Zeng *et al.* (2000), let $||\cdot||_1$ and $||\cdot||_\infty$ denote the l^1 and l^∞ norms respectively on $\mathcal{Z}$ (Kreyszig, 1978:61). Also let

$$||\eta||_\infty = \max_i ||\eta_i||_\infty \ , \tag{2.67}$$

where i denotes the state vector element.

The function $\mathcal{U}(\eta(k), \Xi(k))$ is said to be *uniformly Lipschitz* in a closed s neighbourhood of ([0],[0]) in (η, Ξ) space with positive real constants (K_1, K_2) if an $s > 0$ exists such that for all $\eta_1(k)$, $\eta_2(k)$, $\Xi_1(k)$ and $\Xi_2(k)$, all with $||\cdot||_\infty$ norms $\leq s$, the following local Lipschitz condition holds uniformly $\forall k \in \mathcal{Z}$:

$$||\,\mathcal{U}(\eta_1(k), \Xi_1(k)) - \mathcal{U}(\eta_2(k), \Xi_2(k))||_\infty \leq K_1||\eta_1(k) - \eta_2(k)||_\infty + K_2||\Xi_1(k) - \Xi_2(k)||_\infty \ . \tag{2.68}$$

Furthermore, define

$$||\phi(k)||_\infty = \sup\{||\phi(k)c(k)||_\infty : ||c(k)||_\infty = 1\} \tag{2.69}$$

$$||\phi||_1 = \hat{n} \max_{i,j} ||\phi_{i,j}(k)||_1 \tag{2.70}$$

where $\phi_{i,j}(k)$ is the (i,j)-th element of $\phi(k)$. Finally we note that

$$\begin{aligned} ||G\mathcal{U}(\eta, \Xi)||_\infty &= || \sum_{i=-\infty}^{\infty} \phi(k-i)\mathcal{U}(\eta(i-1), \Xi(i-1))||_\infty \\ &\leq ||\phi||_1 ||\mathcal{U}(\eta, \Xi)||_\infty \ . \end{aligned} \tag{2.71}$$

2.5 Iterative Solution of the Inverse

2.5.1 Introduction

The bounded $\eta(k)$ satisfying Eq. 2.64, which is in the form of the solution of a linear system, is iteratively obtained. (The linear system in this case is the nonlinear system Eq. 2.57, which is a first order nonlinear difference equation, structured as a linear system in Eq. 2.61.) This is done by recasting Eq. 2.64 as a fixed point problem by performing iterative searching of the fixed point of Eq. 2.66. Three of the available iteration schemes that may be used for this purpose are now listed, namely Picard iteration, Mann iteration, and Ishikawa iteration. Consider a mapping $T : B \to B$, with B a non-empty, convex subset of a normed space X. *Picard iteration* is defined by the sequence $(\eta^{(m)} : m \in \mathcal{Z})$, $\eta^{(0)} \in B$, with

$$\eta^{(m+1)} = T(\eta^{(m)}) \ . \tag{2.72}$$

Mann iteration is defined by the sequence $(\eta^{(m)} : m \in \mathcal{Z})$, $\eta^{(0)} \in B$, $\alpha_m \in (0, 1]$, with

$$\eta^{(m+1)} = (1 - \alpha_m)\eta^{(m)} + \alpha_m T(\eta^{(m)}) \; . \tag{2.73}$$

The usual choice for α_m in Mann iteration is $\alpha_m = 1/(1 + m)$ or $\alpha_m = 1/m$.

Finally, *Ishikawa iteration* is defined by the sequence $(\eta^{(m)} : m \in \mathcal{Z})$, $\eta^{(0)} \in B$, $\alpha_m \in (0, 1]$, $\beta_m \in [0, 1]$, with

$$\begin{aligned} \eta^{(m+1)} &= (1 - \alpha_m)\eta^{(m)} + \alpha_m T(\mu^{(m)}) \\ \mu^{(m)} &= (1 - \beta_m)\eta^{(m)} + \beta_m T(\eta^{(m)}) \; . \end{aligned} \tag{2.74}$$

Let η designate a fixed point of Picard iteration. Then

$$\eta = \lim_{m\to\infty} \eta^{(m)} = \lim_{m\to\infty} \eta^{(m+1)} \; ,$$

and by taking the limit on both sides of Eq. 2.73 we get

$$\eta = (1 - \alpha_m)\eta + \alpha_m T(\eta) \; ,$$

from which it is clear (after simplification of this equation) that the fixed points of Picard and Mann iteration are equivalent. It can also be shown that a fixed point of Picard iteration is also a fixed point of Ishikawa iteration.

Mann and Ishikawa iteration may be simplified by using constant scale factors $\alpha_m = \alpha$ and $\beta_m = \beta$. Clearly, if $\alpha_n = 1$ then Picard iteration is obtained from Mann iteration. Furthermore, if $\beta_m = 0$ then Mann iteration is obtained from Ishikawa iteration.

2.5.2 Picard Iteration Method

The adaptation of Eq. 2.64 for Picard iteration gives the sequence $(\eta^{(m)}(k) : m \in \mathcal{Z})$ with

$$\begin{aligned} \eta^{(0)}(k) &= [0] \\ \eta^{(m+1)}(k) &= \textstyle\sum_{i=-\infty}^{\infty} \phi(k - i)\mathcal{U}(\eta^{(m)}(i - 1), \Xi(i - 1)) \; . \end{aligned} \tag{2.75}$$

This may be rewritten in operator form as (cf. Eq. 2.66)

$$\eta^{(m+1)} = G\mathcal{U}(\eta^{(m)}, \Xi) \; . \tag{2.76}$$

If the sequence $(\eta^{(m)})$ converges then

$$\eta(k) = \lim_{m\to\infty} \eta^{(m)}(k) \; . \tag{2.77}$$

Sufficient conditions for the convergence of the Picard iteration scheme Eq. 2.75 are now presented from Zeng *et al.* (2000), with slight modification to the uniqueness aspects, because of its usefulness of these results for subsequent theorems on Mann iteration:

Theorem 2.1: A unique solution $\eta(k) \in l^\infty$, $k \in \mathcal{Z}$, of Eq. 2.57 exists and is obtained by the Picard iteration of Eq. 2.75 if:

1. $\bar{f}$ is uniformly Lipschitz in an s neighbourhood of ([0],[0]) with Lipschitz constants (K_1, K_2),
2. $||\phi||_1 K_1 < 1$,
3. $\Xi(k) \in l^\infty$ with $||\Xi||_\infty \leq s$, and
4. $(||\phi||_1 K_2 ||\Xi||_\infty / (1 - ||\phi||_1 K_1)) \leq s$.

Proof: Let $k \in \mathcal{Z}$. Clearly $\eta^{(0)} \in l^\infty$ and $||\eta^{(0)}||_\infty \leq s$. Suppose $\eta^{(m)} \in l^\infty$ and $||\eta^{(m)}||_\infty \leq s$. Then, by condition 1 and 4:

$$\begin{aligned} ||\eta^{(m+1)}(k)||_\infty &= ||G\mathcal{U}(\eta^{(m)}, \Xi)||_\infty \\ &\leq ||\phi||_1 ||\mathcal{U}(\eta^{(m)}, \Xi)||_\infty \\ &\leq ||\phi||_1 (K_1 ||\eta^{(m)}||_\infty + K_2 ||\Xi||_\infty) \\ &\leq ||\phi||_1 (K_1 s + K_2 ||\Xi||_\infty) \\ &\leq s \; . \end{aligned} \tag{2.78}$$

Thus, by induction $\eta^{(m)} \in l^\infty$ and $||\eta^{(m)}||_\infty \leq s$ for all m. Define

$$H^{(m)}(k) = \eta^{(m+1)} - \eta^{(m)} \tag{2.79}$$

Then, by condition 1 and defining $\mathcal{U}^{(m)} := \mathcal{U}(\eta^{(m)}, \Xi)$:

$$\begin{aligned} ||H^{(m)}(k)||_\infty &= ||\eta^{(m+1)} - \eta^{(m)}||_\infty \\ &= ||G\mathcal{U}^{(m)} - G\mathcal{U}^{(m-1)}||_\infty \\ &= ||G(\mathcal{U}^{(m)} - \mathcal{U}^{(m-1)})||_\infty \\ &\leq ||\phi||_1 ||\mathcal{U}^{(m)} - \mathcal{U}^{(m-1)}||_\infty \\ &\leq ||\phi||_1 K_1 ||\eta^{(m)} - \eta^{(m-1)}||_\infty \\ &= ||\phi||_1 K_1 ||H^{(m-1)}(k)||_\infty \end{aligned} \tag{2.80}$$

Since $||\phi||_1 K_1 < 1$, by the ratio test (Bartle *et al.* 2000:66) the series $\sum_{m=0}^{\infty} ||H^{(m)}(k)||_\infty$ is convergent. Hence

$$\eta^{(m)}(k) = \sum_{j=1}^{m-1} H^{(j)}(k) \; , \tag{2.81}$$

$k \in Z$ is a Cauchy sequence (Bartle *et al.* 2000:81) in l^∞. Since l^∞ is complete (l^∞ is a Banach space), every Cauchy sequence in l^∞ converges to an element of l^∞, and therefore $(\eta^{(m)}(k))$ converges (in the $||\cdot||_\infty$ norm) to an element of l^∞. Denote this limit element as $\eta(k)$, $k \in \mathcal{Z}$. Now, for a $\eta_a, \eta_b \in l^\infty$ it follows that (from Condition 1)

$$||G\mathcal{U}(\eta_a, \Xi) - G\mathcal{U}(\eta_b, \Xi)||_\infty \leq ||\phi||_1 K_1 ||\eta_a - \eta_b||_\infty \; , \tag{2.82}$$

implying the uniform continuity (Bartle *et al.* 2000:136) of $G\mathcal{U}(\eta, \Xi)$ with respect to η, from which

$$G\mathcal{U}(\eta, \Xi) = G\mathcal{U}(\lim_{m\to\infty} \eta^{(m)}, \Xi) = \lim_{m\to\infty} G\mathcal{U}(\eta^{(m)}, \Xi) = \lim_{m\to\infty} \eta^{(m+1)} = \eta \; .$$

Thus η is the fixed point of $G\mathcal{U}(\eta, \Xi)$ and therefore of Eq. 2.64, which implies that $\eta(k)$ is the solution of the inverse system's state equation, Eq. 2.57. Finally, this solution η is unique since, if $\eta_c \in l^\infty$ is another solution, then

$$\begin{aligned} ||\eta - \eta_c||_\infty &= ||G\mathcal{U}(\eta, \Xi) - G\mathcal{U}(\eta_c, \Xi)||_\infty \\ &\leq ||\phi||_1 K_1 ||\eta - \eta_c||_\infty \; . \end{aligned}$$

Since $||\phi||_1 K_1 < 1$ this inequality can only be satisfied if $\eta = \eta_c$. $\diamondsuit$

Condition 2 implies that Eq. 2.75 is a contraction (Kreyszig, 1978:300) between $\eta^{(m)}$ and $\eta^{(m+1)}$. Condition 4 implies that

$$||\phi||_1 (K_1 s + K_2 ||\Xi||_\infty) \leq s \; .$$

This is basically an extension of condition 2 for the additional variable Ξ since, when $||\Xi||_\infty \to 0$ this becomes $||\phi||_1 K_1 \leq 1$, which is satisfied by condition 2. Theorem 2.1 thus essentially says that if Eq. 2.64 is a contraction mapping, then Picard iteration converges to the unique fixed point. In the sequel we show that when this is the case Mann iteration also converges. However, when $||\phi||_1 K_1 > 1$, and Eq. 2.64 is therefore not a contraction, we do not have a general condition of convergence for either Picard, Mann or Ishikawa iteration.

The proof of Theorem 2.1 deviates somewhat from the approach of Zeng (2000), with the uniqueness component of the proof similar to the approach taken in the proof of the Banach fixed point theorem (contraction theorem) in Kreyszig (1978:302).

2.5.3 Mann Iteration Method

Consider again a mapping $T : B \to B$, with B a non-empty, convex subset of a normed space X. For the case where $T : B \to B$ is a contraction, i.e. there exists an $L < 1$ such that the Lipschitz condition

$$||Tx - Ty|| \leq L||x - y||$$

is satisfied for all $x, y \in B$, the convergence of Picard, Mann and Ishikawa iteration is proved in Rhoades *et al.* (2003). However, as in the case above for Eq. 2.64, when T is L-Lipschitz with $L > 1$ (and therefore not contractive), we do not have a general condition of convergence of either Picard, Mann or Ishikawa iteration. Convergence conditions for Mann or Ishikawa iteration do exist for some special cases though:

- Chidume and Chidume (2006) give convergence conditions when $T : B \to B$ is a nonexpansive with B a non-empty, closed, convex subset of a real Banach space X. (An operator $T : B \to B$ is nonexpansive if $||Tx - Ty|| \leq ||x - y||$ is satisfied for all $x, y \in B$.)

- Chidume and Osilike (1999) give convergence conditions when $T : X \to X$ is a uniformly continuous, strongly accretive operator with X an arbitrary real Banach space. (An operator $T : X \to X$ is strongly accretive if $||x - y|| < ||x - y + s(Tx - Ty)||$ is satisfied for all $x, y \in X$ and $s > 0$.)

- Osilike (2000) covers the more general case when $T : X \to X$ is ϕ-strongly accretive and X a real, uniformly smooth Banach space.

- Chidume and Osilike (1999) give convergence conditions when $T : B \to B$ is a uniformly continuous, strong pseudocontraction with B a nonempty closed convex bounded subset of an arbitrary real Banach space X. (An operator $T : B \to B$ is a strong pseudocontraction if there exists a $t > 1$ such that $||x - y|| \leq ||(1 + t)(x - y) - tr(Tx - Ty)||$ is satisfied for all $x, y \in B$ and $r > 0$.)

- Chidume and Udomene (2006) give convergence conditions when $T : B \to B$ is a uniformly continuous pseudocontraction with B a nonempty closed convex bounded subset of an arbitrary real Banach space X. This is however for a different type of iteration not covered in this study.

The situation that is encountered in most of the theoretical examples in the sequel is that Mann iteration (as well as Picard iteration) diverges for a variety of values of α_n, suggesting that none of the above-mentioned special cases apply.

The following two theorems are both a result of this research. The first (Theorem 2.2) gives sufficient conditions for the convergence of Mann iteration in stable inversion in rather empirical terms. While of little practical use, it is used to show in the second (Theorem 2.3) that Mann iteration at least converges under weaker conditions than Picard iteration in stable inversion. The theorems are useful in the situation where solution of the inverse system model is not contractive and Picard iteration is likely to diverge, because it implies that at least Mann iteration still has a chance to converge, or can reasonably be expected to give more accurate best results prior to divergence than Picard iteration.

The adaptation of Eq. 2.65 for Mann iteration gives the sequence $(\eta^{(m)}(k) : m \in \mathcal{Z})$ with

$$\begin{aligned} \eta^{(0)}(k) &= [0] \\ \eta^{(m+1)}(k) &= (1 - \alpha_m)\eta^{(m)}(k) + \alpha_m \textstyle\sum_{i=-\infty}^{\infty} \phi(k - i)\mathcal{U}(\eta^{(m)}(i - 1), \Xi(i - 1)) \; . \end{aligned} \tag{2.83}$$

In the sequel we will assume a constant $\alpha_m = \alpha$. Eq. 2.83 may be restated in operator form (using a constant α) as

$$\eta^{(m+1)} = (1 - \alpha)\eta^{(m)} + \alpha G\mathcal{U}(\eta^{(m)}, \Xi) \tag{2.84}$$

$$= (1-\alpha)\eta^{(m)} + \alpha G\mathcal{U}^{(m)} , \tag{2.85}$$

in which we defined $\mathcal{U}^{(m)} := \mathcal{U}(\eta^{(m)}, \Xi)$. The adaptation for Ishikawa iteration is done in a similar way. Before deriving convergence conditions some definitions are in order, for which we assume $k \in \mathcal{Z}$. As before, define (using Eq. 2.85)

$$\begin{aligned} H^{(m)} &:= \eta^{(m+1)} - \eta^{(m)} && (2.86)\\ &= (1-\alpha)(\eta^{(m)} - \eta^{(m-1)}) + \alpha G(\mathcal{U}^{(m)} - \mathcal{U}^{(m-1)}) \\ &= (1-\alpha)H^{(m-1)} + \alpha\tilde{H}^{(m)} , && (2.87) \end{aligned}$$

in which we defined

$$\tilde{H}^{(m)} := G(\mathcal{U}^{(m)} - \mathcal{U}^{(m-1)}) . \tag{2.88}$$

Taking the norm of $\tilde{H}^{(m)}$ we have (using Eq. 2.71 and Eq. 2.68):

$$\begin{aligned} ||\tilde{H}^{(m)}||_\infty &= ||G(\mathcal{U}(\eta^{(m)}, \Xi) - \mathcal{U}(\eta^{(m-1)}, \Xi))||_\infty \\ &\leq ||\phi||_1 ||\mathcal{U}(\eta^{(m)}, \Xi) - \mathcal{U}(\eta^{(m-1)}, \Xi)||_\infty \\ &\leq ||\phi||_1 K_1 ||\eta^{(m)} - \eta^{(m-1)}||_\infty \\ &= ||\phi||_1 K_1 ||H^{(m-1)}||_\infty . && (2.89) \end{aligned}$$

Define $\tilde{s}_m \leq ||\phi||_1 K_1$ such that

$$||\tilde{H}^{(m)}||_\infty = \tilde{s}_m ||H^{(m-1)}||_\infty . \tag{2.90}$$

Taking the norm of $H^{(m)}$ (Eq. 2.87) and defining $F_m(\alpha, \eta^{(0)}) \in [0, 1]$ we get

$$\begin{aligned} ||H^{(m)}||_\infty &= F_m((1-\alpha)||H^{(m-1)}||_\infty + \alpha||\tilde{H}^{(m)}||_\infty) \\ &= F_m((1-\alpha)||H^{(m-1)}||_\infty + \alpha\tilde{s}_m||H^{(m-1)}||_\infty) \\ &= s_m||H^{(m-1)}||_\infty , && (2.91) \end{aligned}$$

in which we defined s_m as

$$s_m := F_m(1 - \alpha + \alpha\tilde{s}_m) . \tag{2.92}$$

Theorem 2.2: A solution $\eta(k) \in l^\infty$, $k \in \mathcal{Z}$, of Eq. 2.57 exists and is obtained by the Mann iteration of Eq. 2.83 if the following conditions are satisfied:

1. $\bar{f}$ is uniformly Lipschitz in an s neighbourhood of ([0],[0]) with Lipschitz constants (K_1, K_2),
2. $||s_m||_\infty < 1$,
3. $\Xi(k) \in l^\infty$ with $||\Xi||_\infty \leq s$, and

4. $(||\phi||_1 K_2||\Xi||_\infty/(1-||\phi||_1 K_1)) \leq s$.

Proof: Let $k \in \mathcal{Z}$. Clearly $\eta^{(0)} \in l^\infty$ and $||\eta^{(0)}||_\infty \leq s$. Suppose $\eta^{(m)} \in l^\infty$ and $||\eta^{(m)}||_\infty \leq s$. Taking the norm of Eq. 2.84 we get (using conditions 1 and 4):

$$\begin{aligned} ||\eta^{(m+1)}||_\infty &\leq (1-\alpha)||\eta^{(m)}||_\infty + \alpha||G\mathcal{U}(\eta^{(m)},\Xi)||_\infty \\ &\leq (1-\alpha)||\eta^{(m)}||_\infty + \alpha||\phi||_1(K_1||\eta^{(m)}||_\infty + K_2||\Xi||_\infty) \\ &\leq (1-\alpha)s + \alpha||\phi||_1(K_1 s + K_2||\Xi||_\infty) \\ &\leq (1-\alpha)s + \alpha s \\ &= s\ . \end{aligned} \tag{2.93}$$

Thus, by induction $\eta^{(m)} \in l^\infty$ and $||\eta^{(m)}||_\infty \leq s$ for all m. Furthermore, from condition 2 and Eq. 2.91 it follows that $||H^{(m)}||_\infty < ||H^{(m-1)}||_\infty$, and by the ratio test the series $\sum_{m=0}^{\infty}||H^{(m)}(k)||_\infty$ is convergent. Hence

$$\eta^{(m)}(k) = \sum_{j=1}^{m-1} H^{(j)}(k)\ , \tag{2.94}$$

$k \in \mathcal{Z}$ is a Cauchy sequence in l^∞. Since l^∞ is complete (l^∞ is a Banach space), every Cauchy sequence in l^∞ converges to an element of l^∞, and therefore $\{\eta^{(m)}(k)\}$ converges (in the $||\cdot||_\infty$ norm) to an element of l^∞. Denote this limit element as $\eta(k)$, $k \in \mathcal{Z}$. Now, for a $\eta_a, \eta_b \in l^\infty$ it follows that

$$\begin{aligned} &||(1-\alpha)\eta_a + \alpha G\mathcal{U}(\eta_a,\Xi) - ((1-\alpha)\eta_b + \alpha G\mathcal{U}(\eta_b,\Xi))||_\infty \\ &= ||(1-\alpha)(\eta_a-\eta_b) + \alpha G(\mathcal{U}(\eta_a,\Xi) - \mathcal{U}(\eta_b,\Xi))||_\infty \\ &\leq (1-\alpha)||(\eta_a-\eta_b)||_\infty + \alpha||G(\mathcal{U}(\eta_a,\Xi) - \mathcal{U}(\eta_b,\Xi))||_\infty \\ &\leq (1-\alpha)||\eta_a-\eta_b||_\infty + \alpha||\phi||_1 K_1||\eta_a-\eta_b||_\infty \\ &= (1-\alpha+\alpha||\phi||_1 K_1)||\eta_a-\eta_b||_\infty\ , \end{aligned}$$

implying the uniform continuity of $(1-\alpha)\eta + \alpha G\mathcal{U}(\eta,\Xi)$ with respect to η, from which

$$\begin{aligned} (1-\alpha)\eta + \alpha G\mathcal{U}(\eta,\Xi) &= (1-\alpha)\lim_{m\to\infty}\eta^{(m)} + \alpha G\mathcal{U}(\lim_{m\to\infty}\eta^{(m)},\Xi) \\ &= \lim_{m\to\infty}((1-\alpha)\eta^{(m)} + \alpha G\mathcal{U}(\eta^{(m)},\Xi)) \\ &= \lim_{m\to\infty}\eta^{(m+1)} \\ &= \eta\ . \end{aligned} \tag{2.95}$$

Thus η is the fixed point of $(1-\alpha)\eta + \alpha G\mathcal{U}(\eta,\Xi)$ and therefore of Eq. 2.65, which implies that $\eta(k)$ is the solution of the inverse system's state equation, Eq. 2.57. $\diamondsuit$

Note that Theorem 2.2 do not prove uniqueness as in Theorem 2.1, mostly because of the restrictive nature of the definition of F_m. The reason for this choice of F_m (and s_m) is to cover all mechanisms contributing to the possible convergence (or less aggressive divergence) of Mann iteration compared to Picard iteration when the latter diverges, i.e. when $||\phi||_1 K_1 \geq 1$ and Eq. 2.76 is thus not contractive. Substituting $||\phi||_1 K_1 \geq 1$ into $\tilde{s}_m$ in s_m, we observe that the resulting quantity $S = 1-\alpha+\alpha||\phi||_1 K_1 \geq 1$ is already smaller than $||\phi||_1 K_1$, before considering that $\tilde{s}_m \leq ||\phi||_1 K_1$. Further noting that $F_H^{(m)} \leq 1$, it is clear that Condition 2, Theorem 2.2 represents a weaker convergence condition than Condition 2, Theorem 2.1. Even if condition 2, Theorem 2.2 fails, Mann iteration may still converge because it is a sufficient condition. If it does not converge, in view of the above considerations it is reasonable to expect better approximate inversion results than with Picard iteration. The following theorem formalizes these issues:

Theorem 2.3: The convergence conditions in Theorem 2.2 for Mann iteration are weaker than the convergence conditions in Theorem 2.1 for Picard iteration since

- (1) if condition 2, Theorem 2.1 is satisfied, then the condition 2, Theorem 2.2 is also satisfied,
- (2) if condition 2, Theorem 2.1 is violated then condition 2, Theorem 2.2 may still be satisfied, and
- (3) if condition 2, Theorem 2.2 is violated, then condition 2, Theorem 2.1 is also violated.

Proof: (1) In this case $\tilde{s}_m \leq ||\phi||_1 K_1 < 1$. Since $\alpha \in (0, 1]$, it follows that $\tilde{s}_m \leq 1 - \alpha + \alpha\tilde{s}_m < 1$. Noting that $F_H^{(m)} \leq 1$ in Eq. 2.92 it follows that $s_m < 1$, and thus $||s_m||_\infty < 1$.

(2) In this case $\tilde{s}_m < ||\phi||_1 K_1 \geq 1$. If $\tilde{s}_m < 1$, then $\tilde{s}_m \leq 1 - \alpha + \alpha\tilde{s}_m < 1 \leq ||\phi||_1 K_1$, and thus $s_m < 1 \leq ||\phi||_1 K_1$, implying convergence by Condition 2. If $1 \leq \tilde{s}_m \leq ||\phi||_1 K_1$, then $1 \leq 1 - \alpha + \alpha\tilde{s}_m \leq \tilde{s}_m \leq ||\phi||_1 K_1$, and again $s_m \leq ||\phi||_1 K_1$. Therefore $||s_m||_\infty \leq ||\phi||_1 K_1$, and condition 2 may still be satisfied, or have less aggressive divergence than Picard iteration.

(3) If $||s_m||_\infty \geq 1$, then there exists an m such that $s_m \geq 1$, implying by Eq. 2.92 that $1 \leq 1 - \alpha + \alpha\tilde{s}_m \leq \tilde{s}_m$. Recalling that $\tilde{s}_m \leq ||\phi||_1 K_1$, it follows that $||\phi||_1 K_1 \geq 1$. $\diamondsuit$

2.5.4 Filter Incorporation

The results of subsequent theoretical examples show that when stable inversion diverges, incorporating a zero phase low pass filter in the iteration my improve the accuracy of the best results prior to divergence, or even lead to convergence. This is done here by incorporating the filter, which is designated F in operator form, into the inverse system's solution formula, Eq. 2.66, as follows:

$$\eta = FG\mathcal{U}(\eta, \Xi) \ . \tag{2.96}$$

F is a linear operator representing a zero phase filter based on the causal discrete time filter F_0 (i.e. in the frequency domain F is represented by $|F_0(e^{j\omega T_s})|^2$). We can adapt Eq. 2.96 for fixed point iteration in the same way as was done for Eq. 2.66, resulting in the following modified iteration schemes: Picard iteration:

$$\eta^{(m+1)} = FG\mathcal{U}(\eta^{(m)}, \Xi) \tag{2.97}$$

Mann iteration:

$$\eta^{(m+1)} = (1-\alpha_m)\eta^{(m)} + \alpha_m FG\mathcal{U}(\eta^{(m)}, \Xi) \tag{2.98}$$

Ishikawa iteration:

$$\begin{aligned} \eta^{(m+1)} &= (1-\alpha_m)\eta^{(m)} + \alpha_m FG\mathcal{U}(\mu^{(m)}, \Xi) \\ \mu^{(m)} &= (1-\beta_m)\eta^{(m)} + \beta_m FG\mathcal{U}(\eta^{(m)}, \Xi) \end{aligned} \tag{2.99}$$

Note that F above is closely associated with G, and therefore impacts the convergence conditions in Theorem 2.1 and Theorem 2.2 in the same way that $G\mathcal{U}$ does. Note also that by setting

$$\eta = \lim_{i\to\infty} \eta^{(i)} = \lim_{i\to\infty} \eta^{(i+1)} \ ,$$

$$\eta^{(m+1)} = F((1-\alpha_m)\eta^{(m)} + \alpha_m G\mathcal{U}(\eta^{(m)}, \Xi)) \ , \tag{2.100}$$

which we call here the Type II filtering approach. Clearly Eq. 2.100 does not have the same fixed points as Eq. 2.97.

2.6 Example 1: Mann Iteration and Low Pass Filtering - Deterministic Signal

In this example the advantages of using Mann iteration and low pass filtering is demonstrated for a short-duration deterministic signal. First, however, the ability of both Picard and Mann iteration to converge without using a low-pass filter is demonstrated when the system is operated in a relatively low level of nonlinearity regime.

Consider the following NARX system (Chen *et al.* 1989a)

$$\begin{aligned} y(k) \quad = \quad & \theta_1 u(k-4) + \theta_2 u(k-5) + \theta_3 u(k-6) + \theta_4 y(k-6) \\ & \theta_5 u(k-5)y(k-4) + \theta_6 u(k-5)u(k-6)y(k-4) \\ & +\theta_7 u(k-5)^2 u(k-6)y(k-5) \ , \end{aligned} \tag{2.101}$$

with

$$(\theta_1, \ldots, \theta_7) = (0.150, \ -1/12, \ -1/6, \ 1/6, \ -4.0, \ 6.0, \ 11.0) \ . \tag{2.102}$$

This kind of nonlinear system may readily be obtained from nonlinear system identification methods (Chen *et al.* 1989b). When converted to the nonlinear state space formulation, then to the normal form, and finally inverted (more details on this are presented in Chapter 4), this system gives rise to the following inverse system:

$$\begin{aligned} \eta_1(k+1) &= C_1(k)\,\eta_1(k) + C_2\,\eta_2(k) + C_3(k)\,\eta_1(k)\,\eta_2(k) + C_4(k)\,\eta_1(k)^2\,\eta_2(k) + C_0(k) \\ \eta_2(k+1) &= \eta_1(k) \\ u(k-5) &= \eta_1(k) \end{aligned} \tag{2.103}$$

$k = 1, \ldots, N$, with

$$\begin{aligned} C_0(k) &= \frac{1}{\theta_1} y(k) - \frac{\theta_4}{\theta_1} y(k-6) \\ C_1(k) &= -\frac{\theta_2}{\theta_1} - \frac{\theta_5}{\theta_1} y(k-4) \\ C_2 &= -\frac{\theta_3}{\theta_1} \\ C_3(k) &= -\frac{\theta_6}{\theta_1} y(k-4) \\ C_4(k) &= -\frac{\theta_7}{\theta_1} y(k-5)\ . \end{aligned}$$

A sample frequency of 250 Hz is assumed where relevant.

The desired response in question is obtained as the response of Eq. 2.111 to the following relatively short deterministic signal, $u_d(k)$:

$$\bar{u}_d(k) = \begin{cases} 0, & 1 \le k \le 25 \\ c_u(\sin(2\pi(k-31)/20) + 1), & 25 < k \le 46 \\ 0, & 46 < k \le 146 \end{cases} \tag{2.104}$$

$$\tilde{u}_d = \bar{F}_{0.16}\bar{u}_d \tag{2.105}$$

$$u_d = C_T \tilde{u}_d \tag{2.106}$$

$$F_{0.16}(z) = \frac{0.02287z^4 + 0.09148z^3 + 0.13722z^2 + 0.09148z + 0.02287}{1.00z^4 - 1.412z^3 + 1.123z^2 - 0.40807z + 0.06321} \tag{2.107}$$

$$C_T = \begin{cases} 0, & 1 \le k \le 12 \\ 0.5\sin(2\pi(k-13)/24 - \pi/2) + 0.5, & 12 < k \le 24 \\ 1, & 24 < k \le 122 \\ 0.5\sin(2\pi(k-123)/24 + \pi/2) + 0.5, & 122 < k \le 134 \\ 0, & 134 < k \le 146 \end{cases} \tag{2.108}$$

with $c_u = 0.165$, the second equation in operator format, and $\bar{F}$ a non-causal linear operator representing the zero phase version of the low pass filter $F(z)$ with cut frequency 40 Hz (0.16 times the

sample frequency). C_T as given by Eq. 2.108 is essentially a sinusoidal taper function. Two cases are performed in this example, namely for $c_u = 0.25$ (Case 1) and $c_u = 0.445$ (Case 2). A corresponding desired output trajectory $y_d(k)$ was generated for the two $u_d(k)$ by applying each $u_d(k)$ to the original system (Eq. 2.101). The desired input signal and desired response signal is shown in Fig. 2.1 and Fig. 2.2 for Case 1 and Case 2 respectively.

We focus on the stable inversion of Eq. 2.101, i.e. attempting to determine the bounded solution of Eq. 2.103 by stable inversion for the given $y(k) = y_d(k)$ for the two cases. This will be achieved by searching the bounded solution of Eq. 2.103 by using either Picard or Mann iteration. The desired $u(k)$ may be obtained from the best achieved solution of η as $u(k) = \eta_1(k+5)$ (cf. Eq. 2.103). The percentage error between $u_d(k)$ and the input calculated during iteration m of stable inversion, namely $u^{(m)}(k)$, is defined as:

$$\begin{aligned} \mathrm{err}_1(u^{(m)}) &:= 100\frac{\sum_{k=1}^{N}|u^{(m)}(k) - u_d(k)|}{\sum_{k=1}^{N}|u_d(k)|} \\ &= 100\frac{||u^{(m)}(k) - u_d(k)||_1}{||u_d(k)||_1} \ . \end{aligned} \tag{2.109}$$

Similarly

$$\mathrm{err}_1(y^{(m)}) := 100\,||y^{(m)}(k) - y_d(k)||_1/||y_d(k)||_1 \ . \tag{2.110}$$

When stable inversion converges towards the desired input signal, the last iteration may be used for the final results. When it diverges, the results of the "best" needs to be chosen. Since the purpose of stable inversion (and ILC) is the determination of an unknown input signal, the percentage error between $u_d(k)$ and the inputs calculated during successive iterations is not normally available. One approach to circumventing this problem is to evaluate some norm of the calculated input signal for successive iterations, and choose an iteration number based on the behavior of this norm over successive iterations. This is referred to here as the input (or u) - based iteration selection strategy.

Another strategy for selecting the final iteration number is to calculate the output signals that are predicted by the system for the successive, calculated input signals, evaluate the percentage error (or other norm of the error) of each with respect to the desired output signal, and select the iteration number giving the smallest output signal error. This is called here the output (or y) - based iteration selection strategy, and is the one used in these examples (unless specified otherwise).

The results of stable inversion for the two cases using Picard and Mann iteration is shown in Table 2.1. For Case 1 both approaches converge even without using a low pass filter in iteration. For Case 2 both approaches diverge when not using a low pass filter in iteration, with Mann iteration faring only slightly worse than Picard iteration for the input, but far better for the output. When using a low pass filter with cut frequency at 100Hz in Case 2, Picard iteration still diverges with virtually no improvement from the result obtained without the filter (see Fig. 2.4). Mann iteration however

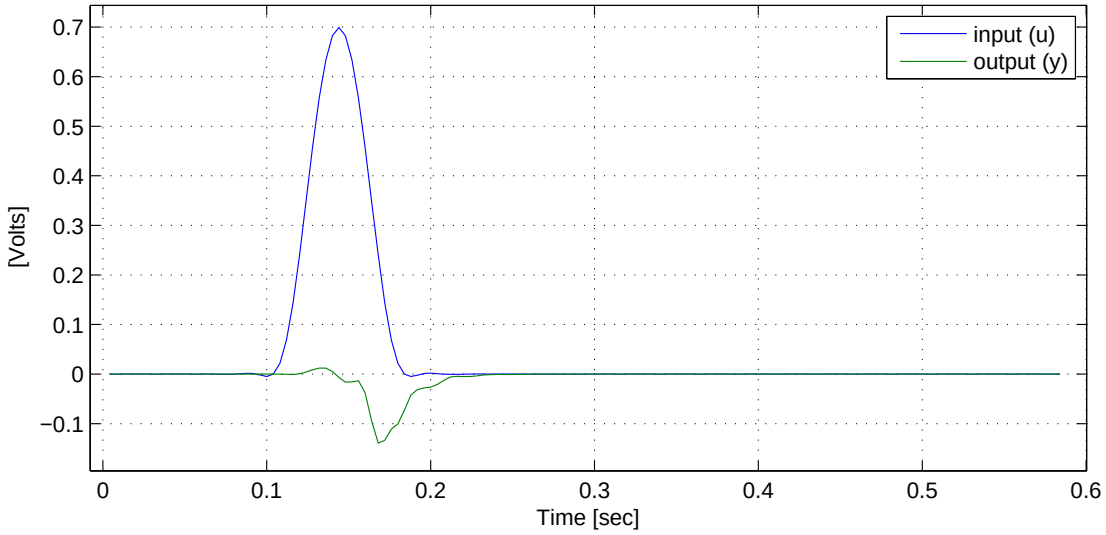

Figure 2.1: Ex. 1, Case 1: $u_d(t)$ and $y_d(t)$.

experiences an oscillatory convergence with best results that are nearly identical to the desired input and output (see Fig. 2.5).

Table 2.1: Example 1: Results of the stable inversion using Picard and Mann iteration. M is the iteration resulting in $\min_m \text{err}_1(u^{(m)})$, i.e. $M = \text{argmin}_m \text{err}_1(u^{(m)})$.

Case	Iteration type	α_m	$\min_m \text{err}_1(u^{(m)})$ [%]	$\text{err}_1(y^{(M)})$ [%]	M = Iter. No.	Comment
1	Picard	1	0.0	0.0	34	–
1	Mann	0.05	0.0	0.0	33	–
2	Picard	1	23.9	523	1	–
2	Mann	0.2	29.3	27.7	3	–
2	Picard	1.0	23.6	503	31	100Hz L.P. Filter
2	Mann	0.1	0.9	1.0	391	100Hz L.P. Filter

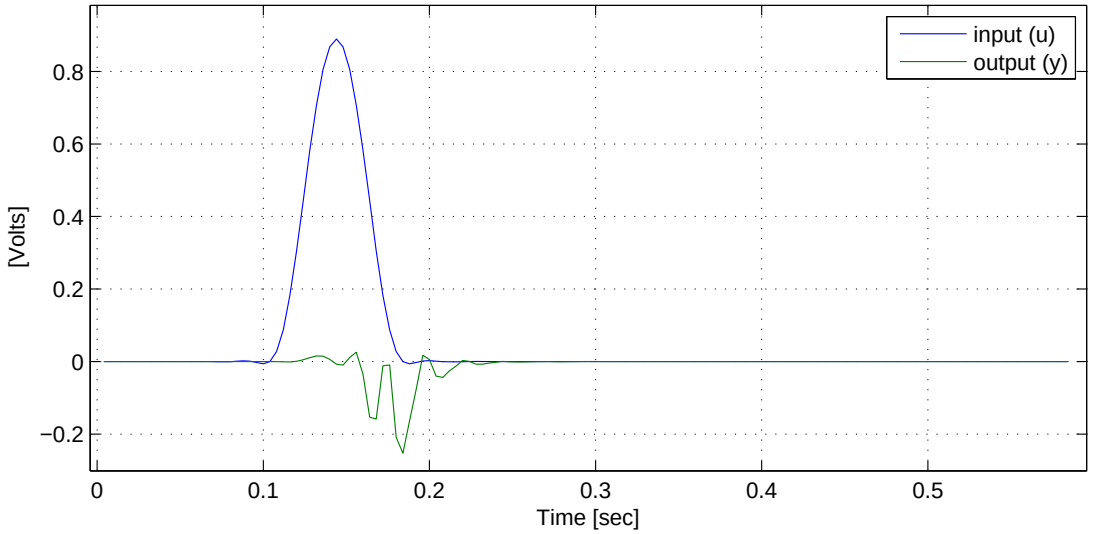

Figure 2.2: Ex. 1, Case 2: $u_d(t)$ and $y_d(t)$.

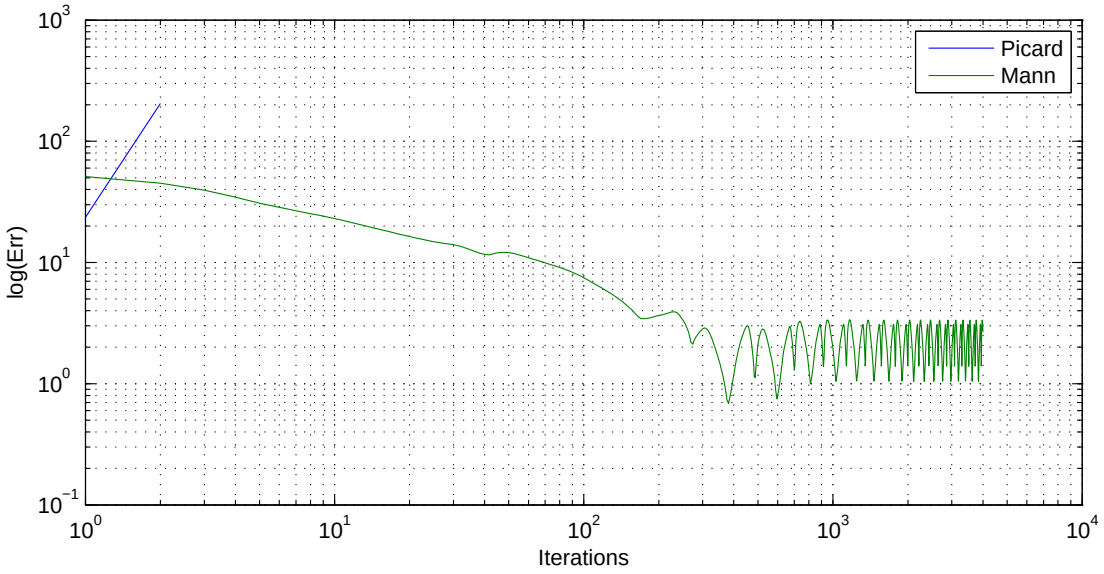

Figure 2.3: Ex. 1, Case 2: Iteration error for calculated input signal for Picard and Mann iteration.

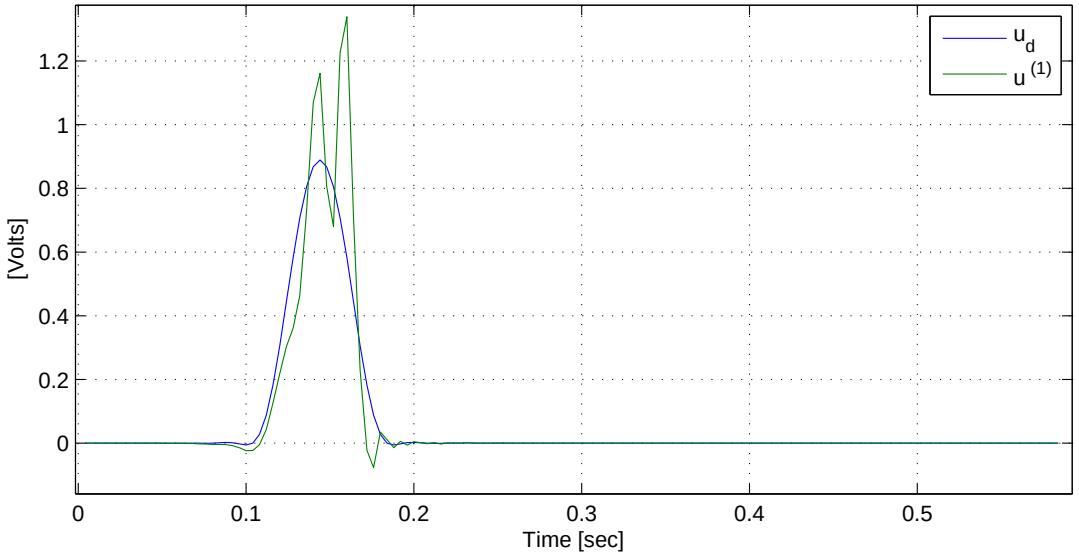

Figure 2.4: Ex. 1, Case 2: Best $u^{(i)}(t)$ for Picard iteration.

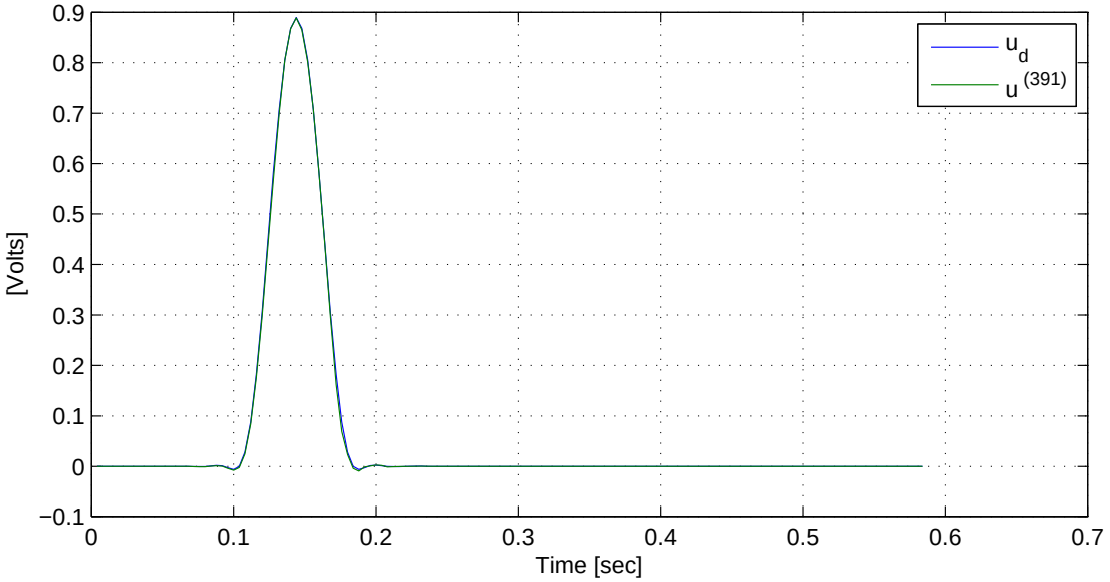

Figure 2.5: Ex. 1, Case 2: Best $u^{(i)}(t)$ for Mann iteration.

2.7 Example 2: Mann Iteration and Low Pass Filtering - Random Signal

In this example the advantages of using Mann iteration and low pass filtering is demonstrated for a random signal. While the better accuracy of Mann iteration is demonstrated, it is also shown that to achieve convergence, and therefore exact tracking, requires the use of both Mann iteration and a low pass filter in this case.

Consider the following NARX system:

$$\begin{aligned} y(k) &= \theta_1 u(k-4) + \theta_2 u(k-5) + \theta_3 u(k-6) + \theta_4 y(k-4) \\ &\quad \theta_5 u(k-5) y(k-4) + \theta_6 u(k-5) u(k-6) y(k-2) \\ &\quad + \theta_7 u(k-5)^2 u(k-6) y(k-1) \ , \end{aligned} \tag{2.111}$$

with

$$(\theta_1, \ldots, \theta_7) = (0.150,\ 0.50,\ 0.50,\ 1/6,\ -2.0,\ 6.0,\ 11.0) \ . \tag{2.112}$$

A sample frequency of 250 Hz is assumed. A desired input trajectory, $u_d(k)$, is constructed as a random signal with a bandwidth of at most about 50 Hz, and a corresponding desired output trajectory $y_d(k)$ is generated by applying $u_d(k)$ to the system (Eq. 2.111). We focus on the stable inversion of this system for the given $y_d(k)$. When converted to the nonlinear state space formulation, then to the normal form, and finally inverted, this system gives rise to the following inverse system:

$$\begin{aligned} \eta_1(k+1) &= C_1(k)\ \eta_1(k) + C_2\ \eta_2(k) + C_3(k)\ \eta_1(k)\ \eta_2(k) + C_4(k)\ \eta_1(k)^2\ \eta_2(k) + C_0(k) \\ \eta_2(k+1) &= \eta_1(k) \\ u(k-5) &= \eta_1(k) \end{aligned} \tag{2.113}$$

$k = 1, \ldots, N$, with

$$\begin{aligned} C_0(k) &= \frac{1}{\theta_1} y(k) - \frac{\theta_4}{\theta_1} y(k-4) \\ C_1(k) &= -\frac{\theta_2}{\theta_1} - \frac{\theta_5}{\theta_1} y(k-4) \\ C_2 &= -\frac{\theta_3}{\theta_1} \\ C_3(k) &= -\frac{\theta_6}{\theta_1} y(k-2) \\ C_4(k) &= -\frac{\theta_7}{\theta_1} y(k-1) \ . \end{aligned}$$

The percentage error of the calculated inputs with respect to $u_d(k)$ is again calculated as in Example 1 (as well as for the calculated outputs). Stable inversion is now performed for model and desired output

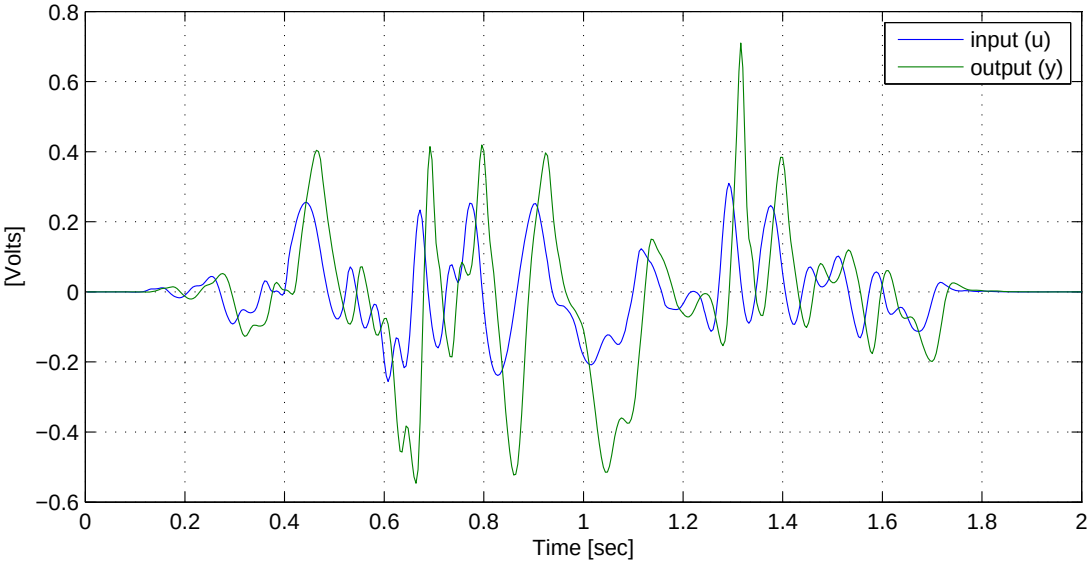

Figure 2.6: Exp. 2: $u_d(t)$ and $y_d(t)$.

as is, and thereafter with a low pass filter incorporated in the iteration. The results of stable inversion using Picard and Mann iteration for the two situations (with and without the filter) is presented in Table 2.2.

Table 2.2: Results of stable inversion. M is the iteration resulting in $\min_m \text{err}_1(u^{(m)})$, i.e. $M = \text{argmin}_m \text{err}_1(u^{(m)})$.

Case	Iteration type	α_n	$\min_m \text{err}_1(u^{(m)})$ [%]	$\text{err}_1(y^{(M)})$ [%]	M = Iter. no.	Comment
1	Picard	1.0	36.5	200.9	1	No filter
1	Mann	0.5	9.4	6.1	4	No filter
2	Picard	1.0	36.1	204.8	1	50Hz L.P.F.
2	Mann	0.5	0.7	0.4	100+	50Hz L.P.F.

For stable inversion without the low pass filter both iteration schemes are divergent, but with Mann iteration much more accurate than Picard iteration. The iteration error for the calculated input is shown in Fig. 2.7 for the iteration schemes, and the best calculated inputs in Fig. 2.8 and Fig. 2.9 for Picard and Mann iteration respectively.

For stable inversion with a low pass filter with a cut frequency of 50 Hz Picard iteration is still divergent, but Mann iteration is now convergent. The iteration error for the calculated input is shown in Fig. 2.10 for the iteration schemes, and the best calculated inputs in Fig. 2.11 and Fig. 2.12 for Picard and Mann iteration respectively.

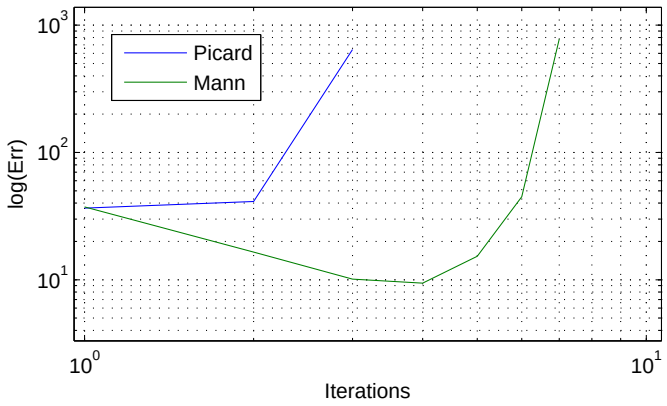

Figure 2.7: Exp. 2 (no low pass filter): Iteration error of calculated inputs for Picard and Mann iteration.

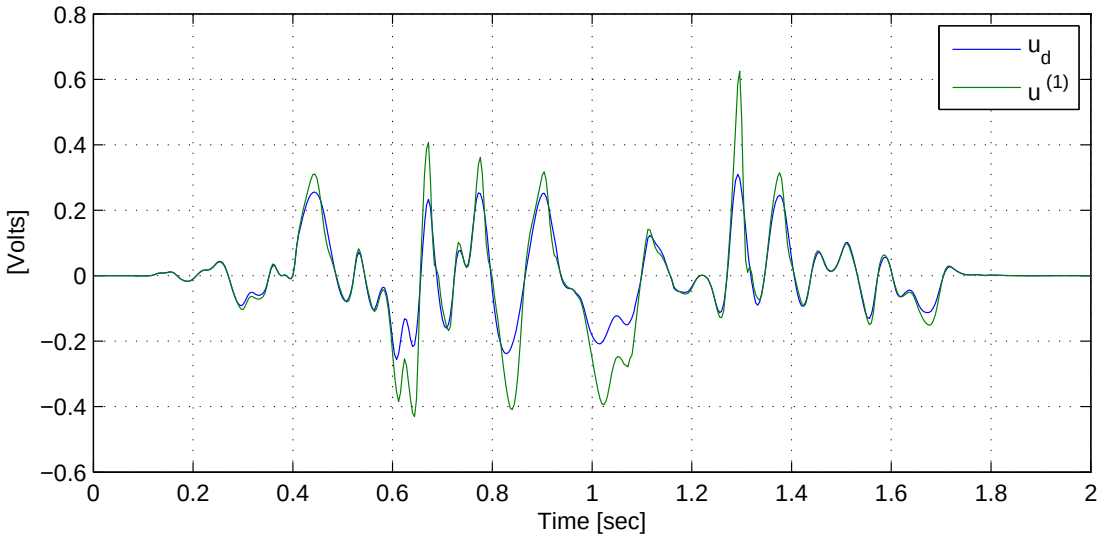

Figure 2.8: Exp. 2 (no low pass filter): Best $u^{(i)}$ for Picard iteration.

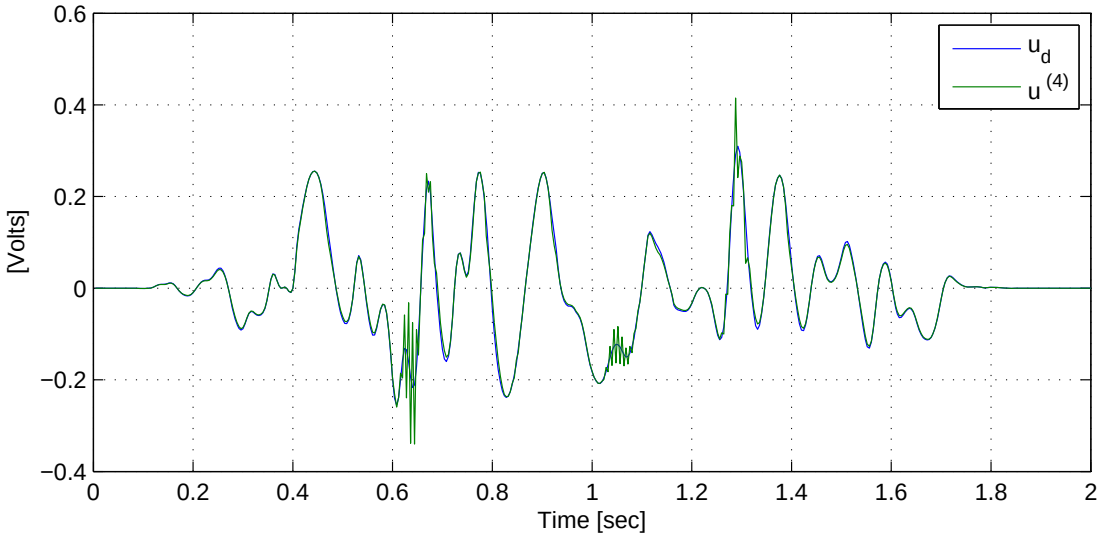

Figure 2.9: Exp. 2 (no low pass filter): Best $u^{(i)}$ for Mann iteration.

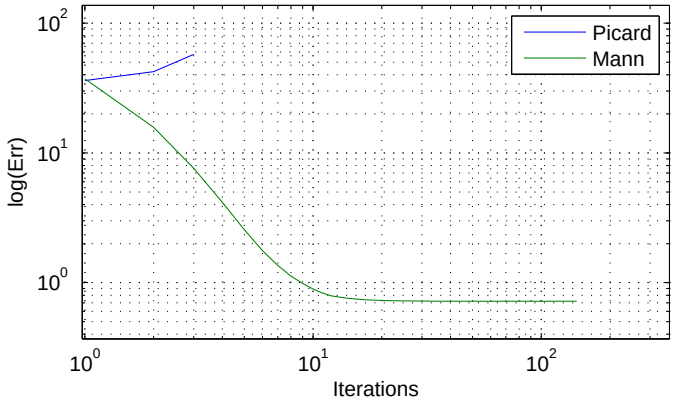

Figure 2.10: Exp. 2 (50Hz low pass filter): Iteration error of calculated inputs for Picard and Mann iteration.

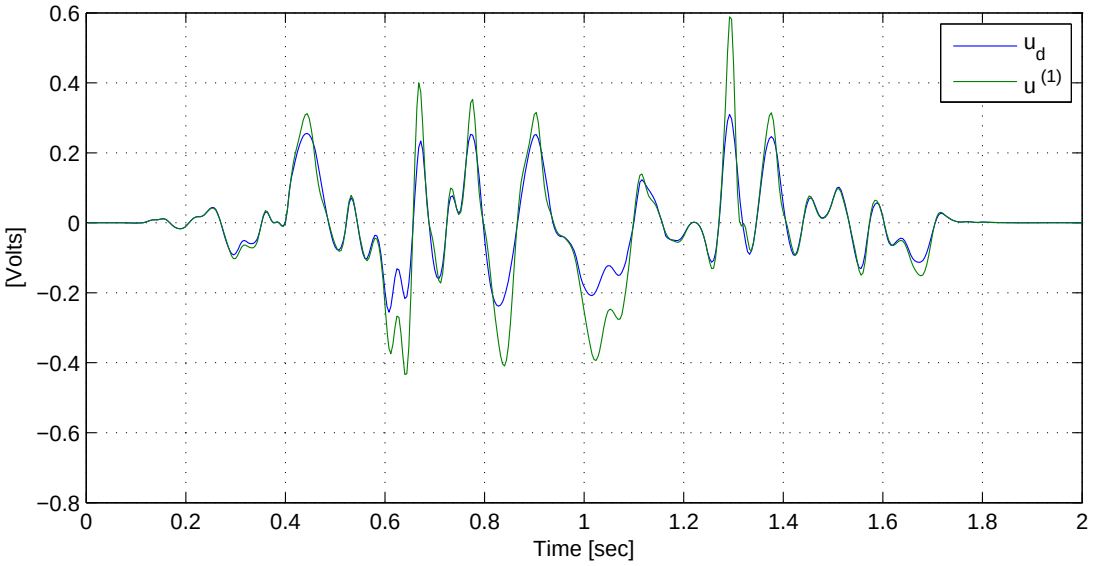

Figure 2.11: Exp. 2 (50Hz low pass filter): Best $u^{(i)}$ for Picard iteration.

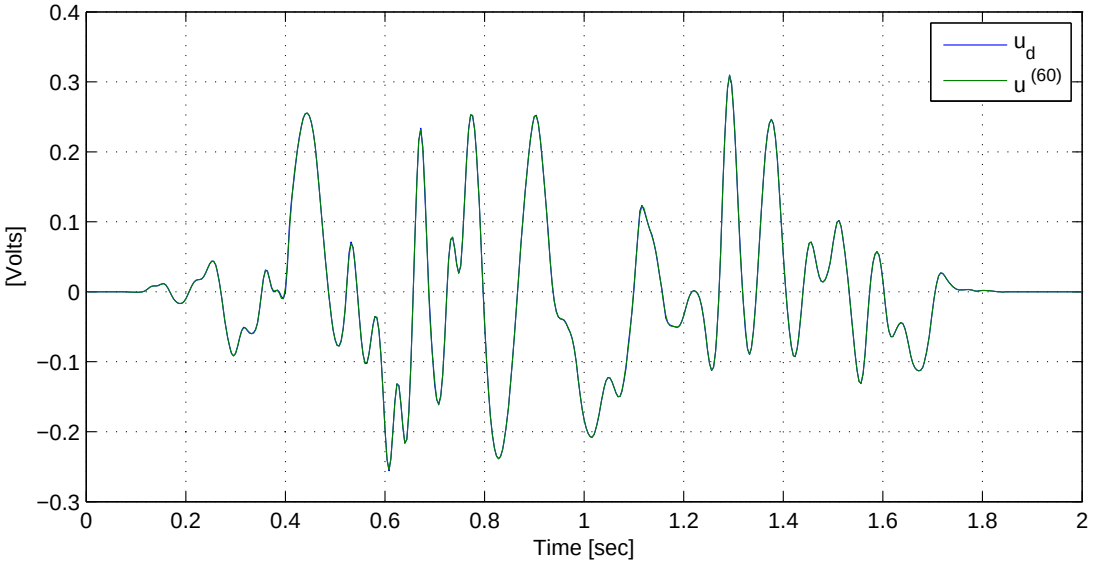

Figure 2.12: Exp. 2 (50Hz low pass filter): Best $u^{(i)}$ for Mann iteration.

2.8 Example 3: Evaluation of Various Mann Iteration Gain Strategies

In this example the advantages of using Mann iteration and low pass filtering is demonstrated for a random signal, however, whereas in examples 1 and 2 a constant gain was used in Mann iteration, in this example various strategies for the gain in Mann iteration are evaluated, including iteration-dependent gains, and time-dependent gains. The iteration-dependent gain strategy involves a monotonically decreasing sequence of gain values, while the time-dependent gain strategy is essentially an amplitude dependent formula (intended to interrupt the process of local divergence, which is a frequent occurrence). It is found that the most accurate results are achieved when employing a combined iteration-dependent and time-dependent gain.

In Example 3 we focus on the stable inversion of the same system as in Example 2, namely Eq. 2.111, however with a larger-amplitude $u_d(k)$ and corresponding $y_d(k)$ (Fig. 2.13). In view of the polynomial nature of the system this generally implies the system is now operated in a more nonlinear regime. Once again, the percentage error between signals is calculated as in Example 1. In a subsequent chapter on ILC we will attempt to invert this system for this $y_d(k)$ by means of ILC. (Recall that ILC iteratively employs stable inversion.) We will compare the success of inversion of this system via ILC with that of the "single pass" stable inversion in this example, the latter serving as a base line for the comparison. Three approaches to handling the gain in Mann iteration will be tried, namely constant gains, monotonically decreasing gain sequences, and time-varying gains (either constant or decreasing):

1. *Preliminary trials using constant gains:* Noting that if $\alpha_m = 1$ we recover Picard iteration from Mann iteration and, in turn, if $\beta_m = 0$ we recover Mann iteration from Ishikawa iteration, it follows that Ishikawa iteration for all combinations of constant $\alpha_m = \alpha$ and $\beta_m = \beta$ (each varying over regular intervals) includes both Mann iteration (also with constant α_m) and Picard iteration. A mapping for the minimum iteration error for the input signal achieved with Ishikawa iteration for different combinations of α and β is shown in Fig. 2.14. A low pass filter of 50 Hz was used. The curve $\beta = 0$ thus represents Mann iteration, and the point $\alpha = 1$ and $\beta = 0$ represents Picard iteration. Clearly Ishikawa iteration (evaluated using a constant α and β) does not significantly improve on the best results that may be obtained with Mann iteration (evaluated at the same α), while Mann iteration does represent a significant improvement on Picard iteration. For the remainder of this study we will therefore focus on Mann iteration only during stable inversion. Noting in Fig. 2.14 that the accuracy of stable inversion is greater for smaller values of α (and β), in the remainder of the example we therefore focus on smaller values of α in particular. We will also employ a zero phase low pass filter with a suitable cut frequency in stable inversion and set $\eta_i^{(0)}(k) = 0$ for all k and i.

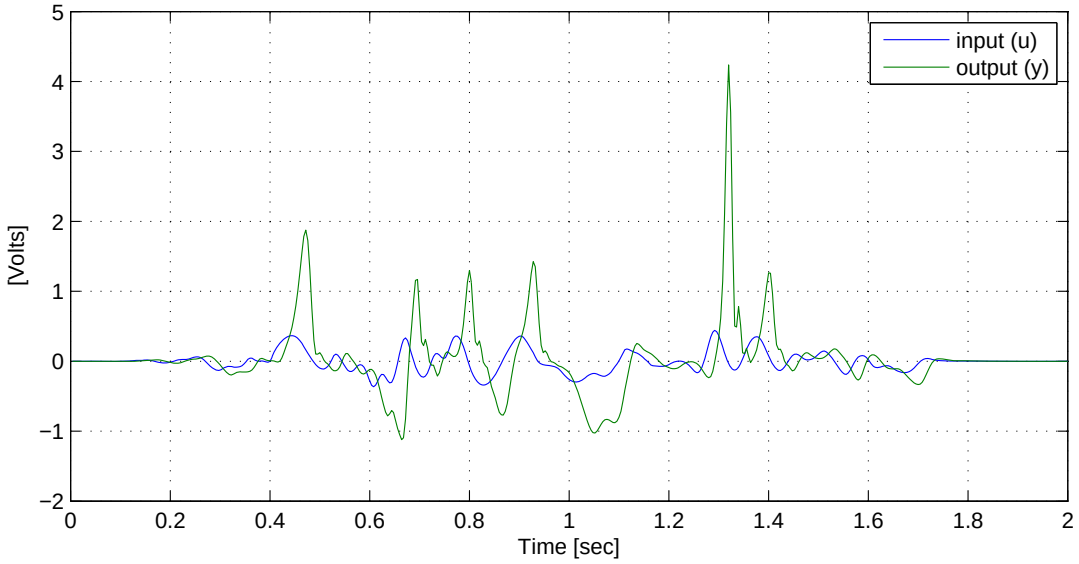

Figure 2.13: Exp. 3: $u_d(t)$ and $y_d(t)$.

2. *Systematic trials using constant gains:* Firstly we evaluate the stable inversion results that are obtained using constant gains over a range of values. For this we use a 50 Hz zero phase low pass filter, a y-based iteration selection strategy (as before), and limit the number of iterations to at most 10000. The whole exercise is repeated for a 70 Hz and 90 Hz filter, and for no filter. The best results are summarized in Table 2.3 (constant gain, time-independent case). The most accurate calculated u signal has an error of 27.6%, and the corresponding output signal an error of 26.0%.

3. *Using monotonically decreasing gain sequences:* Next we evaluate the stable inversion results that are obtained using monotonically decreasing gains (α_n). Chidume and Osilike [1999] suggest the following prototype formula for α_n, $n \geq 0$:

$$\alpha_n = \frac{1}{1+n} \ ,$$

which we generalize here as follows:

$$\alpha_n = n_0 \frac{\alpha_0 - \alpha_{\lim}}{(n+1)^\nu + (n_0 - 1)} + \alpha_{\lim} \ , \tag{2.114}$$

$n \geq 0$ and α_0 the initial value of α_n (i.e. at $n = 0$). Using a 50 Hz zero phase low pass filter a rough optimization exercise of the stable inversion results for $\alpha_{\lim}$, n_0 and ν indicates best results are obtained for $\alpha_{\lim}$ between 0.0001 and 0.0005, $n_0 = 2$ and $\nu = 1$. In the sequel therefore we will use

$$\alpha_n = 2\frac{\alpha_0 - \alpha_{\lim}}{n+2} + \alpha_{\lim} \ , \tag{2.115}$$

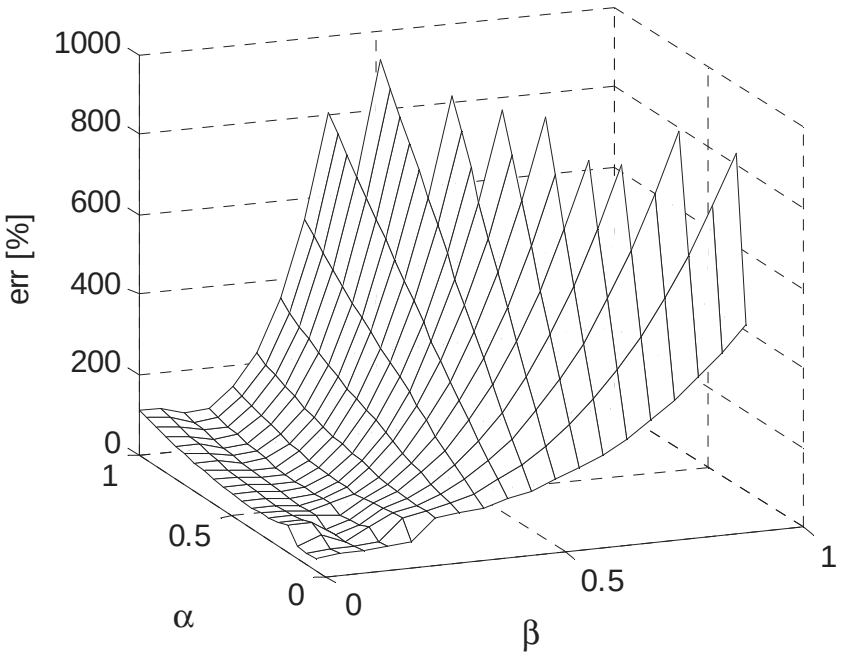

Figure 2.14: Minimum iteration error for the input using a constant α and β, for Ishikawa iteration.

$n \geq 0$. Using a 50 Hz zero phase low pass filter, a y-based iteration selection strategy, at most 10000 iterations, and various combinations of α_0 and α_{lim}, with

$$\alpha_0 \in \{-1.0, -0.7, -0.5, -0.3, -0.1, -0.05\}$$

and

$$\alpha_{\text{lim}} \in \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$$

gives best stable inversion results as in Table 2.3 (decreasing gain, time-independent case). Note that the most accurate calculated u signal is 16.8%, and when simulated back through the system gives an output error of 12.4%. Clearly the use of a decreasing gain sequence gives more accurate stable inversion results, but at the cost of larger numbers of required iterations, and therefore takes longer. Results obtained using a 70 Hz, 90 Hz and absent zero phase low pass filter are also shown in the table.

4. *Time-varying gains:* The above procedure may be repeated using a time-varying gain with the aim of suppressing the localised radical signal growth over successive iterations that often result in divergence of stable inversion. Various approaches exist for designing such a gain function that is dependent on the local magnitude of a signal for a given iteration as a function of time and on the nominal value of α_n. These approaches tend to be highly empirical in nature and are therefore not further elaborated on here. The formula that was used in this case is discussed in Example 4, Case 4. The best results obtained with this formula are presented in Table 2.3 (for both a constant nominal α_n and decreasing nominal α_n, the latter once again obtained for

various combinations of filter frequency, α_0 and $\alpha_{\lim}$). The best results obtained with iteration-independent, time-varying gains represent an improvement over the results of Case 2 and Case 3. The best results obtained with decreasing, time-varying gains represent the best stable inversion results thus far for this system and the given desired output signal. These results demonstrate the advantage of allowing α_n to be time-varying, for both both iteration-independent and iteration-dependent nominal values of α_n.

Table 2.3: Best results of stable inversion of Eq. 2.111. M is the iteration resulting in $\min_m \text{err}_1(u^{(m)})$, i.e. $M = \text{argmin}_m \text{err}_1(u^{(m)})$.

Iteration type α_n	Filter cut freq. [Hz]	$\min_m \text{err}_1(u^{(m)})$ [%]	$\text{err}_1(u^{(M)})$ [%]	M = Iter. no.	Comment
Iteration-independent	50	27.6	26.0	13	$\alpha_n = 0.1$
Time-independent	70	33.4	31.7	913	$\alpha_n = 0.001$
	90	37.7	33.4	808	$\alpha_n = 0.001$
	(None)	66.0	55.0	492	$\alpha_n = 0.001$
Iteration-dependent	50	16.8	12.4	10000	$\alpha_0 = 0.05$, $\alpha_{\lim} = 0.0001$
Time-independent	70	29.7	25.4	3697	$\alpha_0 = 0.05$, $\alpha_{\lim} = 0.0001$
	90	35.6	27.9	3160	$\alpha_0 = 0.05$, $\alpha_{\lim} = 0.0001$
	(None)	65.7	51.7	234	$\alpha_0 = 0.05$, $\alpha_{\lim} = 0.0001$
Iteration-independent	50	15.7	10.3	3776	$\alpha_n = 0.001$
Time-varying	70	19.5	14.5	12	$\alpha_n = 0.5$
	90	26.2	16.1	423	$\alpha_n = 0.01$
	(None)	46.4	46.6	3	$\alpha_n = 0.3$
Iteration-dependent	50	13.3	8.1	8327	$\alpha_0 = 0.05$, $\alpha_{\lim} = 0.0005$
Time-varying	70	18.8	15.1	85	$\alpha_0 = 0.5$, $\alpha_{\lim} = 0.005$
	90	28.4	13.5	3244	$\alpha_0 = 0.3$, $\alpha_{\lim} = 0.0001$
	(None)	48.6	38.6	18	$\alpha_0 = 0.3$, $\alpha_{\lim} = 0.0005$

Chapter 3

Iterative Learning Control of Nonlinear Systems

3.1 Introduction

Firstly a brief overview of relevant aspects of the theory of ILC of discrete-time linear systems is presented. Here the focus will be on model-based or inverse model-based ILC compensators L for the general deterministic square, MIMO case, which is applicable to many laboratory situations. Next, in Section 3.3, the conventional ILC algorithm is presented for model-based or inverse model-based ILC on *nonlinear* systems, again for the deterministic, square, MIMO case. A *general inverse based* (GIB) ILC compensator, which explicitly uses the approximate system inverse model, is proposed, and represents a slight modification of existing approaches for ILC in robotics literature in that it includes a scalar scale factor for controlling the rate of convergence. A related contribution of this research is the demonstration in Section 3.3.5 of how ILC with the GIB compensator employing the scale factor represents a Mann iteration version of ILC without the scale factor (that corresponds to Picard iteration).

As a novel contribution of this research an alternative ILC algorithm is presented in Section 3.4, as well as a modified version in Section 3.4 that enables it to fully match the various properties of the conventional algorithm. Theorems 3.4 to 3.9 are modifications for the alternative and modified alternative ILC algorithms of theorems in the literature of the conventional ILC algorithm.

An example is presented of ILC on a nonlinear system where in a series of case studies, the ability of ILC to achieve exact tracking (i.e. converge) is demonstrated, the ability of the alternative algorithm to sometimes converge when the conventional algorithm diverges is demonstrated, and the conventional and alternative algorithms are compared in a case where both diverge.

3.2 ILC of Linear Systems

3.2.1 Test System Formulation

Consider a $m \times m$ discrete-time, linear time-invariant (LTI) system represented in input-output format by a linear operator T, with input signal u and output signal y

$$y = Tu \; . \tag{3.1}$$

Here we assume that there exists a desired plant output signal $y_d(k)$, $k = 0, \ldots, N-1$, in the range of T which we desire to track as closely as possible, $y_d(0) = y_d(N) = [0]$, T is injective, i.e., there exists a unique $u_d(k)$ such that (Silverman, 1969)

$$y_d = Tu_d \; . \tag{3.2}$$

3.2.2 ILC Algorithm

In order to achieve tracking that is as accurate as possible, we employ an iterative algorithm called *iterative learning control* (ILC). ILC uses a learning capability to improve the tracking of the desired response y_d over repeated trials with the real-time system. For iteration i of the ILC procedure the testing of the input signal, u_i, gives the corresponding output y_i as

$$y^{(i)} = Tu^{(i)} \; , \tag{3.3}$$

with the achieved tracking error for iteration i given by

$$e^{(i)} = y_d - y^{(i)} \; . \tag{3.4}$$

The input signal of the next trial (iteration $i+1$) is then calculated by updating the input signal of the current trial, u_i, as follows

$$u^{(i+1)} = Q(u^{(i)} + Le^{(i)}) \; . \tag{3.5}$$

$u^{(i+1)}$ is intended to result in an improvement in the tracking accuracy, i.e. a reduction in the tracking error as given by $e^{(i+1)}$. L is a linear operator called here the *ILC compensator*, and Q is a linear operator representing a zero-phase filter based on the causal discrete time filter Q_0 (i.e. in the frequency domain Q is represented by $|Q_0(e^{j\omega})|^2$). Eq. 3.5 is a fundamental form of the ILC update formula, and is an example of a *first-order* update formula. It is sometimes referred to as the *classical* update formula. The formula is implemented off-line between trials, i.e. the calculations are not done in real time. This also allows the calculations to be done either non-causally or in the frequency domain. Specifying the ILC compensator L is the aim of the design problem of ILC.

3.2.3 Convergence of ILC

For a proper design of the ILC compensator sufficient conditions may be derived for guaranteeing exact tracking of the desired output y_d, if not on the entire frequency range then at least over a limited frequency range. To this end we construct the *system formulation* of ILC in the iteration domain (called the *linear iterative system formulation* in Norrlöf (2002a) by combining Eq. 3.3, Eq. 3.4, and Eq. 3.5 as follows

$$\begin{aligned} u^{(i+1)} &= Q(u^{(i)} + Le^{(i)}) \\ &= Q(u^{(i)} + L(y_d - y^{(i)})) \\ &= Q(u^{(i)} + L(y_d - Tu^{(i)})) \\ &= Q((I - LT)u^{(i)} + Ly_d) \ . \end{aligned} \tag{3.6}$$

A necessary and sufficient condition for convergence is that

$$|\lambda_l(Q((I - LT))| < 1 \forall l \tag{3.7}$$

for Eq. 3.6 formulated in matrix form containing all time steps in iteration i with $\lambda_l(F)$ the l-th eigenvalue of F (Longman, 2000)). In the frequency domain the system formulation in the iteration domain may be expressed as

$$u^{(i+1)}(\omega) = |Q_0(e^{j\omega})|^2((I - L(e^{j\omega})T(e^{j\omega}))u^{(i)}(\omega) + L(e^{j\omega})y_d(\omega)) \ , \tag{3.8}$$

which is bounded (for bounded u) if (Norrlöf, 2002)

$$\sup_{\omega \in [0,2\pi]} \rho \left(|Q_0(e^{j\omega})|^2(I - L(e^{j\omega})T(e^{j\omega}))\right) < 1 \ , \tag{3.9}$$

with $\rho(F(e^{j\omega})) = \max_l |\lambda_l(F(e^{j\omega}))|$, and $\lambda_l(F(e^{j\omega}))$ the l-th eigenvalue of $F(e^{j\omega})$. While this condition is defined for steady state conditions only, it can be shown to ensure convergence for the transient region of every trial also, and can be regarded as a sufficient condition for convergence (Longman, 2000).

3.2.4 Limit Signals of ILC

If Eq. 3.9 is satisfied, then there exists a bounded limit signal $u^{(\infty)}$, which is the fixed point of Eq. 3.6. Noting that

$$u^{(\infty)} = \lim_{i \to \infty} u^{(i)} = \lim_{i \to \infty} u^{(i+1)} \ , \tag{3.10}$$

by taking the limit on both sides in Eq. 3.6 we obtain (using Eq. 3.30)

$$u^{(\infty)} = Q((I - LT)u^{(\infty)} + Ly_d) \tag{3.11}$$

$$= (I - Q(I - LT))^{-1}QLTu_d \ . \tag{3.12}$$

Clearly, if $Q = 1$ then $u^{(\infty)} = u_d$, from which $y^{(\infty)} = y_d$, implying that exact tracking is achieved, with the fixed point independent of L. When $Q \neq 1$ exact tracking is thus not obtained over all frequencies up to the Nyquist frequency. In the steady state however, if Q is a low pass filter then exact tracking will be achieved at frequencies sufficiently below the filter's cut frequency, again independent of L (which can be confirmed by converting Eq. 3.12 to the discrete frequency domain.)

3.2.5 Transient Behavior

A frequent observation in the ILC literature (Longman, 2000) is that while the convergence of ILC is relatively easily ensured by proper choice of L, often using even very simple forms of L, the convergence error of the input,

$$\Delta^{(i)} := u^{(\infty)} - u^{(i)} \ , \tag{3.13}$$

does not necessarily decay monotonously and, in fact, sometimes may grow strongly before finally decaying. This is obviously not a desirable situation in physical test systems and it is therefore common to seek a condition for monotone convergence.

By inserting Eq. 3.11 and Eq. 3.6 in Eq. 3.13 an iteration-domain formulation may be derived for the input convergence error as

$$\begin{aligned} \Delta^{(i+1)} &= u^{(\infty)} - u^{(i+1)} \\ &= Q(1 - LT)(u^{(\infty)} - u^{(i)}) \\ &= Q(1 - LT)\Delta^{(i)} \ . \end{aligned} \tag{3.14}$$

A sufficient condition for monotone convergence in the Euclidean (2-) norm is then that

$$\sigma_{\max}(Q(I - LT)) < 1 \tag{3.15}$$

for Eq. 3.6 formulated in matrix form containing all time steps in iteration i, and $\sigma_{\max}$ the largest singular value. Alternatively, a sufficient condition in the frequency domain for monotone convergence is (Longman, 2002)

$$\sigma_{\max}(|Q_0(e^{j\omega})|^2(I - L(e^{j\omega})T(e^{j\omega}))) < 1 \ . \tag{3.16}$$

Note that by using similar reasoning as in the derivation of Eq. 3.6 a system formulation in the iteration domain may also be derived for the output signal as

$$y^{(i+1)} = Q((1 - TL)y^{(i)} + TLy_d) \ . \tag{3.17}$$

If we define the output convergence error as

$$\Delta_y^{(i)} := y^{(\infty)} - y^{(i)} \ , \tag{3.18}$$

then the system formulation for the output convergence error is given by

$$\Delta_y^{(i+1)} = Q(1 - TL)\Delta_y^{(i)} \ . \tag{3.19}$$

3.2.6 Model-Based ILC Compensators

Achieving monotone convergence generally requires the use of model-based ILC compensators, with a number of types in existence that are aimed at this need, including the following designs ($\hat{T}$ below represents an approximate model of the real system of T, with $\hat{T}(e^{j\omega}) = |\hat{T}(e^{j\omega})| \arg \hat{T}(e^{j\omega})$):

- **Contraction Mapping ILC Compensator,** given in the frequency domain in SISO form as (Jang and Longman (1994, 1996a)):

$$L(e^{j\omega}) = c|\hat{T}(e^{j\omega})|(-\arg \hat{T}(e^{j\omega})) \ , \tag{3.20}$$

  with c a real scalar. From Eq. 3.20 and Eq. 3.16 it is clear that monotone convergence is ensured for sufficiently small values of c. The references present a time domain version of the compensator for implementation in the time domain, especially for short duration tests (less than about four times the slowest system time constant). Eq. 3.14 in this case becomes in the frequency domain (for $|Q_0(e^{j\omega})| = 1$):

$$\Delta^{(i+1)}(\omega) = (1 - c|\hat{T}(e^{j\omega})|^2)\Delta^{(i)}(\omega) \ . \tag{3.21}$$

  Since physical systems usually have attenuation at high enough frequencies, this ILC compensator can have rather slow convergence at the high frequency end due to the square in Eq. 3.21. This problem is alleviated with the following ILC compensator.

- **Phase-Cancellation ILC Compensator** (Jang and Longman, 1996b) given in the frequency domain in SISO form as

$$L(e^{j\omega}) = c(-\arg \hat{T}(e^{j\omega})) \ . \tag{3.22}$$

  Eq. 3.14 for this compensator becomes in the frequency domain (for $|Q_0(e^{j\omega})| = 1$):

$$\Delta^{(i+1)}(\omega) = (1 - c|T(e^{j\omega})|)\Delta^{(i)}(\omega) \ . \tag{3.23}$$

  Notice that now the learning does not involve the squaring of the system gain anymore. A time domain version has been developed in (Jang and Longman, 1996b).

- **The Modified Phase-Cancellation ILC Compensator,** (Elci *et al.* 1994b) given in the frequency domain in SISO form as

$$L(e^{j\omega}) = \begin{cases} c(-\arg \hat{T}(e^{j\omega})) \ , & |\hat{T}(e^{j\omega})| \leq 1 \\ (c/|\hat{T}(e^{j\omega})|)(-\arg \hat{T}(e^{j\omega})) \ , & |\hat{T}(e^{j\omega})| > 1 \ . \end{cases} \tag{3.24}$$

  The modified phase cancellation compensator is effectively the inverse system. However, at attenuation frequencies (usually high frequencies) where the inverse model tends to have a very high gain, this compensator is a phase cancellation compensator with unit gain. Compared to the inverse compensator this is advantageous because the high gain of the inverse compensator (usually at high frequencies) is problematic for various reasons (that are shortly discussed).

- **The α Pseudoinverse-Based ILC Compensator,** defined in operator form for the SISO case as:

$$L = \frac{\hat{T}^*}{\alpha + \hat{T}^*\hat{T}} \ , \tag{3.25}$$

with $\hat{T}^*$ the adjoint operator of $\hat{T}$. In the frequency domain this compensator is represented as

$$L(e^{j\omega}) = \frac{|\hat{T}(e^{j\omega})|}{\alpha + |\hat{T}(e^{j\omega})|^2}(-\arg \hat{T}(e^{j\omega})) \ . \tag{3.26}$$

Assume a relatively small value for α. For high-gain frequencies of $\hat{T}$ the compensator approximates the modified phase cancellation and inverse compensators, but at attenuation frequencies the value of α begins to dominate and the compensator has a lower gain than the inverse compensator. The compensator is presented in Ghosh and Paden (2004) for the continuous-time case.

- **Inverse ILC Compensators,** defined as (in operator form)

$$L = c\hat{T}^{-1} \ , \tag{3.27}$$

with scale factor $c \in \Re$, and in the frequency domain for the SISO case as

$$L(e^{j\omega}) = \frac{c}{|T(e^{j\omega})|}(-\arg \hat{T}(e^{j\omega})) \ . \tag{3.28}$$

See e.g. Phan and Longman (1989) and Lee-Glauser *et al.* (1996). This is also the form of the ILC compensator in the earliest implementations of automotive service load simulation for fatigue testing (Cryer, 1976). Early implementations employed frequency domain inverses. In case of non-minimum phase systems, when the time-domain inverses have unstable dynamics, the approach was to approximate the true inverse with an approximate causal inverse. This limitation can be removed by using the non-causal stable inversion method for non-minimum phase systems. This allows the most accurate inverse that can be identified for the test system to be used in ILC. The advantages of using an accurate inverse as ILC compensator are the following:

  - Widest possible frequency band of convergence;
  - Monotonic decay of the error of the input signal with respect to the limit input signal;
  - Rapid convergence, with the possibility of convergence in one iteration when using an exact inverse ILC compensator; and
  - Good control over the rate of the convergence of the input signal (this may be achieved using a scale factor in the inverse compensator).

In practice the model uncertainty usually increases with frequency due to noise and/or nonlinear effects manifesting at high frequencies in conjunction with the natural attenuation of typical systems at high frequencies, all of which reduces accuracy of identified models at high frequencies. The model uncertainty at high frequencies in combination with the high gain of the inverse compensator at high frequencies can limit the bandwidth over which convergence can be achieved. A second disadvantage of the inverse compensator is that it is limited to square systems, unlike the pseudo-inverse compensator. The α pseudoinverse compensator is in fact a generalization of the inverse compensator for non-square systems. In practice, however, square systems are common.

The problem of limited convergence bandwidth of the inverse-based compensator may be alleviated by opting for one of the compensators with reduced gain at high frequencies (phase cancellation or α-pseudoinverse compensators), or by incorporating a general zero-phase filter in the compensator to attenuate the high frequency response. Opting for this approach leads to a general, inverse-based compensator, given in operator form as

$$L = C\tilde{L} \ ,$$

with C a zero-phase filter and $\tilde{L}$ the approximate inverse of T. In the frequency domain for the SISO case this becomes

$$L(e^{j\omega}) = |C_0(e^{j\omega})|^2 |\tilde{L}(e^{j\omega})| \arg \tilde{L}(e^{j\omega}) \ .$$

A similar approach may be found in Gunnarsson (1997) and Norrlöf (2002). Note that for specific choices of C the compensator includes all the different choices of model-based compensators described above. By the proper design of C the compensator allows the usually high magnitude of $\tilde{L}$ at the high frequencies to be attenuated for the sake of robustness of convergence against model error. To illustrate, with this compensator the frequency domain version of the condition for monotone convergence, Eq. 3.16, yields

$$|||Q_0(e^{j\omega})|^2 (1 - |C_0(e^{j\omega})|^2 \tilde{L}(e^{j\omega}) T(e^{j\omega}))||_\infty < 1 \ .$$

At frequencies where the product of $\tilde{L}(e^{j\omega})T(e^{j\omega}))$ would normally violate the condition (due to model error in $\tilde{L}$), C may attenuate the product at the relevant frequencies in order to satisfy the inequality and thus retain convergence at that frequency.

3.3 Inverse Model Based ILC of Nonlinear Systems

3.3.1 Test Formulation

Consider a $m \times m$ discrete time, nonlinear, stable, injective test system, represented by a nonlinear operator T with $T([0]) = [0]$. The input signal for the i-th test trial is designated $u^{(i)}$ and the output

signal $y^{(i)}$, so that (with slight abuse of notation)

$$y^{(i)} = T(u^{(i)}) \ . \tag{3.29}$$

The system is relaxed at the beginning of every test trial. We assume there exists a desired plant output $y_d(k)$, $k = 0, \ldots, N-1$, $y_d(0) = y_d(N) = [0]$, which we designate y_d, and which we desire to track as closely as possible. By the injectiveness of the plant there exists a unique $u_d(k)$, $k = 0, \ldots, N-1$, designated u_d, such that (Zeng & Hunt, 2000)

$$y_d = T(u_d) \ . \tag{3.30}$$

3.3.2 Algorithm

The update formula in the conventional algorithm for ILC on both linear and nonlinear systems is frequently given as (Norrlöf & Gunnarsson (2002a) and Markusson (2002))

$$u^{(i+1)} = Q(u^{(i)} + L(e^{(i)})) \ , \tag{3.31}$$

with

$$e^{(i)} = y_d - y^{(i)} \ , \tag{3.32}$$

$y^{(0)} = 0$, and Q a zero-phase, discrete-time linear filter. While usually linear, in this study the ILC compensator L is a generally nonlinear, discrete-time transfer operator.

When T is nonlinear, convergence to the desired u_d can potentially be achieved with Eq. 3.31 even when L is the inverse of an approximate model of the system. However, the convergence generally can not be achieved in one step, irrespective of how accurate a representation of the true inverse of T the compensator L is. This can be rectified by slightly modifying Eq. 3.31 as follows (Markusson (2002) and Smolders *et al.* (2008)):

$$u^{(i+1)} = Q(u^{(i)} + L(y_d) - L(y^{(i)})) \ . \tag{3.33}$$

This algorithm is shown in Fig. 3.1. In case of a nonlinear T this update formula is theoretically capable of achieving convergence in one step, but that requires that $L = T^{-1}$. Note that for linear L Eq. 3.33 reduces to the more conventional form (Eq. 3.31). In the sequel we will use the modified update formula, Eq. 3.33, as the standard representation of the conventional approach to ILC with nonlinear test systems, together with Eq. 3.29 and Eq. 3.30. In view of the alternative algorithm proposed in the next section we will refer to it as the *conventional* algorithm.

3.3.3 Convergence

Next we focus on the convergence of the conventional algorithm. By inserting Eq. 3.29 into Eq. 3.33 the system formulation of Eq. 3.33 in the iteration domain is obtained as

$$u^{(i+1)} \quad = \quad Q(u^{(i)} + L(y_d) - L(T(u^{(i)}))) \tag{3.34}$$

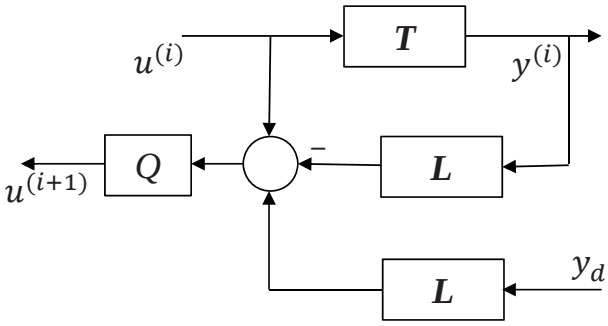

Figure 3.1: Conventional ILC algorithm on a nonlinear system T with general (possibly nonlinear) L and (optional) zero phase filter Q.

$$:= \quad \mathcal{T}(u^{(i)}, y_d) \; . \tag{3.35}$$

Eq. 3.34 is a fixed point problem in the variable u and we may follow the standard route of using the contraction mapping theorem in terms of the nonlinear operator $\mathcal{T}$ to obtain a strong condition for both the existence and uniqueness of a solution. Markusson (2002) provides a more relaxed condition for convergence, at the expense of having to prove the iteration is bounded and the existence of a well-defined limit signal separately. Here we follow the approach of Markusson (2002), starting with the boundedness of the ILC algorithm.

Theorem 3.1: If T is bounded input-bounded output (BIBO) stable in the 2-norm and $\alpha\beta < 1$ with $\alpha = ||Q||_\infty$ and

$$\beta = \sup_{u \neq 0} \frac{||u - L(T(u))||_2}{||u||_2} \; , \tag{3.36}$$

then Eq. 3.33 is bounded input-bounded output (BIBO) stable in the 2-norm. For the proof, see Markusson (2002). $\diamondsuit$

Noting that

$$u^{(\infty)} = \lim_{i \to \infty} u^{(i)} = \lim_{i \to \infty} u^{(i+1)} \; , \tag{3.37}$$

the limit signals for Eq. 3.33 and Eq. 3.34 are defined by

$$u^{(\infty)} \quad = \quad Q(u^{(\infty)} + L(y_d) - L(y^{(\infty)})) \tag{3.38}$$

$$\quad = \quad Q(u^{(\infty)} + L(y_d) - L(T(u^{(\infty)}))) \tag{3.39}$$

If the limit signals exist, then for $Q = 1$ clearly $L(y_d) = L(y^{(\infty)})$, implying $y^{(\infty)} = y_d$ since L is injective, and in turn $u^{(\infty)} = T^{-1}(y^{(\infty)}) = T(y_d) = u_d$ by the injectiveness of T. If $Q \neq 1$, then (from Eq. 3.38)

$$u^{(\infty)} = Q(1 - Q)^{-1}(L(y_d) - L(y^{(\infty)})) \tag{3.40}$$

and $y^{(\infty)} = T(u^{(\infty)})$. We have the following theorem for existence of the fixed points:

Theorem 3.2: If the conditions of Theorem 3.1 hold and L is BIBO stable, then for $Q \neq 1$ Eq. 3.40 is BIBO stable in the 2-norm and $u^{(\infty)}$ and $y^{(\infty)}$ are well defined. If $Q = 1$ then $u^{(\infty)}$ and $y^{(\infty)}$ are well defined, with $y^{(\infty)} = y_d$ and $u^{(\infty)} = u_d$. The proof for the case $Q = 1$ follows from the preceding discussion. For the proof of the case $Q \neq 1$, see Markusson (2002). $\diamondsuit$

The iteration domain formulation for the convergence error of the input signal w.r.t. the limit signal is then (using Eq. 3.34 and Eq. 3.39)

$$\begin{aligned}
\Delta^{(i+1)} &= u^{(\infty)} - u^{(i+1)} && (3.41)\\
&= Q(u^{(\infty)} + L(y_d) - L(T(u^{(\infty)}))) - Q(u^{(i)} + L(y_d) - L(T(u^{(i)})))\\
&= Q(u^{(\infty)} - u^{(i)} - (L(T(u^{(\infty)})) - L(T(u^{(i)})))\\
&= Q\Delta^{(i)} - Q(L(T(u^{(\infty)})) - L(T(u^{(\infty)} - \Delta^{(i)}))) && (3.42)\\
&= Q\Delta^{(i)} - Q\Phi(u^{(\infty)}, \Delta^{(i)}) \,, && (3.43)
\end{aligned}$$

with

$$\Phi(u^{(\infty)}, \Delta^{(i)}) \;:=\; L(T(u^{(\infty)})) - L(T(u^{(\infty)} - \Delta^{(i)})) \,. \qquad (3.44)$$

We have the following sufficient condition for convergence:

Theorem 3.3: If the ILC update formula Eq. 3.33 is BIBO stable, the limit signal of Eq. 3.40 is well defined, and $\alpha\gamma < 1$ with $\alpha = ||Q||_\infty$ and

$$\gamma = \sup_{\Delta \neq 0} \frac{||\Delta - \Phi(u^{(\infty)}, \Delta)||_2}{||\Delta||_2} \,, \qquad (3.45)$$

then $||\Delta^{(i+1)}|| < \alpha\gamma||\Delta^{(i)}||$, that is, the input convergence error is monotonically decreasing and the input $u^{(i)}$ converges to the input limit signal. For the proof, see Markusson (2002). $\diamondsuit$

3.3.4 Choice of ILC Compensator

With regard to model-based designs of L, in case of square test systems L may be the (linear) inverse of the linear model that approximates the nonlinear system (Ghosh & Paden, 2001), or in case of non-square systems it may be the α pseudo-inverse (Ghosh & Paden, 2002) of the linear approximation. The inverse compensator we employ here is given in operator form as

$$L = c\,C\tilde{L} \,, \qquad (3.46)$$

with c a real scalar, C a linear, zero-phase filter, and $\tilde{L}$ the linear or nonlinear approximate inverse of the generally nonlinear system T. This compensator is referred to here as the *general inverse-based*

(GIB) compensator. In this study the focus is on using both linear and nonlinear inverse models in $\tilde{L}$. Such models may be obtained by system identification on behavioral data of the nonlinear system T, and inverting the resulting model. The resulting inverse is often solved in the frequency domain in case of linear models, but may also be solved in the time domain using stable inversion in case of both linear and nonlinear models to circumvent the instability of the inverse associated with nonminimum phase zeros of the normal model.

Eq. 3.31 used in conjunction with the GIB compensator (Eq. 3.46) is also the form of the ILC algorithm in response reconstruction for purposes of automotive fatigue testing. For implementations with a linear inverse in $\tilde{L}$ see Cryer *et al.* (1976), Weal *et al.* (1997), Raath (1993, 1997), DeCuyper *et al.* (1999), and Deckers *et al.* (2012). For an implementation with a nonlinear inverse in $\tilde{L}$ see Smolders *et al.* (2008). For an aeronautical fatigue testing application using a linear inverse in $\tilde{L}$, see Eksteen and Raath (2001; ignore comments about the relative accuracy of the time domain vs. the frequency domain system identification approaches in the paper).

Substitution of the GIB compensator (Eq. 3.46) and Eq. 3.44 into Eq. 3.45 gives

$$\begin{aligned}
\gamma &= \sup_{\Delta \neq 0} \frac{||\Delta - \Phi(u^{(\infty)}, \Delta)||_2}{||\Delta||_2} \\
&= \sup_{\Delta \neq 0} \frac{||\Delta - (L(T(u^{(\infty)})) - L(T(\Delta + u^{(\infty)})))||_2}{||\Delta||_2} \qquad (3.47) \\
&= \sup_{\Delta \neq 0} \frac{||\Delta - cC(\tilde{L}(T(u^{(\infty)})) - \tilde{L}(T(\Delta + u^{(\infty)})))||_2}{||\Delta||_2} \qquad (3.48)
\end{aligned}$$

When the convergence condition of Theorem 3.3, namely $\alpha\gamma < 1$ is violated, the role of a small c in recovering convergence (at the expense of a slower rate of convergence) is clear.

For the ideal GIB compensator with $\tilde{L} = T^{-1}$ and $C = 1$, i.e. $L = cT^{-1}$, Eq. 3.42 becomes

$$\begin{aligned}
\Delta^{(i+1)} &= Q\Delta^{(i)} - Q(cT^{-1}(T(u^{(\infty)})) - cT^{-1}(T(u^{(\infty)} - \Delta^{(i)}))) \\
&= (1-c)Q\Delta^{(i)} \; . \qquad (3.49)
\end{aligned}$$

Clearly, if $c = 1$, $\Delta^{(i+1)} = [0]$, i.e. convergence is achieved in one iteration, with the limit signals as given in Theorem 3.2. This results confirms the advantages of using an accurate inverse model in the ILC compensator: rapid convergence, widest possible frequency band of convergence (up to Nyquist frequency), precise control over the rate of convergence of $u^{(i)}$, and monotonic convergence of $u^{(i)}$.

3.3.5 Connections with Fixed Point Iteration Methods

The system formulation in the iteration domain for the GIB compensator (Eq. 3.46) with $c = 1$ and $Q = 1$ becomes (from Eq. 3.34)

$$u^{(i+1)} = u^{(i)} + cC\tilde{L}(y_d) - cC\tilde{L}(T(u^{(i)})) \; , \qquad (3.50)$$

$$= u^{(i)} + C\tilde{L}(y_d) - C\tilde{L}(T(u^{(i)})) \tag{3.51}$$

$$:= \mathcal{T}_{11}(u^{(i)}, y_d) \ . \tag{3.52}$$

Here we view $\mathcal{T}_{11}(u^{(i)}, y_d)$ as a type of baseline approach for ILC with the GIB compensator (with $c = 1$) and that corresponds to the Picard fixed point iteration approach. If we now formulate a standard Mann iteration scheme with the operator $\mathcal{T}_{11}(u^{(i)}, y_d)$, the result is

$$u^{(i+1)} = (1 - \alpha_i)u^{(i)} + \alpha_i \mathcal{T}_{11}(u^{(i)}, y_d) \tag{3.53}$$

$$= u^{(i)} + \alpha_i C\tilde{L}(y_d) - \alpha_i C\tilde{L}(T(u^{(i)})) \tag{3.54}$$

with $\alpha_i \in (0, 1]$. Comparing Eq. 3.54 with Eq. 3.50 shows that it resembles the system formulation for ILC with $Q = 1$ and with the GIB compensator with $c = \alpha_i$. Thus, performing ILC with the conventional algorithm with $Q = 1$ and the GIB compensator with $c \in (0, 1)$ essentially represents a special case of Mann iteration (using a constant α) with the operator $\mathcal{T}_{11}(u^{(i)}, y_d)$. This will be true for any ILC compensator featuring a constant scale factor. The use of the gain $c \in (0, 1)$ in such compensators thus has the same advantages for achieving convergence compared to the $c = 1$ case as Mann iteration (with a constant α) has compared to Picard iteration.

Note that it is possible to form an Ishikawa iteration scheme with $\mathcal{T}_{11}(u^{(i)}, y_d)$ as follows:

$$\begin{aligned} u^{(i+1)} &= (1 - \alpha_i)u^{(i)} + \alpha_i \mathcal{T}_{11}(\mu^{(i)}, y_d) \\ \mu^{(i)} &= (1 - \beta_i)u^{(i)} + \beta_i \mathcal{T}_{11}(u^{(i)}, y_d) \ , \end{aligned} \tag{3.55}$$

with $\alpha_i \in (0, 1]$ and $\beta_i \in [0, 1]$. It can be shown that a fixed point of Picard iteration is also a fixed point of Mann and Ishikawa iteration. Investigating the connections (if any) between Ishikawa iteration and existing ILC iteration schemes is a subject of future research.

3.4 Alternative ILC Algorithm Using a Nonlinear Inverse Model

3.4.1 Development

We consider the same nonlinear system and associated assumptions as in Section 3.3. Towards deriving a fundamentally different form of the ILC algorithm, we retain the Q filter in an optional capacity, but for the moment consider the $Q = 1$. When the plant output converges to y_d, the plant input at the same time converges to u_d. If the plant input, instead of being the parameter that is updated, is instead directly obtained as output of the ILC compensator L, then as the plant input converges to u_d, the input to the compensator, say $\tilde{y}$, will converge to a limit value $\tilde{y}_d$ so that

$$u_d = L(\tilde{y}_d) \ . \tag{3.56}$$

If we can formulate an update formula in the $\tilde{y}$ parameter, then the goal will be for $\tilde{y}$ to converge to $\tilde{y}_d$, because that will imply desired convergence of u and y to u_d and y_d respectively. This suggests the possibility of an update formula on the output side of the plant instead of on the input side as with conventional ILC (keep in mind L is essentially an inverse model of the plant). The following algorithm follows this approach. As before

$$y^{(i)} = T(u^{(i)}) \ . \tag{3.57}$$

We define a new parameter as input to the ILC compensator which, being an input to the ILC compensator, is related in type to the system output and thus we use the symbol $\tilde{y}$, and have

$$u^{(i)} = L(\tilde{y}^{(i)}) \ . \tag{3.58}$$

The new update formula is in the $\tilde{y}$ parameter, and is as follows:

$$\tilde{y}^{(i+1)} = Q(\tilde{y}^{(i)} + y_d - y^{(i)}) \ , \tag{3.59}$$

with initial value $\tilde{y}^{(0)} = 0$. If $Q = 1$ in Eq 3.59 then clearly, when $y^{(i)}$ converges to y_d, $\tilde{y}^{(i)}$ converges to a limit value that, working back through Eq. 3.57 and Eq. 3.58 for $y^{(i)} = y_d$ must be $\tilde{y}_d$ as defined in Eq. 3.56. In other words, when $\tilde{y}^{(i)}$ converges, by definition u and y converges to u_d and y_d respectively, which is the goal of ILC. Eq. 3.57, Eq. 3.58 and Eq. 3.59 represent an alternative algorithm for ILC on nonlinear systems when using a nonlinear ILC compensator. The essential difference is that the algorithm updates (a version of) the plant output instead of the plant input. The algorithm is shown in Fig. 3.2. Note that in the next section we will modify this algorithm and in the process ostensibly improve it, thus rendering the version presented here essentially a stepping stone or preliminary version. However, the research presented here focuses on implementing and evaluating the algorithm as presented in this section, with the evaluation of the modified version that is presented in the next section being the subject of future research.

3.4.2 Relationship with Conventional Algorithm

The system formulation in the iteration domain of the system input for the alternative algorithm is given as (combining Eq. 3.57, Eq. 3.58 and Eq. 3.59):

$$\begin{aligned} u^{(i+1)} &= L(\tilde{y}^{(i+1)}) \\ &= L(Q(\tilde{y}^{(i)} + y_d - y^{(i)})) \qquad (3.60) \\ &= L(Q(L^{-1}(u^{(i)}) + y_d - T(u^{(i)}))) \ . \qquad (3.61) \end{aligned}$$

Clearly, when L is linear then Eq. 3.61 reduces to Eq. 3.34. In other words, if L is linear then the alternative algorithm is equivalent to the conventional algorithm, even if the system T is nonlinear.

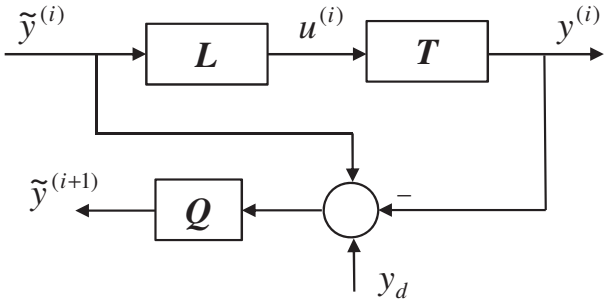

Figure 3.2: Alternative algorithm for ILC on a nonlinear system T using a nonlinear L.

The alternative algorithm is therefore only an option when we employ a nonlinear inverse, and thus for ILC on nonlinear plants only. Note further that by a simple modification the input system formulation for the conventional algorithm (Eq. 3.34) may be restated as (using Eq. 3.58)

$$u^{(i+1)} = Q(u^{(i)} + L(y_d) - L(y^{(i)})) \tag{3.62}$$

$$= Q(L(\tilde{y}^{(i)}) + L(y_d) - L(y^{(i)})) \tag{3.63}$$

Comparison of Eq. 3.60 for $Q = 1$, i.e.

$$u^{(i+1)} = L(\tilde{y}^{(i)} + y_d - y^{(i)}) \ ,$$

and Eq. 3.63 for $Q = 1$, i.e.

$$u^{(i+1)} = L(\tilde{y}^{(i)}) + L(y_d) - L(y^{(i)}) \ ,$$

shows that the essential difference between the two ILC algorithms is that in the alternative algorithm L operates on all three entities together, instead of separately as in the conventional algorithm. This comes at no apparent cost as the alternative algorithm is still capable of achieving convergence in one iteration when $L = T^{-1}$ (as will be shortly shown).

3.4.3 Convergence

By inserting Eq. 3.57 and Eq. 3.58 into Eq. 3.59 the system formulation in the iteration domain for $\tilde{y}$ is derived as

$$\tilde{y}^{(i+1)} = Q(\tilde{y}^{(i)} + y_d - T(L(\tilde{y}^{(i)}))) \ . \tag{3.64}$$

To analyze convergence we follow exactly the same approach as for the conventional algorithm, starting with the boundedness of the ILC algorithm.

Theorem 3.4: If T is BIBO stable in the 2-norm and $\alpha\bar{\beta}_0 < 1$ with $\alpha = ||Q||_\infty$ and

$$\bar{\beta}_0 = \sup_{\tilde{y}\neq 0} \frac{||\tilde{y} - T(L(\tilde{y}))||_2}{||\tilde{y}||_2} , \tag{3.65}$$

then Eq. 3.59 is BIBO stable in the 2-norm. The proof is similar to that of Theorem 3.1. $\diamondsuit$

Next we focus on the existence of convergence points for the fixed point iteration of Eq. 3.64, i.e. the of limit signals. Noting Eq. 3.37 the limit signals for Eq. 3.59 and Eq. 3.64 are respectively

$$\tilde{y}^{(\infty)} = Q(\tilde{y}^{(\infty)} + y_d - y^{(\infty)}) \tag{3.66}$$

$$= Q(\tilde{y}^{(\infty)} + y_d - T(L(\tilde{y}^{(\infty)}))) \tag{3.67}$$

If the limit signals exist, then for $Q = 1$ clearly from Eq. 3.67 $y_d = T(L(\tilde{y}^{(\infty)}))$, implying $\tilde{y}^{(\infty)} = L^{-1}(T^{-1}(y_d)) = L^{-1}(u_d) = \tilde{y}_d$ (cf. Eq. 3.56), in turn $u^{(\infty)} = L(\tilde{y}^{(\infty)}) = L(\tilde{y}_d) = u_d$, and from Eq. 3.57 $y^{(\infty)} = T(u^{(\infty)}) = T(u_d) = y_d$. If $Q \neq 1$, then from Eq. 3.66

$$\tilde{y}^{(\infty)} = Q(1 - Q)^{-1}(y_d - y^{(\infty)}) , \tag{3.68}$$

$u^{(\infty)} = L(\tilde{y}^{(\infty)})$, and $y^{(\infty)} = T(u^{(\infty)})$. We have the following theorem:

Theorem 3.5: If the conditions of Theorem 3.4 hold and L is BIBO stable, then for $Q \neq 1$ Eq. 3.68 is BIBO stable in the 2-norm and $\tilde{y}^{(\infty)}$, $u^{(\infty)}$ and $y^{(\infty)}$ are well defined. If $Q = 1$ then $\tilde{y}^{(\infty)}$, $u^{(\infty)}$ and $y^{(\infty)}$ are well defined, with $\tilde{y}^{(\infty)} = \tilde{y}_d$, $u^{(\infty)} = u_d$ and with $y^{(\infty)} = y_d$. The proof for the case $Q = 1$ follows from the preceding discussion. The proof for case $Q \neq 1$ is similar to Theorem 3.2. $\diamondsuit$

The iteration domain formulation for the convergence error of $\tilde{y}$ w.r.t. the limit signal $\tilde{y}^{(\infty)}$ is then (using Eq. 3.64 and Eq. 3.67)

$$\tilde{\Delta}_0^{(i+1)} = \tilde{y}^{(\infty)} - \tilde{y}^{(i+1)} \tag{3.69}$$

$$= Q(\tilde{y}^{(\infty)} + y_d - T(L(\tilde{y}^{(\infty)}))) - Q(\tilde{y}^{(i)} + y_d - T(L(\tilde{y}^{(i)})))$$

$$= Q(\tilde{y}^{(\infty)} - \tilde{y}^{(i)} - (T(L(\tilde{y}^{(\infty)})) - T(L(\tilde{y}^{(i)})))$$

$$= Q\tilde{\Delta}_0^{(i)} - Q(T(L(\tilde{y}^{(\infty)})) - T(L(\tilde{y}^{(\infty)} - \tilde{\Delta}_0^{(i)}))) \tag{3.70}$$

$$= Q\tilde{\Delta}_0^{(i)} - Q\tilde{\Phi}_0(\tilde{y}^{(\infty)}, \tilde{\Delta}_0^{(i)}) , \tag{3.71}$$

with

$$\tilde{\Phi}_0(\tilde{y}^{(\infty)}, \tilde{\Delta}_0^{(i)}) := T(L(\tilde{y}^{(\infty)})) - T(L(\tilde{y}^{(\infty)} - \tilde{\Delta}_0^{(i)})) . \tag{3.72}$$

We have the following sufficient condition for convergence:

Theorem 3.6: If the ILC update formula Eq. 3.59 is BIBO stable, the limit signal of Eq. 3.67 is well defined, and $\alpha\tilde{\gamma}_0 < 1$ with $\alpha = ||Q||_\infty$ and

$$\tilde{\gamma}_0 = \sup_{\tilde{\Delta}_0\neq 0} \frac{||\tilde{\Delta}_0 - \tilde{\Phi}_0(\tilde{y}^{(\infty)}, \tilde{\Delta}_0)||_2}{||\tilde{\Delta}_0||_2} , \tag{3.73}$$

then $||\tilde{\Delta}_0^{(i+1)}|| < \alpha\tilde{\gamma}_0||\tilde{\Delta}_0^{(i)}||$, that is, the convergence error of $\tilde{y}^{(i)}$ w.r.t. $\tilde{y}^{(\infty)}$ is monotonically decreasing and $\tilde{y}^{(i)}$ converges to $\tilde{y}^{(\infty)}$. The proof is similar to Theorem 3.3. $\diamondsuit$

Substitution of the GIB compensator (Eq. 3.46) and Eq. 3.72 into Eq. 3.73, gives

$$\tilde{\gamma}_0 = \sup_{\tilde{\Delta}_0 \neq 0} \frac{||\tilde{\Delta}_0 - (T(L(\tilde{y}^{(\infty)})) - T(L(\tilde{y}^{(\infty)} - \tilde{\Delta}_0))||_2}{||\tilde{\Delta}_0||_2} \tag{3.74}$$

$$= \sup_{\tilde{\Delta}_0 \neq 0} \frac{||\tilde{\Delta}_0 - (T(cC\tilde{L}(\tilde{y}^{(\infty)})) - T(cC\tilde{L}(\tilde{y}^{(\infty)} - \tilde{\Delta}_0)))||_2}{||\tilde{\Delta}_0||_2} \tag{3.75}$$

When the convergence condition of Theorem 3.6, namely $\alpha\tilde{\gamma}_0 < 1$ is violated, the role of a small c in recovering convergence (at the expense of a slower rate of convergence) is clear, however its effect is not as obvious as in the case of the conventional algorithm.

For the ideal GIB compensator with $\tilde{L} = T^{-1}$ and $C = 1$, i.e. $L = cT^{-1}$, Eq. 3.70 becomes

$$\tilde{\Delta}_0^{(i+1)} = Q\tilde{\Delta}_0^{(i)} - Q(T(cT^{-1}(\tilde{y}^{(\infty)})) - T(cT^{-1}(\tilde{y}^{(\infty)} - \tilde{\Delta}_0^{(i)}))) \; . \tag{3.76}$$

Clearly we do not have the same kind of simplification resulting from use of the ideal GIB compensator as we have for the conventional ILC algorithm (cf. Eq. 3.49). However, if $c = 1$ we again find $\tilde{\Delta}_0^{(i+1)} = 0$, i.e. convergence is still achieved in one iteration, with the limit signals as given in Theorem 3.5. While c will be effective in setting the rate of general convergence, the rate of convergence can not be as precisely controlled as for the conventional case due to the nonlinearity of T in Eq. 3.76.

3.4.4 Connection with Fixed Point Iteration Methods

The system formulation in the iteration domain for the GIB compensator (Eq. 3.46) with $c = 1$, and $Q = 1$ becomes (from Eq. 3.64)

$$\tilde{y}^{(i+1)} = \tilde{y}^{(i)} + y_d - T(cC\tilde{L}(\tilde{y}^{(i)})) \; , \tag{3.77}$$

$$= \tilde{y}^{(i)} + y_d - T(\,C\tilde{L}(\tilde{y}^{(i)})) \tag{3.78}$$

$$:= \bar{\mathcal{T}}_{11}^0(\tilde{y}^{(i)}, y_d) \; . \tag{3.79}$$

If we now formulate a standard Mann iteration scheme with the operator $\bar{\mathcal{T}}_{11}^0(\tilde{y}^{(i)}, y_d)$, we get

$$\tilde{y}^{(i+1)} = (1 - \alpha_i)\tilde{y}^{(i)} + \alpha_i\bar{\mathcal{T}}_{11}^0(\tilde{y}^{(i)}, y_d) \tag{3.80}$$

$$= (1 - \alpha_i)\tilde{y}^{(i)} + \alpha_i(\tilde{y}^{(i)} + y_d - T(\,C\tilde{L}(\tilde{y}^{(i)}))) \tag{3.81}$$

$$= \tilde{y}^{(i)} + \alpha_i y_d - \alpha_i T(\,C\tilde{L}(\tilde{y}^{(i)})) \tag{3.82}$$

with $\alpha_i \in (0, 1]$. Comparing Eq. 3.82 with Eq. 3.77 shows that performing ILC with $Q = 1$, with the alternative algorithm, and with the GIB compensator with $c \in (0, 1)$ is *not* equivalent to Mann iteration with the operator $\bar{\mathcal{T}}_{11}^0(\tilde{y}^{(i)}, y_d)$ for the special case of using a constant $\alpha_i = c$. This will be true for the preliminary algorithm for any ILC compensator featuring a constant scale factor.

3.5 Modified Alternative ILC Algorithm Using a Nonlinear Inverse

In this section we modify the alternative algorithm presented in Section 3.3 in order to obtain it in a way that again allows precise control over the rate of convergence, straight-forward adjustment of the convergence condition (γ) to ensure convergence, and equivalence between use of the GIB compensator for $c \in (0, 1)$ and Mann iteration with the operator obtained when using $c = 1$ in the GIB compensator. We consider the same nonlinear system and associated assumptions as in Section 3.3, and again retain the Q filter in an optional capacity. Define $\tilde{y}_d$ such that

$$u_d = \tilde{L}(\tilde{y}_d) \ , \tag{3.83}$$

with $\tilde{L}$ as defined in the GIB compensator (Eq. 3.46), i.e. as representing the inverse model *per se*. Furthermore,

$$y^{(i)} = T(u^{(i)}) \ , \tag{3.84}$$

and

$$u^{(i)} = \tilde{L}(\tilde{y}^{(i)}) \ . \tag{3.85}$$

We modify the update formula of the preliminary alternative algorithm (Eq. 3.59) as follows:

$$\tilde{y}^{(i+1)} = Q(\tilde{y}^{(i)} + cC(y_d - y^{(i)})) \ , \tag{3.86}$$

with initial value $\tilde{y}^{(0)} = [0]$, and c and C as in the GIB compensator. When $Q = 1$ clearly, when $y^{(i)}$ converges to y_d in Eq. 3.86, $\tilde{y}^{(i)}$ and $u^{(i)}$ converges to $\tilde{y}_d$ and u_d respectively. Eq. 3.84, Eq. 3.85 and Eq. 3.86 represent a modified form of the alternative algorithm for ILC on nonlinear systems when using a nonlinear ILC compensator.

3.5.1 Relationship with Conventional Algorithm

The system formulation in the iteration domain of the system input for the modified alternative algorithm is given as (combining Eq. 3.84, Eq. 3.85 and Eq. 3.86):

$$\begin{aligned} u^{(i+1)} &= \tilde{L}(\tilde{y}^{(i+1)}) \\ &= \tilde{L}(Q(\tilde{y}^{(i)} + cC(y_d - y^{(i)}))) \qquad (3.87) \\ &= \tilde{L}(Q(\tilde{L}^{-1}(u^{(i)}) + cC(y_d - T(u^{(i)})))) \ . \qquad (3.88) \end{aligned}$$

When $\tilde{L}$ is linear Eq. 3.88 reduces to Eq. 3.34 and the modified alternative algorithm becomes equivalent to the conventional algorithm (even for nonlinear T), and is therefore only an option when we employ a nonlinear inverse.

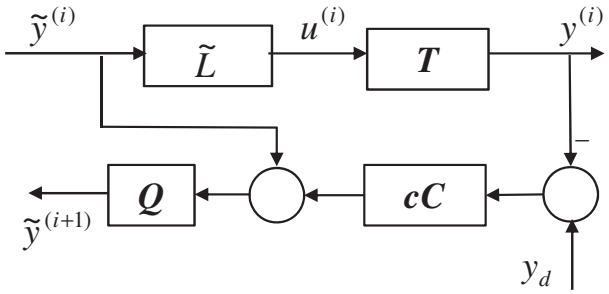

Figure 3.3: Modified version of the alternative algorithm for ILC on a nonlinear system T with a nonlinear L.

Note further that by a simple modification the input system formulation for the conventional algorithm (Eq. 3.34) may be restated as (using Eq. 3.85)

$$u^{(i+1)} = Q(u^{(i)} + L(y_d) - L(y^{(i)})) \tag{3.89}$$

$$= Q(\tilde{L}(\tilde{y}^{(i)}) + cC(\tilde{L}(y_d) - \tilde{L}(y^{(i)}))) \tag{3.90}$$

Comparison of Eq. 3.87 for $Q = 1$, i.e.

$$u^{(i+1)} = \tilde{L}(\tilde{y}^{(i)} + cC(y_d - y^{(i)})) \ ,$$

and Eq. 3.90 for $Q = 1$, i.e.

$$u^{(i+1)} = \tilde{L}(\tilde{y}^{(i)}) + cC(\tilde{L}(y_d) - \tilde{L}(y^{(i)}))$$

shows that the essential difference between the two ILC algorithms is not just that in the modified alternative algorithm $\tilde{L}$ operates on all three entities together, instead of separately as in the conventional algorithm, but is also the location of the cC factor w.r.t. $\tilde{L}$ (i.e. "inside" $\tilde{L}$ vs. "outside" $\tilde{L}$). This comes at no apparent cost as the alternative algorithm is still capable of achieving convergence in one iteration when $L = T^{-1}$ (as will be shortly shown).

3.5.2 Convergence

By inserting Eq. 3.84 and Eq. 3.85 into Eq. 3.86 the system formulation in the iteration domain for $\tilde{y}$ is derived as

$$\tilde{y}^{(i+1)} = Q(\tilde{y}^{(i)} + cC(y_d - T(\tilde{L}(\tilde{y}^{(i)})))) \ . \tag{3.91}$$

To analyze convergence we again follow the same approach as for the conventional algorithm.

Theorem 3.7: If T is BIBO stable in the 2-norm and $\alpha\bar{\beta} < 1$ with $\alpha = ||Q||_\infty$ and

$$\bar{\beta} = \sup_{\tilde{y}\neq 0} \frac{||\tilde{y} - cCT(\tilde{L}(\tilde{y}))||_2}{||\tilde{y}||_2} , \tag{3.92}$$

then Eq. 3.86 is BIBO stable in the 2-norm. The proof is similar to that of Theorem 3.1. $\diamondsuit$

Noting Eq. 3.37 the limit signals for Eq. 3.86 and Eq. 3.91 are defined by respectively

$$\begin{aligned} \tilde{y}^{(\infty)} &= Q(\tilde{y}^{(\infty)} + cC(y_d - y^{(\infty)})) && (3.93)\\ &= Q(\tilde{y}^{(\infty)} + cC(y_d - T(\tilde{L}(\tilde{y}^{(\infty)})))) . && (3.94) \end{aligned}$$

If the limit signals exist, then for $Q = 1$ clearly from Eq. 3.94 $y_d = T(\tilde{L}(\tilde{y}^{(\infty)}))$, implying $\tilde{y}^{(\infty)} = \tilde{L}^{-1}(T^{-1}(y_d)) = \tilde{L}^{-1}(u_d) = \tilde{y}_d$ (cf. Eq. 3.83), in turn $u^{(\infty)} = \tilde{L}(\tilde{y}^{(\infty)}) = \tilde{L}(\tilde{y}_d) = u_d$, and $y^{(\infty)} = T(u^{(\infty)}) = T(u_d) = y_d$. If $Q \neq 1$, then from Eq. 3.93

$$\tilde{y}^{(\infty)} = cCQ(1 - Q)^{-1}(y_d - y^{(\infty)}) , \tag{3.95}$$

$u^{(\infty)} = L(\tilde{y}^{(\infty)})$, and $y^{(\infty)} = T(u^{(\infty)})$. We have the following theorem:

Theorem 3.8: If the conditions of Theorem 3.7 hold and L is BIBO stable, then for $Q \neq 1$ Eq. 3.95 is BIBO stable in the 2-norm and $\tilde{y}^{(\infty)}$, $u^{(\infty)}$ and $y^{(\infty)}$ well defined. If $Q = 1$ then $\tilde{y}^{(\infty)}$, $u^{(\infty)}$ and $y^{(\infty)}$ are well defined, with $\tilde{y}^{(\infty)} = \tilde{y}_d$, $u^{(\infty)} = u_d$ and with $y^{(\infty)} = y_d$. The proof for the case $Q = 1$ follows from the preceding discussion. The proof for case $Q \neq 1$ is similar to Theorem 3.2. $\diamondsuit$

The iteration domain formulation for the convergence error of the input signal w.r.t. the limit signal is then (using Eq. 3.91 and Eq. 3.94)

$$\begin{aligned} \tilde{\Delta}^{(i+1)} &= \tilde{y}^{(\infty)} - \tilde{y}^{(i+1)} && (3.96)\\ &= Q(\tilde{y}^{(\infty)} + cC(y_d - T(\tilde{L}(\tilde{y}^{(\infty)})))) - Q(\tilde{y}^{(i)} + cC(y_d - T(\tilde{L}(\tilde{y}^{(i)})))) \\ &= Q(\tilde{y}^{(\infty)} - \tilde{y}^{(i)} - cC(T(\tilde{L}(\tilde{y}^{(\infty)})) - T(\tilde{L}(\tilde{y}^{(i)}))) \\ &= Q\tilde{\Delta}^{(i)} - QcC(T(\tilde{L}(\tilde{y}^{(\infty)})) - T(\tilde{L}(\tilde{y}^{(\infty)} - \tilde{\Delta}^{(i)}))) && (3.97)\\ &= Q\tilde{\Delta}^{(i)} - Q\tilde{\Phi}(\tilde{y}^{(\infty)}, \tilde{\Delta}^{(i)}) , && (3.98) \end{aligned}$$

with

$$\tilde{\Phi}(\tilde{y}^{(\infty)}, \tilde{\Delta}^{(i)}) := cC(T(\tilde{L}(\tilde{y}^{(\infty)})) - T(\tilde{L}(\tilde{y}^{(\infty)} - \tilde{\Delta}^{(i)}))) . \tag{3.99}$$

We have the following sufficient condition for convergence:

Theorem 3.9: If the ILC update formula Eq. 3.86 is BIBO stable, the limit signal of Eq. 3.94 is well defined, and $\alpha\tilde{\gamma} < 1$ with $\alpha = ||Q||_\infty$ and

$$\tilde{\gamma} = \sup_{\tilde{\Delta}\neq 0} \frac{||\tilde{\Delta} - \tilde{\Phi}(\tilde{y}^{(\infty)}, \tilde{\Delta})||_2}{||\tilde{\Delta}||_2} , \tag{3.100}$$

then $||\tilde{\Delta}^{(i+1)}|| < \alpha\tilde{\gamma}||\tilde{\Delta}^{(i)}||$, that is, the convergence error of $\tilde{y}^{(i)}$ w.r.t. $\tilde{y}^{(\infty)}$ is monotonically decreasing and $\tilde{y}^{(i)}$ converges to $\tilde{y}^{(\infty)}$. The proof is similar to Theorem 3.3. $\diamondsuit$

Substituting Eq. 3.99 into Eq. 3.100, giving

$$\tilde{\gamma} \;=\; \sup_{\tilde{\Delta}\neq 0} \frac{||\tilde{\Delta} - c\,C(T(\tilde{L}(\tilde{y}^{(\infty)})) - T(\tilde{L}(\tilde{y}^{(\infty)} - \tilde{\Delta})))||_2}{||\tilde{\Delta}||_2} . \tag{3.101}$$

When the convergence condition of Theorem 3.9, namely $\alpha\gamma < 1$ is violated, the role of a small c in recovering convergence (at the expense of a slower rate of convergence) is again clear.

For the ideal inverse compensator $\tilde{L} = T^{-1}$ and $C = 1$ Eq. 3.97 becomes

$$\begin{aligned} \tilde{\Delta}^{(i+1)} &= Q\tilde{\Delta}^{(i)} - Qc(T(T^{-1}(\tilde{y}^{(\infty)})) - T(T^{-1}(\tilde{y}^{(\infty)} - \tilde{\Delta}^{(i)}))) \\ &= Q(1-c)\tilde{\Delta}^{(i)} . \end{aligned} \tag{3.102}$$

Clearly we again have the same kind of simplification resulting from use of the ideal GIB compensator as we have for the conventional ILC algorithm (cf. Eq. 3.49). Furthermore, if $c = 1$ we again find $\tilde{\Delta}^{(i+1)} = [0]$, i.e. convergence is still achieved in one iteration, with the limit signals as given in Theorem 3.8. In addition the rate of convergence can be precisely controlled as in the conventional case.

3.5.3 Connection with Fixed Point Iteration Methods

The system formulation in the iteration domain for $Q = 1$ becomes (from Eq. 3.91)

$$\tilde{y}^{(i+1)} = \tilde{y}^{(i)} + cC(y_d - T(\tilde{L}(\tilde{y}^{(i)}))) , \tag{3.103}$$

and for $c = 1$ the system formulation then becomes

$$\begin{aligned} \tilde{y}^{(i+1)} &= \tilde{y}^{(i)} + C(y_d - T(\tilde{L}(\tilde{y}^{(i)}))) && (3.104) \\ &:= \bar{\mathcal{T}}_{11}(\tilde{y}^{(i)}, y_d) . && (3.105) \end{aligned}$$

If we now formulate a standard Mann iteration scheme with the operator $\bar{\mathcal{T}}_{11}(\tilde{y}^{(i)}, y_d)$, we get

$$\begin{aligned} \tilde{y}^{(i+1)} &= (1-\alpha_i)\tilde{y}^{(i)} + \alpha_i\bar{\mathcal{T}}_{11}(\tilde{y}^{(i)}, y_d) \\ &= (1-\alpha_i)\tilde{y}^{(i)} + \alpha_i(\tilde{y}^{(i)} + C(y_d - T(\tilde{L}(\tilde{y}^{(i)})))) \\ &= \tilde{y}^{(i)} + \alpha_i C(y_d - T(\tilde{L}(\tilde{y}^{(i)}))) \end{aligned} \tag{3.106}$$

with $\alpha_i \in (0, 1]$. Comparing Eq. 3.106 with Eq. 3.103 shows that performing ILC with the modified alternative algorithm with $Q = 1$ and $c \in (0, 1)$ is equivalent to Mann iteration with the operator $\bar{\mathcal{T}}_{11}(\tilde{y}^{(i)}, y_d)$ (for the special case of a constant α).

3.6 Example 4

In this example we perform ILC on the same system used in Example 2, Chapter 2, namely Eq. 2.111. In ILC the dynamics of the physical system is usually not exactly known, though the test engineer usually will have an approximate model of the system. When inverted, the resulting approximate inverse model may be used in inverse model-based ILC on the system. While an exact model for the "system" in this case is available, in order to simulate a practical situation (where the exact system model is rarely known) we here also do ILC with the inverse of an approximate model. Such an approximate model may be obtained by system identification on input-output data for the system, and may be either linear or nonlinear.

In Case 1 we demonstrate the convergence of the conventional and alternative ILC algorithms for a short-duration deterministic signal (the same desired input u_d and output y_d that was used in Example 2). In Case 2 we present an example where the alternative ILC algorithm converges while the conventional algorithm diverges for a random signal.

In Case 3 to Case 4 we use the same desired input u_d and output y_d that was used in Example 3. In Case 3 we perform ILC using the inverse of a linear approximate model. In Case 4 we perform ILC using the inverse of a nonlinear approximate model, and compare the results with those of Case 3. Both the linear and nonlinear models of Cases 3 and 4 are obtained from system identification. In each case we will also compare the results of ILC using the conventional and alternative ILC algorithms.

3.6.1 Case 1: ILC for a Deterministic Desired Signal

This case demonstrates the ability of the alternative and conventional ILC algorithms to converge for a short-duration deterministic signal (the same desired input u_d and output y_d that was used in Example 2). The test system is represented by Eq. 2.111

$$\begin{aligned} y(k) \;=\; & \theta_1 u(k-4) + \theta_2 u(k-5) + \theta_3 u(k-6) + \theta_4 y(k-4) \\ & \theta_5 u(k-5) y(k-4) + \theta_6 u(k-5) u(k-6) y(k-2) \\ & + \theta_7 u(k-5)^2 u(k-6) y(k-1) \;, \end{aligned}$$

with

$$(\theta_1, \ldots, \theta_7) = (0.150,\ 0.50,\ 0.50,\ 1/6,\ -2.0,\ 6.0,\ 11.0) \;.$$

The desired response in question is obtained as the response of Eq. 2.111 to the following relatively short deterministic signal, $u_d(k)$:

$$\bar{u}_d(k) = \begin{cases} 0, & 1 \leq k \leq 25 \\ c_u(\sin(2\pi(k-31)/20)+1), & 25 < k \leq 46 \\ 0, & 46 < k \leq 146 \end{cases} \tag{3.107}$$

$$\tilde{u}_d = \bar{F}_{0.16}\bar{u}_d \tag{3.108}$$

$$u_d = C_T\tilde{u}_d \tag{3.109}$$

$$F_{0.16}(z) = \frac{0.02287z^4 + 0.09148z^3 + 0.13722z^2 + 0.09148z + 0.02287}{1.00z^4 - 1.412z^3 + 1.123z^2 - 0.40807z + 0.06321} \tag{3.110}$$

$$C_T = \begin{cases} 0, & 1 \leq k \leq 12 \\ 0.5\sin(2\pi(k-13)/24 - \pi/2) + 0.5, & 12 < k \leq 24 \\ 1, & 24 < k \leq 122 \\ 0.5\sin(2\pi(k-123)/24 + \pi/2) + 0.5, & 122 < k \leq 134 \\ 0, & 134 < k \leq 146 \end{cases} \tag{3.111}$$

with $c_u = 0.165$, the second equation in operator format, and $\bar{F}$ a non-causal linear operator representing the zero phase version of the low pass filter $F(z)$ with cut frequency 40 Hz (0.16 times the sample frequency). C_T as given by Eq. 3.111 is essentially a sinusoidal taper function. The desired input signal and desired response signal is shown in Fig. 3.4. For purposes of obtaining the inverse-model based ILC compensator we obtained the following approximate (nonlinear) NARX system model by system identification:

$$\begin{aligned} y(k) = {} & -0.076129u(t-2) + 0.21444u(t-3) - 0.0040361u(t-10) \\ & +2.6585y(t-1) - 3.6336y(t-2) + 3.2178y(t-3) \\ & -1.8455y(t-4) + 0.56977y(t-5) - 0.039669y(t-7) \\ & +0.12012u(k-3)u(k-4) - 0.12388u(k-3)u(k-10) \\ & -0.12629u(k-3)y(k-2)\ . \end{aligned} \tag{3.112}$$

Due to the nonlinearity of the model, stable inversion of the model is iterative. A gain of 0.1 was used in stable inversion. For ILC a zero-phase low pass ILC filter Q with cut frequency of 40 Hz was used and a ILC gain of $c = 0.3$ was used. The results for both the conventional and alternative ILC algorithms are presented in Fig. 3.5 and Fig. 3.6. Clearly both the conventional and alternative ILC algorithms converge with the alternative algorithm converging more rapidly and to lower convergence errors values. The minimum convergence error values in these figures correspond to virtually imperceptible differences between the desired and achieved input and response signals. Note that sinusoidal tapering (of duration 0.03 sec) was applied during stable inversion during the alternative ILC algorithm to suppress build-up of spurious high frequency oscillations at the start of the signal.

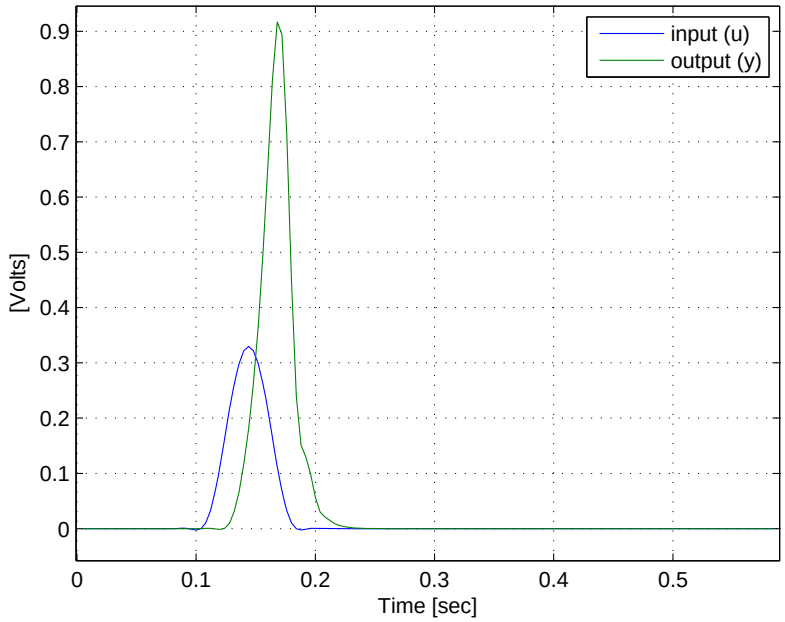

Figure 3.4: Case 1: Desired input $u_d(t)$ and output $y_d(t)$ data.

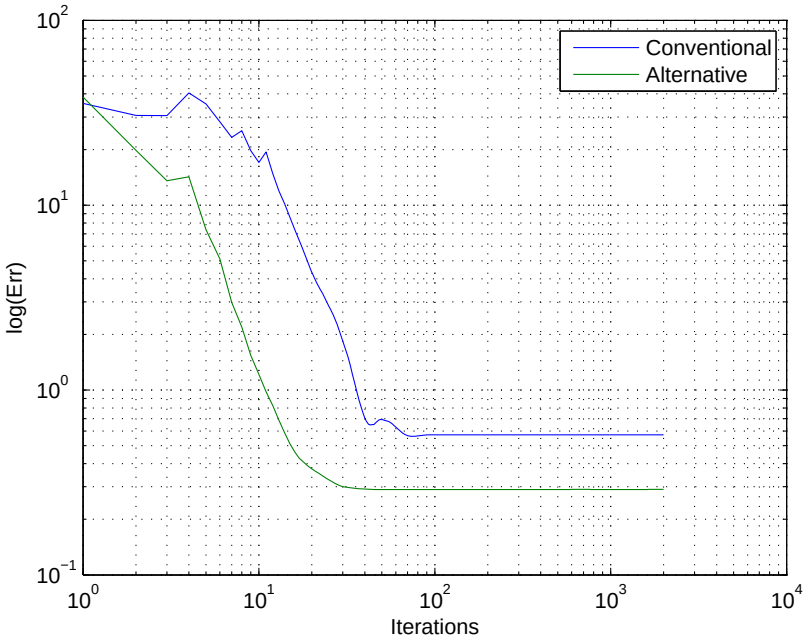

Figure 3.5: Case 1: Convergence error of the input with respect to the desired input for the conventional and alternative ILC algorithms.

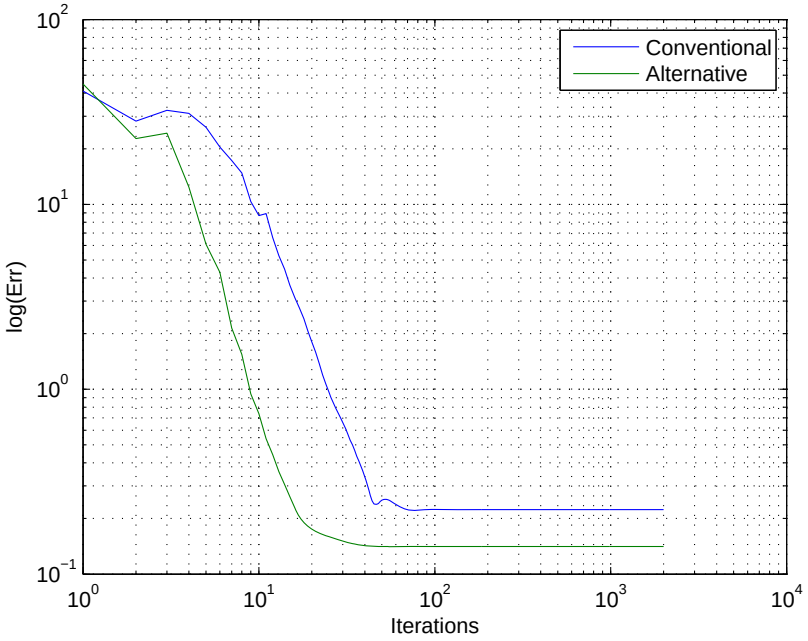

Figure 3.6: Case 1: Convergence error of the output with respect to the desired response for the conventional and alternative ILC algorithms.

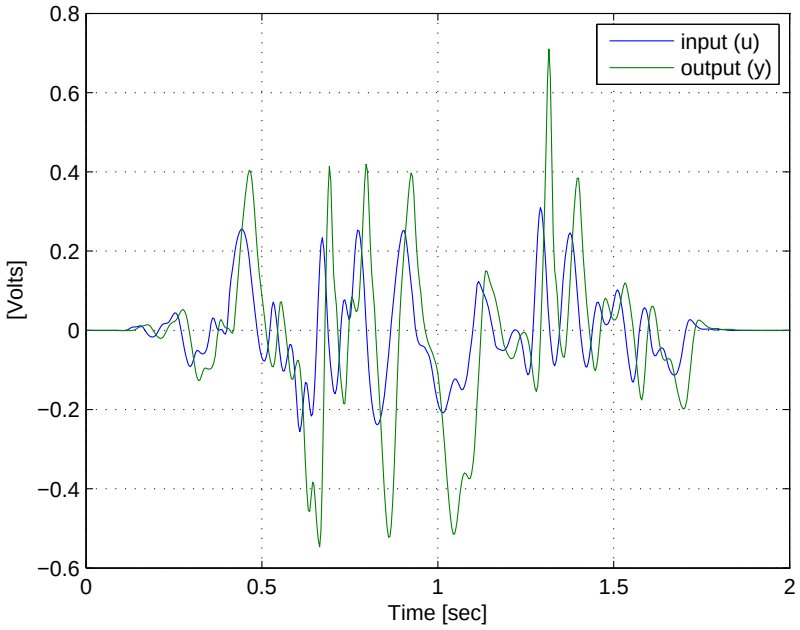

Figure 3.7: Case 2: Desired input $u_d(t)$ and response $y_d(t)$ data.

3.6.2 Case 2: ILC for a Random Desired Signal

In this case ILC is performed on Eq. 2.111 using the low-level random input and corresponding output signal of Example 2, shown in Fig. 3.7, to demonstrate the ability of the alternative ILC algorithm to converge when the conventional algorithm diverges in this particular case. For the ILC compensator the stable inverse of Eq. 3.112 was again used, this time with a stable inversion gain of 0.2. An ILC gain of $c = 0.4$ was used and a zero-phase low pass ILC filter Q was used with a cut frequency 50 Hz.

The results for both the conventional and alternative ILC algorithms are presented in Fig. 3.8 and Fig. 3.9, which shows that the conventional ILC algorithm diverges and alternative ILC algorithm converges. The best achieved response for the conventional ILC algorithm is shown in Fig. 3.10, while that of the alternative ILC algorithm is visually virtually indistinguishable from the desired response signal and therefore not shown. The approach here (as in Case 1) of using the Q filter and not the C filter to achieve the low pass frequency cut off for ILC follows from the fact that that is the intention of the Q filter in normal practise, whereas the C filter is rather intended to be a *shaping* filter used to increase the bandwidth of convergence.

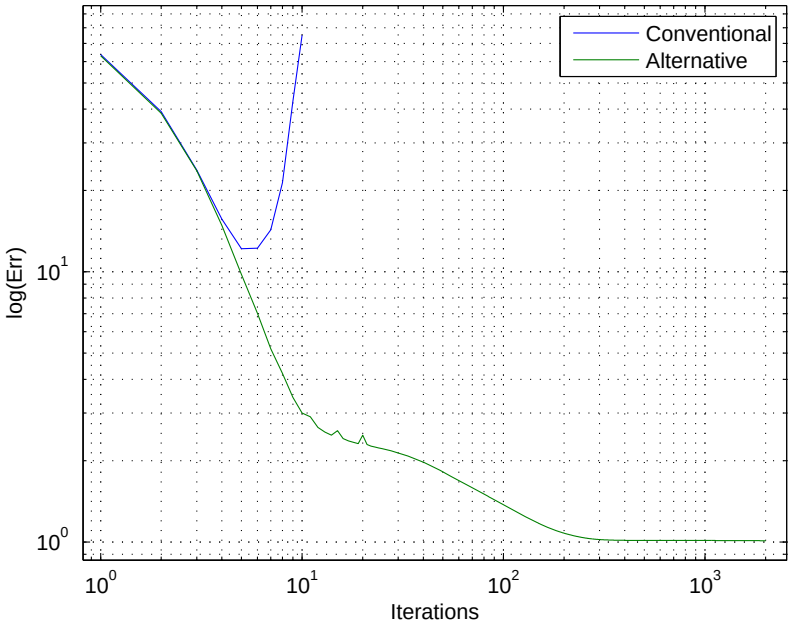

Figure 3.8: Case 2: Convergence error of the input with respect to the conventional and alternative ILC algorithms.

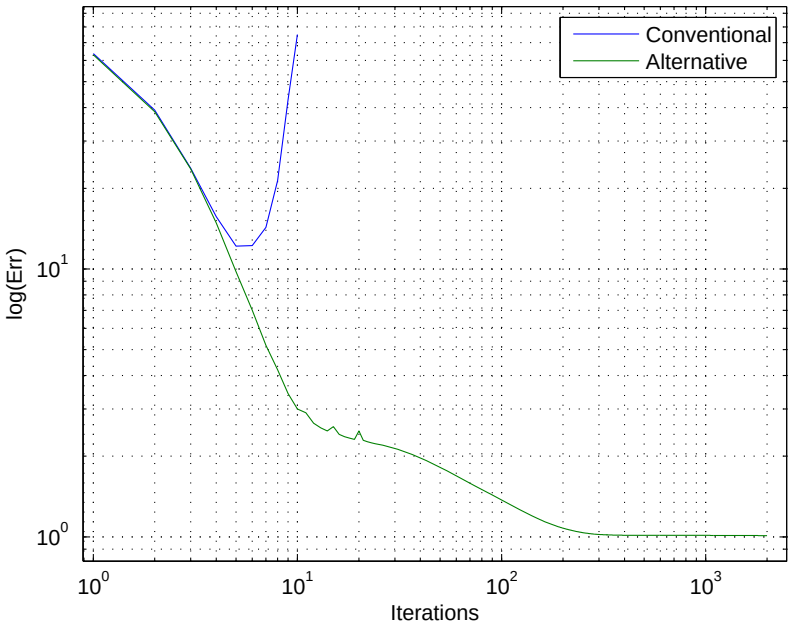

Figure 3.9: Case 2: Convergence error of the output with respect to the conventional and alternative ILC algorithms.

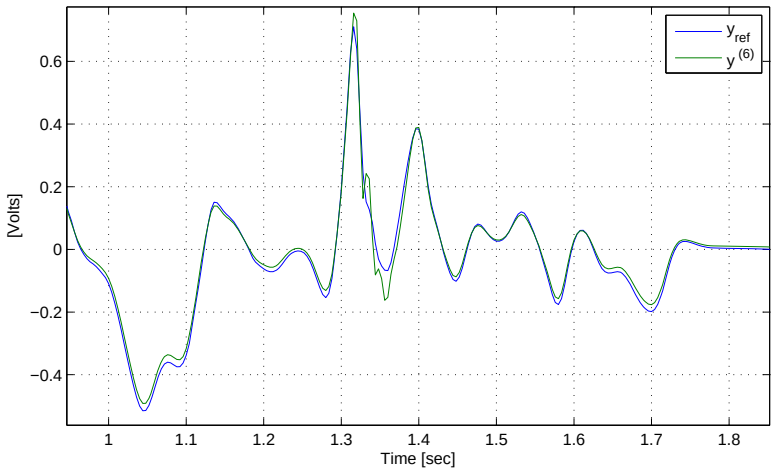

Figure 3.10: Case 2: Most accurate achieved output (response) for the conventional ILC algorithm (only a section is shown).

3.6.3 Case 3: ILC Using an Approximate Linear Inverse Model

In this case ILC is performed on Eq. 2.111 using the higher-level random input and corresponding output signal of Example 3, shown in Fig. 3.11. The purpose is to demonstrate the success of ILC using a linear inverse model-based ILC compensator. This doesn't imply that ILC converges, in fact it diverges in this case, and from a purely ILC point of view that will be a drawback, with the approach to remedy this to tailor the Q and C filters and the c gain to obtain the largest bandwidth possible that still gives convergence of ILC. However, in response reconstruction for structural integrity testing we do not have a purely ILC point of view that requires convergence at all costs, but rather we *use* ILC for a limited number of iterations to obtain the most accurate results possible over the largest bandwidth possible (or over a fixed, given bandwidth), with ILC terminated as soon as the most accurate results have been obtained. Thus, in response reconstruction it is tolerable if ILC diverges, as long as the results obtained before divergence is sufficiently accurate accurate for the purposes of the test.

A zero-phase, low pass ILC filter Q is employed, with the cut frequency of Q matching the values of the stable inversion filter in Example 3, namely respectively 50, 70, 90 Hz, including the option of using no filter. While there is very little signal strength in the desired response above 50 Hz, it is very much the higher frequencies (50 Hz and above) that are responsible for divergence of ILC in this case. It is therefore sensible to investigate the use of low pass ILC Q filters with cut frequencies of 50 Hz and higher to investigate the reduction in the severity of divergence while not severely reducing the signal frequency range.

In order to systematically evaluate ILC for the various values of Q a range of ILC gains will be used for every value of Q, including both iteration independent and iteration dependent gains, from which the best results may subsequently be selected. The formula for the iteration dependent gain $c^{(i)}$ is given as

$$c^{(i)} = 2\frac{c_0 - c_{\text{lim}}}{i+1} + c_{\text{lim}}\,, \tag{3.113}$$

noting that $c^{(0)} = c_0$.

For purposes of the inverse-model based ILC compensator a linear inverse model was obtained by performing system identification, and inverting the resulting model via stable inversion. Stable inversion of a linear model is accomplished in a single pass, and is exact (in the sense that the calculated input signal, when passed back through the model, recovers the given output signal used in the inversion). The following (linear) ARX model of the system was obtained (prior to inversion):

$$\begin{aligned} y(k) \;=\; & -0.0020592u(k-4) + 0.53042u(k-5) - 0.33295u(k-6) + 2.9341y(k-1) \\ & -4.5461y(k-2) + 4.6988y(k-3) - 3.3494y(k-4) + 1.542y(k-5) \\ & -0.34179y(k-6)\,. \end{aligned} \tag{3.114}$$

The best results of ILC on Eq. 2.111 using the stable inverse of this model is presented in Table 3.1. Note that when the ILC compensator L is linear, the conventional and alternative ILC algorithms are equivalent, for which reason we do not distinguish between the two approaches here. The best input signal that was obtained had an error of 18.0%, and the corresponding output, that is obtained by passing the input back through Eq. 2.111 (i.e. the system), had an error of 9.9%. These results do not represent an improvement on the results of stable inversion Eq. 2.111 for the same desired input and output signal in Example 3, for which we get lower input and output errors of 13.3% and 8.1% respectively. It is found that in all cases except when using no Q filter the best ILC results were obtained using iteration-dependent ILC gains.

Table 3.1: Case 3: Best results of ILC. M is the iteration resulting in $\min_m \text{err}_1(u^{(m)})$, i.e. $M = \text{argmin}_m \text{err}_1(u^{(m)})$.

Q-filter cut freq. [Hz]	$\min_m \text{err}_1(u^{(m)})$ [%]	$\text{err}_1(y^{(M)})$ [%]	M = Iter. no.	Comment
50	18.0	9.9	432	$c_0 = 0.1$; $c_{\text{lim}} = .01$ (it. var.)
70	21.4	11.4	86	$c_0 = 0.3$; $c_{\text{lim}} = .025$ (it. var.)
90	41.1	29.4	78	$c_0 = 0.1$; $c_{\text{lim}} = .01$ (it. var.)
None	52.3	75.0	1	$c = 1$ for u; $c = 0.4$ for y

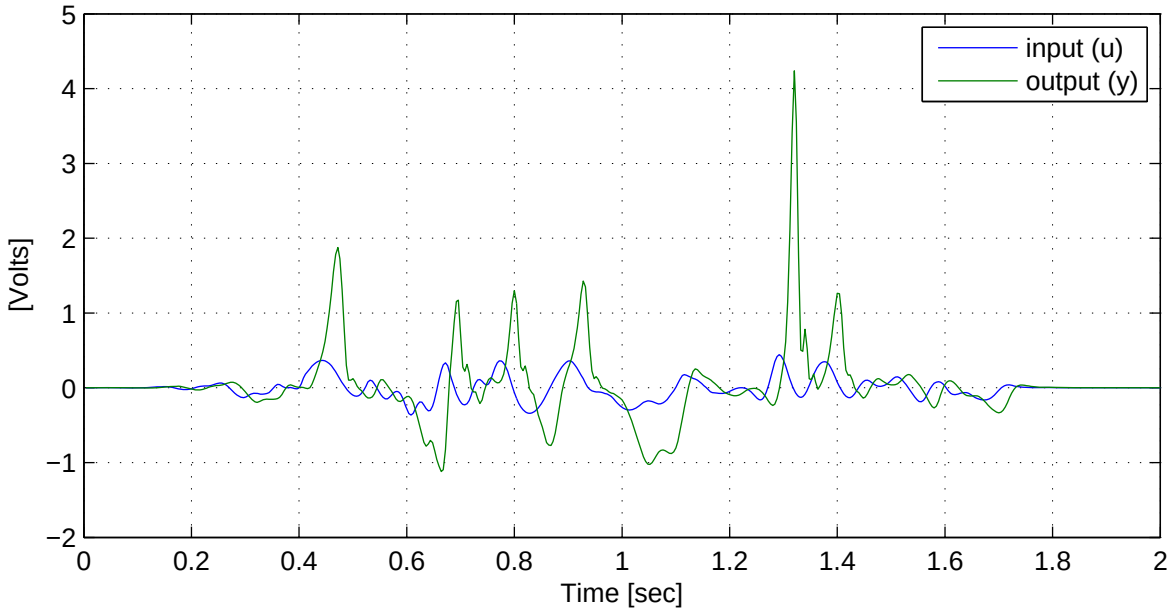

Figure 3.11: Case 3: Desired input $u_d(t)$ and output $y_d(t)$ data.

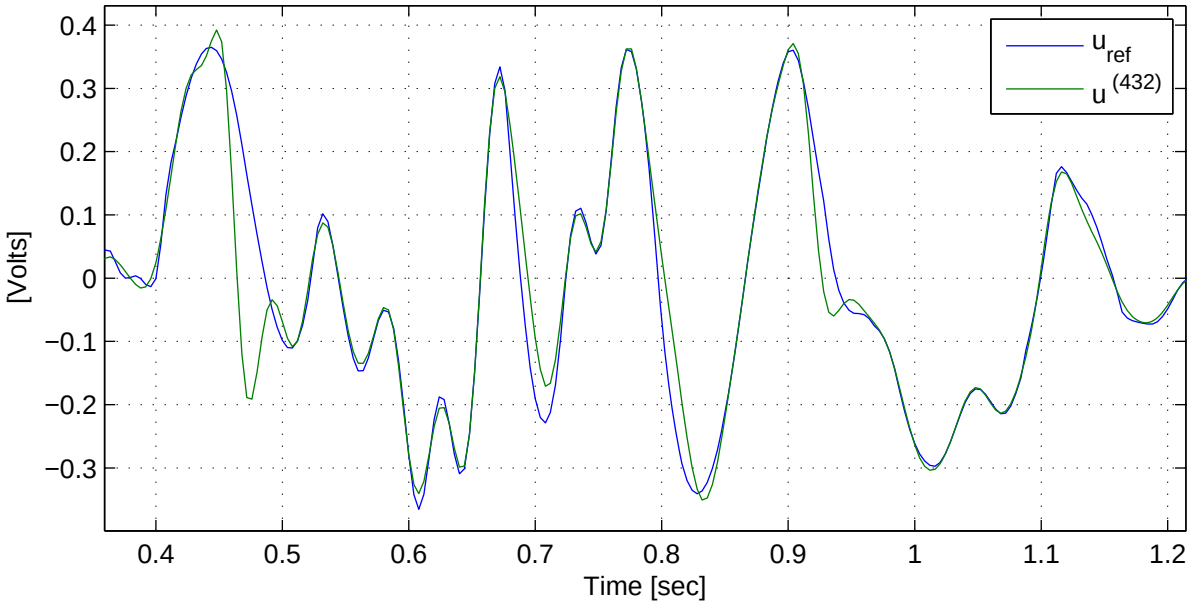

Figure 3.12: Case 3: Best results for ILC input signal $u^{(i)}(t)$.

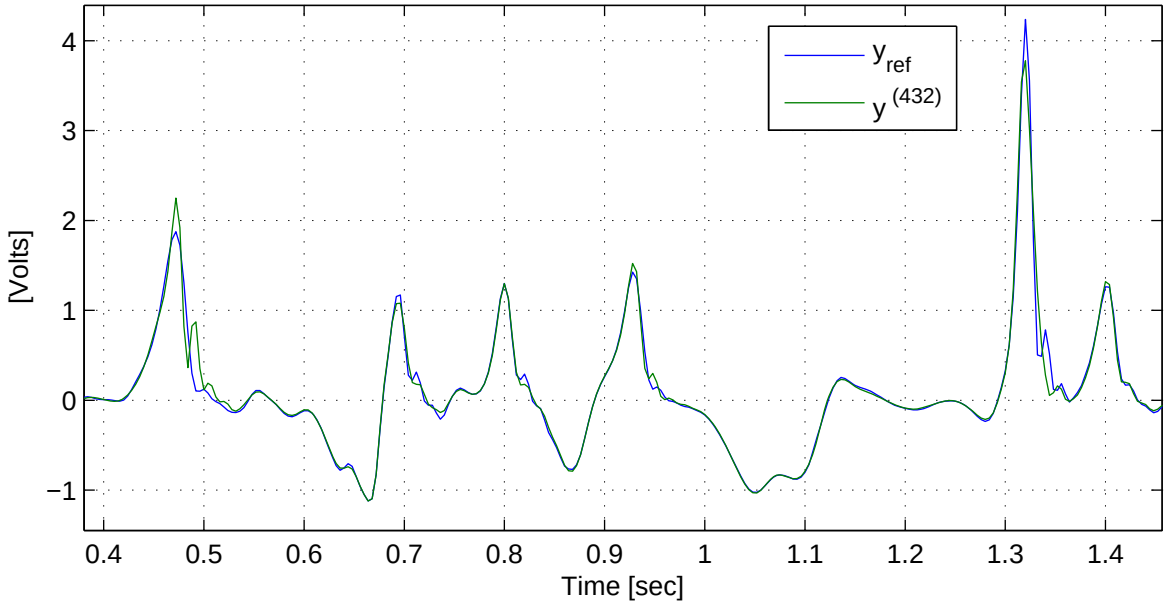

Figure 3.13: Case 3: Best results for ILC output signal $y^{(i)}(t)$.

3.6.4 Case 4: ILC Using an Approximate Nonlinear Inverse Model

In this case, as in Case 3, ILC is performed on Eq. 2.111 using the higher level random input and corresponding output signal of Example 3, shown in Fig. 3.11. However, for the inverse model-based ILC compensator we use here a nonlinear inverse model for purposes of comparison of the results with those of Case 3. For the nonlinear inverse model the following approximate NARX model was obtained by system identification:

$$\begin{aligned} y(k) \;=\; & 0.23043u(t-4) + 0.10461u(t-6) - 0.037279u(t-10) \\ & +1.7955y(t-1) - 2.0957y(t-2) + 1.7392y(t-3) \\ & -0.88389y(t-4) + 0.24564y(t-5) \\ & -0.29621u(k-5)y(k-5) - 1.2585u(k-5)^3 \\ & +7.2289u(k-5)^2y(k-1) - 9.2816u(k-5)u(k-6)y(k-2) \\ & +4.4206u(k-5)u(k-7)y(k-3) - 0.30189u(k-10)y(k-2)y(k-9) \\ & +0.041543y(k-1)^2y(k-10)\ . \end{aligned} \tag{3.115}$$

The model has been obtained using the methods discussed in the next chapter and is not particularly optimized, but is representative of the type of nonlinear models obtained in response reconstruction using automated methods for searching for best models over a range of model structures.

Since the model is nonlinear, stable inversion of the model is iterative. Since the iteration of stable inversion in this case is not convergent, regardless of the choice of gain in stable inversion, a rough optimization was done to determine the choice of gain in stable inversion that gives stable inversion results comparable to the best possible results. This was deemed sufficient as the desired input is usually not available in practise to determine inversion accuracy, systematic optimization can be an intensive exercise and often doesn't result in significant gains over the results obtained with rough optimization, and ILC does not critically depend on the accuracy of the inverse but is capable of iterative correction in the calculation of the input, i.e. is somewhat robust against inaccuracies in the accuracy of the inverse model (within limits of course). The number of iterations in stable inversion was limited to about 1000 iterations (and more where more was needed before divergence occurred).

ILC is evaluated in this case for various combinations of ILC low pass filters Q and ILC gains c, and for both the conventional and alternative ILC algorithms. The best results obtained using the conventional ILC algorithm is shown in Table 3.2, and for the alternative ILC algorithm in Table 3.3. Negative ILC gains c in Table 3.2 and Table 3.3 implies that an iteration dependent ILC gain was used with $c_0 = |c|$ and $c_{\lim} = 0.005$ (cf. Eq. 3.113). Only time-independent ILC gains were used.

Results are presented for both iteration-independent and iteration dependent stable inversion gains, indicated as stable inversion cases A and B respectively, and for time-independent and time dependent stable inversion gains, indicated as stable inversion cases 1 and 2 respectively. Refer to Table 3.4. The formula for iteration dependent stable inversion gains used here is the same as for iteration

dependent ILC gains, namely Eq. 3.113, with $c_0 = c_{\text{SI}}$ and $c_{\text{lim}} = c_{\text{lim,SI}}$. The time-dependent stable inversion gain for stable inversion iteration j and input channel i, designated $C_{v,i}^{(j)}(k)$, was calculated here by the following formula

$$\begin{aligned} \bar{C}_{v,i}^{(j)}(k) &= 1/(800|u^{(j-1)}(k) - u^{(j)}(k)| + 1) \\ \tilde{C}_{v,i}^{(j)} &= \bar{F}_{0.083}\bar{C}_{v,i}^{(j)} - 1 \\ C_{v,i}^{(j)}(k) &= \frac{\tilde{C}_{v,i}^{(j)}(k)}{\max_k |\tilde{C}_{v,i}^{(j)}(k)|} + 1 \end{aligned} \tag{3.116}$$

with the second formula in operator format, and $\bar{F}_{0.083}$ a zero-phase low pass filter (with cut frequency 0.083 of the sample frequency). The $C_{v,i}^{(j)}(k)$ values were suitably delayed for application to the columns of $\eta^{(j)}(k)$. $C_{v,i}^{(j)}(k)$ is multiplied with the time-independent stable inversion gain for iteration j, which for the iteration dependent case is designated $c_{\text{SI}}^{(j)}$ (and is governed by the c_{SI} and $c_{\text{lim,SI}}$ values - cf. Table 3.4).

It is found that both the conventional and alternative ILC algorithms are consistently divergent. The best results were achieved with the alternative ILC algorithm using a 50 Hz low pass filter in Q for stable inversion case B2 (cf. Table 3.3), and was much more accurate than the results achieved with the conventional ILC algorithm.

None of the results in Table 3.2 and Table 3.3 represent an improvement over the results achieved in Case 3 with the linear inverse model for the same mathematical system and desired input and output signals. However, when we repeat the conventional and alternative ILC algorithm tests for the case of a 50 Hz low pass filter in Q ***and* a 50 Hz low pass filter** in stable inversion (see the results presented in Table 3.5), we do observe an improvement over the results in Case 3. The best results are consistently obtained with the time-dependent gain approach in stable inversion, with the conventional ILC algorithm giving slightly better results than the alternative algorithm.

Table 3.2: Case 4: Best ILC results with the conventional ILC algorithm. A negative ILC gain c implies an iteration-dependent ILC gain with $c_0 = |c|$ and $c_{\text{lim}} = 0.005$ (cf. Eq. 3.113). A positive c implies an iteration-independent ILC gain. "SI" refers to stable inversion. "It." for u and y are the iteration numbers m corresponding to $\min_m \text{err}_1(u^{(m)})$ and $\min_m \text{err}_1(y^{(m)})$ respectively.

Q-filter	Input - u			Output - y			SI
cut freq [Hz]	c	It.	$\min_m \text{err}_1(u^{(m)})$ [%]	c	It.	$\min_m \text{err}_1(y^{(m)})$ [%]	Case
50	-0.3	48	52.06	-0.3	48	58.31	A1
70	-0.1	156	62.07	-0.1	156	68.77	A1
90	1.0	1	73.68	0.1	12	80.86	A1
None	-0.1	101	72.87	-0.1	101	73.88	A1
50	-0.1	199	50.22	-0.1	199	58.18	A2
70	1.0	1	65.01	-0.1	69	78.64	A2
90	0.05	25	65.08	0.05	25	72.70	A2
None	1.0	1	71.97	-0.1	41	78.68	A2
50	-0.1	411	48.81	-0.1	411	50.31	B1
70	-0.1	129	68.68	-0.1	129	75.03	B1
90	1.0	1	74.31	0.1	11	82.42	B1
None	1.0	1	75.86	-0.1	60	80.42	B1
50	1.0	1	46.26	-0.1	100	57.51	B2
70	1.0	1	47.22	-0.1	70	59.88	B2
90	1.0	1	48.83	-0.1	56	63.79	B2
None	1.0	1	50.52	-0.7	1	67.45	B2

Table 3.3: Case 4: Best ILC results with the alternative ILC algorithm. A negative ILC gain c implies an iteration-dependent ILC gain with $c_0 = |c|$ and $c_{\text{lim}} = 0.005$ (cf. Eq. 3.113). A positive c implies an iteration-independent ILC gain. "SI" refers to stable inversion. "It." for u and y are the iteration numbers m corresponding to $\min_m \text{err}_1(u^{(m)})$ and $\min_m \text{err}_1(y^{(m)})$ respectively.

Q-filter	Input - u			Output - y			SI
cut freq [Hz]	c	It.	$\min_m \text{err}_1(u^{(m)})$ [%]	c	It.	$\min_m \text{err}_1(y^{(m)})$ [%]	Case
50	-0.1	360	63.45	-0.3	118	69.44	A1
70	1.0	1	75.36	1.0	1	81.06	A1
90	-0.1	154	72.16	-0.1	154	78.28	A1
None	-0.7	1	74.86	-0.7	1	81.07	A1
50	-0.1	238	38.69	-0.1	238	48.87	A2
70	-0.1	238	59.72	-0.1	238	66.94	A2
90	-0.1	129	59.66	-0.1	129	68.85	A2
None	0.4	1	66.50	0.4	1	74.69	A2
50	-0.1	383	59.11	-0.1	383	66.13	B1
70	-0.1	246	73.70	-0.7	2	78.98	B1
90	-0.7	2	76.23	-0.1	27	81.07	B1
None	0.1	8	75.17	0.1	8	80.63	B1
50	-0.3	100	21.70	-0.3	100	15.94	B2
70	-0.1	171	34.05	-0.1	171	32.68	B2
90	0.4	1	71.75	0.4	1	78.83	B2
None	1.0	1	50.52	0.05	10	74.63	B2

Table 3.4: Case 4: Stable inversion case details. "SI" refers to stable inversion.

Case	Iteration dependent?	c_{SI}	$c_{\text{lim,SI}}$	Time dependent?
A1	No	0.01	–	No
A2	No	0.03	–	Yes
B1	Yes	0.10	0.005	No
B2	Yes	0.30	0.100	Yes

Table 3.5: Case 4: Best ILC results with the conventional and alternative ILC algorithms when employing a 50 Hz low pass filter in stable inversion. A negative ILC gain c implies an iteration-dependent ILC gain with $c_0 = |c|$ and $c_{\text{lim}} = 0.005$ (cf. Eq. 3.113). A positive c implies an iteration-independent ILC gain. "SI" refers to stable inversion. "It." for u and y are the iteration numbers m corresponding to $\min_m \text{err}_1(u^{(m)})$ and $\min_m \text{err}_1(y^{(m)})$ respectively.

Q-filter cut freq [Hz]	Input - u			Output - y			SI Case	ILC algorithm
	c	It.	$\min_m \text{err}_1(u^{(m)})$ [%]	c	It.	$\min_m \text{err}_1(y^{(m)})$ [%]		
50	1.0	1	15.82	1.0	1	16.78	A1	Conv.
50	1.0	1	13.47	-0.3	753	7.40	A2	Conv.
50	1.0	1	15.64	1.0	1	20.22	B1	Conv.
50	1.0	1	14.47	-0.3	240	9.95	B2	Conv.
50	1.0	1	14.88	-1.0	9	10.37	A1	Alt.
50	1.0	2	14.42	-1.0	9	8.46	A2	Alt.
50	1.0	1	14.78	0.4	4	17.29	B1	Alt.
50	1.0	2	15.50	-1.0	8	8.65	B2	Alt.

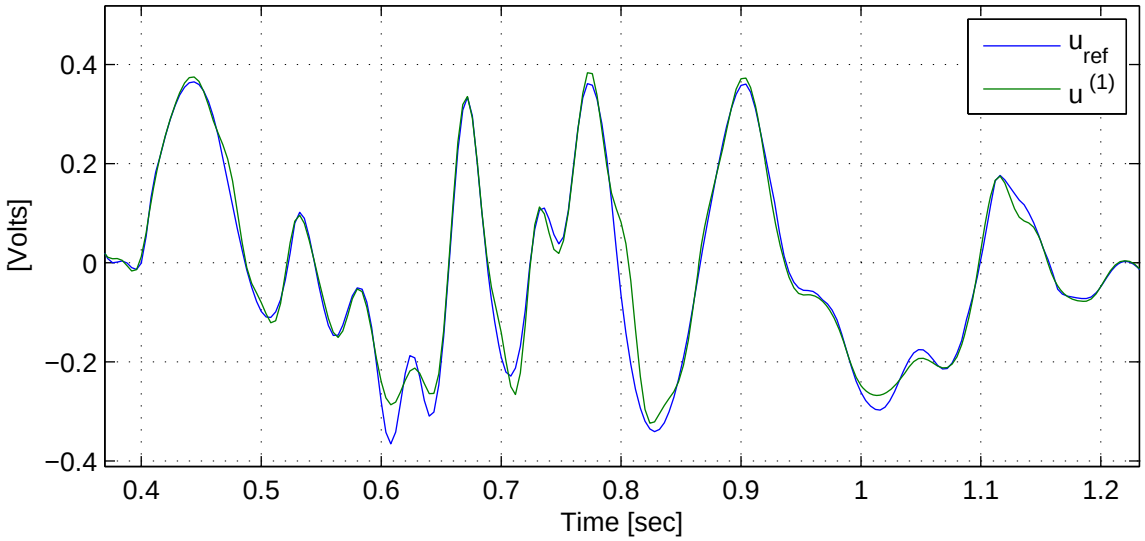

Figure 3.14: Case 4: Best results for ILC input signal $u^{(i)}$.

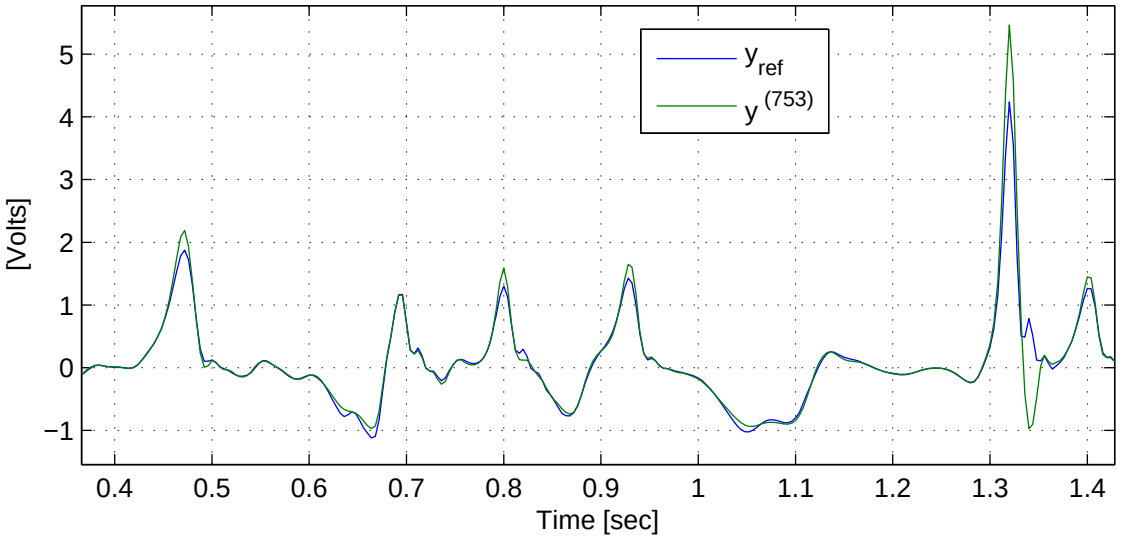

Figure 3.15: Case 4: Best results for ILC output signal $y^{(i)}$.

Chapter 4

Application of ILC in Response Reconstruction

4.1 Introduction

Response reconstruction is a laboratory test method that uses ILC to achieve realistic simulation of field-measured responses in engineering structures, usually for purposes of fatigue testing, but also for other purposes, such as shock testing or vibration testing. When performed on full-scale structures, for example in automotive applications, these are good examples of when we want ILC convergence that is monotone but not too rapid in order to detect dangerous and/or damaging excitation signals before they become too severe. Actuators are applied to the test specimen in such a way as to be able to mimic in-service loading of the structure without excessively impacting the structural dynamic properties of the test specimen. Electro-hydraulic actuators are each typically equipped with a real-time control system to control either actuator load or displacement. The system outputs/responses from the point of view of the response reconstruction procedure are typically strain or acceleration sensors located on the test specimen, where they are appropriately positioned to be able to measure unique specimen responses to unique load inputs. As such the responses may be quite remote from the actuators. The goal of response reconstruction is the accurate reconstruction of desired response histories in the sensors on the test specimen, and is achieved through ILC on the closed-loop test system. The desired responses are usually actual in-service responses measured in the field with the test specimen under normal operation. Replicating the desired responses in the laboratory test specimen by reconstructing them with ILC effectively simulates the normal service conditions under which the desired responses were measured in the field. For more detail regarding the response reconstruction procedure, see sections 1.1 and 1.6.

In the previous chapter (1) the use of nonlinear inverse models of the test system in the general

inverse-based ILC compensator instead of linear inverse models, which is the standard approach, was motivated, and (2) an alternative ILC algorithm was developed for use specifically when employing nonlinear inverse models, but that differs fundamentally from the conventional algorithm. In this chapter we focus on the use of NARX models in particular, obtained by system identification, to source the nonlinear inverse model proposed for use in ILC from. To this end we examine the system identification of NARX models, as well as their inversion, and additionally (3) motivate and develop a multiple-model approach in system identification to improve the accuracy of the nonlinear inverse model in the ILC compensator. Finally we implement all three above-mentioned developments in response reconstruction in a quarter vehicle road simulator case.

4.2 Response Reconstruction Procedure

4.2.1 Test System Definition

With the aim of ILC being the calculation of the system inputs that replicate the desired responses (as close as possible), clearly ILC is essentially an inversion procedure; inverting the entire test system between the control system inputs and the responses on the test specimen for the given desired response signals. The requirement that this experimental inversion procedure of the test system produces a unique input signal for a desired output is the requirement of left invertibility of the test system. This will be satisfied by the same condition as left invertibility of analytical inversion of systems, namely that the system be injective. Amongst others this requires that the test system has as many output channels as input channels (or more output channels). In this research we assume there are the same number of output and input channels (i.e. that the system is "square"). The presentation in this chapter is limited to the SISO case, however, for the sake of brevity and clarity.

The architecture of the real-time control system in discrete-time form is shown in Fig. 4.1. Here $G_s(q)$ represents the generally nonlinear dynamics between the actuator signals sent from the controller to the actuator, and the remote responses measured on the specimen. $G_a(q)$ represents the generally nonlinear dynamics between the same actuator signals and the actuator responses, which are also the feedback signals of the real time control system. $G_c(q)$ represents the (usually) linear controller dynamics.

The nonlinearity of $G_s(q)$ may involve actuators, joints, specimen dynamics, and sensors. A very general model formulation for nonlinear systems is the nonlinear state-space form. We are primarily concerned with the mapping of the closed-loop system between system inputs u and the specimen outputs y which for the i-th iteration may be given in nonlinear state-space form as

$$\begin{aligned} x^{(i)}(k+1) &= f(x^{(i)}(k), u^{(i)}(k)) \\ y^{(i)}(k) &= h(x^{(i)}(k)) \ , \end{aligned} \tag{4.1}$$

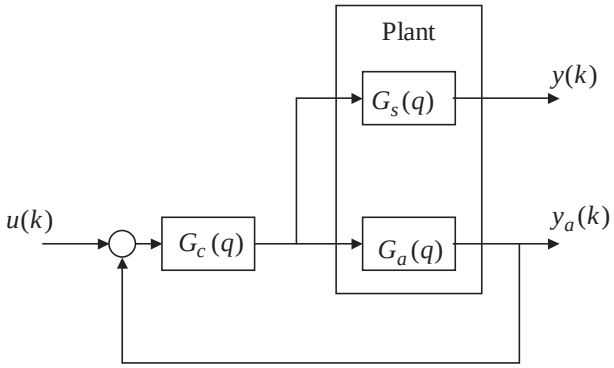

Figure 4.1: Block diagram of the SISO discrete-time control system.

Let T be a nonlinear operator representing the generally nonlinear, closed-loop system between u and y. Thus, for the i-th iteration,

$$y^{(i)} = T(u^{(i)}) \; . \tag{4.2}$$

Assuming the existence of a given desired plant output, $y_d(k)$, it follows there exists a unique input signal $u_d(k)$ corresponding to $y_d(k)$ such that

$$y_d = T(u_d) \; . \tag{4.3}$$

In response reconstruction we use ILC to determine the control system input $u(k)$ that causes the specimen response $y(k)$ to exactly track the desired specimen response, $y_d(k)$, or track it as closely as possible.

4.2.2 System Identification

In past implementations of ILC in response reconstruction the inverse models employed in the ILC compensator have been almost exclusively linear. The two main approaches to obtain a linear model of the closed-loop control system T are non-parametric frequency-domain system identification and parametric time-domain system identification. In most earlier and in many current implementations of ILC in response reconstruction, the non-parametric frequency domain approach is used. Non-parametric models are essentially functions of frequency or time, and are more flexible than parametric models in the sense that no predetermined model structure is imposed (Nelles, 2001). Two basic approaches exist (Ljung, 1999):

- *Fourier transform based identification:* The simplest approach is (for the SISO case) to obtain $\hat{T}$ as:

$$\hat{T}_{(np)}(e^{j\omega}) = \frac{y(\omega)}{u(\omega)} \; ,$$

with $y(\omega)$ and $u(\omega)$ the *discrete Fourier transform* (DFT) of $u(k)$ and $y(k)$ respectively. Ljung (1999) refers to this as the *empirical transfer function estimate* (ETFE). It is called "empirical" because of the "rawness" of the estimate, meaning that the only constraint applied in its calculation is the linearity of the system. Whenever $u(\omega) = 0$ at some frequencies the EFTE is considered as undefined at those frequencies. The use of this approach in ILC is discussed by Longman (2000).

For the general case when the input is non-periodic the ETFE is considered a crude estimate as the estimate at different frequencies are uncorrelated and erratic. Let N be the length of the input signal. While the estimate is asymptotically (as $N \to \infty$) unbiased, the variance doesn't decrease with increasing N. However, in case of a periodic input the ETFE is unbiased and the variance decays proportional to $1/N$.

- *PSD-based identification:* Letting $\phi_u(\omega)$ be the power spectral density (PSD) of the input u, and $\phi_{yu}(\omega)$ be the output-input cross spectral density, for the SISO case $\hat{T}$ may also be obtained as

$$\hat{T}_{(np)}(e^{j\omega}) = \frac{\phi_{yu}(\omega)}{\phi_u(\omega)} \ .$$

For the MIMO case the corresponding formulation is $\hat{T}_{(np)}(e^{j\omega}) = \phi_{yu}(\omega)\phi_u^{-1}(\omega)$. Special care needs to be taken to prevent correlation among the channels of u, which may render $\phi_u(\omega)$ singular (De Cuyper *et al.* 1999). This type of transfer function estimate involves a trade-off between bias and variance based on the choice of windowing parameters, as result of which general statements about the dependence of variance on N alone are not possible. Note that in case of a closed-loop control system this type of transfer function estimate will give erroneous results if applied in the straight-forward way.

If the model is intended to be used as an inverse-model based ILC compensator, then for the desirable properties of the ILC algorithm associated with such a compensator to be achieved, the model needs to be as accurate as possible from the point of view of achieving desirable transient behavior of ILC and maximising the width of the convergent frequency band in ILC. In case of nonlinear systems this means a nonlinear model is sought. Nonlinear models have in recent years been developed as extensions of parametric time domain models with both their formulation and identification relatively straight-forward in many cases. For this reason the focus in this research is on parametric time domain models, specifically with regard to nonlinear extensions of the approach, and their use in ILC for response reconstruction testing purposes. The identification of parametric time domain models is discussed in Section 4.3 for the linear case and in Section 4.4 for the nonlinear case.

The remainder of the algorithm focuses on the design of elements required in the implementation of the ILC procedure, including the nonlinear inverse model $\tilde{L}$, the scalar c, and the filters C and Q.

4.2.3 Design of $\tilde{L}$ (inversion of $\hat{T}$)

The system model $\hat{T}$ is left inverted to obtain the ILC compensator as $\tilde{L} = \hat{T}^{-1}$. In the case of a linear MIMO non-parametric model $\hat{T}_{(np)}(e^{j\omega})$ (i.e. an FRF) the model can simply be inverted via matrix inversion at every frequency for which the model is defined, assuming the matrix is full rank. Inversion of parametric (time domain) models $\hat{T}_{(p)}$ in the general case of non-minimum phase systems is less straight-forward, and requires stable inversion, i.e. the use of the dichotomic partitioning of the dynamics of the inverse into the stable and unstable part.

4.2.4 Design of C

The SISO zero-phase filter C is designed to achieve local adjustments (i.e. focused on specific frequency bands) in the convergence criterion of ILC (see for example Eq. 3.48) to locally (where needed) increase robustness against model uncertainty, typically at high frequencies.

4.2.5 Choice of c

The real scalar c is selected to set the rate of convergence (cf. Eq. 3.49). When the inverse model in $\tilde{L}$ is exact, c will be exactly the rate of convergence; but when the inverse is approximate, c will approximate the rate of convergence. c can also be used separate from C as a frequency-independent scale factor to achieve frequency-independent adjustments in the convergence criterion of ILC (cf. Eq. 3.48) to increase robustness of convergence against model uncertainty, but at the cost of a global reduction in the rate of convergence.

4.2.6 Design of Q

The SISO zero-phase filter Q is designed as a low pass filter ideally with cut frequency to limit the ILC test bandwidth to the range of frequencies over which convergence occurs. Noting the role of Q in Theorem 3.3 clearly Q plays a major role in achieving convergence, but at the expense of inducing bias in the limit signals (c.f. Theorem 3.2). In practice in fatigue testing the need is usually to do response reconstruction on a bandwidth wider than that over which convergence occurs, and Q is usually selected as a low-pass filter with cut frequency being the required upper test frequency band rather than the convergent frequency band of ILC.

4.2.7 ILC

Either the conventional or alternative ILC algorithms may be employed in response reconstruction, with the conventional algorithm given as

$$\begin{aligned} y^{(i)} &= T(u^{(i)}) \,, \\ u^{(i+1)} &= Q(u^{(i)} + L(y_d) - L(y^{(i)})) \,, \end{aligned} \tag{4.4}$$

with $L = cC\tilde{L}$, which is able to achieve convergence in one step when $\tilde{L} = \hat{T}^{-1}$. When using a very accurate inverse model $\tilde{L}$ in the ILC compensator L, this algorithm provides precise control of the rate of convergence of the input, $u^{(i)}$, with monotone convergence of the input, and convergence over the entire frequency band up to the Nyquist frequency. Due to the desirability of these properties, and the fact that the more accurate the inverse model is, the more these properties will be achieved, when an accurate inverse model is not available we still aim for the most accurate inverse possible. For this reason there is much merit in future research into successful identification and inversion of more general nonlinear models than currently being implemented in response reconstruction.

The alternative ILC algorithm as presented Section 3.4 is given by

$$\begin{aligned} u^{(i)} &= L(\tilde{y}^{(i)}) \,, \\ y^{(i)} &= T(u^{(i)}) \,, \\ \tilde{y}^{(i+1)} &= Q(\tilde{y}^{(i)} + y_d - y^{(i)}) \,. \end{aligned}$$

with with $L = cC\tilde{L}$.

The modified alternative ILC algorithm as presented Section 3.5 is given by

$$\begin{aligned} u^{(i)} &= \tilde{L}(\tilde{y}^{(i)}) \,, \\ y^{(i)} &= T(u^{(i)}) \,, \\ \tilde{y}^{(i+1)} &= Q(\tilde{y}^{(i)} + cC(y_d - y^{(i)})) \,. \end{aligned}$$

When using an accurate inverse in the ILC compensator ($\tilde{L}$), this algorithm again provides precise control of the rate of convergence of $\tilde{y}^{(i)}$, with monotone convergence of $\tilde{y}^{(i)}$, and convergence over the entire frequency band up to the Nyquist frequency. The more accurate the inverse model is, the more accurately these properties will be realized in ILC.

For the purposes of response reconstruction learning is allowed to continue until either convergence has occurred, or an optimum value of the norm of the tracking error has been reached, after which the $u^{(i)}$ and corresponding $y^{(i)}$ for the final iteration is saved, and ILC is terminated. In case of ILC for fatigue testing purposes the rest of the fatigue test consist of merely replaying the final

achieved $u^{(i)}$ to achieve the desired service load reconstruction that was the purpose for which ILC was employed in the first place. Fatigue analysis may be done on the final achieved $y^{(i)}$ in order to evaluate the achieved test acceleration compared to the desired test acceleration and determine the required test duration to a simulate a given normal-usage life for the structure.

4.2.8 Discussion

When a relatively accurate model of the real-time test system is available and the ILC compensator is a simple, non-model based design, the ILC convergence criterion can be evaluated to determine the approximate frequency range for convergence *a priori*. In case of inverse model-based ILC, the most accurate approach is to employ the inverse of the most accurate available model in the ILC compensator, as result of which the ILC convergence criterion will automatically predict convergence up to the Nyquist frequency and will not be of value in evaluating convergence of ILC with the inverse model (due to the inverse model being the exact inverse of the model used to represent the system in the convergence criterion). In this study we do not try to salvage the convergence criterion for inverse-based ILC and thus discard attempts at quantitative evaluation of the convergence criterion, instead using it as a qualitative guide.

The design of $\tilde{L}$, C, c, and Q as outlined above follows largely after Norrlöf and Gunnarsson (2002b), with the difference that here C is aimed purely at robustness of convergence of ILC, with the rate of convergence adjusted with c instead, while in Norrlöf and Gunnarsson (2002b) a single filter $(1 - H)$ handles both robustness of convergence and rate of convergence. Thus, by generalising from the linear case, at frequencies that we have high model certainty of $\tilde{L} = \hat{T}^{-1}$, there is little need to scale C down to improve the robustness of convergence, and we will let $C = 1$ at these frequencies. Afterwards, having shaped C aimed at robustness of convergence only, c is then chosen to represent the rate of convergence, keeping in mind that it will also contribute to robustness of convergence, but in a frequency insensitive way. This approach incorporates both the $(1 - H)$ filter of Norrlöf and Gunnarsson (2002b) in C, and the constant, scalar iteration gain in c as used in traditional fatigue testing implementations of ILC, and thus represents a slightly more general framework for ILC than either of the approaches.

Designing C *a priori* may be difficult (either for lack of an accurate model of T or the difficulty of analyzing the convergence criterion), as result of which in the fatigue testing field frequently $C = 1$ is used and c is set to a rather low value, such as 0.1 - 0.2, to maximise the chances of convergence over the desired frequency band and reduce the rate of convergence. As a consequence it is a frequent outcome to, instead of having a given bandwidth over which convergence occurs and have the test limited to that bandwidth (using the Q filter), find that either the high frequencies or another frequency band diverges within the desired test bandwidth and for iterative learning to be stopped at some optimum value of a norm of the achieved tracking error before the divergent frequency band results

in increase of the tracking error. (This is also the approach followed in the demonstration example, Section 4.7). Future work in the employment of ILC in response reconstruction may benefit from use of the C filter, especially to curb the early onset of divergence at high frequencies due to the typical inaccuracy of model estimates $\hat{T}$ at these frequencies. In ILC on nonlinear systems the potential is limited however, as even when not inputting any energy at higher frequencies, nonlinear effects such as free-play in joints or frequent contact between components of the specimen may result in strong response at higher frequencies and ultimately cause divergence at higher frequencies.

4.3 Linear Parametric System Identification in the Time Domain

4.3.1 The ARX Model

System identification is performed to obtain a parametric time-domain model which may subsequently be inverted to give an inverse system model for use in ILC. During system identification the entire test system is modeled, including digital to analogue (D-A) conversion, actuator control system, hydraulic delivery system, actuators, specimen, response measurement system, and analogue to digital (A-D) conversion. The approach taken here is to use the input-output formulation, also called external dynamics formulation, which includes the well-known ARX model. The alternative approach is the state-space formulation, also called the internal dynamics formulation. In a non-parametric sense the test system with input signal $u(k)$ and output $y(k)$ may be modeled as a linear time-invariant system in input-output format as (Nelles (2001), Ljung (1999)):

$$y(k) = G(q)u(k) \ , \tag{4.5}$$

with

$$G(q) = \sum_{i=0}^{\infty} g_i q^{-i} \ , \tag{4.6}$$

$g_i \in \mathcal{R}$, and q the backward shift operator, i.e. $q^{-n}u(k) = u(k-n)$. Eq. 4.5 represents the discrete-time version of the convolution integral using the impulse response when employing a *zero order hold* (ZOH) during sampling of the input. The sequence $\{g(k)\}$ represents the discrete-time version of the impulse response. This deterministic model may be generalized by the addition of a stochastic noise component $v(k)$, with

$$v(k) = H(q)e(k) \ , \tag{4.7}$$

to obtain a basic model formulation

$$y(k) = G(q)u(k) + H(q)e(k) \ , \tag{4.8}$$

in which $e(k)$ is white noise and $H(q)$ a linear filter, allowing any noise spectrum for $v(k)$ to be modeled. When the deterministic part of the model ($G(q)u(k)$) is discarded, the model is referred to

as a *time series*. When $H(q)$ consists only of a numerator it is referred to as a *moving average* (MA) model, and when it consists only of a denominator it is referred to as an *autoregressive* (AR) model. When it has both it is referred to as an ARMA model. Adding an exogenous input $u(k)$ to the AR time series model gives the so-called ARX model (auto-regressive model with exogenous input):

$$y(k) = \frac{B(q)}{A(q)}\,u(k) + \frac{1}{A(q)}\,e(k)\ , \tag{4.9}$$

with

$$A(q) \;=\; 1 + \sum_{i=1}^{n_a} a_i q^{-i} \in \Re[q]\ , \tag{4.10}$$

$$B(q) \;=\; \sum_{i=0}^{n_b} b_i q^{-i} \in \Re[q]\ . \tag{4.11}$$

Comparison with Eq. 4.8 shows that $G(q) = B(q)/A(q)$ and $H(q) = 1/A(q)$. The ARX model is an example of an *equation error*, referring to the fact that we may write

$$A(q)y(k) = B(q)\,u(k) + e(k)\ ,$$

where e enters the equation rather than being added to the output, and implying that the denominator is common to both the deterministic and stochastic parts. When the noise model (i.e. $H(q)$) is independent of the dynamics of the deterministic part (i.e. $G(q)$), the model is of the *output error* class of models. The most basic of these is itself called the *output error* (OE) model, and is given as

$$y(k) = \frac{B(q)}{F(q)}\,u(k) + e(k)\ , \tag{4.12}$$

where e is added to the output. When $F(q) = 1$ we get the *finite impulse response* (FIR) model

$$y(k) = B(q)u(k) + e(k)\ . \tag{4.13}$$

When $G(q)$ represents the impulse response, as in Eq. 4.6, it is essentially non-parametric (with an infinite number of elements); however in the ARX model $G(q)$ has been parameterized as a rational function with a finite number of parameters. The ARX and related models thus represent *parametric* models.

4.3.2 Model Simulation

Predicting the unknown output for a given input using the model is referred to as *model simulation*. Since the noise term is unknown beforehand the model is used in the deterministic sense only in simulation. In this case the ARX model becomes

$$\hat{y}(k) = \frac{B(q)}{A(q)}u(k)\ , \tag{4.14}$$

where $\hat{y}(k)$ is the simulated model output. Expanding $A(q)$ and $B(q)$ in Eq. 4.14 and multiplying with the denominator gives

$$\begin{aligned} &\hat{y}(k) + a_1\hat{y}(k-1) + \ldots + a_{n_a}\hat{y}(k-n_a) = \\ &b_0 u(k) + b_1 u(k-1) + \ldots + b_{n_b} u(k-n_b) \; . \end{aligned} \tag{4.15}$$

Solving for the current $\hat{y}(k)$ in terms of u and past values of $\hat{y}$ gives

$$\hat{y}(k) = B(q)u(k) + (1 - A(q))\hat{y}(k) \; . \tag{4.16}$$

4.3.3 Prediction and Estimation

Prediction is when we use the model to predict the output at a given time k for the given input signal and past *observed* outputs (i.e. actual instead of predicted past outputs). It can be shown the optimal predictor (resulting in the lowest prediction error) for the basic model formulation (cf. Eq. 4.8) is given by (Ljung, 1999)

$$\hat{y}(k|k-1) = \frac{G(q)}{H(k)}\, u(k) + \left(1 - \frac{1}{H(q)}\right) y(k) \; , \tag{4.17}$$

which, for the ARX model, becomes

$$\begin{aligned} \hat{y}(k|k-1) &= B(q)u(k) + (1 - A(q))y(k) && (4.18) \\ &= \theta^{\mathrm{T}}\varphi(k) \; , && (4.19) \end{aligned}$$

with

$$\begin{aligned} \theta &= [a_1 \; \ldots \; a_{n_a} \; b_0 \; b_1 \; \ldots \; b_{n_b}]^{\mathrm{T}} && (4.20) \\ \varphi(k) &= [-y(k-1) \; \ldots \; -y(k-n_a) \; u(k) \; u(k-1) \; \ldots \; u(k-n_b)]^{\mathrm{T}} \; . && (4.21) \end{aligned}$$

The θ vector represents the parameters of the model, and determining the θ vector is the purpose of the system identification exercise. Such a model, consisting of a product between a known data vector $\varphi(k)$ and an unknown parameter vector θ is called a *linear regression*, with $\varphi(k)$ called the regression vector. For a given parameter estimate θ the *prediction error* is defined as (using Eq. 4.17)

$$\begin{aligned} \varepsilon(k) &:= y(k) - \hat{y}(k|k-1) && (4.22) \\ &= \frac{1}{H(q)}\, y(k) - \frac{G(q)}{H(k)}\, u(k) && (4.23) \\ &= y(k) - \theta^{\mathrm{T}}\varphi(k) \; . && (4.24) \end{aligned}$$

Eq. 4.24 may be developed for $k = 0, \ldots, N-1$ and grouped into matrix form as

$$\varepsilon_{[0,N-1]} = Y - \Phi\, \theta \; , \tag{4.25}$$

with $\varepsilon_{[0,N-1]} = [\varepsilon(0), \ldots, \varepsilon(N-1)]^{\mathrm{T}}$, $Y = y_{[0,N-1]} = [y(0), \ldots, y(N-1)]^{\mathrm{T}}$ and $\Phi = [\varphi(0), \ldots, \varphi(N-1)]^{\mathrm{T}}$, called the regression matrix. When selecting a scalar-valued cost function V of the prediction error $\varepsilon_{[0,N-1]}$, the *ideal* parameter estimate $\hat{\theta}$ may be defined as the one that minimizes this cost function, i.e.

$$\begin{aligned} \hat{\theta} &= \arg\min_{\theta} V(\varepsilon_{[0,N-1]}) \\ &= \arg\min_{\theta} V(u, y, \theta) \ . \end{aligned} \tag{4.26}$$

Choosing a quadratic cost function for V, namely

$$V = \sum_{k=0}^{N-1} \varepsilon^2(k) \ , \tag{4.27}$$

enables the optimal estimate to be determined by linear optimization methods, specifically the least squares method. This allows the analytic solution of the best parameter estimate $\hat{\theta}$ to be determined from

$$\Phi^{\mathrm{T}} \, \Phi \, \hat{\theta} \ = \Phi^{\mathrm{T}} Y \ . \tag{4.28}$$

4.3.4 Inconsistency of the ARX Model

A set of identification data with length $N \to \infty$ is said to be *informative enough* with respect to a given model structure if it can distinguish between different models in the model structure, i.e. if when two models have the same one-step-ahead prediction for the data set, they must have the same transfer function. If a model structure (such as the ARX model structure) is flexible enough to include the true test system and the identification data is informative enough with respect to the model structure, then the true system will be found by the limit estimate, θ^*, which is defined as the estimate that is obtained when the data length $N \to \infty$, and say that the estimate is *consistent*. A problem of the ARX model structure is that the $1/A(q)$ noise model of the ARX model is not very realistic, with additive disturbances, as modeled in output error models, being much more common. This results in the parameters estimated for the ARX model being biased and *inconsistent* whenever the real noise mechanism doesn't conform to the ARX model's very special noise model (Nelles, 2001). *Inconsistency* refers to that fact that the error persists even for infinite data length. A few strategies to circumvent this problem are as follows:

- The *instrumental variable* identification procedure may be used. This is a multi-step identification method that comes at the expense of added effort (Ljung, 1999).
- Use a more general model structure than the ARX model, such as the OE model. With model structures where the transfer function model $G(q)$ and noise model $H(q)$ are distinct (i.e. independently parameterized), such as the OE model, consistent identification is achieved even

when the noise model structure cannot exactly model the process noise dynamics. The downside is that parameter identification methods are also more complex for OE models, and that the unconditional consistency applies only for open-loop systems (which is not usually the case in response reconstruction).

- A prefilter strategy can be used with the ARX model and least squares identification in an iterative manner to identify what is effectively an OE model, with the associated consistency advantages of the OE model, but with the convenience of the ARX model and least squares identification (for more detail see the discussion of prefiltering in Section 4.6).

In response reconstruction, however, we will usually settle for use of the ARX model identified in the straight-forward manner with least squares despite its inherent bias and inconsistency, primarily for the sake of its convenience.

4.3.5 Characterization of Model Fit

If the true system is not included in the model structure used in system identification, then the parameter estimate θ_N is inconsistent, no matter how large N is, and must necessarily differ from the true system, in other words, be biased, even for the limit estimate. We now examine the frequency domain representation of the quadratic loss function for the limit estimate for the case of inconsistent estimates (Ljung, 1999). Noting that for the prediction error (cf. Eq. 4.7)

$$\varepsilon(k) = \frac{1}{\hat{H}(q,\theta)}(y(k) - \hat{G}(q,\theta)u(k)) \tag{4.29}$$

$$= \frac{1}{\hat{H}(q,\theta)}(G_0(q)u(k) + v(k) - \hat{G}(q,\theta)u(k)) \tag{4.30}$$

$$= \frac{1}{\hat{H}(q,\theta)}((G_0(q) - \hat{G}(q,\theta))u(k) + v(k)) \ , \tag{4.31}$$

with $\hat{G}$ representing the model of the physical system G_0, the loss function in terms of the PSD of the prediction error Φ_ε becomes

$$E\{I(\theta)\} = \frac{1}{4\pi}\int_{-\pi}^{\pi} \Phi_\varepsilon(\omega,\theta)d\omega \tag{4.32}$$

$$= \frac{1}{4\pi}\int_{-\pi}^{\pi} \left(|G_0(e^{j\omega}) - \hat{G}(e^{j\omega},\theta)|^2 \ \Phi_u(\omega) + \Phi_v(\omega)\right) \frac{1}{|\hat{H}(e^{j\omega},\theta)|^2}\, d\omega \ , \tag{4.33}$$

with Φ_u and Φ_v the PSDs of the input and noise signals respectively, $E\{\cdot\}$ representing mathematical expectation. If the noise term can be neglected Eq. 4.33 reduces to

$$E\{I(\theta)\} \approx \frac{1}{4\pi}\int_{-\pi}^{\pi} |G_0(e^{j\omega}) - \hat{G}(e^{j\omega},\theta)|^2 Q(\omega)d\omega \tag{4.34}$$

$$Q(\omega) = \frac{\Phi_u(\omega)}{|\hat{H}(e^{j\omega},\theta)|^2}\, d\omega \ . \tag{4.35}$$

Interpreting $Q(\omega)$ as a frequency dependent weighting factor it is observed that the model fit is optimized at frequencies where (1) $\Phi_u(\omega)$ is large, or (2) the noise model's frequency response is small. The former mechanism opens up the possibility of influencing the model fit by the shape of the input signal's spectrum, specifically optimizing it in frequencies at which the model is intended to be used. The second mechanism shows that the choice of noise model also influences the model fit. In case of the ARX model the noise model $\hat{H}(q) = 1/A(q)$ typically has a low pass characteristic, implying that $1/|\hat{H}(e^{j\omega}, \theta)|^2$ has a high pass characteristic, thus accentuating the high frequency end of the model fit and implying that the ARX model's fit is biased towards the high frequency end. By accentuating low frequencies in the input spectrum we may counter the natural bias towards high frequencies of the ARX model fit. Note that in case of OE models, with a unity noise model, the fit is consequently weighted only by the input spectrum.

4.3.6 Model Variance

Consider a parameter vector estimate, $\theta(\mathcal{D})$, for a model that is obtained from system identification on a set of input and output data (hereafter referred to as *identification* data) for a given test system. $\mathcal{D}$ contains all choices (variables) associated with the identification data including, amongst others, the length of the data, N, the order (i.e. 'size') of the model, n, and the variance of the noise input signal to the noise model λ_0. $\theta(\mathcal{D})$ defines a model $\hat{T} = [\hat{G}\ \hat{H}]$ of the true system $T = [G\ H]$ (cf. Eq. 4.8). Defining a quadratic norm $J(\mathcal{D})$ on the model error $\tilde{T} = \hat{T}(\mathcal{D}) - T$, it can be shown that

$$J(\mathcal{D}) = J_P(\mathcal{D}) + J_B(\mathcal{D}) \ ,$$

where $J_P(\mathcal{D})$ is due to the inherent variance of the parameter estimate, and $J_B(\mathcal{D})$ is due to the bias of the model estimate. If the model is consistent then the bias error becomes negligible and $J(\mathcal{D})$ is dominated by the variance error $J_P(\mathcal{D})$. Ljung (1999) gives an asymptotic (as $N \to \infty$) formula for the variance error $J_P(\mathcal{D})$. From this formula it is observed for the variance error associated with the system dynamics $G(q)$, *per se*, or for the case of purely additive output noise (i.e. $H(q) = 1$), that

$$J(\mathcal{D}) \approx J_P(\mathcal{D}) \propto \lambda_0 \frac{n}{N} \tag{4.36}$$

for large n (i.e. $n \to \infty$). Thus, the variance of the parameter estimate can be reduced by reducing the noise variance, reducing the model size, or increasing the duration of the identification data. These guidelines are still valid even when $H(q) \neq 1$. In the frequency domain, for $\hat{G}$ in particular, we have

$$\text{Cov}\ \hat{G}(e^{j\omega}) \approx \frac{n}{N} \frac{\phi_v(\omega)}{\phi_u(\omega)} \tag{4.37}$$

with ϕ_u and ϕ_v the spectra (i.e. PSDs) of u and v respectively, and Cov $\hat{G}(e^{j\omega})$ defined as

$$\text{Cov}\ \hat{G}(e^{j\omega}) = E\ |\hat{G}(e^{j\omega}) - E\ \hat{G}(e^{j\omega})|^2 \tag{4.38}$$

Another factor that impacts the variance of $\theta(\mathcal{D})$ is the sample frequency, f_s, of the identification data. For the signals to be as informative as possible, the sample frequency should be high enough – at least twice the highest frequency content of the data deemed the interesting part of the process. However, the short sample period that results from a high sample frequency translates into an increase in the variance of the estimate if it is much shorter than the dominant time constants of the system. Conversely, when the sample period exceeds the dominant time constants of the system, the variance drastically increases. The minimum occurs when the sample period roughly equals the dominant time constants of the system. Too fast sampling is therefore better than too slow sampling. This, together with the fact that the dominant time constants are often not known a priori, often leads to the recommendation to sample at approximately $10\times$ the bandwidth of the system. Care should be taken though, as too fast sampling is directly linked to two mechanisms resulting in *inaccuracy in the low frequency regime* of the estimated model, namely (1) model numerical problems (cf. Section 4.6) and (2) model bias towards high frequencies in case of ARX models (cf. Section 4.3.5).

4.3.7 Choice of Excitation

The analysis of the bias distribution in Section 4.3.5 shows that model quality will be better in the frequencies where the power in the input signal's spectrum is greater. Thus, when we expect the model bias to be significant, such as is typically the case for the ARX model's noise model or when modelling a nonlinear system with a linear model, the input spectrum should ideally be tailored in view of the intended frequency range for which the model is to be applied (or for nonlinear systems operate at the same nominal operating point as in practice). Since parameter variance is also reduced by increasing the input signal power and reducing the noise power, cf. Eq. 4.37, a good choice of input signal amplitude range will have a good (i.e. high) signal to noise ratio at frequencies of interest. In case of nonlinear systems it is often found that lower amplitude input signals may result in better model fits to be achieved, at the expense of the model representativity being aimed at this operating range.

A number of possible designs may be used for the input signal, such as steps, rectangular waves, pseudo random binary signals, or filtered white noise (Nelles (2001) and Ljung (1999)). The use of filtered white noise is recommended because it allows input signals to be generated to any desired spectrum.

4.3.8 Identification of Closed-Loop Systems

Consider a typical feedback control system as shown in Fig. 4.2. Note that the G_c-G_a feedback loop of the real time system involved in response reconstruction (cf. Fig. 4.1) may be expressed in a form very similar to Fig. 4.2, with the remaining G_s part an open-loop system in series with the closed-

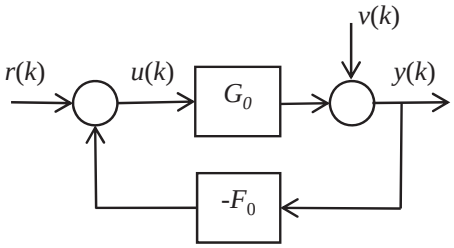

Figure 4.2: Closed loop control system in system identification.

loop system (thus representing no new conceptual complication). The main complication when trying to do system identification of the plant $G_0(q)$ is the fact that the reference input $r(k)$ is correlated with the noise $v(k)$ due to the presence of the feedback loop. Two approaches exist (Ljung, 1999) to do system identification of the plant $G_0(q)$, namely the *direct* and *indirect* approach. In direct identification the plant $G_0(q)$ is identified from $u(k)$ and $y(k)$ as if it is open loop (thus ignoring the effects of the feedback loop). In indirect identification the closed-loop system between $r(k)$ and $y(k)$ is identified, and the plant estimate $\hat{G}(q)$ is subsequently inferred from the closed-loop system. The complete closed-loop system is given by

$$y(k) = T_0(q)r(k) + v_{cl}(k) \tag{4.39}$$

$$= \frac{G_0(q)}{1 + G_0(q)F_0(q)}r(k) + \frac{1}{1 + G_0(q)F_0(q)}v(k) \tag{4.40}$$

Thus, in indirect identification the closed loop system $T_0(q)$ between $r(k)$ and $y(k)$ is identified, resulting in the estimate $\hat{T}(q)$. For this purpose we will typically use an ARX model in the linear case. Since in response reconstruction we seek a model of the entire test system to use in the ILC procedure, the indirect approach is therefore the approach we employ, with no further subsequent effort to obtain $\hat{G}(q)$ from the estimated $\hat{T}(q)$. With regard to the variance and bias of the estimate obtained with the indirect approach, we have the following results: The asymptotic expression (i.e. for large N) of the variance of $\hat{T}(q)$ is

$$\text{Cov } \hat{T}(e^{j\omega}) = \frac{n}{N}\frac{\phi_{v,cl}(\omega)}{\phi_r(\omega)}\,, \tag{4.41}$$

with n and N as before and ϕ_r and $\phi_{v,cl}$ the spectra of r and v_{cl} respectively. Furthermore, if the model is flexible enough to represent the true system $T_0(q)$, the estimate will be consistent. This is usually not the case with the noise model of the ARX model, however, in view of the relatively low noise levels of laboratory test systems the use of the ARX model in the linear case for response reconstruction purposes will generally be acceptable.

4.3.9 Handling of Offsets

Input and output signals often have non-zero offsets, or equilibrium points, whereas in system identification the need is for input and output data in terms of deviations from the equilibrium point, i.e. around a zero offset. Various approaches exist to accommodate this situation, with the one adopted here to explicitly remove the non-zero offsets of input and output data by preprocessing. Whenever the model is to be used for simulation the offset values are afterwards added again to the input and calculated output signal. In order to easier and more accurately estimate the offsets the approach followed here is to precede the input signal during testing with a period during which the input signal offset is held constant, and follow the input signal with another such period. This allows the output signal offset to be explicitly measured during these pre - and post "dead times", and to calculate trends in the offset over the duration of the test.

4.4 Nonlinear Parametric System Modeling: The NARX Model

4.4.1 Input-Output Models with Feedback

We now focus on the identification of nonlinear parametric models of the nonlinear test system in response reconstruction. A very general nonlinear system description is given by the nonlinear state space model (cf. Eq. 4.1). The use of nonlinear state space (internal dynamics) models, though desirable due to their lower model order than external dynamics models, is considerably complicated by the need to estimate not only the model parameters during system identification, but the internal states also, leading to a general preference for the simpler input-output (external dynamics) models. Nonlinear input-output models are capable of representing a wide range of systems (Leontaritis & Billings, 1985), but are not as general as nonlinear state space models, facing limitations with non-unique nonlinearities (such as backlash and hysteresis) and non-invertible nonlinearities (Nelles, 2001:562). Under assumption that the system is locally linearizable around the equilibrium point, it may be represented by the nonlinear input-output model structure (Chen & Billings, 1989a), which in prediction format is given as

$$\hat{y}(k|k-1) = F(\varphi(k)) \ . \tag{4.42}$$

For the so-called NARX (Nonlinear ARX) model structure $\varphi(k)$ is given as:

$$\varphi(k) = [u(k-1) \ \ldots \ u(k-n_b) \ y(k-1) \ \ldots \ y(k-n_a)]^{\mathrm{T}} \ . \tag{4.43}$$

The NARX model parameters are estimated with the model in predictor format (as here). When the model is trained in simulation rather than in prediction, we get the NOE (nonlinear output error) model, which is therefore given by

$$\varphi(k) = [u(k-1) \ \ldots \ u(k-n_b) \ \hat{y}(k-1) \ \ldots \ \hat{y}(k-n_a)]^{\mathrm{T}} \ . \tag{4.44}$$

The NARX and NOE models are examples of a nonlinear model with output feedback. (We do not discuss models without output feedback here.) The disadvantages of models with output feedback are (Nelles, 2001:558):

- The model orders n_a and n_b are difficult to estimate in an efficient way, resulting in trial and error being the usual approach for their determination.
- Model stability cannot generally be proven, and is instead demonstrated using extensive simulations with the model.

One advantage of output-feedback models is that they are relatively compact compared to non-output feedback models (for example NFIR models). This is an important advantage in case of MIMO systems, whose size grows strongly out of proportion with the length of $\varphi(k)$ (the dimension of the input space), which is referred to as the "curse of dimensionality".

4.4.2 NARX Model Issues

The major advantage of the NARX model is that when it is formulated in terms of polynomial functions (see below) the model parameters are easy to identify using linear optimization methods such as least squares because it is then linear in the parameters. Some disadvantages for the NARX model in particular are similar to the ARX model, namely (Nelles, 2001:559)

- Numerical problems during identification in case of high sample frequency (discussed in more detail in Section 4.6.1);
- Unrealistic noise model, leading to inconsistent model structure and thus biased parameter estimates (cf. Section 4.3.4); and
- Emphasis of the model fit on higher frequencies due to the low pass characteristic of the noise model (cf. Section 4.3.5).

In response reconstruction the need is to use the model in simulation rather than in the (one-step-ahead) prediction format. NOE models may potentially be more accurate than NARX models when the intended usage is simulation (which is the usage in which the NOE model is identified), rather than one-step-ahead prediction (which is the usage in which the NARX model is identified). Furthermore, because NOE models are identified in simulation they will reveal model instability during identification already, whereas the NARX may only reveal it during simulation and not during identification (because a one-step-ahead predictor model may be stable whereas the same model in simulation may be unstable). This problem is more serious when the model is used to extrapolate in regions that were not covered in the identification input data because these regions are more prone

to instability. The disadvantage is that NOE models require more complex (nonlinear) identification procedures.

4.4.3 Design of Excitation Signal

The choice of excitation signal for use in identification of nonlinear models is much more crucial than for linear models. Because these models are much more complex than linear models, their identification data must contain much more information. In response reconstruction the test system will generally be available for experimentation, and may generally be excited over a large range of its operating regime if needed. This is indeed necessary to avoid the model being employed in extrapolation (i.e. when the model is employed in a dynamic regime that did not feature in the data the model was estimated with) for some operating conditions during subsequent use, which carries a high risk of model inaccuracy or instability. In view of the fact that the model size of nonlinear models is orders of magnitudes larger than linear models, as shown in Section 4.4.4, it follows that the identification data need to be as long as practically possible in order to capture as much information of the process as possible, and thereby minimise the model parameter estimate variance. The recommended approach for the design of excitation data for nonlinear systems therefore, as with linear systems, is to generate random data that conforms to a prescribed PSD, either by filtering of white noise or some other generating method. The PSD may be designed to match the excitation aspects expected to dominate in the generation of field responses of the structure, and/or may try to compensate for expected bias of the NARX model towards higher frequencies by accentuating low frequency components in the excitation. In summary, the need is for much longer data than for linear systems, that is varying over the widest range of operating regimes possible, and with a good spread of energy over the whole frequency range of the test system.

4.4.4 Polynomial NARX Model Formulation

The general NARX structure of Eq. 4.42 and Eq. 4.43 may be approximated by implementing a polynomial formulation, leading to the one-step-ahead predictor becoming:

$$\begin{aligned}\hat{y}(k|k-1) &= \sum_{j_1=1}^{n_z} \theta_{j_1}\varphi_{j_1}(k) \\ &+ \sum_{j_1=1}^{n_z}\sum_{j_2=j_1}^{n_z} \theta_{j_1 j_2}\varphi_{j_1}(k)\varphi_{j_2}(k)\dots \\ &+ \sum_{j_1=1}^{n_z}\dots\sum_{j_d=j_{d-1}}^{n_z} \theta_{j_1\dots j_d}\varphi_{j_1}(k)\dots\varphi_{j_d}(k)\ ,\end{aligned} \tag{4.45}$$

with $\varphi(k)$ as in Eq. 4.43, and d the degree of nonlinearity. Model structures with output feedback (such as the NARX model) that are formulated in this way are sometimes referred to as *Kolmogorov-Gabor polynomial models*. The primary advantage of the polynomial NARX model formulation, as already mentioned, is that the model is linear in the parameters: If the parameters and mononomial

regressors are grouped into separate column vectors θ and $\tilde{\varphi}(k)$ respectively then the prediction error becomes a linear regression:

$$\varepsilon(k) = y(k) - \theta^{\mathrm{T}}\tilde{\varphi}(k) \ . \tag{4.46}$$

Therefore the parameters may be estimated with linear optimization using least squares. This is an important advantage from the point of view of speed of calculation of the parameters of the polynomial NARX model formulation over other nonlinear model structures, and over other NARX model formulations. Some disadvantages of the polynomial NARX model formulation are:

1. Polynomial models suffer severely from the curse of dimensionality because the model size grows exponentially with the degree of nonlinearity;
2. Oscillatory interpolation behavior for high polynomial degrees of nonlinearity; and
3. Poor model accuracy or instability during model extrapolation.

In order to mitigate the problem of excessive model size due to the curse of dimensionality (Item 1 above) polynomial models are usually limited to a degree of nonlinearity of about 3, which is also low enough to prevent the oscillatory interpolation behavior problem (Item 2) from becoming a serious issue. The poor extrapolation behavior of the polynomial formulation (Item 3) is remedied with sufficient attention to the design of the excitation signals for obtaining system identification data (cf. Section 4.4.3). With due attention to these limitations and in view of the simplicity of the identification of the polynomial NARX model (using least squares) the polynomial NARX model becomes very competitive. It remains the classical approach to formulating the NARX model structure and is still very widely used.

4.4.5 Subset Methods for Identification of Polynomial NARX Models

Because of the large number of regressors potentially involved in Eq. 4.45, it is desirable to use orthogonal least squares (OLS) methods to identify only a subset of the parameters, namely the most significant parameters (Chen *et al.* (1989b), Zheng *et al.* (1999), and Nelles (2001)). Two of these methods, one based on the Classical Gram-Schmidt procedure and one based on the Modified Gram-Schmidt procedure, were implemented in this research. Two problems associated with this approach are (Nelles, 2001:581):

- While resulting in much smaller models than the full-set model of Eq. 4.45, the calculation process itself becomes very intensive for originally large models.
- Due to being based on the prediction error, the selection procedure is negatively impacted by too high sample frequencies in the presence of output feedback. (This is briefly discussed in Section 4.6.)

Zheng *et al.* (1999) furthermore point out the need to bias the parameter selection criterion in favour of monomials of lower degrees of nonlinearity, as well as the inadequacy of typical identification data sets to not just identify models with which to predict simulation outputs with relative accuarcy, but identify system invariants such as fixed points, limit cycles and domains of attraction (Arrowsmith & Place, 1992).

4.5 Inversion of the NARX Model

4.5.1 Conversion to State Space Form

Consider the SISO polynomial NARX formulation model (from Eq. 4.42 and 4.43)

$$y(k) = F(\varphi(k)) \ , \tag{4.47}$$

with

$$\varphi(k) = [u(k-r) \ \ldots \ u(k-n_b) \ y(k-1) \ \ldots \ y(k-n_a)]^{\mathrm{T}} \ , \tag{4.48}$$

and $r \geq 1$. Here $y(k)$ represents the simulation output of the model (thus we discard the prediction vs. simulation distinction in notation here). For reasons shortly discussed, we assume that there is a term containing $u(k-r)$ only in Eq. 4.47, and that the $u(k-r)$-only term is linear, and there are no other terms containing $u(k-r)$, however without imposing similar restrictions on the other $u(k-i)$, $i > r$. (This renders the model input affine, as is shortly discussed in Section 4.5.3.) This restriction can be implemented during the system identification phase. The resulting NARX model is thus of the form

$$y(k) = \theta_1 u(k-r) + \tilde{F}(u(k-r-1), \ldots, u(k-n_b), y(k-1), \ldots \ y(k-n_a)) \ . \tag{4.49}$$

To obtain a corresponding state space system we can use the following state vector

$$x(k) = [u(k-1) \ \ldots \ u(k-n_b) \ y(k-1) \ \ldots \ y(k-n_a)]^{\mathrm{T}} \ , \tag{4.50}$$

from which there exists a $(n_a + n_b - r) \times (n_a + n_b)$ matrix P such that $\varphi(k) = Px(k)$, and the model (Eq. 4.47) may be expressed in terms of x as

$$y(k) = F(Px(k)) \ .$$

Using the state vector x Eq. 4.49 may be converted to the following state space form:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) && (4.51)\\ &= f_0(x(k)) + bu(k) && (4.52)\\ y(k) &= h(x(k)) \ , && (4.53) \end{aligned}$$

with $u(k), y(k) \in \Re$, $x(k) \in \Re^n$, $n = n_a + n_b$,

$$f(x(k)) + bu(k) = \begin{bmatrix} x_1(k+1) & = u(k) & \\ x_2(k+1) & = u(k-1) & = x_1(k) \\ \vdots & \vdots & \vdots \\ x_{n_b}(k+1) & = u(k-n_b+1) & = x_{n_b-1}(k) \\ x_{n_b+1}(k+1) & = y(k) & = F(Px(k)) \\ x_{n_b+2}(k+1) & = y(k-1) & = x_{n_b+1}(k) \\ \vdots & \vdots & \vdots \\ x_{n_b+n_a}(k+1) & = y(k-n_a+1) & = x_{n_b+n_a-1}(k) \end{bmatrix} \tag{4.54}$$

and

$$h(x(k)) = F(Px(k)) \; . \tag{4.55}$$

Since F is a polynomial model structure $(x^\circ, u^\circ) = ([0], [0])$ is an equilibrium point of the system.

4.5.2 Conversion to Normal Form

To invert the state space model (Eq. 4.52 and Eq. 4.53) we first convert the model to the normal form, for which we choose a coordinate transformation $z = \psi(x)$ with

$$\begin{aligned} z(k) &= \psi(x(k)) \\ &= [h(x(k)),\ h \circ f_0(x(k)),\ \ldots,\ h \circ f_0^{r-1}(x(k)),\ x_{r+1}(k),\ \ldots,\ x_n(k)]^{\mathrm{T}} \; , \end{aligned} \tag{4.56}$$

with the exponent of f_0 referring to the order of the composition. By the definition of f_0 and h in Eq. 4.52 and Eq. 4.53 and x in Eq. 4.50, it follows that $z_i(k) = \psi_i(x(k)) = y(k+i-1)$, $i = 1, \ldots, r$. The remaining states $\psi_{r+1}(x), \ldots, \psi_n(x)$ were simply chosen to result in the Jacobian of $\psi(x)$ being nonsingular at the point $x^\circ = [0]$, so that $z = \psi(x)$ qualifies as a local coordinate transformation in a neighbourhood of x°. Thus, in terms of u and y the state vector z becomes (cf. Eq. 4.50)

$$z(k) = [y(k)\ \ldots\ y(k+r-1)\ u(k-r-1)\ \ldots\ u(k-n_b)\ y(k-1)\ \ldots\ y(k-n_a)]^{\mathrm{T}} \; . \tag{4.57}$$

In the new z coordinates the system takes on the normal form (cf. Eq. 2.17 - 2.19):

$$\begin{array}{lll}
z_1(k+1) & = y(k+1) & = z_2(k) \\
\vdots & \vdots & \vdots \\
z_{r-1}(k+1) & = y(k+r-1) & = z_r(k) \\
z_r(k+1) & = y(k+r) & = \alpha(z(k)) + \beta(z(k), u(k)) \\
z_{r+1}(k+1) & = u(k-r) & = q_1(z(k)) \\
z_{r+2}(k+1) & = u(k-r-1) & = z_{r+1}(k) \\
\vdots & \vdots & \vdots \\
z_{n_b}(k+1) & = u(k-n_b+1) & = z_{n_b-1}(k) \\
z_{n_b+1}(k+1) & = y(k) & = z_1(k) \\
z_{n_b+2}(k+1) & = y(k-1) & = z_{n_b+1}(k) \\
\vdots & \vdots & \vdots \\
z_{n_b+n_a}(k+1) & = y(k-n_a+1) & = z_{n_b+n_a-1}(k)
\end{array} \tag{4.58}$$

and

$$y(k) = z_1(k) \tag{4.59}$$

4.5.3 Inversion of Normal Form

When performing inversion of a model a known output signal, y, is applied to the inverse of the model. To obtain the inverse model, we observe the following for the normal form of the model (Eq. 4.58 and Eq. 4.59):

- $z_{r+1}(k+1), \ldots, z_{n_b+n_a}(k+1)$ in Eq. 4.58 are not functions of $u(k)$ explicitly and, therefore, the elimination of $u(k)$ in these functions during inversion is thus not required,
- q_1 does not depend on $z_2, \ldots, z_r$,
- $z_{n_b+1}(k+1), \ldots, z_{n_b+n_a}(k+1)$ in Eq. 4.58 are already known (from $y(k)$).

The only unknown states are $z_{r+1}(k), \ldots, z_{n_b}(k)$, which we designate here $\eta_1(k), \ldots, \eta_{n_b-r}(k)$, i.e.

$$\eta_i(k) = z_{r+i}(k) \ ,$$

with $i = 1, \ldots, n_b - r$. The inverse system is now obtained from the normal form as

$$\eta(k+1) = \bar{f}(\eta(k), c(k)) \tag{4.60}$$

$$u(k) = \bar{h}(\eta(k), c(k)) \ . \tag{4.61}$$

The state equation (Eq. 4.60) is obtained from Eq. 4.58 as (cf. Eq. 4.57)

$$\begin{array}{lll} \eta_1(k+1) & = u(k-r) & = q_1(\eta(k), z_1(k), z_{n_b+1}(k), \ldots, z_{n_b+n_a}(k)) \\ & & = q_1(\eta(k), y(k), \ldots, y(k-n_a)) \\ \eta_2(k+1) & = u(k-r-1) & = \eta_1(k) \\ \vdots & \vdots & \vdots \\ \eta_{n_b-r}(k+1) & = u(k-n_b+1) & = \eta_{n_b-r-1}(k) \end{array} \tag{4.62}$$

in which the known values $(z_1(k), z_{n_b+1}(k), \ldots, z_{n_b+n_a}(k)) = (y(k), \ldots, y(k-n_a))$ are inserted, giving rise to the constant term $c(k)$. The output equation of the inverse system (Eq. 2.58) is obtained as the inverse of $z_r(k+1)$ in Eq. 4.58 for $u(k)$. The inverse system may be solved for the bounded solution of $\eta(k)$ as in Section 2.4.

We note here that after having solved $\eta(k)$, $u(k)$ may be determined by advancing $\eta_1(k)$ by $r+1$ time instants (since $\eta_1(k) = u(k-r-1)$) instead of using Eq. 2.58. When determining $u(k)$ in this way then Eq. 2.58 is not needed anymore, and thus neither is $z_1(k+1), \ldots, z_r(k+1)$ in Eq. 4.58, which Eq. 2.58 is determined from. The reason is that Eq. 2.58 is now neither used for the purpose of first eliminating $u(k)$ in $z_{r+1}(k+1), \ldots, z_{n_b+n_a}(k+1)$ to obtain Eq. 4.60 via the typical inversion procedure, nor for the purpose of solving $u(k)$ after having solved $\eta(k)$. Therefore, since $z_1(k+1), \ldots, z_r(k+1)$ are not needed anymore, Eq. 4.62 can be conveniently derived directly from the NARX model Eq. 4.49 by selecting the state vector

$$\eta(k) = [u(k-r-1) \ \ldots \ u(k-n_b)]^{\mathrm{T}},$$

as

$$\begin{array}{rcl} \eta_1(k+1) & = & u(k-r) \\ & = & (y(k) + \tilde{F}(u(k-r-1), \ldots, u(k-n_b), y(k-1), \ldots \ y(k-n_a)))/\theta_1 \\ & = & (y(k) + \tilde{F}(\eta_1(k), \ldots, \eta_{n_b-r}(k), y(k-1), \ldots \ y(k-n_a)))/\theta_1 \\ & = & q_1(\eta(k), y(k), \ldots, y(k-n_a)) \\ \eta_i(k+1) & = & \eta_{i-1}(k) \end{array} \tag{4.63}$$

with $i = 2, \ldots, n_b - r$.

The model structure of the $u(k-r)$ term in Eq. 4.47 and Eq. 4.48 becomes the model structure of the $\beta(z(k), u(k))$ term in $z_r(k+1)$ in Eq. 4.58, and therefore influences the local relative degree as defined in Eq. 2.10. While the use of a polynomial model structure for the $u(k-r)$ term in Eq. 4.47 may ensure a relative degree r, this will generally only apply in a neighbourhood u°, and not globally. For example, a polynomial model term in Eq. 4.47 containing a quadratic $u(k-r)$ factor will lead to $\frac{\partial y(k+r)}{\partial u(k)}$ vanishing for some u and will limit the neighbourhood of u° in which the system has relative degree r (cf. Eq. 2.10). Therefore, here we simplify the situation by allowing only a linear $u(k-r)$ term

in Eq. 4.47, of the form $\theta_1 u(k-r)$, resulting in the $bu(k)$ term in Eq. 4.52, however without imposing similar restrictions on the other terms in Eq. 4.47 containing $u(k-i)$ factors. This specialization of the NARX polynomial model renders it *input-affine* or *control-affine* (Nelles, 2001:548,606), which in case of state space models refers to models of the general form

$$x(k+1) = f(x(k)) + b(x(k))u(k) \ .$$

The effect of this is to simplify the inversion and thus the construction of a standard procedure for the identification and subsequent inversion of nonlinear models for use in software for performing response reconstruction by means of ILC.

Note, finally, if $r = n_b$ then $u(k-r)$ may be solved directly from the NARX model and the inverse becomes a quasi-static model.

4.5.4 Example

In the following example we convert a NARX model with $r \geq 1$ first to the state space system of Eq. 4.51 and Eq. 4.52, and then to the normal form (Eq. 4.58 and 4.59) to demonstrate the procedure of constructing the normal form from an existing state space system. Consider the following NARX polynomial model:

$$\begin{aligned} y(k) &= \theta_1 u(k-4) + \theta_2 u(k-5)^2 u(k-6) y(k-1) \\ &\quad + \theta_3 u(k-5) y(k-4) + \theta_4 y(k-4) + \theta_5 u(k-5) + \theta_6 u(k-6) \ , \end{aligned} \tag{4.64}$$

from which $n_a = 4$, $n_b = 6$ and $r = 4$. Choosing a state vector as in Eq. 4.50, namely

$$x(k) = [u(k-1) \ \ldots \ u(k-6) \ y(k-1) \ \ldots \ y(k-4)]^{\mathrm{T}} \ , \tag{4.65}$$

gives the state space system Eq. 4.52 and Eq. 4.53 as

$$\begin{aligned} x(k+1) &= f_0(x(k)) + bu(k) \\ y(k) &= h(x(k)) \ , \end{aligned}$$

with $u(k), y(k) \in \Re$, $x(k) \in \Re^n$, $n = n_a + n_b = 10$. For the state equation we have (keep in mind Eq. 4.65)

$$f_0(x(k)) + bu(k) = \begin{bmatrix} x_1(k+1) &= u(k) \\ x_2(k+1) &= x_1(k) \\ &\vdots \\ x_6(k+1) &= x_5(k) \\ x_7(k+1) &= \theta_1\, x_4 + \theta_2\, x_5^2\, x_6\, x_7 + \theta_3\, x_5\, x_{10} + \theta_4\, x_{10} + \theta_5\, x_5 + \theta_6\, x_6 \\ x_8(k+1) &= x_7(k) \\ &\vdots \\ x_{10}(k+1) &= x_9(k) \end{bmatrix} \tag{4.66}$$

and for the output equation we have

$$h(x(k)) = \theta_1\, x_4 + \theta_2\, x_5^2\, x_6\, x_7 + \theta_3\, x_5\, x_{10} + \theta_4\, x_{10} + \theta_5\, x_5 + \theta_6\, x_6 \tag{4.67}$$

To obtain the normal form of this system we choose a coordinate transformation $z = \psi(x)$ with the new state vector as in Eq. 4.56, giving

$$\begin{aligned}
z_1(k) &= y(k) &&= h(x(k)) \\
&&&= \theta_1\, x_4 + \theta_2\, x_5^2\, x_6\, x_7 + \theta_3\, x_5\, x_{10} + \theta_4\, x_{10} + \theta_5\, x_5 + \theta_6\, x_6 \\
z_2(k) &= y(k+1) &&= h \circ f_0(x(k)) \\
&&&= \theta_1\, x_3 + \theta_2\, x_4^2\, x_5\, [y(k)]_{=z_1} + \theta_3\, x_4\, x_9 + \theta_4\, x_9 + \theta_5\, x_4 + \theta_6\, x_5 \\
z_3(k) &= y(k+2) &&= h \circ f_0^2(x(k)) \\
&&&= \theta_1\, x_2 + \theta_2\, x_3^2\, x_4\, [y(k+1)]_{=z_2} + \theta_3\, x_3\, x_8 + \theta_4\, x_8 + \theta_5\, x_3 + \theta_6\, x_4 \\
z_4(k) &= y(k+3) &&= h \circ f_0^3(x(k)) \\
&&&= \theta_1\, x_1 + \theta_2\, x_2^2\, x_3\, [y(k+2)]_{=z_3} + \theta_3\, x_2\, x_7 + \theta_4\, x_7 + \theta_5\, x_2 + \theta_6\, x_3 \\
z_5(k) &= u(k-5) &&= x_5 \\
z_6(k) &= u(k-6) &&= x_6 \\
z_7(k) &= y(k-1) &&= x_7 \\
z_8(k) &= y(k-2) &&= x_8 \\
z_9(k) &= y(k-3) &&= x_9 \\
z_{10}(k) &= y(k-4) &&= x_{10}
\end{aligned} \tag{4.68}$$

$\psi(x)$ may be confirmed to have a nonsingular Jacobian. The inverse coordinate transform is given as

$$\begin{aligned}
x_1(k) &= u(k-1) &&= \tfrac{1}{\theta_1}(z_4 - \theta_2\,[u(k-2)^2]_{=x_2}\,[u(k-3)]_{=x_3}\,z_3 \\
&&&\quad -\theta_3\,[u(k-2)]_{=x_2}\,z_7 - \theta_4\,z_7 - \theta_5\,[u(k-2)]_{=x_2} - \theta_6\,[u(k-3)]_{=x_3}) \\
x_2(k) &= u(k-2) &&= \tfrac{1}{\theta_1}(z_3 - \theta_2\,[u(k-3)^2]_{=x_3}\,[u(k-4)]_{=x_4}\,z_2 \\
&&&\quad -\theta_3\,[u(k-3)]_{=x_3}\,z_8 - \theta_4\,z_8 - \theta_5\,[u(k-3)]_{=x_3} - \theta_6\,[u(k-4)]_{=x_4}) \\
x_3(k) &= u(k-3) &&= \tfrac{1}{\theta_1}(z_2 - \theta_2\,[u(k-4)^2]_{=x_4}\,z_5\,z_1 \\
&&&\quad -\theta_3\,[u(k-4)]_{=x_4}\,z_9 - \theta_4\,z_9 - \theta_5\,[u(k-4)]_{=x_4} - \theta_6\,z_5) \\
x_4(k) &= u(k-4) &&= \tfrac{1}{\theta_1}(z_1 - \theta_2\,z_5^2\,z_6\,z_7 - \theta_3\,z_5\,z_{10} - \theta_4\,z_{10} - \theta_5\,z_5 - \theta_6\,z_6) \\
x_5(k) &= u(k-5) &&= z_5 \\
x_6(k) &= u(k-6) &&= z_6 \\
x_7(k) &= y(k-1) &&= z_7 \\
x_8(k) &= y(k-2) &&= z_8 \\
x_9(k) &= y(k-3) &&= z_9 \\
x_{10}(k) &= y(k-4) &&= z_{10}
\end{aligned} \tag{4.69}$$

The resulting normal form of the system is

$$\begin{aligned}
z_1(k+1) &= y(k+1) &&= z_2(k) \\
z_2(k+1) &= y(k+2) &&= z_3(k) \\
z_3(k+1) &= y(k+3) &&= z_4(k) \\
z_4(k+1) &= y(k+4) &&= \theta_1\,u(k) + \theta_2\,[u(k-1)^2]_{=x_1}\,[u(k-2)]_{=x_2}\,z_4 \\
&&&\quad +\theta_3\,[u(k-1)]_{=x_1}\,z_1 + \theta_4\,z_1 + \theta_5\,[u(k-1)]_{=x_1} + \theta_6\,[u(k-2)]_{=x_2} \\
z_5(k+1) &= u(k-4) &&= \tfrac{1}{\theta_1}(z_1 - \theta_2\,z_5^2\,z_6\,z_7 - \theta_3\,z_5\,z_{10} - \theta_4\,z_{10} - \theta_5\,z_5 - \theta_6\,z_6) \\
z_6(k+1) &= u(k-5) &&= z_5(k) \\
z_7(k+1) &= y(k) &&= z_1(k) \\
z_8(k+1) &= y(k-1) &&= z_7(k) \\
z_9(k+1) &= y(k-2) &&= z_8(k) \\
z_{10}(k+1) &= y(k-3) &&= z_9(k) \\
y(k) &= z_1(k)
\end{aligned} \tag{4.70}$$

With $z_1(k), \ldots, z_4(k), z_7(k), \ldots, z_{10}(k)$ known, by setting $(\eta_1(k), \eta_2(k)) = (z_5(k), z_6(k))$ the following 2 dimensional system may be extracted from Eq. 4.70

$$\begin{aligned}
\eta_1(k+1) &= u(k-4) &&= \tfrac{1}{\theta_1}(z_1 - \theta_2\,\eta_1^2\,\eta_2\,z_7 - \theta_3\,\eta_1\,z_{10} - \theta_4\,z_{10} - \theta_5\,\eta_1 - \theta_6\,\eta_2) \\
\eta_2(k+1) &= u(k-5) &&= \eta_1(k)
\end{aligned} \tag{4.71}$$

By inserting the known values for $z_1(k), \ldots, z_4(k), z_7(k), \ldots, z_{10}(k)$ into Eq. 4.71, we obtain the state equation of Eq. 4.60:

$$\begin{aligned}
\eta(k+1) &= \bar{f}(\eta(k), c(k)) \\
&= \begin{bmatrix} C_1(k)\,\eta_1(k) + C_2\,\eta_2(k) + C_3(k)\,\eta_1(k)^2\,\eta_2(k) + c_1(k) \\ \eta_1(k) \end{bmatrix}
\end{aligned} \tag{4.72}$$

with

$$\begin{aligned} c(k) &= \left(\frac{1}{\theta_1}z_1(k) - \frac{\theta_4}{\theta_1}z_{10}(k),\ 0\right)^{\mathrm{T}} \\ C_1(k) &= -\frac{\theta_3}{\theta_1}z_{10}(k) - \frac{\theta_5}{\theta_1} \\ C_2 &= -\frac{\theta_6}{\theta_1} \\ C_3(k) &= -\frac{\theta_2}{\theta_1}z_7(k) \end{aligned}$$

which may be evaluated by the substitution of Eq. 4.68. This state equation for the inverse system may be solved for the bounded solution of $\eta(k)$ as in Section 2.4. We use the initial condition $\eta^{(0)} = [0]$. Having determined the bounded $\eta(k)$, $u(k)$ is finally obtained as $u(k) = \eta_1(k+5)$ (since $\eta_1 = z_5 = u(k-5)$ - cf. Eq. 4.68).

4.6 The Multiple-Model Method

In this section some mechanisms of inaccuracy during the identification of ARX and NARX models are discussed, following which possible remedies are discussed and the multiple-model method is selected to address the problem. The method is further developed for implementation in response reconstruction, which requires the identification and subsequent inversion of the models in the approach.

4.6.1 Mechanisms of Inaccuracy in the Identification of NARX Models

The following are two mechanisms of general model inaccuracy of parametric time domain models:

- **Variance of the parameter estimate due to finite data length and model size:** In Section 4.3.6, Eq. 4.36 the dependence of variance on the identification data length N and model size n was given for large N, with the variance being directly proportional to n/N. The model error due to parameter variance due to finite N and model size n is not known to be inherently frequency biased.

- **Bias of the parameter estimate due to ARX and NARX model inconsistency:**

  When the model is not flexible enough to accurately model the physical system, parameter estimation is inconsistent and necessarily results in bias error. The bias of ARX (and by extension NARX) models due to their unrepresentative noise models was already discussed in Section 4.3.4. In practice the types of nonlinearity of physical systems are widely varying and it is therefore generally difficult, if not impossible, to have a nonlinear model flexible enough to fully represent the physical system. The frequency dependence of the model error due to the inconsistency will

vary on a case by case basis depending on which part of the physical test system's dynamics the model cannot accommodate.

The following are two mechanisms of general model inaccuracy of parametric time domain models relating to the choice of sample frequency:

- **Sensitivity of parameter variance to sampling frequency:** A short sample period translates into an increase in the variance of the estimate if it is much shorter than the dominant time constants of the system. Conversely, when the sample period exceeds the dominant time constants of the system, the variance drastically increases. The minimum occurs when the sample period roughly equals the dominant time constants of the system. Too fast sampling is therefore better than too slow sampling (a value of $10\times$ the bandwidth of the system is often recommended). Refer to Section 4.3.6 for more details.

- **Sensitivity of structure and parameter identification of feedback models to sample frequency:** When the sample frequency becomes very high there is a tendency for consecutive $y(k-i)$ and $u(k-i)$ terms to become highly correlated. In polynomial NARX models this results in different monomial regressors with a similar structure becoming virtually indistinguishable, leading to suboptimal model structure identification during orthogonal least square identification of subset NARX models. In the extreme the output $y(k)$ becomes nearly identical to $y(k-1)$, leading to the latter becoming the only selected regressor and causing proper structure identification to become impossible. This is discussed in Billings and Aguirre (1995), which also suggests a lower sample frequency for structure identification, and a somewhat higher sample frequency for parameter identification.

The above-mentioned model inaccuracies are general in nature, meaning the model error cannot be characterized as consistently having a frequency based tendencies of one type or another. Two mechanisms of model inaccuracy of parametric time domain models that impact the low frequency accuracy specifically are:

- **Low frequency inaccuracy due to model numerical problems:** When the sample frequency becomes high enough, at some stage the low frequency trends, which are slowly varying, tend to fall below the A-D resolution and thus become invisible to the identification methods of parametric models such as the ARX model that examine only a relatively small window of the data history at a time. The resulting model numerical problems effectively limit the lowest frequency that can be modeled with a particular model and sample frequency. The problem is illustrated in the following analysis (Goodwin, 1985). Consider the time-domain equivalent of a simple lag, $G(s) = 1/(\tau s + 1)$, and apply Euler's discretization formula, resulting in

$$y(k+1) = y(k) - \frac{T_s}{\tau}(y(k) - u(k)) \ , \tag{4.73}$$

with T_s the sample period. Assume $u(k)$ and $y(k)$ are scaled to the range $(-1, 1)$ and let b be the precision of the A-D converter in terms of number of bits. If the T_s/τ factor is smaller than the value represented by the least significant bit, i.e. $1/2^b$, the second term on the right hand side disappears and Eq. 4.73 becomes useless. Thus, the model requires that $\tau/T_s \ll 2^b$. Assuming $b = 10$ and letting $f_B = 1/(2\pi\tau)$ be the bandwidth and $f_s = 1/T_s$ the sample frequency, it is therefore required that $f_B > f_s/1000$. In other words, the lowest-frequency dynamics that can be modeled with such an A-D system is $f_s/1000$. Recalling the suggestion that $f_s = 10 f_B$ (cf. Section 4.3.6), the range of useable frequencies f becomes

$$\frac{f_s}{1000} < f < f_B = \frac{f_s}{10} , \tag{4.74}$$

which is just 2 decades. Goodwin states that, if anything, this analysis is still generous. Clearly the situation can be alleviated by either reducing the sample frequency or increasing the A-D resolution. The former approach however carries the risk overestimating the usually unknown time constants. The latter approach is useful, with between 16 bit and 24 bit A-D and D-A conversion having become common in recent times.

- **Low frequency inaccuracy due to ARX and NARX model bias:** The other possible cause of low frequency inaccuracy is bias of the identification procedure towards the high frequencies that may be experienced by ARX (and by extension NARX) models. The limit estimate may be described as (cf. Section 4.3.5):

$$\theta^* = \arg\min_{\theta} \int_{-\pi/T_s}^{\pi/T_s} \left| G_0(e^{j\omega T_s}) - \hat{G}(e^{j\omega T_s}, \theta) \right|^2 Q(\omega, \theta) d\omega \tag{4.75}$$

$$Q(\omega, \theta) = \frac{\phi_u(\omega)}{\left| \hat{H}(e^{j\omega T_s}, \theta) \right|^2} , \tag{4.76}$$

with $\phi_u(\omega)$ the spectrum of the input. It is clear from Eq. 4.75 that Q, and thus $\hat{H}$ and ϕ_u, act as a frequency-dependent weighting function for penalizing the misfit between $\hat{G}$ and G_0. The impact of using a high sample frequency is to increase the frequency range of the integral. This is usually not detrimental since $G_0(e^{j\omega T_s}) - G(e^{j\omega T_s}, \theta)$ usually tends to zero at higher frequencies. However, when the noise model is coupled to the dynamics, as is the case for ARX models where $H(e^{j\omega T_s}) = 1/A(e^{j\omega T_s})$, then the product

$$\frac{\left| G_0(e^{j\omega T_s}) - G(e^{j\omega T_s}, \theta) \right|^2}{\left| H(e^{j\omega T_s}, \theta) \right|^2} \tag{4.77}$$

does not tend to zero at high frequencies. As a result the model fit is biased towards the high frequency region, resulting in the concentration of inaccuracy in the low frequency region. NARX models exhibit similar bias tendencies based on a similar mechanism (Nelles, 2001:559). Three corrective measures aimed at improving low-frequency bias of ARX and NARX models are the following:

– Increasing the weight of the Q-factor in the low frequency region by modifying the input spectrum;

– Increasing the weight of the Q-factor in the low frequency region by prefiltering the identification data; and

– Reducing the sample frequency in order to shrink the band of inaccuracy as much as possible. This suggestion forms the basis of the multiple model approach.

These three corrective measures are now briefly discussed in turn, not just from the point of view of relieving low frequency bias, but of low frequency model accuracy and model accuracy generally. The last measure, leading to the multiple model approach, will be the main focus here.

4.6.2 Methods to Improve the Accuracy of NARX Model Identification:

Increasing the energy of the **input spectrum**, $\phi_u(\omega)$, in the low frequency region will strengthen the Q factor in the low frequencies. In view of the risk of bias in the ARX and NARX model, it is considered prudent to routinely accentuate the low frequency region of the input signal regardless of what additional measures to improve accuracy are employed.

With regard to the **prefiltering** approach, note that applying a prefilter to the prediction error before doing parameter estimation (cf. Eq. 4.30) results in (for the SISO case):

$$\begin{aligned} \varepsilon_F(k,\theta) &= L(q)\varepsilon(k,\theta) \\ &= \frac{L(q)}{\hat{H}(q,\theta)}(y(k) - \hat{G}(q,\theta)u(k)) \ , \end{aligned} \tag{4.78}$$

implying that the effect of prefiltering is equivalent to changing the noise model to $\hat{H}(q,\theta)/L(q)$, in which case the Q factor in Eq. 4.76 becomes

$$Q(\omega,\theta) = \frac{\left|L(e^{j\omega T_s})\right|^2 \phi_u(\omega)}{\left|\hat{H}(e^{j\omega T_s},\theta)\right|^2} \ . \tag{4.79}$$

We can design $L(q)$ specifically to increase the low frequency values of Q, and apply it to the prediction error, which clearly is equivalent to applying it to u and y if the predictor is linear and time invariant (cf. Eq. 4.78). In case of nonlinear models $L(q)$ is applied to the prediction error. Note that proper use of prefiltering not only allows the low-frequency bias to be alleviated, but by approximating the inverse of the noise model itself with the prefilter, the noise model may be made unity and the ARX model with least square estimation may be used to identify a corresponding OE model, which gives a consistent estimate despite the non-unity nature of the true process noise dynamics. (The estimate will be consistent only in case of open-loop identification. This requires an existing estimate of the noise

model, making it a multi-stage approach. Refer to Nelles (2001:502) where this method is referred to as *Repeated least squares and filtering for OE model estimation.*)

The first two general inaccuracy mechanisms listed above (variance due to finite data length and model inconsistency) have little bearing on the choice of sample frequency or systematic frequency characterization of the error. Instead they point to the need for adequate data length (even more so for nonlinear models), and the general inconsistency of the ARX and NARX noise model. As mentioned, the latter problem may be partly addressed with a prefiltering approach (with guaranteed results for the open-loop case and linear models). Additionally, the prefiltering approach may also address the low frequency bias problem of the ARX and NARX model (Spinelli *et al.* 2005). However, it will not address the problem of low frequency model inaccuracy associated with limited A/D and D/A resolution. In view of this, and faced with the prospect of prefilter design on an *ad hoc* basis, we prefer a third approach, which is referred to here as the **multiple model** approach (Ljung, 1999:494), which is well suited to standardization. In the multiple-model approach the total frequency band is partitioned into two or more smaller bands, each of which is modeled separately with a submodel that applies only to part of the total test frequency band. This allows the sample frequency of each submodel to be reduced to what would be an optimal value for the given frequency band of the specific submodel (about 10× the high frequency limit of the submodel in question). In line with this is the suggestion by Billings and Aguirre (1995) that the identification data sample frequency be viewed as an adjustable parameter that may be adjusted after data gathering to what would be an optimal value for the identification task at hand. The adjustment is achieved by appropriate anti-aliasing filtering (in anticipation of the new sample frequency) followed by resampling of the data at a slower rate.

The accuracy of the low frequency model in a multiple model approach is of particular importance for response reconstruction tests because the low to medium frequency range usually contributes by far the greatest share of fatigue damage in structures subject to fatigue testing. (For this reason it is important for the low frequency model to have the flexibility needed to model test system nonlinearities that may be of particular relevance at lower frequencies as accurate as possible.) While the multi-model approach will adress low-frequency inaccuracy problems associated with low A-D resolution and low-frequency bias of the ARX and NARX models, it will not solve the inconsistency problem of ARX and NARX models (which, however, is not a low frequency specific problem). But neither will the prefilter approach in the case of closed-loop systems, which is usually the case in response reconstruction testing. Two more reasons why the multi-model approach could potentially improve the accuracy of the low frequency models are:

- The lower sample frequency of the low frequency model alleviates the problem of sensitivity of structure identification and parameter identification to sample frequency in feedback models such as the ARX and NARX models (the fourth inaccuracy mechanism mentioned above), which also improves the accuracy of the low frequency models.

- It is intuitive that as result of the multi-model approach the low frequency models will have lower model orders due to the lower sample frequency of these models, which in case of polynomial NARX models will result in dramatically smaller low frequency models because of the curse of dimensionality phenomenon. Verifying this is a matter of future research, but if true this is an important advantage of the multi-model approach for response reconstruction tests that may sometimes contain up to 12 or more axes (e.g. in case of road simulators; see Wright, 1993:96). This is because when using polynomial NARX models it will allow much more comprehensive modelling of nonlinearity in the low frequency models than would have been possible with the unadjusted sample frequency due to the lower model order needed for the model with the lower sample frequency allowing higher degrees of nonlinearity to be identified.

In view of the above the multiple model approach is therefore considered the most robust, convenient and tractable approach to solving the problems of low-frequency inaccuracy and the curse of dimensionality within the framework of polynomial NARX models and linear regression for purposes of response reconstruction for fatigue testing with large multi-axis test setups.

4.6.3 Implementation of the Multi-Model Method

The implementation of the multi-model method is now briefly developed for respectively model identification, simulation and inversion. With the use of the multi-model approach having been developed for these operations, its employment in ILC is exactly as for a single (uni-) model.

During **identification** of a given submodel the anti-aliasing filtering and re-sampling is done to both the input and output data before identification (which is the way prefiltering is usually done). We do not apply the upper and lower frequency limits of the submodel before identification, only the anti-aliasing filtering (which is a low pass filter at at most half the frequency of the adjusted sample frequency (referred to as the Nyquist frequency). Consequently, for every submodel the identification is done on all frequencies from zero up to the Nyquist frequency of the given submodel. The rationale for this is the following:

- To a low frequency submodel the high frequency behavior appears as instantaneous static relationships, and can be handled by including an instantaneous term, $b_0 u(k)$, in the submodel.

- To a high frequency submodel the low frequency behavior resembles integrators. Though an integrator may be incorporated in the noise model, which is equivalent to differentiating the data, this will accentuate the high-frequency content and pushes the parameter fit into the high frequency range. A better option would have been to high-pass filter the data, and in this way explicitly remove the low frequency behavior, which may in fact be necessary for identification of OE models (to prevent dynamic behavior from being overshadowed). However, in case of ARX

and NARX models, which are equation error models and use the least squares identification method, this is not necessary and the presence of the low frequency data may be tolerated during identification (this is as result of the inherent flexibility of least squares, and is also alleviated by the use of higher order models than strictly necessary (Ljung, 1999)).

Thus, identification of a submodel is done on all data up to the particular submodel's Nyquist frequency irrespective of the frequency band intended to be applied to a model during subsequent simulation of the model (via band pass filtering). This is in order as long as we generally include an instantaneous term (in low frequency models especially), use the ARX or NARX model with least squares estimation, and use higher model orders than is strictly necessary. Fig. 4.3 shows the identification procedure of a nonlinear multi-model.

While **simulation** of the submodel is not directly employed in ILC, it is used to evaluate the accuracy of the identified submodel. For simulation of a given submodel the following steps are followed:

1. The anti-alias filtering is applied to the input data;
2. The input data is re-sampled to the desired sample frequency of the submodel;
3. The submodel is simulated with the resampled input data;
4. The calculated output is interpolated again to the original (master) frequency, f_s; and
5. The upper and lower frequency limits of the submodel are finally applied via band pass filtering to the calculated output. All filters (anti-aliasing and band pass filters) are implemented in zero-phase fashion.
6. In order to compute the simulated result over the entire frequency band of the multi-model, the filtered output signals of the different submodels are superimposed.

Fig. 4.4 shows the simulation procedure of a multi-model.

The basic approach is thus to do both identification and simulation of a submodel up to the Nyquist frequency, and only to apply the chosen model frequency band after simulation on the calculated output. The primary reason for this preference is because it allows the multiple model approach to more effectively accommodate nonlinear models. When dealing with linear systems the intended upper and lower frequency limits of the submodel may be enforced before identification of and simulation with the submodel because of the frequency independence of linear systems. In case of nonlinear systems there may be a cross influence amongst frequencies which, in order for a submodel to be as accurate as possible, requires it to operate on the widest frequency range possible in order to cover as many dynamical cross-influences as possible.

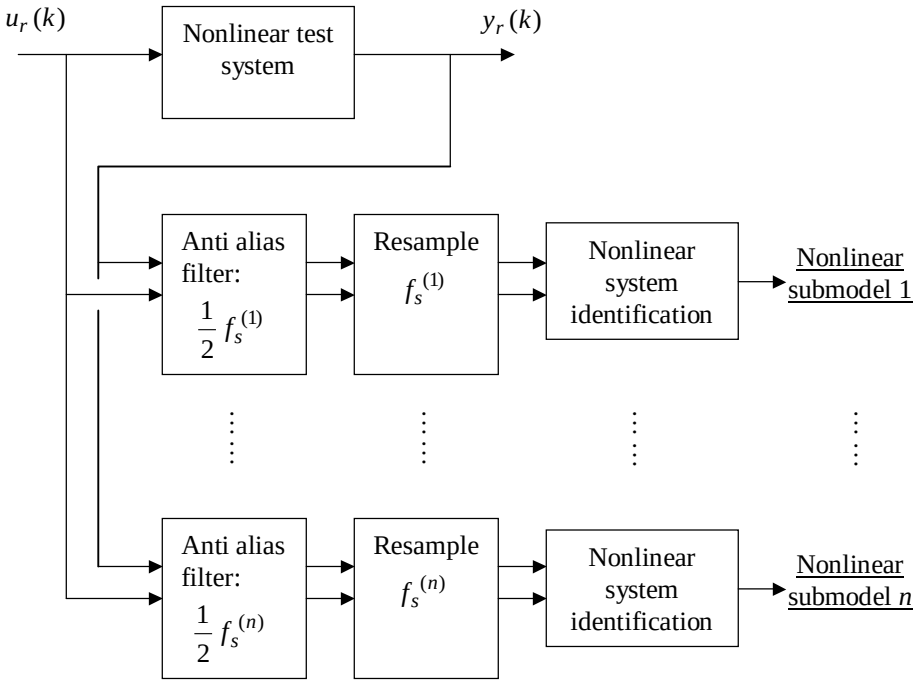

Figure 4.3: Identification of a nonlinear multi-model.

Finally, to implement the multiple model method in ILC in response reconstruction, we need a strategy for the **inversion** of multi-models. This is accomplished by reversing the procedure for model simulation as described above. In other words, the inverse is formulated for every submodel and is solved for the input signal over the entire frequency range up to the Nyquist frequency limit. Afterwards the upper and lower frequency limits of the specific submodel is applied to the calculated input signal. The input signals calculated in this way for the respective submodels are superimposed to give the solution of the inverse system over the entire frequency band of the model. Fig. 4.5 shows the inversion procedure of a multi-model. Fig. 4.6 gives an overall view of the model identification and validation procedure. The comparison of y_r with y_{rm} demonstrates the accuracy of the multi-model. The comparison of u_r with u (and y_r with y) demonstrates the accuracy of the inverse multi-model. Comparison of y_r with y_m demonstrates the success of the actual inversion procedure itself of the multi-model. Even for a poor multi-model, if the inverse multi-model is the exact inverse of the normal multi-model, this will give a very good match of y_r with y_m. The comparison of y_r with y_m thus gives an indication of the accuracy of the inversion itself.

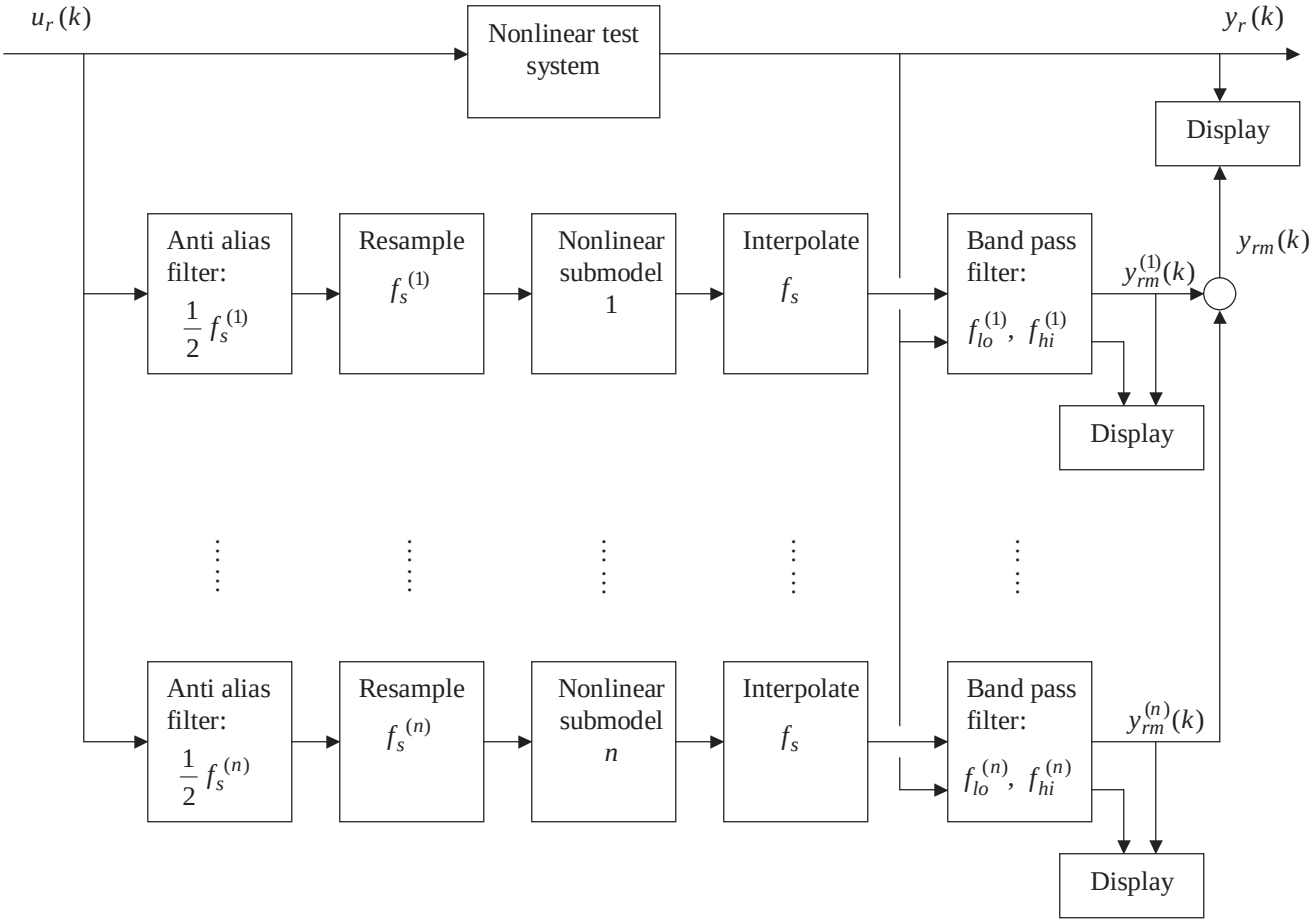

Figure 4.4: Simulation of a multi-model and comparison with the actual system output.

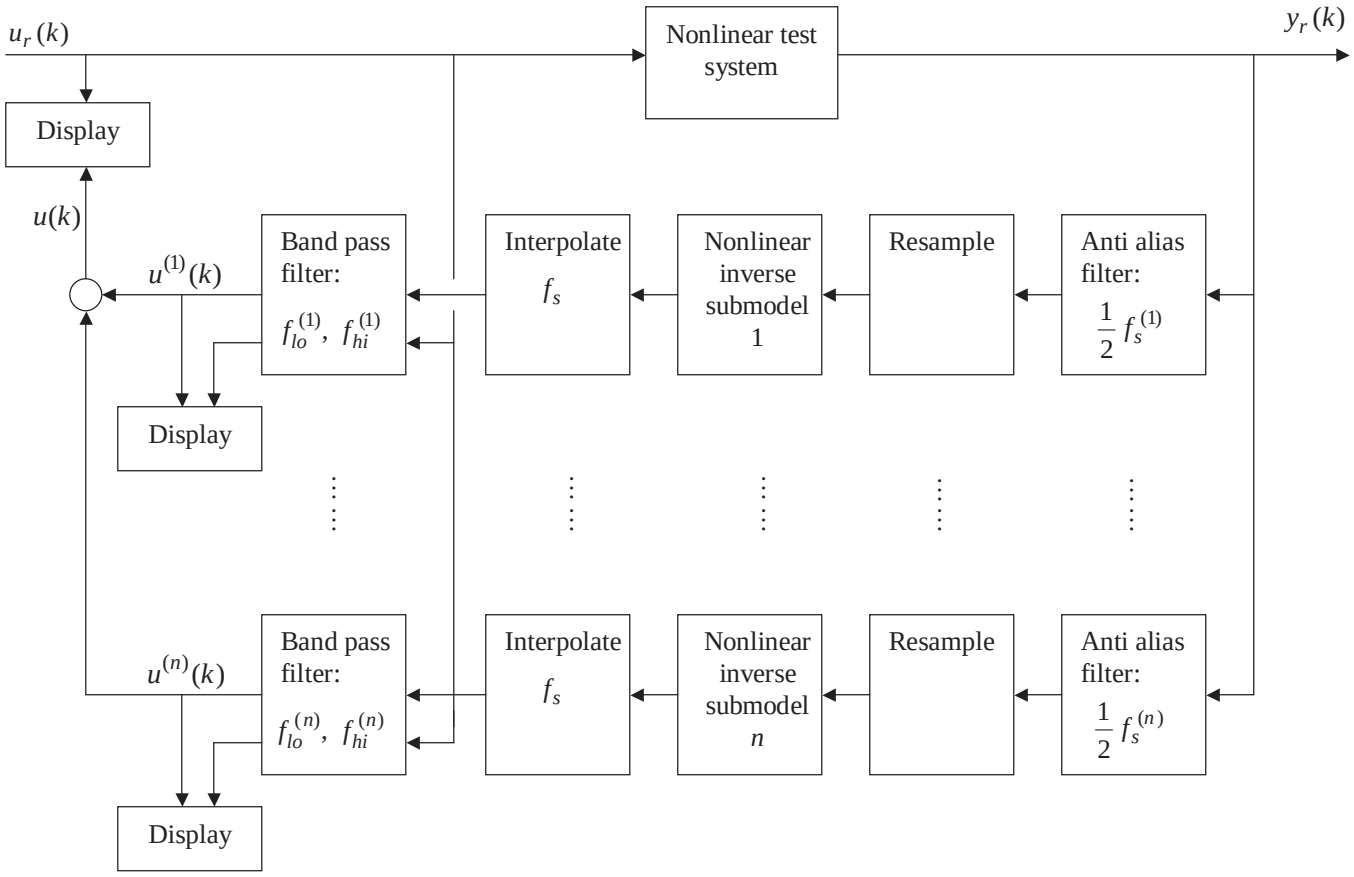

Figure 4.5: Simulation of an inverse multi-model and comparison with the actual system input.

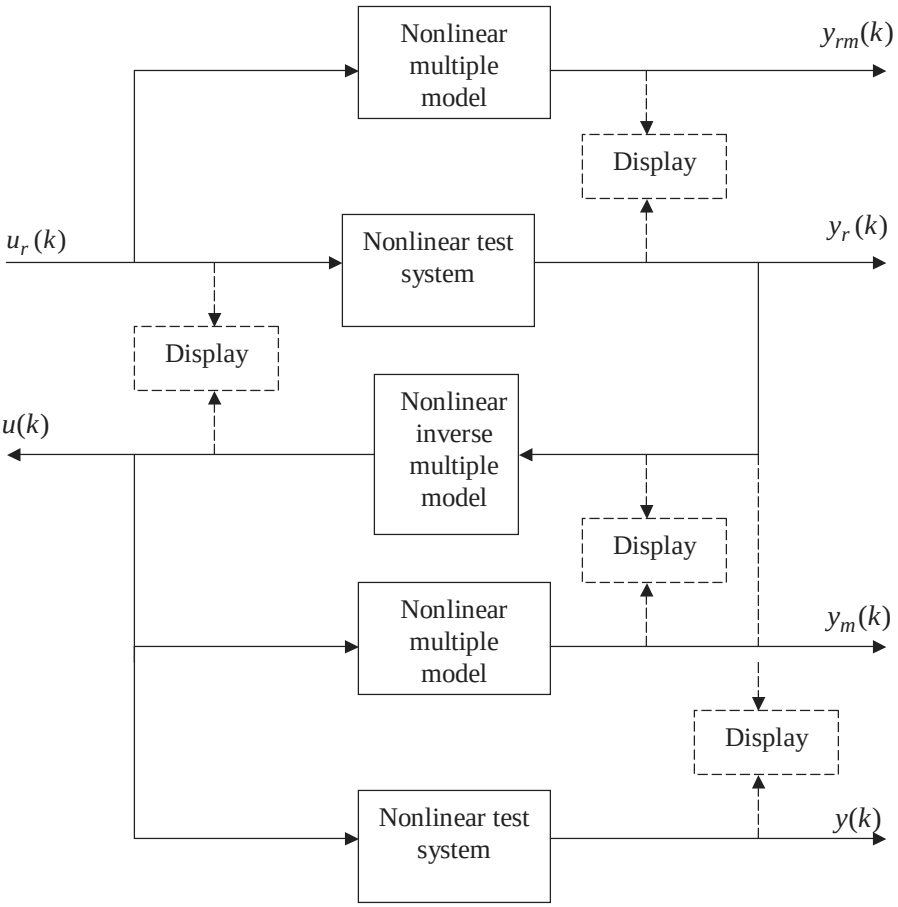

Figure 4.6: Overall view of the system identification and model validation procedure.

4.7 Example 5: Quarter Vehicle Road Simulator

4.7.1 Introduction

This quarter vehicle road simulator test is a demonstration example to practically investigate the various suggestions and advances made in this research, including the use of nonlinear inverse models, of inverse multiple-models in ILC, and of using the alternative ILC algorithm.

While the study focuses only on the accuracy of ILC in two strain channels, the purposes of such tests, and road simulation tests in general is usually structural integrity validation of the whole vehicle against fatigue.

For this laboratory set-up synthetic desires response data for the strain gauges was used instead of field measured strain data. Such data was generated using an actuator drive signal with a spectrum that has high power at lower frequencies and tapers of at higher frequencies as characteristic of typical road inputs. The actuator drive signal was tested and the corresponding strain measurements were saved and designated as desired response data for purposes of subsequent response reconstruction testing.

4.7.2 Test Setup

The test setup consisted of a monoshock swingarm motorcycle rear suspension, with the frame fitted with rigidly connected dummy weights. The shock absorber (i.e. damper assembly) is a coil-over-spring design with an external oil reservoir, and that sits in front of the wheel and connects directly to the swingarm via a bracket through a pivot point. There were two response channels on the test specimen:

- Channel A was a 0° - 90° rosette strain gauge on the shock's lower bracket connection to the swingarm. See Fig. 4.10.

- Channel B was a 0° - 90° rosette strain gauge just behind the front bearing of the swingarm on the right hand side upper surface of the swingarm. See Fig. 4.11.

AD and DA conversion was synchronized and was done through 24 bit and 16 bit National Instruments PCI cards respectively. The test specimen was tyre coupled to a 40 kN linear actuator. A picure of the test rig is shown in Fig. 4.7, and a diagrammatic representation of the test rig is shown in Fig. 4.8.

4.7.3 System Identification Procedure

In order to obtain identification (ID) data input/drive signals were generated from a tailored PSD spectrum curve over a frequency range between 1 - 249 Hz. The spectrum was tailored to result in broad-band random drive signals that gave a realistic spectrum for the corresponding measured responses. Drive signals were produced for two excitation levels, dubbed "medium amplitude" and "low amplitude".

The drive signals had generous duration of about 190 sec, and used about 30% of this duration for gradual ramp up and another 30% for gradual ramp down. The intention with this was to vary the operating point of the system for the sake of maximum representavity of the estimated nonlinear models and least chances of the models being employed in extrapolatory sense during ILC. This will not be entirely possible however as the general aim was for the "medium amplitude" data to still be of lower general severity than the envisaged desired responses, which was done from a practical point of view for the following reasons: Firstly the intensity levels of the drive signal that translate well to intensity levels of the desired response is unknown at the outset of response reconstruction, and driving a test system at very high energy states for the durations required for nonlinear identification data is often undesirable from a rig and specimen safety point of view. Secondly, system identification is frequently done with relatively low level drive signals (Raath, 1993a) which, when employing linear system identification as is usually done, have the best chance of resulting in models with a high accuracy on the ID data (though this is not necessarily optimal when employing the models in response reconstruction on nonlinear systems at nominal operating points of much higher energy). The intention was to take a middle road approach where system identification is done on a higher severity data set than usually employed in response reconstruction (to accommodate the needs of nonlinear system identification) but not as severe as to match the desired response data, and evaluate the feasibility of nonlinear methods in this scenario, which was deemed the most viable for practical implementation. If the models indeed turn out to be employed in an extrapolation sense in response reconstruction (which is bound to occur) and is unstable for the desired response data, this is not a critical set-back as the models are employed in the inverse sense only, in which case they are necessarily stably solved by the stable inversion method.

The subsequent testing of the drive signals to obtain corresponding response signals was a comprehensive process involving rig warm-up, repetitions of test to check repeatability, averaging of successive responses obtained for the repetitions (a standard approach in frequency domain identification also, aimed at reducing the variance of the estimate), and linear detrending of signals. Channel B showed considerable non-repeatability, as result of which it was decided to discard the low-amplitude ID data and only use the medium amplitude ID data during system identification for Channel B for the sake of a better signal to noise ratio. One possible cause of the non-repeatability is the proximity of the main swing-arm hinge point and its inherent free play in combination with the low value of the

bending strain so close to the hinge point. Channel A showed better repeatability than channel B and thus both the low amplitude and medium amplitude ID data was used during system identification for Channel A.

System identification was done on three bandwidths, namely 99, 149 and 249 Hz. The 99 and 149 Hz identification data sets were obtained by low pass filtering the 249 Hz bandwidth tests. The approach here was to identify NARX models and to reduce the number of parameters in the model using the Modified Gram-Schmidt procedure (cf. Section 4.4.5). The three sets of models were intended for use in ILC testing on the same three bandwidths.

A first round of preliminary ID trials was conducted to optimize various settings associated with stable inversion of identified models for the subsequent second round of preliminary ID trials. The second round of preliminary ID trials was then done to optimize the sample frequency of each submodel. The inversion settings were again optimized for the optimal sample frequencies, after which final systematic ID trials were conducted to optimize model order and the model polynomial's degree of nonlinearity. Following identification of the final models with the optimal sample frequencies, inversion settings, model orders and degrees of nonlinearity, the inversion settings were optimized for the final estimated models. It was found that the inversion settings that were obtained from optimization on the ID data were not very optimal for predicting the desired drive signal (which is normally not available) from the desired response signal. Consequently, in order for the focus not to be too much distracted from determining the potential of nonlinear system identification and model inversion in response reconstruction, the inversion parameters were again optimized, this time on the desired drive and response data. (Future research can focus on various strategies to determine the optimum iteration number in stable inversion without knowledge of the desired drive signal.) The results of inversion of the various linear and nonlinear and uni- and multi-models are presented in Tables 4.1 to 4.3. In the case of multi-models it was usually found that divergence of the stable inversion iteration process occurred at the higher frequency end first. The approach taken here was to then adjust the gain and number of iterations of lower frequency submodels downward accordingly so that the PSD spectrum of the inverse multi-model over the combined (total) frequency range is more or less continuous (i.e. do not show obviously large jumps at the boundaries of sub-models). This was not extremely rigorously applied but was biased toward higher progress of inversion iteration at the lower frequencies than rigorous application would have allowed. Nevertheless, in a number of cases it meant that in lower frequency sub-models inversion iteration was stopped short of the point that yields lowest inversion error. This does not necessarily translate into a equivalent loss of accuracy of ILC using such an inversion process because it is very similar to ILC with the full strength inverse (albeit without divergence) that is scaled down, which is synonymous to ILC with a lower ILC iteration gain (c).

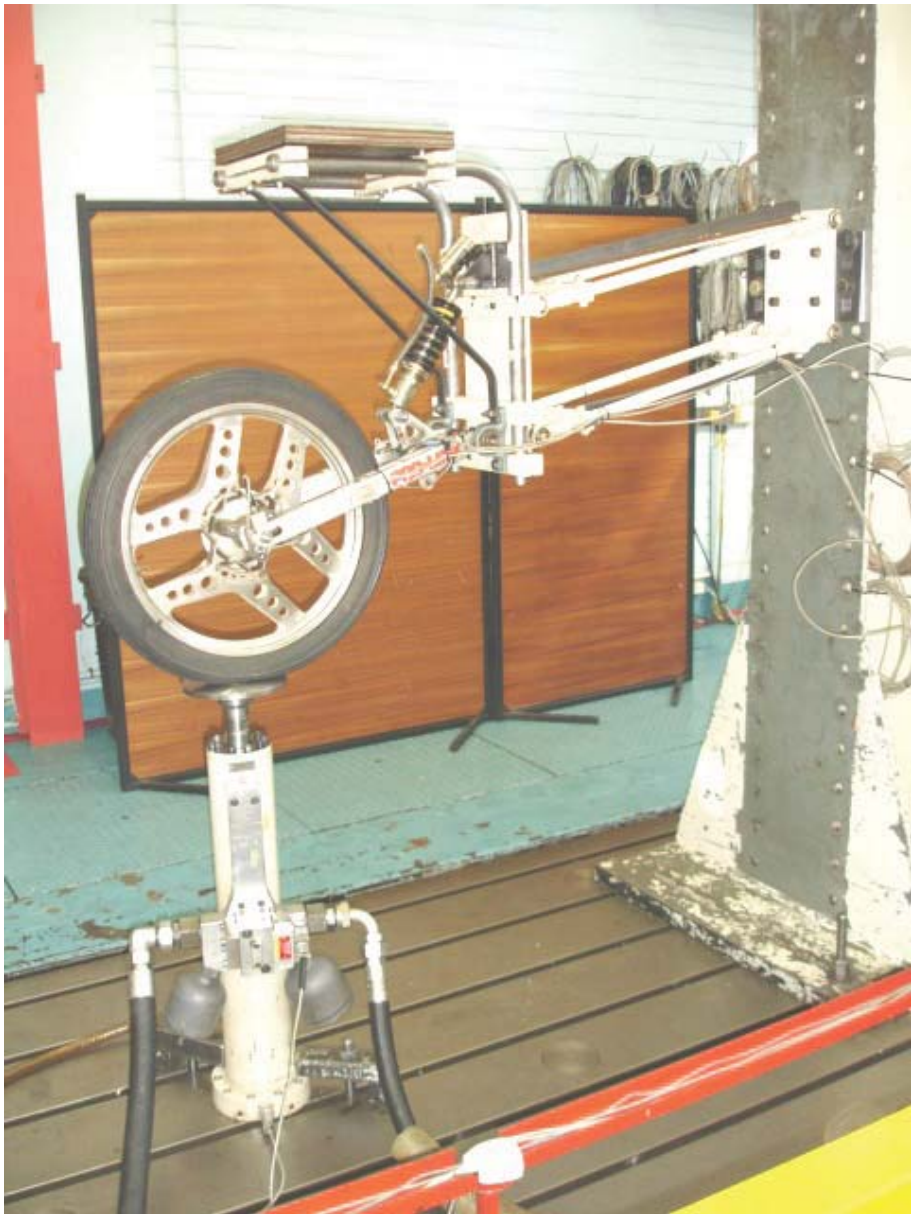

Figure 4.7: Quarter vehicle road simulator test rig.

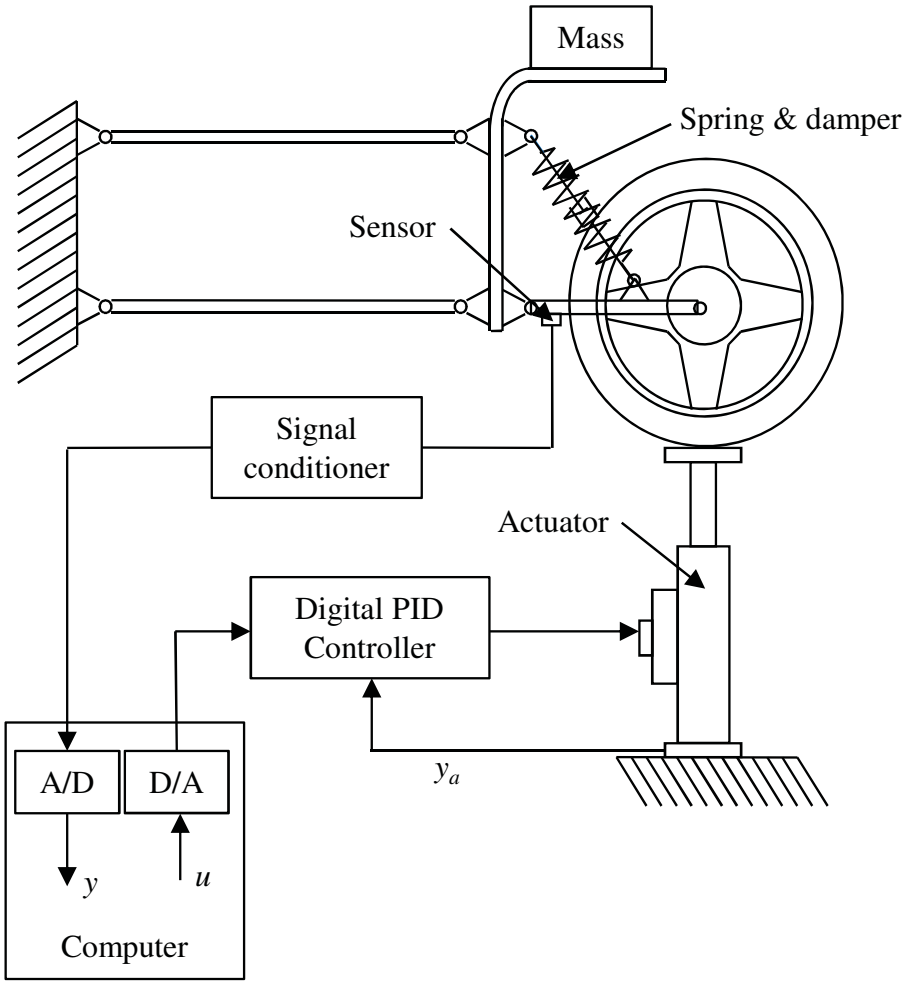

Figure 4.8: Diagram of quarter vehicle road simulator test rig.

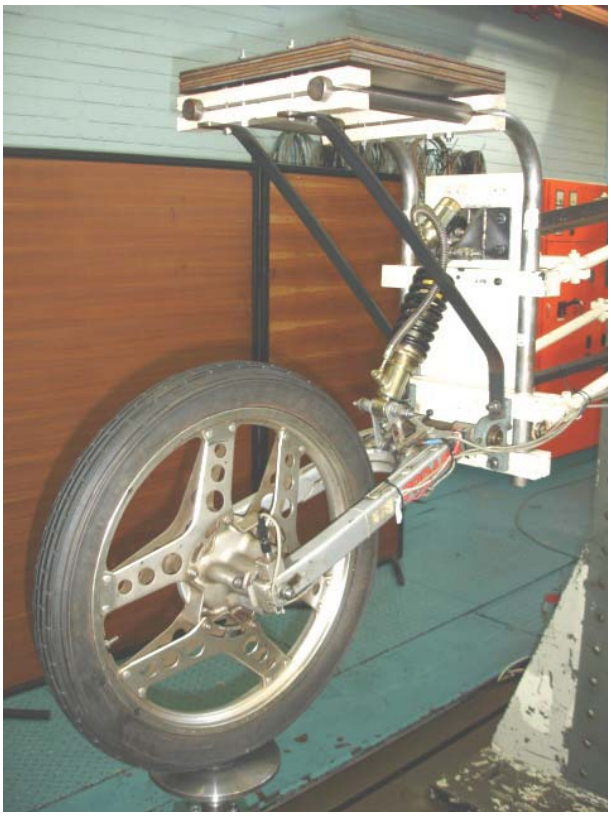

Figure 4.9: Detail of the motor cycle rig.

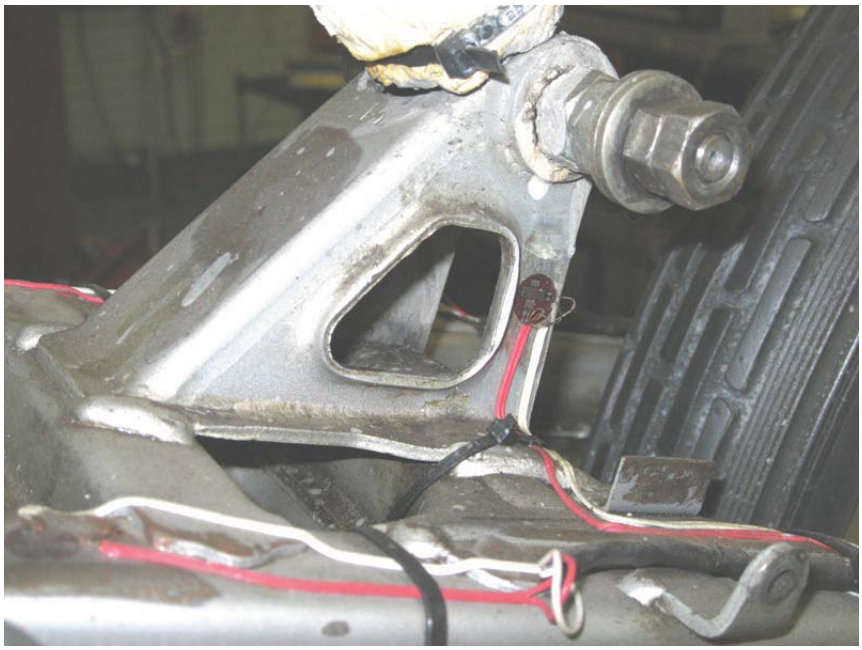

Figure 4.10: Strain channel A.

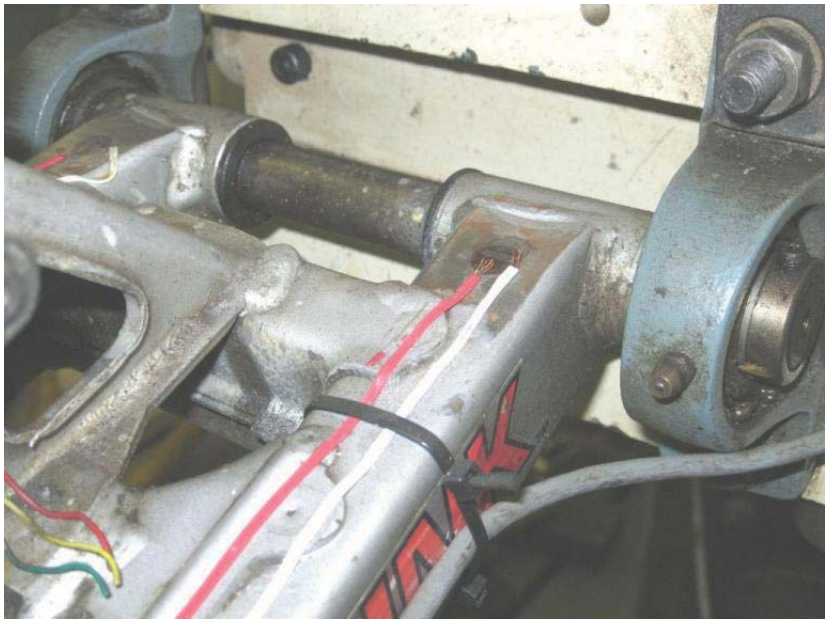

Figure 4.11: Strain channel B (on the right hand side).

4.7.4 Overall ILC Results

The best results from various tests for ILC on different bandwidths using linear inverse models and for ILC using nonlinear inverse models are summarized in Table 4.1. The percentage error between $u_d(k)$ and the input calculated during iteration m of stable inversion, namely $u^{(m)}(k)$, is here defined as:

$$\begin{aligned} \text{err}_2(u^{(m)}) &:= 100\frac{(\sum_{k=1}^{N}(u^{(m)}(k)-u_d(k))^2)^{1/2}}{(\sum_{k=1}^{N}u_d(k)^2)^{1/2}} \\ &= 100\frac{||u^{(m)}(k)-u_d(k)||_2}{||u_d(k)||_2}\,, \end{aligned} \tag{4.80}$$

and differs from Eq. 2.109 for historical reasons only. Similarly

$$\text{err}_2(y^{(m)}) := 100\,||y^{(m)}(k)-y_d(k)||_2/||y_d(k)||_2\,. \tag{4.81}$$

Results are presented for both the achieved input u and the achieved output y. Only the overall results are presented in these tables, namely the achieved accuracy (specifically, error values) over the whole test bandwidth. It is clear from these results that ILC with non-linear inverse models give significantly better results than ILC with linear models.

The best results from various tests for ILC using single inverse models (referred to here as the "uni-model" approach) and for ILC using inverse multiple-models, for both linear and nonlinear inverses, are summarized in Table 4.2. Again the results are the overall results, and are presented for different test bandwidths. It is clear from these results that ILC with multi-models give better results than ILC with single models.

Finally, the best overall results from various tests for ILC with the conventional algorithm and for ILC with the alternative algorithm of Section 3.4 are summarized in Table 4.3 for different bandwidths. The results are generally mixed, with the conventional algorithm giving better results in the 99 Hz and 149 Hz bandwidth tests, and the alternative algorithm giving generally better results in the 249 Hz bandwidth test.

Table 4.1: Best results for ILC with linear and nonlinear inverse models in $\tilde{L}$ for tests on different bandwidths.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	**1 - 99 Hz**		**1 - 149 Hz**		**1 - 249 Hz**	
-	-	**u**	**y**	**u**	**y**	**u**	**y**
A	Linear	42.8	47.9	63.9	51.0	76.8	61.9
A	Nonlinear	39.5	47.3	49.6	49.2	52.5	55.4
B	Linear	66.9	63.4	59.6	55.2	63.1	61.2
B	Nonlinear	47.9	51.7	46.9	48.8	47.0	50.6

Table 4.2: Best results for ILC with single and multiple inverse models in $\tilde{L}$ for tests on different bandwidths.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	**1 - 99 Hz**		**1 - 149 Hz**		**1 - 249 Hz**	
-	-	**u**	**y**	**u**	**y**	**u**	**y**
A	Uni-model	50.6	51.5	62.9	51.0	77.1	61.9
A	Multi-model	39.5	47.3	49.6	49.2	52.5	55.4
B	Uni-model	48.9	51.7	50.0	53.1	54.3	54.0
B	Multi-model	47.9	51.7	46.9	48.8	47.0	50.6

Table 4.3: Best results for ILC with the conventional and alternative algorithms for tests on different bandwidths.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	**1 - 99 Hz**		**1 - 149 Hz**		**1 - 249 Hz**	
-	-	**u**	**y**	**u**	**y**	**u**	**y**
A	Conventional	39.5	47.3	49.6	49.2	55.1	60.0
A	Alternative	43.2	54.0	58.3	59.3	52.5	55.4
B	Conventional	47.9	52.4	46.9	49.1	47.0	52.1
B	Alternative	49.9	51.7	48.5	48.8	53.2	50.6

4.7.5 ILC Results in Individual Frequency Bands

In Table 4.4 to Table 4.12 below the errors in individual frequency bands are given for the best u and y results featured in Table 4.1 to Table 4.3 above, with the results for every test bandwidth presented in a separate table.

Considering the results for u and y in Tables 4.4, 4.7, and 4.10 the improvement in accuracy by using nonlinear inverse models is observed in the individual frequency bands also, specifically in the low (1 - 15 Hz) and upper-medium to high (> 30 Hz) frequency ranges. The low frequency bands show clear improvements in accuracy which is due to the greater accuracy of the inverse model at low frequencies. While the improved accuracy of the nonlinear models is easy to show using model simulation with the identification data (i.e. the data that was used to identify the models), showing it for the desired response data (by simulating the model with the input that gives the desired response) is difficult because the nonlinear models tend to be unstable with this data set due to the lower levels of the "medium amplitude" data set compared to the desired response data, leading to the models to be used in an extrapolatory sense and thus be unstable. While the model is used in the inverse sense only and is therefore stably solved, inversion of the model still runs the risk of divergence of the inversion algorithm. As a consequence of the divergence of the inversion algorithm the inverse may have worse error norm values than a linear inverse model, despite the nonlinear normal model being more accurate than the linear normal model, which is indeed observed here with inversion of the desired response data. Again this is not as detrimental as may be feared since, while the nonlinear inverse may suffer divergence, the effect of this is to limit the number of iterations allowed to a selected optimum and thereby reduce the amplitude levels (magnitude) of the calculated inverse while still preserving the advantages of the more accurate nonlinear dynamics in the sense of better characterization of the dynamics at the amplitude levels involved. ILC with such an inverse is synonymous with ILC incorporating a lower iteration gain (i.e. c value), which is as good or better than ILC with a higher gain. The advantage of such a nonlinear inverse model in ILC is thus that, despite its divergence and consequent occasional poorer error values than a linear inverse in individual frequency bands (cf. Tables 4.4 to 4.6), it captures the nonlinearity of the system better and can thus lead to more accurate ILC, which is indeed what is observed in these results.

On the high frequency end the nonlinear inverse may have much lower error norm values than the linear inverse. This is because the linear inverse may have very high magnitude at high frequencies due to the natural attenuation at high frequencies of the model being inverted, which is aggravated by the typical model inaccuracy at high frequency (due to noise, low signal strength during identification, and nonlinearities at high frequencies - see Section 3.2.6, which also discusses various approaches to attenuate the magnitude of the inverse at high frequencies.) Normally the erroneously high magnitude of the inverse at high frequencies will reduce the bandwidth of convergence of ILC, and the common way of counteracting this in ILC is to use a C filter designed with attenuation at high frequencies

(cf. Section 4.2.4). Future work in the use of ILC in response reconstruction should indeed begin to exploit the C filter to improve the convergent bandwidth. The approach taken here, however, which is frequent in ILC performed for response reconstruction purposes, is to use $C = 1$ and instead rely on the constant c to improve convergence properties (and adjust the rate of convergence). With this as background, clearly the stable inversion algorithm for inversion of nonlinear models allows iterations to be limited prior to the onset of excessive magnitude of the output of the inverse model at high frequencies. Especially in case of multi-models the inversion algorithm gain of high frequency models may be particularly reduced, independently from the inversion gain of the other sub-models, allowing reduction of the inverse model output at high frequencies without severely impacting inverse model output at lower frequencies. (As previously mentioned, in such cases the gain and number of iterations of lower frequency sub-models were also down-tuned at least somewhat to retain some level of continuity of the overall spectrum). As a consequence ILC results with nonlinear inverses with these methods tend to be more accurate at high frequencies than ILC with linear models, especially when employing a unity C.

The improvements in ILC accuracy obtained by using multiple-models are observed in Table 4.5, Table 4.8, and Table 4.11 for u and y in the individual frequency bands also. In a number of cases improvement is observed at high frequencies. Apart from one exception this is ascribed to the use of nonlinear inverse multi-models where the inversion gains may be reduced and/or iteration during inversion may be cut short to reduce inverse model output generally and at high frequencies particularly. This is of course not possible with linear uni-models, in which the inversion is one step (i.e. non-iterative and the high frequencies are not modeled separately).

The exception that was mentioned is the 30 - 99 Hz frequency range in the 99 Hz bandwidth test, where the unimodel was in fact nonlinear. In this case, while the inversion iteration of the unimodel may be halted prematurely to limit divergence at high frequencies, the inversion gain may not be reduced at high frequencies independently of the lower frequency range (at least not in the standard stable inversion approach).

Moderate increases in accuracy can also be observed at low frequencies in the 99 Hz and 149 Hz bandwidth tests. In the 249 Hz bandwidth test the increase in accuracy at low frequencies become especially pronounced. This can be generally correlated with the accuracy of the identified models (evaluated for the desired drive and response data set) where, for both channels, the accuracy of the 1 - 15 Hz frequency band is usually as good or significantly better than for the uni-model, particularly for the inverse model, where the multi-model tend to give much better results than the uni-model in the 1 - 15 Hz frequency band. It is also observed for the simulation accuracy of the identified models that as the test bandwidth increases, and thus the sample frequency of the uni-model relative to the low frequency band increases, the accuracy of the uni-model in the low frequency band tend to decrease. The probable explanation for this is that the higher values of the sample frequency of

the uni-model begins to result in the sample-frequency dependent inaccuracy mechanisms in system identification at low frequencies to begin to take effect in the uni-model (cf. Section 4.6). While the effect is not very pronounced for the normal model (probably due to the high A-D precision that was used, namely 24 bit, and the relatively strong spectrum of the input signals on the low frequency end, even for the "low amplitude" signals), it is quite apparent in the inverse model accuracy.

Mixed results are observed for using the alternative ILC algorithm compared to the conventional ILC algorithm in Table 4.6, Table 4.9 and Table 4.12 for u and y in the individual frequency bands also. It is observed that the alternative algorithm gives worse results on the low frequency end except for the 249 Hz bandwidth tests for channel A, where it gives much better results. On the mid to high frequency end the alternative algorithm gives similar to generally better results than the conventional algorithm. Interpreting these results is difficult but it should be stressed that these results are for the unmodified alternative algorithm, with the modified version considered more viable and promising than the unmodified version. Evaluation of the modified version of the alternative algorithm is a subject of future study.

Table 4.4: Best results for ILC over 1 - 99 Hz with linear and nonlinear inverse models in $\tilde{L}$ - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz	
-	-	u	y	u	y	u	y
A	Linear	28.2	33.5	76.1	61.7	198.5	118.3
A	Nonlinear	32.0	38.2	132.1	75.3	118.5	120.0
B	Linear	56.2	50.6	77.6	69.0	277.5	105.7
B	Nonlinear	44.2	37.3	113.3	65.6	103.9	91.8

Table 4.5: Best results for ILC over 1 - 99 Hz with single and multiple inverse models in $\tilde{L}$ - individual frequency bands. Lin. and NL indicates whether the uni- or multi-model was linear or nonlinear respectively.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz	
-	-	u	y	u	y	u	y
A	Uni-model (Lin.)	37.5	37.3	69.5	62.1	189.3	102.8
A	Multi-model (NL)	32.0	38.2	132.1	75.3	118.5	120.0
B	Uni-model (NL)	38.5	37.9	91.1	59.9	228.2	94.2
B	Multi-model (NL)	44.2	37.3	113.3	65.6	103.9	91.8

Table 4.6: Best results for ILC over 1 - 99 Hz with the conventional and alternative algorithms - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]					
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz	
-	-	u	y	u	y	u	y
A	Conventional	32.0	38.2	132.1	75.3	118.5	120.0
A	Alternative	41.6	47.7	124.7	62.2	104.8	99.6
B	Conventional	44.2	36.8	113.3	60.2	103.9	100.2
B	Alternative	46.5	37.3	110.9	65.6	103.0	91.8

Table 4.7: Best results for ILC over 1 - 149 Hz with linear and nonlinear inverse models in $\tilde{L}$ - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]							
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz	
-	-	u	y	u	y	u	y	u	y
A	Linear	52.5	42.4	67.8	55.5	174.8	102.4	159.0	205.3
A	Nonlinear	45.8	43.0	85.2	72.1	106.7	109.1	99.6	109.9
B	Linear	49.7	38.5	82.3	65.3	196.8	101.8	126.0	83.5
B	Nonlinear	39.6	33.9	77.2	67.8	132.2	83.2	95.3	88.7

Table 4.8: Best results for ILC over 1 - 149 Hz with single and multiple inverse models in $\tilde{L}$ - individual frequency bands. Lin. and NL indicates whether the uni- or multi-model was linear or nonlinear respectively.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]							
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz	
-	-	u	y	u	y	u	y	u	y
A	Uni-model (Lin.)	58.3	42.4	84.0	55.5	154.1	102.4	107.5	205.3
A	Multi-model (NL)	45.8	43.0	85.2	72.1	106.7	109.1	99.6	109.9
B	Uni-model (NL)	41.9	36.2	107.8	61.8	180.7	100.4	88.2	99.7
B	Multi-model (NL)	39.6	33.9	77.2	67.8	132.2	83.2	95.3	88.7

Table 4.9: Best results for ILC over 1 - 149 Hz with the conventional and alternative algorithms - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]							
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz	
-	-	u	y	u	y	u	y	u	y
A	Conventional	45.8	43.0	85.2	72.1	106.7	109.1	99.6	109.9
A	Alternative	55.7	53.7	88.4	78.8	120.3	105.7	101.2	104.3
B	Conventional	39.6	31.8	77.2	70.5	132.2	85.0	95.3	80.7
B	Alternative	40.8	33.9	78.5	67.8	154.4	83.2	92.6	88.7

Table 4.10: Best results for ILC over 1 - 249 Hz with linear and nonlinear inverse models in $\tilde{L}$ - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]									
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz		149 - 249 Hz	
-	-	u	y	u	y	u	y	u	y	u	y
A	Linear	63.5	55.1	69.2	67.4	137.9	110.5	171.4	179.0	472.2	227.2
A	Nonlinear	45.1	40.9	88.6	82.2	99.7	114.2	100.6	109.5	97.9	105.2
B	Linear	52.8	42.0	76.6	70.8	159.5	115.1	124.4	101.1	322.9	124.0
B	Nonlinear	41.5	31.5	85.6	72.6	126.8	89.0	89.1	88.1	128.4	105.1

Table 4.11: Best results for ILC over 1 - 249 Hz with single and multiple inverse models in $\tilde{L}$ - individual frequency bands. Lin. and NL indicates whether the uni- or multi-model was linear or nonlinear respectively.

Chan.	**Description**	$\min_m \text{err}_2(\cdot)$ [%]									
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz		149 - 249 Hz	
-	-	u	y	u	y	u	y	u	y	u	y
A	Uni-model (Lin.)	77.3	55.1	109.1	67.4	112.7	110.5	109.3	179.0	100.1	227.2
A	Multi-model (NL)	45.1	40.9	88.6	82.2	99.7	114.2	100.6	109.5	97.9	105.2
B	Uni-model (NL)	42.9	39.8	90.1	62.9	193.1	91.5	98.4	80.2	127.0	109.1
B	Multi-model (NL)	41.5	31.5	85.6	72.6	126.8	89.0	89.1	88.1	128.4	105.1

Table 4.12: Best results for ILC over 1 - 249 Hz with the conventional and alternative algorithms - individual frequency bands.

Channel	**Description**	$\min_m \text{err}_2(\cdot)$ [%]									
-	-	1 - 15 Hz		15 - 30 Hz		30 - 99 Hz		99 - 149 Hz		149 - 249 Hz	
-	-	u	y	u	y	u	y	u	y	u	y
A	Conventional	51.5	55.7	85.0	79.4	102.2	112.2	105.1	120.6	97.6	119.4
A	Alternative	45.1	40.9	88.6	82.2	99.7	114.2	100.6	109.5	97.9	105.2
B	Conventional	41.5	29.7	85.6	72.3	126.8	99.4	89.1	87.3	128.4	111.4
B	Alternative	45.9	31.5	78.5	72.6	157.3	89.0	90.4	88.1	108.9	105.1

Chapter 5

Conclusions

The contributions of this research lie in three main areas, namely stable inversion, the ILC algorithm, and response reconstruction as used in structural integrity testing.

5.1 Stable Inversion

It is frequently found that the stable inversion of NARX models, which result from system identification during response reconstruction tests in this research, prove to be divergent, and often strongly so. This research made two novel contributions aimed at improving the accuracy of stable inversion under such circumstances (either by recovery of convergence or lessening the strength of divergence), namely incorporating alternative iteration schemes from the field of fixed point iteration, and incorporating a low pass filter in the algorithm. In a set of three examples in Chapter 2 the gains in accuracy when stable inversion of NARX models diverges was confirmed for stable inversion using the Mann iteration scheme, and for using a low pass filter in the algorithm. The results demonstrated either the recovery of convergence using these two measures (examples 1 and 2), or a reduction in the strength of divergence (and thus increased accuracy of inversion results prior to divergence; Example 3). The examples also demonstrated improvements in accuracy of stable inversion by using different strategies for the gain employed in Mann iteration instead of a constant gain value, namely a time-dependent and iteration dependent gain (with the latter being a novel contribution to Mann iteration to the author's knowledge).

5.2 Iterative Learning Control

Towards improving the convergence properties of the ILC algorithm an alternative ILC algorithm was developed that functions inherently differently to the conventional algorithm. Further modifications

were also made to get the alternative algorithm to fully match the conventional algorithm in some of the most important fundamental properties of the algorithm. The algorithm in unmodified form was demonstrated in an example consisting of a set of five case studies to either result in dramatically improved convergence properties, or comparable to slightly inferior results to the conventional algorithm.

The ability of ILC to converge over a given frequency band depends on the ILC compensator's dynamics over the band, with best results achieved when employing an inverse model of at least "reasonable" accuracy in the compensator. While the accuracy of the inverse model is not essential for the convergence of ILC, the more accurate it is the better the chances of achieving convergence are, or at least of achieving good results prior to divergence. A novel contribution of this research is the incorporation of nonlinear inverse models in the ILC compensator in case of ILC of nonlinear systems, with the intention of improving the accuracy of the inverse model in these cases and thereby improve the convergence properties of ILC. While this has been suggested by Markusson (2002) in a purely ILC context, it was not recommended in the paper due to issues of complexity of the inversion process. In Chapter 4 the stable inversion process for NARX models in particular, which are the type of nonlinear models used in this research, was developed for slight limitations imposed on the model prior to to inversion. The inescapable reality of this approach is however that stable inversion, which is conducted for every iteration of ILC, is itself iterative for nonlinear inverses, which substantially impacts ILC by increasing complexity and computational intensity. In response reconstruction tests for purposes full-scale fatigue testing the added complexity during the ILC phase is considered justified if it can result in sufficient gains in accuracy. The use of a nonlinear inverse model in the ILC compensator was examined in the examples in Chapter 3, where it was found to indeed contribute to more accurate results in a case where ILC was divergent.

5.3 Response Reconstruction

In Chapter 3 a number of model based and inverse based ILC compensators in the literature that are meant to obtain monotonic convergence of ILC of linear systems were presented. These were shown to be special cases of a very general ILC compensator,

$$L = cC\tilde{L} \, ,$$

which is called here the *general inverse-based* (GIB) compensator, with $\tilde{L}$ the approximate system inverse, C a zero phase filter, and c a real scalar. As a novel contribution this ILC compensator is recommended for general use in response reconstruction. While a closely related form, $L = c\tilde{L}$, is currently already used in response reconstruction, the addition of the C filter is new in the response reconstruction setting to the knowledge of the author.

Response reconstruction utilizes ILC to achieve the tracking of desired response signals that simulate service loading of the specimen for purposes of structural integrity testing. The success of response reconstruction depends on the ILC procedure being well-behaved (i.e. maintain progressive convergence without temporary wide divergence) during the iteration process, and on the accuracy of the final reconstruction results. While the preferred outcome of ILC is to achieve convergence of the algorithm, ideally over a frequency band sufficient for the requirements of the test, this is frequently not achieved in dynamic response reconstruction tests (such as fatigue tests, and shock and/or vibration tests) primarily due to the limited accuracy of the inverse models employed in the ILC compensator over the frequency band of the test. Two mechanisms contributing to reduced accuracy of the inverse model are the following:

- *Nonlinearity of the test system:* When the test system is sufficiently nonlinear, the accuracy of a linear inverse model may be inadequate to achieve convergence of ILC or at least achieve reconstruction results of sufficient accuracy prior to divergence. The incorporation of a nonlinear inverse model in the ILC compensator aimed at alleviating this was referred to in Section 5.2 as a novel contribution of this research. The nonlinear models used in this research is of the NARX type, and in Chapter 4 the stable inversion process for NARX models in particular was developed. In a quarter vehicle road simulator demonstration rig the use of nonlinear inverse models of the NARX type in the ILC compensator was shown to lead to appreciable gains in the accuracy of response reconstruction.

- *Wide frequency band of the model:* Response reconstruction for fatigue testing purposes can sometimes require a test band with upper limit in the range of 50 - 100 Hz. In case of shock and vibration tests the upper frequency limit can far exceed 100 Hz. This implies that the test engineer is sometimes confronted with tests requiring modelling over wide frequency bands. Ljung (1999) reported on this problem already in the first edition of his book in 1987 in the context of linear system identification, recommending to partition the frequency band in separate bands that are modelled separately. Chapter 4 lists a number of mechanisms of inaccuracy in the identification of ARX and NARX models, including mechanisms ascribed to the high relative sample frequency arising in wide band models. The motivation and (novel) development of a multi-model scheme for nonlinear inverse models for use in ILC in response reconstruction was presented in Chapter 4. The use of the multiple model approach was furthermore evaluated in the quarter vehicle road simulator and was shown to also lead to appreciable gains in the accuracy of response reconstruction.

The alternative ILC algorithm (in unmodified form), that was developed in Chapter 3, was also evaluated on the quarter vehicle road simulator, however the results were of comparable to slightly inferior accuracy to the results of the conventional algorithm. In view of the promising

results obtained with the alternative algorithm in Chapter 3, it is recommended to further evaluate the alternative algorithm, and in particular the modified form, in the response reconstruction setting in future research.

5.4 Recommendations for Future Research

Recommendations for further research on stable inversion:

- The time dependent gain employed in Mann iteration in stable inversion was found to be of value. There are many possibilities for the design of the time-dependence function, and the best approach need to be found.
- This research didn't devote much attention to the use of Ishikawa iteration in stable inversion. Further work is needed to investigate the possible utility of Ishikawa iteration in improving the results of stable inversion of NARX models, including the use of iteration dependent and time dependent gains.
- Theorem 2.3 for showing that Mann iteration converges under weaker conditions than Picard iteration may be improved with further research, possibly leading to or requiring changes to Theorem 2.2.

Recommendations for further research in ILC:

- The modified alternative ILC algorithm needs to be evaluated in comparison with the alternative and conventional algorithms in examples similar to those in Chapter 3 and in response reconstruction.
- This research showed that the use of Mann and Ishikawa iteration in stable inversion has an analogy in ILC in that the conventional and alternative ILC algorithms developed here both have parallels in the Picard and Mann iteration schemes (when using the GIB compensator). It is furthermore shown that the application of Ishikawa iteration to ILC results in novel ILC iteration schemes for both the conventional and alternative ILC algorithms. The adaptation of ILC along the lines of Ishikawa was done in Chapter 3 for the conventional algorithm (with extension to the alternative algorithm being obvious), but still needs to be evaluated in theoretical examples and practical implementation in response reconstruction.
- ILC in this research was evaluated with either constant or iteration dependent gains. The possible contribution of time-dependent ILC gains still need to be evaluated in theoretical examples and practical implementation in response reconstruction.

- The models in the stable inversion in this research are all analytical (i.e. smooth). ILC need to be investigated for inverting non-smooth models (that may be obtained by system identification but are difficult to invert) by approximating such models with smooth models (e.g. NARX models) and then using the smooth approximate models in stable inversion in ILC that is done on the non-smooth model as if on a physical system. If successful this may improve the accuracy of ILC on non-smooth physical systems by virtue of the better accuracy of the ILC-based inversion of identified non-smooth models of such systems than of stable inversion of smooth approximate models of such systems. This implies the use of ILC for purposes of inversion in the ILC compensator of ILC on a system, i.e. a nested ILC.

Recommendations for further research on response reconstruction:

- Alternative nonlinear model formulations and their identification need to be investigated, amongst others for modelling of non-smooth physical systems, possibly including neural network methods. The inversion of such alternative model formulations need to be studied and developed, possibly using ILC on the model to invert the model. This may be done along the lines of the Newton method of ILC (employing a time-varying linear inverse), or by using a simplified (smooth and/or lower degree of nonlinearity) nonlinear inverse model in the ILC compensator (that is solved with stable inversion or a third level of nested ILC).

- Investigate the effect of reduced sample frequency on the model order of a low frequency nonlinear submodel of a system in a multiple model approach to confirm that it indeed leads to lower model orders. If so this is a major advantage of the multiple model approach because the curse of dimensionality cause the size of nonlinear models to dramatically increase with the number of regressors (input and output terms). If low frequency submodels naturally involve lower model orders (fewer regressors) this will mean the typical and highly fatigue-relevant nonlinearity in low frequencies may be better modelled because nonlinear system identification is often limited by the computational intensity of searching for large numbers of potentially relevant model terms.

Appendix A

Bibliography

Ahn, H.-S., Chen, Y. and Moore, K.L. 2007. Iterative learning control: brief survey and categorization. *IEEE Transactions On Systems, Man, And Cybernetics*:37(6):1099-1121.

Arimoto, S., Kawamura, S. and Miyazaki, F. 1984. Bettering operation of robots by learning. *Journal of Robotic Systems*,1(2):123-140.

Amann, N. and Owens, D. 1994. Non-minimum phase plants in iterative learning control. *International Conference on Intelligent Systems Engineering*, Hamburg-Harburg, Germany:107-112.

Arrowsmith, D.K. and Place, C.M. 1992. *Dynamical Systems*, Chapman & Hall/CRC.

Bannantine, J.A., Comer, J.J. and Handrock, J.L. 1990. *Fundamentals of metal fatigue analysis*, Prentice Hall.

Bartle, R.G. and Sherbert, D.R. 2000. *Introduction to real analysis*, Third Edition, John Wiley and Sons.

Billings, S.A. 2013. *Nonlinear system identification*, John Wiley and Sons.

Billings, S.A. and Aguirre, L.A. 1995. Effects of sampling time on the dynamics and identification of nonlinear models. *International Journal of Bifurcation and Chaos*,5(6):1541-1556.

Burden, R.L. and Faires, J.D. 1993. *Numerical analysis*. 5th ed. PWS Publishing Company, Boston.

Castillo, B. and Di Gennaro, S. 1991. Asymptotic output tracking for SISO nonlinear discrete time systems. *Proc. of the 30th IEEE Conf. on Decision and Control*:1802-1806.

Cater, C.R. 1997. *Advances in Dynamic Response Reconstruction using non-linear time domain system identification*. Unpublished masters thesis. Pretoria: University Pretoria.

Chen, C.-T. 1984. *Linear system theory and design*. Second Edition, Harcourt Brace College Publishers.

Chen, D. and Paden, B. 1992. Stable inversion of nonlinear nonminimum phase systems. *JAPAN/USA Symposium on Flexible Automation*:1:791-797.

Chen, S. and Billings, S.A. 1989. Representations of non-linear systems: the NARMAX model. *Int. Journal of Control*:49(3):1013-1032.

Chen, S. and Billings, S.A. 1989. Orthogonal least squares methods and their application to non-linear system identification. *Int. Journal of Control*:50(5):1873-1896.

Chidume, C.E. and Chidume, C.O. 2006. Iterative approximation of fixed points of nonexpansive mappings. *Journal of Mathematical Analysis and Applications*:318:288-295.

Chidume, C.E. and Udomene, A. 2006. Strong convergence theorems for uniformly continuous pseudocontractive maps. *Journal of Mathematical Analysis and Applications*:323:88-99.

Chidume, C.E. and Osilike, M.O. 1999. Iterative solutions of nonlinear accretive operator equations in arbitrary banach spaces. *Nonlinear Analysis*:36:863-872.

Osilike, M.O. 2000. Iterative solutions of nonlinear equations of the accretive type. *Nonlinear Analysis*:42:291-300.

Craig, J.J. 1984 Adaptive control of manipulators through repeated trials. *IEEE American Control Conference*:1566-1573.

Cryer, B.W., Nawrocki, P.E. and Lund, R.A. 1976. A road simulation system for heavy duty vehicles. *SAE Transactions*, SAE paper 760361:1322-1334.

Deckers, K., Guillaume, P., Lefeber, D. and De Baere, D. 2012. Turning point based fatigue testing: combining multisines with turning point replication. *Mechanical Systems and Signal Processing*:30:23-31.

De Cuyper, J., Coppens, D., Liefooghe, C. and Debille, J. 1999. Advanced system identification methods for improved service load simulation on multi axial test rigs. *European Journal of Mechanical and Environmental Engineering*:44(1):27-39.

Devasia, S., Chen, D. and Paden, B. 1996. Nonlinear inversion-based output tracking. *IEEE Trans. on Automatic Control*:AC-41:(7):930-942.

Devasia, S. and Paden, B. 1994 Exact output tracking for nonlinear time-varying systems. *Proc. of the 33th IEEE Conf. on Decision and Control*:2346-2355.

Dobson, B. and Schwab, H. 2006. *Accelerated testing*, SAE International.

Eksteen, J. and Raath, A. 2001. Time-domain reconstruction of dynamic multiaxial responses in aeronautical fatigue testing. *Journal of Aircraft*:38(1):147-153.

Eksteen, J., Vorster, J., Grove, A.P. and Heyns, P.S. 2012. Field Measurements and Accelerated Fatigue Testing in the Laboratory on an Extra-Heavy Commercial (Class 8) Waste Collection

Vehicle. *Proc. of the 21st WasteCon Conf. and Exhibition, East London, South Africa*:333-341.

Elci, H., Longman, R.W., Phan, M., Juang, J-N. and Ugoletti, R. 1994a. Discrete frequency-based learning control for precision motion control. *Proc. IEEE Conf. on Systems, Man, and Cybernetics*, San Antonio, Texas, Oct. 1994.

Elci, H., Longman, R.W., Phan, M., Juang, J-N. and Ugoletti, R. 1994b. Automated learning control through model updating for precision motion control. *ASME J. of Adaptive Structures and Composite Materials: Analysis and Application*:AD-45/MD:54:299-314.

Elci, H., Longman, R.W., Phan, M., Juang, J-N. and Ugoletti, R. 2002. Simple learning control made practical by zero-phase filtering: applications to robotics. *IEEE Transactions on Circuits and Systems*:49(6):753-767.

Fang, Y. and Chow, T.W.S. 1998. Iterative learning control of linear discrete-time multivariable systems. *Automatica*:34(11):1459-1462.

Gao, J. and Chen, D. 1998. Iterative learning control for non-minimum phase systems. *Proc. of the American Control Conf.*

Ghosh, J. and Paden, B. 2001. Iterative learning control for nonlinear nonminimum phase plants. *ASME Journal of Dynamic Systems, Measurement and Control*:123:21-30.

Ghosh, J. and Paden, B. 2002. A pseudoinverse-based iterative learning control. *IEEE Trans. on Automatic Control*:47(5):831-837.

Ghosh, J. and Paden, B. 2004. Pseudo-inverse based iterative learning control for linear nonminimum phase plants with unmodeled dynamics. *ASME Journal of Dynamic Systems, Measurement and Control*:126:661-665.

Goldsmith, P.B. 2002. On the equivalence of LTI iterative learning control and feedback control. *Automatica*:38:703-708.

Goodwin, G.C., Graebe, S.F. and Salgado, M.E. 2001. *Control System Design*. Prentice Hall, Upper Saddle River, N.J.

Goodwin, G.C. 1985. Some observations on robust estimation and control. *Proc. of the 7th IFAC Symposium on Identification and System Parameter Estimation*, York, U.K:851-859.

Grenander, U. and Szegö, G. 1958. *Toeplitz forms and their applications*, Berkeley, Los Angeles: University of California Press.

Grimble, M. J. 1994. *Robust industrial control*. Prentice Hall International.

Gunnarsson, S. and Norrlöf, M. 1997 On the use of learning control for improved performance in robot control systems. *Proc. of the European control conf. on Decision and Control*:586-591.

Halfpenny, A. 2006. Methods for accelerating dynamic durability tests. *9th Int. Conf. on Recent*

Advances in Structural Dynamics, Southampton, UK, 2006.

Hideg, L.M. and Judd, R.P. 1988 Frequency domain analysis of learning systems. *Proc. of the 27th IEEE Conf. on Decision and Control*:586-591.

Huang, Z. 1997. A new convergence result for fixed-point iteration in bounded interval of $\Re^n$. *Computers Math. Appl.*:34(12):33-36.

Hunt, L.R. and Meyer, G. 1997. Stable inversion for nonlinear systems. *Automatica*:33(8):1549-1554.

Ishikawa, S. 1974. Fixed points by a new iteration method. *Proc. of the Am. Math. Soc.*:44:147-150.

Isidori, A. 1995. *Nonlinear control systems*, Third Edition, Springer-Verlag, London.

Jang, H.S. and Longman, R.W. 1994. A new learning control law with monotonic decay of the tracking error norm. *Proc. of the Thirty-Second Annual Allerton Conf. on Communication, Control and Computing*, Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, USA:314-323.

Jang, H.S. and Longman, R.W. 1996a. An update on a monotonic learning control law and some fuzzy logic learning gain adjustment techniques. *Advances in the Astronautical Sciences*:90:301-318.

Jang, H.S. and Longman, R.W. 1996b. Design of digital learning controllers using a partial isometry. *Advances in the Astronautical Sciences*:93:137-152.

Juang, J.-N. and Phan M.Q. 2001, *Identification and control of mechanical systems*. Cambridge University Press.

Kinosita, K., Sogo, T. and Adachi, N. 2002 Iterative learning control using adjoint systems and stable inversion. *Asian Journal of Control*:43(1):60-67.

Kreyszig, E. 1978. *Introductory functional analysis with aplications*. John Wiley and Sons.

Lee-Glauser, G,. Juang, J.-N. and Longman, R.W. 1996. Comparison and combination of learning controllers: computational enhancement and experiments. *Journal of Guidance, Control and Dynamics*:19:1116-1123.

Leontaritis, I.J. and Billings, S.A. 1985. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *Int. Journal of Control*:41(2):303-328.

Liu, K. and Miller, D.W. 1995. Time domain state space identification of structural systems. *ASME Journal of Dynamic Systems, Measurement and Control*:117:608-618.

Ljung, L. 1999. *System identification: theory for the user*. 2nd ed. Prentice Hall, Englewood Cliffs, N.J.

Longman, R.W. 2000. Iterative learning control and repetitive control for engineering practice.

Int. J. of Control:73(10):930-954.

Longman, R.W. 2002. Simple learning control made practical by zero-phase filtering: applications to robotics. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*:49(6):753-767.

Lin, T., Owens, D.H. and HÄTÖNEN, J. 2006. Newton method based iterative learning control for discrete non-linear systems. *Int. J. of Control*:79(10):12631276.

Manabe, T. and Miyazaki, F. 1994. Learning control based on local linearization by using DFT. *Journal of Robotic Systems*:11(2):129-141.

Mann, W.R. 1953. Mean value in iteration. *Proc. of the Am. Math. Soc.*:4:506-510.

Markusson, O. 2002. *Model and system inversion with applications in nonlinear system identification and control*, Doctoral thesis, Royal Institute of Technology, Stockholm, Sweden.

Mianzo, L., Fricke, D. and Chabaan, R. 1998. Road profile control methods for laboratory vehicle road simulators. *Autotestcon Proceedings*:222-228.

Monaco, S. and Normand-Cyrot, D. 1987. Minimum-phase nonlinear discrete-time systems and feedback stabilization. *Proc. of the 26th IEEE Conf. on Decision and Control*:979-986.

Moore, K.L., Dahleh, M. and Bhattacharyya, S.P. 1992. Iterative learning control: a survey and new results. *Journal of Robotic Systems*:9(5):563-594.

Nelles, O. 2001. *Nonlinear system identification.* Springer.

Nijmeijer, H. and van der Schaft, A.J. 1990. *Nonlinear dynamical control systems.* Springer-Verlag, New York.

Norrlöf, M. 2000. *Iterative learning control: analysis, design and experiments*, PhD Thesis, Linköpings Universitet, Linköping, Sweden, Dissertation No. 653, Download: http:/www/ control.isy.liu.se/ publications/.

Norrlöf, M. and Gunnarsson, S. 2002. Time and frequency domain convergence properties in iterative learning control. *Int. J. Contr.* :75(14):1114-1126.

Norrlöf, M. and Gunnarsson, S. 2002b. Experimental comparison of some classical iterative learning control algorithms. *IEEE Transactions on Robotics and Automation*:18(4):636-641.

Phan, M.Q. and Longman, R.W. 1988. A mathematical theory of learning control for linear discrete multivariable systems. *Proc. of the AIAA/AAS Astrodynamics Conference*, Minneapolis, Minnesota, USA:740-746.

Phan, M. and Longman, R.W. 1989. Indirect learning control with guaranteed stability. *Proc. of the 1989 Conf. on Information Sciences and Systems*, John Hopkins University, Baltimore, MD:125-

131.

Phan, M.Q., Longman, R.W. and Moore, K.L. 2000. Unified formulation of linear iterative learning control. *AAS/AIAA Spaceflight Mechanics Meeting*, Jan. 2000, Paper No. AAS 00-106.

Proakis, J.G. and Manolakis, D.G. 1992 *Digital signal processing principles, algorithms, and applications*. Second Edition, Macmillan Publishing Company.

Raath, A.D. 1993a. *Structural Dynamic Response Reconstruction in the Time Domain.* Unpublished doctoral thesis. Pretoria: University Pretoria.

Raath, A.D. 1993b. Service load simulation testing in the time domain. *Environmental Engineering*:6(3):8-16.

Raath, A.D. 1997. A new time domain parametric dynamic system identification approach to multiaxial service load simulation testing in components. *Int. Journal of Fatigue*:19(5):409-414.

Rugh, W. J. 1996. *Linear system theory*. Prentice Hall, Upper Saddle River.

Sarachik, P.E. and Kreindler, E. 1965. Concerning adjoints of discrete-time systems. *IEEE Trans. on Automatic Control*:10(3):350-352.

Shafiullah, A.K.M. and Wu, C.Q. 2013. Generation and validation of loading profiles for highly accelerated durability tests of ground vehicle components. *Engineering Failure Analysis*:33:116.

Silverman, L.M. 1969. Inversion of multivariable systems. *IEEE Trans. on Automatic Control*:AC-14(3):270-276.

Smolders, K., Volckaert, M. and Swevers, J. 2008., Tracking control of nonlinear lumped mechanical continuous-time systems: a model-based iterative learning approach. *Mechanical Systems and Signal Processing*:22:18961916.

Spinelli, W., Piroddi, L. and Lovera, M. 2005. On the role of prefiltering in nonlinear system identification. *IEEE Trans. on Automatic Control*:50(10):1597-1602.

Telang, N.P. and Hunt, L.R. 2001. Frequency domain computations for nonlinear steady-state solutions. *IEEE Trans. on Signal Processing*:49(8):1728-1733.

Weal, P., Liefooghe, C. and Dressler, K. 1997. Product durability engineering - improving the process. *Sound and Vibration*, 30th Anniversary Issue, January 1997:68-79.

Wang, H. and Chen, D. 2002. Robust inversion-based learning control for nonminimum phase systems. *Proc. IEEE Conf. on Systems, Man, and Cybernetics*, 2002.

Zeng, G. and Hunt, L.R. 2000. Stable inversion for nonlinear discrete-time systems. *IEEE Trans. on Automatic Control*:45(6):1216-1220.

Zheng, G.L. and Billings, S.A. 1999. Qualitative validation and generalization in non-linear system identification. *Int. Journal of Control*:72(17):1592-1608.

Zhou, K., Doyle, J.C. and Glover, K. 1996. *Robust and optimal control.* Prentice Hall, Upper Saddle River, N.J.

Zou, Q. and Devasia, S. 1999. Preview-based stable inversion for output tracking. *Proc. of the American Control Conf.*:5:3544-3548.