RAATH, ANTON DANIEL

STRUCTURAL DYNAMIC RESPONSE RECONSTRUCTION IN THE TIME DOMAIN

PhD UP 1993

Structural Dynamic Response Reconstruction in the Time Domain

by

Anton Daniel Raath

a dissertation submitted in partial fulfilment of

the requirements for the degree

Philosophiae Doctor

in the Department of Mechanical and Aeronautical Engineering

of the Faculty of Engineering

of the

University of Pretoria

November 1992

Abstract

To assist in the development of mechanical structures which are subjected to dynamic loads, structural dynamic testing, using a test rig loaded by servo-hydraulic actuators to reproduce operational measured responses in the laboratory, may form an essential element of the development process.

The input loads acting on the structure under operational conditions can in most cases not be measured directly, and instead the structural dynamic responses to these loads are recorded. The input forcing functions then need to be determined to effect a simulation of the operational conditions.

With this work, a time domain based testing system has been developed to enable the reproduction of service-acquired dynamic responses on any actual full scale structure in the laboratory, taking into account the full multiple axis dynamics of the system. The system is able to determine the input forcing functions in such a way that, when applied to the test-structure, an accurate reproduction of the in-service measured responses are reproduced on the computer controlled laboratory test rig.

The test structure is instrumented with suitable transducers which are used to record the structural dynamic response under operational conditions. The test structure is thereafter installed in a servo-hydraulic actuator test rig in the laboratory. The test rig is excited with synthetic random inputs while simultaneously recording the responses to these inputs. Using the experimental input-output data, a dynamic model of the test system is found by using parametric dynamic system identification techniques. By using the service aquired vibration responses together with the dynamic model, the system inputs may be determined. A series of iterations around this first approximation finally provides a high degree of accuracy in the simulation.

To prove the integrity of the developed system, it has been applied to a number of case studies using a variety of different engineering structures, and very accurate results were achieved.

Opsomming

In die ontwikkeling van meganiese strukture wat onderwerp word aan dinamiese belastings, vorm struktuurdinamiese toetsing, met behulp van 'n toetsopstelling wat van servo-hidrouliese aktueerders gebruik maak, 'n essensiële element.

Die struktuuropwekkingskragte kan in die meeste gevalle nie direk gemeet word nie en die gevolglike struktuurdinamiese responsies word dus gemeet. Die opwekkingskragte moet dan uit hierdie responsies afgelei word ten einde 'n simulasie van die operasionele toestande te kan uitvoer.

Met hierdie studie is 'n tyddomein gebaseerde toetstegniek ontwikkel waarmee die operasioneel gemete dinamiese responsies van enige struktuur in die laboratorium nageboots kan word, deur inagneming van die volledige multi-assige dinamika van die stelsel. Die tegniek bepaal die opwekkingskragte op so 'n wyse dat wanneer die struktuur daaraan blootgestel word, 'n akkurate simulasie van die operasionele responsie op die rekenaar beheerde multi-assige toetsopstelling in die laboratorium plaasvind.

Die struktuur word met geskikte omsetters geïnstrumenteer wat die struktuurdinamiese responsies tydens operasionele toestande opneem. Daarna word die toetsstruktuur in die laboratorium in 'n servo-hidrouliese aktueerder toetsopstelling ingebou. Die toetsopstelling word dan met behulp van sintetiese, stogastiese seine opgewek terwyl die responsies tegelykertyd gemeet word. Uit die ingangsresponsie data word 'n dinamiese model van die stelsel in die tyddomein bereken deur gebruik te maak van parametriese dinamiese stelselidentifikasietegnieke. Met behulp van die model sowel as die operasioneel gemete responsies, kan die stelselopwekkingseine bereken word. Enkele iterasies word daarna gebruik wat lei na 'n simulasie van hoë akkuraatheid.

Die integriteit van die ontwikkelde stelsel is beproef deur verskeie gevallestudies op ingenieurstrukture en akkurate resultate is verkry.

Acknowledgements

I wish to thank my supervisor Professor Stephan Heyns for the opportunities and support in making this research possible.
Professor Gilmer Klintworth for his enormous contributions in establishing the laboratory where all the experimental work was done, and for introducing me to the field of structural dynamic response reconstruction.
My colleagues at the Centre for Structural Mechanics laboratory for their unparalleled enthusiasm and support.
A word of appreciation is due to the client for partial financial support of this project.
My uncle Professor Danie Page for meticulously attending to the manuscript.
I would also like to extend my gratitude to my parents for their constant support.
Finally I am especially indebted to my family, Christa, Neill and Uhna, for their endless forbearance and encouragement, and for letting me use their time.

Contents

Chapter 1

Introduction

Dynamic synthesis, finite element techniques, fatigue prediction, modal- and structural dynamic analysis and many other related fields have today developed to an extremely high level of sophistication and accuracy. This is mainly due to the development of improved analysis algorithms and techniques together with the highly developed computer technology which allows these analysis techniques to be implemented with ease, see e.g. Coackley and Butcher [1992].

On the other hand, the design of mechanical engineering products dictates optimized designs with lower safety margins but higher reliability, in the competitive markets today, with a resultant strong emphasis towards product development, rather than a single design and manufacture.

To assist in the development process laboratory testing of full scale prototypes or components has equally grown in sophistication, enabling the accurate simulation of operational conditions in the laboratory, Lund and Donaldson [1983].

To this end the ability to recreate actual measured service conditions in a laboratory is of great value in the development of dynamically loaded structures. The applications may be varied, ranging from endurance fatigue testing of full scale structures or components, see e.g. Marsh [1988], deriving vibration test severities, see Richards [1990], studying vibration problems, optimizing vibration isolation systems and achieving a general understanding of the dynamic behaviour and resultant induced stresses of structures under actual operational conditions.

1.1 System Overview

The object is to measure the dynamic response of the test structure during operational conditions and to recreate these conditions in a laboratory on the full scale structure mounted in a test rig loaded by servo-hyraulic actuators. It is in general not possible to measure the actual loads acting on the structure under operational conditions, and instead the dynamic responses to these loads are measured at positions remote from the load inputs. These remotely measured responses are subsequently utilized to derive the system input forcing-functions, which means that this is an inverse dynamic problem.

With this work, a time domain based system has been developed to enable the reproduction of service acquired dynamic responses on any actual full scale structure in the laboratory, taking the full multiple axis dynamics of the system into account. In short, the method functions as follows:

- The structural dynamic response of the test structure is recorded with suitable transducers under operational conditions.
- The test structure is thereafter installed in a laboratory test rig and excited by servo-hydraulic actuators, using synthetically generated input signals, while simultaneously recording the responses.
- From the above experimental input and output data, a time domain based dynamic model of the complete system, including actuators, test rig and control electronics is determined, using parametric dynamic system identification techniques.
- Using the dynamic model, together with the pre-recorded operational reponses, the input forcing-functions to the system which would result in these responses are determined in an off-line manner.
- In most instances a linear dynamic model would be used, requiring a process of iterations around the initial estimate to achieve an accurate simulation of the desired operational responses.
- Although not provided for in this study, in highly nonlinear systems it is also possible to employ a specific nonlinear model based on an analytical modelling of the system at hand.

1.2 Existing Systems

Similar systems functioning in the frequency domain, and essentially performing the same function, have been in existence for a number of years. In a Phd dissertation by Dodds [1972], the frequency domain based concept was initially suggested, see also Craig [1975]. This led to an American system "Remote Parameter Control" - (RPC), which was developed by MTS Systems Corporation and is described by Klinger & Stranzenbach [1979]. A similar German system "Iterative Transfer Function Compensation" - (ITFC), was developed by Schenck AG, which is discussed by Craig [1979]. These systems are extensively used in the vehicle development industries. A system based on ITFC was also developed in South Africa by Mecalc, and is known as "Multi-Axis Structural Simulation Software" - (MAS^3).

The most recent significant development in the field of multiaxis test control is described in Fletcher [1990], where a technique was developed to accomodate more response transducers than actuators, i.e. a non-square system transfer matrix. This technique has been called global simulation. Sulisz, LaCombe & Fletcher [1992] also utilize the frequency domain approach in the "TS&SICS" software. They employed modal analysis techniques to modify the test rig to improve the dynamic characteristics of the test rig.

To the author's knowledge no such systems functioning in the time domain are in existence. The question arises why a time domain based system be used in preference to the already existing frequency domain techniques and what is to be gained by the development of such a system in the time domain ? These questions will be discussed in the next section.

1.3 Frequency vs Time Domain Modelling and Analysis

In paragraph 1.1 reference was made to a time domain model of the complete system. Traditionally frequency response functions both in the electronic and structural disciplines are described in the frequency domain.

Various considerations suggest substantial advantages in the time domain over the frequency domain. Over the past few decades the general trend in the fields of signal processing, structural dynamic synthesis, modal analysis, and control systems design has shown a marked movement towards the time domain. Today's control systems literature refers to the frequency domain techniques as "classical techniques", whereas the time domain based techniques are called "modern methods". Two fundamental reasons have essentially contributed to this trend, firstly a great deal of research has been done over the past few decades into the time domain based control systems design and analysis, especially the understanding of multivariable control systems. Secondly, the ability of today's computers to handle the more intensive time domain based calculations, has enabled the practical implementation of these techniques.

An interesting discussion is given by Söderström and Stoica [1989] who state that the nonparametric frequency domain techniques give moderately accurate results, whereas parametric time domain techniques are required for high accuracy. Burg's [1967] development of the Maximum Entropy Method (MEM) for spectral estimation as well as the AR and other time domain spectral estimators, see e.g. Priestly [1981], Kay [1988], are a clear indication of the general trend from the FFT based to the time domain based spectral estimation methodologies.

This, and other factors, which will be enlightened below, prompted the author to investigate the feasibility of such a system, and the subsequent development thereof in the time domain.

- *Periodicity of signals*: In frequency domain analysis all time signals need to be transformed to the frequency domain via an FFT algorithm, which relies on the assumption of a periodic signal. In practice, neither the input, nor response signals are periodic, and periodicity must hence be enforced by applying an appropriate weighting window, see Chatfield [1980]. This invariably results in a small distortion of the data, as well as loss of significant data, which leads to inaccuracies. On the other hand these problems are not present when analyzing in the time domain.

- *Amount of data required*: The experimental determination of frequency response functions, as is typical in modal analysis, requires a significant amount of data to ensure reliable results. Data of 10 minutes to 1 hour is not uncommon when working in the frequency domain. Reliable results have on the other hand been obtained when analysing in the time domain from as little as 20 to 60 seconds of data. This was also found in comparative studies on modal parameter estimation algorithms in both the time and frequency domains, see Galyardt and Quantz [1987]. Furthermore in the frequency domain the minimum amount of data required is one FFT length of typically 1024 data points, whereas these restrictions do not apply to the time domain, see Zhao [1985]. This is particularly advantageous when dealing with data consisting of short impulses.

- *Time versus frequency resolution*: In structural fatigue testing a high degree of accuracy in the simulation is required. In particular, the accurate simulation of operational measured stress amplitudes is essential since even small deviations in the stress amplitudes will cause significant errors in the actual endurance fatigue life. This fact may easily be verified through a Rainflow and cumulative fatigue damage analysis, see e.g. Bannantine [1990]. Accurate stress amplitudes may only be guarenteed by employing a sufficiently high sample frequency. In the frequency domain, the sample frequency is restricted by virtue of it's fixed relation to the spectral resolution. A high sample frequency may ensure a high resolution of the time signals, but will cause a poor frequency resolution, which in certain applications inhibits the use of the frequency domain techniques. The time domain on the other hand is not affected by this problem.

- *Spectral leakage*: Because of a discrete frequency spectrum which results in the leakage of specific frequencies to adjacent spectral lines, see Bendat and Piersol [1971], very low frequency components in the operational measured responses can very often not be simulated successfully in the frequency domain. This situation typically occurs on vehicles, where low frequency trends are found in the measured response data. To prevent leakage problems, it is generally required to filter out these low frequency trends when applying frequency domain techniques, see Raath and von Fintel [1989] and [1989 b]. In the time domain on the other hand, where the calculations are not frequency dependent, all frequencies can be accomodated with ease.

- *Local linearization of nonlinear systems*: A further advantage in the time domain is the ability to re-identify a dynamic model using the actual simulation input-output data, which provides for a local linearization of a nonlinear system. This technique was applied successfully in the time domain by Raath [1991 b] on a highly nonlinear rubber component. In the frequency domain however the amount of data is insufficient from which to identify the frequency transfer function, and furthermore the simulation input excitation signals do not sufficiently cover all spectral lines, resulting in a poor transfer function. This is discussed by Raath and von Fintel [1989] and [1989 b].

- *Practical noise contaminated data*: Dynamic System Identification techniques allow the process noise to be specifically modelled, resulting in a high accuracy in the identification process as discussed by Ljung [1987]. On the other hand, noise is treated in the frequency domain by simply taking a sufficient number of FFT averages, see e.g. Galyardt and Quantz [1987], Bendat and Piersol [1971].

- *Ability to accomodate nonlinear and time-varying systems*: Although this aspect has not been dealt with in the present study, the time domain is best suited to accomodate both nonlinear, see Billings and Voon [1984], Leontaritis and Billings [1985] as well as time-varying systems, Åström and Eykhoff [1971]. Nonlinear analysis cannot be dealt with in the frequency domain. Time-varying or non-stationary systems amongst others are best dealt with by adding a third, time axis onto the frequency response function, see Braun [1986].

Naturally there are also a number of disadvantages of the time domain over the frequency domain, which are listed below

- *Calculation intensity*: Time domain calculations are generally found to be more cumbersome than the frequency domain calculations.

- *Dynamic model order selection*: In implementing the time domain techniques parametric modelling requires that both the structure and order of the dynamic model be specified. In the present study the structure of the model has been generalized, and it is only required to specify the model orders. Several techniques for the estimation of the model orders are available in the literature, some of which have been implemented. In the frequency domain however no prior knowledge of model structures and orders are required, which is one of its greatest advantages. A pre-analysis in the frequency domain could provide a simple means of estimating the model orders for the subsequent time domain system identification process. Such a proposal has been detailed in section 10.2 .

1.4 Detailed System Description

The general functioning of the method has already been broadly outlined, and will now be described in more detail.

1.4.1 Recording of Operational Responses

The structure is instrumented with transducers which may be accelerometers, strain gauges, displacement transducers or any other type of transducer which is able to render a signal analogous to the vibration response of the structure. The structure is then subjected to normal operational conditions, for instance aircraft flight tests, or a vehicle is driven over the desired road conditions, while recording the vibration response using the transducers. The recorded operational responses are utilized to derive the input signals to the test rig. The minimum number of transducers required would be equal to the number of actuators to be used on the test rig. If fewer transducers than actuators are used, the system is in analogy to an over-constrained system, and cannot be solved. As far as the positioning of the transducers is concerned, each transducer should be positioned in such a way that the signal recorded by a specific transducer shows a high correlation with each corresponding intented actuator input load.

1.4.2 System Model

In the specific application, a general linear dynamic model is sought in which the user is not burdened by the requirement of specifying the analytical mechanics of the system at hand. This type of model is thus formulated to describe any linear dynamic system and is termed a “black-box” model. Figure 1.1 shows a typical laboratory test rig configuration.

Synthetic Excitation Signals

Synthetic excitation signals are generated in the computer from a prescribed power spectral density function and conceptually constitute pseudo random white noise. These signals are sent simultaneously to all actuators, thereby exciting the entire test rig, while simultaneously recording the responses using the same transducers that recorded the operational vibrations.

Dynamic System Model Identification

It is important to realize that the identified model constitutes the entire path from the computer disc file containing the synthetic excitation signals **u**, and includes the DAC interfaces, smoothing filters, analogue PID feedback control system, servo-valves, actuators, loading members, test structure, transducers, signal conditioners, anti-aliasing filters, ADC interfaces up to the digital response file **y** on disc. This is clearly depicted in figure 1.1 .

From the known synthetic excitation signals, and the measured response to these excitations, a parametric dynamic model is identified in the time domain, using Dynamic System Identification techniques. This model takes the fully coupled multiple-input multiple-output dynamics of the system into account.

1.4.3 System Model Inversion

Having identified a dynamic model which describes the input-output characteristics of the system, the model is used together with the remotely measured operational responses to calculate the system inputs, which resulted in those responses. When applied to the test rig, these inputs would result in the desired operational responses on the test structure. In System Identification, and for that matter in all dynamic systems modelling and analysis, the model describes the response behaviour, given the system inputs ($\mathbf{y} = f(\mathbf{u})$). In this case however, the inverse of this situation is desired, whereby the system responses are known, and the system inputs, which resulted in those responses, need to be determined. We therefore require an *inverse* model for this purpose, in which two possible paths may be followed. The identified dynamic model is either inverted, or alternatively

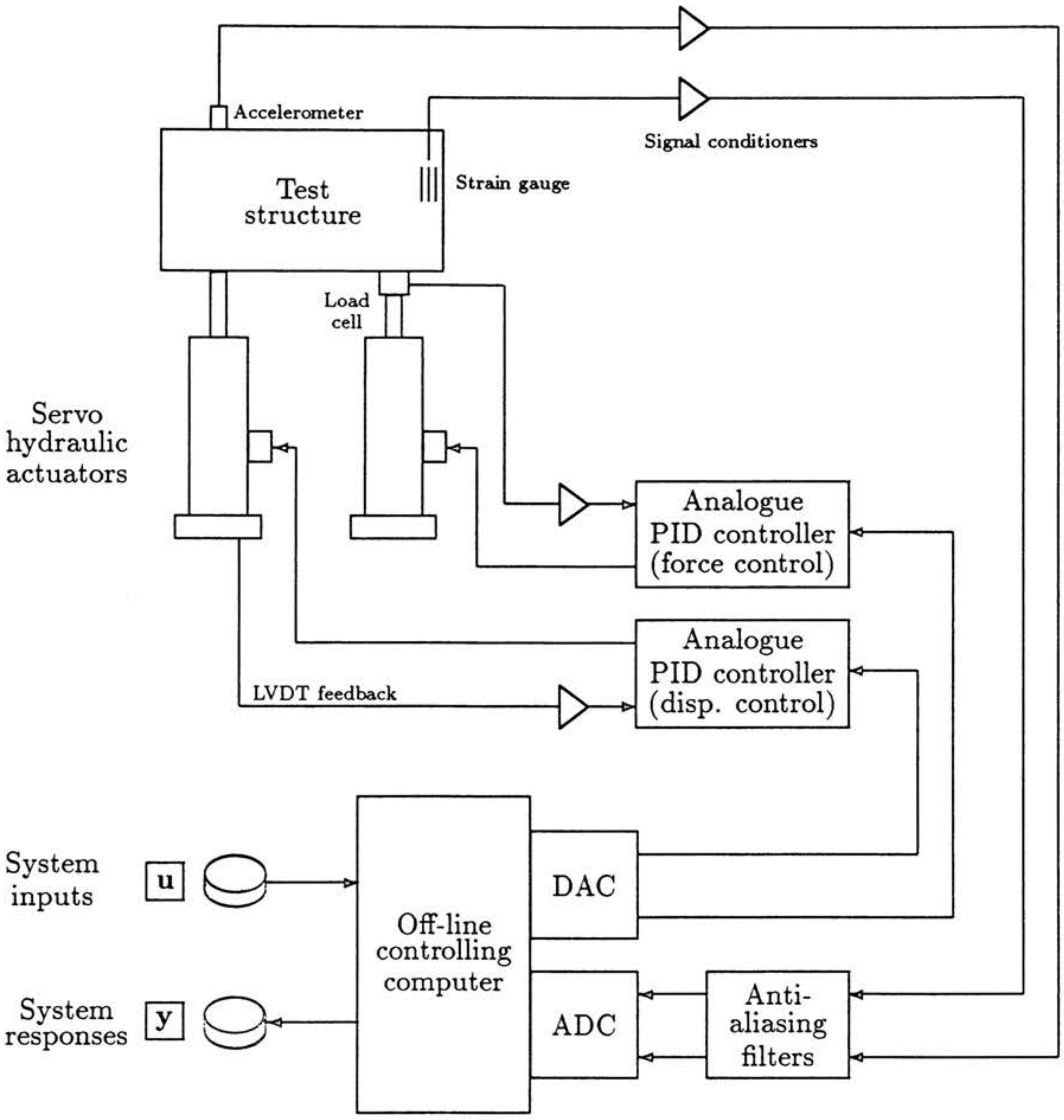


Figure 1.1: Typical test rig configuration

an inverse model is directly identified, giving a description of the system from the responses to the system inputs ($\mathbf{u} = f(\mathbf{y})$).

1.4.4 Linear Solution to System Inputs

Using the inverse model, the service acquired response data is transformed to the system input signals (or actuator drive signals), which should theoretically, when applied to the test rig, result in the desired operational responses.

1.4.5 Iteration Process

The test rig is then excited by these calculated drive signals, while the responses are again recorded. This operation should give the correct response signals had the system been linear, which is invariably not the case, and therefore requires some degree of iteration. Comparing the desired operational response to the laboratory achieved response, shows a response error, which is again input to the inverse model, and gives the required correction to the drive signals.

The original drive signals are then updated by these corrections to achieve new drive signals. This process is repeated in a few iterations until the required accuracy between the operational responses and the laboratory responses has been achieved.

1.4.6 Re-identification of System Model

The iteration process primarily makes provision for nonlinearities. As already mentioned, a linearized model around the working point may be re-identified during the iteration cycles, using the input-output data from the test rig. These input-output signals are then already close to the true system inputs and outputs and should in most instances give an improved model around the working point. This feature is a further advantage of the time domain from which it is possible to re-identify a model as opposed to the frequency domain which would need significantly more data.

1.5 Structural Fatigue Testing through Service Load Simulation Testing

The main application of structural dynamic response reconstruction is in the field of structural fatigue testing through service load simulation testing. This has also been the predominant emphasis of the applications of the developed time domain based system. A few comments regarding the background to service load simulation, especially in the vehicle durability fatigue testing environment are in order.

Practical applications

Service load simulation testing is an established field and is utilized successfully by many vehicle manufacturers. Probably one of the most comprehensive is the facility described by Petersen and Weissberger [1982]. Most of these applications are road simulators of varying complexity, see e.g. Zomotor, Schwarz and Weiler [1982]. Georgiev [1989] formalises a number of the input-output relations applicable to four channel road simulators.

Necessity for Service Load Simulation Testing

A great deal of engineering and scientific effort is spent on modelling the real world situation, and by replacing this real world situation by simplified equivalents. Yet in many situations realistic simulations of actual operational conditions are still demanded, e.g. Funk and Horst [1987],

In multiaxial loading configurations it is in general not possible to reduce the actual service histories by means of reconstruction techniques from Rainflow counting and cumulative damage calculations, such as described in e.g. Conle and Topper [1983]. The only method of testing is by a realistic service load simulation test, since the applications lead to multiaxial fatigue where the dynamics of the test structure furthermore plays a major role in the generation of reactive stress levels. Rainflow counting and cumulative damage calculations may however still be utilized to accelerate the test, but this would have to be based purely on a relative damage content of different terrain types. Under specific conditions however acceleration of simulation tests are possible. Some of these techniques are discussed by Dodds [1992] and Hurd [1992].

Digital Compensation in Single Channel Systems

The application of adaptive control for single channel servo-hydraulic testing systems is extremely valuable. Styles [1990] describes the model reference control technique which is implemented on-line, but utilizes a fixed frequency transfer function. Digital compensation using the inverse Fourier transform in the adaptive control mode is also discussed by Sherratt [1990], who points out the importance of utilizing adaptive control for testing of structures into the crack propagation regime, where the stiffness of the structure is reduced due to crack propagation. The need for these type of adaptive schemes are also becoming important in multi-actuator test rigs.

1.6 Summary

A system has been developed which enables the reproduction of operational measured responses on full scale structures using servo-hydraulic actuators. This requires a dynamic model of the system, which is identified by parametric system identification techniques. The inverse dynamic problem is encountered in the process, in which it is required to find the input forcing functions from remotely measured responses. This means that it is necessary to find an inverse solution to the dynamic model.

Similar systems have already been in existence for a number of years, which all function in the frequency domain. What makes this new development unique is essentially that it is formulated in the time domain. The inverse dynamic problem is also solved in a unique manner in this application, by reversing the data vectors, which allows the direct identification of an inverse dynamic model. The use of a time domain formulation in comparison to the frequency domain results in several advantages, which is seen as a new contribution to the field of service load simulation testing.

1.7 Document Overview

The development of this time domain structural dynamic response recontruction system has called for the synthesis of a number of disparate disciplines, amongst others dynamic system identification, structural dynamics, digital signal processing, control systems analysis, fatigue analysis and testing.

The first six chapters are utilized to develop the required mathematical techniques, while the remaining chapters are devoted to case studies.

- A short overview of the field of Dynamic System Identification is presented in Chapter 2, which establishes basic terminology from the literature. The theory in Chapter 2 in most instances applies to systems with a single input and single output. A number of key issues are proven there, which are specific to the intended application, and lay the basic foundations on which the entire study has been based.

- The specific system requirements are laid down in Chapter 3, which deals with requirements relating to the identification process, dealing with noise contaminated data, and the model requirements.

- Chapter 4 extends these techniques to the identification of multiple input-multiple output "black-box" dynamic models.

- Techniques for inverting the model, stability analysis for both normal and inverted models, together with methods of selecting the appropriate inverting technique are dealt with in Chapter 5.

- Having discussed all the building blocks in the preceding chapters, Chapter 6 is devoted to a detailed description of the system as a whole and how it is put together, and includes the iteration process. The development of computer programs is also discussed in Chapter 6.

- Several qualification tests were performed on various analytical test systems to prove the integrity of the computer programs, which are discussed in Chapter 7.

- The developed time domain techniques were applied to four practical case studies on actual laboratory test rigs, which are presented in Chapter 8.

- Comparisons to the frequency domain based techniques are drawn in Chapter 9.

- Finally, recommendations for future developments and refinements, are concluded in Chapter 10.

Chapter 2

Relevant Aspects of Dynamic System Identification

2.1 Introduction

The procedure of the proposed testing system was detailed to some extent in section 1.4. The first step in realizing such a system is the development of a dynamic model, utilizing dynamic system identification techniques. The model would be derived from experimental input-output data, obtained from the laboratory test rig. This chapter is hence a short overview of the field of dynamic system identification, and is primarily intended to establish the basic principles and terminology from the literature. In certain cases, deviations are made from the literature, to develop additional terminology and theoretical aspects which are specifically adapted to the intended aplication of simulating operational responses on a servo-hydraulic test rig. Further details on the field of dynamic system identification may be found in any textbook on the subject, see for instance Ljung [1987], Söderström and Stoica [1989], Goodwin and Payne [1977], Norton [1986], Sinha and Kuszta [1983].

Dynamic system identification involves the construction of mathematical models of dynamic systems using measured input and output data from the system. It has found application in an extremely wide field, ranging from ecology, econometrics, socio-economic systems, physics, medicine, biomedical systems, chemical processes, transportation, hydrology, electric power systems, aeronautics and astronautics.

In principle an experiment is performed on the dynamic system, from which the input and output data sequences are obtained. Using this data, a mathematical model of the system is constructed which describes the relation between the given input and output.

In many applications, no measurable input acts on the system, and it is only possible to describe the dynamic response of the system. This is called *Time Series Analysis* and is typically used in economic studies and signal processing systems.

Nature can mathematically be described by nonlinear, time dependent, partial and ordinary differential equations, with data contaminated by non-Gaussian non-stationary correlated noise. The goal of the engineer or analyst is to make sufficient and plausible assumptions to be able to describe this real world situation. It is then also the prime purpose of System Identification to construct models that are "good enough" for the specific intended purpose.

A plaudible aspect of System Identification is the ability to create a model with the basic knowledge of the mathematical formulation of the system at hand and obtaining unknown constants or parameters describing the system. This type of analysis is therefore based upon prior knowledge. On the other hand we may not know anything about the structure of the system, in which case we may construct a "black-box" model.

This chapter presents the underlying mathematical techniques for obtaining efficient and unbiased models of single-input single-output dynamic systems. The presentation is specifically adapted with the objective of proving several vital aspects, which are unique to the application of simulating operational conditions on a laboratory test rig. The resultant conclusions, are fundamental to the remaining chapters. A number of additional aspects on the background to dynamic system identification are given by Raath [1989] and [1990].

Section 2.2 gives a basic overview of linear dynamic systems with specific attention to the impulse response and the description of the noise or disturbances. The key issue of prediction is covered in section 2.3, while the different model types and model formulations are studied in section 2.4. Section 2.5 is devoted to the underlying principles of estimating the unknown model parameters and gives a discussion of some of the more popular parameter estimation techniques, which are relevant to the developed system. A summary is given in section 2.6 .

2.2 Linear Time-Invariant Dynamic Systems

Consider a linear time-invariant dynamic system with a single input $u(t)$ and single output $y(t)$. The system is said to be causal if the output at a certain time depends on the input up to that time only.

2.2.1 Impulse Response

The response of a linear, time-invariant, causal system, can be described by:

$$y(t) = \int_{\tau=0}^{\infty} g(\tau)u(t-\tau)d\tau \tag{2.1}$$

where

$$g(\tau)_{\tau=0}^{\infty} \tag{2.2}$$

is the *impulse response*, and is thus a complete dynamic characterization of the system.

In practical applications, the input-output data would be sampled, giving a discrete version of equation (2.1), namely:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) \qquad t = 0, 1, 2... \tag{2.3}$$

The above equation is valid for an input signal which remains constant between sampling intervals and is generally referred to as a *zero order hold*, which simplifies the integration of equation 2.1.

2.2.2 Disturbances

A practical system is always subjected to disturbances, which may be grouped into input disturbances or measurement noise. Input disturbances result for instance on an aircraft whose ailerons are controlled, but which is also subject to uncontrollable gusts. These input disturbances are also not directly measurable, only their effects on the output are measurable. For this reason, the totality of all disturbances and noise is lumped into an additive disturbance term $v(t)$ at the output, resulting in

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t) \tag{2.4}$$

Classical descriptions of disturbances are steps, impulses, sine waves and ramps. It is natural to employ a probabilistic description of disturbances by describing them as realizations of stochastic processes, see Papoulis [1986].

With this framework, we may describe the disturbance sequence by

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) \tag{2.5}$$

where $e(t)$ is a sequence of independent identically distributed random variables with a certain probability density function, see Ljung [1987] section 2.1. We normalize $h(k)$ such that it is *monic*, which means $h(0) = 1$, giving

$$v(t) = e(t) + h(1)e(t-1) + h(2)e(t-2) + \dots\dots\dots \tag{2.6}$$

In practice, the second order properties of the sequence $e(t)$ are described namely the mean and variance λ.

A complete description of the output of the system is thus given by

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + \sum_{k=0}^{\infty} h(k)e(t-k) \tag{2.7}$$

2.2.3 Transfer Functions

Introducing the *forward shift operator* q by

$$q\,u(t) = u(t+1) \tag{2.8}$$

and the *delay operator* by

$$q^{-1}\,u(t) = u(t-1), \tag{2.9}$$

(2.3) may be written as

$$\begin{aligned} y(t) &= \sum_{k=1}^{\infty} g(k)u(t-k) \\ &= \sum_{k=1}^{\infty} g(k)q^{-k}u(t) \\ &= [\sum_{k=1}^{\infty} g(k)q^{-k}]u(t) \\ &= G(q)u(t) \end{aligned} \tag{2.10}$$

Classical descriptions of disturbances are steps, impulses, sine waves and ramps. It is natural to employ a probabilistic description of disturbances by describing them as realizations of stochastic processes, see Papoulis [1986].

With this framework, we may describe the disturbance sequence by

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) \tag{2.5}$$

where $e(t)$ is a sequence of independent identically distributed random variables with a certain probability density function, see Ljung [1987] section 2.1. We normalize $h(k)$ such that it is *monic*, which means $h(0) = 1$, giving

$$v(t) = e(t) + h(1)e(t-1) + h(2)e(t-2) + ... \tag{2.6}$$

In practice, the second order properties of the sequence $e(t)$ are described namely the mean and variance λ.

A complete description of the output of the system is thus given by

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + \sum_{k=0}^{\infty} h(k)e(t-k) \tag{2.7}$$

2.2.3 Transfer Functions

Introducing the *forward shift operator* q by

$$q\ u(t) = u(t+1) \tag{2.8}$$

and the *delay operator* by

$$q^{-1}\ u(t) = u(t-1), \tag{2.9}$$

(2.3) may be written as

$$\begin{aligned} y(t) &= \sum_{k=1}^{\infty} g(k)u(t-k) \\ &= \sum_{k=1}^{\infty} g(k)q^{-k}u(t) \\ &= [\sum_{k=1}^{\infty} g(k)q^{-k}]u(t) \\ &= G(q)u(t) \end{aligned} \tag{2.10}$$

It is however essential that H(q) be stable; that is,

$$\sum_{k=0}^{\infty} \mid h(k) \mid < \infty \tag{2.17}$$

2.3.2 One Step Ahead Prediction of Disturbance $v(t)$

Having observed v up to time $t-1$, we wish to predict the value of $v(t)$ based on these observations.

The inherent feature of a disturbance is that its value is not known beforehand. However information about past disturbances could be important for making qualified guesses about coming values.

Since H is monic, (2.16) may be written as

$$\begin{aligned} v(t) &= e(t) + \sum_{k=1}^{n} h(k)e(t-k) & (2.18) \\ &= e(t) + h(1)e(t-1) + h(2)e(t-2) + ... + h(n)e(t-n) & (2.19) \end{aligned}$$

$v(t)$ up to time $t-1$ is known, from which $e(t)$ up to time $t-1$ may be computed. The computation of $e(t)$, given $v(t)$, is known as the invertibility of the noise model, and will be enlightened in the next section.

Having computed $e(t)$ up to time $t-1$, means that the second term of (2.19) is known at time $t-1$. Using this information, we may predict the value of $v(t)$ at time t, which is given by

$$\begin{aligned} \hat{v}(t) &= \sum_{k=1}^{n} h(k)e(t-k) & (2.20) \\ &= h(1)e(t-1) + h(2)e(t-2) + ... + h(n)e(t-n) & (2.21) \end{aligned}$$

Note that the prediction is based on a probabilistic setting, where we predict the most likely value of $v(t)$ based on the probability density function of the known values of $\{e(t)\}$ which have been calculated by the noise model inversion.

Invertibility of the Noise Model

In the previous section we required that the noise model be invertible, which means that knowing $\{v(t)\}$ up to time $t-1$, implies that we are able to calculate $\{e(t)\}$ up to time $t-1$, see Ljung [1987] section 3.2.

$H^{-1}(q)$ is consequently defined by

$$H^{-1}(q) = \frac{1}{H(q)} \tag{2.22}$$

and require that the function $1/H(z)$ be analytic in $|z| \geq 1$; i.e. it has no poles on or outside the unit circle. We can alternatively require that the function $H(z)$ has no zeros on or outside the unit circle.

2.4 Models of Linear Time-Invariant Systems

2.4.1 General

From (2.15) the description of a linear time-invariant system is given by

$$y(t) = G(q)u(t) + H(q)e(t) \tag{2.23}$$

or alternatively by

$$y(t) = [\sum_{k=0}^{\infty} g(k)q^{-k}]u(t) + [\sum_{k=0}^{\infty} h(k)q^{-k}]e(t) \tag{2.24}$$

Since it is not possible to enumerate the infinite sequences and as both $g(k)$ and $h(k)$ decay as $k \to \infty$, one chooses to work with a finite number of values. This does not mean a finite number of terms in the summations, but that they are approximated by, for instance, a rational function, which may be approximated by an infinite series.

The aim is therefore to specify the functions $G(q), H(q)$ as well as the probability density function f_e of the disturbance $e(t)$. In general a Gaussian (normal) distribution of $e(t)$ is assumed and therefore a zero mean and variance λ are specified.

$$e(t) \sim N(0, \lambda) \tag{2.25}$$

2.4.2 Generalized Linear Time Invariant Model Structure

The generalised linear time-invariant parameterized model structure takes the form

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \tag{2.26}$$

where $A(q), B(q), C(q), D(q)$ and $F(q)$ are polynomials in the delay operator q^{-1}, Söderström and Stoica [1989] section 6.2, Ljung [1987] section 4.2. Comparison with (2.23) gives a transfer function of

$$G(q) = \frac{B(q)}{A(q)F(q)} \tag{2.27}$$

and a noise model of

$$H(q) = \frac{C(q)}{A(q)D(q)} \tag{2.28}$$

2.4.3 Special Cases of the Generalized Model Structure

ARX Model

The auto regressive model with external input **ARX**, takes the form

$$A(q)y(t) = B(q)u(t) + e(t) \tag{2.29}$$

with

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + ... + a_{n_a}q^{-n_a} \tag{2.30}$$

$$B(q) = b_0 + b_1q^{-1} + b_2q^{-2} + ... + b_{n_b}q^{-n_b} \tag{2.31}$$

(Note that $A(q)$ is a monic function)
The ARX model may be represented in difference equation form as

$$\begin{aligned} y(t) &+ a_1y(t-1) + ... + a_{n_a}y(t-n_a) = \\ b_0u(t) &+ b_1u(t-1) + ... + b_{n_b}u(t-n_b) + e(t) \end{aligned} \tag{2.32}$$

The error term $e(t)$ describes that part of the output $y(t)$ which cannot be predicted from past data and is also called the *innovations* term at time t

Defining $\hat{y}(t) = y(t) - e(t)$, the one step ahead predictor for the ARX model becomes

$$\begin{aligned} \hat{y}(t) = -a_1 y(t-1) &- a_2 y(t-2) - ...... - a_{n_a} y(t-n_a) + \\ b_0 u(t) + b_1 u(t-1) &+ b_2 u(t-2) + ... + b_{n_b} u(t-n_b) \end{aligned} \qquad (2.33)$$

In terms of polynomial functions this becomes

$$\hat{y}(t) = B(q)u(t) + [1 - A(q)]y(t) \qquad (2.34)$$

ARMAX Model

The auto regressive moving average model with external input **ARMAX**, takes the form

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \qquad (2.35)$$

With noise contaminated data, this model is an improvement on the ARX model in that the noise is modelled as a moving average process.
$A(q)$ and $B(q)$ are defined as in (2.30) and (2.31) and

$$C(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + ...c_{n_c} q^{-n_c} \qquad (2.36)$$

again $C(q)$ is monic.
The ARMAX model in difference equation form is

$$\begin{aligned} y(t) &+ a_1 y(t-1) + a_2 y(t-2) + ...a_{n_a} y(t-n_a) = \\ b_0 u(t) &+ b_1 u(t-1) + b_2 u(t-2) + ... + b_{n_b} u(t-n_b) + \\ e(t) &+ c_1 e(t-1) + c_2 e(t-2) + ...c_{n_c} e(t-n_c) \end{aligned} \qquad (2.37)$$

Regarding $e(t) = y(t) - \hat{y}(t)$ as the *innovations* term, then clearly the one step ahead predictor for the ARMAX model becomes

$$\begin{aligned} \hat{y}(t) = -a_1 y(t-1) &- a_2 y(t-2) - ...... - a_{n_a} y(t-n_a) + \\ b_0 u(t) - b_1 u(t-1) &+ b_2 u(t-2) + ... + b_{n_b} u(t-n_b) + \\ c_1 e(t-1) &+ c_2 e(t-2) + ... + c_{n_c} e(t-n_c) \end{aligned} \qquad (2.38)$$

Note that in the ARX model, a term b_0 has been included in (2.31), (2.32) or (2.33). Ljung [1987] section 4.2 does *not* include such a leading b_0 term in the ARX formulation. The inclusion of this term indicates a zero-delay term, which means that a term is present in the model which relates the current input at time t namely $u(t)$ to the current output $y(t)$. This is also used by Goodwin and Payne [1977] section 4.2, where the summation also runs from 1 instead of 0 as indicated in (2.12). The inclusion of the zero delay terms b_0 also makes the analysis more complete as will be illustrated for the direct transmission matrix **D** of the state space formulation, see appendix B. The same principle has been applied to the ARMAX model.

Recursive Nature of the ARMAX Model

In the specific application of reproducing responses on a servo-hydraulic test rig, it will be shown below that the ARMAX model reduces to the ARX model. Now $e(t-1)$ is the innovation at time $t-1$ and similarly for $e(t-2)......e(t-n_c)$. These innovations terms are not directly known or measureable.
However by substituting terms like $e(t-1) = y(t-1) - \hat{y}(t-1)$, the one step ahead predictor for the ARMAX model becomes:

$$\begin{aligned} \hat{y}(t) = -a_1 y(t-1) &- ......- a_{n_a} y(t-n_a) + \\ b_0 u(t) + b_1 u(t-1) &+ ......+ b_{n_b} u(t-n_b) + \\ c_1(y(t-1) &- \hat{y}(t-1)) + c_2(y(t-2) - \hat{y}(t-2)) + \\ ......+ c_{n_c}(y(t-n_c) &- \hat{y}(t-n_c)) \end{aligned} \tag{2.39}$$

In attempting to predict the output at time t i.e. $y(t)$, not only knowledge of the *predicted outputs* ($\hat{y}$) up to time $t-1$ would be required, but also the *measured output* up to time $t-1$.

$$u(t-n_b) \quad ...... \quad u(t-2), u(t-1), u(t) \tag{2.40}$$

$$y(t-n_a) \quad ...... \quad y(t-2), y(t-1) \tag{2.41}$$

$$\hat{y}(t-n_c) \quad ...... \quad \hat{y}(t-2), \hat{y}(t-1) \tag{2.42}$$

The above would thus be required to predict $y(t)$ as $\hat{y}(t)$. This implicitly implies that the ARMAX model can only be used in a recursive fashion from one sampling instant to the next. If it is not used in a recursive manner, the ARMAX model reduces to the ARX model. When simulating the output of the ARMAX model with another input sequence than was used to estimate the model parameters, the ARMAX model also effectively reduces to the ARX model.

The ARMAX model is however a prime choice with superior performance when used in on-line digital control systems. To start the recursion at t=1, requires

$$u(1-n_b) \quad ..... \quad u(1) \tag{2.43}$$

$$y(1-n_a) \quad ..... \quad y(0) \tag{2.44}$$

In terms of polynomial functions the ARMAX one step ahead predictor becomes

$$C(q)\hat{y}(t) = B(q)u(t) + [C(q) - A(q)]y(t) \tag{2.45}$$

$$or \;\; \hat{y}(t) = \frac{B(q)}{C(q)}u(t) + [1 - \frac{A(q)}{C(q)}]y(t) \tag{2.46}$$

Other Special Cases

By employing different combinations of the polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$, other special cases of the generalized linear time-invariant model structure are obtained. Typical cases are ARARMAX, ARARX, output error (OE), and the Box-Jenkins (BJ) model, see Söderström and Stoica [1989] section 6.2.

Deterministic and Stochastic Models

In the intended application, it becomes necessary to differentiate between two model types, on the basis of whether the noise is specifically modelled or not. This has lead to the following two definitions:

Definition: A deterministic model is defined to be of the form

$$y(t) = G(q)u(t) + e(t) \tag{2.47}$$

Definition: A stochastic model is defined to be of the form

$$y(t) = G(q)u(t) + H(q)e(t) \tag{2.48}$$

The above definitions differ slightly from those traditionally found in the literature where a deterministic model is usually defined as in (2.47) but without the $e(t)$ term, see Ljung [1987] section 3.3, which should strictly speaking be referred to as $\hat{y}(t) = G(q)u(t)$, since this is an aproximation to the true system output. Goodwin and Sin [1984] section 7.4 define a *deterministic* ARMA model and a *stochastic* ARMA model with the same approach as Ljung. The desired definition in this application is however to distinguish between modelled and unmodelled noise, which leads to the above two definitions.

2.4.4 Regression Form of Generalized Model Structure

Linear Regressions

Consider the one step ahead ARX predictor (2.33).

$$\begin{aligned} \hat{y}(t) = -a_1 y(t-1) \quad &- \quad ...... - a_{n_a} y(t-n_a) + \\ b_0 u(t) + b_1 u(t-1) \quad &+ \quad .... + b_{n_b} u(t-n_b) \end{aligned} \tag{2.49}$$

By defining the regression vector $\phi(t)$ as

$$\phi(t) = [-y(t-1) - ... - y(t-n_a)\ u(t)\ u(t-1).....u(t-n_b)]^T \tag{2.50}$$

and the parameter vector

$$\theta = [\ a_1\ a_2\ .....\ a_{n_a}\ b_0\ b_1\ .....\ b_{n_b}\ ] \tag{2.51}$$

Equation (2.49) may be written as a linear regression equation

$$\hat{y}(t) = \theta^T \phi(t) = \phi^T(t)\theta \tag{2.52}$$

The linear regression form leads to a convenient least squares solution of the parameter vector, which will be enlightened in section 2.5.2 .

Pseudo Linear Regressions

In analogy to the ARX model, the one step ahead predictor for the ARMAX model from (2.39) may also be considered

$$\begin{aligned} \hat{y}(t) = -a_1 y(t-1) &- \ldots\ldots - a_{n_a} y(t-n_a) + \\ b_0 u(t) + b_1 u(t-1) &+ \ldots\ldots + b_{n_b} u(t-n_b) + \\ c_1(y(t-1) - \hat{y}(t-1)) &+ \ldots\ldots + c_{n_c}(y(t-n_c) - \hat{y}(t-n_c)) \end{aligned} \quad (2.53)$$

In this case however the regression vector ϕ is also a function of the parameter vector θ, by virtue of the $\hat{y}(t-1)\ldots\ldots\hat{y}(t-n_c)$ terms.

$$\begin{aligned} \phi(t,\theta) \;=\; & [-y(t-1)\ldots\ldots - y(t-n_a)u(t)u(t-1)\ldots\ldots u(t-n_b) \\ & (y(t-1)-\hat{y}(t-1))(y(t-2)-\hat{y}(t-2))\ldots\ldots(y(t-n_c)-\hat{y}(t-n_c)] \end{aligned} \quad (2.54)$$

together with the parameter vector

$$\theta = [a_1\ a_2\ldots\ldots a_{n_a}\ b_0\ b_1\ldots\ldots b_{n_b}\ c_1\ c_2\ldots\ldots c_{n_c}] \quad (2.55)$$

(2.53) may be written as

$$\hat{y}(t) = \phi^T(t,\theta)\theta \quad (2.56)$$

However this is no linear regression, due to the nonlinear effect of θ in the regression vector $\phi(t,\theta)$ and is referred to as a pseudo-linear regression - Ljung [1987]. In this case the least squares approach is no longer valid.

2.5 Parameter Estimation Techniques

2.5.1 General Philosophy and Classification

Having selected the model structure, the main purpose of system identification is the calculation of the unknown parameters θ_i. It will be assumed that the order of the system, i.e. the number of parameters to be determined, is known. In all parameter estimation techniques the aim is to minimize the sequence of prediction errors $\{\epsilon(t)\}$. Fundamentally two classifications result, Ljung [1987] section 7.2, namely prediction error methods (PEM) and correlation methods.

Prediction Error Methods (PEM)

A loss function or performance index is defined as

$$J_N = \frac{1}{N}\sum_{k=1}^{N}\ell[\epsilon(k)] \qquad (2.57)$$

where $\ell(.)$ is a scalar valued positive function. The parameter estimate is then obtained by minimizing J_N.

$$\hat{\theta}_N = arg\ min(J_N) \qquad (2.58)$$

This leads to a host of different techniques such as:

- Least squares (LS)
- Weighted least squares (WLS)
- Maximum likelihood parameter estimation (ML)
- Maximum a-posteriori method (MAP)

Correlation Approach

Having found a "good" model, means that the prediction error sequence should be uncorrelated with past data. The sequence $\{\epsilon(t)\}$ should thus comprise white noise with no characteristic frequency spectrum. If the prediction error sequence is correlated with past data, it means that there was more information in the data than was picked up by $\hat{y}(t)$.

To do this we select a finite dimensional vector sequence $\zeta(t)$, which is derived from the given input-output data, and determine the parameter estimate from the condition that the prediction error be uncorrelated with this sequence, see Norton [1986]. This leads to

$$\frac{1}{N}\sum_{k=1}^{N}\zeta(k)\epsilon(k) = 0 \qquad (2.59)$$

In contrast to the PEM techniques which are based on minimizing some function, the correlation techniques give the parameter estimate $\hat{\theta}$ from the solution of some equation. Typical correlation methods are:

- Instrumental variable method (IV)
- Four step instrumental variable techniques (IV4)

Minimizing the the loss function

A variety of different techniques are available for minimizing the loss function, depending on the precise model formulation. In the next paragraph it will be shown that the least squares technique applied to the ARX model leads to a convenient closed form solution. In the case of an autoregressive (AR) time series model, a Toeplitz matrix follows and the Yule-Walker equations are formed which also gives a closed solution by using the Durbin-Levinson algorithm, see Giordano and Hsu [1985] sections 3.1 and 3.2, or Bennett [1979] section 3.2.2.

In most other cases a closed sulution is not possible, and iterative numerical search schemes are required. A host of techniques are available for this purpose, the most common are the quasi-Newton methods, see Burden and Faires [1985] section 9.3, and in particular the Gauss-Newton algorithm, see Söderström and Stoica [1989] section 7.6. A popular alternative algorithm was developed by Marquardt [1963], and is often used for the ARMA model.

2.5.2 Least Squares Parameter Estimation

The LS technique employs a quadratic norm for describing the loss function, see Ljung [1987] section 7.3.

$$V_N = \frac{1}{N}\sum_{k=1}^{N}[\epsilon(k)]^2 \tag{2.60}$$

or

$$V_N = \frac{1}{2}\mathbf{e}^T(k)\epsilon(k) \tag{2.61}$$

The unique feature of the least squares method is that the loss function is quadratic and may be minimized analytically with respect to θ.

The least squares technique is applicable to those model types which may be written in the form of a linear regression. The typical linear regression model is of the ARX type.

Consider an ARX model with $n_a = n_b = n$, and parameter vector

$$\theta = [\,-a_1 \;\; -a_2 \;\cdots\; -a_n \; b_0 \; b_1 \; b_2 \;\cdots\; b_n\,] \tag{2.62}$$

and a regression vector

$$\phi(k) = [\,y(k-1)\; y(k-2)\;\cdots\; y(k-n_a)\; u(k)\; u(k-1)\;\cdots\; u(k-n_b)\,]^T \tag{2.63}$$

The predictor may therefore be written as a linear regression in the form

$$\hat{y}(k) = \phi^T\theta \tag{2.64}$$

Assume the input-output data is known from $k = 1$ to $k = N$.
From the model

$$\begin{aligned} y(k) \;\; = \;\; & -a_1 y(k-1) - a_2 y(k-2) - \;\cdots\; - a_{n_a} y(k-n_a) \\ & + b_0 u(k) + b_1 u(k-1) + \;\cdots\; b_{n_b} u(k-n_b) + e(k), \end{aligned} \tag{2.65}$$

$N - n$ equations may be written as

$$\begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} y_n & \cdots & y_2 & y_1 & u_{n+1} & \cdots & u_2 & u_1 \\ y_{n+1} & \cdots & y_3 & y_2 & u_{n+2} & \cdots & u_3 & u_2 \\ y_{n+2} & \cdots & y_4 & y_3 & u_{n+3} & \cdots & u_4 & u_3 \\ \vdots & & & \vdots & \vdots & & & \vdots \\ y_{N-1} & \cdots & & y_{N-n} & u_N & \cdots & & u_{N-n} \end{bmatrix} \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_n \\ b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} + \begin{bmatrix} e_{n+1} \\ e_{n+2} \\ e_{n+3} \\ \vdots \\ e_N \end{bmatrix}$$

or

$$\mathbf{y}_N = \phi_N \theta_N + \epsilon_N \tag{2.66}$$

Defining the loss function as

$$J_N = \frac{1}{2}\sum_{k=n+1}^{N} \epsilon^2(k) = \frac{1}{2}\epsilon^T(N)\epsilon(N) \tag{2.67}$$

Temporarily dropping the arguments,

$$\begin{aligned} J = \frac{1}{2}\epsilon^T\epsilon &= \frac{1}{2}[\mathbf{y} - \phi\theta]^T[\mathbf{y} - \phi\theta] \\ &= \frac{1}{2}[-\theta^T\phi^T\mathbf{y} + \mathbf{y}^T\mathbf{y} + \theta^T\phi^T\phi\theta - \mathbf{y}^T\phi\theta] \end{aligned} \tag{2.68}$$

We wish to estimate the parameter vector θ by minimizing the performance index J_N with respect to θ.

$$\frac{\partial J}{\partial\theta} = \phi^T\phi\theta - \phi^T\mathbf{y} = \mathbf{0} \tag{2.69}$$

From which the least squares estimate is obtained as

$$\hat{\theta} = [\phi^T\phi]^{-1}\phi^T\mathbf{y} \tag{2.70}$$

provided that the matrix $[\phi^T\phi]^{-1}$ is nonsingular.

2.5.3 Weighted Least Squares

In many applications the performance index is defined by weighing the predictions errors,

$$J_N = \frac{1}{2}\epsilon^T(N)\mathbf{W}(N)\epsilon(N) \tag{2.71}$$

using a positive definite weighting matrix $\mathbf{W}(N)$. This situation may typically arise when the noise after a certain time is reduced to a lower level than was previously measured. The remaining analysis is analogous to the ordinary least squares.

2.5.4 Least Squares Estimation in the Presence of Noise Contaminated Data

Two important aspects of the least squares estimate need to be considered, namely the bias and consistency of the estimates. From the basic model in a linear regression form we obtain

$$y(t) = \phi\theta + \epsilon(t) \tag{2.72}$$

which gives a least squares estimate

$$\hat{\theta} = [\phi^T\phi]^{-1}\phi^T\mathbf{y} \tag{2.73}$$

Substituting (2.66) into the above gives

$$\begin{aligned}\hat{\theta} &= [\phi^T\phi]^{-1}\phi^T[\phi\theta + \epsilon] \\ &= \theta + [\phi^T\phi]^{-1}[\phi^T\epsilon]\end{aligned} \tag{2.74}$$

The second term $[\phi^T\phi]^{-1}[\phi^T\epsilon]$ thus gives the error in the estimated parameter vector, which we shall denote by $\tilde{\theta}$.

We have therefore $\hat{\theta} = \theta + \tilde{\theta}$.

Taking expectations of $\tilde{\theta}$ under the assumption that ϕ is deterministic, leads to

$$\begin{aligned}E[\tilde{\theta}] &= E[[\phi^T\phi]^{-1}[\phi^T\epsilon]] \\ &= [\phi^T\phi]^{-1}\phi^T E[\epsilon] = 0\end{aligned} \tag{2.75}$$

The estimate will therefore be consistent and unbiased if

$$E[\epsilon(t)] = 0 \tag{2.76}$$

If ϕ is of a stochastic nature and it is assumed that ϕ and $\epsilon(t)$ are independent, we may write

$$E[\hat{\theta}] = \theta + E[(\phi^T\phi)^{-1}\phi^T]E[\epsilon] = \theta \tag{2.77}$$

if $E[\epsilon(t)] = 0$

In summary then: the least squares estimate will be consistent and unbiased if

- ϕ and $\epsilon(t)$ are uncorrelated and
- the expected value $E[\epsilon(t)]$ of the residual errors is zero.

The above are sufficient but not necessary conditions.

When noise is present in the system the output data is correlated with the residual errors and the least squares technique does not give an unbiased estimate. The solution to this problem is given by the instrumental variable technique which is discussed in section 2.5.7 .

2.5.5 Maximum Likelihood Parameter Estimation

The ML parameter estimation technique, which is based on a stochastic setting, is one of the most successful approaches to obtaining unbiased estimates of the parameter vector.

Once the measurements $y(1), y(2), ...y(N)$ have been made, the joint probability density function $p(\mathbf{y}|\theta)$ is a function of the unknown parameters θ only. The maximum likelihood estimate of θ is that value which maximizes $p(\mathbf{y}|\theta)$. We therefore choose θ in such a way that the observations $y(i)$ are most likely to occur. In principle, given the data $\mathbf{y}$, the global maximum of $p(\mathbf{y}|\theta)$ is found. In practice however $p(\mathbf{y}|\theta)$ is usually a complicated function of θ, making it difficult to find the global maximum. If instead the measured data set $\mathbf{y}$ is arranged to consist of a number of smaller independent sets $y_1, y_2, ...y_N$, then

$$\begin{aligned} p(y_1, y_2, ...y_N|\theta) &= p(y_1|\theta)p(y_2|\theta)...p(y_N|\theta) \\ &= \prod_{k=1}^{N} p(y_k|\theta) \qquad (2.78) \end{aligned}$$

since the observations were assumed to be uncorrelated.

This is called the likelihood function L . Finding the global maximum is still difficult from the above product of probability density functions. Taking the logarithm of (2.78) gives

$$\log(L) = \sum_{k=1}^{N} \log p(y_k|\theta) \qquad (2.79)$$

which is referred to as the log likelihood function. Since $\log(L)$ is a monotonic function of L, attaining its maximum when L is maximum, it is simpler to maximize $\log(L)$ rather than L.

The maximum likelihood estimator is thus obtained from maximizing the log likelihood function with respect to the parameter vector.

$$\frac{\partial \log L}{\partial \theta}|_{\theta=\hat{\theta}_{ML}} = 0 \tag{2.80}$$

Maximum Likelihood Estimation in the Presence of Gaussian White Noise

It will be shown that for the special case of zero-mean white Gaussian observation noise, the ML estimator reduces to the least squares estimator. The following analysis is typically applicable to the ARX model. Recall the normal (or Gaussian) distribution of a variable x as

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.81}$$

where μ and σ^2 are the population mean and variance. ($\sigma^2 = \frac{1}{N}\sum(x_i - \mu)^2$ and σ = standard deviation).

Assuming that $\{v_i\}$ is a zero-mean, (unknown variance) σ^2, Gaussian sequence $\{v_i\} \sim N(0, \sigma^2)$ uncorrelated with $\{u_i\}$, the population mean of $\{v_i\}$ is zero i.e. $E\{v_i\} = \mu = 0$, then

$$\begin{aligned}
L(v_1, v_2, ...v_N; \theta) &= f(v_1;\theta) f(v_2;\theta)......f(v_N;\theta) \\
&= [\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{v_1^2}{2\sigma^2}}][\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{v_2^2}{2\sigma^2}}]..... \\
&= (\frac{1}{\sigma\sqrt{2\pi}})^N e^{-(\frac{\sum_{i=1}^N v_i^2}{2\sigma^2})} \\
&= (\frac{1}{\sigma\sqrt{2\pi}})^N e^{-\frac{(\mathbf{v}_N^T \mathbf{v}_N)}{2\sigma^2}}
\end{aligned} \tag{2.82}$$

Recalling also the linear regression form (2.66) from which

$$\mathbf{v}_N = \mathbf{y}_N - \phi_N \theta, \tag{2.83}$$

the likelihood function can be expressed as

$$L[\mathbf{y}_N; \theta] = (\frac{1}{\sigma\sqrt{2\pi}})^N e^{-\frac{(\mathbf{y}_N - \phi\theta)^T (\mathbf{y}_N - \phi\theta)}{2\sigma^2}}. \tag{2.84}$$

Taking logarithms, the log likelihood function becomes

$$\log L[\mathbf{y}_N;\theta] = -\frac{N}{2}\log 2\pi - \frac{N}{2}\log\sigma^2 - \frac{(\mathbf{y}_N - \phi\theta)^T(\mathbf{y}_N - \phi\theta)}{2\sigma^2}. \tag{2.85}$$

Setting to zero the partial derivatives with respect to the unknown quantities θ and σ, we obtain

$$\frac{\partial \log L}{\partial\theta^T} = \frac{\partial}{\partial\theta^T}[\frac{-1}{2\sigma^2}[\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\phi\theta - \theta^T\phi^T\mathbf{y} + \theta^T\phi^T\phi\theta]] = 0 \tag{2.86}$$

$$\frac{1}{\hat{\theta}_{ML}}(\phi^T\phi\hat{\theta}_{ML} - \phi^T\mathbf{y}) = 0 \tag{2.87}$$

and

$$\frac{\partial \log L}{\partial\sigma} = -\frac{\partial}{\partial\sigma}(N\log\sigma) - \frac{\partial}{\partial\sigma}(\frac{\mathbf{v}_N^T\mathbf{v}_N}{2\sigma^2}) = 0 \tag{2.88}$$

$$\frac{N}{\hat{\sigma}_{ML}} - \frac{\mathbf{v}_N^T\mathbf{v}_N}{\hat{\sigma}_{ML}^3} = 0 \tag{2.89}$$

From (2.89) the ML estimate for the variance of the observation noise may be determined as:

$$\hat{\sigma}_{ML}^2 = \frac{1}{N}\mathbf{v}_N^T\mathbf{v}_N = \frac{1}{N}\sum_{k=1}^{N} e_k^2 \tag{2.90}$$

where

$$e_k = y_k - \phi_k^T\hat{\theta}_{ML}. \tag{2.91}$$

Furthermore, from (2.91)

$$\hat{\theta}_{ML} = [\phi^T\phi]^{-1}\phi^T\mathbf{y}. \tag{2.92}$$

It is thus seen that for the special case of zero mean white Gaussian observation noise, the maximum likelihood estimate is identical to the ordinary least squares estimate obtained in (2.70). It may therefore also be stated that the ML and LS estimators are identical when applied to an ARX model.

2.5.6 Maximum A-Posteriori (MAP) Estimate

In contrast to the ML estimator which considers the data to comprise random variables with PDF $P(y|\hat{\theta})$, the Bayesian approach considers the parameter vector itself to consist of random variables. We thus consider θ to be a random vector with a prior (i.e. before the observations have been made) probability density $P(\hat{\theta})$.

After the observations have been made, the posterior PDF $P(\hat{\theta}|y)$ is determined, given the observed data and the prior PDF $P(\hat{\theta})$.

The optimal parameter estimate is then made by maximizing the posterior PDF $P(\hat{\theta}|y)$ - that parameter vector θ which gives the most likely value given the prior information and the observed data. This is known as the maximum a-posteriori (MAP) estimate.

We therefore have

$$\hat{\theta}_{MAP} = argmax P(\hat{\theta}|y) \tag{2.93}$$

where from Bayes's rule

$$P(\hat{\theta}|y) = \frac{P(y|\hat{\theta})P(\hat{\theta})}{P(y)}. \tag{2.94}$$

Following the reasoning of the ML estimator,

$$\log P(\hat{\theta}|y) = \log P(y|\hat{\theta}) + \log P(\hat{\theta}) - \log P(y) \tag{2.95}$$

$$\frac{\partial P(y)}{\partial \theta} = 0 \tag{2.96}$$

and the MAP estimate is given by

$$\hat{\theta}_{MAP} = argmax[\log P(y|\hat{\theta}) + \log P(\hat{\theta})]. \tag{2.97}$$

The ML estimate ignores the prior information $P(\hat{\theta})$, giving

$$\hat{\theta}_{ML} = argmax[\log P(y|\hat{\theta})]. \tag{2.98}$$

Note that

$$LLF(\hat{\theta}|y) = \log P(y|\hat{\theta}) = \sum_{i=1}^{N} \log P(y_i|\hat{\theta}) \tag{2.99}$$

2.5.7 Instrumental Variable Methods

It was shown in section 2.5.4 that in the presence of noise, the least squares technique does not give unbiased estimates due to correlation between the output data and the residual errors.

From (2.69) it was required that

$$\phi^T\phi\hat{\theta} - \phi^T\mathbf{y} = 0 \tag{2.100}$$

or

$$\phi^T(\mathbf{y} - \phi\hat{\theta}) = 0. \tag{2.101}$$

However $\mathbf{y} - \phi\theta = \mathbf{v}$. Following a heuristic argument, equation (2.101) therefore requires $E[\phi^T\mathbf{v}] = 0$. This is in general not possible because ϕ and $\mathbf{v}$ are correlated. We therefore require a different vector to ϕ^T, which is not correlated with the noise $\mathbf{v}$, and consequently replace $E[\phi^T\mathbf{v}]$ by $E[\zeta\mathbf{v}]$ which will now give a zero expected value due to ζ and $\mathbf{v}$ being uncorrelated.

Returning to equation (2.101) it therefore follows that

$$\zeta^T(\mathbf{y} - \phi\hat{\theta}) = 0 \tag{2.102}$$

$$\zeta^T\mathbf{y} = \zeta^T\phi\hat{\theta} \tag{2.103}$$

or

$$\hat{\theta}_{IV} = [\zeta^T\phi]^{-1}\zeta^T\mathbf{y} \tag{2.104}$$

where

$$E[\zeta^T\mathbf{v}] = 0 \tag{2.105}$$

and $[\zeta^T\phi]$ is nonsingular.

This gives an unbiased estimate of the parameter vector $\hat{\theta}$. The elements of ζ are called the instruments or instrumental variables.

The principle may further be clarified by repeating equations (2.72) through (2.74) for the IV estimate. In this case

$$y(t) = \phi\theta + \mathbf{v}(t) \tag{2.106}$$

which gives a least squares estimate

$$\hat{\theta} = [\zeta^T\phi]^{-1}\zeta^T\mathbf{y}. \tag{2.107}$$

Substituting (2.106) into the above gives

$$\begin{aligned} \hat{\theta} &= [\zeta^T\phi]^{-1}\zeta^T[\phi\theta + \mathbf{v}] && (2.108)\\ &= \theta + [\zeta^T\phi]^{-1}[\zeta^T\mathbf{v}] && (2.109)\\ &= \theta + \tilde{\theta} && (2.110)\end{aligned}$$

It is required that

$$E[\tilde{\theta}] = E[[\zeta^T\phi]^{-1}\zeta^T\mathbf{v}] = 0 \qquad (2.111)$$

because by specific choice of ζ, $\mathbf{v}$ and ζ are independent. It follows that

$$E[\tilde{\theta}] = E[\zeta^T\phi]^{-1}\zeta^T]E[\mathbf{v}] = 0 \qquad (2.112)$$

if $E[\mathbf{v}] = 0$

In summary then the IV estimate is consistent and unbiased,

- because ζ and $\mathbf{v}$ are uncorrelated
- E[$\mathbf{v}$] is however still required to be zero.

Determining Suitable Instruments

As conditions for the choice of ζ,

- $E[\zeta^T\mathbf{v}] = 0$ and
- $[\zeta^T\phi]$ is nonsingular.

The instruments must therefore be uncorrelated with the noise, but correlated with the regression variables ϕ.

Following figure 2.1, an approximate model $M(q)/N(q)$ is generated through say an ordinary least squares analysis, i.e.

$$\begin{aligned} M(q) &\approx A(q) && (2.113)\\ N(q) &\approx B(q) && (2.114)\end{aligned}$$

The approximate model is then simulated with the input data to achieve $\zeta(t)$ which is correlated with $y(t)$ due to the common input and approximate model, but is uncorrelated with $\mathbf{v}$. The final estimate is then obtained from (2.104) .

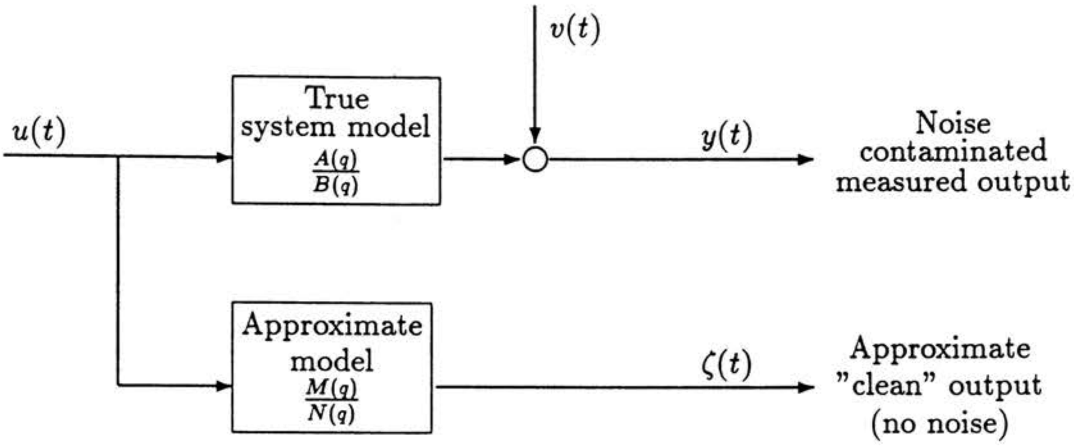


Figure 2.1: Block diagram representation of IV technique

2.6 Summary

This chapter was concerned with the underlying mathematical techniques for obtaining unbiased models of any linear dynamic system. Dynamic system identification is however an extremely wide field, and has a vast literature base. At this point thě emphasis has primarily been on single-input single-output systems, but will be extended to include multiple-input multiple-output systems in chapter 4. The bulk of the material in this chapter has been taken from the literature. In the application for controlling servo-hydraulic test rigs, a number of issues have however been uniquely established which will be shown to be fundamental to the specific application. These aspects are summarized below:

- The concepts of stochastic and deterministic models for the present application were defined in section 2.4.3.
- Under specific conditions when the prediction and measurement operations do not follow one another from one sampling instant to the next, i.e. the model is not used in a recursive fashion, stochastic models reduce to deterministic models.

- The ARX model would attempt to describe the system dynamics as well as the measurement noise, although it is not suitably structured for that purpose. In such a case, the ARMAX model would be the preferred choice.

- In the presence of Gaussian stationary white noise, the maximum likelihood estimator reduces to the ordinary least squares estimator for the ARX model.

As far as the parameter estimation techniques are concerned, the PEM methods probably give the most convenient solution. This would exclude the least squares technique because of biased estimates in the presence of correlated measurement noise. To this end the maximum likelihood estimators are better suited. The final estimate may thereafter be obtained through a follow-up using the correlation approach, and in particular the instrumental variable IV technique. Should the measurement noise levels however be relatively low, i.e. a high signal to noise level, then the least squares technique would on the other hand be employed, since the parameter estimate gives a closed solution, which does not require the minimizing of a function.

Chapter 3

Preliminary Model Requirements and Assessment

3.1 Introduction

The theory presented in Chapter 2 has also found application in the modal analysis field. Since the primary application of the present study is in structural dynamic response reconstruction, it is considered essential to firstly take a brief look at what has been done in modal analysis using time domain identification. This is briefly discussed in section 3.2.

In Chapter 2 it was shown that Dynamic System Identification has matured into a well established field of techniques. The current chapter assesses these techniques in an attempt to establish a basic model structure and parameter estimation technique, which would be tailored to suit the specific intended application of simulating operational measured responses on a laboratory test rig. Section 3.3 establishes a general philosophy of the route which is followed in developing such a dynamic model.

The general requirements of any mathematical model of a dynamic system are highly applications dependent, meaning that both the type and structure of the model are a function of the intended use of the model. Broadly speaking, this would imply that a suitable model should be found which describes the input-output behaviour well enough and which may later be inverted for subsequent calculation of the system inputs from operational measured responses. These requirements are detailed in section 3.4.

The treatment of noise in the system input and output signals is discussed in section 3.5, where it is found that a model structure which specifically models the noise may be used for the identification process per se, but cannot be used in the subsequent off-line application of the identified model.

In section 3.6 an assessment is made of the generalized model structure from equation (2.26), against the background of the established model requirements, and the ARX model is selected as the basic building block. An analogous assessment is also made in section 3.7 for the parameter estimation technique.

Since it is desired to accomodate more than one actuator on the test rig, a multiple-input multiple-output dynamic model would be required. In Chapter 4 it will be shown that the MIMO model can be made up by combining single-input single-output models. It is hence firstly required to select the specific SISO model, together with a number of other aspects. This may be done without restricting the final MIMO model. Chapter 3 is hence primarily aimed at single-input single-output systems. The requirements of finally establishing a multiple-input multiple-output dynamic model are dealt with in the next chapter.

3.2 A Brief Look at Time Domain System Identification Applications in Modal Analysis

Time domain system identification techniques have been applied to modal analysis, and because of the close relation to structural dynamic response reconstruction, these techniques will briefly be discussed.

3.2.1 Modal Analysis versus Structural Dynamic Response Reconstruction

System identification in modal analysis essentially revolves around obtaining the natural frequencies and mode shapes of a structure. The input to the structure is usually a force which may be measured via a loadcell directly at the point of load application, while the response is typically in the form of an acceleration measured on the structure, which differs from the present study of structural dynamic response reconstruction, as detailed in section 1.4.2. Modal analysis is also primarily concerned with modal mass, stiffness and damping matrices, whereas the current requirement calls for a black box model description.

3.2.2 Time Domain Structural Identification for Modal Analysis

In the application of structural modal identification in the time domain, attention has essentially been concentrated on identifying the system eigenvectors when only output, or response, data is available. Time series analysis is hence a characteristic of these methods. The response data is typically in the form of free response data.

The time domain methods utilize a least squares procedure to fit a multi-degree of freedom model to the measured response, see Cooper and Wright [1986]. Depending on the type of model used, two basic categories of methods have been utilized. One category utilizes exponentially weighted trigonometric functions which require an iterative procedure. One of these methods has become known as the Smith Least Squares method, see Smith [1984].

The second category utilizes time series modelling in the form of AR or ARMA models, see Wang and Fang [1986], Zhao-qian and Yang [1984], Yingxian [1986]. The time series analysis makes extensive use of autocorrelation and partial autocorrelation analysis to estimate model orders. The techniques are well presented by Makridakis, Wheelwright and McGee [1983] as well as Cryer [1979]. Other researchers have extended the time series metods to the state space formulation, see Helsel, Evensen and Pandit [1988].

A third category which is somewhat further removed from the current application is the multiple input Least Squares Complex Exponential or Polyreference technique, see Leuridan, Lipkens, Van der Auweraer, and Lembregts [1986].

3.2.3 Applicability to Structural Dynamic Response Reconstruction

The modal analysis applications differ in a number of respects to the present application, and are hence considered only to be related to structural dynamic response reconstruction. The main reasons are the lack of input modelling and the specific application of modal paramaters in comparison to a black box time domain model.

3.3 General Philosophy for the Development of a Dynamic Model

In the introduction to this Chapter it was pointed out that the model is applications dependent. In this case the application is the calculation of actuator input signals using the operational measured responses. This requirement places a number of restrictions on the specific model formulation, which will be discussed below.

3.3.1 The General System Identification Procedure

The system identification procedure essentially involves making qualified choices of a number of aspects:

- *Experiment design*: This means choosing the correct experimental conditions such as the type of excitation signals, the sampling frequency, the frequency bandwidth to be covered, and many other aspects, to obtain informative experimental input-output data.

- *Pre-analysis of data*: It is normally required to analyse the above input-output data by a choice of various methods.

- *Pre-processing of data*: Prior to the actual identification, a certain degree of pre-processing of the data is required such as filtering or detrending. Also here, certain choices need to be made such as filter frequencies etc.

- *Choosing the type of model set*: This essentially entails choosing between physically parameterized and black-box models, time varying and stationary models, linear and nonlinear models etc.

- *Selecting the model order*: This means selecting the number of parameters, or order of the model, to be estimated.

- *Choosing the model parameterization*: The choice of model parameterization involves choosing between ARX, ARMAX, BJ, OE etc. models. It is also further required to select the manner in which certain models are parameterized.

- *Choosing the parameter estimation algorithm*: Depending on the model type and structure, as well as a number of other aspects, a choice needs to be made about the parameter estimation algorithm, or even combinations of a few of the algorithms.

In making the above choices the following questions amongst others need to be answered:

- Is the model flexible enough ?

- Is the model too complex ?

- Which model structure of two or more candidates should be chosen ?

- Which model order of two or more candidates should be chosen ?

3.3.2 Development Philosophy

Development of a Generalized Structural Testing System

Clearly the above procedure might be considerably simplified by prior knowledge. The prime objective is to develop a generalized structural testing system. The implications of such a requirement means that the entire process should, broadly speaking, require a minimum of user inputs, and both the model type and structure should, where possible, be pre-defined. It would therefore be desirable not to burden the user with having to specify either the model type or structure.

Fixed Model Type and Structure

The requirement of a fixed model type and structure means that the model should be of a general nature, capable of catering for any specific type of dynamic test structure, without retaining unnecessary flexibility, by allowing model structures which would never be relevant for the specific intended application. It is however also essential that a fixed model structure should not restrict the ability to use the best possible model for the specific application.

Efficiency of Parameter Estimation Algorithm

In contrast to most applications in dynamic system identification, where a great deal of effort might be required in estimating the parameters in a process of estimation and cross validation, in this case, it would be required to identify the dynamic model efficiently in the shortest possible time. This may sound trivial, but would nevertheless weigh very heavily in favour of efficient parameter estimation algorithms, especially in the event of a large number of input and response channels. In structural testing using servo-hydraulic actuators, it is common even for experienced operators to adjust various parameters such as the input excitation signal amplitudes, bandwidths, excitation signal length, PID settings and a great many other variables. The objective in doing so, is to optimize the specific test rig to achieve optimal performance and an acceptable dynamic model. During this process, it would be required to fit many models, in an attempt to find the best model, hence requiring an efficient parameter estimation algorithm.

3.4 Dynamic model requirements

A multitude of different types and structures of models are generally available, as was shown in chapter 2. The intended application dictates specific requirements from the model, and furthermore serves to narrow down the number of possible model candidates. This section lays down the basic requirements of the model for the specific application.

3.4.1 General model requirements

A few general requirements could be stated as follows:

- The model should lend itself to convenient analyses to study the behaviour of the system at hand. This is not an essential requirement for structural testing, but would be required when studying characteristics like the response of the dynamic model, damping characteristics or other aspects.

- It would be required to be able to use the model for simulation purposes, because the model will be simulated with the operational measured responses to determine the system inputs.

- Since the system would operate in a time independent situation it would be natural to assume a stationary model.

3.4.2 Specific model requirements

MIMO vs SISO

A multiple-input multiple-output description is necessary since in most applications, more than one actuator would be applied to the test structure. The various input channels are in general not separable, leading in most cases to a significant amount of cross-coupling between the various inputs and outputs. This cross-coupling dependence should be fully encompassed in a completely coupled multiple-input multiple-output description of the entire system where the effect of all inputs on all outputs is taken fully into account.

Model accuracy

The model should describe the system dynamics as accurately as possible, without loss of parsimony. Since the model will primarily be used in the inverted state, an inaccurate model cannot be expected to produce accurate system inputs from measured responses. The model should however be parsimonious in the sense that by using an over specified model structure or order nothing is gained in terms of accuracy.

Modelling from physical knowledge vs Black-box models

The intention is to be able to test any type of structure without prior knowledge of the physical mechanics of the specific test structure. It is therefore desirable not to get involved in attempting to describe the physical nature of the test structure through modelling by various mechanical elements. This would necessitate the use of a general parameterized black-box type dynamic model.

Invertability

Since the prime objective is to determine the system inputs from knowledge of the system responses, an inverse model will finally be required. As will be discussed in chapter 5, the inversion of the system model very often creates an unstable inverse, which is of no use, unless pole placement or an optimal servo-system is employed. It would consequently be necessary to select a model which lends itself to convenient inversion, and furthermore ensures that a stable inverse may be achieved.

Simultaneous multiple input excitation

For a multiple-input multiple-output application it would be desirable that the dynamic model be identified through a process of simultaneous multiple input excitation, whereby all input actuators are excited at the same time. This approach is very convenient as the model identification process may be executed in a shorter time especially in the case of a high number of actuators, which is an important aspect, as was explained in section 3.2.2. The model and identification process will be required to accomodate such an excitation procedure.

Continuous vs. discrete

Since all data input to, and recorded from the system will be sampled, it would be natural to work in a discrete form, also from the model point of view. The sampling interval should in general be relatively small to ensure that the operational measured responses may be accurately simulated, especially in a structural fatigue testing application. The choice of sampling interval can however have implications on the stability of the model Åström and Wittenmark [1984].

Linear vs nonlinear models

As mentioned before, the intention is to be able to test any type of structure without prior knowledge of the physical mechanics of the specific test structure. Unfortunately, as is well known, a general nonlinear form of model does not exist. In any type of nonlinear analysis the type of nonlinearity must be known beforehand, and the dynamic modelling should be specifically formulated for the nonlinearity. It would however be undesirable to burden the user with attempting to determine which form of nonlinearity is applicable for each test structure, and a linear model is therefore rather proposed. By using a few iterations, the laboratory responses may be brought very close to the service measured responses using a linear model, which is re-identified between iterations, and is in fact a linearized model of a nonlinear system.

Linear modelling is utilized in many applications of engineering analysis. In modal analysis, linear models are applied successfully. In the application to servo-hydraulic test rigs, the iteration process is specifically intended to provide for nonlinearities. This process is also extremely effective as is shown in Raath [1991 b] where a highly nonlinear rubber mounting was tested.

3.4.3 Summarized model characteristics

From the above requirements, as well as those established in section 3.2.2, the model would be:

- Linear
- Discrete
- Multivariable
- Time invariant
- Black-box type
- Parsimonious

- Allow convenient and stable inversion.
- Allow use of the model for simulation purposes.
- Accomodate simultaneous multiple input excitation during identification.
- Lend itself to convenient analyses to study the system behaviour.
- Allow for a fixed pre-defined model type and structure, without restricting the ability to employ the best possible model.
- Employ an efficient, unbiased parameter estimation algorithm.

The above summarized requirements rule out a number of model formulations, and already narrow down the possible model candidates considerably.

3.5 Stochastic Modelling

3.5.1 General

The modelling of noise or disturbances requires specific discussion. Different methods of how noise is introduced into models is discussed by Mahmoud and Singh [1981]. The dynamic system description is a model, which describes the input-output characteristics of the system. It is important to realize that in the specific application however, the "system" refers to the entire path from an input digital disc file through the digital to analogue interfaces, smoothing filters, analogue servo-hydraulic control system, actuators, loading interfaces, test structure, response transducers and signal conditioners, anti-aliasing filters, analogue to digital interfaces up to the final digital response disc file. The "system path" is therefore from disc file to disc file, as shown in figure 1.1. The choice of a specific model structure from the family of general model structures (2.26) is, amongst others, governed by the position where the noise or disturbances enter the specific process.

3.5.2 Noise Disturbances on Servo-hydraulic Test Rigs

Input Noise

In contrast to a modal analysis where the inputs are defined as the force inputs measured at the excitation positions on the structure, in this case the system inputs will always be generated from within the controlling computer, which implies that the input signals can inherently never contain noise or disturbances. Although the excitation signals during the identification process are of a stochastic nature, they are completely known and hence do not contain any form of disturbances. Furthermore, disturbances as are generally found in controlled plants, are also not applicable, implying that general state noise will not be found.

Output Noise

On the other hand, the response transducers could pick up noise and the digitally measured response will certainly contain stochastic noise. During the dynamic system identification phase, stochastic elements should be contained in the model. This is particularly important to prevent the system dynamics from attempting to describe the stochastic part of the system, Ljung [1987].

Noise Due to Analogue Servo-valve Control

The analogue PID controllers are optimized to suit the specific test structure in such a manner that when, for instance, an actuator is driven in displacement feedback control mode, the actuator displacement follows the command displacement input as accurately as the process will allow. The errors in the specific analogue control will in general be of a stochastic nature, and may be treated as an additive noise or disturbance term in the response signals.

3.5.3 Noise Modelling Formulations

It has been stated in section 2.6 that in the presence of noise, the ARX model will attempt to model both the dynamics of the system as well as the noise. It will therefore be necessary to prevent the ARX model parameters from modelling the noise. The only method of achieving this is to model the noise explicitly, thereby achieving ARX parameters which describe the dynamics only. Noise modelling would imply the use of either of the following model types:

- An ARMAX model.
- An ARARX model.
- An ARARMAX model.
- The state space Innovations representation using a time-invariant Kalman filter.
- The directly parameterized state space Innovations formulation.
- The Maine-Iliff MMLE3 state space formulation.

The first three model types may be implemented in a single-input single-output model. The last three formulations are implemented in the complete multiple-input multiple-output state space setting and are described in section 4.6.

ARMAX Model

The ARMAX model is probably the most practical model to use since it contains a moving average modelling of the innovations or noise. In most practical cases transducer output noise is found to be of a moving average nature. The specific assumption which has to be made is that the noise is indeed of a stochastic nature.

ARARX or ARARMAX Models

If it should be found that the noise is not well modelled by a moving average process, the above two model types may be more suitable. In the intended application however, where a robust modelling tool is sought, the ARMAX model would from a practical point of view be the preferred choice.

Innovations Representation Using a Time-invariant Kalman Filter

This formulation, sometimes also referred to as the natural formulation, is complex, requiring solution of the stationary discrete Ricatti equation. Setting aside complexity, this formulation is in practice very time consuming.

The Maine-Iliff MMLE3 Formulation

The above comments would generally also apply in this case. This formulation would in general primarily be intended when modelling state space systems from physical knowledge of the actual system, see Maine and Iliff [1986].

The Directly Parameterized Innovations Formulation

This formulation is simpler in that the innovations terms are directly parameterized, Ljung [1987]. In a black-box state space setting this would certainly be a preferred formulation compared to the Innovations representation using a time-invariant Kalman filter.

3.6 Assessment of SISO Model Structures

3.6.1 Off-line Application

In contrast to on-line digital control systems, where stochastic control through state estimation is implemented, the proposed control of servo-hydraulic test rigs to reproduce operational measured responses, functions in a completely off-line mode. This fact has specific implications on a number of aspects regarding the choice and use of the model structure.

3.6.2 Reduction of Stochastic Models to Deterministic Models in an Off-line Application

It was shown in section 2.4.3 that in an off-line application the ARMAX model essentially reduces to the ARX model. When simulating an ARMAX model with data other than that used to identify the model, the ARMAX model also reduces to an ARX model. In determining the system inputs from operational measured responses, the model would indeed be simulated using the operational responses, which is not the data used in the actual identification process, and an identified ARMAX model would hence indeed reduce to an ARX model. This would also apply to any of the other stochastic model structures like the ARARX or ARARMAX models, which would reduce to a deterministic ARX model.

3.6.3 Reduction of the Generalized Linear Time-invariant Model Structure

Recalling the generalized model structure from equation (2.26),

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \tag{3.1}$$

it would hence only be required to implement the impulse response

$$G(q) = \frac{B(q)}{A(q)F(q)} \tag{3.2}$$

without the noise model. The generalized model structure for this specific application would subsequently simplify to

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + e(t) \tag{3.3}$$

which essentially narrows down to either the ARX model, with $A(q)$ and $B(q)$, or the complete above model as given by equation (3.3).

3.6.4 ARX Model

The ARX model structure is a *prime choice* from a simplicity point of view, as well as the advantage of being able to write the model as a linear regression. It has also found wide application in control systems. The most important reason for selecting the ARX model however, is the direct relation to the state space model, which is discussed in chapter 4.

3.6.5 Application of the Identified Model

The sole purpose of modelling the stochastic components of the system is to prevent the dynamic model from attempting to model the noise as well. The conclusion is hence drawn that an initial noise model should be included in the identification process. By definition, the noise and inputs do not form a causal system. Corrections for the stochastic part of the output measurement is consequently not possible. After the actual modelling process the noise model is of no further use, since the stochastic model effectively reduces to a deterministic model in an off-line situation. The stochastic part of the model is thus omitted after the actual identification, and only the deterministic system dynamics is taken into account.

3.7 Assessment of Parameter Estimation Techniques

In section 2.5.1 two classifications were established, namely the prediction error methods, and the correlation approach.

3.7.1 Bias and Consistency of the Least Squares Method

If the process noise is not true white noise, and is correlated with the outputs, the ordinary least squares method does not give consistent, unbiased results, as discussed in section 2.5.4 . In this case the instrumental variable technique is a better choice, and in particular the IV4 method. The necessity of using the instrumental variable method would depend on the noise characteristics of a particular test rig setup.

In section 3.4.3 it was mentioned that the ARX model parameters will attempt to model both the system dynamics as well as the noise. The solution to this problem was given by explicitly modelling the noise through one of the stochastic model formulations. An alternative method of overcoming this problem is to employ the instrumental variable technique instead of the least squares method. The IV technique is said to only model the system dynamics, and essentially ignores the noise, Ljung [1987].

3.7.2 Suggested Parameter Estimation Procedure

The following procedure is suggested, but may vary slightly from one test to another:

- Identify a deterministic ARX model to obtain an initial parameter vector, using the ordinary least squares technique, or where necessary, the IV4 method.

- Identify a stochastic model, like ARMAX or ARARX, using the above determined parameters for $A(q)$ and $B(q)$, while making a crude estimate for the $C(q)$ parameters, utilizing a prediction error method (PEM).

- Discard the $C(q)$ parameters, while only retaining the $A(q)$ and $B(q)$ parameters for the final deterministic ARX model.

The first step above, would only be required for large models, where it might be required to speed up the identification process.

3.8 Summary

From the vast field of dynamic system identification a number of important fundamental aspects have been established in the process of selecting both appropriate basic model structures as well as the parameter estimation techniques for the application to servo-hydraulic test rigs.

After the identification process has been completed a deterministic model is required, since a stochastic model cannot be accomodated in the off-line application of determining the system inputs from the operational measured responses. *The ARX model is consequently suggested as the basic deterministic model structure.*

During the actual identification process it will in general be required to use a basic model structure which specifically models the noise or disturbances. The reason for this requirement is that an ARX model will attempt to model the process noise if stochastic modelling terms are not provided in the model formulation. The ARMAX model structure is suggested for this purpose, however specific applications may require other structures like an ARARX model. The choice of the final basic model type to be used during the identification process, could vary from one test rig to the next, and would be governed by specific aspects such as the chosen transducers, the control mode of the actuators and the general rig configuration.

Should the ARMAX model, which consists of the three basic parameter polynomials $A(q), B(q), C(q)$, for instance have been chosen for the initial identification process, only the $A(q)$ and $B(q)$ parameters would be retained, while discarding the $C(q)$ parameters, since the noise model serves no further purpose thereafter.

As far as the parameter estimation technique is concerned, the following procedure is suggested:

- Identify a deterministic model to obtain an initial parameter vector
- Using this initial parameter vector, identify a stochastic model
- Discard the noise model, retaining an ARX model

The parameter estimation methods would be the ordinary least squares, followed by the IV approach.

Thus far no attention has been given to the requirement of a multiple-input multiple-output model. This aspect is addressed in the next chapter.

Chapter 4

Development of the MIMO Dynamic Model

4.1 Introduction

In chapter 2 relevant aspects from the dynamic system identification theory were presented. Having furthermore established the necessary model requirements in chapter 3, we are now able to develop a specific model for the intended application. From the summarized model requirements in section 3.4.3, it has already been possible to eliminate a number of possible model types and model structures, which has led to the selection of a *basic ARX model.*

Having chosen the basic ARX model structure, it becomes necessary to turn attention to the multivariable aspects. For the specific application, a fully coupled model describing the complete dynamics of the multiple-input multiple-output system is required. Various multivariable model formulations are available to do this, and are discussed in this chapter. One possibility is the matrix fraction description which may be formulated for the various basic model types and gives matrix polynomials in the delay operator q. Alternatively the pulse transfer function may be used, but this requires the use of three-dimensional matrices, which in general is inconvenient. A third alternative is the state space model formulation, which has been developed to a high degree especially in the control systems field. Texts on the state space formulation and its implications from a control systems point of view may be found in Rosenbrock [1970], Kailath [1980], Strejc [1981] and Apte [1981].

The purpose of this chapter is to evaluate the various multiple-input multiple-output model structures in order to select the most appropriate type of model and thereafter to develop the chosen model structure into an appropriate form for subsequent determination of the system inputs from the measured responses.

In section 4.3 which discusses multivariable model descriptions, it is found that the state space model structure is in general the most appropriate model to describe the general multiple-input multiple-output dynamics of the system. It is however also found difficult in practice to implement a generalized black-box state space model. It will be shown that the coupled multiple-input multiple-output dynamic system may be separated into multiple-input single-output (MISO) ARX models, which becomes possible due to the fact that the separate models are independent of one another.

These separate ARX models are then combined into a complete global state space model describing the total multiple input-output dynamics of the system.

4.2 Multivariable ARX Model Parameterizations

The general ARX model in polynomial form is given by

$$A(q)y(t) = B(q)u(t) + e(t) \tag{4.1}$$

4.2.1 Multiple-input Multiple-output Systems Modelling

Consider a simple second order system with two inputs and two outputs. Assuming an ARX structure, output y_1 would be influenced by both inputs u_1 and u_2, which may be written as

$$\begin{aligned} y_1(k) + a_{1_1} y_1(k-1) + a_{2_1} y_1(k-2) &= \\ b_{10_1} u_1(k) + b_{11_1} u_1(k-1) + b_{12_1} u_1(k-2) & \\ +b_{20_1} u_2(k) + b_{21_1} u_2(k-1) + b_{22_1} u_2(k-2) & \end{aligned} \tag{4.2}$$

The response of any linear dynamic system at time instant k, is determined by two parameters namely, the inputs, as well as the current state of the system at time instant k. Analysing equation 4.2, it is clear that output y_1 is influenced by both inputs u_1 and u_2, as well as the current and past states of y_1, by virtue of the autoregressive modelling of y_1. The state of the system is however not only described by y_1, but also by y_2, hence also necessitating the inclusion of y_2 terms in the above equation. This would give

$$\begin{aligned} y_1(k) + a_{1_1} y_1(k-1) + a_{2_1} y_1(k-2) &= \\ b_{10_1} u_1(k) + b_{11_1} u_1(k-1) + b_{12_1} u_1(k-2) & \\ + b_{20_1} u_2(k) + b_{21_1} u_2(k-1) + b_{22_1} u_2(k-2) & \\ + c_{10_1} y_2(k) + c_{11_1} y_2(k-1) + c_{12_1} y_2(k-2) & \end{aligned} \tag{4.3}$$

Similarly for the second output channel

$$\begin{aligned} y_2(k) + a_{1_2} y_2(k-1) + a_{2_2} y_2(k-2) &= \\ b_{10_2} u_1(k) + b_{11_2} u_1(k-1) + b_{12_2} u_1(k-2) & \\ + b_{20_2} u_2(k) + b_{21_2} u_2(k-1) + b_{22_2} u_2(k-2) & \\ + c_{10_2} y_1(k) + c_{11_2} y_1(k-1) + c_{12_2} y_1(k-2) & \end{aligned} \tag{4.4}$$

Note that in equation (4.3), the c parameters are used to designate correspondence with the *output* y_2, which acts as an *input* to the ARX model. The symbol "c" is traditionally utilized for the moving average modelling parameters of the ARMAX model.

4.2.2 Independence of Fully Coupled Multivariable ARX Models

In the previous section, it was established that in a two input two output system, y_1 would be determined by u_1, u_2, y_2 and similarly for the second output channel, i.e.

$$y_1 = f_1(u_1, u_2, y_2) \tag{4.5}$$

$$y_2 = f_2(u_1, u_2, y_1) \tag{4.6}$$

The identified parameter vectors for these two models are however *independent of one another*, and may be written as the following matrices:

$$\mathcal{A}_1 = \left[\begin{array}{ccc} 1 & a_{1_1} & a_{2_1} \end{array}\right] y_1$$

$$\mathcal{B}_1 = \left[\begin{array}{ccc} b_{10_1} & b_{11_1} & b_{12_1} \\ b_{20_1} & b_{21_1} & b_{22_1} \\ c_{10_1} & c_{11_1} & c_{12_1} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ y_2 \end{array}$$

and for the second output channel,

$$\mathcal{A}_2 = \left[\begin{array}{ccc} 1 & a_{1_2} & a_{2_2} \end{array}\right] y_2$$

$$\mathcal{B}_2 = \left[\begin{array}{ccc} b_{10_2} & b_{11_2} & b_{12_2} \\ b_{20_2} & b_{21_2} & b_{22_2} \\ c_{10_2} & c_{11_2} & c_{12_2} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ y_1 \end{array}$$

4.2.3 Simulation of Multivariable ARX Models

To be of any use, it would be required to be able to simulate the model, as detailed in section 3.3. The separate ARX models defined by equations (4.3) and (4.4) are difficult to simulate, since they are still coupled to one another. Values of y_2 are required in a one step ahead prediction of y_1 in equation (4.3), while the converse is true for equation (4.4). It is hence required to combine these separate ARX models in some manner to effect a convenient simulation of the complete MIMO dynamic system. It therefore becomes necessary to investigate alternative MIMO model formulations, which is done in the next section.

4.3 Multivariable Model Descriptions

Two basic formulations for multiple-input multiple-output systems are presented in this section.

4.3.1 Matrix Fraction Descriptions for Multivariable Systems

Consider a multiple-input multiple-output (MIMO) system with r inputs and p outputs. The input now becomes an r dimensional vector $\mathbf{u}(t)$ and the output a p dimensional vector $\mathbf{y}(t)$.

The matrix fraction description of the ARX model then becomes

$$\mathbf{y}(t)+\mathbf{A}_1\mathbf{y}(t-1)+\ldots\mathbf{A}_{n_a}\mathbf{y}(t-n_a) = \mathbf{B}_0\mathbf{u}(t)+\mathbf{B}_1\mathbf{u}(t-1)+\ldots\mathbf{B}_{n_b}\mathbf{u}(t-n_b)+\mathbf{e}(t) \tag{4.7}$$

Where
$\mathbf{A}_i$ - are $p \times p$ matrices and
$\mathbf{B}_i$ - are $p \times r$ matrices.
Introducing the matrix polynomials

$$\mathbf{A}(q) = \mathbf{I} + \mathbf{A}_1 q^{-1} + \ldots\ldots + \mathbf{A}_{n_a} q^{-n_a} \tag{4.8}$$

$$\textit{and}\ \mathbf{B}(q) = \mathbf{B}_0 + \mathbf{B}_1 q^{-1} + \ldots\ldots + \mathbf{B}_{n_b} q^{-n_b} \tag{4.9}$$

which means that $\mathbf{A}(q)$ is a matrix whose elements are polynomials in q^{-1}; this leads to the *Matrix Fraction Description* (MFD) [Strejc, 1981]. Defining the parameter matrix

$$\theta = [\mathbf{A}_1\ \mathbf{A}_2\ldots\ldots\mathbf{A}_{n_a}\ \mathbf{B}_0\ \mathbf{B}_1\ldots\ldots\mathbf{B}_{n_b}]^T \tag{4.10}$$

we may also write (4.7) as a linear regression

$$\mathbf{y}(t) = \theta^T\phi(t) + \mathbf{e}(t) \tag{4.11}$$

Similar MFD descriptions exist for the other model structures like ARMAX.

4.3.2 State Space Descriptions

The state space formulation, which is detailed in Appendix A, gives a very convenient formulation for multivariable MIMO systems Åström and Wittenmark [1984].

For a system with r inputs and p outputs, the continuous state space formulation becomes

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4.12}$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{4.13}$$

The state equation (4.12) may be integrated under the assumption that the input signal remains constant over a sampling period. Details of the integration may be found in Appendix A. This leads to the discrete state space description, which is given by

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) \tag{4.14}$$
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \tag{4.15}$$

4.3.3 Relation Between SISO ARX and State Space Formulations

A convenient relation exists between the ARX model and the state space formulation for single-input single-output systems Ogata [1987]. The results of this relation, applied to a third order single-input single-output ARX model, will simply be demonstrated here without proof. This relationship has been expanded to multiple-input multiple-output systems by the author, and will be discussed in detail in section 4.5, and Appendices B and C.

Consider a single-input single-output system defined by a third order ARX model. The difference equation is given by

$$y(k)+a_1y(k-1)+a_2y(k-2)+a_3y(k-3) = b_0u(k)+b_1u(k-1)+b_2u(k-2)+b_3u(k-3) \tag{4.16}$$

The corresponding state space description of the above model is given by

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} -a_1b_0 + b_1 \\ -a_2b_0 + b_2 \\ -a_3b_0 + b_3 \end{bmatrix} \begin{bmatrix} u(k) \end{bmatrix} \tag{4.17}$$

$$\begin{bmatrix} y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_0 \end{bmatrix} \begin{bmatrix} u(k) \end{bmatrix} \tag{4.18}$$

4.3.4 Assessment of Multivariable Model Descriptions and Final Choice of the State Space Formulation

It was shown above that a convenient direct relation exists between the ARX model and the state space formulation. This was shown for a SISO system, but will also be shown to hold true for a MIMO system in section 4.5. It has often been stated in the literature that the state space formulation is virtually the only convenient formulation to work with in a multivariable setting, see Ljung [1988]. A vast theory base has furthermore also been developed for the state space formulation over the past few decades, see Phillips and Nagle [1990].

The state space model structure is thus considered to be *the* formulation, and is chosen as the final model structure. The actual parameterization of the state space model however remains to be defined, which is addressed in the next section.

4.4 Multivariable State Space Model Parameterizations

The general black-box state space model parameterization takes the form

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \\ x_5(k+1) \\ x_6(k+1) \\ x_7(k+1) \\ x_8(k+1) \\ x_9(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & X & 0 & X & 0 & 0 & 0 & X \\ 0 & 0 & X & 0 & X & 0 & 0 & 0 & X \\ 0 & 0 & X & 1 & X & 0 & 0 & 0 & X \\ 0 & 0 & X & 0 & X & 0 & 0 & 0 & X \\ 0 & 0 & X & 0 & X & 1 & 0 & 0 & X \\ 0 & 0 & X & 0 & X & 0 & 1 & 0 & X \\ 0 & 0 & X & 0 & X & 0 & 0 & 1 & X \\ 0 & 0 & X & 0 & X & 0 & 0 & 0 & X \\ 0 & 0 & X & 0 & X & 0 & 0 & 0 & X \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} + \begin{bmatrix} X & X \\ X & X \\ X & X \\ X & X \\ X & X \\ X & X \\ X & X \\ X & X \\ X & X \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

(4.1

with output equation

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} + \begin{bmatrix} X & X \\ X & X \\ X & X \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{4.20}$$

where the X elements indicate parameter positions. The above structure has thus been illustrated for $n = 9$ states, $p = 3$ outputs, $r = 2$ inputs.

The parameterization is uniquely defined by the p numbers ν_i which together constitute the *multi-index* $\bar{\nu}_n$ where

$$\bar{\nu}_n = \{\nu_1, \nu_2, ...\nu_p\} \tag{4.21}$$

In this case

$$\bar{\nu}_n = \{3, 2, 4\} \tag{4.22}$$

This indicates the model orders of each channel and corresponds to the positions of the column numbers with parameters in the $\mathbf{\Phi}$ matrix of (4.19), as well as the non-zero column entries in the $\mathbf{C}$ matrix of (4.20).

The above state space parameterization is similar to Ljung [1987] Appendix 4A. The main difference being an exchange of the parameter rows with columns. This may be done since the state space representation is not unique. Ogata [1987] section 5.2 gives alternative formulations for transforming z-transform type transfer functions into a number of different state space formulations through amongst others the direct programming method and the nested programming method. Appendix B of the current document uses a modified version of these principles to expand Ogata's formulations to multivariable state space formulations.

4.4.1 State Space Model Structure Selection and Evaluation

System identification with a MIMO state space model is well established in the literature, see e.g. Milne [1988], where a predetermined model structure is selected, and all parameters are determined simultaneously. See also Ljung [1991]. Various methods for selecting the multi-indices have furthermore also been described in the literature, see Ljung [1987] appendix 4A.

What is however of concern in the state space formulation is the large number of parameters that constitute the model, and which have to be found (a total of 51 parameters in the above case). The state space identification technique could therefore be extremely cumbersome in practice, especially when the model order i.e. the multi-indices also have to be found. It is therefore essential to attempt to break the large multivariable problem down into smaller sub-problems, which is discussed in the next section.

4.5 Conversion of MIMO ARX Models to a Single State Space Description

The relation between the ARX difference equation model and the state space formulation was presented in section 4.3.3. This was shown for a SISO system according to Ogata [1987]. An analogous relation applicable to MIMO systems, has been derived by the author, and is dealt with below. This formulation leads to an observable canonical form.

4.5.1 Combination of MIMO ARX Models to Observable Canonical Form

The required combining of the separate ARX models into a single MIMO state space model in observable canonical description is derived in Appendix B. The application of this method to the two models in equations (4.3) and (4.4) leads to a restriction in that the leading c terms c_{10_1} and c_{10_2} are required to be zero in order to realize an observable canonical description. This leads to:

$$\mathbf{\Phi} = \begin{bmatrix} -a_{1_1} & 1 & c_{11_1} & . \\ -a_{2_1} & . & c_{12_1} & . \\ c_{11_2} & . & -a_{1_2} & 1 \\ c_{12_2} & . & -a_{2_2} & . \end{bmatrix} \tag{4.23}$$

$$\mathbf{\Gamma} = \begin{bmatrix} (b_{11_1} - a_{1_1}b_{10_1} + c_{11_1}b_{10_2}) & (b_{21_1} - a_{1_1}b_{20_1} + c_{11_1}b_{20_2}) \\ (b_{12_1} - a_{2_1}b_{10_1} + c_{12_1}b_{10_2}) & (b_{22_1} - a_{2_1}b_{20_1} + c_{12_1}b_{20_2}) \\ (b_{11_2} + c_{11_2}b_{10_1} - a_{1_2}b_{10_2}) & (b_{21_2} + c_{11_2}b_{20_1} - a_{1_2}b_{20_2}) \\ (b_{12_2} + c_{12_2}b_{10_1} - a_{2_2}b_{10_2}) & (b_{22_2} + c_{12_2}b_{20_1} - a_{2_2}b_{20_2}) \end{bmatrix} \tag{4.24}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} b_{10_1} & b_{20_1} \\ b_{10_2} & b_{20_2} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{4.25}$$

or

$$\mathbf{y}(k) = \mathbf{Cx}(k) + \mathbf{Du}(k) \tag{4.26}$$

The above formulation is in the well known observable canonical form.

$$\mathbf{x}(k+1) = \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) \tag{4.27}$$

$$\mathbf{y}(k) = \mathbf{Cx}(k) + \mathbf{Du}(k) \tag{4.28}$$

4.5.2 Combination of MIMO ARX Models to State Space Form

When it is essential to retain the leading c terms c_{10_1} and c_{10_2}, an observable canonical form is not possible. A state space conversion is however still possible, Strejc [1981], and is presented in Appendix C .

4.6 Noise Models in the State Space Formulation

From (2.15) we had the description of a dynamic system

$$y(t) = G(q)u(t) + H(q)e(t) \tag{4.29}$$

where the noise was described by

$$v(t) = H(q)e(t) \tag{4.30}$$

It is common for state space descriptions to split the noise term $v(t)$ into contributions from measurement noise $v(t)$ and process noise $w(t)$ acting on the states of the system see Åström and Wittenmark [1984], Goodwin and Sin [1984]. This gives

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) + \mathbf{w}(k) & (4.31)\\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) + \mathbf{v}(k) & (4.32) \end{aligned}$$

where $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are sequences of independent random variables with zero mean and covariances

$$\begin{aligned} E[\mathbf{w}(k)\mathbf{w}^T(k)] &= \mathbf{R}_1(\theta) & (4.33)\\ E[\mathbf{v}(k)\mathbf{v}^T(k)] &= \mathbf{R}_2(\theta) & (4.34)\\ E[\mathbf{w}(k)\mathbf{v}^T(k)] &= \mathbf{R}_{12}(\theta) & (4.35) \end{aligned}$$

4.6.1 Innovations Representation and the Time Invariant Kalman Filter

When the direct transmission matrix $\mathbf{D} = [0]$, the output prediction error of (4.32) leads to $e(t) = y(t) - C\hat{x}(t)$ by virtue of (4.30), where $\hat{x}(t)$ is the estimate of the states at time t. The state space formulation together with the covariances may be used in a predictor Ljung [1987], given by

$$\begin{aligned} \hat{\mathbf{x}}(t+1) &= \mathbf{\Phi}\hat{\mathbf{x}}(t) + \mathbf{\Gamma}u(t) + K[y(t) - \mathbf{C}\hat{\mathbf{x}}(t)] \quad (4.36) \\ \hat{y}(t) &= \mathbf{C}\hat{\mathbf{x}}(t) \quad (4.37) \end{aligned}$$

where K is known as the Kalman gain, and is obtained from

$$K = [\mathbf{\Phi}PC^T + R_{12}][CPC^T + R_2]^T \quad (4.38)$$

and P is the positive semi definite solution to:

$$P = \mathbf{\Phi}P\mathbf{\Phi}^T + R_1 - [\mathbf{\Phi}PC^T + R_{12}][CPC^T + R_2]^{-1}[\mathbf{\Phi}PC^T + R_{12}]^T \quad (4.39)$$

The above equation is known as the stationary discrete matrix Ricatti equation.

The covariance of the state estimation error is the matrix P, therefore

$$\mathbf{P} = E[\mathbf{x}(t) - \hat{\mathbf{x}}(t)][\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^T \quad (4.40)$$

The prediction error denoted by e(t) is given by

$$e(t) = y(t) - C\hat{x}(t) \quad (4.41)$$

and is that part of the output which cannot be predicted from past data i.e. the innovation. (4.36) and (4.37) may now be written as

$$\begin{aligned} \hat{\mathbf{x}}(t+1) &= \mathbf{\Phi}\hat{\mathbf{x}}(t) + \mathbf{\Gamma}u(t) + K[e(t)] \quad (4.42) \\ y(t) &= \mathbf{C}\hat{\mathbf{x}}(t) + e(t) \quad (4.43) \end{aligned}$$

The innovations covariance is given by

$$E[\mathbf{e}(t)\mathbf{e}^T(t)] = \mathbf{R} = CPC^T + R_2 \quad (4.44)$$

Equations (4.42) and (4.43) are referred to as the innovations formulation of the state space description, because the innovations term appears explicitly.

4.6.2 Directly Parameterized Innovations Representation

A convenient simplified form of the innovations formulation is obtained by directly parameterizing K in terms of θ. This is called the *directly parameterized innovations form.* Åström and Wittenmark [1984]. This means that the R matrices are no longer required and the solution of the discrete Ricatti equation is also alleviated. General black box models are essentially directly parameterized innovations models.

4.6.3 Omitting the Noise Model

As mentioned in section 3.5.5, the noise model is discarded after the actual identification process, since it serves no further purpose in the inverted model. Two alternatives are available namely:

- Implement a noise model in each multiple-input single-output model. The stochastic part is then dropped before converting each model to the final multiple-input multiple-output state space model.

- Implement a noise model in each multiple-input single-output model, convert these models to state space, retaining the stochastic parts. The state space model then contains a stochastic model which is then refined in ă state space system identification process. Only thereafter is the noise model dropped.

The latter alternative would however burden the user with specifying the innovations covariance matrix which would be undesirable, and the former solution will therefore further be pursued.

4.7 Summary

Summarizing what has been achieved up to this point: From the large field of dynamic system identification, the variety of possible structures has been narrowed down by selecting the ARX model as the basic identification process. This is achieved by a prior identification of a stochastic model such as the ARMAX model, whereafter only the $A(q)$ and $B(q)$ parameters are retained for the subsequent deterministic ARX model.

The state space formulation has been chosen as the final MIMO model structure. The complex multivariable problem has further been broken down into smaller problems by identifying a separate ARX model for each output channel, and thereafter combining these models into a complete coupled multiple-input multiple-output state space model.

The multivariable state space model can now be further developed into such a form as is required to determine the system inputs from the known operational responses. This means that the identified model must be inverted to achieve the final inverse multiple-input multiple-output state space model. These aspects are dealt with in the next chapter.

Chapter 5

Inversion of the dynamic model

5.1 Introduction

This chapter forms the final stage in the development of an appropriate model for the specific application. Having found a multivariable state space model which gives a complete description of the dynamic input-output behaviour of the system, the operational measured responses may be used together with the model to calculate the system inputs or actuator drive signals, which will then result in the desired responses.

To achieve this objective requires an *inverse model*, i.e. given the system responses, the inverse model may be simulated to give the system inputs which caused those responses. In the frequency domain, the frequency response function matrix is inverted, see Allemang [1980], Craig [1979], Klinger & Stranzenbach [1979]. Alternative methods such as modal coordinate transformation have also been utilized to find excitation forces from response measurements, see Desanghere and Snoeys [1985].

A multitude of different techniques may be employed in achieving the final fully coupled inverse state space model, see Raath [1991 c]. Inversion of the dynamic model is however not a trivial exercise since the inverse model is more often than not found to be unstable and can therefore not be simulated. Pre-analysis of the eigenvalues of the state matrix provides an indication of the stability properties of the identified system. To alleviate the problem of instability, a number of different inversion techniques may be employed, at least one of which should provide a stable inverse. Unstable inverses are not found in the frequency domain, although other problems arise in the frequency domain such as numerical instability near resonances and singular spectral lines i.e. singular FRF matrices at specific spectral lines, see e.g. Desanghere and Snoeys [1985].

If, during the identification of the dynamic model, one could simply exhange the inputs and outputs, it should then be possible to identify an inverse model directly. There is however a problem with such a scheme, in as much as the causality would be violated (see section 2.2) ie. the system would be expected to respond before any excitation had been applied. By reversing the input and output data sequences, the system would be able to react correctly, and the causality would again be restored. This proposal has been implemented, and is discussed in section 5.7.2. Apart from its simplicity it has furthermore proved to solve model instability problems, which will be enlightened in section 5.4.2.

The general philosophy and classification of inversion techniques and their formulations has been developed in section 5.2. The key issue of stability, which fundamentally decides which inversion algorithm is to be applied, is discussed in section 5.3, while the various inversion techniques and formulations are presented in sections 5.4 through 5.9 . Finally guidelines as to which algorithm to select, are summarized in section 5.10 .

The theoretical developments in sections 5.2, 5.4, 5.6 and 5.7 are novel, while section 5.5 is taken from Strejc [1981] and is presented for the sake of completeness. The theory in section 5.9 is a slightly modified version of a presentation from Ogata [1987] and is well established in the control systems literature, but the application thereof in this specific form is novel.

5.2 General Philosophy and Classification of Inverse Models

5.2.1 General Philosophy

Point of Inversion

The process of obtaining the final inverse state space model may be broken down into three fundamental phases, namely

- The identification phase
- The conversion to state space phase
- The inversion phase

It is however important to appreciate that the order of applying the above three phases need not necessarily be as given above. This considerably broadens the scope of possible inverse techniques and formulations by applying the inversion phase at different points in the complete process.

- Firstly the following method may be pursued:
  - Identify an ARX model for each output channel
  - Combine each of the above models to one multivariable state space model
  - Invert the state space model
- Secondly the path below may be followed:
  - Identify an ARX model for each output channel
  - Invert each model to give an inverse ARX model
  - Combine each of the above inverse models to one multivariable state space model
- Alternatively the inversion process may even be implemented during identification:
  - Identify a direct inverse ARX model for each input channel
  - Combine each of the above inverse models to one multivariable state space model

If the true order of the system is known, then the above three alternatives will give the same final resultant inverse state space model.

Type of Inverse Model

It is furthermore possible to formulate a number of different inverse models, which all serve the same purpose. The reason for such a pursuance is the ability to find alternative stability properties for the different inverse models, which all represent an inverse description of the same dynamic system.

5.2.2 Classification of Inverse Model Types

The above mentioned formulations may be classified according to the order of the data, and are detailed below.

Models having identical input-output characteristics but exhibiting different stability characteristics, may be formulated by inverting the order of the data. Consider for instance the input vector **u** consisting of N data points. The reversed data vector $\tilde{\mathbf{u}}$ is then defined as follows:

$$\begin{aligned} u(1) &= \tilde{u}(N) \qquad (5.1)\\ u(2) &= \tilde{u}(N-1) \\ &\vdots \\ u(k-2) &= \tilde{u}(r) \\ u(k-1) &= \tilde{u}(r-1) \\ u(k) &= \tilde{u}(r-2) \\ &\vdots \\ u(N-1) &= \tilde{u}(2) \\ u(N) &= \tilde{u}(1) \end{aligned}$$

A similar reversal of the output vector may be applied. By introducing the new index r , the reversed data vector may be conveniently used in the standard ARX model formulations.

The classification may best be done by the following definitions.

- *Definition* : A "Normal" model is defined such that, given the system inputs, the system responses may be simulated.

- *Definition* : An "Inverse" model is defined such that, given the system responses, the system inputs may be simulated.
- *Definition* : A "Forward" model is defined such that, when simulating the model, the data is in its natural state.
- *Definition* : A "Reversed" model is defined such that, when simulating the model, the order of the data is reversed.

The above definitions may now be combined to give the following classification of model types namely

- Normal models
- Forward Inverse models (fi)
- Reversed Inverse models (ri)

Other combinations of the above definitions may of course also be possible, but would make little sense, and the above three model classes are the only practical model types to formulate.

5.3 Stability Analysis of Multivariable Dynamic Systems

Following heuristic arguments, unstable dynamic models practically relate to an ever increasing output for a given bounded input. For a system which has an unstable inverse, the input shows an ever increasing amplitude when simulated from a bounded response. It may for instance be found that when simulating a model in the forward direction, i.e. determining the response output from a given input, that the system behaves perfectly well. However if the system has an unstable inverse, the input will diverge when simulating the inverted model with the given response data. When studying the problem from a numerical analysis point of view, it is found that instability practically stems from a poorly conditioned numerical problem, with rounding-off errors being one of the major causes of numerical instability.

Two fundamental issues need to be addressed: namely how do we determine the stability of the inverted model from an analysis of the normal model, and what can be done about the problem of instability. Several aspects need to be described beforehand to appreciate the key issues of stability.

5.3.1 Solution to the Discrete State Equations

Consider a system described by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) &\quad (5.2) \\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) &\quad (5.3) \end{aligned}$$

Given the initial state vector $\mathbf{x}(0)$, and assuming that $\mathbf{u}(0) = \mathbf{0}$, the states may be propagated as follows,

$$\begin{aligned} \mathbf{x}(1) &= \mathbf{\Phi x}(0) \\ \mathbf{x}(2) &= \mathbf{\Phi x}(1) + \mathbf{\Gamma u}(1) \\ &= \mathbf{\Phi}^2\mathbf{x}(0) + \mathbf{\Gamma u}(1) \\ \mathbf{x}(3) &= \mathbf{\Phi x}(2) + \mathbf{\Gamma u}(2) \\ &= \mathbf{\Phi}^3\mathbf{x}(0) + \mathbf{\Phi\Gamma u}(1) + \mathbf{\Gamma u}(2) \\ &\vdots \\ \mathbf{x}(k) &= \mathbf{\Phi}^k\mathbf{x}(0) + \mathbf{\Phi}^{k-2}\mathbf{\Gamma u}(1) + \mathbf{\Phi}k - 3\mathbf{\Gamma u}(2) + \cdots + \mathbf{\Gamma u}(k-1). \\ &= \mathbf{\Phi}^k\mathbf{x}(0) + \sum_{j=0}^{k-1} \mathbf{\Phi}^{k-j-1}\mathbf{\Gamma u}(j) &\quad (5.4) \end{aligned}$$

The state vector at any instant k is thus a function of the initial states and all inputs up to point $k-1$. Making use of the definition of the controllability matrix W_c from Appendix A, equation (5.4) may also be written as

$$\mathbf{x}(n) = \mathbf{\Phi}^n\mathbf{x}(0) + \mathbf{W}_c\mathbf{u}. \quad (5.5)$$

5.3.2 Diagonal Form of the Discrete State Equations

Returning the attention to stability, assume that $\mathbf{\Phi}$ has distinct eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$, where n is the order of the system. Then there exists a transformation matrix $\mathbf{T}$ such that $\mathbf{\Phi}$ may be diagonalized

$$\mathbf{T\Phi T^{-1}} = \begin{bmatrix} \lambda_1 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \cdot \\ & & & & \lambda_n \end{bmatrix}$$

Comparing (5.4) or (5.5) with the above, Φ^k is found to diverge if the eigenvalues are such that $|\ \lambda_i\ | \geq 1$. Since the eigenvalues of the state matrix may also be complex, this means that for stability, the eigenvalues are required to be inside the unit circle on the complex plane.

5.3.3 Definitions of Stability

Various forms of stability may be defined for state space systems:

Poles and Zeros of Discrete Multivariable Systems

A total of eight different types of poles and zeros of discrete multivariable systems exist Patel and Munro [1982]. Only two of these which are relevant to the determination of stability of models and their inverses will be discussed.

Poles

In discrete multivariable systems the poles are given by the eigenvalues of the discrete state matrix $\mathbf{\Phi}$.

Transmission Zeros

The transmission zeros of a discrete multivariable state space system are calculated by solving the pseudo eigenvalue problem.

Poles and Transmission Zeros of Models and their Inverses

The poles and zeros of a Laplace form transfer function are given by the zeros of the numerator and denominator respectively. When inverting the transfer function the poles and zeros interchange. Likewise the poles and transmission zeros of multivariable state space systems, similarly interchange. This applies only to the forward inverse model.

Therefore the stability of a forward inverse model may be evaluated through the poles of the forward inverse model, or alternatively by the transmission zeros of the normal model.

In the case of reversed inverse models, the model is first inverted and its stability evaluated by the poles.

5.4 Inversion of Multivariable ARX Models

The following simple analysis, applied to a two channel system, illustrates the inversion of multivariable ARX models. Conversion to both forward inverse and reversed inverse models are presented.

For the first output channel assume a third order ARX model with two delays on both inputs, resulting in the following parameters:

$$\mathcal{A}_1 = \left[\begin{array}{cccc} 1 & a_{1_1} & a_{2_1} & a_{3_1} \end{array}\right] y_1$$

$$\mathcal{B}_1 = \left[\begin{array}{cccc} & & b_{12_1} & b_{13_1} \\ & & b_{22_1} & b_{23_1} \\ c_{10_1} & c_{11_1} & c_{12_1} & c_{13_1} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ y_2 \end{array}$$

Similarly for the second output channel a second order model with one delay on each input channel may be assumed, resulting in

$$\mathcal{A}_2 = \left[\begin{array}{ccc} 1 & a_{1_2} & a_{2_2} \end{array}\right] y_2$$

$$\mathcal{B}_2 = \left[\begin{array}{ccc} & b_{11_2} & b_{12_2} \\ & b_{21_2} & b_{22_2} \\ c_{10_2} & c_{11_2} & c_{12_2} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ y_1 \end{array}$$

The corresponding difference equations are given by

$$\begin{aligned}
y_1(k) + a_{1_1} y_1(k-1) + a_{2_1} y_1(k-2) + a_{3_1} y_1(k-3) &= \\
b_{12_1} u_1(k-2) + b_{13_1} u_1(k-3) & \\
+ b_{22_1} u_2(k-2) + b_{23_1} u_2(k-3) & \\
+ c_{10_1} y_2(k) + c_{11_1} y_2(k-1) + c_{12_1} y_2(k-2) + c_{13_1} y_2(k-3) & \qquad (5.6)
\end{aligned}$$

and

$$\begin{aligned}
y_2(k) + a_{1_2} y_2(k-1) + a_{2_2} y_2(k-2) &= \\
b_{11_2} u_1(k-1) + b_{12_2} u_1(k-2) & \\
+ b_{21_2} u_2(k-1) + b_{22_2} u_2(k-2) & \\
+ c_{10_2} y_1(k) + c_{11_2} y_1(k-1) + c_{12_2} y_1(k-2) & \qquad (5.7)
\end{aligned}$$

5.4.1 Forward Inverse Conversion

Channel no. 1

Incrementing k by 2, and solving for $u_1(k)$ from (5.6), results in the forward inverse model for input channel 1:

$$\begin{aligned}
u_1(k) + \frac{b_{13_1}}{b_{12_1}} u_1(k-1) = \frac{1}{b_{12_1}} [& y_1(k+2) + a_{1_1} y_1(k+1) + a_{2_1} y_1(k) + a_{3_1} y_1(k-1) \\
& - c_{10_1} y_2(k+2) - c_{11_1} y_2(k+1) - c_{12_1} y_2(k) - c_{13_1} y_2(k-1) \\
& - b_{22_1} u_2(k) - b_{23_1} u_2(k-1) \quad]
\end{aligned}$$

By shifting both output channels by -2, which shall be denoted by $\bar{\mathbf{y}}$, the standard ARX difference equation model is achieved:

$$\begin{aligned}
u_1(k) + \frac{b_{13_1}}{b_{12_1}} u_1(k-1) = \frac{1}{b_{12_1}} [\bar{y}_1(k) & + a_{1_1} \bar{y}_1(k-1) + a_{2_1} \bar{y}_1(k-2) + a_{3_1} \bar{y}_1(k-3) \\
- c_{10_1} \bar{y}_2(k) & - c_{11_1} \bar{y}_2(k-1) - c_{12_1} \bar{y}_2(k-2) - c_{13_1} \bar{y}_2(k-3) \\
- b_{22_1} u_2(k) & - b_{23_1} u_2(k-1)]
\end{aligned}$$

The resulting final parameters for input channel u_1 are therefore given by

$$\mathcal{A}_{1_{fi}} = \left[\begin{array}{cc} 1 & \frac{b_{13_1}}{b_{12_1}} \end{array}\right] u_1$$

$$\mathcal{B}_{1_{fi}} = \frac{1}{b_{12_1}} \left[\begin{array}{cccc} 1 & a_{1_1} & a_{2_1} & a_{3_1} \\ -c_{10_1} & -c_{11_1} & -c_{12_1} & -c_{13_1} \\ -b_{22_1} & -b_{23_1} & & \end{array}\right] \begin{array}{c} \bar{y}_1 \\ \bar{y}_2 \\ u_2 \end{array}$$

Three points of interest need to be mentioned. Firstly $\mathcal{A}_1$ for the normal model was a third order Autoregressive model whereas $\mathcal{A}_{1_{fi}}$ has now, after inversion, become a first order Autoregressive model. This is clearly due to the two initial delays in both inputs. The second point of interest is that both outputs y_1 and y_2 have now been delayed by two sampling periods to $\bar{y}_1$ and $\bar{y}_2$, also because of the two initial delays. Finally it is noticed that $\mathcal{B}_{1_{fi}}$ has no delays on its inputs, which in this case are $\bar{y}_1$ and $\bar{y}_2$.

Channel no. 2

Attempting a similar analysis for the second channel leads to incrementing k by 1, and solving for $u_2(k)$ from (5.7), which gives the forward inverse model for input channel 2 as:

$$\begin{aligned} u_2(k) + \frac{b_{22_2}}{b_{21_2}} u_2(k-1) = \frac{1}{b_{21_2}} [& -c_{10_2} y_1(k+1) - c_{11_2} y_1(k) - c_{12_2} y_1(k-1) \\ & + y_2(k+1) + a_{1_2} y_2(k) + a_{2_2} y_2(k-1) \\ & - b_{11_2} u_1(k) - b_{12_2} u_1(k-1) \quad] \end{aligned}$$

In this case, it will be necessary to shift both output channels by -1, which is also denoted by $\bar{y}$, to achieve the standard ARX difference model:

$$\begin{aligned} u_2(k) + \frac{b_{22_2}}{b_{21_2}} u_2(k-1) = \frac{1}{b_{21_2}} [-c_{10_2} y_1(k) & - c_{11_2} y_1(k-1) - c_{12_2} y_1(k-2) \\ + y_2(k) & + a_{1_2} y_2(k-1) + a_{2_2} y_2(k-2) \\ - b_{11_2} u_1(k) & - b_{12_2} u_1(k-1)] \end{aligned}$$

The resulting final parameters for input channel u_2 are therefore given by

$$\mathcal{A}_{2_{fi}} = \begin{array}{c} \left[\begin{array}{cc} 1 & \frac{b_{22_2}}{b_{21_2}} \end{array} \right] \end{array} u_2$$

$$\mathcal{B}_{2_{fi}} = \frac{1}{b_{21_2}} \left[\begin{array}{ccc} -c_{10_2} & -c_{11_2} & -c_{12_2} \\ 1 & a_{1_2} & a_{2_2} \\ -b_{11_2} & -b_{12_2} & \end{array} \right] \begin{array}{c} \bar{y}_1 \\ \bar{y}_2 \\ u_1 \end{array}$$

In the inversion of channel 1 we defined $\bar{\mathbf{y}}(r) \equiv \mathbf{y}(k-2)$ whereas in the inversion of channel 2 we defined $\bar{\mathbf{y}}(r) \equiv \mathbf{y}(k-1)$. On the surface this seems acceptable, however when converting the two inverse models to the state space formulation this poses a problem, since the two models would not be compatible in a final state space model. We are therefore compelled to restrict the model structures in such a manner that in the normal models, the number of delays on the inputs are required to be equal.

5.4.2 Reversed Inverse Conversion

The reversed inverse conversion overcomes the delay restrictions required by the forward inverse conversion. In this case no data shifting is required.

Channel no. 1

Solving for $u_1(k-3)$ from (5.6) results in the reversed inverse model for input channel 1 being

$$\begin{aligned} u_1(k-3) + \frac{b_{12_1}}{b_{13_1}} u_1(k-2) = \frac{1}{b_{13_1}} [& a_{3_1} y_1(k-3) \; + \; a_{2_1} y_1(k-2) + a_{1_1} y_1(k-1) + y_1(k) \\ & -c_{13_1} y_2(k-3) \; - \; c_{12_1} y_2(k-2) - c_{11_1} y_2(k-1) - c_{10_1} y_2(k) \\ & -b_{23_1} u_2(k-3) \; - \; b_{22_1} u_2(k-2)] \end{aligned}$$

By reversing the order of the data which is denoted by $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{y}}$, and introducing the new index r as defined in (5.1), this results in

$$\begin{aligned}\tilde{u}_1(r) + \frac{b_{12_1}}{b_{13_1}}\tilde{u}_1(r-1) = \frac{1}{b_{13_1}}[a_{3_1}\tilde{y}_1(r) \quad &+ \quad a_{2_1}\tilde{y}_1(r-1) + a_{1_1}\tilde{y}_1(r-2) + \tilde{y}_1(r-3) \\ -c_{13_1}\tilde{y}_2(r) \quad &- \quad c_{12_1}\tilde{y}_2(r-1) - c_{11_1}\tilde{y}_2(r-2) - c_{10_1}\tilde{y}_2(r-3) \\ -b_{23_1}\tilde{u}_2(r) \quad &- \quad b_{22_1}\tilde{u}_2(r-1)]\end{aligned}$$

The resulting final parameters for input channel $\tilde{u}_1$ are therefore given by

$$\mathcal{A}_{1_{ri}} = \left[\begin{array}{cc} 1 & \frac{b_{12_1}}{b_{13_1}} \end{array}\right]\tilde{u}_1$$

$$\mathcal{B}_{1_{ri}} = \frac{1}{b_{13_1}}\left[\begin{array}{cccc} a_{3_1} & a_{2_1} & a_{1_1} & 1 \\ -c_{13_1} & -c_{12_1} & -c_{11_1} & -c_{10_1} \\ -b_{23_1} & -b_{22_1} & & \end{array}\right]\begin{array}{c}\tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{u}_2\end{array}$$

Channel no. 2

Solving for $u_2(k-2)$ from (5.7) results in the reversed inverse model for input channel 2:

$$\begin{aligned}u_2(k-2) + \frac{b_{21_2}}{b_{22_2}}u_2(k-1) = \frac{1}{b_{22_2}}[\quad &- \quad c_{12_2}y_1(k-2) - c_{11_2}y_1(k-1) - c_{10_2}y_1(k) \\ &+ \quad a_{2_2}y_2(k-2) + a_{1_2}y_2(k-1) + y_2(k) \\ &- \quad b_{12_2}u_1(k-2) - b_{11_2}u_1(k-1)].\end{aligned}$$

After reversing the order of the data, and introducing the new index r as defined in (5.1), the above equation becomes

$$\begin{aligned}\tilde{u}_2(r) + \frac{b_{21_2}}{b_{22_2}}\tilde{u}_2(r-1) = \frac{1}{b_{22_2}}[-c_{12_2}\tilde{y}_1(r) \quad &- \quad c_{11_2}\tilde{y}_1(r-1) - c_{10_2}\tilde{y}_1(r-2) \\ +a_{2_2}\tilde{y}_2(r) \quad &+ \quad a_{1_2}\tilde{y}_2(r-1) + \tilde{y}_2(r-2) \\ -b_{12_2}\tilde{u}_1(r) \quad &- \quad b_{11_2}\tilde{u}_1(r-1)].\end{aligned}$$

The resulting final parameters for input channel $\tilde{u}_2$ are therefore given by

$$\mathcal{A}_{2_{ri}} = \left[\begin{array}{cc} 1 & \frac{b_{21_2}}{b_{22_2}} \end{array}\right] \tilde{u}_2$$

$$\mathcal{B}_{2_{ri}} = \frac{1}{b_{22_2}} \begin{array}{c} \left[\begin{array}{ccc} -c_{12_2} & -c_{11_2} & -c_{10_2} \\ a_{2_2} & a_{1_2} & 1 \\ -b_{12_2} & -b_{11_2} & \end{array}\right] \end{array} \begin{array}{c} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{u}_1 \end{array}$$

In the inversion of the first channel we defined

$$\begin{aligned}
u(1) &= \tilde{u}(N) \\
u(2) &= \tilde{u}(N-1) \\
&\vdots \\
u(k-3) &= \tilde{u}(r) \\
u(k-2) &= \tilde{u}(r-1) \\
u(k-1) &= \tilde{u}(r-2) \\
u(k) &= \tilde{u}(r-3) \\
&\vdots \\
u(N-2) &= \tilde{u}(3) \\
u(N-1) &= \tilde{u}(2) \\
u(N) &= \tilde{u}(1)
\end{aligned}$$

while in the inversion of the second channel the following definition was utilized:

$$\begin{aligned}
u(1) &= \tilde{u}(N) \\
u(2) &= \tilde{u}(N-1) \\
&\vdots \\
u(k-2) &= \tilde{u}(r) \\
u(k-1) &= \tilde{u}(r-1) \\
u(k) &= \tilde{u}(r-2) \\
&\vdots
\end{aligned}$$

$$\begin{aligned} u(N-2) &= \tilde{u}(3) \\ u(N-1) &= \tilde{u}(2) \\ u(N) &= \tilde{u}(1) \end{aligned}$$

These seemingly differing definitions however pose no problems since the two inversions are independent of each other. The conversion of the above two reversed inverse models to the state space formulation will also not be influenced by these two differing definitions.

Inverting an ARX model to the reversed inverse formulation therefore requires no restrictions on the input delays, and is in general the preferred formulation over the forward inverse case.

Delays in the Reversed Inverse Model

Comparing the inverse model parameter matrices $\mathcal{B}_{1_{ri}}$ and $\mathcal{B}_{2_{ri}}$ for the reversed inverse model with those obtained for the forward inverse model $\mathcal{B}_{1_{fi}}$ and $\mathcal{B}_{2_{fi}}$, the reversed inverse model always has zero delay terms. In the direct identification of reversed inverse ARX models, which will be presented in section 5.7.2, the estimation of delays becomes unneccessary since the number of delays will always be zero. Advantage is taken of this fact, which is further pursued in section 6.2.2 .

5.5 Inversion of Multivariable State Space Models: Forward Inverse

In this section the inversion of a normal multivariable state space model to a forward inverse model is presented. This theory was taken from Strejc [1981], where it is simply referred to as an "inverse", and is presented here for the sake of completeness.

We start with a definition, Brockett [1965], relating to the transport delay of the system:

Definition: Let the system be described by the equations

$$\mathbf{x}(k+1) = \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) \tag{5.8}$$

$$\mathbf{y}(k) = \mathbf{Cx}(k) + \mathbf{Du}(k) \tag{5.9}$$

Denoting

$$\begin{aligned} h_0 &= \mathbf{D} \\ h_1 &= \mathbf{C\Phi}^0\mathbf{\Gamma} \\ h_2 &= \mathbf{C\Phi}^1\mathbf{\Gamma} \\ h_3 &= \mathbf{C\Phi}^2\mathbf{\Gamma} \\ &\vdots \\ h_i &= \mathbf{C\Phi}^{i-1}\mathbf{\Gamma} \end{aligned} \tag{5.10}$$

then m defined by the relation

$$m = min_{i=0,1,2....}\{i : h_i \neq 0\} \tag{5.11}$$

is the so-called relative order of the system .
Three points are of interest:

- Comparing (5.10) with (2.3) and (5.4), the series $h_0, h_1, h_2, \cdots$ represents the ordinates of the discrete impulse response.
- If $h_0 = h_1 = h_2 \cdots = h_{m-1} = 0$, and $h_m \neq 0$, then m is the number of delay periods of the output with respect to the input.
- In the discrete transfer function of the system the relative order is given by the difference in degree of the denominator and numerator polynomials.

The inverted system may now be defined as follows:

Relative Order $m = 0$

Consider first a system with relative order $m = 0$, for which
$\mathbf{D} \neq [0]$

From (5.9) $\mathbf{u}(k)$ is solved:

$$\mathbf{u}(k) = [-\mathbf{D}^{-1}\mathbf{C}]\mathbf{x}(k) + [\mathbf{D}^{-1}]\mathbf{y}(k) \tag{5.12}$$

This gives the output equation, which may be substituted into (5.8), giving:

$$\mathbf{x}(k+1) = [\mathbf{\Phi} - \mathbf{\Gamma D}^{-1}\mathbf{C}]\mathbf{x}(k) + [\mathbf{\Gamma D}^{-1}]\mathbf{y}(k) \tag{5.13}$$

which is the state equation

Relative Order $m = 1$

Next consider a system with relative order $m = 1$. In this case
$\mathbf{D} = [0]; \mathbf{C\Gamma} \neq [0]$

From (5.9) k is incremented:

$$\mathbf{y}(k+1) = \mathbf{Cx}(k+1)$$

Substituting (5.8) into the above equation, we again solve for $\mathbf{u}(k)$ giving the output equation

$$\mathbf{u}(k) = [-(\mathbf{C\Gamma})^{-1}\mathbf{C\Phi}]\mathbf{x}(k) + [(\mathbf{C\Gamma})^{-1}]\mathbf{y}(k+1) \tag{5.14}$$

Substituting the above into (5.8), gives the state equation

$$\mathbf{x}(k+1) = [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Gamma})^{-1}\mathbf{C\Phi}]\mathbf{x}(k) + [\mathbf{\Gamma}(\mathbf{C\Gamma})^{-1}]\mathbf{y}(k+1) \tag{5.15}$$

Relative Order $m = 2$

For a system with relative order $m = 2$,
$\mathbf{D} = \mathbf{C\Gamma} = [0]; \mathbf{C\Phi\Gamma} \neq [0]$

Incrementing k from (5.8) and (5.9) results in

$$\mathbf{x}(k+2) = \mathbf{\Phi x}(k+1) + \mathbf{\Gamma u}(k+1) \tag{5.16}$$
$$\mathbf{y}(k+2) = \mathbf{Cx}(k+2) \tag{5.17}$$

Substituting (5.8) into (5.16), the result of which is substituted into (5.17), the following equation is obtained:

$$\mathbf{y}(k+2) = \mathbf{C\Phi}^2\mathbf{x}(k) + \mathbf{C\Phi\Gamma u}(k),$$

from which we may solve for the output equation

$$\mathbf{u}(k) = [-(\mathbf{C\Phi\Gamma})^{-1}\mathbf{C\Phi}^2]\mathbf{x}(k) + [(\mathbf{C\Phi\Gamma})^{-1}]\mathbf{y}(k+2) \tag{5.18}$$

Substituting the above into (5.8) gives the state equation

$$\mathbf{x}(k+1) = [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi\Gamma})^{-1}\mathbf{C\Phi}^2]\mathbf{x}(k) + [\mathbf{\Gamma}(\mathbf{C\Phi\Gamma})^{-1}]\mathbf{y}(k+2) \tag{5.19}$$

Relative Order m

In general,

$$\begin{aligned}
\mathbf{x}(k+1) &= [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi^{m-1}\Gamma})^{-1}\mathbf{C\Phi}^{m}]\mathbf{x}(k) + [\mathbf{\Gamma}(\mathbf{C\Phi^{m-1}\Gamma})^{-1}]\mathbf{y}(k+m) \tag{5.20}\\
\mathbf{u}(k) &= [-(\mathbf{C\Phi^{m-1}\Gamma})^{-1}\mathbf{C\Phi^{m}}]\mathbf{x}(k) + [(\mathbf{C\Phi^{m-1}\Gamma})^{-1}]\mathbf{y}(k+m) \tag{5.21}
\end{aligned}$$

Employing the definition of relative order from (5.10),

$$h_m = \mathbf{C\Phi}^{m-1}\mathbf{\Gamma}$$

the inverted model's state and output equations become

$$\begin{aligned}
\mathbf{x}(k+1) &= [\mathbf{\Phi} - \mathbf{\Gamma}(h_m)^{-1}\mathbf{C\Phi}^{m}]\mathbf{x}(k) + [\mathbf{\Gamma}(h_m)^{-1}]\mathbf{y}(k+m) \tag{5.22}\\
\mathbf{u}(k) &= [-(h_m)^{-1}\mathbf{C\Phi^{m}}]\mathbf{x}(k) + (h_m)^{-1}\mathbf{y}(k+m) \tag{5.23}
\end{aligned}$$

Depending on the specific dynamic system the above forward inverse model may however exhibit unstable eigenvalues and one of the other inversion techniques would then be required.

5.6 Inversion of Multivariable State Space Models: Reversed Inverse

Very often the forward inverse conversion is unsuitable due to an unstable inverse. As described in section 5.2, it is possible to formulate an alternative inverse description of the dynamic system by reversing the order of the data. The resultant model was termed a *reversed inverse model*, which was derived for multivariable ARX models in section 5.4.2 . In this section the inversion to a reversed inverse model will be developed in two alternative methods for multivariable state space systems.

5.6.1 Relative Order Dependent Reversed Inverse Conversion

We first consider the conversion of a normal multivariable state space model to the reversed inverse model where the relative order of the system, which was defined in section 5.5, must be taken into account.

Let the system be described by the equations:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) && (5.24) \\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) && (5.25) \end{aligned}$$

By reversing the order of the data we define a new index r as follows

$$\begin{aligned} r &= k \\ r-1 &= k+1 \\ r-2 &= k+2 \\ &\vdots && (5.26) \end{aligned}$$

In terms of our new indices we obtain

$$\begin{aligned} \mathbf{x}(r-1) &= \mathbf{\Phi x}(r) + \mathbf{\Gamma u}(r) && (5.27) \\ \mathbf{y}(r) &= \mathbf{Cx}(r) + \mathbf{Du}(r) && (5.28) \end{aligned}$$

Relative Order $m = 0$

Consider first a system with relative order $m = 0$, for which
$\mathbf{D} \neq [0]$

From the output equation (5.28) $\mathbf{u}(r)$ is solved directly:

$$\mathbf{u}(r) = -\mathbf{D}^{-1}\mathbf{Cx}(r) + \mathbf{D}^{-1}\mathbf{y}(r) \qquad (5.29)$$

which gives the output equation for the inverse model. We increment both the above equation and the state equation (5.27) and obtain

$$\begin{aligned} \mathbf{u}(r+1) &= -\mathbf{D}^{-1}\mathbf{Cx}(r+1) + \mathbf{D}^{-1}\mathbf{y}(r+1) && (5.30) \\ \mathbf{x}(r+1) &= \mathbf{\Phi}^{-1}\mathbf{x}(r) - \mathbf{\Phi}^{-1}\mathbf{\Gamma u}(r+1) && (5.31) \end{aligned}$$

By substituting (5.30) into (5.31), and solving for $\mathbf{x}(r+1)$, gives the inverted system's state equation

$$\mathbf{x}(r+1) = [(\mathbf{I} - \boldsymbol{\Phi}^{-1}\boldsymbol{\Gamma}\mathbf{D}^{-1}\mathbf{C})^{-1}\boldsymbol{\Phi}^{-1}]\mathbf{x}(r) - [(\mathbf{I} - \boldsymbol{\Phi}^{-1}\boldsymbol{\Gamma}\mathbf{D}^{-1}\mathbf{C})^{-1}\boldsymbol{\Phi}^{-1}\boldsymbol{\Gamma}\mathbf{D}^{-1}]\mathbf{y}(r+1) \tag{5.32}$$

Note that it will be necessary to shift $\mathbf{y}(r+1)$ to $\mathbf{y}(r)$ to effect the simulation of the state equation.

Relative Order $m = 1$

Next consider a system with relative order $m = 1$. In this case
$\mathbf{D} = [0]; \mathbf{C}\boldsymbol{\Gamma} \neq [0]$

The output equation (5.28) incremented by -1 which is substituted into the state equation (5.27) from which $u(r)$ is solved. This gives the inverted system's output equation:

$$\mathbf{u}(r) = [-(\mathbf{C}\boldsymbol{\Gamma})^{-1}\mathbf{C}\boldsymbol{\Phi}]\mathbf{x}(r) + (\mathbf{C}\boldsymbol{\Gamma})^{-1}\mathbf{y}(r-1) \tag{5.33}$$

The state equation (5.27) is incremented by 1 into which the above equation is substituted. Solving for $\mathbf{x}(r+1)$ gives the state equation for the inverse system:

$$\mathbf{x}(r+1) = [\boldsymbol{\Phi} - \boldsymbol{\Gamma}(\mathbf{C}\boldsymbol{\Gamma})^{-1}\mathbf{C}\boldsymbol{\Phi}]^{-1}\mathbf{x}(r) - [\boldsymbol{\Phi} - \boldsymbol{\Gamma}(\mathbf{C}\boldsymbol{\Gamma})^{-1}\mathbf{C}\boldsymbol{\Phi}]^{-1}\boldsymbol{\Gamma}(\mathbf{C}\boldsymbol{\Gamma})^{-1}\mathbf{y}(r) \tag{5.34}$$

In this case it will be necessary to shift $\mathbf{y}(r)$ to $\mathbf{y}(r-1)$ in order to simulate the inverted system's output equation.

Relative Order $m = 2$

For a system with relative order $m = 2$,
$\mathbf{D} = \mathbf{C}\boldsymbol{\Gamma} = [0]; \mathbf{C}\boldsymbol{\Phi}\boldsymbol{\Gamma} \neq [0]$

The state equation (5.27) is incremented by -1 while also substituting (5.27) into this equation

$$\mathbf{x}(r-2) = \boldsymbol{\Phi}^2\mathbf{x}(r) + \boldsymbol{\Phi}\boldsymbol{\Gamma}\mathbf{u}(r) + \boldsymbol{\Gamma}\mathbf{u}(r-1) \tag{5.35}$$

The output equation (5.28) is incremented by -2 into which (5.35) is substituted to obtain

$$\mathbf{y}(r-2) = \mathbf{C\Phi}^2\mathbf{x}(r) + \mathbf{C\Phi\Gamma u}(r) + \mathbf{C\Gamma u}(r-1)$$

Solving for $\mathbf{u}(r)$ from the above leads to the output equation for the inverted system.

$$\mathbf{u}(r) = [-(\mathbf{C\Phi\Gamma})^{-1}\mathbf{C\Phi}^2]\mathbf{x}(r) + [(\mathbf{C\Phi\Gamma})^{-1}]\mathbf{y}(r-2) \qquad (5.36)$$

Substituting (5.36) into the state equation (5.27) and incrementing by 1 leads to the inverted system's state equation

$$\mathbf{x}(r{+}1) = [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi\Gamma})^{-1}\mathbf{C\Phi}^2]^{-1}\mathbf{x}(r) - [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi\Gamma})^{-1}\mathbf{C\Phi}^2]^{-1}\mathbf{\Gamma}(\mathbf{C\Phi\Gamma})^{-1}\mathbf{y}(r{-}1) \qquad (5.37)$$

In this case it will be required to

- shift $\mathbf{y}(r)$ to $\mathbf{y}(r-1)$ for simulating the state equation (5.37)
- shift $\mathbf{y}(r)$ to $\mathbf{y}(r-2)$ for simulating the output equation (5.36) .

Relative Order m

In general we have for the relative order dependent reversed inverse conversion

$$\begin{aligned}\mathbf{x}(r+1) &= [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi}^{\mathbf{m-1}}\mathbf{\Gamma})^{-1}\mathbf{C\Phi}^{m}]^{-1}\mathbf{x}(r) \\ &- [\mathbf{\Phi} - \mathbf{\Gamma}(\mathbf{C\Phi}^{\mathbf{m-1}}\mathbf{\Gamma})^{-1}\mathbf{C\Phi}^{m}]^{-1}\mathbf{\Gamma}(\mathbf{C\Phi}^{\mathbf{m-1}}\mathbf{\Gamma})^{-1}\mathbf{y}(r-m+1)\end{aligned} \qquad (5.38)$$

and

$$\mathbf{u}(r) = [-(\mathbf{C\Phi}^{\mathbf{m-1}}\mathbf{\Gamma})^{-1}\mathbf{C\Phi}^{\mathbf{m}}]\mathbf{x}(r) + [(\mathbf{C\Phi}^{\mathbf{m-1}}\mathbf{\Gamma})^{-1}]\mathbf{y}(r-m) \qquad (5.39)$$

Utilizing the definition of relative order (5.10),

$$h_m = \mathbf{C\Phi}^{m-1}\mathbf{\Gamma}$$

the inverted model's state and output equations for the relative order dependent reversed inverse conversion become

$$\mathbf{x}(r{+}1) = [\mathbf{\Phi} - \mathbf{\Gamma}(h_m)^{-1}\mathbf{C\Phi}^m]^{-1}\mathbf{x}(r) - [\mathbf{\Phi} - \mathbf{\Gamma}(h_m)^{-1}\mathbf{C\Phi}^m]^{-1}\mathbf{\Gamma}(h_m)^{-1}\mathbf{y}(r{-}m{+}1) \qquad (5.40)$$

and

$$\mathbf{u}(r) = [-(h_m)^{-1}\mathbf{C}\mathbf{\Phi}^{\mathbf{m}}]\mathbf{x}(r) + [(h_m)^{-1}]\mathbf{y}(r-m) \tag{5.41}$$

In applying these equations it will be required to

- shift $\mathbf{y}(r)$ to $\mathbf{y}(r-m+1)$ for simulating the state equation (5.40)
- shift $\mathbf{y}(r)$ to $\mathbf{y}(r-m)$ for simulating the output equation (5.41) .

This reversed inverse model may likewise exhibit unstable eigenvalues, and one of the other inversion techniques would then be required.

5.6.2 Relative Order Independent Reversed Inverse Conversion

The inconvenience of taking account of the relative order of the system, together with the necessity of shifting the output vectors in the above formulations, may be avoided by an alternative inversion formulation. By reversing the data, it is found as in equations (5.27) and (5.28) that

$$\mathbf{x}(r-1) = \mathbf{\Phi}\mathbf{x}(r) + \mathbf{\Gamma}\mathbf{u}(r) \tag{5.42}$$
$$\mathbf{y}(r) = \mathbf{C}\mathbf{x}(r) + \mathbf{D}\mathbf{u}(r) \tag{5.43}$$

A new state vector, which is denoted by $\tilde{\mathbf{x}}$ is defined such that

$$\tilde{\mathbf{x}}(r) \equiv \mathbf{x}(r-1) \tag{5.44}$$
$$\tilde{\mathbf{x}}(r+1) \equiv \mathbf{x}(r) \tag{5.45}$$

The state and output equations (5.42) and (5.43) then become

$$\tilde{\mathbf{x}}(r) = \mathbf{\Phi}\tilde{\mathbf{x}}(r+1) + \mathbf{\Gamma}\mathbf{u}(r) \tag{5.46}$$
$$\mathbf{y}(r) = \mathbf{C}\tilde{\mathbf{x}}(r+1) + \mathbf{D}\mathbf{u}(r) \tag{5.47}$$

from which we may solve for $\tilde{\mathbf{x}}(r+1)$, giving

$$\tilde{\mathbf{x}}(r+1) = \mathbf{\Phi}^{-1}\tilde{\mathbf{x}}(r) - \mathbf{\Phi}^{-1}\mathbf{\Gamma}\mathbf{u}(r) \tag{5.48}$$

Substituting the above equation into (5.47) and solving for $\mathbf{u}(r)$, leads to the final output equation of the reversed inverse system:

$$\mathbf{u}(r) = [-(\mathbf{D} - \mathbf{C}\mathbf{\Phi}^{-1}\mathbf{\Gamma})^{-1}\mathbf{C}\mathbf{\Phi}]\tilde{\mathbf{x}}(r) + [\mathbf{D} - \mathbf{C}\mathbf{\Phi}^{-1}\mathbf{\Gamma}]^{-1}\mathbf{y}(r) \tag{5.49}$$

The state equation is found by substituting the above output equation into (5.48)

$$\tilde{\mathbf{x}}(r+1) = \mathbf{\Phi}^{-1}[\mathbf{I} + \mathbf{\Gamma}(\mathbf{D} - \mathbf{C}\mathbf{\Phi}^{-1}\mathbf{\Gamma})^{-1}\mathbf{C}\mathbf{\Phi}^{-1}]\tilde{\mathbf{x}}(r) + [-\mathbf{\Phi}^{-1}\mathbf{\Gamma}(\mathbf{D} - \mathbf{C}\mathbf{\Phi}^{-1}\mathbf{\Gamma})^{-1}]\mathbf{y}(r) \tag{5.50}$$

The convenience of the above formulation lies in the independence of the relative order. With any of following conditions:

$$\begin{aligned} \mathbf{D} &= 0 \\ \mathbf{C}\mathbf{\Gamma} &= 0 \\ \mathbf{C}\mathbf{\Phi}\mathbf{\Gamma} &= 0 \\ \mathbf{C}\mathbf{\Phi}^2\mathbf{\Gamma} &= 0 \\ &\vdots \end{aligned} \tag{5.51}$$

the above state and output equations (5.50) and (5.49) remain valid, which makes this a very convenient reversed inverse formulation. This formulation is however restricted by the existence of $\mathbf{\Phi}^{-1}$

Any one or both of the above reversed inverse formulations may of course exhibit unstable eigenvalues which would then make them unsuitable.

5.7 Direct Identification of Multivariable Inverse ARX Models

Pending the outcome of the stability properties of the final multivariable state space model, it could be found that the model is not invertable, and the effort of identifying the model has been fruitless. By identifying an inverse model directly the stability properties are automatically checked in the process. Since the ultimate goal is an inverse model the direct identification of inverse models is also the simplest approach to follow. This is achieved by a simple reversal of all input and output channels. Again two alternatives are possible through either the forward inverse or reversed inverse models.

5.7.1 Identification of Forward Inverse models

Consider a two channel system which consists of a third order system with two delays for channel 1, and a second order system with one delay for the second channel, as in section 5.4 . The corresponding difference equations are given by (5.6) and (5.7) and are repeated below.

$$\begin{aligned} y_1(k) + a_{1_1} y_1(k-1) + a_{2_1} y_1(k-2) + a_{3_1} y_1(k-3) &= \\ b_{12_1} u_1(k-2) + b_{13_1} u_1(k-3) & \\ +b_{22_1} u_2(k-2) + b_{23_1} u_2(k-3) & \\ +c_{10_1} y_2(k) + c_{11_1} y_2(k-1) + c_{12_1} y_2(k-2) + c_{13_1} y_2(k-3) & \qquad (5.52) \end{aligned}$$

and

$$\begin{aligned} y_2(k) + a_{1_2} y_2(k-1) + a_{2_2} y_2(k-2) &= \\ b_{11_2} u_1(k-1) + b_{12_2} u_1(k-2) & \\ +b_{21_2} u_2(k-1) + b_{22_2} u_2(k-2) & \\ +c_{10_2} y_1(k) + c_{11_2} y_2(k-1) + c_{12_2} y_2(k-2) & \qquad (5.53) \end{aligned}$$

Employing the notation $u_i(k) \equiv u_{i_k}$ and a similar notation for $y_i(k)$, we may then write $N-3$ equations from the model as in section 2.5.2:

$$\begin{bmatrix} y_{1_4} \\ y_{1_5} \\ y_{1_6} \\ \vdots \\ y_{1_N} \end{bmatrix} \approx \begin{bmatrix} y_{1_3} & y_{1_2} & y_{1_1} & u_{1_2} & u_{1_1} & u_{2_2} & u_{2_1} & y_{2_4} & y_{2_3} & y_{2_2} & y_{2_1} \\ y_{1_4} & y_{1_3} & y_{1_2} & u_{1_3} & u_{1_2} & u_{2_3} & u_{2_2} & y_{2_5} & y_{2_4} & y_{2_3} & y_{2_2} \\ y_{1_5} & y_{1_4} & y_{1_3} & u_{1_4} & u_{1_3} & u_{2_4} & u_{2_3} & y_{2_6} & y_{2_5} & y_{2_4} & y_{2_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & & & & \end{bmatrix} \begin{bmatrix} -a_{1_1} \\ -a_{2_1} \\ -a_{3_1} \\ b_{12_1} \\ b_{13_1} \\ b_{22_1} \\ b_{23_1} \\ c_{10_1} \\ c_{11_1} \\ c_{12_1} \\ c_{13_1} \end{bmatrix} \tag{5.54}$$

or

$$\mathbf{y}_N = \phi_N \theta_N + \epsilon_N \tag{5.55}$$

where ϵ_N is the innovations vector. The above may now be converted to a forward inverse system according to the procedure outlined in section 5.4.1 where it was also required to shift both output channels by -2, which was denoted $\bar{\mathbf{y}}$, in order to obtain the standard ARX formulation. For the forward inverse model it hence follows that

$$\begin{bmatrix} u_{1_4} \\ u_{1_5} \\ u_{1_6} \\ \vdots \\ u_{1_N} \end{bmatrix} = \begin{bmatrix} u_{1_3} & \bar{y}_{1_4} & \bar{y}_{1_3} & \bar{y}_{1_2} & \bar{y}_{1_1} & \bar{y}_{2_4} & \bar{y}_{2_3} & \bar{y}_{2_2} & \bar{y}_{2_1} & u_{2_4} & u_{2_3} \\ u_{1_4} & \bar{y}_{1_5} & \bar{y}_{1_4} & \bar{y}_{1_3} & \bar{y}_{1_2} & \bar{y}_{2_5} & \bar{y}_{2_4} & \bar{y}_{2_3} & \bar{y}_{2_2} & u_{2_5} & u_{2_4} \\ u_{1_5} & \bar{y}_{1_6} & \bar{y}_{1_5} & \bar{y}_{1_4} & \bar{y}_{1_3} & \bar{y}_{2_6} & \bar{y}_{2_5} & \bar{y}_{2_4} & \bar{y}_{2_3} & u_{2_6} & u_{2_5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & & & & \end{bmatrix} \frac{1}{b_{12_1}} \begin{bmatrix} -b_{13_1} \\ 1 \\ a_{1_1} \\ a_{2_1} \\ a_{3_1} \\ -c_{10_1} \\ -c_{11_1} \\ -c_{12_1} \\ -c_{13_1} \\ -b_{22_1} \\ -b_{23_1} \end{bmatrix} \tag{5.56}$$

The parameter vector θ_N is seen to coincide with the results obtained in section 5.4.1. A similar formulation may also be written for the second channel. The comments on shifting the data by -2 for channel 1, and by -1 for channel 2 as mentioned in section 5.4.1, will again lead to incompatibility when combining the two models to the state space formulation. The restriction of ensuring equal delays on each channel is therefore also valid in the identification of forward inverse ARX models.

5.7.2 Identification of Reversed Inverse Models

Over and above the restrictions on equal delays on all channels in the identification of forward inverse models it is also necessary to know the true order and structure of the system. This inconvenience can be overcome by the formulation of the reversed inverse data, where both the inputs and outputs, as well as the order of the data is reversed.

By simply reversing the order of the data, and simultaneously also reversing the order of the $N-3$ equations of (5.54), results in a change in order of the last column giving:

$$a_{3_1}\begin{bmatrix} \tilde{y}_{1_4} \\ \tilde{y}_{1_5} \\ \tilde{y}_{1_6} \\ \vdots \\ \tilde{y}_{1_N} \end{bmatrix} = \begin{bmatrix} \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{y}_{1_1} & \tilde{u}_{1_4} & \tilde{u}_{1_3} & \tilde{u}_{2_4} & \tilde{u}_{2_3} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} & \tilde{y}_{2_1} \\ \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{u}_{1_5} & \tilde{u}_{1_4} & \tilde{u}_{2_5} & \tilde{u}_{2_4} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} \\ \tilde{y}_{1_5} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{u}_{1_6} & \tilde{u}_{1_5} & \tilde{u}_{2_6} & \tilde{u}_{2_5} & \tilde{y}_{2_6} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & & & & & \end{bmatrix} \begin{bmatrix} -a_{2_1} \\ -a_{1_1} \\ 1 \\ b_{13_1} \\ b_{12_1} \\ b_{23_1} \\ b_{22_1} \\ c_{13_1} \\ c_{12_1} \\ c_{11_1} \\ c_{10_1} \end{bmatrix} \tag{5.57}$$

Solving for the fourth column results in

$$
-b_{13_1}\begin{bmatrix} \tilde{u}_{1_4} \\ \tilde{u}_{1_5} \\ \tilde{u}_{1_6} \\ \vdots \end{bmatrix} = \begin{bmatrix} \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{y}_{1_1} & \tilde{u}_{1_3} & \tilde{u}_{2_4} & \tilde{u}_{2_3} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} & \tilde{y}_{2_1} \\ \tilde{y}_{1_5} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{u}_{1_4} & \tilde{u}_{2_5} & \tilde{u}_{2_4} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} \\ \tilde{y}_{1_6} & \tilde{y}_{1_5} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{u}_{1_5} & \tilde{u}_{2_6} & \tilde{u}_{2_5} & \tilde{y}_{2_6} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{y}_{1_N} & & & & & & & & & & \end{bmatrix} \begin{bmatrix} -a_{3_1} \\ -a_{2_1} \\ -a_{1_1} \\ -1 \\ -b_{12_1} \\ b_{23_1} \\ b_{22_1} \\ c_{13_1} \\ c_{12_1} \\ c_{11_1} \\ c_{10_1} \end{bmatrix} \tag{5.58}
$$

Rearranging the above matrix equation finally gives

$$
\begin{bmatrix} \tilde{u}_{1_4} \\ \tilde{u}_{1_5} \\ \tilde{u}_{1_6} \\ \vdots \end{bmatrix} = \begin{bmatrix} \tilde{u}_{1_3} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{y}_{1_1} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} & \tilde{y}_{2_1} & \tilde{u}_{2_4} & \tilde{u}_{2_3} \\ \tilde{u}_{1_4} & \tilde{y}_{1_5} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{1_2} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{y}_{2_2} & \tilde{u}_{2_5} & \tilde{u}_{2_4} \\ \tilde{u}_{1_5} & \tilde{y}_{1_6} & \tilde{y}_{1_5} & \tilde{y}_{1_4} & \tilde{y}_{1_3} & \tilde{y}_{2_6} & \tilde{y}_{2_5} & \tilde{y}_{2_4} & \tilde{y}_{2_3} & \tilde{u}_{2_6} & \tilde{u}_{2_5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & \tilde{y}_{1_N} & & & & & & & & & \end{bmatrix} \frac{1}{b_{13_1}} \begin{bmatrix} -b_{12_1} \\ a_{3_1} \\ a_{2_1} \\ a_{1_1} \\ 1 \\ -c_{13_1} \\ -c_{12_1} \\ -c_{11_1} \\ -c_{10_1} \\ -b_{23_1} \\ -b_{22_1} \end{bmatrix} \tag{5.59}
$$

The parameters obtained in this way are directly related to those obtained in section 5.4.2 for the first channel. A similar analysis may be done for the second channel. Furthermore no data shifting is required, which makes the identification of reversed inverse ARX models an attractive approach. It must however be bourne in mind that the above identified models may likewise be unstable.

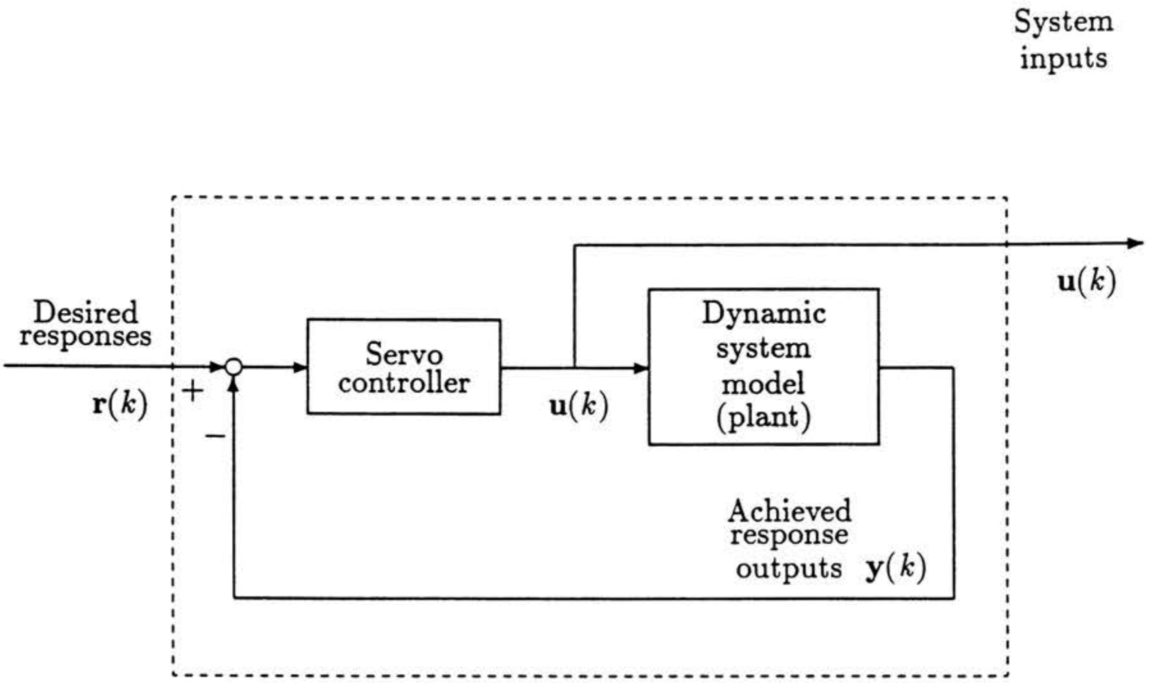


Figure 5.1: Application of a servo-controller to determine the system inputs from desired system responses

5.8 Linear Quadratic Optimal Servo-controller

In some instances it is not possible to implement any of the previous techniques, due to the system exhibiting an unstable inverse. The most appropriate solution then is the synthesis of the model using a linear quadratic optimal servo-controller. A servo-controller is by definition a system which lets the output of the plant follow a pre-specified desired path, using the output feedback. In on-line multivariable systems control this is one of the most successful methods of controlling the plant in such a manner that the desired responses are achieved. In single input single output systems it is of course also possible to implement a controller using pole placement. However multivariable systems will invariably be tested, therefore dictating the use of a linear quadratic optimal servo-controller.

Figure 5.1 shows the application of a typical closed loop servo-controller. In this case however the system inputs are extracted during the simulation of the closed loop system, which is indicated on the right hand side of the figure. These system inputs are then stored and thereafter used to excite the actual test rig. It is important to realize that the inputs, outputs and states in figure 5.1 are vectors, indicating a multiple-input multiple-output system.

5.8.1 Development of the LQ Servo-controller

The development of the closed loop servo-controller is given in Appendix D. Only the basic principles of how the servo-controller is employed in the specific application for determining the system inputs from the desired responses are discussed here, and the reader is referred to the appendix for relevant details.

5.8.2 System Performance

The general performance of any dynamically controlled plant may be described in terms of the step response of the closed loop system. Aspects such as rise time, overshoot and damping ratio are common parameters when describing the performance. It must however be bourne in mind that a dynamic plant with poor performance in terms of its step response may be extremely difficult to control. Closed loop control in the true sense of the word essentially relates to stability. There is however a small degree of control over the plant performance in the sense of optimizing the analogue PID controllers of the actuator servo-valves. Apart from this we have no control over the dynamic performance of the test structure. Although the dynamic performance of the test system, or plant, may be poor in terms of its step response, the optimal servo-controler will always improve the performance of the closed loop system.

5.8.3 Iterative Corrections

When testing a system with poor dynamic response, the servo-controller may be unable to give an absolute "perfect" response, and the closed loop system output will follow the desired system responses as well as possible, but with a certain degree of error. This response error, being the difference between the desired response and the closed loop response, is thereafter again input to the closed loop system as a desired response from which the system inputs are determined to correct for this deviation in response. The overall procedure may be described as follows:

- Simulate the closed loop system using the desired system responses as reference inputs $\mathbf{r}(k) = \mathbf{y}_{desired}$, giving the system inputs $\mathbf{u}_0(k)$
- Subtract the closed loop system responses from the desired system responses $\mathbf{e}_{y_0}(k) = \mathbf{y}_{desired} - \mathbf{y}_0$
- Simulate the closed loop system using the response error as reference inputs $\mathbf{e}_{y_0}(k)$ to determine the error in system inputs $\mathbf{e}_{u_0}(k)$
- The error inputs $\mathbf{e}_{u_0}(k) = \triangle\mathbf{u}_0(k)$ are added to the previously determined inputs to give a corrected input $\mathbf{u}_1(k) = \mathbf{u}_0(k) + \triangle\mathbf{u}_0(k)$
- The corrected input $\mathbf{u}_1$ is input to a simulation of the system model to give the system response $\mathbf{y}_1$
- Subtracting the above determined responses from the desired system responses again gives an error in system response $\mathbf{e}_{y_1}(k) = \mathbf{y}_{desired} - \mathbf{y}_1$

This procedure is repeated in a number of iterative corrections, until the difference between the model simulated responses and the desired operational responses is within acceptable limits.

5.8.4 Dynamic System Delays

When simulating the response of the servo system, a certain amount of delay is found between the reference input and the system output. This is naturally to be expected, since we are simulating a closed loop dynamic system. The Linear Quadratic optimal servo-controller is designed for two fundamental purposes, namely to implement in an on-line real time controller, or alternatively to simulate the system response through deterministic synthesis. In both these cases the transport delay through the complete closed loop system is of no significance. However, when using the servo-controller to calculate the system inputs from the desired responses in an iterative fashion, this poses a problem. The response signals from one iteration to the next are required to be in phase when attempting to subtract them from one another to supply a response error. Likewise the system inputs need to be in phase from one iteration to the next in order to apply corrections to the system inputs.

A simple yet effective method of estimating the delays is to shift the servo system response phase by incremental sample intervals from say 1 to 20 with respect to the desired system response. At each phase shift the sum of squared errors between the servo response and desired response is calculated, and the minimum value found. This would give the "correct" required phase shift. The actual phase shift may however not be a multiple of the fixed sample interval – therefore the quotes. This poses no serious difficulties since shifting the data to the nearest multiple of the sample period gives a slightly higher error, which is then corrected for in the next correction step.

5.9 Other System Inverting Methods

Several other techniques for finding the inverse of multivariable dynamic systems have been developed over the past few decades. Most of these methods are also applicable to the continuous state equations.

One of the more well known techniques is Silverman's structure algorithm, Silverman [1969]. Other methods developed by Dorato [1969], Bengtsson [1974], Moylan [1977], Patel [1977] and other researchers have also concentrated on three fundamental aspects, namely:

- The stability of the inverse
- The existence of an inverse, where existence relates to stability
- Minimal order inverses

The fundamental problem with these techniques is the necessity of having to differentiate the response data in order to determine the system inputs. In a deterministic setting there is no problem, and these algorithms perform exceptionally well. However when the responses exhibit a general stochastic nature, and it is required to numerically differentiate the data, practical implementation problems arise. Even with the most sophisticated numerical differentiation algorithms inaccuracies dominate so strongly that the methods break down completely.

Silverman's structure algorithm was implemented, with surprizingly poor results. The minimal sampling rate for reasonably acceptable results, was found to be at least fifteen times the maximum frequency content of the data. This is necessary simply due to the inherent poor performance of numerical differentiation in general.

Due to these difficulties, these techniques were not further pursued.

5.10 Guidelines on Selecting the Appropriate Inversion Technique

The key issue to selecting the inversion method is stability. However when an unstable inverse is found other issues need to be considered for selecting the appropriate technique. Figure 5.2 shows the summarized procedure in selecting the appropriate inversion technique, which gives a complete picture of the entire process. Application of the techniques to a number of numerical experiments has shown that three of the possible seven routes seem to give the most practical methods. These are discussed below.

5.10.1 Direct Inverse Method

The most natural choice is the identification of an inverse model, and in particular the reversed inverse model, where the inputs and outputs are exhanged and the order of the data is reversed. In doing so we immediately get an indication of the final performance of the model in the desired inverted state, without having to find out afterwards that the model is not invertable due to the system exhibiting an unstable inverse. The identified model is thereafter converted to the state space formulation using the procedures outlined in Appendices B or C. Should the direct identification of a reversed inverse model however prove unsuccessful, a normal forward model needs to be identified. It has been found most convenient to convert the normal model to the state space formulation prior to inversion.

5.10.2 Inverting the State Space Model

Having identified a normal ARX model which has been converted to the state space formulation, thereafter requires the inversion of the state space model, which may be done either through a forward or a reversed inverse conversion.

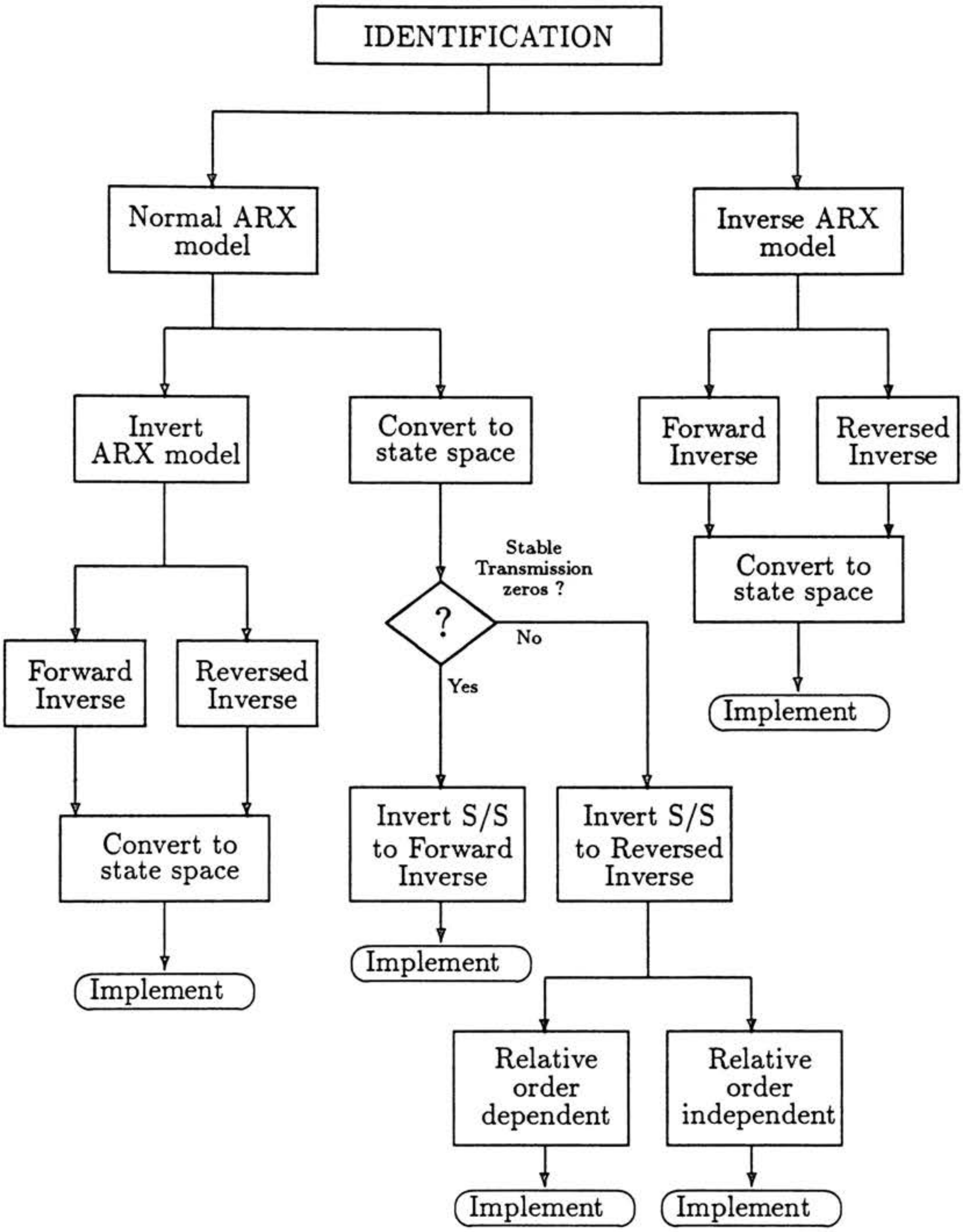


Figure 5.2: Summarized procedure in selecting the appropriate inverting technique

Inverting to a Forward Inverse

In section 5.3.3 it was pointed out that the transmission zeros of the normal forward model become the poles of the forward inverse model. Attempting the inversion to the forward inverse model therefore requires the precondition that the the transmission zeros fall within the unit circle on the complex plane. Should this be the case, equations (5.22) and (5.23) give the state and output equations of the forward inverse model.

Inverting to a Reversed Inverse

If stable transmission zeros are not found, the state space model may be inverted to a reversed inverse. As mentioned in section 5.6, the relative order independent reversed inverse would be the preferred choice. The inversion is given by equations (5.49) and (5.50).

5.10.3 Optimal Servo-controller

Deterministic synthesis using an optimal servo-controller may finally be used when all of the above techniques fail to give satisfactory results. The dynamic performance, in terms of the system step response, determines the success of applying the optimal servo-controller, which is not the case with any of the other inversion algorithms. Because of the off-line application, the servo-controller may be used in an iterative fashion by applying corrections to the system inputs until the desired responses are accurately achieved.

5.11 Summary

This chapter has been the final step in the development of a multiple-input multiple-output inverse description of the laboratory test system. Instability of the inverse model was addressed, and a variety of techniques were established to find an appropriate inverse formulation which is able to render a stable inverse.

The direct identification of inverse ARX models on each input channel, with subsequent conversion to the state space formulation, is considered to be *the* approach. Naturally the initial stochastic modelling, through for instance an ARMAX model, still remains valid.

Chapter 6

System Implementation

6.1 Introduction

All the relevant building blocks have been discussed in the first five chapters. In particular, the basic system description was given in chapter 1, while relevant system identification aspects were established in chapter 2. In chapter 3 the basic SISO model structure was selected, and expanded to MIMO systems through the state space formulation in chapter 4. Finally the development of an inverse multivariable state space description was presented in chapter 5.

The current chapter describes in detail how these building blocks are combined into a complete system which enables the practical application to real engineering problems. This is achieved by a parallel discussion of the developed computer programs which are required to accomplish such a system.

The techniques have been implemented on an 80386 based IBM-compatible computer. Because of the excessive number of matrix and vector operations required, the most suitable environment was found to be MATLAB's 386 package together with several of the tool boxes, see Laub and Little [1986] and Ljung [1988]. MATLAB extensively makes use of the LINPACK and EISPACK algorithms, which are generally known to be some of the most reliable and computationally efficient matrix handling routines.

As far as the hardware is concerned, existing analogue to digital and digital to analogue interfaces on an HP1000 mini computer were used, and files transfered to and from the mini computer and the 386 machine. An

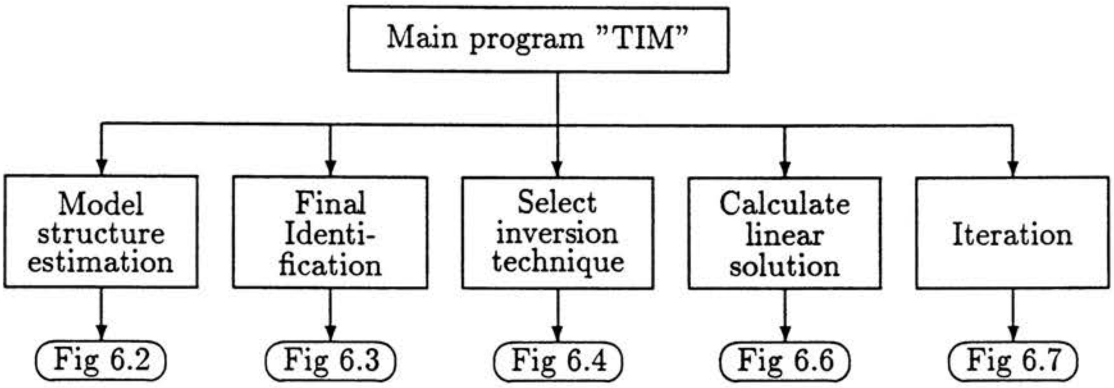


Figure 6.1: Main program structure

ADC/DAC front-end which interfaces directly to a 386 or 486 machine has since been developed to alleviate the necessity of having to transfer data files between the two computers.

The main program has been called TIM. Figure 6.1 shows the main program structure, which consists of five basic steps required to simulate operational measured responses on the laboratory test rig.

The complete procedure may essentially be grouped into three basic functions.

- Identification of the dynamic model
- Inversion of the dynamic model
- Application of the inverse dynamic model

These procedures are discussed in sections 6.2 through 6.4

6.2 Identification of the Dynamic Model

The following basic steps are required to obtain a complete description of the dynamic system:

- Generation of synthetic excitation signals

- Determination of experimental input-output data
- Selection of the model structure for each channel
- Estimation of the model parameters for each channel
- Combination of ARX models to a MIMO state space description

6.2.1 Experimental Input-output Data

Synthetic Excitation Signals

Giving sufficient attention to the selection of excitation signals is imperative in achieving a useful, descriptive dynamic model. Pseudo random white noise, sines, sine sweeps, impulses, pseudo random binary sequences and ARMA processes are but a few of the many typical input signals used for parametric system identification. In the field of modal analysis, chirps, periodic random signals etc. have been used to overcome the periodicity problems of FFT analysis. In time domain parametric modelling, periodicity is of no concern, and a greater degree of freedom is available in the selection of the input excitation signals. On servo-hydraulic test rigs, it is common to use pseudo random white noise generated from a prescribed power spectral density function.

Persistent Excitation

Identifiability is ensured by persistently exciting inputs, see Söderström and Stoica [1989] chapter 5. In principle this means that all modes of the dynamic system should be excited during the identification experiment.

A signal is said to be persistently exciting of order n if the covariance matrix of order n :

$$\mathbf{R}_u(n) = \begin{bmatrix} R_u(0) & R_u(1) & R_u(2) & \cdots & R_u(n-1) \\ R_u(1) & R_u(0) & R_u(1) & \cdots & R_u(n-2) \\ R_u(2) & R_u(1) & R_u(0) & \cdots & R_u(n-3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_u(n-1) & R_u(n-2) & R_u(n-3) & \cdots & R_u(0) \end{bmatrix} \qquad (6.1)$$

is nonsingular. In the frequency domain this condition is equivalent to requiring that the spectral density is nonzero in at least n points. This will ensure that a linear system of order $n/2$ may be consistently identified.

Spectral Properties

The spectral bandwidth of the input signals is an indicator of the bandwidth for which the model is applicable. The frequency content of the operational measured desired responses tell which bandwidth is to be covered by the model, and therefore dictates the bandwith of the excitation signals during the identification process. The frequency domain gives a very efficient indication of the acceptability of the input signals through the coherence function between the inputs and outputs.

6.2.2 Model Structure Estimation

Figure 6.2 shows the flow chart logic for the estimation of the multiple-input single-ouput model structures on each channel. A short section of the experimental input-output data is required for this purpose. Standard system identification procedures of cross validation through simulation of the identified model, as well as residual analysis, are followed in estimating the model structures. It furthermore becomes necessary to also perform a stability analysis on the identified models, since a model exhibiting an unstable inverse is of no use in the particular application.

Model Type

The identification of a direct inverse model is always first attempted. Should this prove to be unsuccessful, a normal model is then identified, upon which it will be required to perform a stability analysis for subsequent inversion of the model.

Order Estimation

One of the most important aspects in dynamic system identification remains the selection of the order of the system. In a multiple-input multiple-output setting this choice becomes even more difficult and the problem of evaluating the different multi-indices is encountered. As mentioned in

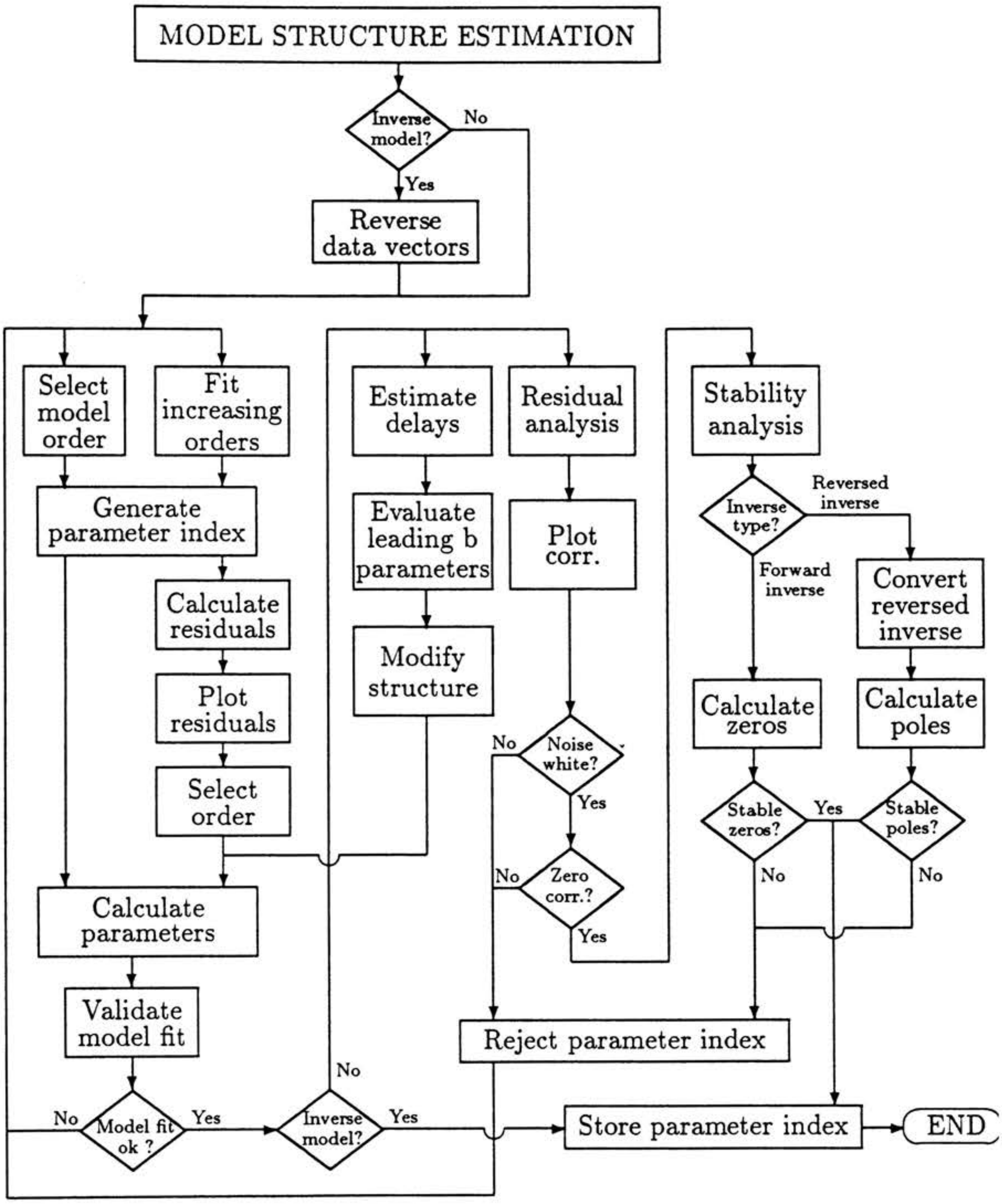


Figure 6.2: Flow chart logic for model structure estimation

section 4.4, this process is in general a difficult one, and the problem is solved easier by breaking down the complex multiple-input multiple-output problem into smaller multiple-input single-output problems, and thereafter combining these separate models into one complete state space system. The problem of order estimation has now been reduced, but still remains.

One of the simplest methods of order estimation is to fit increasing order models and evaluate the prediction errors from each model by plotting. It is however essential to employ a cross-validation dataset to ensure consistent results. The model with the lower prediction error is then selected, bearing in mind the important aspects of parsimony, i.e. not choosing a higher order model if there is not a significant decrease in the prediction errors. Akaike's final prediction error Ljung [1987] is useful in comparing a variety of model orders.

Delay Estimation

The estimation of delays, which is only applicable to normal models, is also a simple exercise in evaluating the leading b terms in the ARX or ARMAX model. The leading b terms are zeroed when they are less than a pre-selected ratio of the maximum b parameter. In doing so, the number of small leading b terms are found. The model is thereafter re-identified, and would generally result in a small change in the nonzero parameters. As shown in section 5.4.2, reversed inverse models always have zero delays, and delay estimation is therefore not applicable to these models. This makes the identification of reversed inverse models an attractive approach.

Order Reduction

It is always necessary to reduce the order of a model to the minimum, retaining only the important input output characteristics of the system. To achieve this objective means eliminating states that have little or no influence on the input output behaviour of the system.

Residual Analysis

Under ideal conditions, the residuals should be white and independent of the MISO model inputs. If this is not the case, then the model structure was insufficient, and did not pick up all the information from the data. The correlation function of the residuals, as well as the cross-correlation between the MISO model inputs and the output, are computed and analysed for whiteness and independence. Makridakis, Wheelwright and McGee [1983] give descriptions of these procedures.

"Full Order" Approach

The estimation of the structure of a multiple-input single-output ARX model involves selection of the following:

- Selection of the number of a parameters (n_a)
- Selection of the number of b parameters (n_b), for each input channel
- Selection of the number of delays (n_k), for each input channel

In a MIMO dynamic system the above decisions need to be made for each output channel. If the number of inputs or outputs are equal to m, then it is required to select $m(4m-1)$ different numbers, giving a total of 3, 14, 33, 60, ... numbers for $m = 1, 2, 3, 4, ...$ that need to be pre-selected, in order to define the structure of the MIMO model. This most certainly seems like an enormous task, and is indeed the case. Furthermore, taking the possible combinations of these numbers into account the task may quite rightly be considered to be virtually impossible in a practical application. Various techniques are available in the literature for determining the order of linear dynamic systems, Woodside [1971], Guidorzi [1975], Tse and Weinert [1975], however, in the intended application, where a high number of channels would apply, these techniques become extremely cumbersome. The incentive for a simpler approach is hence enormous.

By adopting a "full order" approach, the values of 3, 14, 33, 60, ... may be reduced to 1, 2, 3, 4, ... i.e. to the number of channels m.
Definition: A "full order" model is defined such that

$$n_b = n_a + 1 \qquad (6.2)$$

$$n_k = 0 \qquad \forall \; u_i \qquad (6.3)$$

$$n_k = 1 \qquad \forall \; y_i \qquad (6.4)$$

This means that it is only required to select the number of a parameters (n_a) for each output channel, which vastly simplifies the model structure selection process.

The motivation for the "full order" approach may be found by considering the conversion of the ARX models to the observable canonical state space formulation, given in Appendix B, for which a simple two channel application was shown in section 4.5. Essentially this means that all entries in the state and output equations are utilized by the identified ARX parameters or combinations of these parameters.

When applying the ARX to state space conversion of appendix C, equation (6.4) becomes

$$n_k = 0 \qquad \forall \;\; y_i \tag{6.5}$$

The "full order" approach to estimating the model structure, is in particular applicable to the direct identification of reversed inverse models.

Stability Analysis

When identifying a reversed inverse model the stability analysis is automatically done in the model fit validation process. Stability analysis is hence only applicable to normal models, and is done by analysis of the zeros or poles, as detailed in section 5.3.3.

6.2.3 Final Identification of the Model

After the model structures or parameter indices have been defined, it is required to determine the model parameters. Where only a short section of the input-output data was required to find the parameter indices, all the data is now used in the final parameter estimation phase. The procedure of estimating the ARX model parameters is outlined in figure 6.3, which also includes the combination of the ARX models to the final MIMO state space formulation.

Initial Parameter Vector

The ordinary least squares parameter estimation technique for the ARX model, as described in section 2.5.2, is employed as an initial identification phase in order to estimate the initial parameter vector. When the data is contaminated by stochastic disturbances, the IV4 technique is generally selected, which gives an improved estimate of the parameters since the

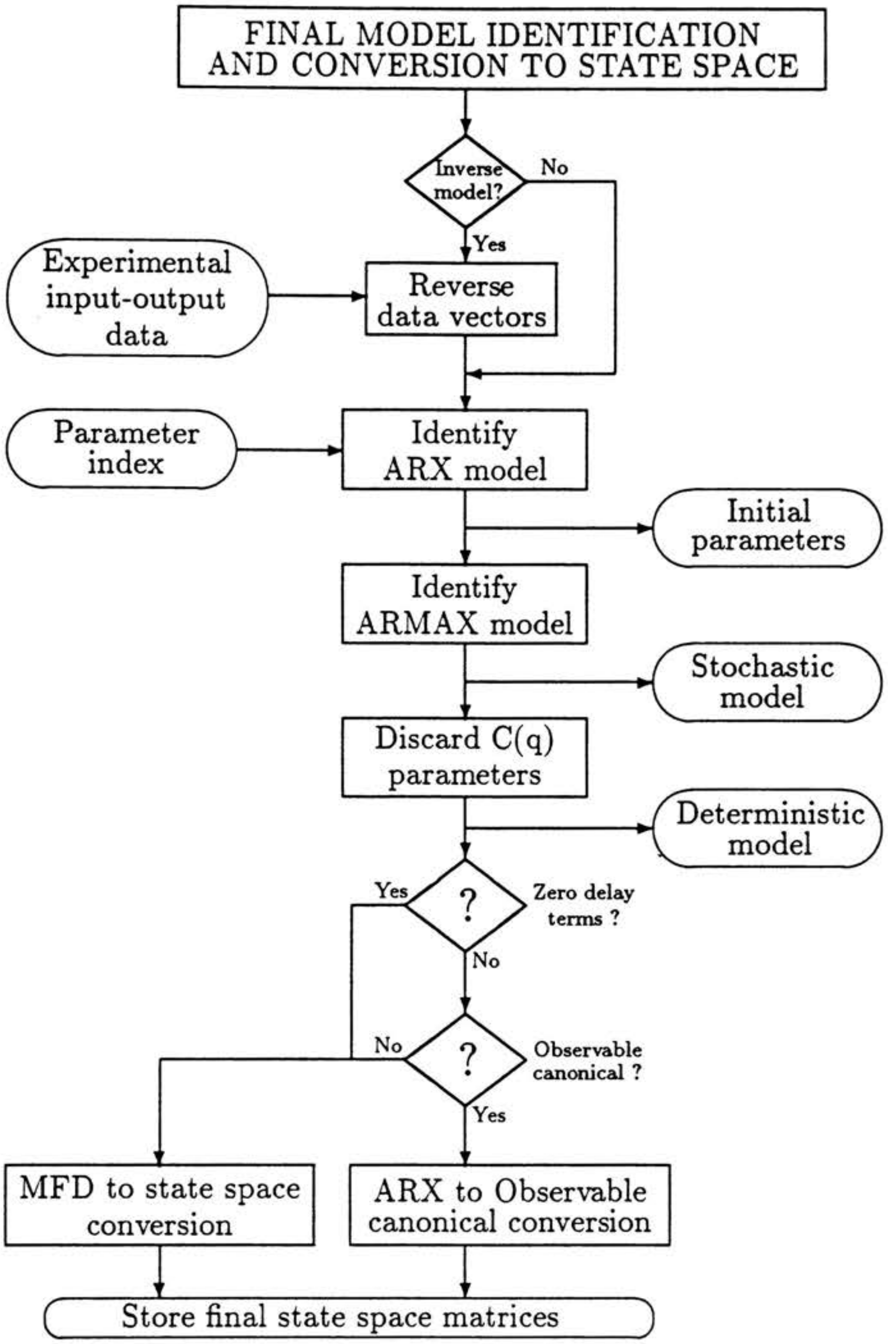


Figure 6.3: Flow chart logic for final model identification and conversion to state space formulation

instrumental variable technique is said to care less about the noise and only estimates the desired system dynamics.

Final Parameter Identification

The above initial parameter vector is subsequently used in the estimation of the stochastic model by means of a Gauss-Newton search scheme in a prediction error method PEM. The $C(q)$ parameters are thereafter discarded to give the final deterministic ARX model.

Combination of ARX Models to the MIMO State Space Description

Depending on the zero delay terms, and the requirement of an observable canonical description, the ARX models for each channel are combined to the state space formulation according to the procedures detailed in Appendices B or C.

6.2.4 Assessing the model fit

In order to assess how well the model describes the experimental data, a "fit" parameter η, which is applied separately to each channel, has been defined. For a forward model, the error in response prediction $e_y(k) = y(k) - \hat{y}(k)$ may be written as

$$e_y(k) = y_{measured}(k) - y_{simulated}(k) \qquad (6.6)$$

Assume that we have N data points. The model fit is then defined as

$$\eta_y = 100 * \frac{\sum_{k=1}^{N} |e_y(k)|}{\sum_{k=1}^{N} |y_{measured}(k)|} \qquad [\%] \qquad (6.7)$$

When a direct inverse model is identified, it is required to define how well the model describes the actual inputs $u(k)$ which were used to excite the structure. In analogy to the above, the error in input prediction $e_u(k) = u(k) - \hat{u}(k)$ may be written as

$$e_u(k) = u_{excitation}(k) - u_{simulated}(k) \qquad (6.8)$$

The model fit for the inverse model may then be defined as

$$\eta_u = 100 * \frac{\sum_{k=1}^{N} |e_u(k)|}{\sum_{k=1}^{N} |u_{excitation}(k)|} \qquad [\%] \qquad (6.9)$$

6.3 Inversion of the Dynamic Model

6.3.1 Stability of the Identified Model

Depending on the model order together with the parameter estimation algorithm used, some models are found to be unstable. Using the least squares technique, a stable ARX model may be found, but when attempting to estimate the same order of model using the four step instrumental variable technique, the resulting model may be found to be unstable. In a number of experiments the IV4 technique was found in general to give unstable models more readily than the least squares estimation technique.

The stability of the model is also a function of the order of the model. Unnecessarily high order models were found to give more stability problems than do lower order models, although both models may seem to give virtually the same small prediction errors. The same can be said about the stability of the inverse model, where it is preferable to use a minimal order model to prevent stability problems.

6.3.2 Selection of the Inversion Technique

The inversion of a normal model has already been described in detail in chapter 5. Section 5.10 gives an overview of the suggested procedures in selecting the correct inversion technique, which is shown in figure 6.4. Testing for stability of the forward inverse model is done by plotting the transmission zeros of the normal model which become the poles of the inverted system. In the case of the reversed inverse model the inversion is first completed and thereafter the poles of the inverse model evaluated.

The direct identification of a reversed inverse model through reversal of the input-output data vectors is considered to be a prime choice by virtue of its simplicity. Transport delay through the estimation of the system delays is also not necessary in this case, as is the requirement of equal delays on each channel, which is required for the forward inverse formulation.

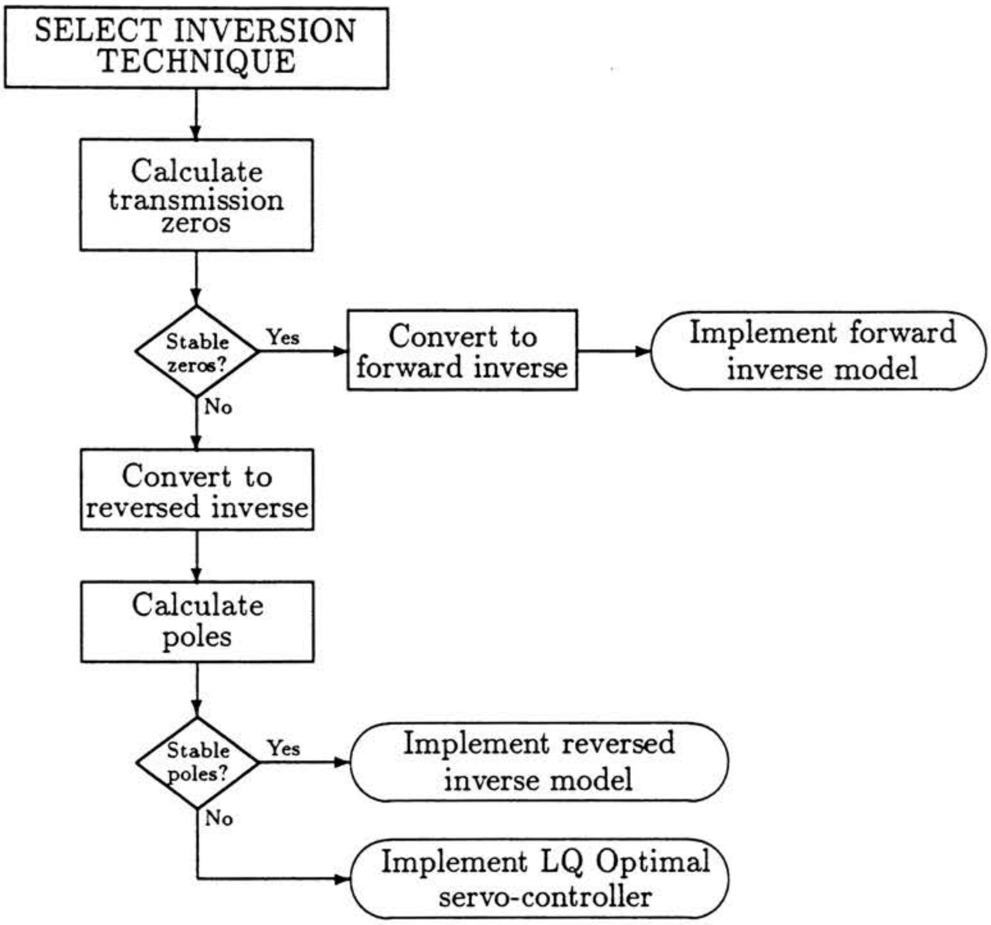

SELECT INVERSION TECHNIQUE
Calculate transmission zeros
Stable zeros?
Yes
Convert to forward inverse
Implement forward inverse model
No
Convert to reversed inverse
Calculate poles
Stable poles?
Yes
Implement reversed inverse model
No
Implement LQ Optimal servo-controller

Figure 6.4: Flow chart logic for selecting the inversion technique

6.3.3 Linear Quadratic Optimal Servo-controller

When all of the previous techniques fail to yield a stable inverse model the linear quadratic optimal servo-controller is employed in determining the system inputs from the remotely recorded operational responses. The normal model is however not inverted in this case. This procedure, which was detailed in section 5.8 and Appendix D, is schematically shown in figure 6.5

6.4 Application of the Model

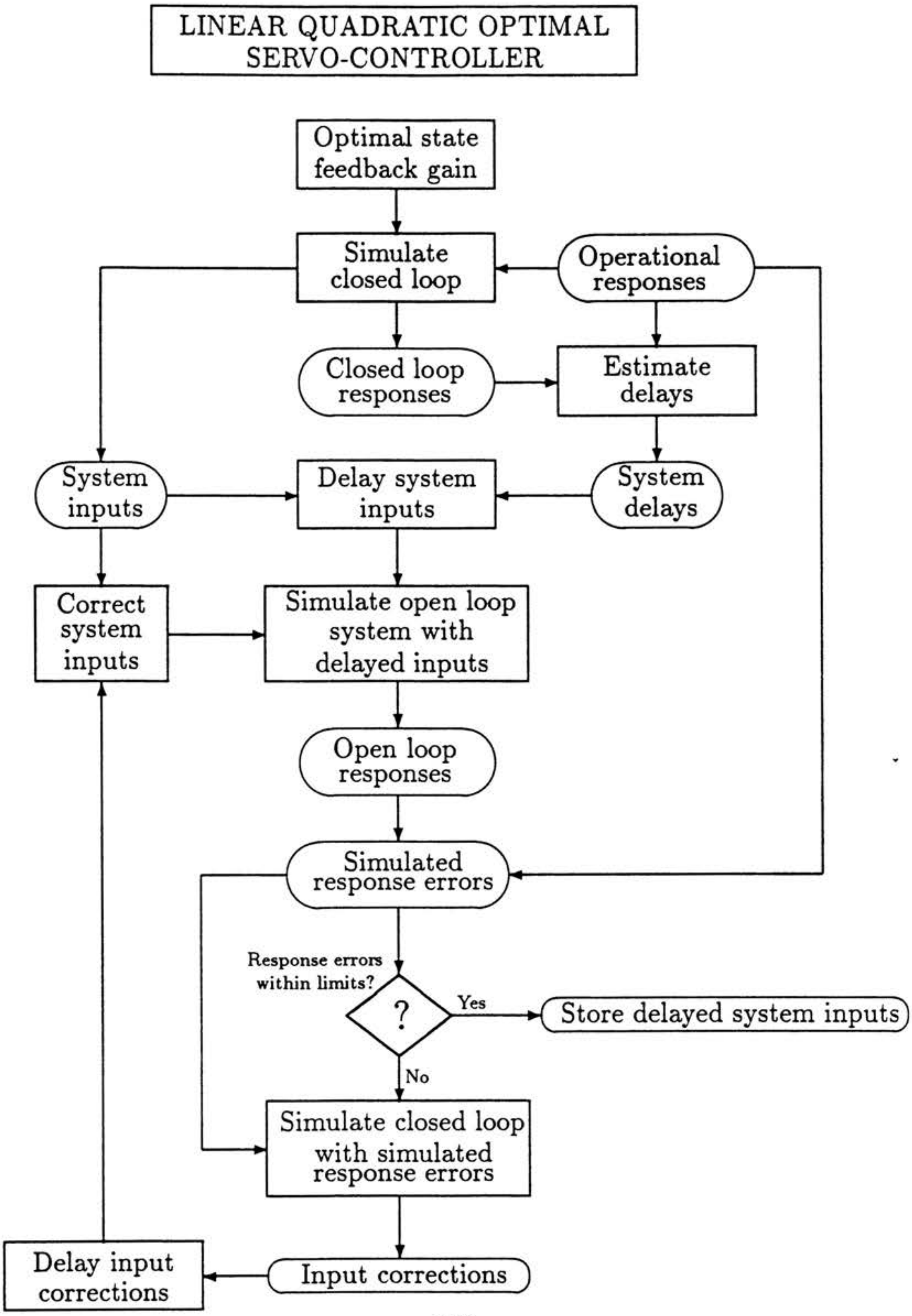
The procedures for obtaining an inverse state space model, giving a complete description of the MIMO dynamic system, were detailed in sections 6.2 and 6.3. It now remains to be shown how this model is utilized in obtaining the system inputs from the remotely measured responses.

6.4.1 Calculation of the Linear Solution Inputs

Details of this procedure are shown in figure 6.6. The initial conditions of the test rig are recorded during the identification process, and are applied in pre-processing the operational responses. The system inputs are found by a simulation of the inverse state space model, using the operational measured responses. The inputs calculated in this manner are termed the *linear solution*, and represent an initial attempt at the system inputs. In the case of the optimal servo-controller, a number of iterative corrections may be required for systems which exhibit a poor step response.

6.4.2 Iteration Process

The laboratory test rig is subsequently excited with the above determined linear solution inputs, while simultaneously recording the system responses, with the same transducers that were utilized during the recording of the operational responses. The accuracy of the laboratory simulation is thereafter ascertained by a comparison of the laboratory responses with the operational measured responses. The difference between the desired operational and measured laboratory responses, leads to the error in response. This response error is then simulated with the inverse model to give the resultant



LINEAR QUADRATIC OPTIMAL SERVO-CONTROLLER
Optimal state feedback gain
Simulate closed loop
Operational responses
Closed loop responses
Estimate delays
System inputs
Delay system inputs
System delays
Correct system inputs
Simulate open loop system with delayed inputs
Open loop responses
Simulated response errors
Response errors within limits?
?
Yes
Store delayed system inputs
No
Simulate closed loop with simulated response errors
Delay input corrections
Input corrections

Figure 6.5: Flow chart logic for determining system inputs from operational responses using LQ Optimal servo-controller

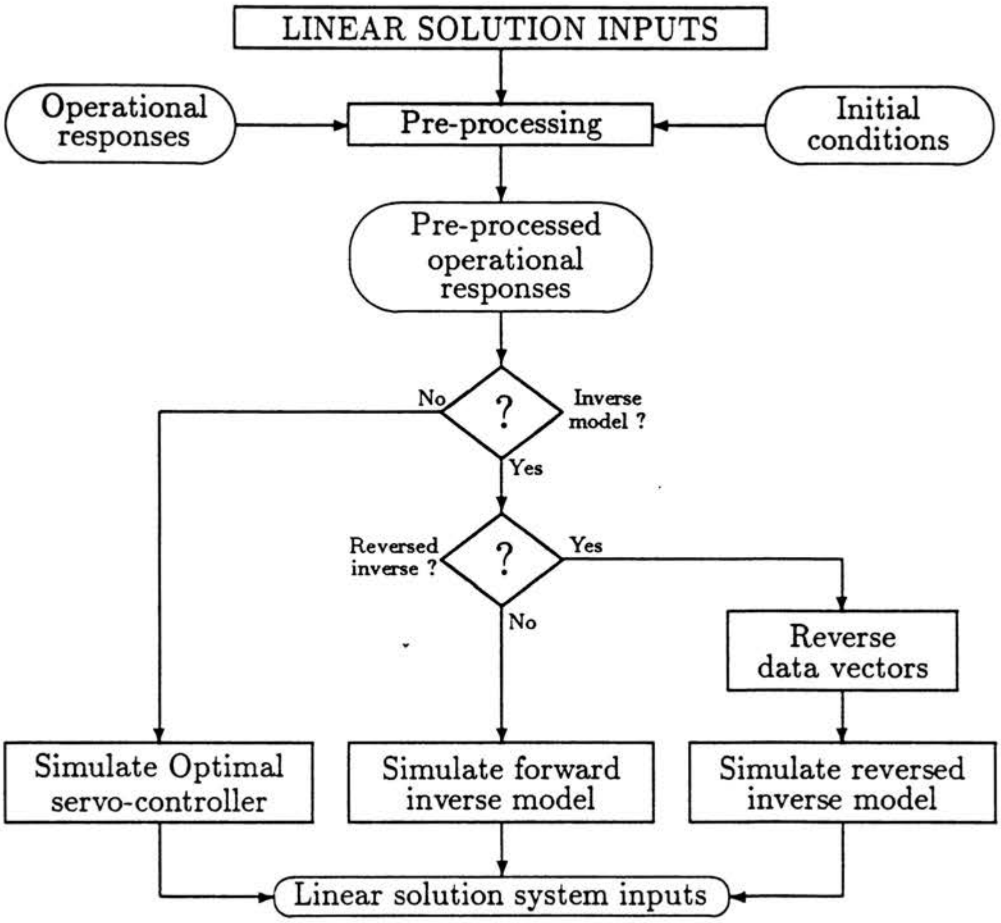

LINEAR SOLUTION INPUTS
Operational responses
Pre-processing
Initial conditions
Pre-processed operational responses
No
?
Inverse model ?
Yes
Reversed inverse ?
?
Yes
No
Reverse data vectors
Simulate Optimal servo-controller
Simulate forward inverse model
Simulate reversed inverse model
Linear solution system inputs

Figure 6.6: Flow chart logic for calculation of system inputs from remotely recorded operational responses

error in inputs, which are added to the linear solution inputs. In this way corrections are applied to achieve updated input signals. By repeating this process in a number of iterations, provision for nonlinear systems is quite easily made. The iteration procedure is detailed in figure 6.7 .

6.4.3 Assessing the simulation accuracy

In order to quantify the simulation accuray which is indicated by the improvement in response accuracy from one iteration to the next, a "fit" parameter has been defined. In analogy to the model fit parameters η_y and η_u defined in equations 6.7 and 6.9, a fit parameter η_{sim} has been defined which describes how well the laboratory achieved response corresponds to the desired operational response. Defining the response error as

$$e_{sim}(k) = y_{desired}(k) - y_{achieved}(k), \qquad (6.10)$$

Assume that we have N data points. The simulation accuracy may then be defined as

$$\eta_{sim} = 100 * \frac{\sum_{k=1}^{N} |e(k)|}{\sum_{k=1}^{N} |y_{desired}(k)|} \qquad [\%] \qquad (6.11)$$

6.4.4 Re-identifying a Model

Should the linearized working point for the operational measured responses differ greatly from the corresponding working point which was found during the identification process, it is then possible to use the input-output data of the linear system inputs to re-identify the model, and thereby find a model which is more suited to the specific desired working point. This procedure is particularly advantageous when testing systems which contain rubber elements with inherent nonlinear stiffness characteristics, as is typical in automotive structures.

6.5 Summary

This chapter concludes the development of the time domain based structural testing system for the simulation of operational responses on servo-hydraulically driven test rigs.

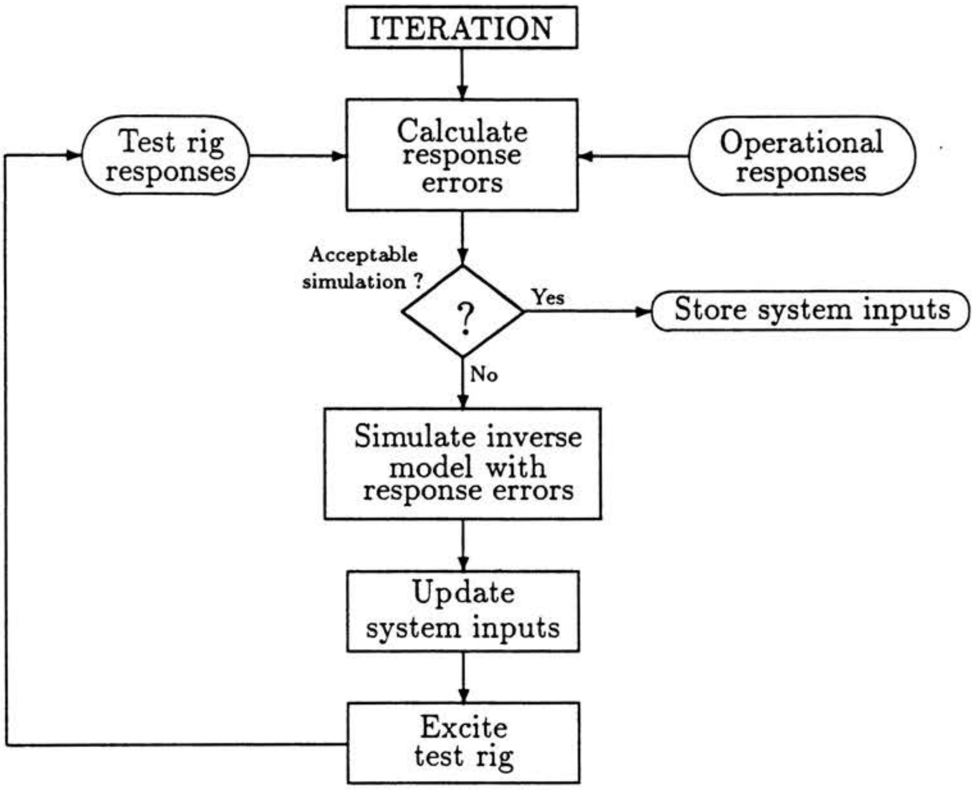


Figure 6.7: Flow chart logic for iteration process

A number of important issues were established in this chapter, which are essential to the practical application of these techniques:

- To alleviate the inverse instability problems, a reversed inverse model is directly identified.
- This choice also has no restrictions on equal delays on each output channel, as was established in section 5.4.2.
- The estimation of the model structures for each channel was shown to be virtually an impossible task and the concept of a "full order" model was introduced to solve this problem.
- The model fit parameters η_y and η_u as well as the simulation accuracy parameter η_{sim} were developed.

The feasibility of the developed system remains to be proven, and the remaining chapters are hence devoted to case studies.

Chapter 7

Applications: Qualification Tests for the Integrity of Computer Programs

7.1 Introduction

The first six chapters dealt with the theoretical development and the implementation of the proposed sytem. The remaining chapters are devoted to the application of the system in a variety of case studies. Three issues are addressed in the next three chapters:

- Proof of the integrity of all computer programs is presented in this chapter.
- Practical case studies are discussed in chapter 8.
- A comparison to the frequency domain techniques is finally done in chapter 9.

The integrity of the computer programs were proven by applying them to a number of test cases. The nature of the developed techniques are such that results cannot be compared to standard text book solutions as is normally done. In this case synthetic single-input single-output as well as multivariable state space models were devised. These models were simulated with pseudo random excitation signals giving the required input and output data from which to identify an appropriate model. Both direct inverse and normal forward models were identified, together with all the different inversion techniques.

The purpose of the tests performed in this chapter is only to qualify the postulated system. The most practical method of achieving this objective is the direct qualification of the actual computer programs. All tests described in this section made use of initial synthetic state space models which served as "test rigs".

Four case studies are presented, which illustrate various aspects of single-input single-output as well as multivariable systems, thereby proving the integrity of the computer programs:

- A simple second order single-input single-output system is firstly studied to illustrate the various different alternatives to both the identification and inversion techniques.

- Secondly the above methods are expanded to include multivariable systems by consideration of a two input two output discrete system. Both forward and reversed inverse identification methods are applied together with the multivariable inversions.

- Next, a two channel spring-mass-damper system is used to show the identification of multivariable black-box state space models. The normal forward inversion algorithm together with the iteration process is also dealt with.

- Finally the optimal servo-controller is applied to a two channel system to prove the integrity of this system inversion technique.

7.2 Single-input Single-output System

The purpose of this case study is essentially to illustrate stability aspects in the inversion of models, as well as to illustrate alternative identification procedures. Depending on the location of the poles and zeros of the model, different techniques are required to obtain a stable inverse.

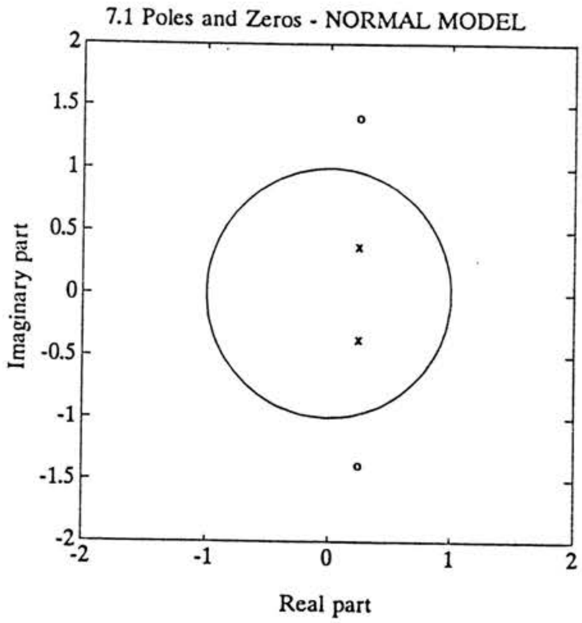


Figure 7.1: Pole and zero locations of the normal model

Normal Model

We start by defining a simple second order model with the following parameters:

$$\mathcal{A} = \begin{bmatrix} 1 & -0.5 & 0.2 \end{bmatrix} \tag{7.1}$$

$$\mathcal{B} = \begin{bmatrix} 0.2 & -0.1 & 0.4 \end{bmatrix} \tag{7.2}$$

The poles of this model are located at $0.2500 \pm 0.3708i$ and the zeros at $0.2500 \pm 1.3919i$, and are shown in figure 7.1 .

Forward Inverse Model

Applying an analysis similar to section 5.4.1 leads to the forward inverse model whose parameters are given by

$$\mathcal{A}_{fi} = \begin{bmatrix} 1 & -0.5 & 2 \end{bmatrix} \tag{7.3}$$

$$\mathcal{B}_{fi} = \begin{bmatrix} 5 & -2.5 & 1 \end{bmatrix} \tag{7.4}$$

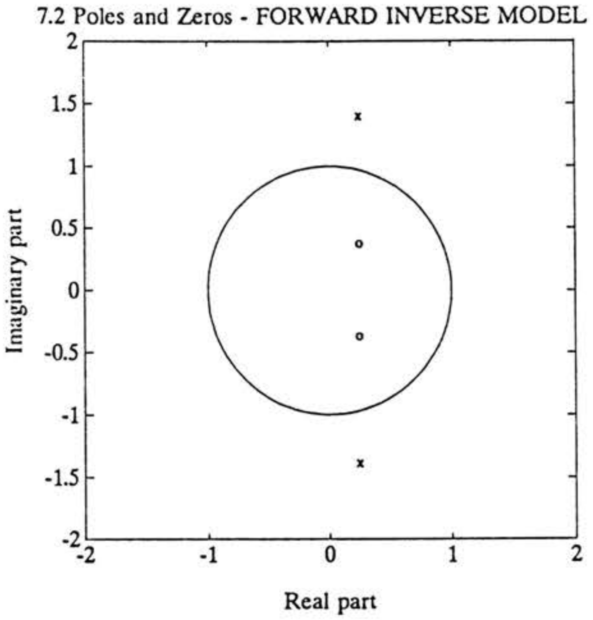


Figure 7.2: Pole and zero positions of the forward inverse model

The poles of the forward inverse model are located at $0.2500 \pm 1.3919i$ and the zeros at $0.2500 \pm 0.3708i$, which shows the expected interchange of the poles and zeros between the normal and forward inverse models.

Figure 7.2 shows the pole positions of the forward inverse model, indicating instability. Identical results are obtained by first converting the model to state space and thereafter applying the inversion algorithm, which was discussed in section 5.5. The forward inverse conversion therefore fails to give a suitable solution, and a reversed inverse is required to achieve stability.

Reversed Inverse Model

The reversed inverse model may be found by applying the inversion procedure described in section 5.4.2, resulting in the following parameters for the reversed inverse model:

$$\mathcal{A}_{ri} = \left[\begin{array}{ccc} 1 & -0.25 & 0.5 \end{array} \right] \tag{7.5}$$

$$\mathcal{B}_{ri} = \left[\begin{array}{ccc} 0.5 & -1.25 & 2.5 \end{array} \right] \tag{7.6}$$

Figure 7.3 shows the pole and zero positions for the reversed inverse model, which are located at $0.1250 \pm 0.6960i$ and $1.2500 \pm 1.8540i$ respectively, and *in this case stability is achieved.*

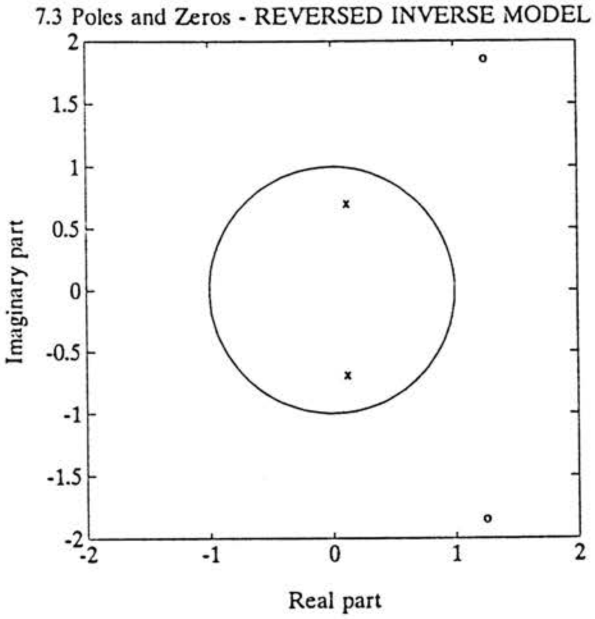


Figure 7.3: Pole and zero positions of the reversed inverse model

By simulating the normal model with any input signal the reversed inverse model may be identified on the input-output data, as described in section 5.7.2 , giving the same parameters for the above reversed inverse model. Alternatively, identical results are obtained by converting the model to state space, and applying the relative order independent reversed inverse conversion given by equations (5.49) and (5.50).

By simulating the reversed inverse model the system input signal is achieved identically.

7.3 Two Channel Discrete State Space System

The above case study may similarly be applied to a system with two inputs and two outputs. In this case a system with a high degree of cross coupling is chosen to illustrate the applicability to the inversion of multivariable systems. The analysis is presented on two models, which only differ in one element of the input matrix $\mathbf{\Gamma}$

Normal Model

We start by defining a system with four states in discrete form. Since we are generating the dynamic model, we have the convenience of also using the observable canonical form to compare with results obtained by both the identification and inversion techniques.

The discrete state equation is given by

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & -0.7 & 0 & -0.20 \\ 1 & 1.3 & 0 & 0.10 \\ 0 & 0.4 & 0 & -0.05 \\ 0 & -0.1 & 1 & 0.30 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0.30 & 0.1 \\ -0.05 & -0.3 \\ 0.10 & 0.8 \\ 0.30 & -0.6 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{7.7}$$

with output equation

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{7.8}$$

The eigenvalues or poles of the above model, shown in figure 7.4, are given by

$$eig(\Phi) = \begin{bmatrix} 0.722 + 0.505i \\ 0.722 - 0.505i \\ 0.078 + 0.377i \\ 0.078 - 0.377i \end{bmatrix} \tag{7.9}$$

7.3.1 Stable Reversed Inverse - Unstable Forward Inverse

Using equations (5.49) and (5.50) , we may convert to the reversed inverse model, while equations (5.22) and (5.23) give the forward inverse model. The eigenvalues of these two inversions are given below, and are also indicated in figure 7.5 .

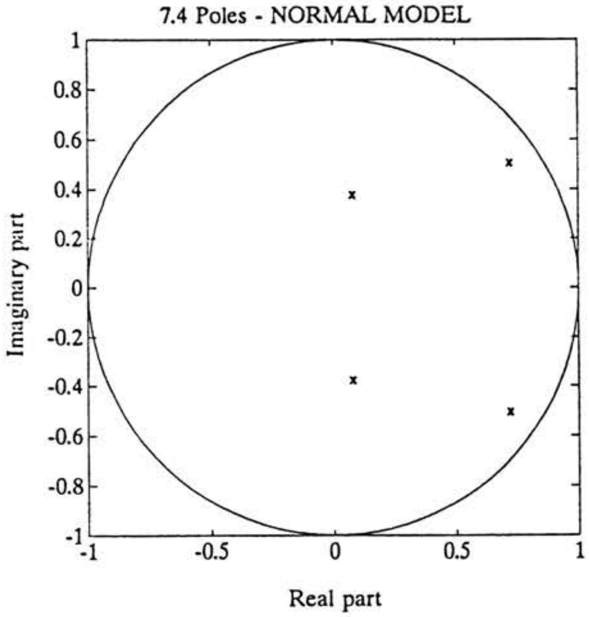


Figure 7.4: Pole positions of the normal model

$$eig(\mathbf{\Phi}_{ri}) = \begin{bmatrix} 0.478 + 0.541i \\ 0.478 - 0.541i \\ 0 \\ 0 \end{bmatrix} \qquad eig(\mathbf{\Phi}_{fi}) = \begin{bmatrix} 0.917 + 1.038i \\ 0.917 - 1.038i \\ 0 \\ 0 \end{bmatrix} \quad (7.10)$$

Reversed Inverse Model

Figure 7.5 (A) shows the pole positions of the reversed inverse model, indicating stability. Having established a stable reversed inverse model, the identification of a direct reversed inverse model now becomes possible, for which identical pole positions are obtained. By simulating the reversed inverse model the system input signals are identically achieved.

Forward Inverse Model

Attempting the forward inversion algorithm however, leads to instability with the eigenvalues shown in figure 7.5 (B)

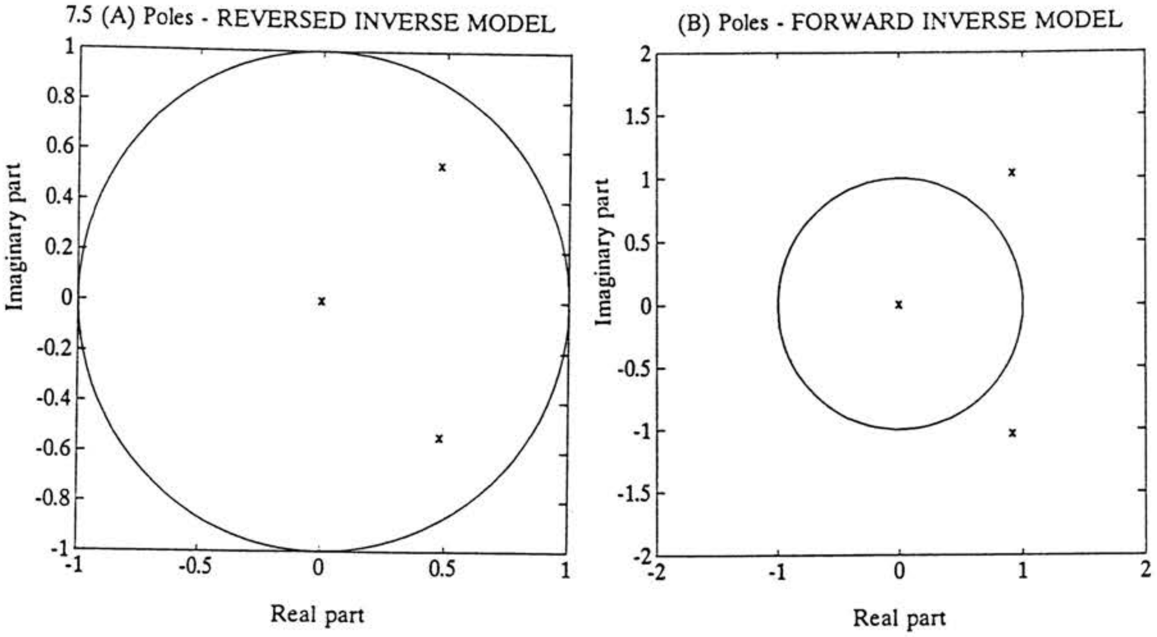


Figure 7.5: Pole positions of the reversed- and forward inverse models

7.3.2 Stable Forward Inverse - Unstable Reversed Inverse

Modifying one element of the input matrix $\mathbf{\Gamma}$ results in the reversed inverse model becoming unstable, while the forward inverse now gives stability. By replacing the value of $\mathbf{\Gamma}(2,1) = -0.05$ with a value $\mathbf{\Gamma}(2,1) = -0.5$, the following eigenvalues are obtained, which are shown in figure 7.6 .

$$eig(\mathbf{\Phi}_{ri}) = \begin{bmatrix} 1.261 + 0.325i \\ 1.261 - 0.325i \\ 0 \\ 0 \end{bmatrix} \qquad eig(\mathbf{\Phi}_{fi}) = \begin{bmatrix} 0.7436 + 0.1919i \\ 0.7436 - 0.1919i \\ 0 \\ 0 \end{bmatrix} \quad (7.11)$$

Reversed Inverse Model

For the reversed inverse model the unstable eigenvalues shown in figure 7.6 (A) are now obtained. Because of the unstable reversed inverse, it is now not possible to identify a direct reversed inverse model. (It was found possible to identify the reversed inverse model only on the second input channel, but not on the first input channel).

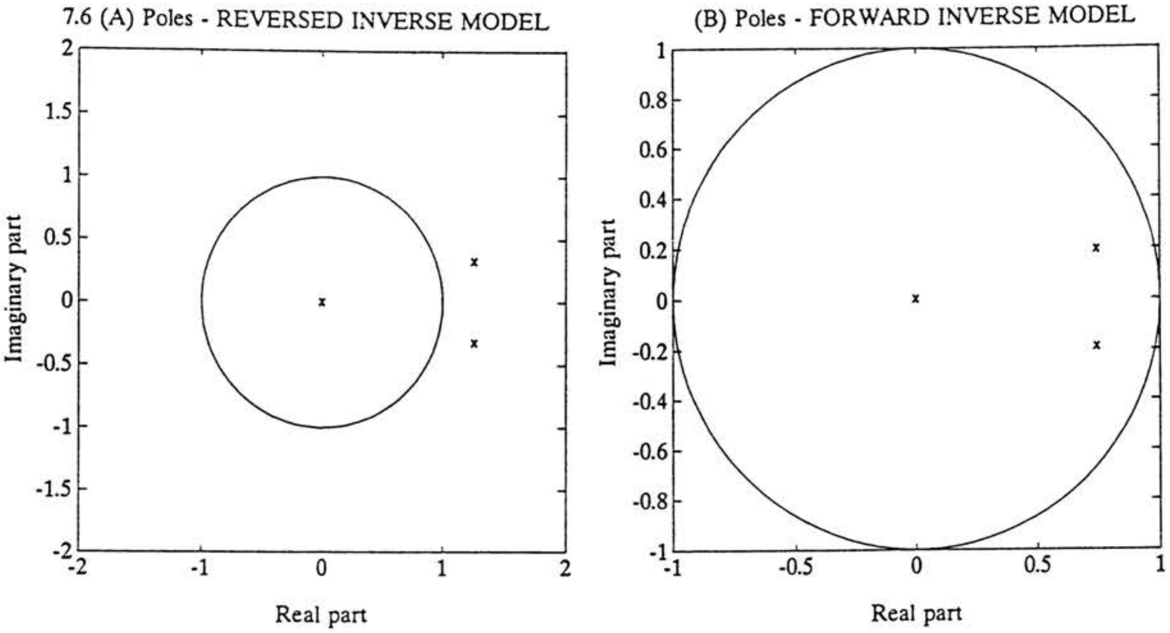


Figure 7.6: Pole positions of the reversed- and forward inverse models with $\Gamma(2,1) = -0.5$

Forward Inverse Model

In this case the eigenvalues shown in figure 7.6 (B) are obtained, which are now clearly stable.

7.3.3 Unstable Forward- and Reversed Inverses

From the above analysis, it is clear that depending on the model, a stable inverse may be found either through the reversed inverse or the forward inverse conversion. Unfortunately in some cases, both these methods fail to yield a stable inverse, and other methods need to be applied. Such a situation is found by replacing the value of $\Gamma(3,2) = 0.80$ with a value $\Gamma(3,2) = 0.20$ in the original model of equation (7.7). In this case the following real eigenvalues for the reversed inverse and forward inverse models are obtained:

$$eig(\mathbf{\Phi}_{ri}) = \begin{bmatrix} 3.000 \\ 0.800 \\ 0 \\ 0 \end{bmatrix} \qquad eig(\mathbf{\Phi}_{fi}) = \begin{bmatrix} 1.250 \\ 0.333 \\ 0 \\ 0 \end{bmatrix} \tag{7.12}$$

These results are also shown in figures 7.7 (A) and (B). Due to the

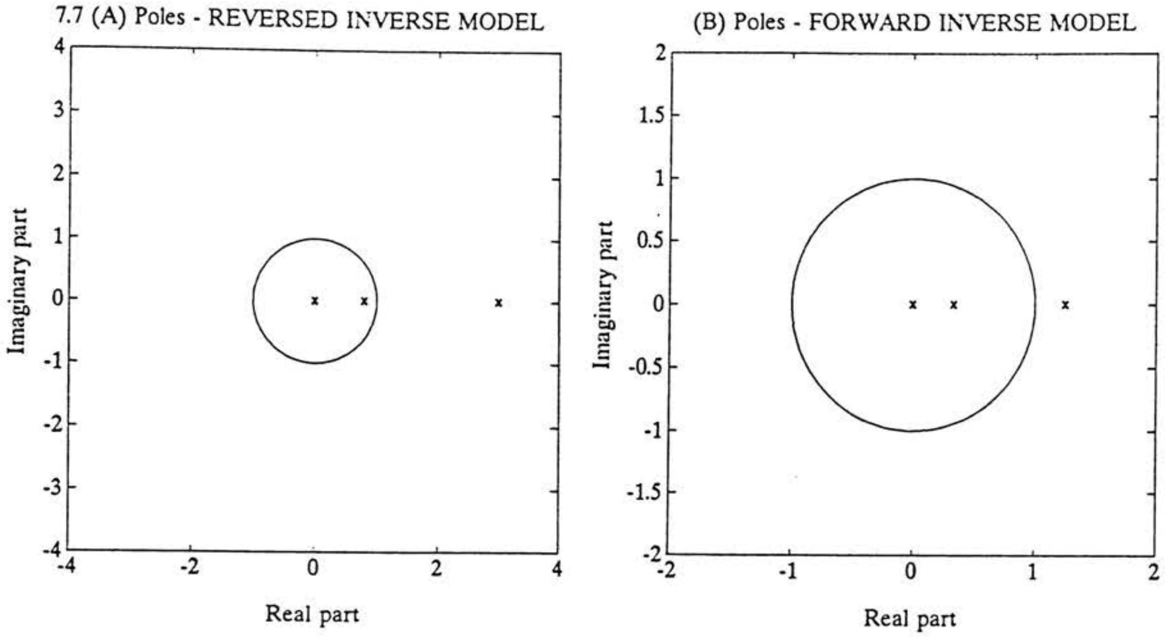


Figure 7.7: Pole positions of the reversed- and forward inverse models with $\mathbf{\Gamma}(3,2) = 0.2$

unstable reversed inverse it is again not possible to identify a direct reversed inverse model. (In this case, it was found possible to identify the reversed inverse model only on the first input channel, but not on the second input channel).

In the above case the model cannot be inverted, and another solution must be found. One possible method of finding the system inputs from the responses is to apply the linear quadratic optimal servo-controller which will be discussed in section 7.5.

7.4 Two Channel Spring-mass-damper System

Figure (7.8) shows a modified version of a simple two degree of freedom spring-mass-damper system taken from D'Souza and Garg [1984], together with the relevant model parameters. Applying Newtonian or Lagrangian dynamics results in the equations of motion:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \end{Bmatrix} + \begin{bmatrix} (c_1+c_2) & -c_2 \\ -c_2 & (c_2+c_3) \end{bmatrix} \begin{Bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{Bmatrix}$$

$$+ \begin{bmatrix} (k_1+k_2) & -k_2 \\ -k_2 & (k_2+k_3) \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} F_1(t) \\ F_2(t) \end{Bmatrix} \tag{7.13}$$

The numerical values of the mass, damping and stiffness matrices are chosen so as to achieve significant cross-coupling between the channels, since it is also desired to demonstrate the applicability of the identification process in a highly cross-coupled situation. These are given by:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 25 & -10 \\ -10 & 60 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 1100 & -300 \\ -300 & 400 \end{bmatrix} \tag{7.14}$$

with the basic units of [kg], [Ns/m] and [N/m] respectively. Employing x_i as generalized coordinates, the continuous state and output equations are given by

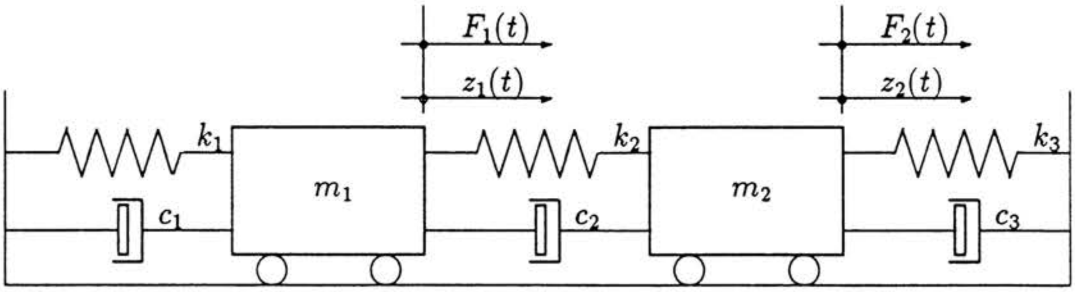


Figure 7.8: Two degree of freedom system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(k_1+k_2)}{m_1} & \frac{-(c_1+c_2)}{m_1} & \frac{k_2}{m_1} & \frac{c_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} & \frac{-(k_2+k_3)}{m_2} & \frac{-(c_2+c_3)}{m_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

and

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{7.15}$$

With the above matrices, the units for the force vector [**u**] are in Newtons, while the displacement vector [**y**] has its units in millimeters. The reason for this choice is to achieve values for [**u**] and [**y**] which have the same order of magnitude, resulting in a well conditioned numerical problem, which is essential for good results in the identification process. The model has furthermore also been chosen so as to achieve a significant amount of cross-coupling for the purpose of identifying the model in a multivariable situation. The continuous state equation may be integrated according to equations (A.12) and (A.13) from appendix A, to give the equivalent discrete state equation.

7.4.1 Model Identification

Pseudo random white noise excitation from 0.2 to 6 Hz was input as forces to both masses. Figure 7.9 shows the inputs and resultant output responses of the two masses, and is referred to as the "experimental" data. This data was used to identify the model. A direct reversed inverse model could not be identified successfully due to instability, and instead it was necessary to identify normal forward ARX models for each output channel. These ARX models were thereafter combined to the state space formulation, giving a total of four states. Figure 7.10 shows the simulated model outputs plotted over the experimental outputs. Also shown in dotted lines, are the residuals, indicating a perfect fit, which proves the integrity of the identification process in a multivariable setting.

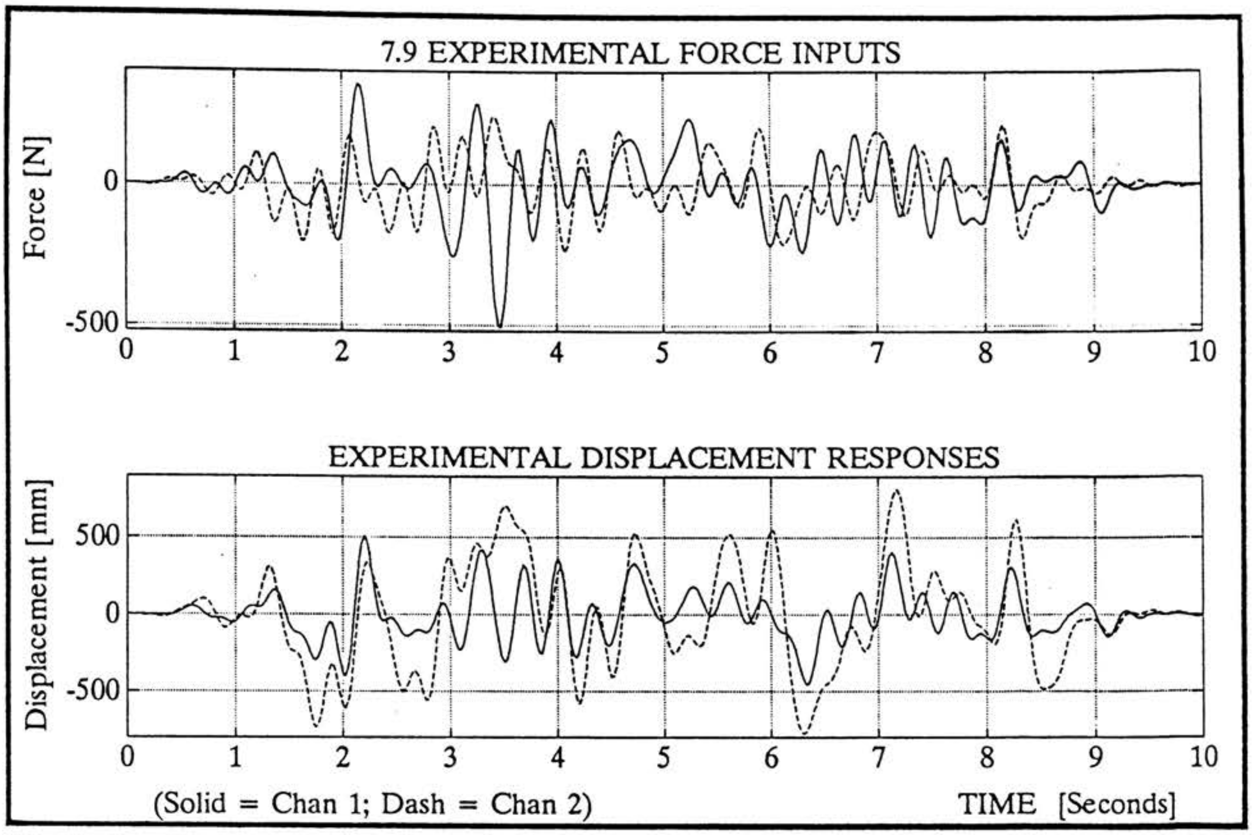


Figure 7.9: "Experimental" input-output data for m-c-k system

7.4.2 Model Inversion

Since a reversed inverse model could not be identified, the inversion to a reversed inverse model naturally also gave an unstable inverse. Stable transmission zeros were however found, and the state space model was consequently inverted to a forward inverse model.

7.4.3 "Practical Test Rig"

Since the correlation between the "experimental" and simulated model data was identical, the original system was slightly modified to achieve a slight "practical mismatch" between the identified model and the original system, similar to which one might typically find between any mathematical model and the actual measured practical test rig. The mass, damping, and stiffness matrices were hence modified to:

$$\mathbf{M} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.9 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 27 & -12 \\ -12 & 52 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 1000 & -250 \\ -250 & 370 \end{bmatrix} \tag{7.16}$$

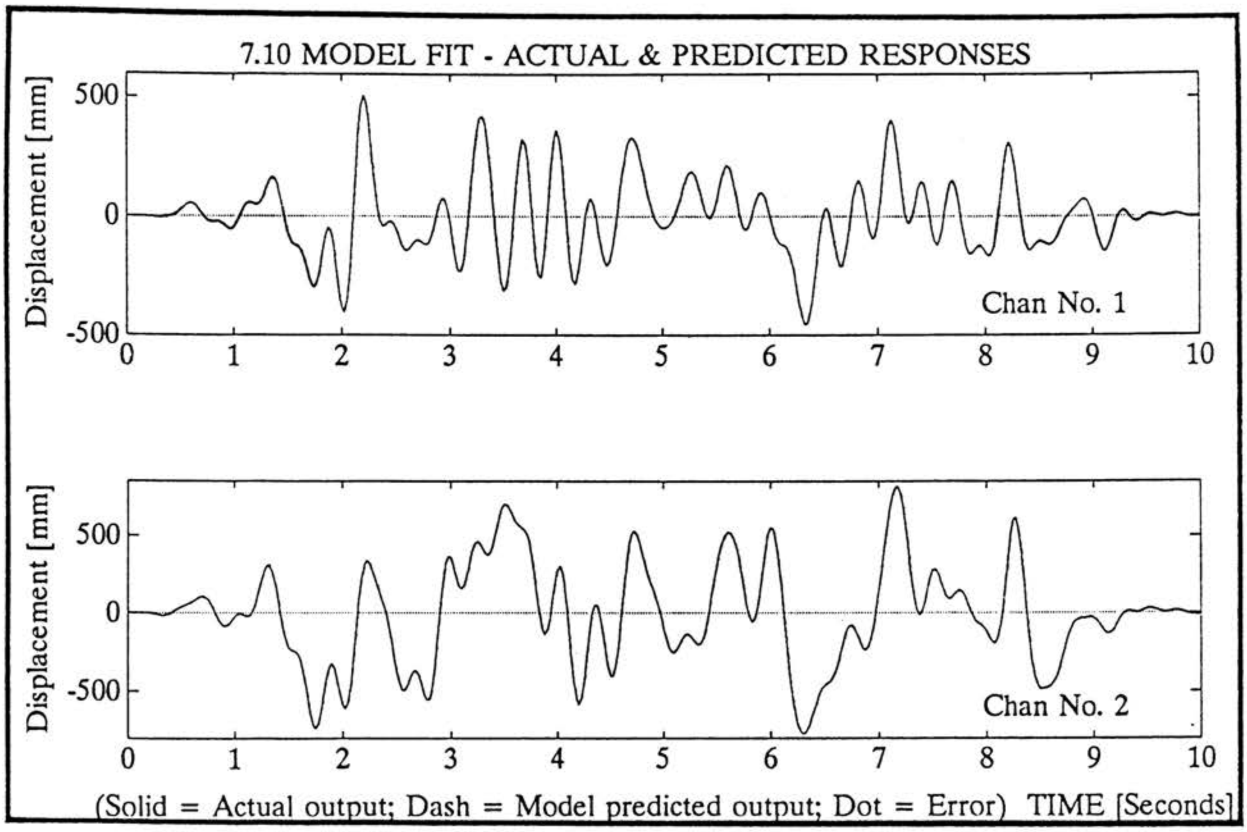


Figure 7.10: Model fit for m-c-k system

Using the above system, which will be referred to as the "test rig", new pseudo random signals with spectral content ranging from 0.1 to 6 Hz for the first input channel , and 0.05 to 2 Hz for the second input channel, were input to achieve the "operational measured" desired responses, which are shown in figure 7.11 .

7.4.4 Linear Solution Responses

The desired responses were input to the inverted state space model to give the "test rig" drive signals, which were again used to excite the "test rig". Figure 7.12 shows the "test rig" achieved responses plotted with the desired "operational measured" responses. Also shown in dotted lines in the figure, are the errors in responses for the two output channels.

7.4.5 Iteration Process

Using the response error signals the drive signal corrections could be found from the inverted model to determine the updated drive signals. Applying

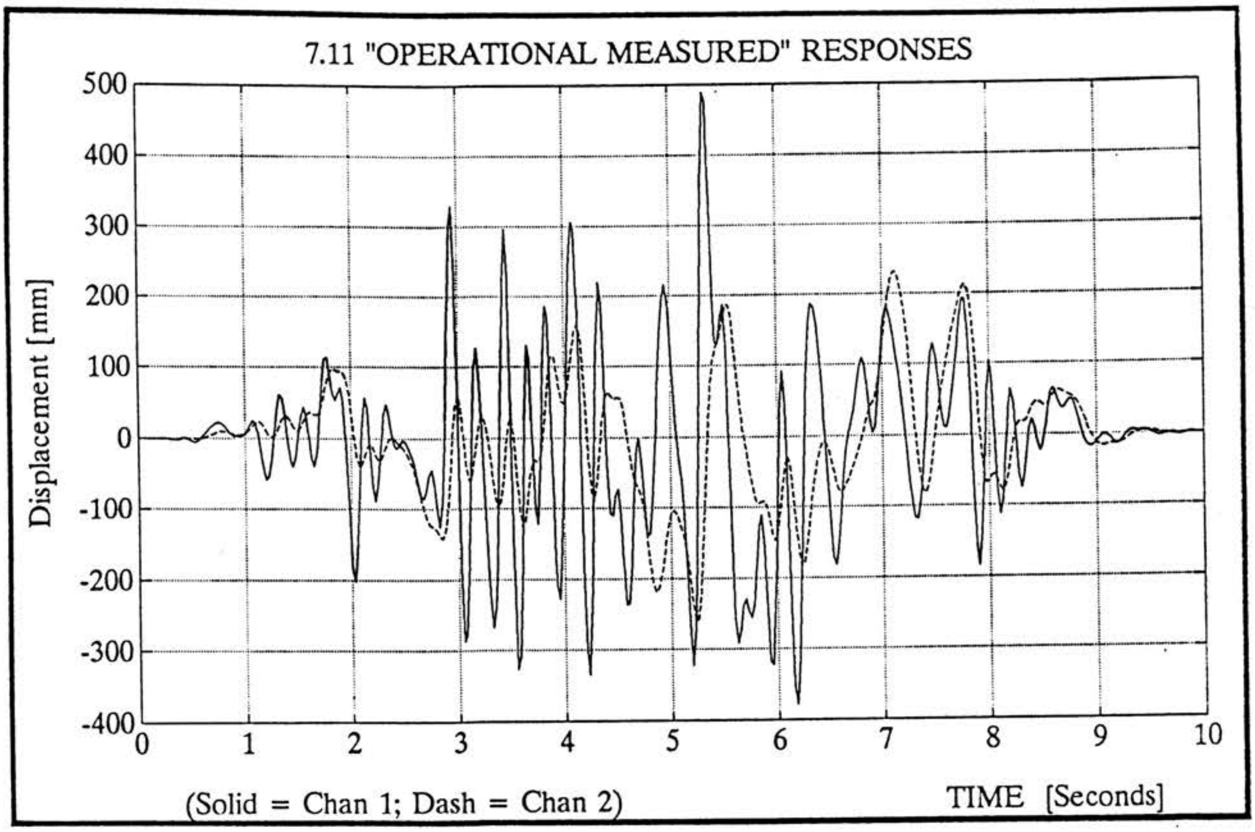


Figure 7.11: "Operational measured" desired responses from "test rig"

STAGE OF TEST	PERCENT "FIT"		FIGURE
	Chan. 1	Chan. 2	NUMBER
Linear solution (η_{sim})	26.4%	17.0 %	7.12
Iteration No. 1 (η_{sim})	7.0%	4.1 %	7.13
Iteration No. 3 (η_{sim})	0.5%	0.3 %	7.14

Table 7.1: Simulation results η_{sim} for m-c-k system

these updated drive signals to the "test rig", resulted in an improved response as shown in figure 7.13 . Similar results after three iterations are presented in figure 7.14, which ultimately also proves the integrity of the iteration process in a multiple-input multiple-output application.

Recalling the simulation accuracy parameter η_{sim} defined in equation (6.11), the simulation results are summarized in table 7.1.

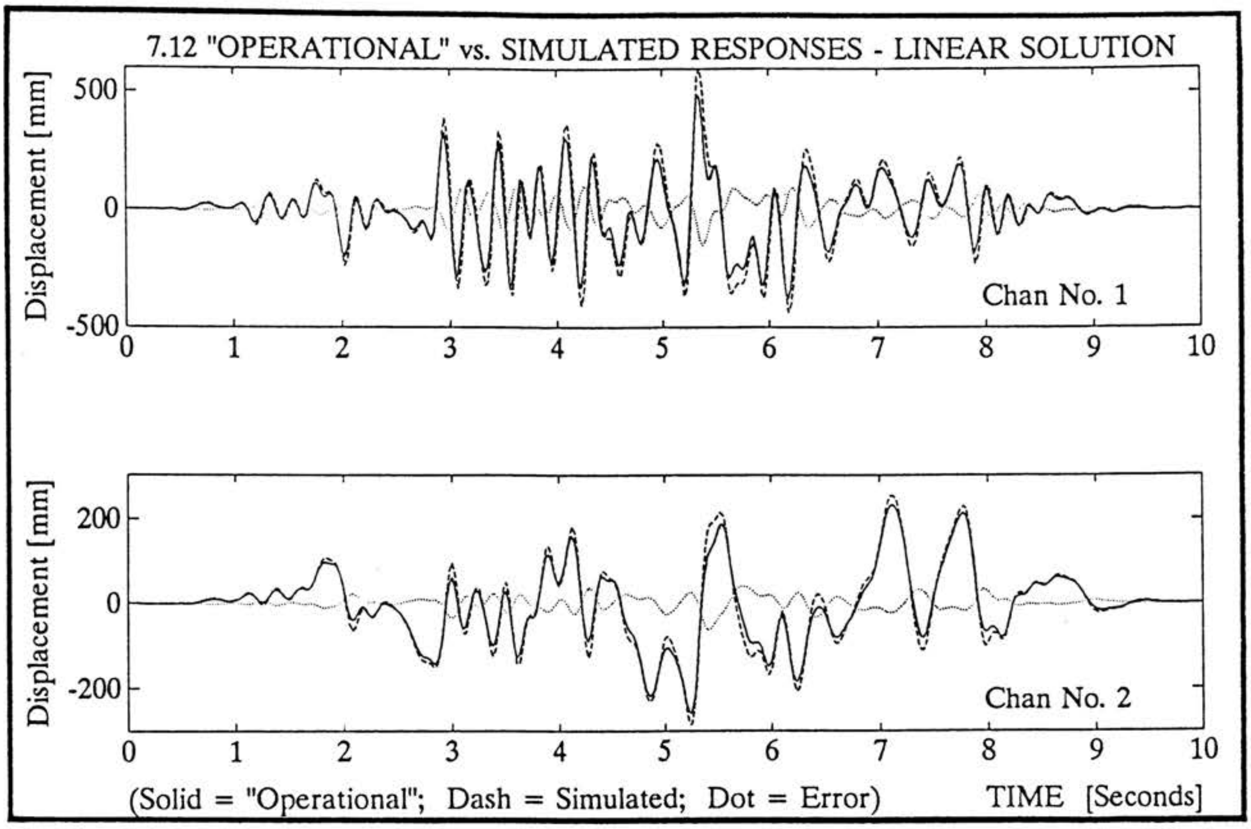


Figure 7.12: Desired and achieved responses - linear solution

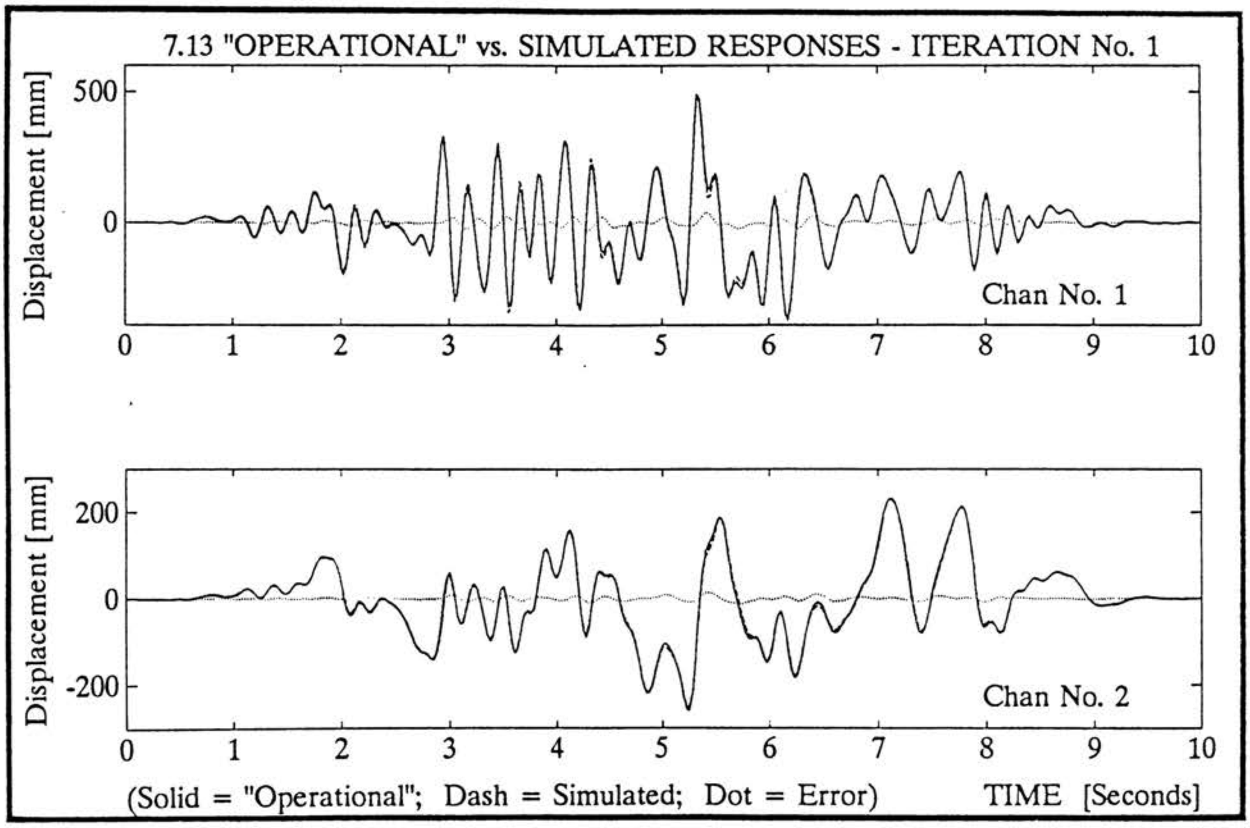


Figure 7.13: Desired and achieved responses - Iteration No. 1

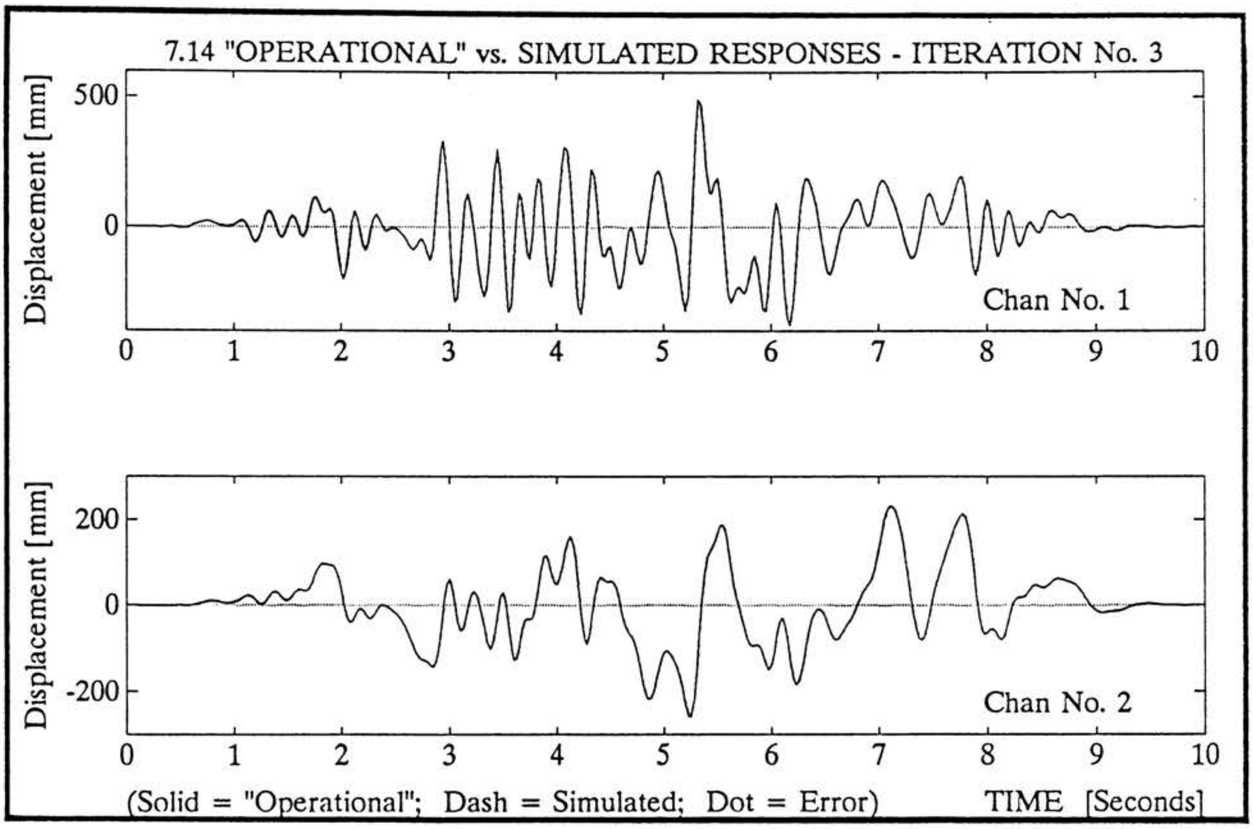


Figure 7.14: Desired and achieved responses - after three iterations

7.5 Application of the Optimal Servo-controller

By simply introducing no-delay terms in the previous two channel system, instability can occur for both the forward inverse and reversed inverse models.

For this purpose the direct transmission matrix is modified to

$$\mathbf{D} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \tag{7.17}$$

Following a similar procedure to the previous section, pseudo random excitation inputs are applied, giving the system input-output data, from which a normal forward model is identified. Attempting both forward inverse and reversed inverse conversions leads to the following poles or eigenvalues

$$eig(\mathbf{\Phi}_{ri}) = \begin{bmatrix} -1.828 \\ 0.506 + 0.872i \\ 0.506 - 0.872i \\ 0.130 \end{bmatrix} \qquad eig(\mathbf{\Phi}_{fi}) = \begin{bmatrix} 7.680 \\ 0.498 + 0.858i \\ 0.498 - 0.858i \\ -0.547 \end{bmatrix} \tag{7.18}$$

which are clearly unstable in both cases. The identification of a direct inverse model is naturally also not possible.

The only solution to finding the system inputs from the system responses is the implementation of the linear quadratic optimal servo-controller.

Dynamic System Delays

The dynamic system delays are found to be 3 on each input channel as discussed in section 5.8.4, and are found by varying the number of delay periods as shown in figure 7.15 .

Response to Initial Inputs

Using the system inputs as extracted from the LQ optimal servo-controller synthesis, the responses to these optimal inputs from a subsequent simulation of the dynamic model, give the result shown in figure 7.16. Although being optimal in the sense of the LQ controller, the calculated inputs are clearly not entirely correct, and a number of corrections are required.

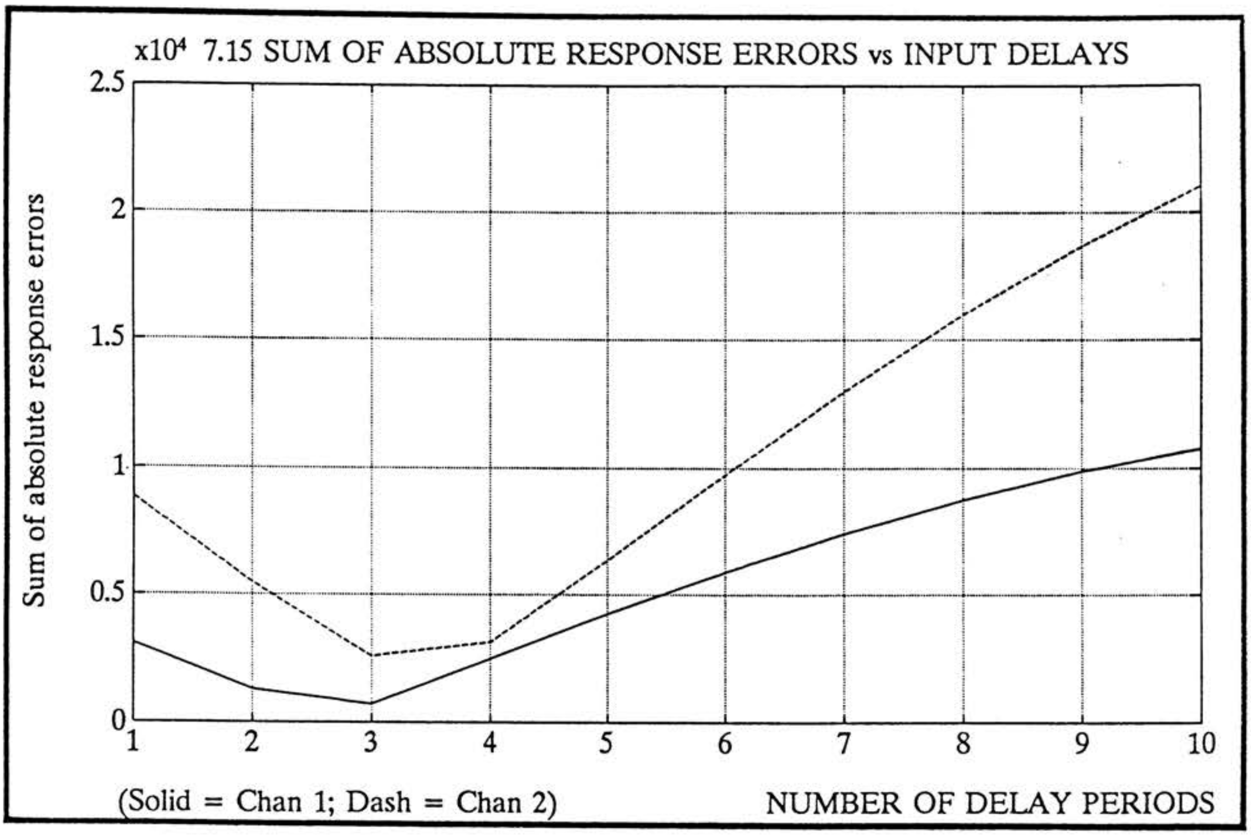


Figure 7.15: Determination of the required number of input delay periods

Iterative Corrections

As was explained in section 5.8.3, iterative corrections are required to achieve the final inputs. The sum of absolute errors is reduced by each correction step, which is shown in figure 7.17. Figures 7.18 and 7.19 show the results for the first two correction steps. The simulation results are also shown in table 7.2.

Clearly the linear quadratic optimal servo-controller performs well in obtaining the system inputs from the system responses. The success of this

STAGE OF TEST	PERCENT "FIT"		FIGURE NUMBER
	Chan. 1	Chan. 2	
LQ optimal input (η_{sim})	9.4 %	14.6 %	7.16
Input correction No. 1 (η_{sim})	1.3 %	3.4 %	7.18
Input correction No. 2 (η_{sim})	0.5 %	1.1 %	7.19

Table 7.2: Simulation results η_{sim} for LQ optimal servo-controller application

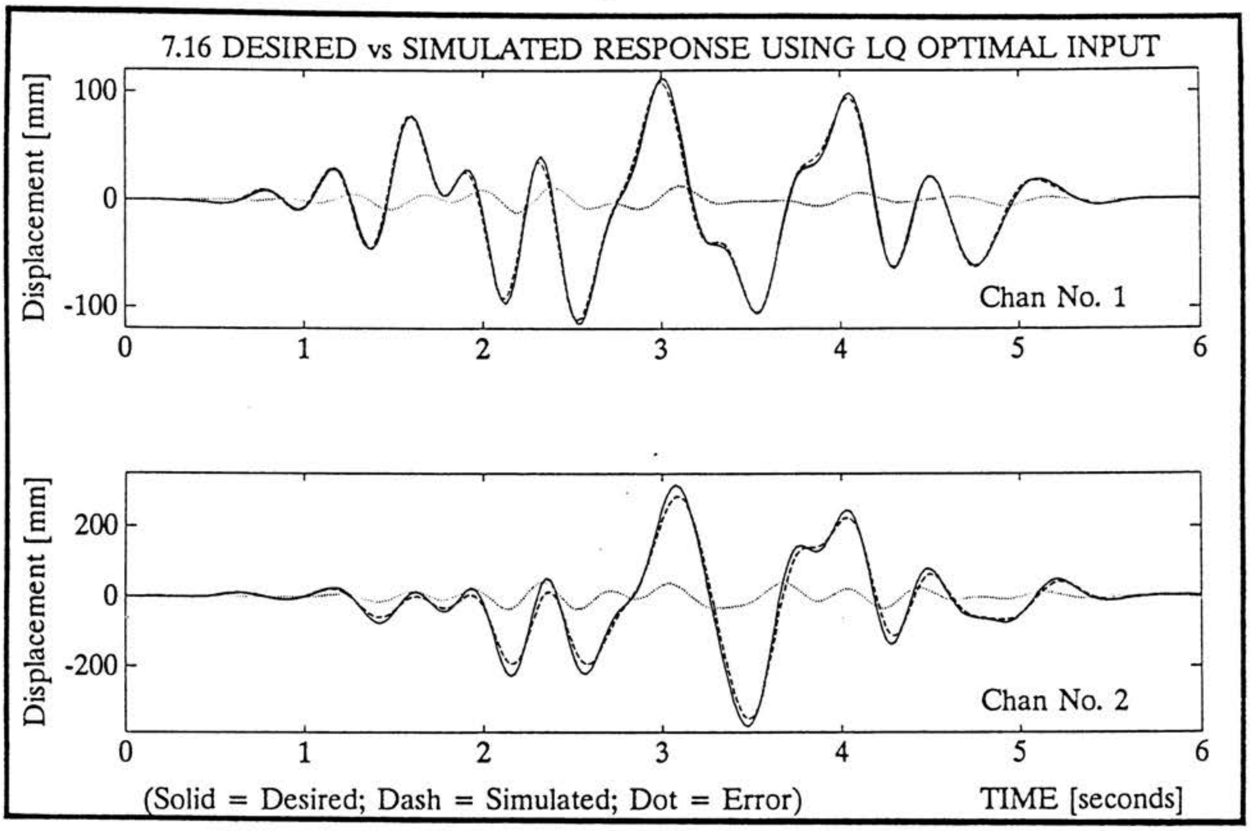


Figure 7.16: Response to LQ optimal servo-controller calculated inputs

method is however also a function of the dynamic performance in terms of its step response of the specific system at hand.

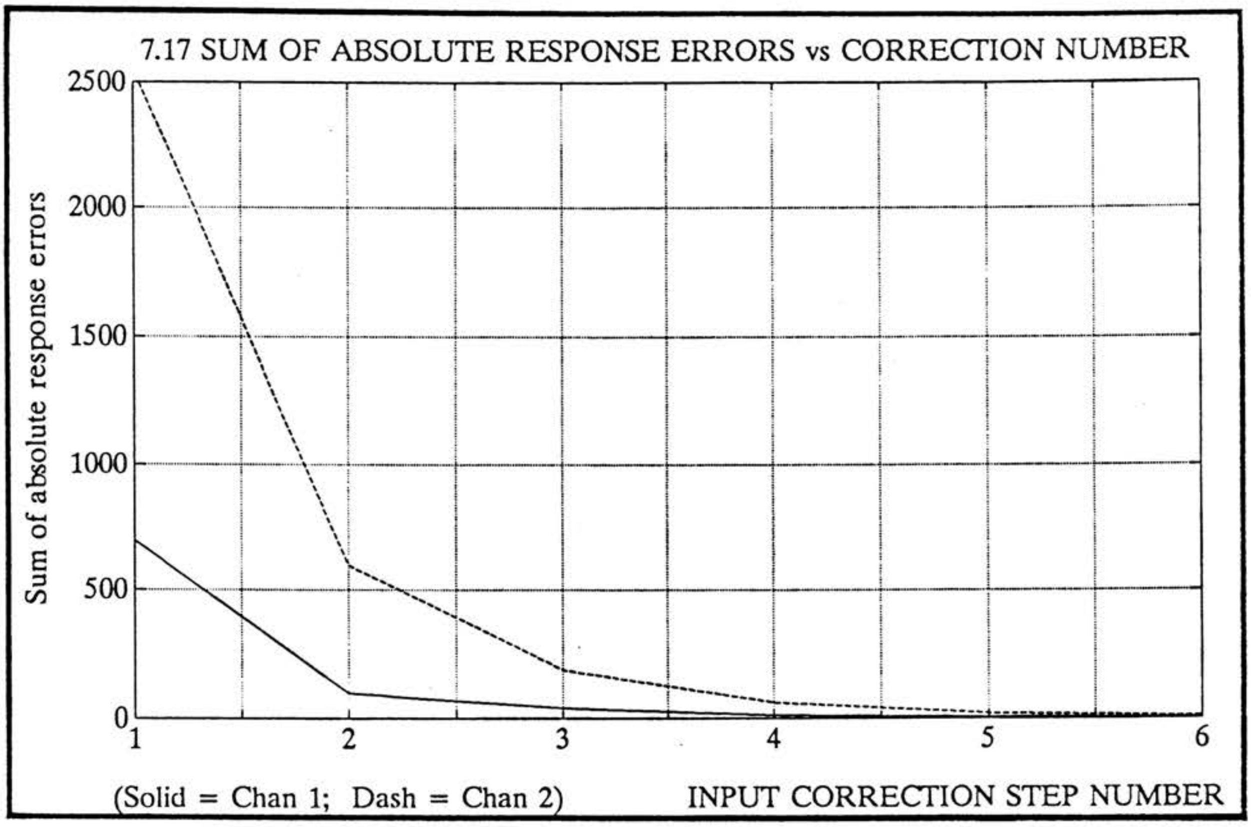


Figure 7.17: Sum of absolute response errors at each input correction step

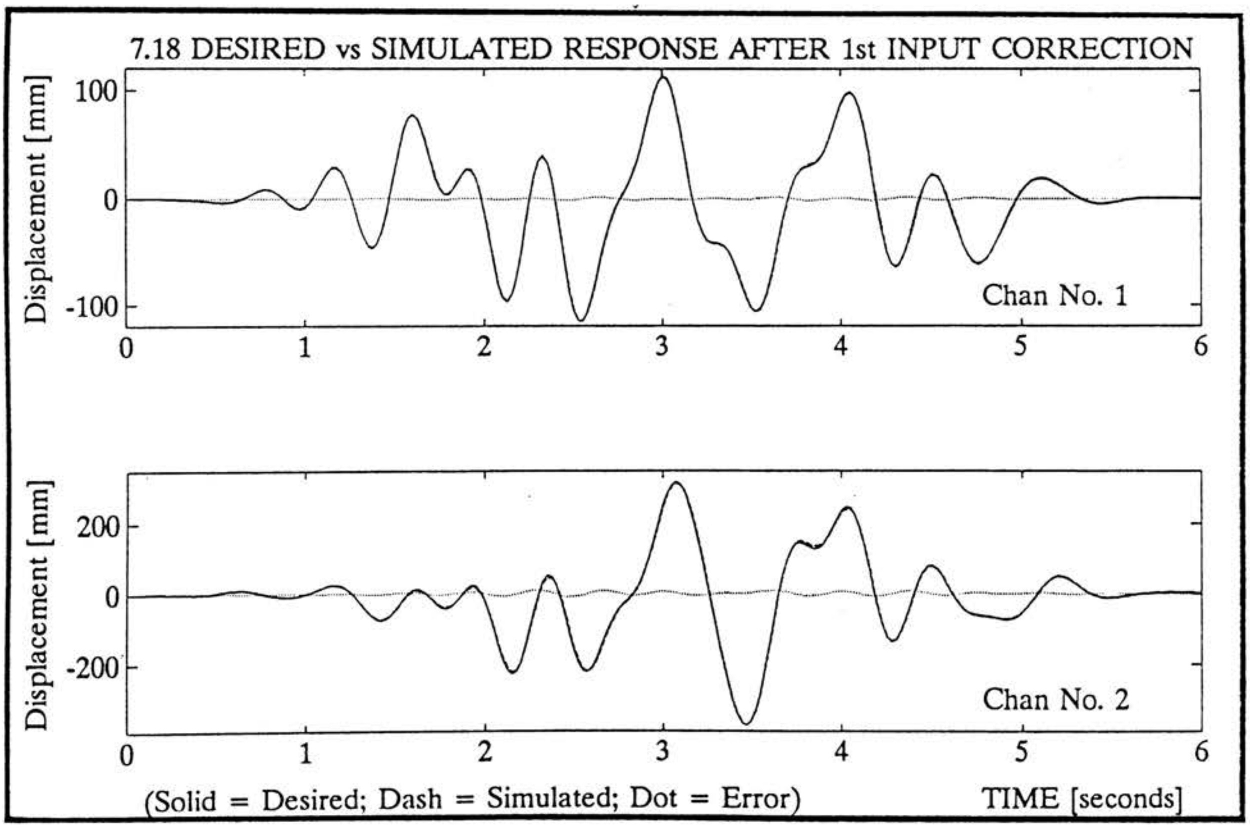


Figure 7.18: Desired and simulated responses after first input correction step

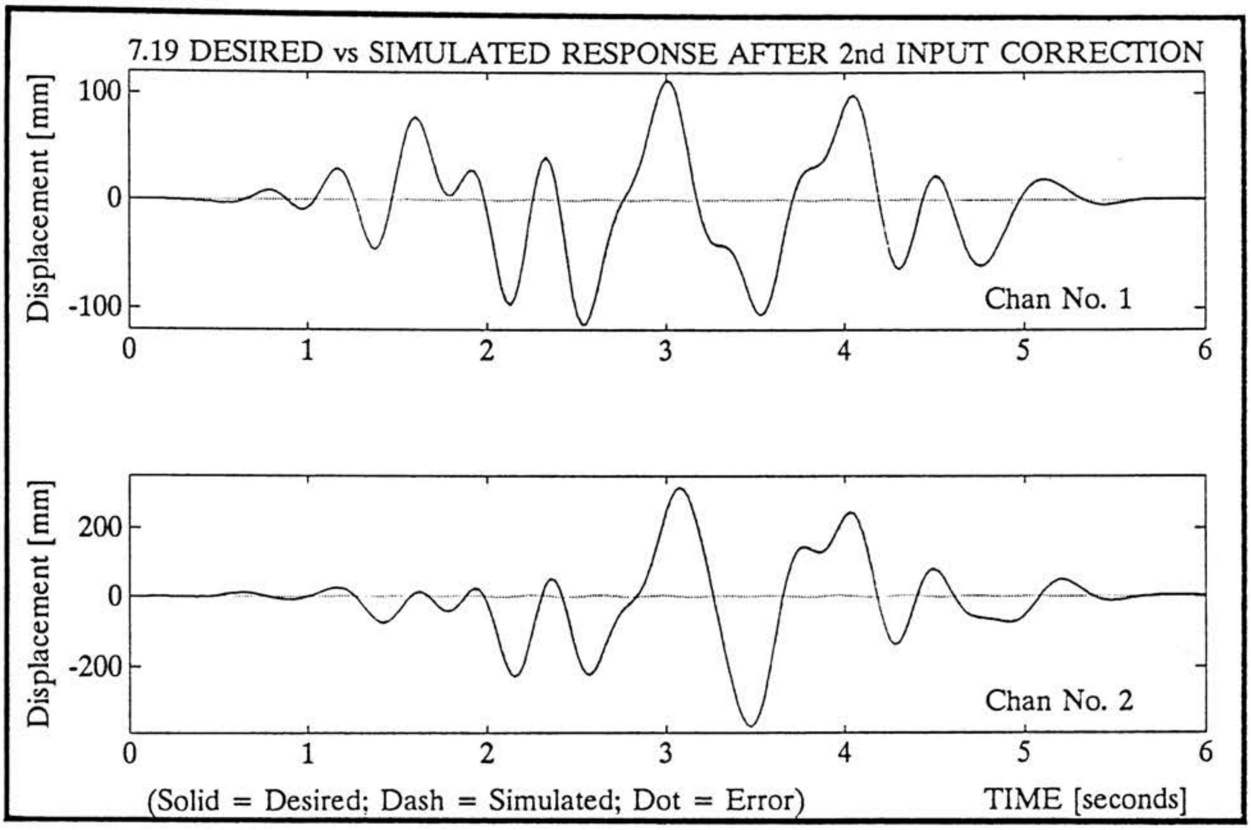


Figure 7.19: Desired and simulated responses after second input correction step

7.6 Summary

The objective of this chapter was to prove the integrity of the computer programs. This was achieved by applications to a variety of analytical simulation tests.

Dynamic System Identification in a Multivariable Setting

This aspect was addressed in section 7.4, where the specific system was chosen in such a way that significant cross-coupling between the two channels was achieved. The identification of multiple-input single-output ARX models on each output channel was utilized, whereafter these models were combined into a complete multivariable state space model. A comparison of the experimental outputs with the simulated outputs gave a perfect fit, indicating that the generation of a multivariable model from experimental input output data may in fact be achieved successfully.

Inversion Techniques

Virtually all inversion algorithms presented in chapter 5 were applied, giving identical stability results for the various chosen inversion methods. Where instability was found, alternative inversion algorithms were required to invert the model.

Identifiability of Reversed Inverse Models

In section 7.3 it was shown that if the system exhibits unstable eigenvalues after converting to the reversed inverse model, it is then also not possible to directly identify a reversed inverse model. At most, it may be possible to identify the reversed inverse model only on some of the input channels, but not on all. An important point to be mentioned however, is that a normal model may always be found, however the inversion of the normal model may then not be possible due to unstable inverse model eigenvalues.

Linear Quadratic Optimal Servo-controller

When instability of the inverse model cannot be overcome it is still possible to find the system inputs from the desired responses by implementing a linear quadratic optimal servo-controller. This was demonstrated in the case study of section 7.5, where a number of off-line corrections were required to achieve the final input signals.

Chapter 8

Applications: Practical Case Studies

8.1 Introduction

Four practical case studies, in order of increasing complexity, were performed, each addressing different important aspects of typical real world practical engineering problems:

- The first case study involves a single servo-hydraulic actuator test rig, which serves as a high speed spring tester.
- A three channel vehicle suspension was used for the next case study by simulating actual measured operational conditions on a laboratory test rig.
- A four channel simulation of measured responses on a passenger vehicle was employed for the third case study.
- Finally a five channel simulation of measured responses on a pick-up truck chassis served as the final practical case study.

Photographs of the mechanical laboratory test rigs are presented in Appendix E. For comparative purposes the first two case studies were also repeated using the frequency domain techniques, which is dealt with in chapter 9.

Accuracy of Laboratory Simulations

In all practical case studies, measured response signals were recorded during actual operational conditions and simulated on laboratory test rigs. In none of the cases were any form of synthetic "operational responses" utilized. This results in a much tougher and revealing test. It is therefore essential to appreciate that the simulation inaccuracies are essentialy ascribed to the mechanical restrictions of the various servo-hydraulic test rigs.

8.2 Case Study 1: Single Servo-hydraulic Actuator High Speed Spring Tester

The fatigue life of shock loaded springs as is found in engine valve springs or pneumatic equipment may conveniently be simulated in the laboratory using a servo-hydraulic actuator. In these applications the springs are invariably exposed to extremely high velocities and accelerations. Typical displacement response curves were chosen and simulated on a test rig. Full details of the tests may be found in Raath [1991]. This section describes the application of the time domain based techniques, whereas a comparison to the frequency domain techniques is presented in the next chapter.

8.2.1 Test Rig and Transducers

A diagrammatic layout of the test system is given in figure 8.1 which consists of a single actuator with its analogue PID controller capable of testing five springs simultaneously. Since the servo-hydraulic actuator is capable of producing a maximum velocity of 1.8 m/s, a 1 : 6.2 ratio crank was used to increase the velocity. Also shown in the figure is a linear hydrostatic bearing to guide the main shaft transmitting the loads to the springs. The actuator was driven in displacement control using it's internal LVDT as feedback to the analogue PID controller. An external LVDT measuring the actual spring displacement was used as the system response transducer.

8.2.2 Desired Operational Response

The displacement of the springs was digitized from a high speed film of a light pneumatic jack-hammer. Two cases were considered, namely a single cycle as well as a continuous operation of twenty cycles, which are depicted in figure 8.2 .

8.2.3 Dynamic Model Identification

In the specific application it was essential to simulate the *shape* of the desired response curve accurately and not only the peak amplitudes. The wave form was shown to influence the dynamics of the spring, which in turn influenced the spring stresses and hence the fatigue life. To achieve this objective, required a relatively high sampling frequency, which was chosen at 256 Hz. The spectral content of the operational measured response data is shown in figure 8.3, dictating the high sample frequency. A total of 4 seconds of data was used in the identification, resulting in 1024 data points. The experimental input-output data is shown in figure 8.4 . A direct reversed inverse model of 19th order was identified, and figure 8.5 shows the actual vs predicted input excitation. The identified model was thereafter transformed to state space using the equations from appendix B, resulting in 19 states.

8.2.4 Results for Single Cycle Operation

The single cycle desired operational response from figure 8.2 was input to the reversed inverse state space model to give the linear solution to the actuator drive signal. The test rig was excited with this signal, and figure 8.6 shows the laboratory achieved response plotted with the desired operational measured response. Also shown in the figure is the error in response. Figure 8.7 shows the same results after one iteration, while figure 8.8 shows the results after four iterations. Results for the model fit and simulation accuracy are shown in table 8.1.

8.2.5 Results for Continuous Operation

The above results were similarly also obtained for the continuous operation desired response of figure 8.2 . The corresponding results are shown in figures 8.9 through 8.12 as well as table 8.2.

STAGE OF TEST	PERCENT "FIT"	FIGURE NUMBER
Model fit (η_u)	7.6 %	8.5
Linear solution (η_{sim})	10.0 %	8.6
Iteration No. 1 (η_{sim})	3.1 %	8.7
Iteration No. 4 (η_{sim})	1.8 %	8.8

Table 8.1: Model fit η_u and simulation results η_{sim} for single cycle operation

STAGE OF TEST	PERCENT "FIT"	FIGURE NUMBER
Linear solution (η_{sim})	9.7 %	8.9 and 8.10
Iteration No. 1 (η_{sim})	3.4 %	8.11 and 8.12

Table 8.2: Simulation results η_{sim} for continuous operation

8.2.6 Conclusion

The simulation of operational responses on impact loaded systems is easily achieved by the time domain technique. These systems are invariably restricted by very little data, which makes the time domain technique ideal for such an application.

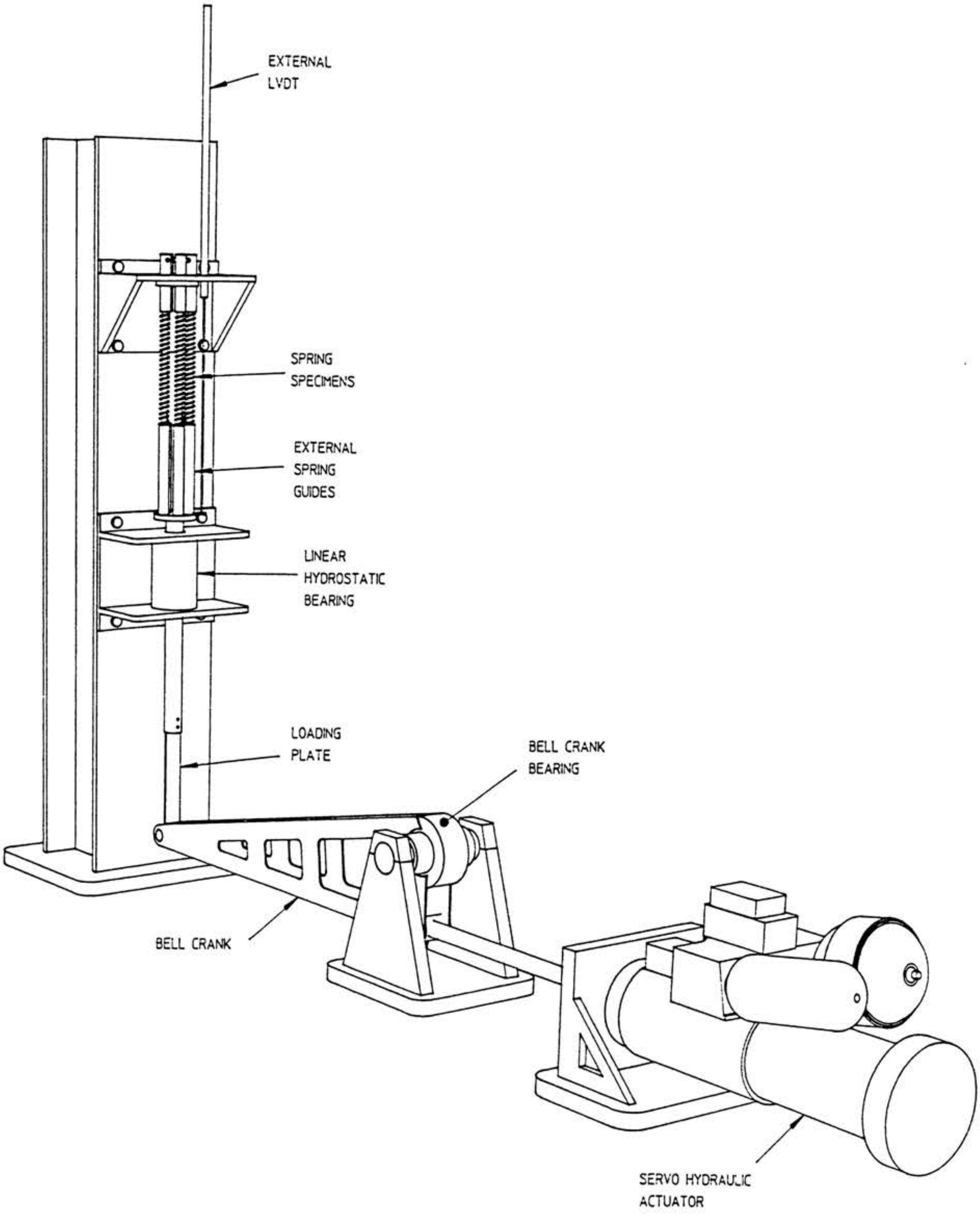


Figure 8.1: High speed spring loading simulator

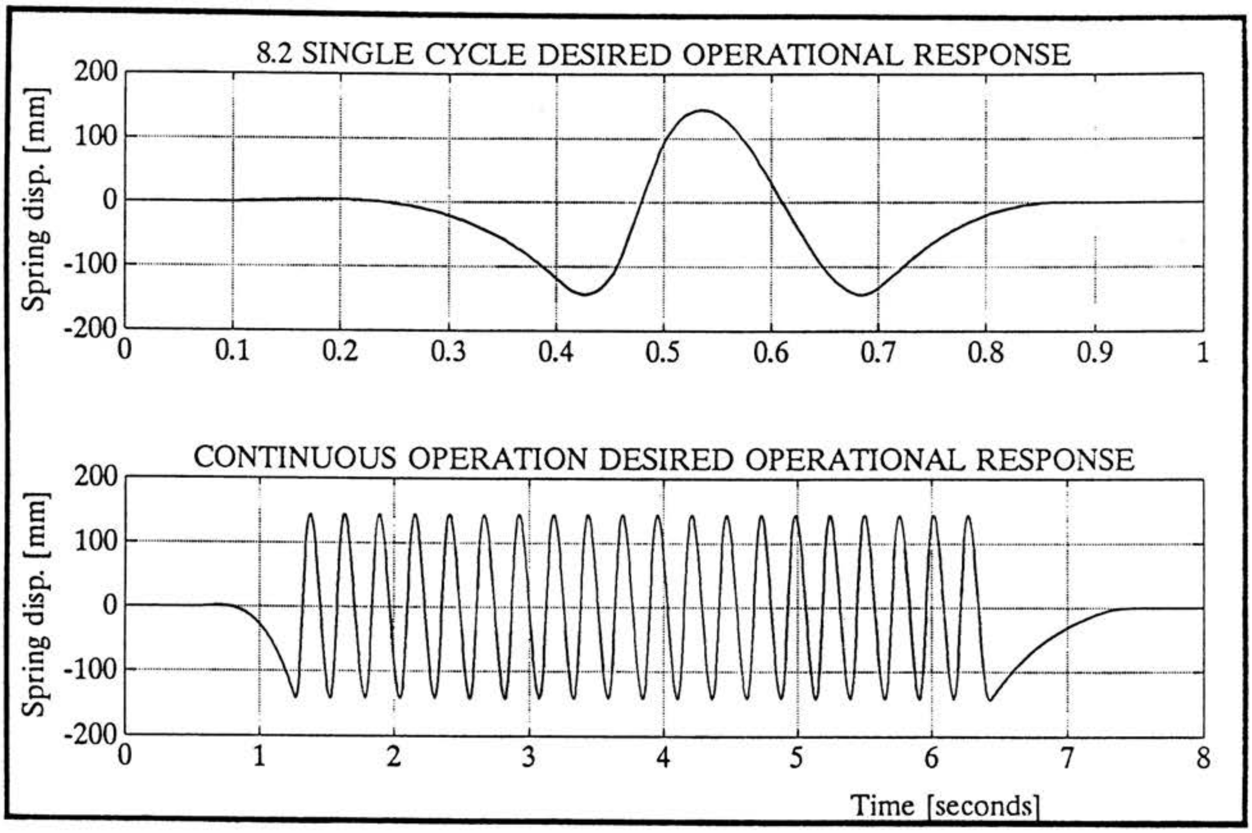


Figure 8.2: Desired operational spring displacement responses

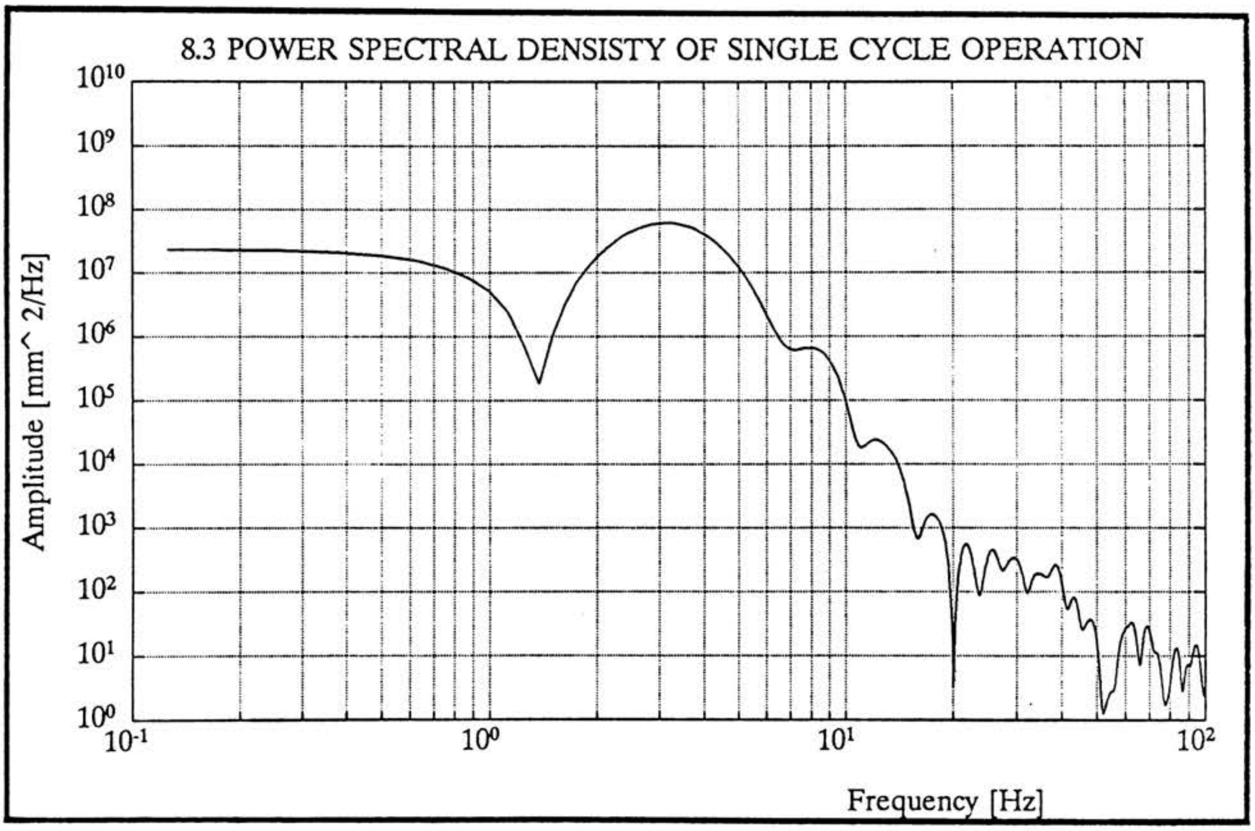


Figure 8.3: Spectral content of desired operational response

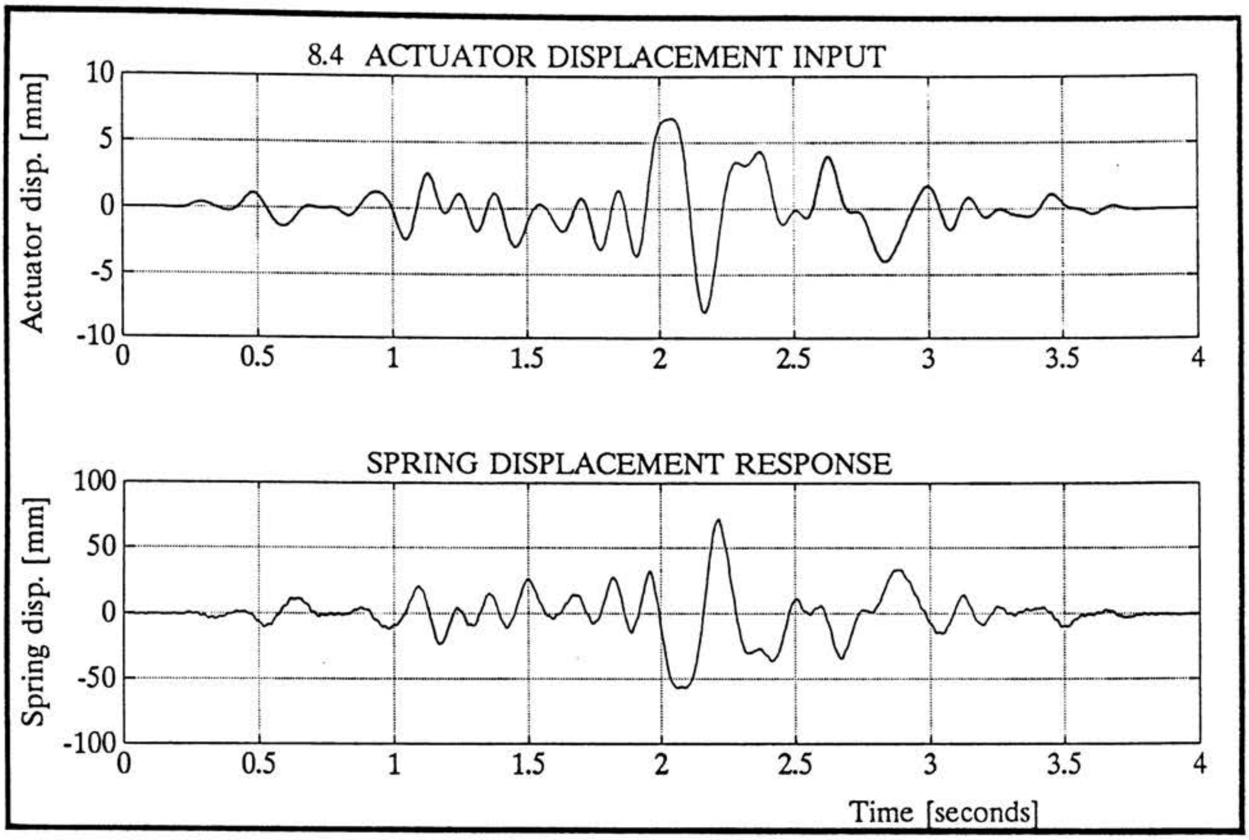


Figure 8.4: Experimental input-output data for identifying the dynamic model

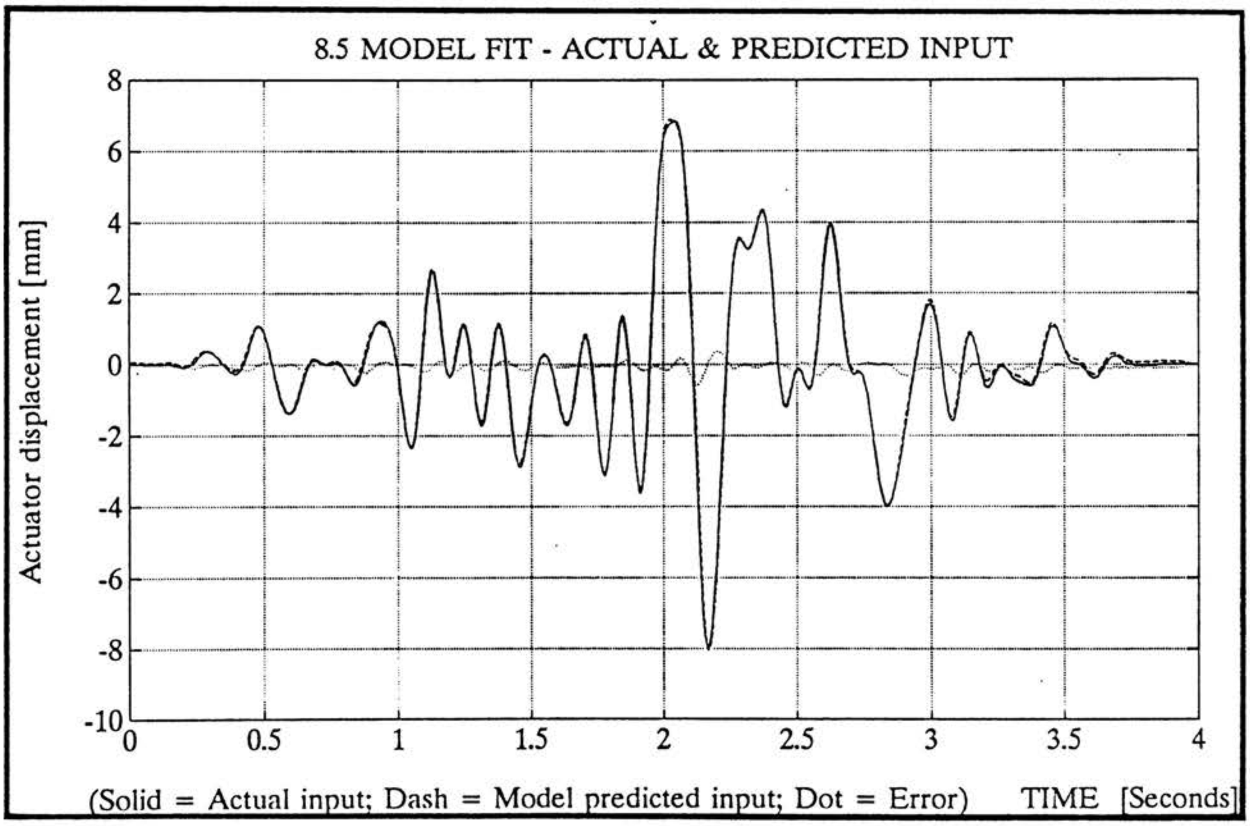


Figure 8.5: Model fit - actual and predicted system input

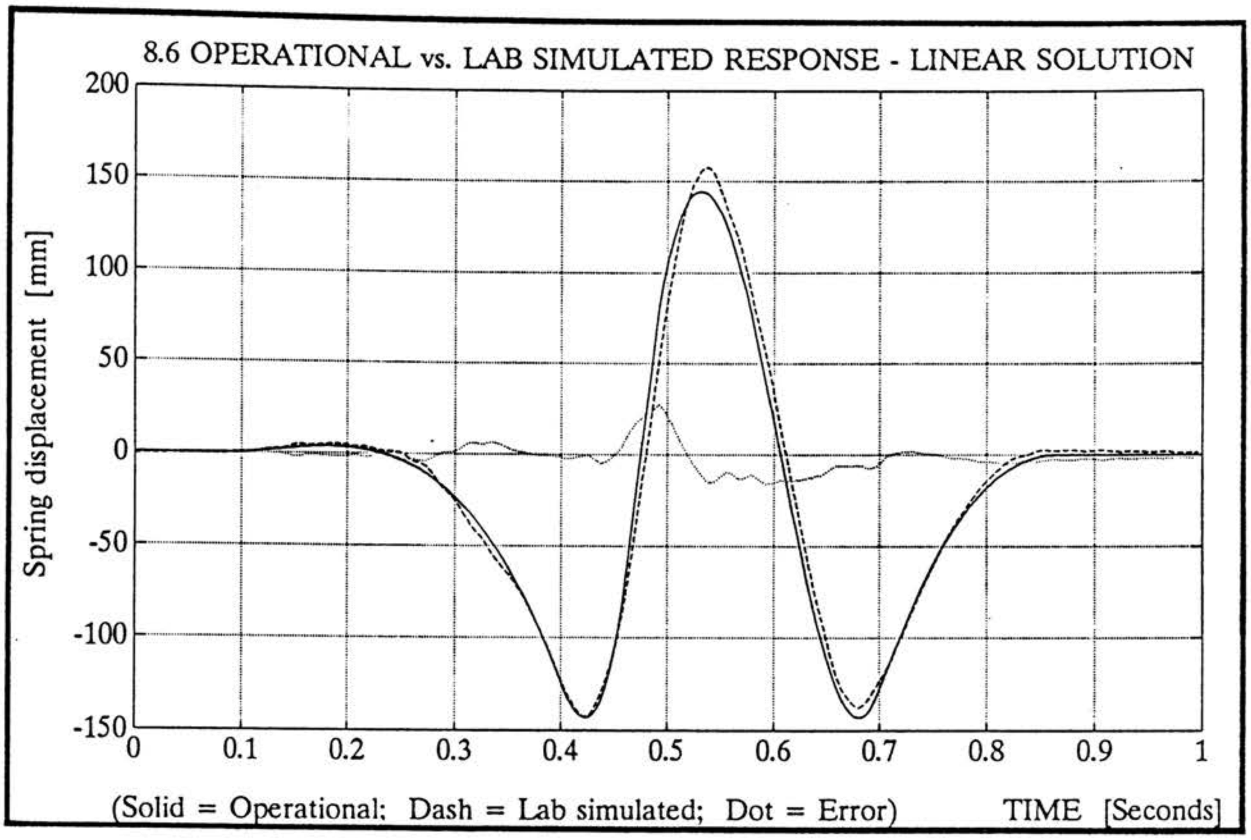


Figure 8.6: Desired operational- and laboratory achieved response - linear solution

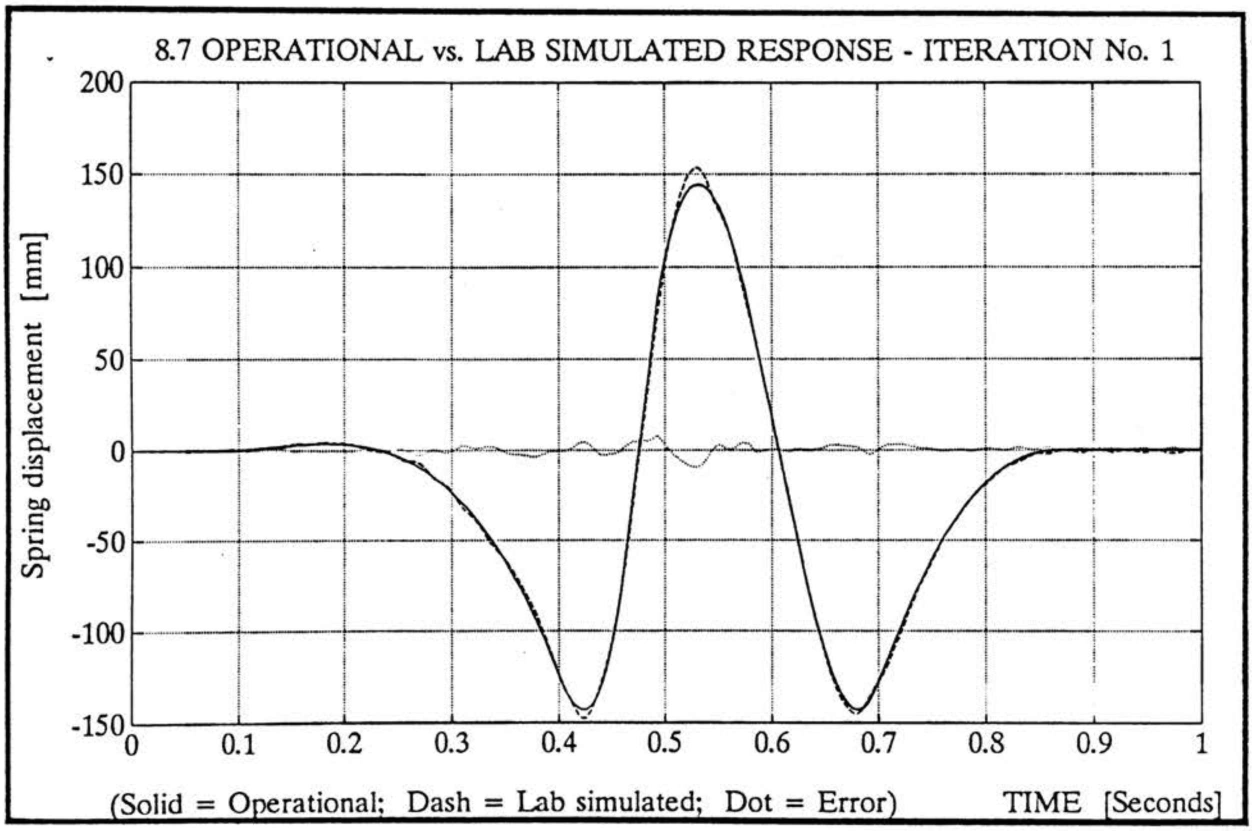


Figure 8.7: Desired operational- and laboratory achieved response - after one iteration

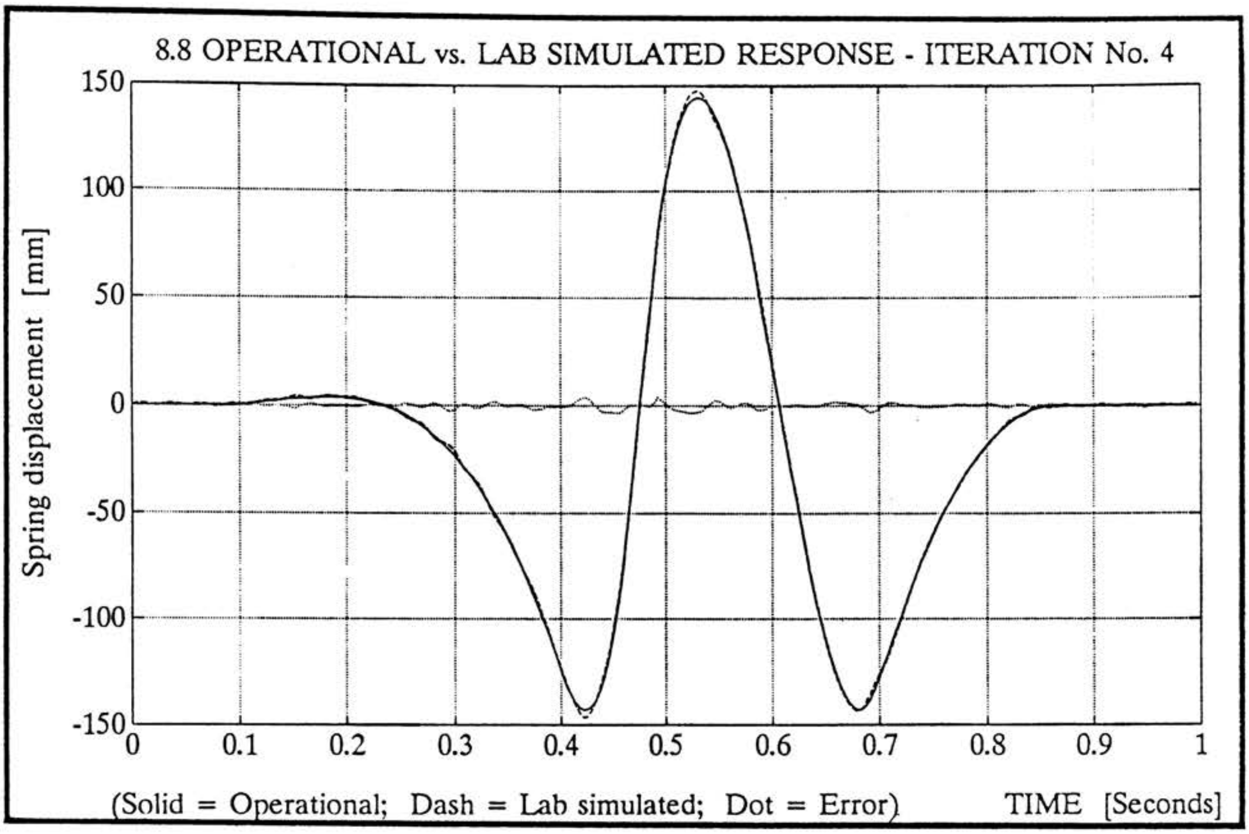


Figure 8.8: Desired operational- and laboratory achieved response - after four iterations

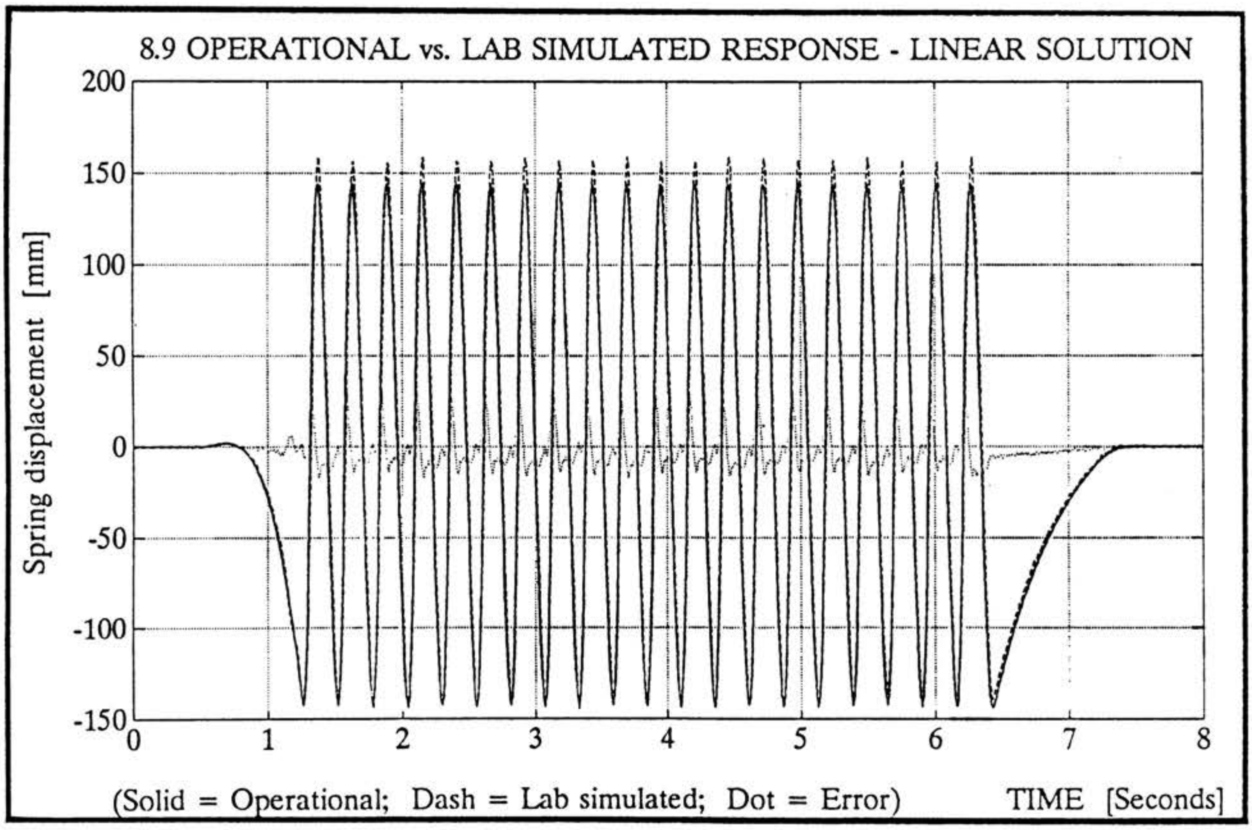


Figure 8.9: Desired operational- and laboratory achieved response - (Continuous operation) - linear solution

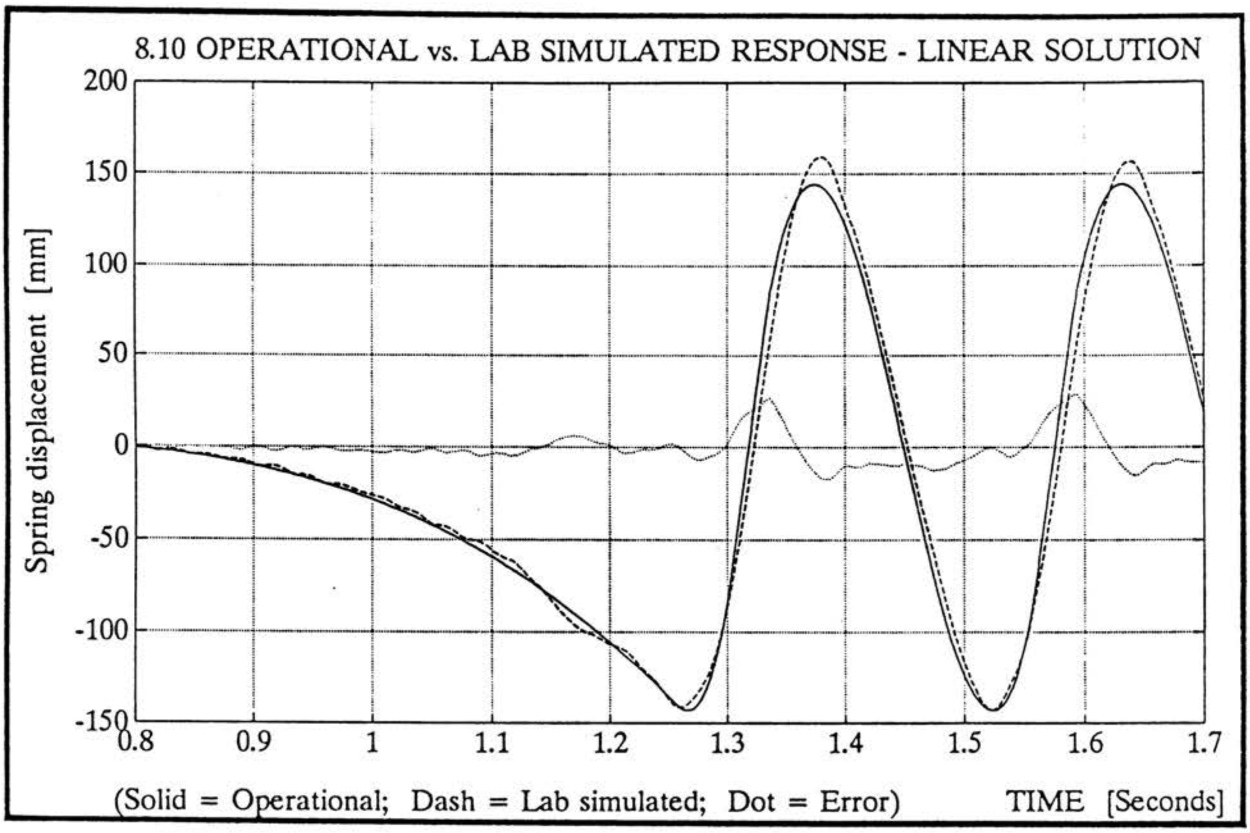


Figure 8.10: Detail extract of figure 8.9

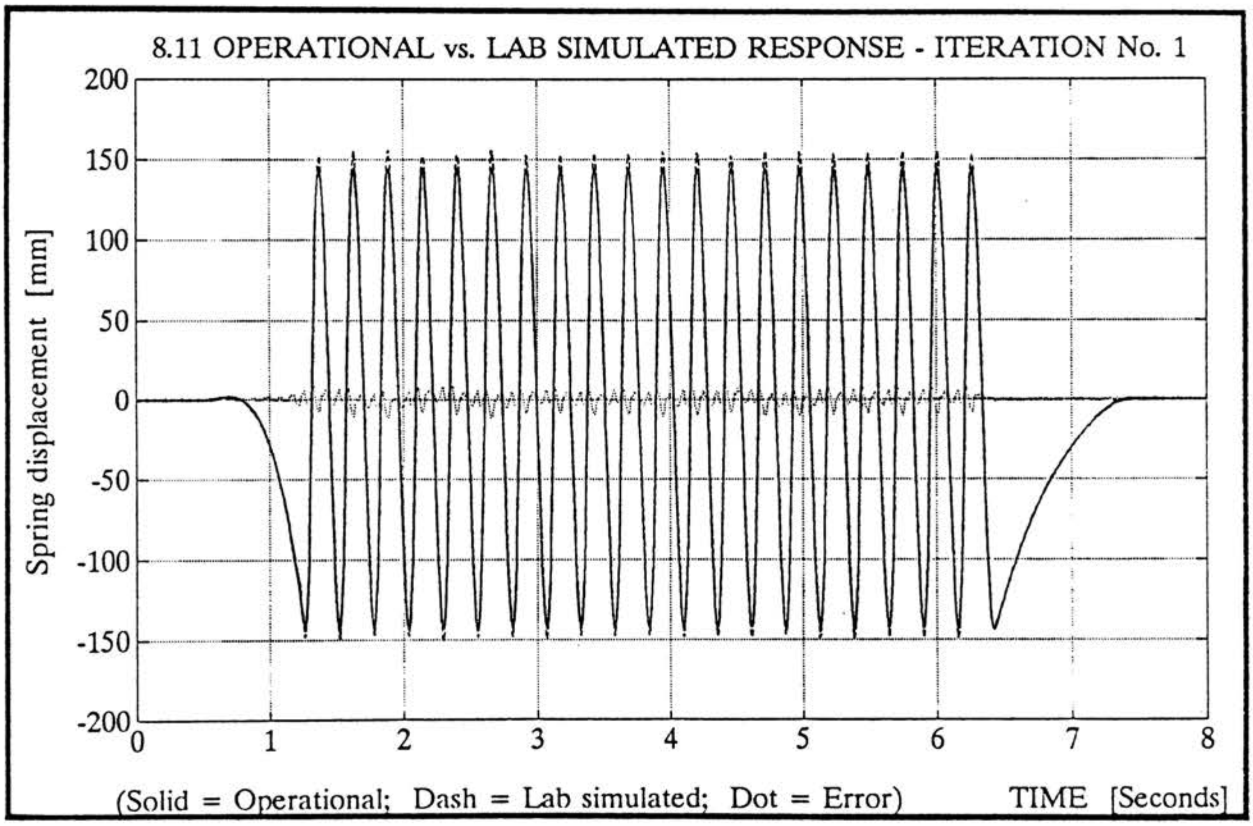


Figure 8.11: Desired operational- and laboratory achieved response - (Continuous - after one iteration

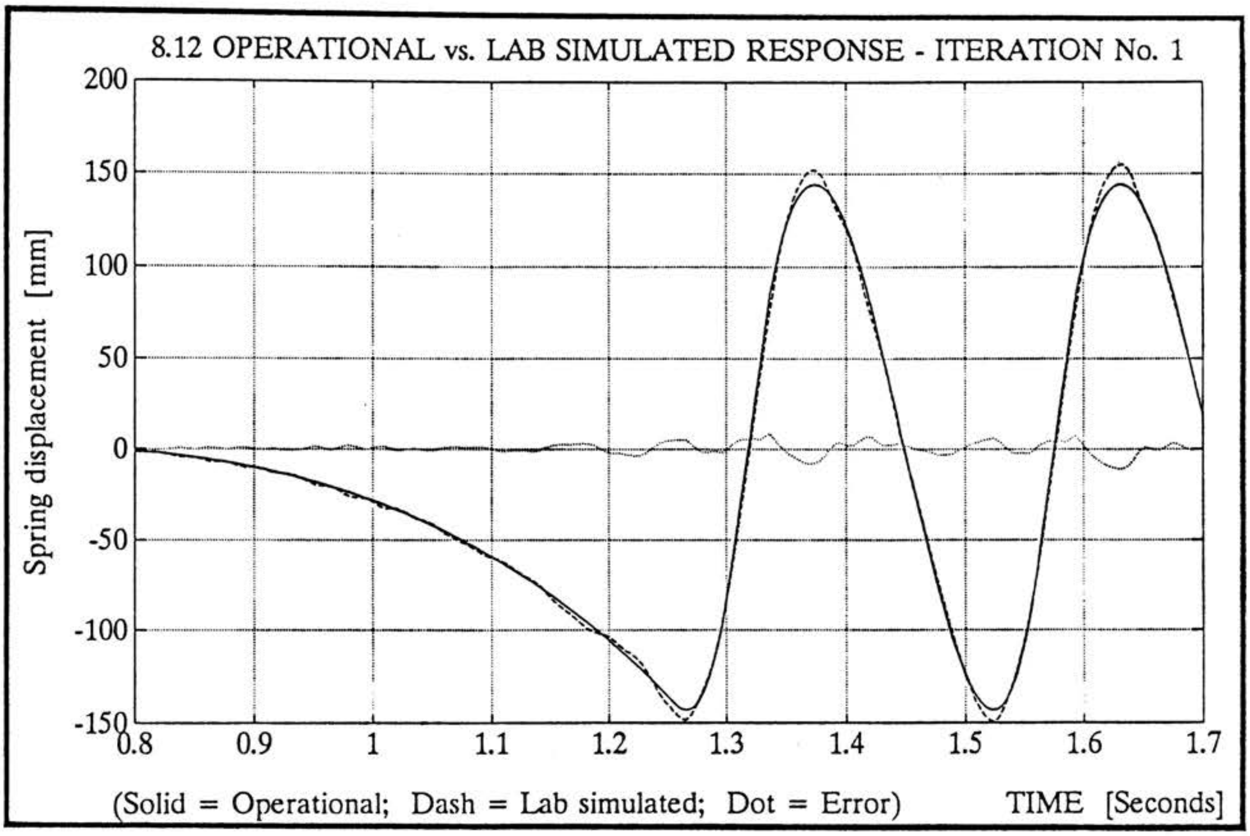


Figure 8.12: Detail extract of figure 8.11

8.3 Case Study 2: Simulation of Operational Responses on a Vehicle Suspension

The local manufacturers of a minibus modified the design of their rear gearbox crossmember to accomodate a new gearbox in the vehicle. Due to a number of fundamental problems, several failures occurred and the Centre for Structural Mechanics was approached by the vehicle manufacturers to solve the problem. As part of the final qualification of a newly designed crossmember, an equivalent distance of 300 000 km was simulated on a laboratory test rig in an endurance fatigue test. Details of this investigation may be found in Raath [1990 c]. Both the frequency and time domain techniques were employed to determine the actuator inputs for the simulation of the operational measured responses. This section describes the time domain application, while section (9.3) gives a comparison with the frequency domain techniques.

8.3.1 Instrumentation of Vehicle

The crossmember serves as reaction point to the two front suspension torsion bars, and furthermore supports the gearbox. The loading of the crossmember thus consists of two very large moments, as well as a vertical load from the gearbox. Figure 8.13 shows the general configuration of loadings on the crossmember. The vehicle components were therefore instrumented to record signals analogous to these three loads. Shear strain gauges were applied to the two torsion bars to record the torsional strain data. For the loads exerted by the gearbox two small loadcells were installed between the gearbox and crossmember mounting interface.

8.3.2 Recording of Operational Data

The vehicle was driven over a number of pre-selected routes while recording the two strain gauged channels and the two loadcells onto an analogue FM tape recorder. These signals served as operational responses, which were subsequently duplicated on the laboratory test rig. For practical reasons, the data from the two loadcells were summed to give the total load onto the crossmember as exerted by the gearbox. Figures 8.14 and 8.15 show two of the recorded operational response sections. These two sections were chosen from a total of twelve possible sections with the specific intention of comparing results with those obtained with the frequency domain technique, which are presented in section 9.3 . It was found that of all twelve data sections, the frequency domain technique gave the most favourable results on data section "A" (figure 8.14). On the other hand, data section "B" (figure 8.15) was found difficult to simulate in the frequency domain.

8.3.3 Laboratory Test Rig

To achieve the correct stiffness on the crossmember mounting interfaces, part of the vehicle chassis was mounted to the laboratory strong floor onto which the crossmember and other relevant suspension components could be installed. The torsional loads acting on the crossmember were transmitted via the torsion bars which in turn used two 40 kN servo-hydraulic actuators loading the wishbones through bell-cranks as shown in figure 8.13. The bell-cranks have a 1:2 ratio which enables a high vertical velocity on the wishbones, as is typically required for the simulation of vertical wheel vibrations. A certain degree of geometric nonlinearity is also introduced into the system by the bell-cranks due to the conversion of linear to rotational motion and vice versa. All three servo-hydraulic actuators were driven in displacement control. The same transducers used for data aquisition purposes were also used as response transducers on the test rig.

8.3.4 Determination of System Model

Test Rig Excitation

An estimation of the spectral content of the road recorded data served as an indication of the required frequency bandwidth as well as dictating the sample frequency of 64 Hz, which was specifically chosen to suit the frequency domain techniques, discussed in section 9.3 . Synthetic excitation signals were derived from a prescribed power spectral density function using an inverse Fourier transform. All three actuators were driven simultaneously while also recording the responses, providing the experimental input-output data required for the model identification.

STAGE OF TEST	PERCENT "FIT"			FIGURE NUMBER
	Chan. 1	Chan. 2	Chan. 3	
Model fit (η_u)	5.0 %	5.7 %	6.1 %	8.16
Linear solution (η_{sim})	2.7 %	4.2 %	7.7 %	8.17
Iteration No. 1 (η_{sim})	0.8 %	4.3 %	3.6 %	8.18

Table 8.3: Model fit η_u and simulation results η_{sim} for data section "A"

Identification of Model

An extract of 6 seconds of data, (384 data points) was used to estimate the model orders for the three channels. A direct reversed inverse model was attempted and the orders for the three channels were quite easily found to be a 9th order for each torsion bar channel and a 14th order for the gearbox loading actuator. The inverse model was found to be successful and the final model was identified using 32 seconds of data (2048 data points) giving a total of 32 states for the final state space parametric model. Figure 8.16 shows an extract of the true actuator inputs and the model predicted inputs, showing an acceptable model fit for the three channels.

8.3.5 Results for Data Section "A"

The desired operational responses from figure 8.14 were input to the inverse parametric state space model to give the linear solution to the actuator drive signals. The test rig was excited with these signals, and figure 8.17 shows the laboratory achieved responses plotted with the desired operational measured responses. Also shown in the figure is the error in responses.

Using the response error signals, the drive signal corrections could be found from the inverse parametric model to finally achieve the updated drive signals. Applying these updated drive signals to the test rig resulted in an improved response, as shown in figure 8.18.

The model fit and simulation accuracy for data section "A" is shown in table 8.3.

STAGE OF TEST	PERCENT "FIT"			FIGURE NUMBER
	Chan. 1	Chan. 2	Chan. 3	
Linear solution (η_{sim})	3.5 %	5.2 %	20.2 %	8.19
Iteration No. 1 (η_{sim})	2.2 %	2.0 %	18.3 %	8.20

Table 8.4: Simulation results η_{sim} for data section "B"

8.3.6 Results for Data Section "B"

The same procedure was also applied to data section "B" (figure 8.15), giving the results depicted in figures 8.19 and 8.20, together with table 8.4.

8.3.7 Conclusion

The simulation of operational responses on a vehicle suspension for the purpose of a durability test was accurately achieved using the time domain based testing system. As mentioned before, the same test was also repeated using the frequency domain techniques, and is described in section (9.3), which essentially concentrates on the specific advantages of the time domain compared to the frequency domain in a practical application.

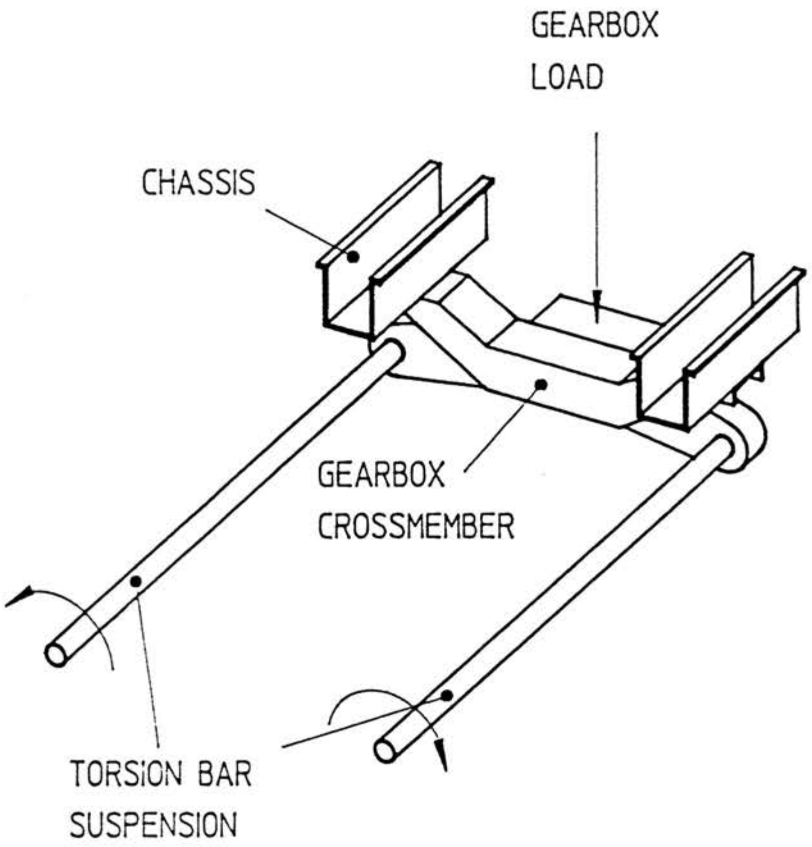


Figure 8.13: General configuration of crossmember loadings

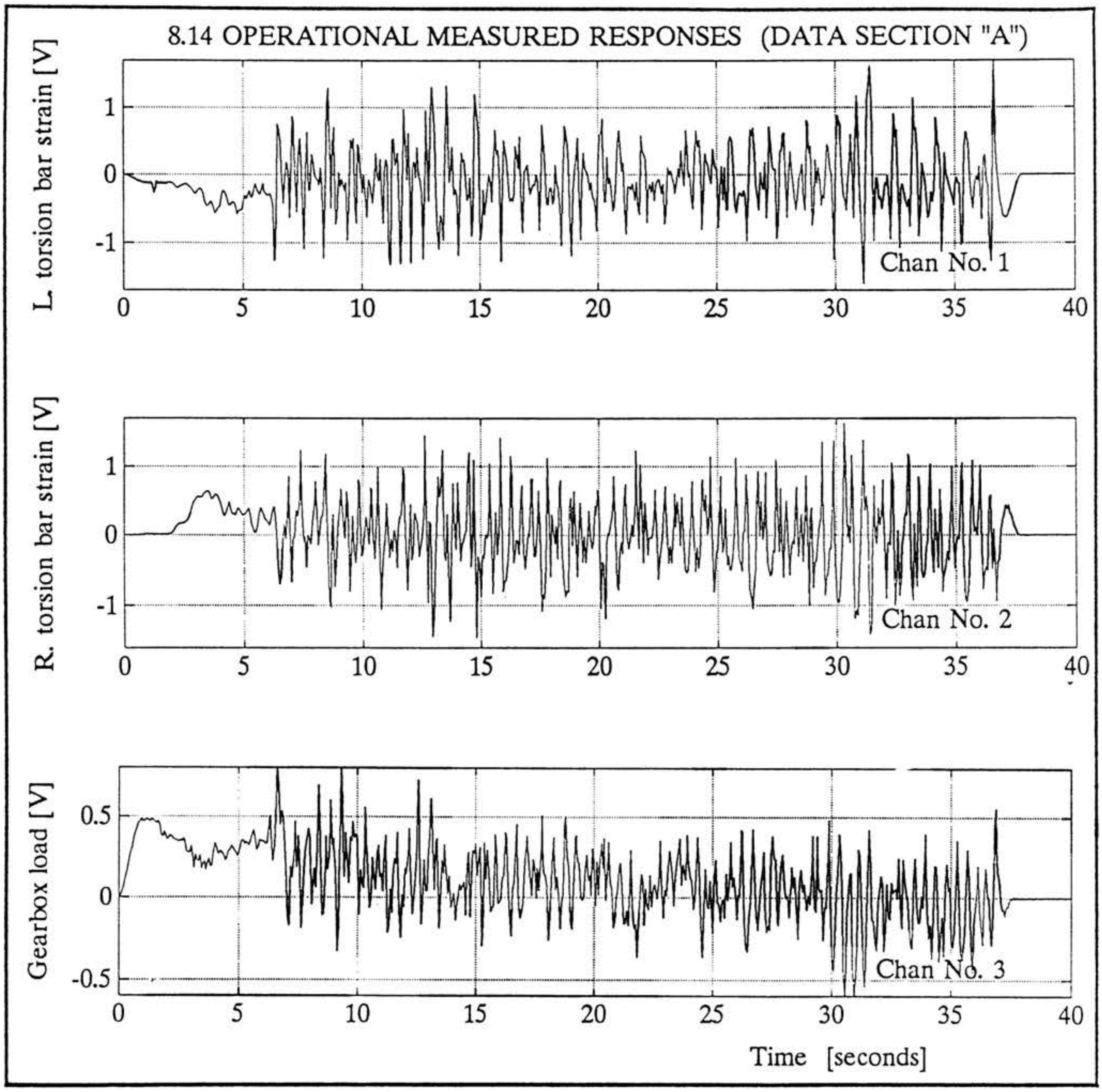


Figure 8.14: Operational measured responses (Data section "A")

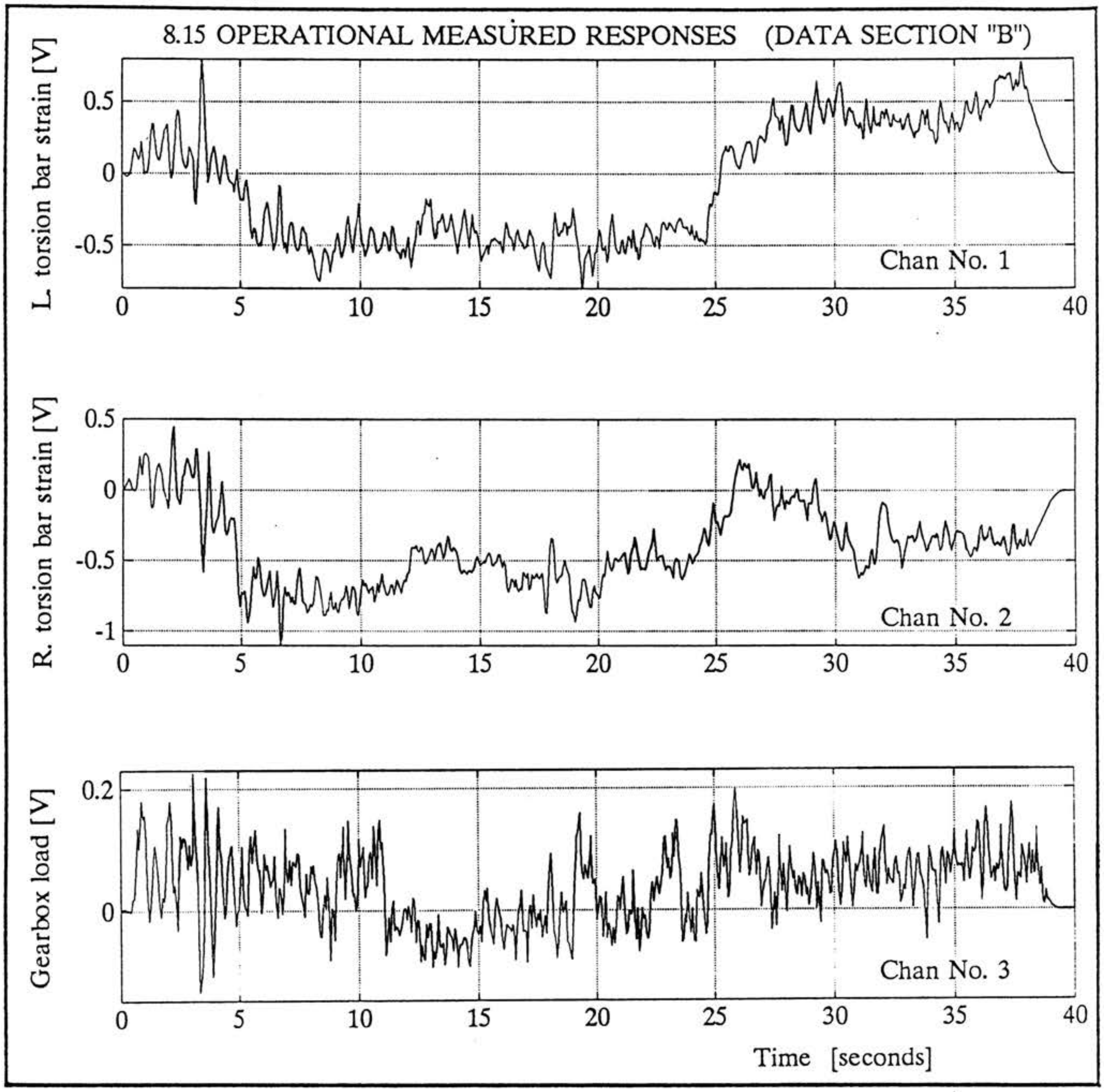


Figure 8.15: Operational measured responses (Data section "B")

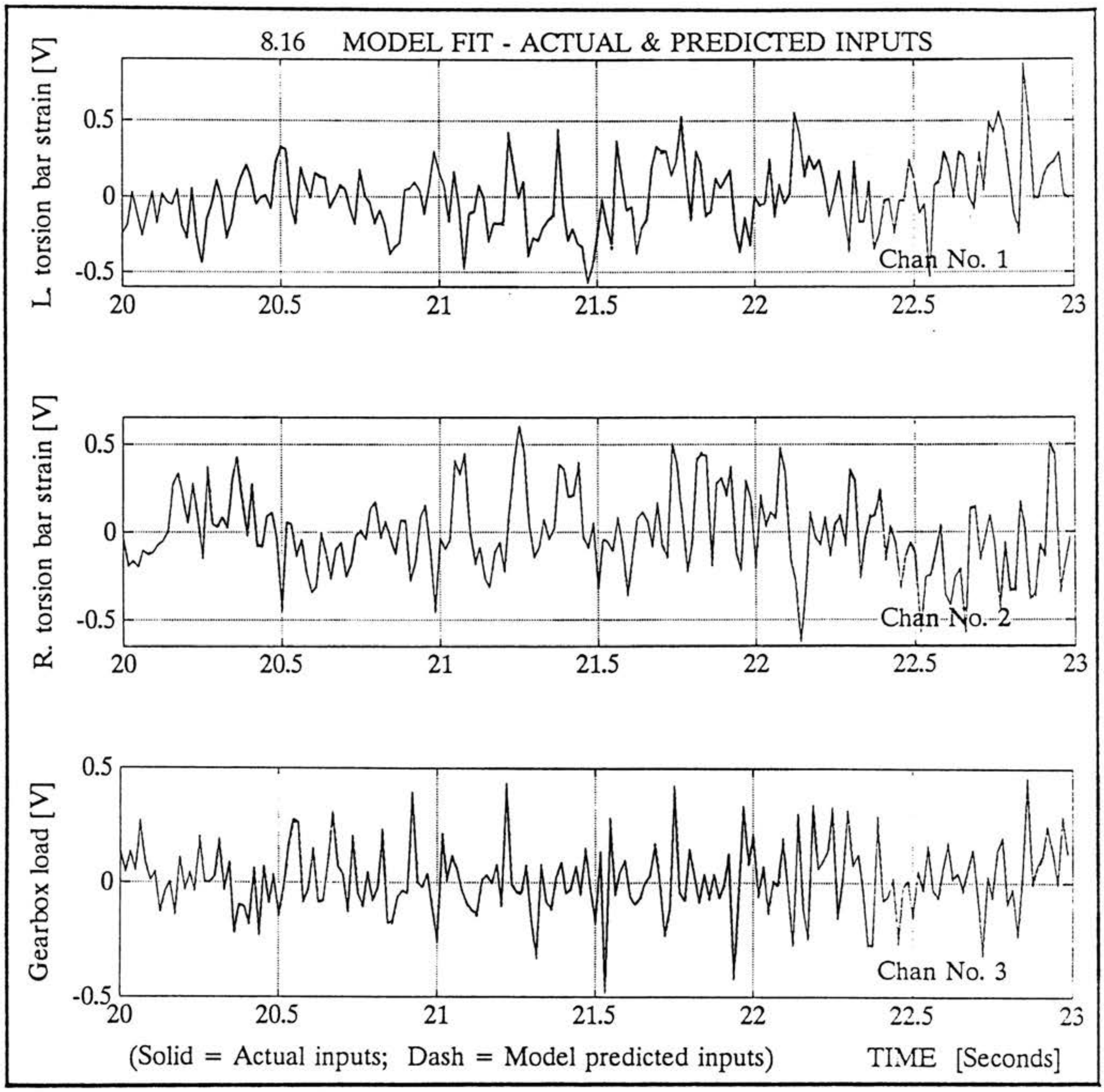


Figure 8.16: Model fit - Actual and model predicted inputs

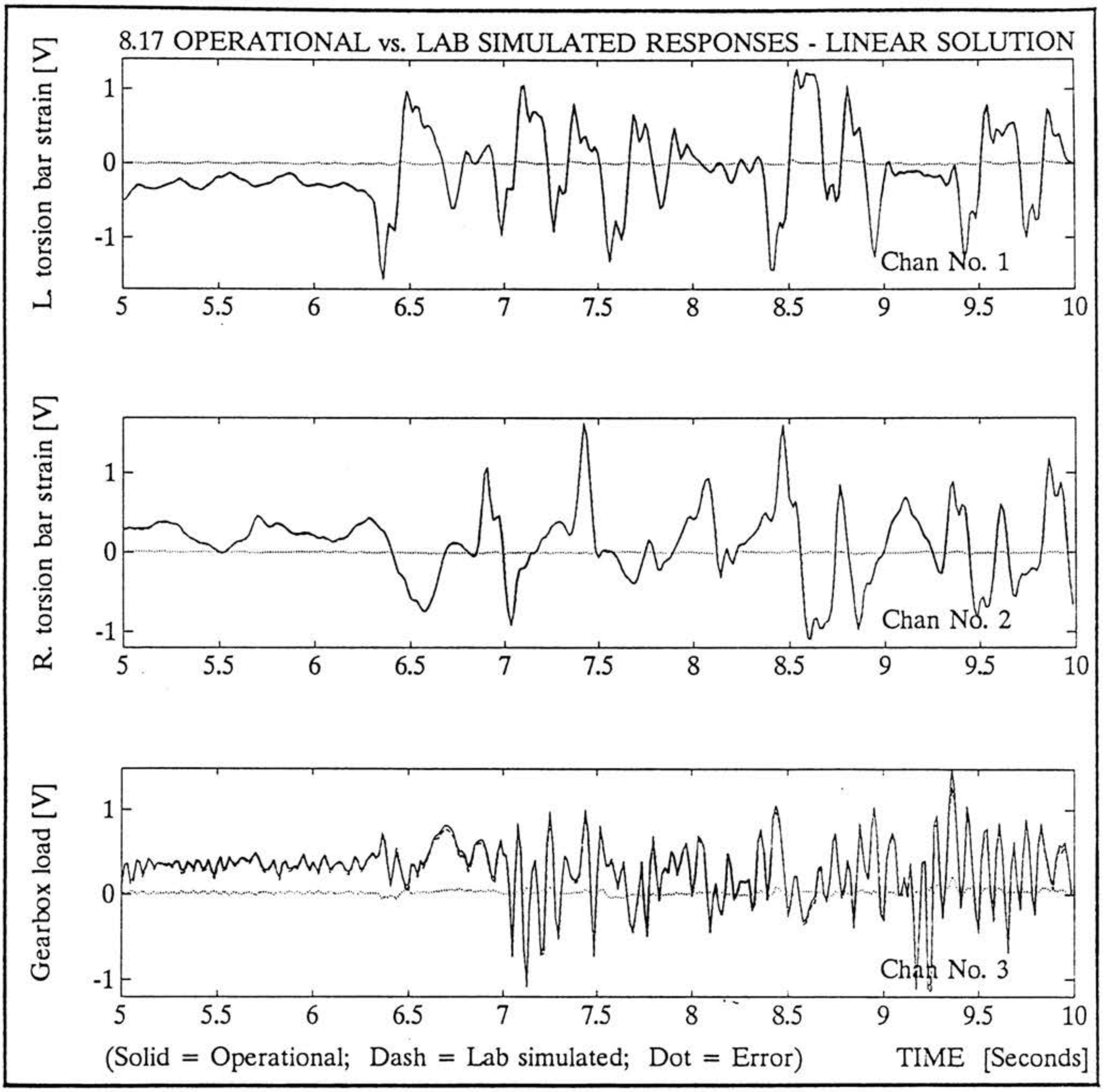


Figure 8.17: Desired operational- and laboratory achieved responses (Data section "A") - linear solution

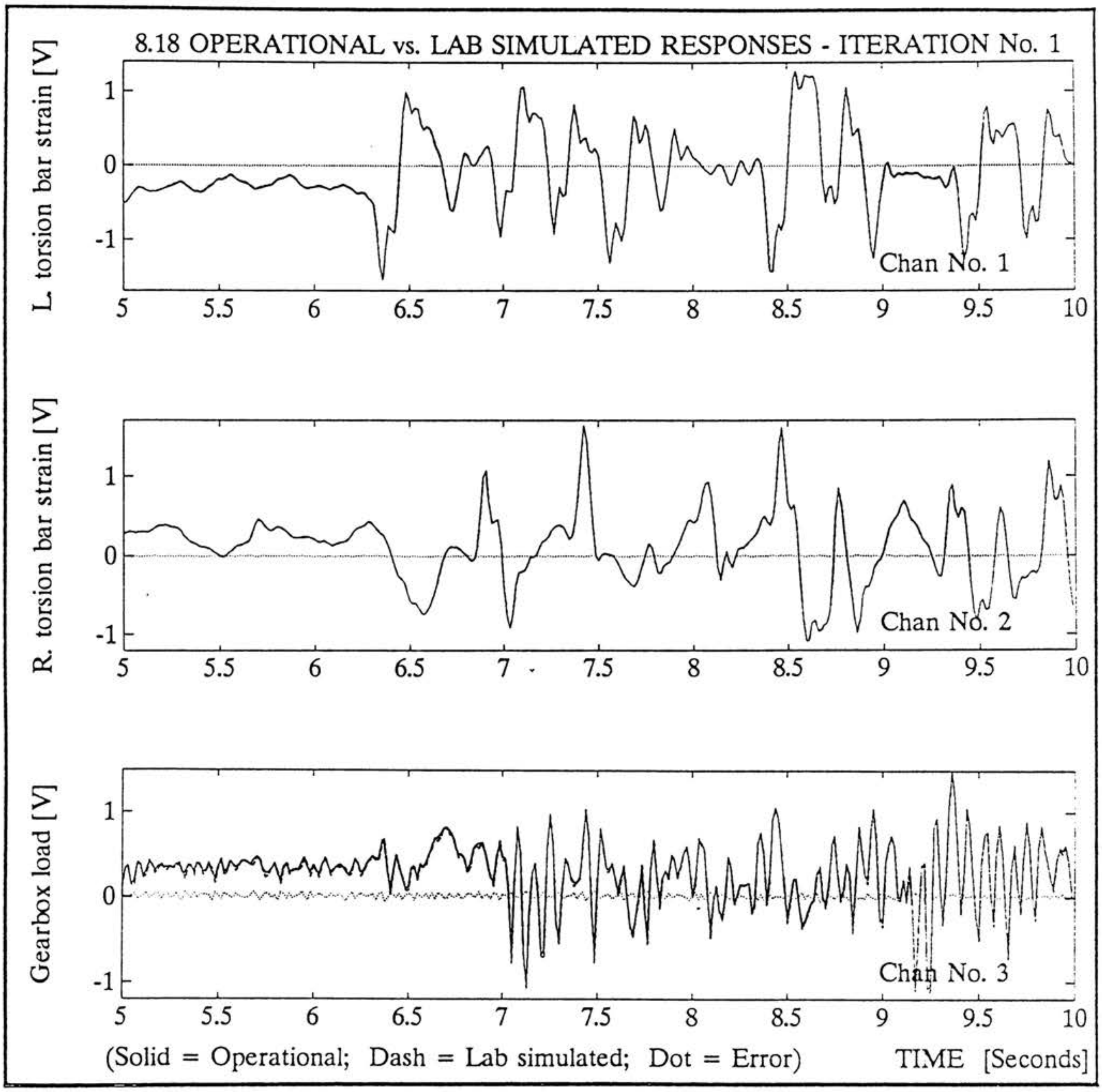


Figure 8.18: Desired operational- and laboratory achieved responses (Data section "A") - after one iteration

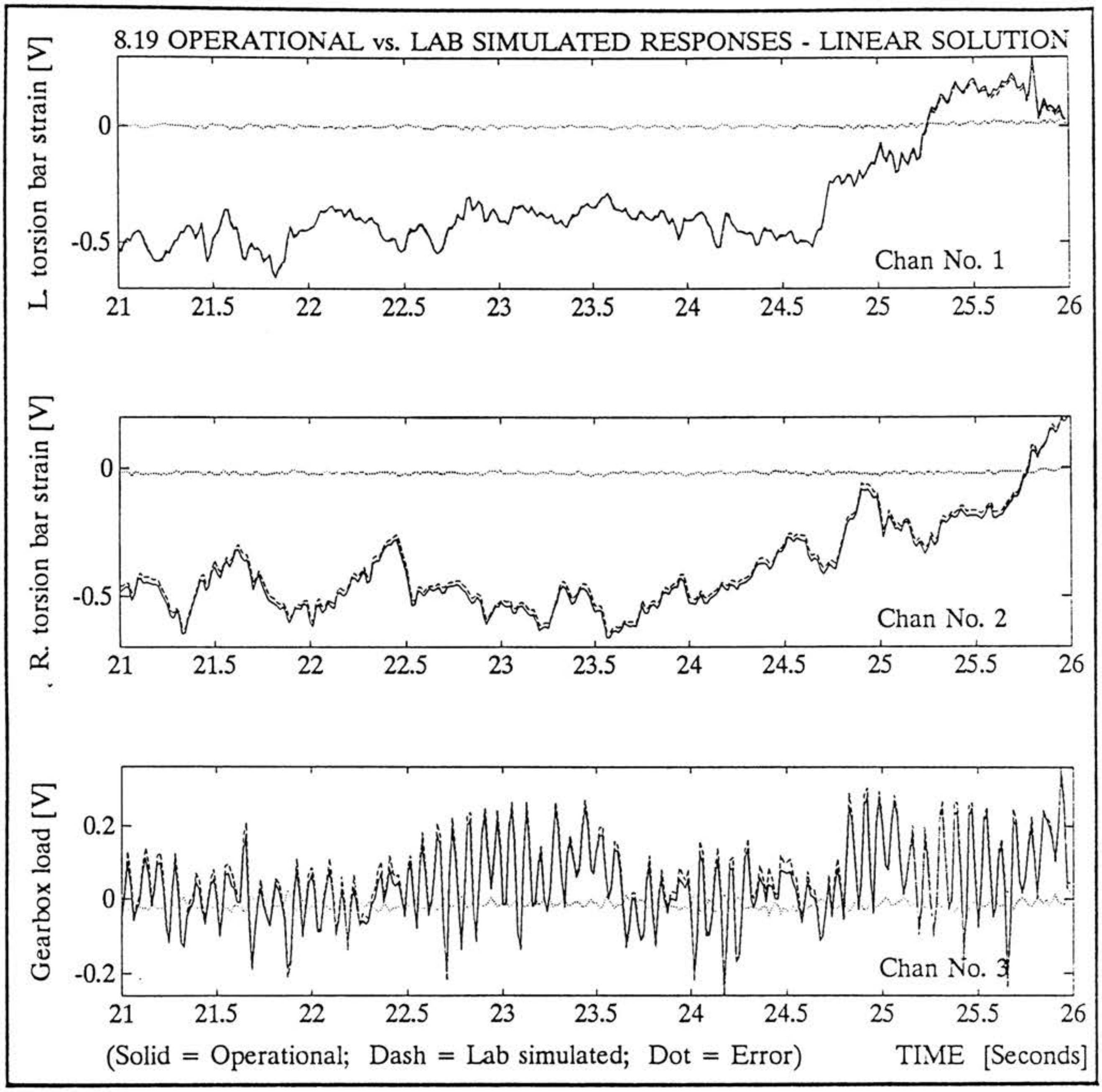


Figure 8.19: Desired operational- and laboratory achieved responses (Data section "B") - linear solution

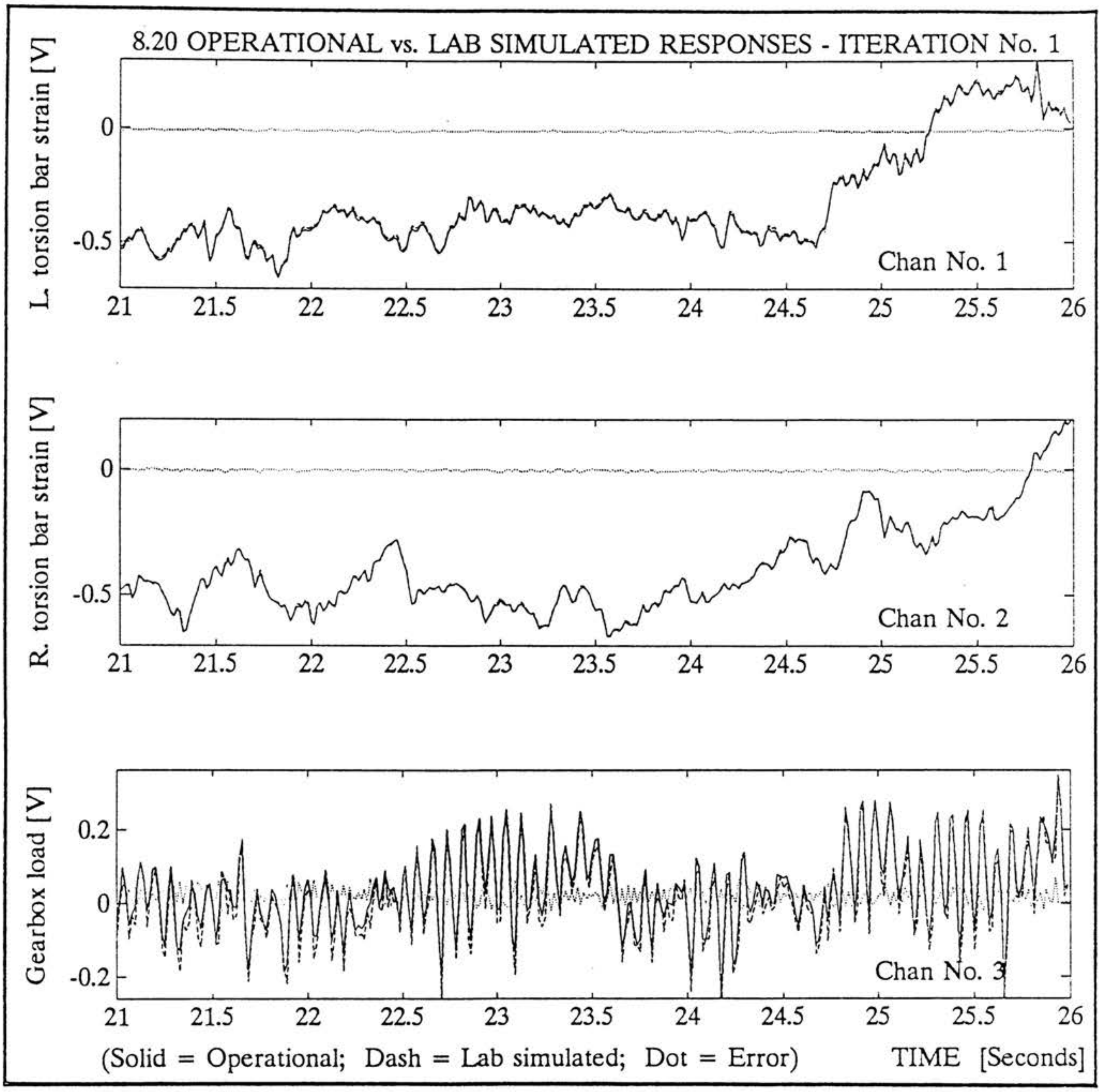


Figure 8.20: Desired operational- and laboratory achieved responses (Data section "B") - after one iteration

8.4 Case Study 3: Four Channel Simulation of Operational Responses on a Passenger Vehicle

The structural integrity of the rear longitudinal beams of a passenger vehicle was established through a durability test on a laboratory test rig. The precise details of this test may be found in Raath [1991 (b)], but are considered superfluous for the purpose of this discussion and only the essential results are presented here.

8.4.1 Test Configuration

Photographs of the test configuration are shown in Appendix E. The vehicle body was mounted on a test stand and loaded by four servo-hydraulic actuators, two of which provided the rear wheel displacements, and two applied vertical loading inputs to the rear longitudinal beams, simulating the boot load.

The relative displacement of the semi-trailing arms was recorded via two displacement transducers, while two strain gauge channels provided an indication of the bending strains on the longitudinal beams. These four transducers acted as response transducers both for the recording of operational data during road recordings, as well as on the laboratory test rig. A significant amount of cross-coupling was found in as much as the strain gauge readings are influenced by contributions from the wheel loading actuators as well as those utilized to simulate the boot load.

8.4.2 Dynamic Model Identification

After exciting all actuators simultaneously a direct reversed inverse model was identified on the experimental input-output data, giving a sixth order model on each of the wheel displacements, and a ninth order on each of the longitudinal strain gauge channels. This produced the final state space model with 30 states.

STAGE OF TEST	PERCENT "FIT"			
	Chan. 1	Chan. 2	Chan. 3	Chan. 4
Iteration No. 3 (η_{sim})	0.8%	0.8%	6.9%	3.6%

Table 8.5: Simulation results η_{sim} for passanger vehicle (figure 8.21)

8.4.3 Simulation Results

Three iterations were required to achieve the desired results, and an extract of the road recorded and laboratory simulated responses is shown in figure 8.21, with the simulation accuracy parameter of the final drive signals shown in table 8.5.

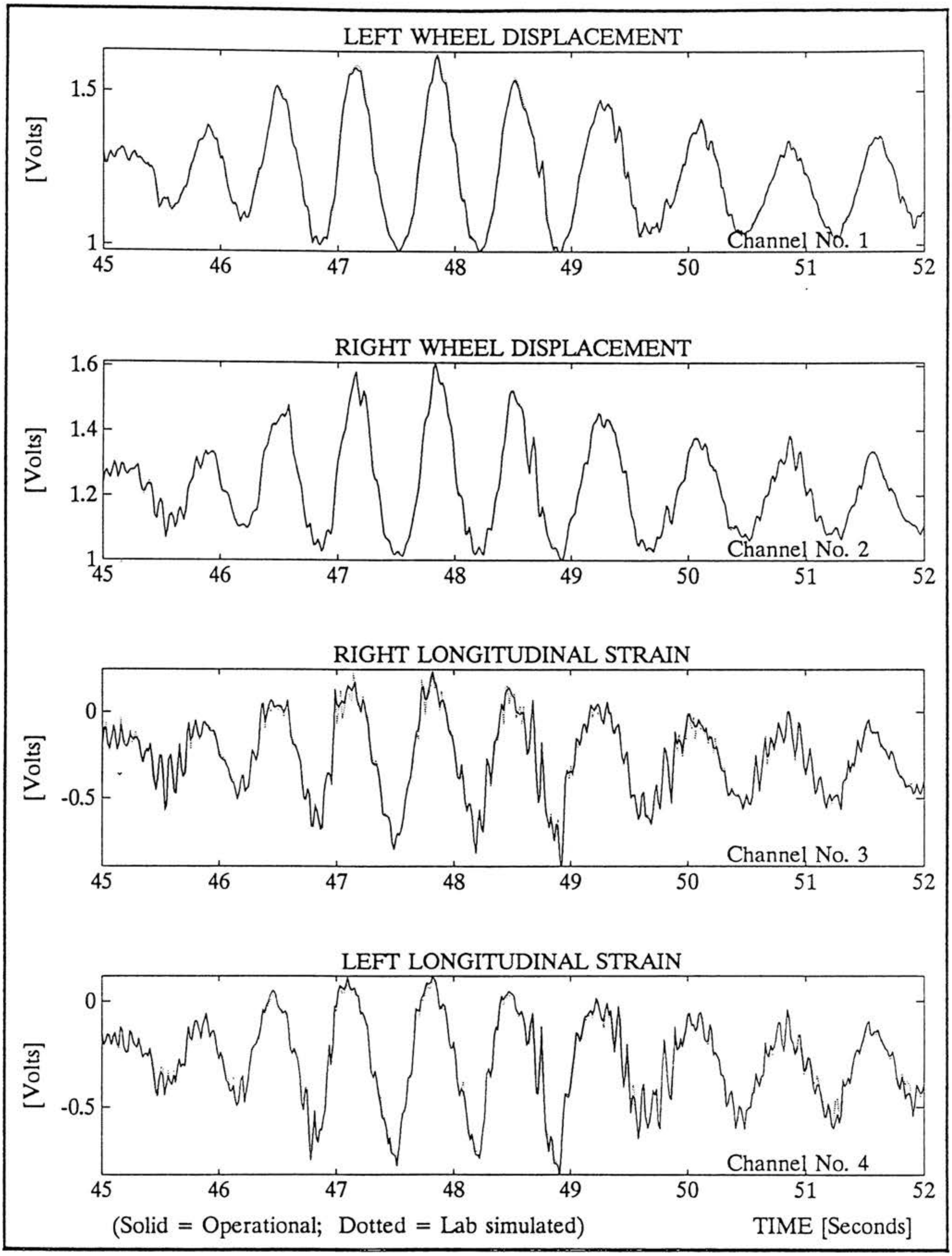


Figure 8.21: Road recorded- and laboratory simulated responses on passenger vehicle

8.5 Case Study 4: Five Channel Simulation of Operational Responses on a Vehicle Chassis

The final case study involved the application to the simulation of operational responses on the front section of a pick-up truck chassis using five simultaneous servo-hydraulic actuators. As with the previous case study, only the essential results are presented, and details of this test may be found from Wannenburg and Immelman [1991].

8.5.1 Test Configuration

The rear end of the chassis was mounted to a test stand while applying the vertical displacements to the front wheel spindles, using two actuators working through bell-cranks. Two additional actuators simulated the vertical loads exerted by the vehicle cab through the front cab mountings on the chassis. Finally the fifth actuator applied the loads which are transferred by the engine onto the chassis via the rubber engine mountings. Due to severe resonance problems, the test rig was later modified by moving the chassis mounting to the front. Figures E.7 through E.10 show the original test rig prior to the modifications.

Strain gauges were utilized to record the shear strains on the two front torsion bars, while another set of strain gauges provided the bending strains on the longitudinal chassis beams to the rear of the two front wheels. The engine load was recorded using strain gauges on one of the engine mounting brackets. These five strain gauge channels were used as response transducers during both the operational road recordings on the vehicle, as well as the subsequent simulation of the operational data on the chassis in the laboratory.

8.5.2 Test Excecution and Results

Upon excitation of the test rig a direct reversed inverse model was identified giving a total of 34 states for the final reversed inverse state space model. A number of different data sections were simulated in the endurance test,

STAGE OF TEST	PERCENT "FIT"				
	Chan. 1	Chan. 2	Chan. 3	Chan. 4	Chan. 5
Iteration No. 3 (η_{sim})	1.8%	0.8%	2.1%	3.1%	8.6%

Table 8.6: Simulation results η_{sim} for vehicle chassis (figure 8.22)

and figure 8.22 shows a typical extract of one of these sections, while the final simulation results are given in table 8.6.

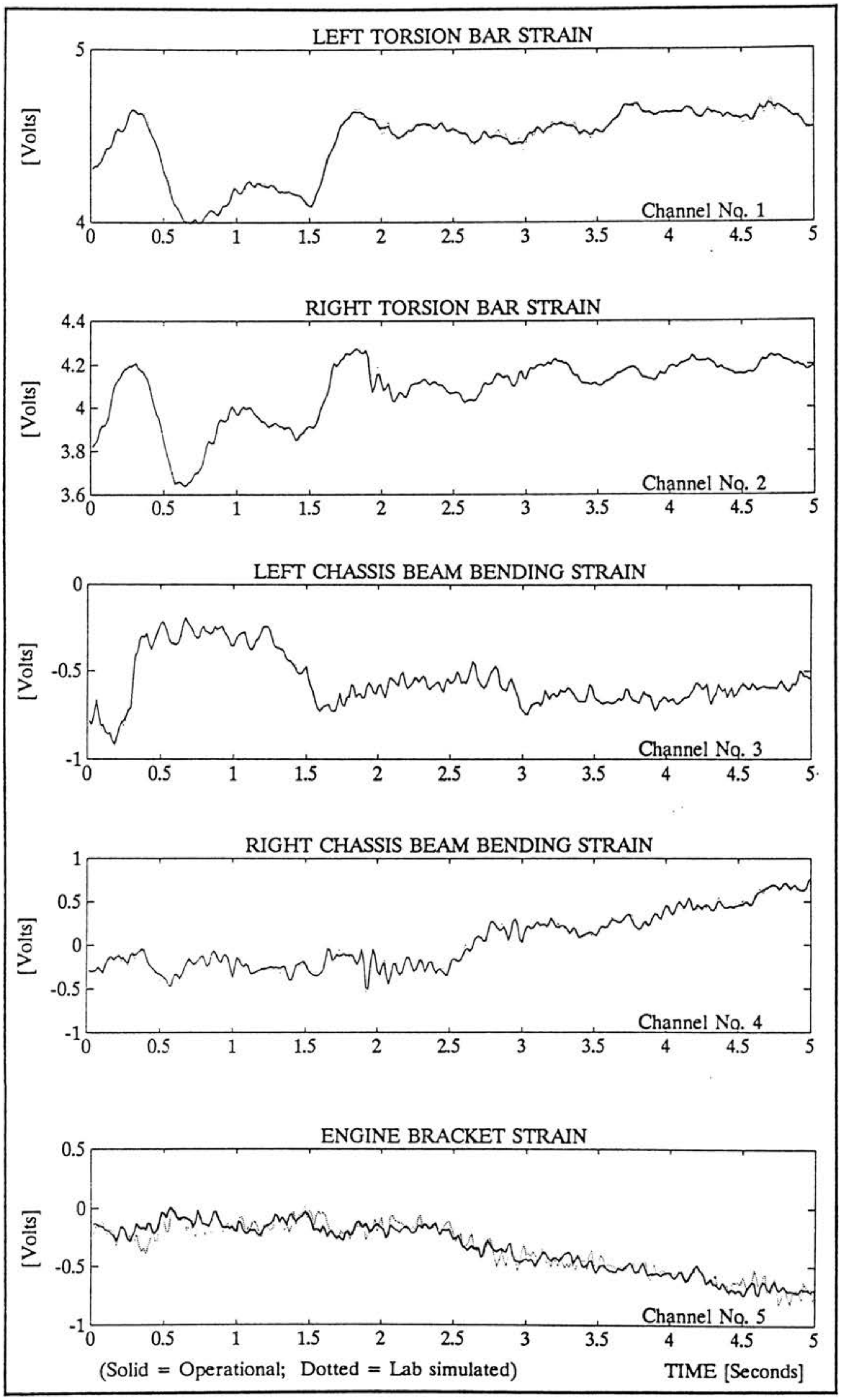


Figure 8.22: Road recorded- and laboratory simulated responses on vehicle chassis

8.6 Summary

Dynamic Model Identification

Estimation of the model order and delays for each channel was done by simply fitting a number of models and displaying the sum of squared residuals upon which the order of the system could be determined.

In all four case studies it was found most convenient to fit a direct reversed inverse model in each case, which has the further advantage that the delays need not be estimated. Normal forward models were also attempted, and found to exhibit unstable eigenvalues both for the forward inverse and reversed inverse conversions. The linear quadratic optimal servo-controller was also applied in the case of the second case study (section 8.3), and performed exceptionally well, but was found to be very time consuming and could probably prove impractical in a production testing environment. Should no stable inverse be found through any of the other methods, the linear quadratic optimal servo-controller can as a last resort always supply the system inputs from the the given operational measured responses.

Simulation Accuracy

The accuracy of simulation was found to be good in all case studies. Even in the case of extremely short impulses, as was the case with the high speed spring tester, the time domain technique seems to perform exceptionally well. As was mentioned in section 8.1, the deviations between the operational measured- and laboratory simulated responses may essentially be ascribed to the mechanical inability of the test rig and servo-hydraulic actuators to achieve a perfect simulation.

Chapter 9

Applications: Comparison with Frequency Domain Techniques

9.1 Introduction

Referring to section 1.3, several advantages of the time domain compared with a frequency domain based system were postulated, such as the sample frequency, the amount of data required, frequency resolution restrictions, and the consequential general accuracy of the two methods. Several of these aspects were addressed by comparing results using the developed time domain technique with those obtained through the frequency domain technique. The first two case studies presented in the previous chapter, namely the high speed spring tester and the simulation of operational responses on a vehicle suspension, were utilized for this purpose, by also applying the frequency domain techniques to these case studies.

9.2 Frequency Domain Application: Single Servo-hydraulic Actuator High Speed Spring Tester

The first case study which was discussed in section 8.2, was also attempted with the frequency domain based techniques. Identical conditions were utilized to arrive at an accurate comparison.

9.2.1 Identification of Transfer Function

Using the same excitation signal as shown in figure 8.4 the system transfer function was identified, giving an acceptable coherence. It is usually required to utilize significantly more data in the frequency domain; however a direct comparison with the results from section 8.2 was required, necessitating the short identification data process.

9.2.2 Results for Single Cycle Operation

It was firstly attempted to simulate the single cycle operational response of figure 8.2 . Since the frequency domain technique requires a minimum of two FFT's, (2048 points or 8 seconds) for the desired operational response, a section of zeros was inserted in the data before and after the single cycle of figure 8.2 . Figure 9.1 shows the desired operational response together with the laboratory achieved response for the linear solution to the system input data. Also shown is the error in response. One iteration was attempted, the results of which are shown in figure 9.2 .

9.2.3 Results for Continuous Operation

The continuous operation desired response of figure 8.2 was also attempted. Again only one iteration was applied, whereafter the test was aborted. Results of this test are shown in figures 9.3 and 9.4 .

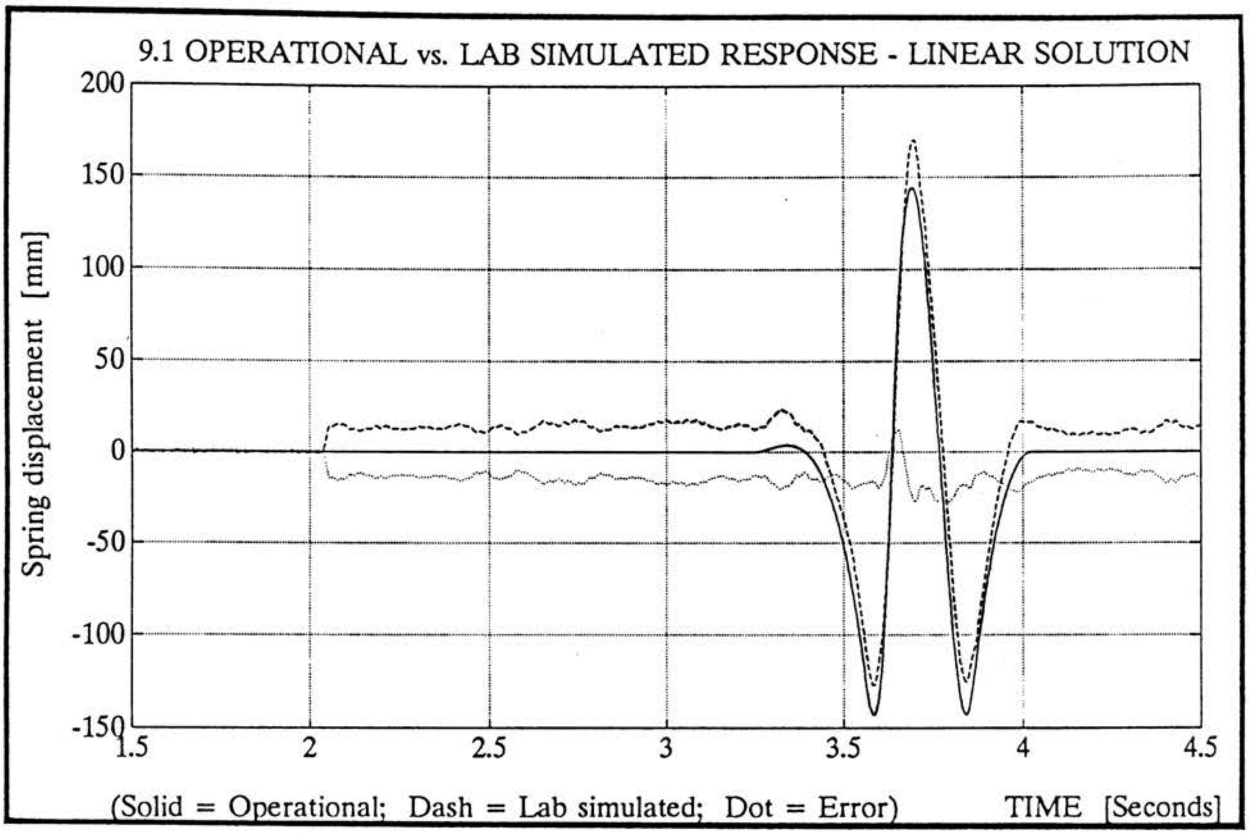


Figure 9.1: Desired operational- and laboratory achieved response - linear solution - Using frequency domain technique

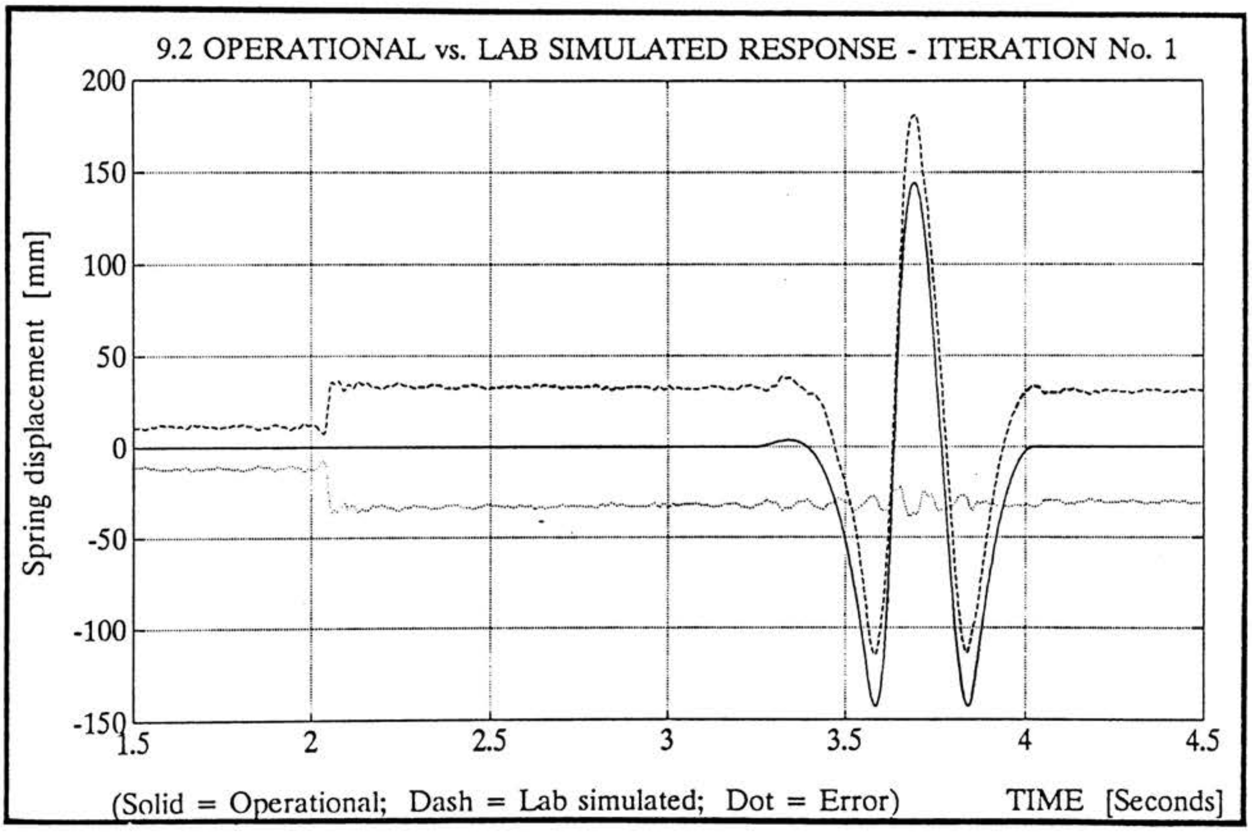


Figure 9.2: Desired operational- and laboratory achieved response - after one iteration - Using frequency domain technique

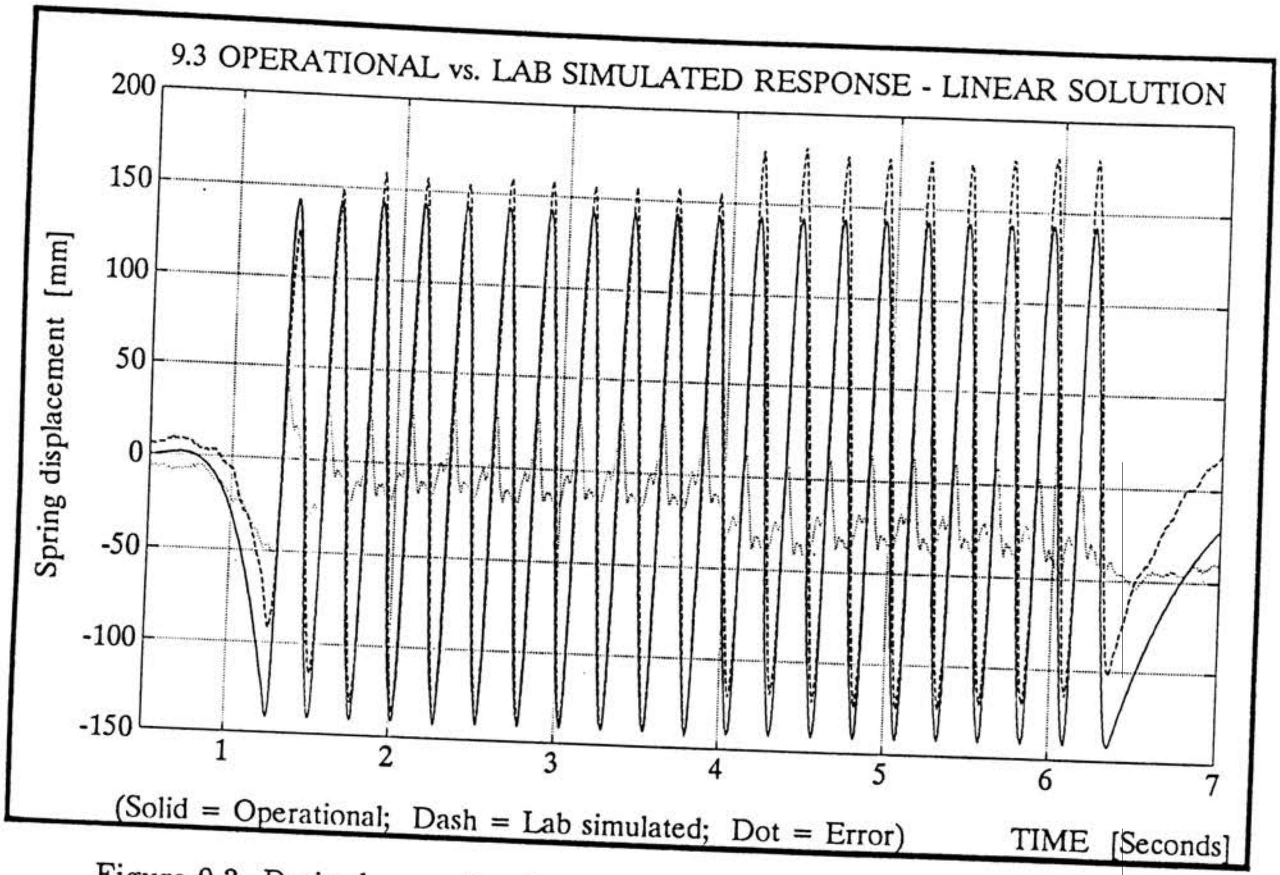


Figure 9.3: Desired operational- and laboratory achieved response - linear solution - Using frequency domain technique

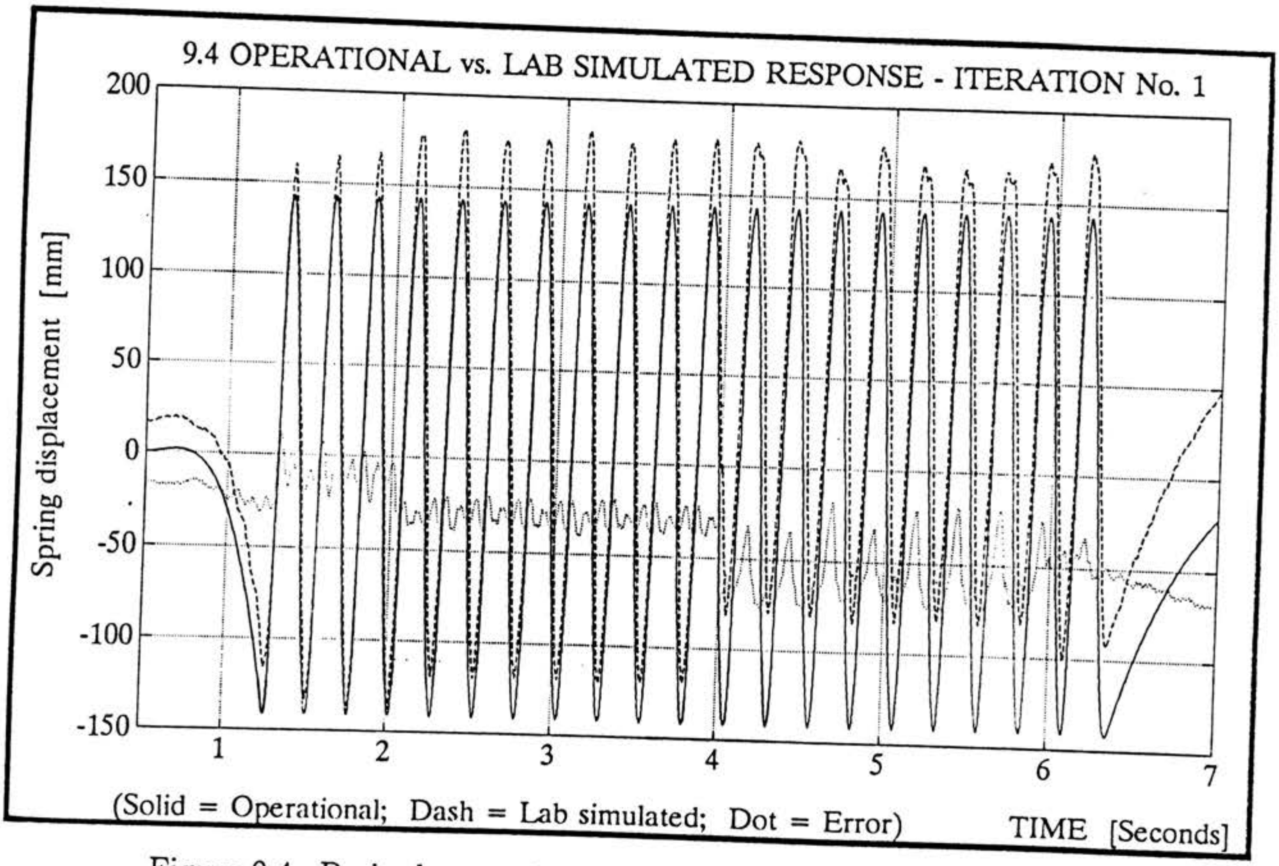


Figure 9.4: Desired operational- and laboratory achieved response - after one iteration - Using frequency domain technique

9.2.4 Discussion of Results

In both the single cycle and continuous operation modes the responses show significant steps in the data. This may be explained by the FFT processing which is affected by the frequency resolution. Using a 1024 point FFT with a 256 Hz sample frequency results in a spectral resolution of 0.25 Hz, which causes leakage of specific frequencies to neighboring spectral lines. Due to the requirement of a high sample frequency to ensure an acceptable time resolution, the sample frequency of 256 Hz could not be reduced to increase the spectral line resolution. It would seem that these results also indicate that the frequency domain techniques are unsuited to short data sections. In particular, applications of simulating systems where impulses are experienced, makes the frequency domain techniques difficult to apply successfully.

9.3 Frequency Domain Application: Simulation of Operational Responses on a Vehicle Suspension

9.3.1 General

The frequency domain techniques were also applied to the case study presented in section 8.3 . All testing conditions remained as before, including the test rig, sample frequency of $64Hz$, excitation signal PSD, as well as the identification signal length of 32 seconds.

As mentioned in section 8.3, two specific sections of data, from a possible twelve data sections, were chosen for reasons of comparing results of the time- and frequency domain techniques.

9.3.2 Identification of Transfer Function

The identification signal length of 32 seconds was applied as in section 8.3, except that the frequency domain technique identifies one channel at a time, therefore using a total of 96 seconds of data for the three channels. The coherence functions on the diagonal elements of the system transfer function were found to be very good, and from the system transfer function only a small amount of cross coupling was found between the gearbox force response and the two torsion bar input loads, as was expected. Following the inversion of the system transfer function, the linear solution inputs to the drive signals could be determined.

9.3.3 Test Results and Discussion for Data Section "A"

This data section was chosen specifically due to superior performance in the frequency domain application when compared with all the other possible data sections.

The response to the linear solution system inputs is shown in figure 9.5, together with the desired response signals and the response error. The same results are shown in figure 9.6 after one iteration. The simulation accuracy results are given in table 9.1.

STAGE OF TEST	PERCENT "FIT"			FIGURE NUMBER
	Chan. 1	Chan. 2	Chan. 3	
Linear solution (η_{sim})	37.6%	48.3%	48.1%	9.5
Iteration No. 1 (η_{sim})	9.3%	35.4%	18.6 %	9.6

Table 9.1: Simulation results η_{sim} for data section "A" - Using frequency domain technique

Apart from a small offset on the second channel a close inspection of figure 9.6 indicates a good simulation. Results for the time domain application for the same section of data were given in figures 8.17 and 8.18. Tables 8.3 and 9.1 also compare the simulation accuracies η_{sim} for the time- and frequency domain methods. Both techniques show an accurate simulation, with the time domain results marginally superior to the frequency domain results.

9.3.4 Test Results and Discussion for Data Section "B"

In contrast to the data section discussed above, section "B" was chosen specifically due to very poor performance in the frequency domain application when compared with all the other possible data sections.

The linear solution responses are shown in figure 9.7, together with the desired response signals and the response error. The same results after one iteration are shown in figure 9.8. A global view of the same data is presented in figure 9.9, which shows steps in the laboratory simulated responses around 16 seconds and around 24 seconds. A close up view of the data around 24 seconds is shown in figure 9.10 . In this case the frequency domain determined responses compare poorly with the desired responses.

The reason for this poor response is the very low frequency component present in the data, which typically stems from data obtained when driving the vehicle over a culvert. An estimate of this low frequency is approximately 0.06 Hz, which cannot be well described by the transfer function exhibiting a spectral resolution of 0.125 Hz . The result of this low frequency component causes leakage to the first spectral line of the transfer function, which in turn results in offset problems and in fact corrupts the data.

A comparison with figure 8.20, shows that the time domain technique is not affected by such low frequency trends.

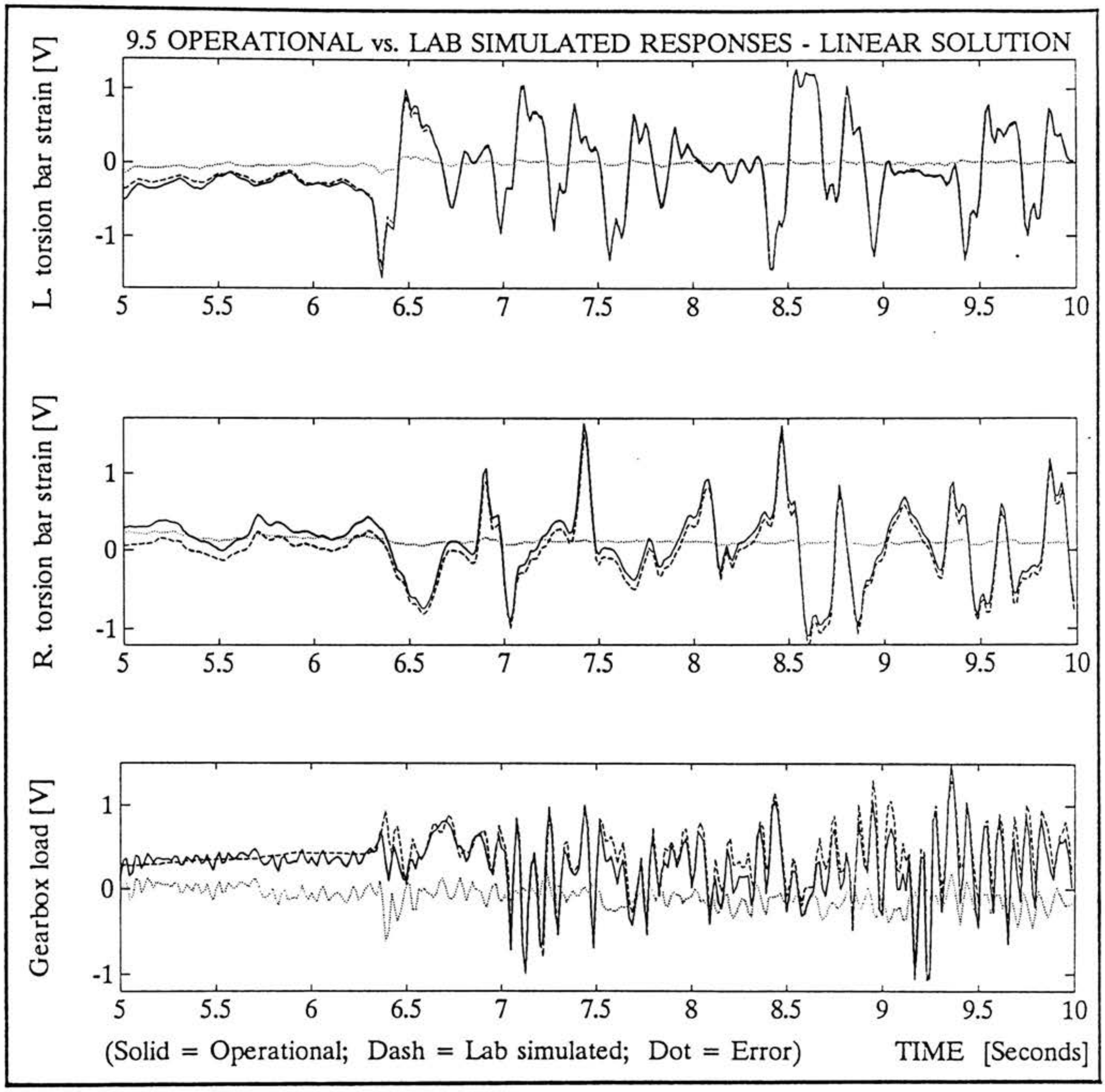


Figure 9.5: Results for data section "A" - Using frequency domain technique - Linear solution

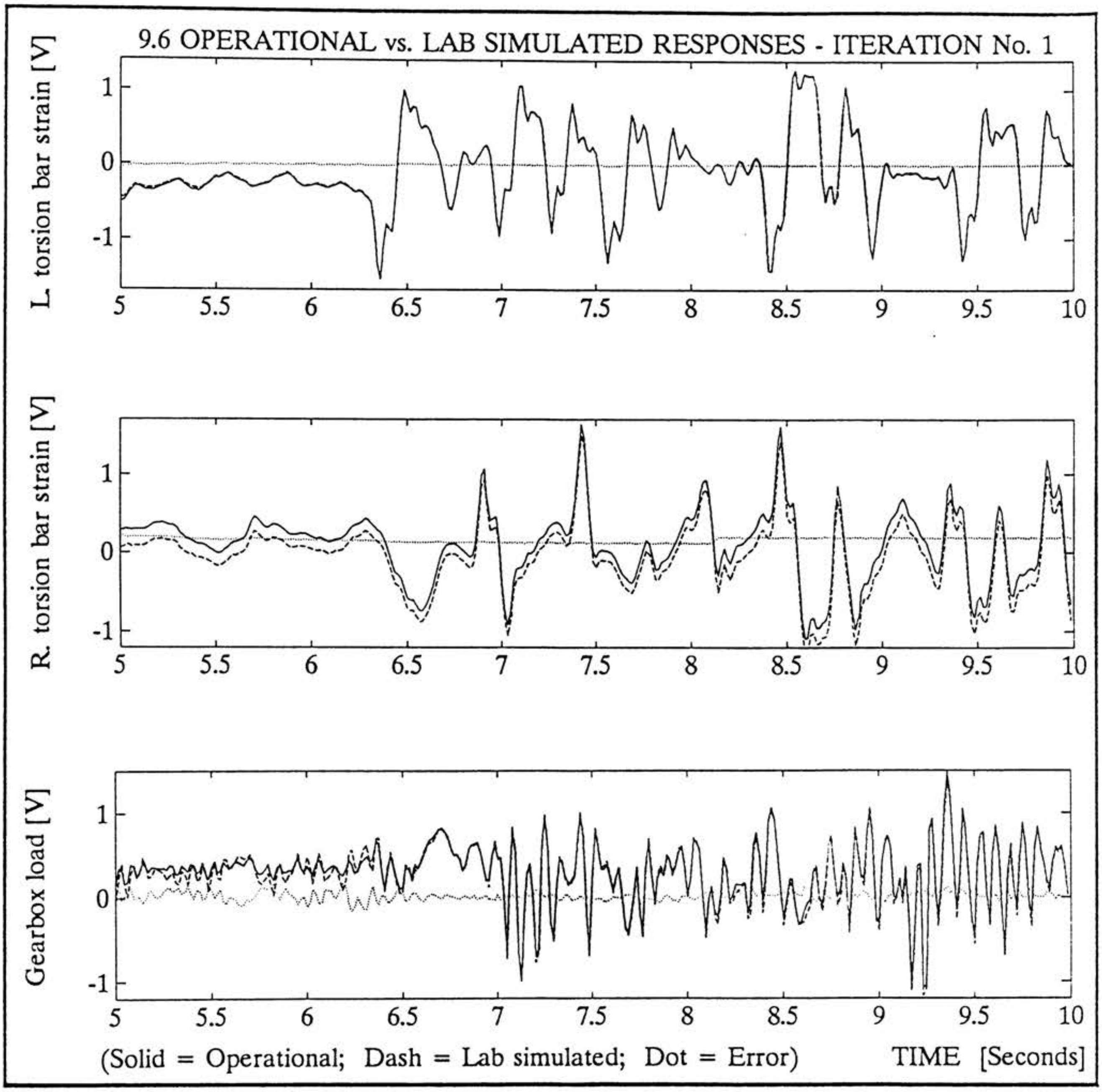


Figure 9.6: Results for data section "A" - Using frequency domain technique - After one iteration

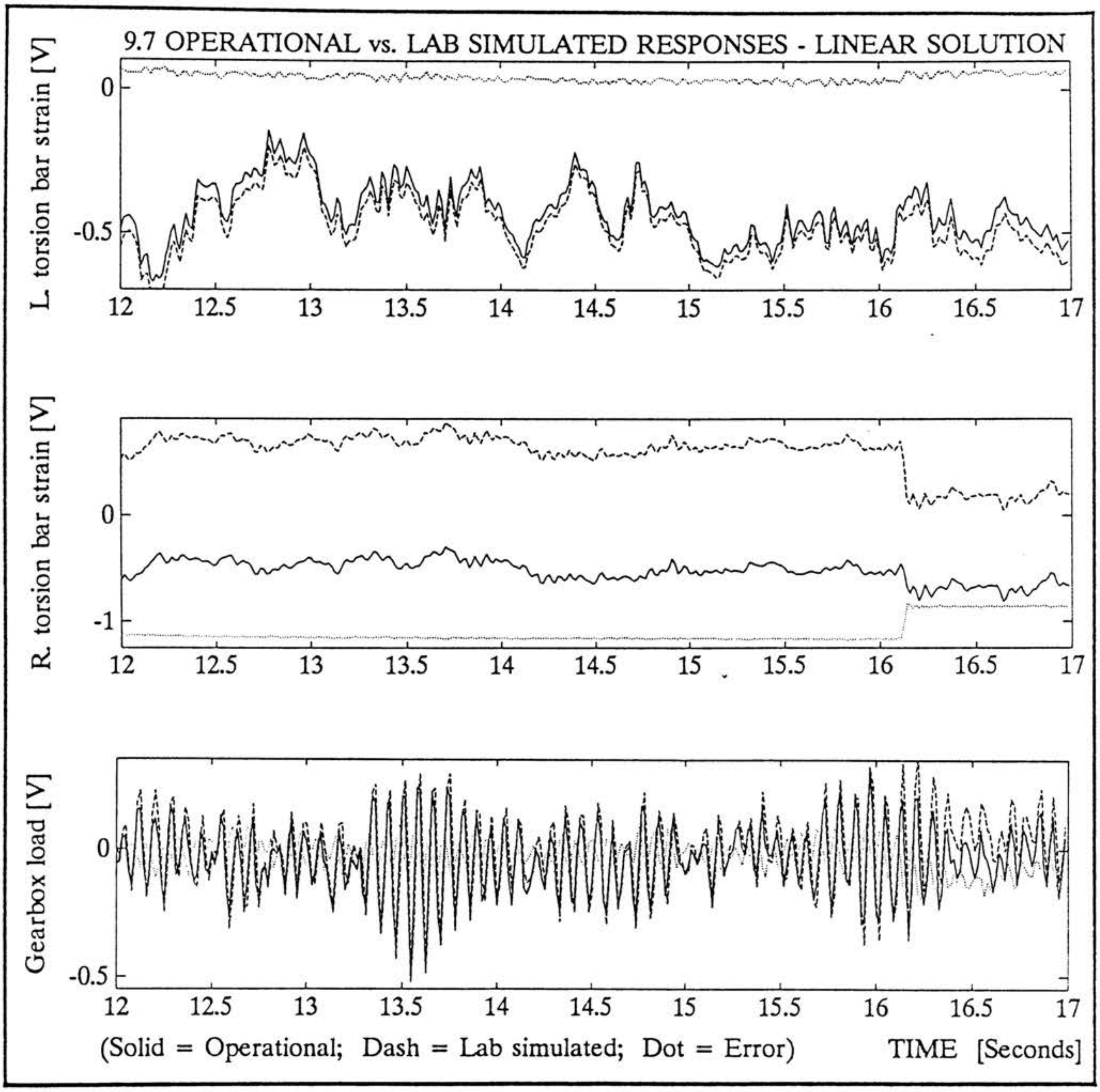


Figure 9.7: Results for data section "B" - Using frequency domain technique - Linear solution

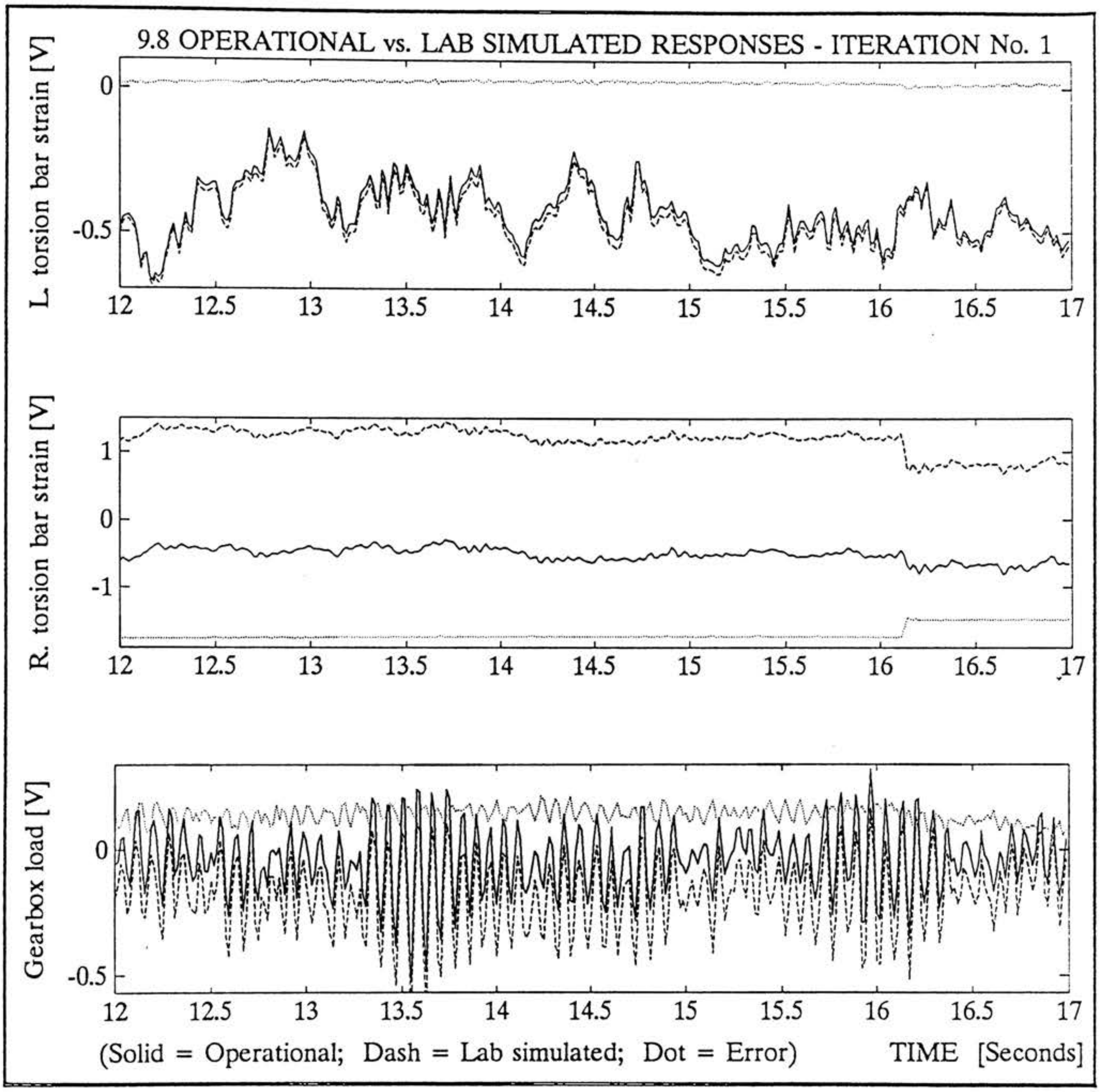


Figure 9.8: Results for data section "B" - Using frequency domain technique - After one iteration

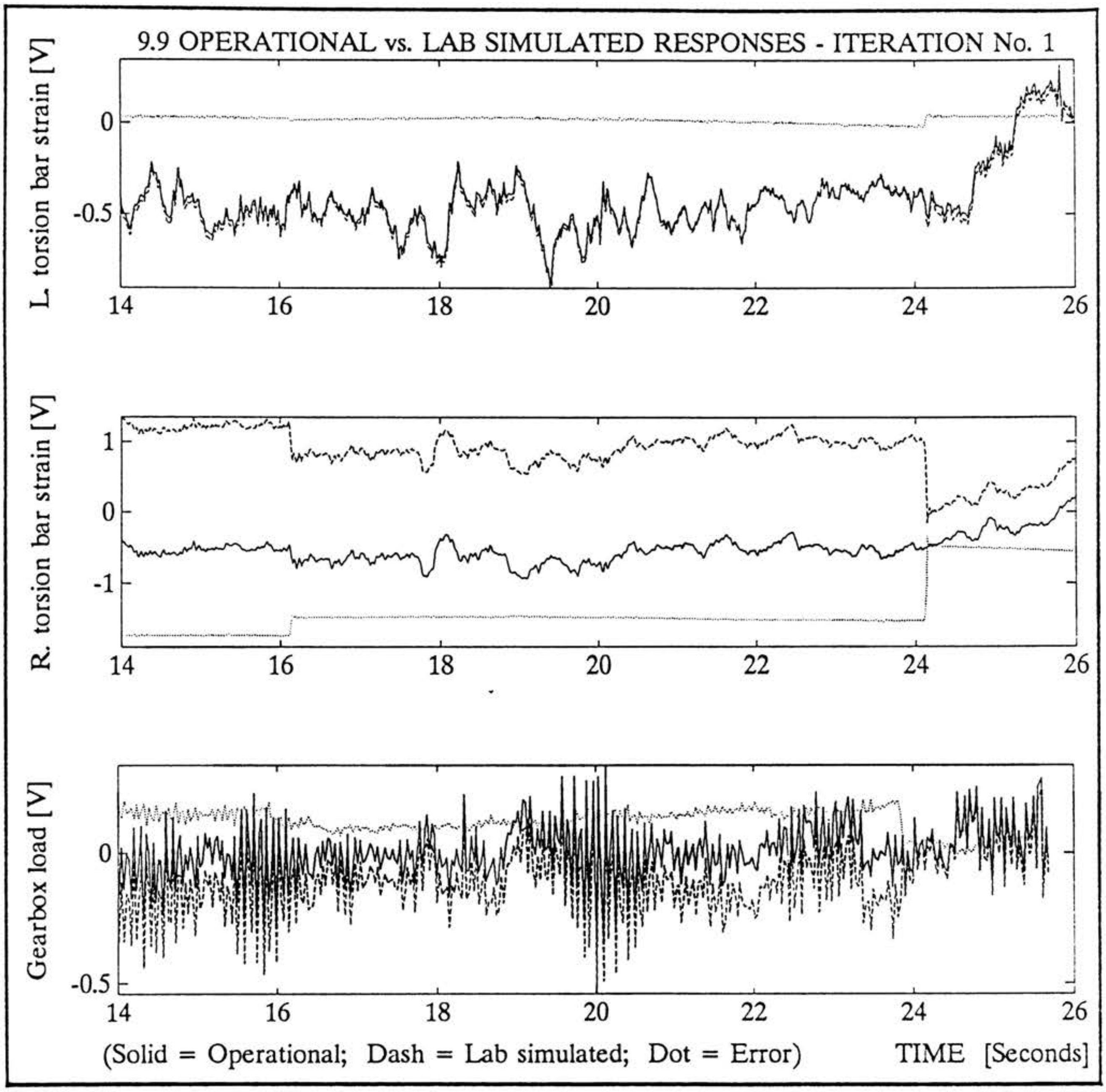


Figure 9.9: Global view of results for data section "B" - Using frequency domain technique

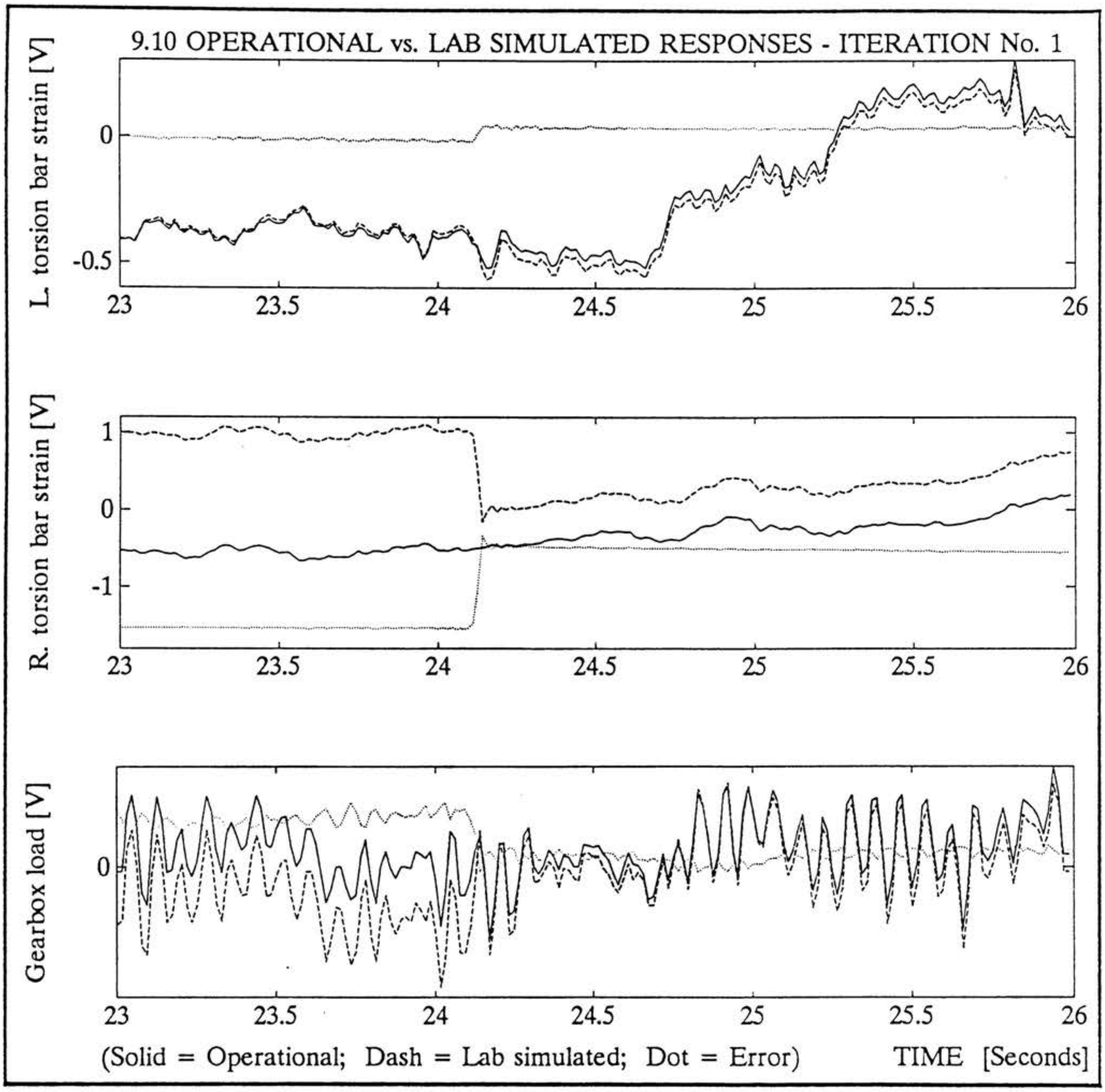


Figure 9.10: Detail view of figure 9.9

9.4 Summary

In this chapter a comparison between the developed time domain and the existing frequency domain techniques was drawn by an application of both techniques to the high speed spring tester case study, as well as the simulation of operational responses on a vehicle suspension.

From the results it seems that the new time domain technique exhibits a number of advantages over the existing frequency domain technique, in as far as versatility, and accuracy is concerned. In particular, two features have been enlightened by these two case studies, namely the ability of the time domain technique to successfully deal with short data sections especially where an impulse is concerned. The second feature is that the time domain technique is not affected by low frequency components which cause difficulties in the frequency domain through the discrete spectral resolution.

Chapter 10

Conclusions and Recomendations for Future Research

10.1 Conclusion

The simulation of operational measured responses on a servo-hydraulic actuator loaded laboratory test-rig requires that the correct actuator input signals be determined. These inputs can in most cases not be measured directly, and instead the response to these inputs are measured under operational conditions, at positions remote from the inputs.

A time domain based computer controlled testing system has been developed which is able to determine the input signals to a servo-hydraulic test rig using remotely measured operational responses, taking the fully coupled multiple axis dynamics of the test system into account.

Parametric dynamic system identification techniques were adapted in a number of ways for the specific application. In particular, a direct reversed inverse model is identified on each input channel, using the principle of a "full order" model. These separate models are subsequently combined in a fully coupled multiple-input multiple-output state space model. Using the pre-recorded operational responses, the test-rig inputs are determined. Provision for nonlinearities is made by a process of iterations, which finally provides an accurate simulation of the remotely measured operational responses on the laboratory test structure.

The test system has been implemented on an 80386 IBM-compatible computer in the MATLAB environment. To prove the integrity of the developed computer software, a number of simple test runs were done on computer simulated "test rigs".

The system was furthermore applied to four practical case studies. The first was a single servo-hydraulic actuator high speed spring tester. Secondly the operational measured responses on a vehicle suspension were reproduced on a full scale laboratory test rig. The third case study concerned the simulation of service acquired vibrational responses on a passenger vehicle. The final case study utilized five servo-hydraulic actuators to simulate operational measured vibrations on a pick-up truck chassis.

As a comparison with existing frequency domain based systems the frequency domain techniques were also applied to two of the above case studies.

These initial tests have indicated several advantages of the time domain based method compared with the existing frequency domain techniques namely:

- The time domain method is not restricted by a discrete spectral resolution and is hence not affected by spectral leakage when the operational responses contain low frequency trends, as is typical in vehicle response data.
- Significantly less data is required for the time domain method.
- Accuracy is generally higher.
- Offsets in the operational measured response data are easily handled.

The development of the time domain technique naturally drew heavily on established dynamic system identification and digital control systems methodologies. Also the basic concept i.e. identification, linear solution and iteration steps come from the original equivalent frequency domain systems. However, to the author's knowledge, a number of aspects might be considered novel to the field of service load reconstruction testing. These might hence be seen as specific contributions to the field of service load reconstruction testing, notably the following:

- The concept of utilizing a time domain approach for obtaining the servo-hydraulic actuator inputs from remotely measured response data, in contrast to the existing frequency domain based approach.

- Identification of separate models for each channel, and the subsequent combination of these separate models into the state space formulation. The conversion of SISO models as well as complete MIMO models to state space is well known in the literature.

- The concept of a "full order" model.

- The formalization of inverse model types and the terminology of a reversed inverse model.

- Reversal of the data vectors in obtaining the reversed inverse models.

- The direct identification of reversed inverse models.

- Off-line application of a linear quadratic optimal servo-controller in calculating the system inputs from the desired operational responses, together with the concept of utilizing iterative corrections to the system inputs from the LQ optimal servo-controller, as well as the method of finding the system delays.

10.2 Recomendations for Future Research

It was naturally not possible to cover all aspects in this study and a number of issues still need to be improved.

Recursive Identification Techniques

The current system does not make use of recursive system identification techniques. A recursion in the number of data points, will certainly prove to be advantageous in the laboratory testing environment where the accuracy of the dynamic model may be improved by adding additional input-output data to an already identified model. This procedure is well established and is described by Ljung and Söderström [1985]. The traditional approach to recursive identification is in the on-line application. Solbrand, Ahlen and Ljung [1985] also describe an off-line recursive procedure applied to a concatenated string of repeated data sequences.

Scaled Corrections During Iteration Process

In applications with the frequency domain techniques, under certain conditions, it has been found advantageous to apply a scaled down correction to the system drive signals between the iterations. This feature has as yet not been implemented in the time domain technique. In comparison with the frequency domain, the time domain technique is able to deal with very large offsets, which can naturally not be scaled down. The offsets will hence have to be dealt with separately.

Coherence Function

The coherence function is probably one of the best indicators to establish optimized input excitation signals for the identification process. Whether the coherence function, which gives the coherence at each spectral line, is as relevant in the time domain as is the case for the frequency domain, would probably have to be established experimentally.

Model Order Estimation

One of the disadvantages of the time domain technique is the requirement of specifying the model orders on each channel prior to the actual identification process. Several techniques are available in the literature which allow the estimation of the order of linear systems. Norton [1986] section 8.4 describes a recursive technique whereby the recursion is in the model order. Although none of these methods were used in this study, they could certainly prove to be very useful and should be implemented in the computer programs. A further method which is proposed, is the prior identification of a frequency domain transfer function to enable the estimation of the order of the system for the subsequent time domain identification. The method is envisaged to comprise the following steps:

- Identify a frequency domain transfer function and ensure a favourable coherence.

- Simulate the transfer function using the input data to obtain simulated output data, or alternatively invert the transfer function and simulate the inverted transfer function using the measured responses to give simulated inputs.

- Using the above simulated input-output data, which is based on a mathematical description, attempt to estimate the model orders on each channel.

- Having so obtained the model orders identify the final model, using the actual experimental input-output data.

Computer Software

The computer software is reasonably friendly, but could certainly improve.

Provision for Nonlinear Systems

Nonlinear systems can only be analysed in the time domain, which makes the developed system superior to the frequency domain techniques. In highly nonlinear systems, provision should be made to identify such models directly, possibly through a direct state space identification procedure. The prediction error methods should prove to give consistent results by using small perturbations on the system parameters.

Generation of Synthetic Input Signals for Identification

In the time domain it is possible to use higher sample frequencies, without spectral resolution problems, and an 8192 point FFT may be considered in the conversion from the PSD to the time signals, which would not be possible in the frequency domain unless the entire analysis including the actual frequency domain transfer function provides for such a high number of spectral lines.

Adaptive Control for Fatigue Testing

After the test rig inputs have been determined, the actual fatigue test may be executed using these calculated inputs. The validity of these inputs will however deteriorate as fatigue cracks start to develop in the test structure, since this will cause a subsequent deterioration of stiffness, see Sherratt [1990]. A solution to this problem would be to drive the test in force control mode, and hence ensure stationary load inputs. Where suspension components in vehicles are tested, large displacements are generally applied, and the test can then only be run in displacement control mode. The only solution would then be to identify a second dynamic model between the actuator displacement inputs and the actuator loadcells. Using the initial measured loads from the loadcells as desired responses, this second dynamic model would be re-identified from time to time to update the actuator displacement drive signals, so as to ensure stationary load inputs to the test rig.

On-line Control

Implementing the test system in an on-line setting, could probably have significant advantages. Aspects such as adaptive control might even be considered in such an application.

H_∞ and Robust Control

The H_∞ and robust control techniques, which are probably considered as state-of-the-art in nonlinear multivariable control system design, might be applied to the present application.

10.3 Applications in Areas Other than Structural Testing

The current time domain technique has been developed essentially for the structural fatigue testing environment as well as other applications requiring the simulation of operational measured responses on servo-hydraulically driven test rigs. The scope for further applications in a variety of different fields is considered to be vast. A few of the many possible areas of investigation are given below.

Dynamic Finite Element Analysis at Critical Points

Forced response dynamic finite element analyses are in general excessively time consuming, especially when attempting to perform a time domain finite element analysis from structural inputs for subsequent fatigue analysis through rainflow counting and cumulative fatigue damage techniques. By employing dynamic system identification on the finite element model, such an analysis may be done within a considerably shorter time. Since a structural finite element model is in most cases linear, this technique should give very accurate results.

The procedure is envisaged as follows:

- Assume that the dynamic structure is loaded by m input loads.
- Perform a dynamic finite element analysis using a short section of input data - typically one to two seconds of data (say 200 to 300 data points), with a reasonably wide frequency bandwidth. The data for the various inputs should be uncorrelated.
- Select at least m critical points of interest on the structure, which will be utilized as response positions. These points could be chosen as critically stressed points, but should exhibit a strong correlation to the m input degrees of freedom.
- From the above finite element system inputs and responses identify a dynamic model for each channel, and combine to a complete discrete state space model.
- Very large amounts of data may now be simulated in a fraction of the time required by the finite element analysis using the state space model.
- The simulated stress vs. time data may thereafter conveniently be used for subsequent fatigue analyses.
- Another m points may thereafter be chosen without repeating the finite element anaysis.

Deriving Input Loads to a Finite Element Analysis from Remotely Measured Response Data

The prime objective of the developed time domain technique is to derive the system inputs from remotely measured response data, since the system inputs can in most cases not be measured directly. This principle is also directly applicable to a finite element analysis by following similar steps to those described in the previous paragraph. As an example, the procedure could be implemented on a vehicle chassis by deriving the suspension input loads to the chassis from *remotely measured operational strain data*. This would in general give significantly improved results over the usual procedure of attempting to determine the input displacements from *measured input acceleration data*. Such a system called Remote Parameter Analysis (RPA) has also recently come into being, see Pountney and Dakin [1992], but is based on static modelling. The time domain method makes provision for a dynamic modelling.

Measurement of Vehicle Terrain Profiles

The measurement of the vertical profile of various roads and terrains for the purpose of vehicle inputs, is an area of possible application for dynamic system identification techniques. It is envisaged to apply the developed techniques as follows:

- A fifth wheel equipped with an accelerometer may be used to traverse the terrain of which the profile is required.

- By exciting the wheel, using a simple single actuator vertical input, a dynamic model may be identified between the displacement input and the accelerometer response. In this specific application, a nonlinear dynamic model of the tyre may produce improved results over a linear model.

- Having traversed the terrain and recorded the system response, an inverse dynamic model could supply the vertical input which caused the measured response.

Identification of Nonlinear Vehicle Models

It is very often required to determine how a vehicle would perform on a specific terrain from a fatigue point of view. A variety of dynamic modelling packages are commercially avaliable to model the specific suspension components, whereafter a finite element model would be required to translate these results into a stress time history for a number of critically stressed areas. Significant improvements in accuracy may be achieved by identifying the complete vehicle between vertical terrain profile inputs and the critically stressed areas. This model should ideally be nonlinear and may be realized by driving the vehicle over a short section of test track with known vertical profile to supply the experimental input-output data for the identification process. The model may thereafter be used to relate various terrains with one another from a fatigue point of view by predicting the stress time response of the vehicle from any other known terrain profile.

References

Allemang R.J (1980): Investigation of Some Multiple Input/Output Frequency Response Function Experimental Modal Analysis Techniques. *Phd dissertation; University of Cincinnati, 1980. U.M.I. Dissertation Service.*

Apte Y.S. (1981): *Linear Multivariable Control Theory.* Tata McGraw-Hill Publishing Co Ltd, New Delhi.

Åström K.J. and Eykhoff P. (1971): System identification - a survey. *Automatica*, vol. 7, pp.123-167, Pergamon Press.

Åström K.J. and Wittenmark B. (1984): *Computer Controlled Systems, Theory and Design.* Prentice Hall Inc., Englewood Cliffs, N.J.

Bannantine J.A., Comer J.J. and Handrock J.L. (1990): *Fundamentals of Metal Fatigue Analysis.* Prentice Hall Inc., Englewood Cliffs, N.J.

Bengtsson G. (1974): Minimal System Inverses for Linear Multivariable Systems. *Journal of Mathematical Analysis and Applications* vol.46, pp 261-274.

Bennett R.J. (1979): *Spatial Time Series.* Pion Ltd, London.

Billings S.A. and Voon W.S.F. (1984): Least Squares Parameter Estimation Algorithms for non-linear Systems *International Journal of System Science* vol. 15, No. 6, pp 601-615.

Braun S. (1986): *Mechanical Signature Analysis theory and applications.* Academic Press Inc.(London).

Brockett R.W. (1965): Poles, Zeros and Feedback: State Space Interpretation. *IEEE Transactions on Automatic Control AC-10*, pp 129-135.

Burden R.L. and Faires J.D. (1985): *Numerical Analysis.* third edition, PWS Publishers

Burg J.P. (1967): Maximum Entropy Spectral Analysis. *Proceedings of the 37th Annual International Meeting of the Society of Exploration Geophysicists*, Oklahoma City, Oklahoma.

Chatfield C. (1980): *The Analysis of Time Series: An Introduction.* second edition, Chapman and Hall.

Coackley W.M. and Butcher N.A. (1992): Development of a technique for analysing the durability aspects of automotive structures by computational methods *Environmental Engineering*, March 1992.

Conle A. and Topper T.H. (1983): Fatigue Service Histories: Techniques for Data Collection and History Reconstruction *SAE Transactions*, vol. 91, p. 299-312.

Cooper J.E. and Wright J.R. (1986): Comparison of some Time Domain System Identification Methods for Free Response Data. *Proceedings of 4th IMAC*, Los Angeles, CA, 1986, vol. 2, pp. 831-836

Craig J.B. (1979): ITFC - How it works and where to use it. *Carl Schenck (UK) Publication*, September 1979.

Craig J. (1975): The Laboratory Simulation of Vehicle Response to Road Profile Excitation. *Symposium on Dynamic Analysis of Structures.* Birniehill Institute, National Engineering Laboratory, East Kilbride, Glasgow

Cryer J.D. (1979): *Time Series analysis.* PWS Publishers.

Desanghere G. and Snoeys R. (1985) Indirect Identification of Excitation Forces by Modal Coordinate Transformation *Proceedings of 3rd IMAC*, Orlando, Florida, 1985, vol. 1, pp. 685-690.

Dodds C.J. (1972): The Response of Vehicle Components to Random Road Surface Undulations. *Phd dissertation; Department of Mechanical Engineering, University of Glasgow.*

Dodds C.J. (1992): Engine Mount Durability Testing - a Discussion of Possible Test Methods *Environmental Engineering*, June 1992.

Dorato P. (1969): On the Inverse of Linear Dynamical Systems. *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-5, No. 1, January 1969.

D'Souza A.F. and Garg V.K. (1984): *Advanced Dynamics Modeling and Analysis.* Prentice Hall Inc., Englewood Cliffs, N.J.

Fletcher J.N. (1990): Global Simulation: A New Technique for Multiaxis Test Control. *Proceedings of the IES, 36th Annual Technical Meeting*, pp. 147-156, April 1990.

Franklin G.F. and Powell J.D. (1980): *Digital Control of Dynamic Systems.* Reading Mass: Addison-Wesley Publishing Co., Inc.

Funk W. and Horst J. (1987): Vehicle Gear Tests in a Test Bed with Very Realistic Service Load Simulation *The Theory of Machines and Mechanisms*, Seville, Spain, September 1987, pp. 907-910.

Galyardt D. and Quantz C. (1987): Comparative Results from Time Domain and Frequency Domain Modal Parameter Estimation Algorithms. *Proceedings of 3rd IMAC*, London, England, 1987, vol. 1, pp. 115-121.

Georgiev N.I. (1989): Road Simulation - Problems and Realizations *Phd dissertation; Department of Electronic Instruments Engineering, Higher School of Mechanical and Electrical Engineering, V.I.LENIN, Sofia, Bulgaria.*

Giordano A.A. and Hsu F.M. (1985): *Least Square Estimation with Applications to Digital Signal Processing.* John Wiley and Sons.

Goodwin G.C. and Payne R.L. (1977): *Dynamic System Identification: Experiment Design and Data Analysis.* Academic Press.

Goodwin G.C. and Sin K.S. (1984): *Adaptive Filtering Prediction and Control.* Prentice Hall Inc., Englewood Cliffs, N.J.

Guidorzi R. (1975): Canonical Structures in the Identification of Multivariable Systems. *Automatica*, vol. 11, pp. 361-374.

Helsel R.J., Evensen H.A. and Pandit S.M. (1988): Experimental Confirmation of Time-Domain Analysis via Autoregressive Models in State Space *Proceedings of 6th IMAC*, Kissimmee, Florida, 1988, pp. 243-249.

Hurd A. (1992): Using data energy content to accelerate road simulation tests *Environmental Engineering*, March 1992.

Kailath T. (1980): *Linear Systems.* Prentice Hall Inc., Englewood Cliffs, N.J.

Kay S.M (1988) *Modern Spectral Estimation, Theory and Applications* Prentice Hall Inc., Englewood Cliffs, N.J.

Klinger F. and Stranzenbach E. (1979): Fatigue testing of cars on road simulators using high speed digital computers. *MTS Systems, VFI*, April 1979.

Laub A.J. and Little J.N. (1986): Control System Toolbox User's Guide. *The MathWorks Inc*, August 27, 1986.

Leontaritis I.J. and Billings S.A. (1985) Input-Output Parametric Models for Non-linear Systems; Part I: Deterministic Non-linear Systems; Part II: Stochastic Non-linear Systems *International Journal of Control*, 1985, vol. 41, No. 2, pp 303-328, pp 329-344.

Leuridan J., Lipkens J., Van der Auweraer H., and Lembregts F. (1986): Global Modal Parameter Estimation Methods: an Assessment of Time versus Frequency Domain Implementation *Proceedings of 4th IMAC*, Los Angeles, CA, 1986, vol. 2, pp. 1586-1595.

Ljung L. (1987): *System Identification Theory for the user.* Prentice Hall Inc., Englewood Cliffs, N.J.

Ljung L. (1988) and (1991): System Identification Toolbox User's Guide. *The MathWorks Inc*, April 1988, and May 1991.

Ljung L. and Söderström T. (1985): *Theory and Practice of Recursive Identification.* The MIT Press, Cambridge, Massachusetts, London, England

Lund R.A. and Donaldson K.H. (Jr.) (1983): Approaches to Vehicle Dynamics and Durability Testing. *SAE Transactions*, vol. 91, pp. 289-298.

Makridakis S. ,Wheelwright S.C. and McGee V.E. (1983): *Forecasting Methods and Applications.* second edition, John Wiley and Sons.

Mahmoud M.S. and Singh M.G. (1981): *Large Scale Systems Modelling.* International series on Sytems and Control, vol. 3, Pergamon Press England

Maine R.E. and Iliff K.W. (1986): Application of Parameter Estimation to Aircraft Stability and Control. *NASA Reference Publication 1168*, Edwards California.

Marquardt D.W. (1963): An algorithm for least-squares estimation of non-linear parameters. *Journal of the society for industrial and applied mathematics*, II; pp. 431-441.

Marsh K.J. (1988): *Full-scale Fatigue Testing of Components and Structures* Butterworths and Co. Ltd

Milne G.M. (1988): State Space Identification Tool for use with MATLAB *The MathWorks Inc*, March, 1988.

Moylan P.J. (1977): Stable Inversion of Linear Systems. *IEEE Transactions on Automatic Control*, February 1977.

Norman J. and Craig J. (1980): Service load simulation testing of complex structures. *CME* pp. 47-48, March 1980.

Norton J.P. (1986): *An Introduction to Identification.* Academic Press, Florida.

Ogata K. (1987): *Discrete Time Control Systems.* Prentice Hall Inc., Englewood Cliffs, N.J.

Papoulis A. (1986): *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Company, Singapore.

Patel R.V. (1977): Minimal Order Inverses for Linear Systems with Zero and Arbitrary Initial States *International Journal of Control*, vol. 30, No. 2, p 245-258.

Patel R.V. and Munro N. (1982): *Multivariable System Theory and Design.* Pergamon Press, England.

Petersen J. and Weissberger G. (1983) The Conception, Description, and Application of a New Vehicle Endurance Test System at AUDI NSU. *SAE Transactions.*

Phillips C.L. and Nagle H.T. Jr (1990): *Digital Control System Analysis and Design.* Prentice Hall Inc., Englewood Cliffs, N.J.

Pountney R.E. and Dakin J.D. (1992) Integration of Test and Analysis for Component Durability. *Environmental Engineering.* June 1992.

Priestly M.B. (1981): *Spectral Analysis and Time Series.* Academic Press Inc., London.

Raath A.D. (1989): Basic Parametric Methods in System Identification. Report no 89-170 *Centre for Structural Mechanics. Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. and Von Fintel W.H. (1989): Load Reconstitution on a Helicopter Airframe. Report no 89-140 *Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. and von Fintel W.H. (1989 b): Load Reconstitution on a Helicopter. Report no 89-141 *Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1990): Dynamic System Identification. Report no 90-050 *Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1990 b): Dynamic System Identification for General Multivariable Black-box State Space Formulations. Report no 90-079*Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1990 c): Gearbox Rear Crossmember - Redesign, Prototype Implementation and Qualification. Report no 90-096*Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1991): Laboratory Fatigue Evaluation of Multistrand Springs. Report no 91-023*Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1991 b): Laboratory Durability Tests on Differential Mounting Bracket and Rear Longitudinal Beams. Report no 91-035*Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Raath A.D. (1991 c): Inversion of Discrete Linear Multivariable Systems. Report no 91-043*Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Richards D.P. (1990): A Review of Analysis and Assessment Methodologies for Road Transportation Vibration and Shock Data *Environmental Engineering*, December 1990.

Rosenbrock H.H. (1970): *State-space and Multivariable Theory.* Thomas Nelson and Sons Ltd, London.

Sherratt F. (1990): Digital Compensation for Single-channel Servo-hydraulic Component Testing Systems - a Re-appraisal. *Environmental Engineering*, September 1990.

Silverman L.M. (1969): Inversion of Multivariable Linear Systems. *IEEE Transactions on Automatic Control*, vol. AC-14, No. 3, June 1969.

Smith K.E. (1984): *An Evaluation of a Least-Squares Time-Domain Parameter Identification Method for Free-Response Measurements. Proceedings of 2nd IMAC*, Orlando, Florida, 1984, vol. 2, pp. 610-615.

Söderström T. and Stoica P. (1989): *System Identification.* Prentice Hall International (UK) Ltd.

Solbrand G. Ahlen A. Ljung L. (1985): Recursive Methods for Off-line Identification. *International Journal of Control*, 1985, vol. 41, No. 1, pp 177-191.

Styles S.S. (1990): Digital Hydraulic Controllers - a Path to High-accuracy Component Testing *Environmental Engineering*, June 1990.

Strejc V. (1981): *State Space Theory of Discrete Linear Control.* John Wiley and Sons.

Styles D.D. and Dodds C.J. (1976): Simulation of Random Environments for Structural Dynamics Testing. *Experimental Mechanics*, vol. 16, Nr. 11, 4/6-424, November 1976.

Sulisz D.W., LaCombe A.J. and Fletcher J.N. (1992): The Application of Modal Analysis and Modeling Techniques to Laboratory Simulation Testing. *Proceedings of 10th IMAC*, February 1992, pp. 870-876.

Tse E. and Weinert H.L. (1975): Structure determination and parameter identification for multivariable stochastic linear systems. *IEEE Transactions on Automatic Control*, vol. AC-20, No. 5, pp. 603-613.

Wang Z. and Fang T. (1986): A Time Domain Method for Identifying Modal Parameters. *Journal of Applied Mechanics, Transactions of the ASME*, March 1986, vol. 53, pp. 28-32.

Wannenburg J. and Immelman C.J. (1991): Chassis Fatigue Life Investigation. Report no 91-076 *Centre for Structural Mechanics, Laboratory for Advanced Engineering (Pty) Ltd* - Pretoria.

Woodside C.M. (1971): Estimation of the Order of Linear Systems. *Automatica*, vol. 7, pp.727-733, Pergamon Press.

Yingxian Y. (1986): Time Domain Identification of Vibration Parameters. *Proceedings of 4th IMAC*, Los Angeles, CA, 1986, vol. 1, pp. 290-296.

Zhao X. (1985): Identification of Mechanical System Modal Parameters using Time Series approach *Proceedings of 3rd IMAC*, Orlando, Florida, 1985, vol. 1, pp. 536-540.

Zhao-qian T. and Yang Q. (1984): Modal Analysis using the Time Series Analysis Method (AR or ARMA model) and its Computer Program Design. *Proceedings of 2nd IMAC*, Orlando, Florida, 1984, vol. 1, pp. 559-565.

Zomotor A., Schwarz K. and Weiler W. (1983) Simulation Methods for evaluating passenger car ride comfort and the fatigue strength of vehicle components. *SAE Transactions*.

Appendix A

State Space Descriptions

A.1 Continuous Time State Space Formulation

Time-invariant linear dynamic systems may be described by linear constant differential equations. The state space formulation stems from the mathematical principle of writing an n-th order linear differential equation into n first-order differential equations, using an n dimensional auxiliary state vector.

For a system with r inputs and p outputs the continuous state space formulation becomes

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{A.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{A.2}$$

where $\mathbf{x}(t) = n$-dimensional state vector
$\mathbf{u}(t) = r$-dimensional input vector
$\mathbf{y}(t) = p$-dimensional output vector
$\mathbf{A} = n \times n$ state matrix
$\mathbf{B} = n \times r$ input matrix
$\mathbf{C} = p \times n$ output matrix
$\mathbf{D} = p \times r$ direct transmission matrix

(A.1) is referred to as the continuous state equation, while
(A.2) is referred to as the continuous output equation

A.2 Discrete State Space Description

In digital control and system identification it is common to work with a discrete version of the state equations. This is achieved through a zero-order-hold sampling of the continuous state equations, under the fundamental assumption that the input signal remains constant over a sampling interval. Under this assumption we may integrate (A.1). Given the state at sampling instant t_k, the state at some future time t is obtained by solving (A.1).

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-s)}Bu(s)ds \qquad \text{(A.3)}$$

The state at the next sampling instant t_{k+1} is given by

$$\begin{aligned} x(t_{k+1}) &= e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Bu(s)ds && \text{(A.4)} \\ &= e^{A(t_{k+1}-t_k)}x(t_k) + Bu(t_k)\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}ds && \text{(A.5)} \end{aligned}$$

Because u is constant between t_k and t_{k+1}.
The sampled system therefore becomes

$$\begin{aligned} \mathbf{x}(t_{k+1}) &= \mathbf{\Phi x}(t_k) + \mathbf{\Gamma u}(t_k) && \text{(A.6)} \\ \mathbf{y}(t_k) &= \mathbf{Cx}(t_k) + \mathbf{Du}(t_k) && \text{(A.7)} \end{aligned}$$

We shall henceforth write the above equations as

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k) && \text{(A.8)} \\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) && \text{(A.9)} \end{aligned}$$

or

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{\Phi x}_k + \mathbf{\Gamma u}_k && \text{(A.10)} \\ \mathbf{y}_k &= \mathbf{Cx}_k + \mathbf{Du}_k && \text{(A.11)} \end{aligned}$$

where k signifies the sampling instant and

$$\Phi = e^{Ah} \tag{A.12}$$

$$\Gamma = \int_0^h e^{As} ds B \tag{A.13}$$

where h is the sampling period.

Computation of Φ and Γ

Several different ways of evaluating (A.12) and (A.13) are available [Åström and Wittenmark 1984]. Since the purpose here is a computer application, the series expansion of the matrix exponential will be used, which is a good machine calculation method. We define

$$\Upsilon = \int_0^h e^{\mathbf{A}s} ds = \mathbf{I}h + \frac{\mathbf{A}h^2}{2!} + \frac{\mathbf{A}^2h^3}{3!} + \ldots\ldots\ldots\ldots \tag{A.14}$$

and compute Φ and Γ from

$$\Phi = \mathbf{I} + \mathbf{A}\Upsilon \tag{A.15}$$

$$\Gamma = \Upsilon\mathbf{B} \tag{A.16}$$

With a sufficient number of terms, the series expansion is quite feasible.

A.3 Canonical Forms

The discrete state and output matrices may be transformed to, amongst others, two convenient forms using transition matrices. The *observable* canonical and the *controllable* canonical forms are standard control and system identification forms. The observable and controllable forms are also called *companion* forms, see Apte [1981].

A.3.1 Observable Canonical Form

We define the nonsingular *observability matrix* by

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix} \tag{A.17}$$

The observable canonical form is then

$$\mathbf{x}(k+1) = \begin{bmatrix} -a_1 & 1 & 0 & . & . & . & 0 \\ -a_2 & 0 & 1 & . & . & . & 0 \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & 0 \\ -a_{n-1} & 0 & 0 & . & . & . & 1 \\ -a_n & 0 & 0 & . & . & . & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_{n-1} \\ b_n \end{bmatrix} u(k) \tag{A.18}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & . & . & . & 0 \end{bmatrix} \mathbf{x}(k) + [b_o]u(k) \tag{A.19}$$

where $\mathbf{\Phi}$ has the characteristic equation

$$det[\lambda\mathbf{I} - \mathbf{\Phi}] = \lambda^n + a_1\lambda^{n-1} + \ldots + a_n = 0 \tag{A.20}$$

The advantage of the observable canonical form is the convenient way of finding the input-output model.

A.3.2 Controllable Canonical Form

We define the nonsingular *controllability matrix* by

$$W_c = [\,\mathbf{\Gamma} \;\; \mathbf{\Phi\Gamma} \; \ldots \; \mathbf{\Phi}^{n-2}\mathbf{\Gamma} \;\; \mathbf{\Phi}^{n-1}\mathbf{\Gamma}\,] \tag{A.21}$$

The controllable canonical form is then given by

$$\mathbf{x}(k+1) = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} b_1 & b_2 & \cdots & b_{n-1} & b_n \end{bmatrix} \mathbf{x}(k) + [b_o]u(k) \tag{A.22}$$

Appendix B

Conversion of ARX to Observable Canonical State Space Models

The derivations in this appendix are based on a similar presentation taken from Ogata [1987], which is valid for SISO systems. In this appendix however, the formulations have been expanded to accomodate MIMO systems. The formulation of the discrete input matrix (equation B.29) in its present form is also novel, see also Raath [1990 b].

A mathematically rigorous treatment of the conversion from the ARX difference equation models to the state space formulation in terms of r-inputs and p-outputs is not given because of the difficulty in presenting partially filled matrices. Instead a specific example for a 3 input and 3 output system is presented. In this way all the necessary steps are illustrated far more clearly, from which it is quite simple to extend to any number of inputs and outputs.

It is furthermore assumed that no delays are present from the system inputs u_i, which makes the presentation more complete. Introducing delays is a simple exercise by zeroing leading b terms.

We start by defining a third order ARX model for output channel y_1. The ARX parameters may be written in matrix form as follows:

$$\mathcal{A}_1 = \left[\begin{array}{cccc} 1 & a_{1_1} & a_{2_1} & a_{3_1} \end{array} \right] y_1 \qquad \text{(B.1)}$$

$$\mathcal{B}_1 = \left[\begin{array}{cccc} b_{10_1} & b_{11_1} & b_{12_1} & b_{13_1} \\ b_{20_1} & b_{21_1} & b_{22_1} & b_{23_1} \\ b_{30_1} & b_{31_1} & b_{32_1} & b_{33_1} \\ & c_{11_1} & c_{12_1} & c_{13_1} \\ & c_{21_1} & c_{22_1} & c_{23_1} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ y_2 \\ y_3 \end{array} \tag{B.2}$$

For output channel 2 we select a second order ARX model, with the following parameters:

$$\mathcal{A}_2 = \left[\begin{array}{ccc} 1 & a_{1_2} & a_{2_2} \end{array}\right] y_2 \tag{B.3}$$

$$\mathcal{B}_2 = \left[\begin{array}{ccc} b_{10_2} & b_{11_2} & b_{12_2} \\ b_{20_2} & b_{21_2} & b_{22_2} \\ b_{30_2} & b_{31_2} & b_{32_2} \\ & c_{11_2} & c_{12_2} \\ & c_{21_2} & c_{22_2} \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ y_1 \\ y_3 \end{array} \tag{B.4}$$

And for output channel 3 we have a fourth order ARX model, with third orders on input u_1 and output y_1 , while a second order model is employed for output y_2.

$$\mathcal{A}_1 = \left[\begin{array}{cccc} 1 & a_{1_3} & a_{2_3} & a_{3_3} \end{array}\right] y_3 \tag{B.5}$$

$$\mathcal{B}_1 = \left[\begin{array}{ccccc} b_{10_3} & b_{11_3} & b_{12_3} & b_{13_3} & b_{14_3} \\ b_{20_3} & b_{21_3} & b_{22_3} & b_{23_3} & b_{24_3} \\ b_{30_3} & b_{31_3} & b_{32_3} & b_{33_3} & \\ & c_{11_3} & c_{12_3} & c_{13_3} & \\ & c_{21_3} & c_{22_3} & & \end{array}\right] \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ y_1 \\ y_2 \end{array} \tag{B.6}$$

The above three ARX models will be combined into a complete three input three output state space model in the observable canonical form.

The corresponding difference equations may be written as follows:

$$\begin{aligned}
y_1(k) &+ a_{1_1}y_1(k-1) + a_{2_1}y_1(k-2) + a_{3_1}y_1(k-3) = \\
b_{10_1}u_1(k) &+ b_{11_1}u_1(k-1) + b_{12_1}u_1(k-2) + b_{13_1}u_1(k-3) \\
+b_{20_1}u_2(k) &+ b_{21_1}u_2(k-1) + b_{22_1}u_2(k-2) + b_{23_1}u_2(k-3) \\
+b_{30_1}u_3(k) &+ b_{31_1}u_3(k-1) + b_{32_1}u_3(k-2) + b_{33_1}u_3(k-3) \\
&+ c_{11_1}y_2(k-1) + c_{12_1}y_2(k-2) + c_{13_1}y_2(k-3) \\
&+ c_{21_1}y_3(k-1) + c_{22_1}y_3(k-2) + c_{23_1}y_3(k-3) \qquad (B.7)
\end{aligned}$$

and for output channel y_2 we have

$$\begin{aligned}
y_2(k) &+ a_{1_2}y_2(k-1) + a_{2_2}y_2(k-2) = \\
b_{10_2}u_1(k) &+ b_{11_2}u_1(k-1) + b_{12_2}u_1(k-2) + \\
+b_{20_2}u_2(k) &+ b_{21_2}u_2(k-1) + b_{22_2}u_2(k-2) + \\
+b_{30_2}u_3(k) &+ b_{31_2}u_3(k-1) + b_{32_2}u_3(k-2) + \\
&+ c_{11_2}y_1(k-1) + c_{12_2}y_1(k-2) + \\
&+ c_{21_2}y_3(k-1) + c_{22_2}y_3(k-2) + \qquad (B.8)
\end{aligned}$$

and for output channel y_3,

$$\begin{aligned}
y_3(k) &+ a_{1_3}y_3(k-1) + a_{2_3}y_3(k-2) + a_{3_3}y_3(k-3) = \\
b_{10_3}u_1(k) &+ b_{11_3}u_1(k-1) + b_{12_3}u_1(k-2) + b_{13_3}u_1(k-3) + b_{14_3}u_1(k-4) \\
+b_{20_3}u_2(k) &+ b_{21_3}u_2(k-1) + b_{22_3}u_2(k-2) + b_{23_3}u_2(k-3) + b_{24_3}u_2(k-4) \\
+b_{30_3}u_3(k) &+ b_{31_3}u_3(k-1) + b_{32_3}u_3(k-2) + b_{33_3}u_3(k-3) \\
&+ c_{11_3}y_1(k-1) + c_{12_3}y_1(k-2) \\
&+ c_{21_3}y_2(k-1) + c_{22_3}y_2(k-2) \qquad (B.9)
\end{aligned}$$

Taking the z-transform of (B.7), gives

$$\begin{aligned}
Y_1(z) &= b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + \\
&+ z^{-1}(b_{11_1}U_1 + b_{21_1}U_2 + b_{31_1}U_3 + c_{11_1}Y_2 + c_{21_1}Y_3 - a_{1_1}Y_1) \\
&+ z^{-2}(b_{12_1}U_1 + b_{22_1}U_2 + b_{32_1}U_3 + c_{12_1}Y_2 + c_{22_1}Y_3 - a_{2_1}Y_1) \\
&+ z^{-3}(b_{13_1}U_1 + b_{23_1}U_2 + b_{33_1}U_3 + c_{13_1}Y_2 + c_{23_1}Y_3 - a_{3_1}Y_1) \\
&\qquad (B.10)
\end{aligned}$$

where we have dropped the z argument in the latter part of the equation. This may be written as

$$\begin{aligned}
Y_1(z) &= b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + \\
&+ z^{-1}\{b_{11_1}U_1 + b_{21_1}U_2 + b_{31_1}U_3 + c_{11_1}Y_2 + c_{21_1}Y_3 - a_{1_1}Y_1 \\
&+ z^{-1}[b_{12_1}U_1 + b_{22_1}U_2 + b_{32_1}U_3 + c_{12_1}Y_2 + c_{22_1}Y_3 - a_{2_1}Y_1 \\
&+ z^{-1}(b_{13_1}U_1 + b_{23_1}U_2 + b_{33_1}U_3 + c_{13_1}Y_2 + c_{23_1}Y_3 - a_{3_1}Y_1)]\}
\end{aligned} \tag{B.11}$$

similarly taking z-transforms of (B.8) and (B.9) we obtain

$$\begin{aligned}
Y_2(z) &= b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + \\
&+ z^{-1}\{b_{11_2}U_1 + b_{21_2}U_2 + b_{31_2}U_3 + c_{11_2}Y_1 + c_{21_2}Y_3 - a_{1_2}Y_2 \\
&+ z^{-1}[b_{12_2}U_1 + b_{22_2}U_2 + b_{32_2}U_3 + c_{12_2}Y_1 + c_{22_2}Y_3 - a_{2_2}Y_2]\}
\end{aligned} \tag{B.12}$$

and

$$\begin{aligned}
Y_3(z) &= b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + \\
&+ z^{-1}\{b_{11_3}U_1 + b_{21_3}U_2 + b_{31_3}U_3 + c_{11_3}Y_1 + c_{21_3}Y_2 - a_{1_3}Y_3 \\
&+ z^{-1}[b_{12_3}U_1 + b_{22_3}U_2 + b_{32_3}U_3 + c_{12_3}Y_1 + c_{22_3}Y_2 - a_{2_3}Y_3 \\
&+ z^{-1}(b_{13_3}U_1 + b_{23_3}U_2 + b_{33_3}U_3 - a_{3_3}Y_3 \\
&+ z^{-1}(b_{14_3}U_1 + b_{24_3}U_2))]\}
\end{aligned} \tag{B.13}$$

Using the above three equations we may now define state variables as follows:

$$\begin{aligned}
X_3 &= z^{-1}(b_{13_1}U_1 + b_{23_1}U_2 + b_{33_1}U_3 + c_{13_1}Y_2 + c_{23_1}Y_3 - a_{3_1}Y_1) \\
X_2 &= z^{-1}[b_{12_1}U_1 + b_{22_1}U_2 + b_{32_1}U_3 + c_{12_1}Y_2 + c_{22_1}Y_3 - a_{2_1}Y_1 + X_3] \\
X_1 &= z^{-1}\{b_{11_1}U_1 + b_{21_1}U_2 + b_{31_1}U_3 + c_{11_1}Y_2 + c_{21_1}Y_3 - a_{1_1}Y_1 + X_2\}
\end{aligned} \tag{B.14}$$

$$X_5 = z^{-1}[b_{12_2}U_1 + b_{22_2}U_2 + b_{32_2}U_3 + c_{12_2}Y_1 + c_{22_2}Y_3 - a_{2_2}Y_2]$$

$$X_4 = z^{-1}\{b_{11_2}U_1 + b_{21_2}U_2 + b_{31_2}U_3 + c_{11_2}Y_1 + c_{21_2}Y_3 - a_{1_2}Y_2 + X_5\} \tag{B.15}$$

$$\begin{aligned} X_9 &= z^{-1}(b_{14_3}U_1 + b_{24_3}U_2) \\ X_8 &= z^{-1}(b_{13_3}U_1 + b_{23_3}U_2 + b_{33_3}U_3 - a_{3_3}Y_3 + X_9) \\ X_7 &= z^{-1}[b_{12_3}U_1 + b_{22_3}U_2 + b_{32_3}U_3 + c_{12_3}Y_1 + c_{22_3}Y_2 - a_{2_3}Y_3 + X_8] \\ X_6 &= z^{-1}\{b_{11_3}U_1 + b_{21_3}U_2 + b_{31_3}U_3 + c_{11_3}Y_1 + c_{21_3}Y_2 - a_{1_3}Y_3 + X_7\} \end{aligned} \tag{B.16}$$

Using the above defined states, equations (B.11) through (B.13) become

$$\begin{aligned} Y_1(z) &= b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1 \\ Y_2(z) &= b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4 \\ Y_3(z) &= b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6 \end{aligned} \tag{B.17}$$

Substituting the above three equations into the nine state equations gives

$$\begin{aligned} zX_1 &= b_{11_1}U_1 + b_{21_1}U_2 + b_{31_1}U_3 \\ &+ c_{11_1}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) \\ &+ c_{21_1}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) \\ &- a_{1_1}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z)) + X_1) + X_2 \end{aligned} \tag{B.18}$$

$$\begin{aligned} zX_2 &= b_{12_1}U_1 + b_{22_1}U_2 + b_{32_1}U_3 \\ &+ c_{12_1}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) \\ &+ c_{22_1}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) \\ &- a_{2_1}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) + X_3 \end{aligned} \tag{B.19}$$

$$\begin{aligned} zX_3 &= b_{13_1}U_1 + b_{23_1}U_2 + b_{33_1}U_3 \\ &+ c_{13_1}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) \\ &+ c_{23_1}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) \\ &- a_{3_1}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) \end{aligned} \tag{B.20}$$

$$
\begin{aligned}
zX_4 &= b_{11_2}U_1 + b_{21_2}U_2 + b_{31_2}U_3 \\
&+ c_{11_2}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) \\
&+ c_{21_2}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) \\
&- a_{1_2}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) + X_5
\end{aligned}
\tag{B.21}
$$

$$
\begin{aligned}
zX_5 &= b_{12_2}U_1 + b_{22_2}U_2 + b_{32_2}U_3 \\
&+ c_{12_2}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) \\
&+ c_{22_2}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) \\
&- a_{2_2}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4)
\end{aligned}
\tag{B.22}
$$

$$
\begin{aligned}
zX_6 &= b_{11_3}U_1 + b_{21_3}U_2 + b_{31_3}U_3 \\
&+ c_{11_3}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) \\
&+ c_{21_3}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) \\
&- a_{1_3}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) + X_7
\end{aligned}
\tag{B.23}
$$

$$
\begin{aligned}
zX_7 &= b_{12_3}U_1 + b_{22_3}U_2 + b_{32_3}U_3 \\
&+ c_{12_3}(b_{10_1}U_1(z) + b_{20_1}U_2(z) + b_{30_1}U_3(z) + X_1) \\
&+ c_{22_3}(b_{10_2}U_1(z) + b_{20_2}U_2(z) + b_{30_2}U_3(z) + X_4) \\
&- a_{2_3}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) + X_8
\end{aligned}
\tag{B.24}
$$

$$
\begin{aligned}
zX_8 &= b_{13_3}U_1 + b_{23_3}U_2 + b_{33_3}U_3 \\
&- a_{3_3}(b_{10_3}U_1(z) + b_{20_3}U_2(z) + b_{30_3}U_3(z) + X_6) + X_9
\end{aligned}
\tag{B.25}
$$

$$
zX_9 = b_{14_3}U_1 + b_{24_3}U_2 \tag{B.26}
$$

Taking the inverse $z-$transform of the above nine equations, and writing the result into matrix form, gives the following state equation:

$$
\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) \tag{B.27}
$$

where the matices $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are given by:

$$\boldsymbol{\Phi} = \begin{bmatrix} -a_{1_1} & 1 & \cdot & c_{11_1} & \cdot & c_{21_1} & \cdot & \cdot & \cdot \\ -a_{2_1} & \cdot & 1 & c_{12_1} & \cdot & c_{22_1} & \cdot & \cdot & \cdot \\ -a_{3_1} & \cdot & \cdot & c_{13_1} & \cdot & c_{23_1} & \cdot & \cdot & \cdot \\ c_{11_2} & \cdot & \cdot & -a_{1_2} & 1 & c_{21_2} & \cdot & \cdot & \cdot \\ c_{12_2} & \cdot & \cdot & -a_{2_2} & \cdot & c_{22_2} & \cdot & \cdot & \cdot \\ c_{11_3} & \cdot & \cdot & c_{21_3} & \cdot & -a_{1_3} & 1 & \cdot & \cdot \\ c_{12_3} & \cdot & \cdot & c_{22_3} & \cdot & -a_{2_3} & \cdot & 1 & \cdot \\ 0 & \cdot & \cdot & 0 & \cdot & -a_{3_3} & \cdot & \cdot & 1 \\ 0 & \cdot & \cdot & 0 & \cdot & 0 & \cdot & \cdot & \cdot \end{bmatrix} \tag{B.28}$$

and

$$\boldsymbol{\Gamma} = \boldsymbol{\Psi\Omega} + \boldsymbol{\Lambda} \tag{B.29}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} -a_{1_1} & c_{11_1} & c_{21_1} \\ -a_{2_1} & c_{12_1} & c_{22_1} \\ -a_{3_1} & c_{13_1} & c_{23_1} \\ c_{11_2} & -a_{1_2} & c_{21_2} \\ c_{12_2} & -a_{2_2} & c_{22_2} \\ c_{11_3} & c_{21_3} & -a_{1_3} \\ c_{12_3} & c_{22_3} & -a_{2_3} \\ 0 & 0 & -a_{3_3} \\ 0 & 0 & 0 \end{bmatrix} \tag{B.30}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} b_{11_1} & b_{21_1} & b_{31_1} \\ b_{12_1} & b_{22_1} & b_{32_1} \\ b_{13_1} & b_{23_1} & b_{33_1} \\ b_{11_2} & b_{21_2} & b_{31_2} \\ b_{12_2} & b_{22_2} & b_{32_2} \\ b_{11_3} & b_{21_3} & b_{31_3} \\ b_{12_3} & b_{22_3} & b_{32_3} \\ b_{13_3} & b_{23_3} & b_{33_3} \\ b_{14_3} & b_{24_3} & 0 \end{bmatrix} \tag{B.31}$$

$$\boldsymbol{\Omega} = \begin{bmatrix} b_{10_1} & b_{20_1} & b_{30_1} \\ b_{10_2} & b_{20_2} & b_{30_2} \\ b_{10_3} & b_{20_3} & b_{30_3} \end{bmatrix} \tag{B.32}$$

The output equation is obtained by taking the inverse $z-$transform of (B.17) and writing the result into matrix form, which gives:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \\ x_7(k) \\ x_8(k) \\ x_9(k) \end{bmatrix} + \begin{bmatrix} b_{10_1} & b_{20_1} & b_{30_1} \\ b_{10_2} & b_{20_2} & b_{30_2} \\ b_{10_3} & b_{20_3} & b_{30_3} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix} \tag{B.33}$$

or

$$\mathbf{y}(k) = \mathbf{Cx}(k) + \mathbf{Du}(k) \tag{B.34}$$

The above formulation is in the well known observable canonical form.

$$\begin{aligned} \mathbf{x}(k+1) &= \boldsymbol{\Phi}\mathbf{x}(k) + \boldsymbol{\Gamma}\mathbf{u}(k) & \text{(B.35)} \\ \mathbf{y}(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k) & \text{(B.36)} \end{aligned}$$

Appendix C

Conversion of MFD to State Space Models

In appendix B the conversion from the ARX difference equation models to the observable canonical state space form compelled at least a single delay on the outputs, which means that the parameters $c_{10_1}, c_{20_1}, \ldots c_{(p-1)0_1}$ must be zero. In some instances, this restriction may not be desirable, or even possible, and a conversion is hence sought whereby these parameters need not equal zero. A conversion to the state space formulation is then still possible, albeit not in the observable canonical form. The conversion makes use of the matrix fraction description MFD, as discussed in section 4.3.1, hence the title of this appendix. The presentation in this appendix is based on a similar presentation in Strejc [1981] and is given for the sake of completeness.

Consider again a system with three inputs and three outputs. We start by defining a second order ARX model for output channel y_1

$$\begin{aligned} y_1(k) &+ a_{1_1} y_1(k-1) + a_{2_1} y_1(k-2) = \\ b_{10_1} u_1(k) &+ b_{11_1} u_1(k-1) + b_{12_1} u_1(k-2) \\ +b_{20_1} u_2(k) &+ b_{21_1} u_2(k-1) + b_{22_1} u_2(k-2) \\ +b_{30_1} u_3(k) &+ b_{31_1} u_3(k-1) + b_{32_1} u_3(k-2) \\ +c_{10_1} y_2(k) &+ c_{11_1} y_2(k-1) + c_{12_1} y_2(k-2) \\ +c_{20_1} y_3(k) &+ c_{21_1} y_3(k-1) + c_{22_1} y_3(k-2) \end{aligned} \tag{C.1}$$

for output channel y_2 we also have a second order ARX model

$$\begin{aligned}
y_2(k) &+ a_{1_2}y_2(k-1)+a_{2_2}y_2(k-2) = \\
b_{10_2}u_1(k) &+ b_{11_2}u_1(k-1)+b_{12_2}u_1(k-2)+ \\
+b_{20_2}u_2(k) &+ b_{21_2}u_2(k-1)+b_{22_2}u_2(k-2)+ \\
+b_{30_2}u_3(k) &+ b_{31_2}u_3(k-1)+b_{32_2}u_3(k-2)+ \\
+c_{10_2}y_1(k) &+ c_{11_2}y_1(k-1)+c_{12_2}y_1(k-2)+ \\
+c_{20_2}y_3(k) &+ c_{21_2}y_3(k-1)+c_{22_2}y_3(k-2)+
\end{aligned} \tag{C.2}$$

and the same for output channel y_3

$$\begin{aligned}
y_3(k) &+ a_{1_3}y_3(k-1)+a_{2_3}y_3(k-2) = \\
b_{10_3}u_1(k) &+ b_{11_3}u_1(k-1)+b_{12_3}u_1(k-2) \\
+b_{20_3}u_2(k) &+ b_{21_3}u_2(k-1)+b_{22_3}u_2(k-2) \\
+b_{30_3}u_3(k) &+ b_{31_3}u_3(k-1)+b_{32_3}u_3(k-2) \\
+c_{10_3}y_1(k) &+ c_{11_3}y_1(k-1)+c_{12_3}y_1(k-2) \\
+c_{20_3}y_2(k) &+ c_{21_3}y_2(k-1)+c_{22_3}y_2(k-2)
\end{aligned} \tag{C.3}$$

Using the delay operator q^{-1}, the above three equations may be written in matrix fraction description as follows:

$$\left\{\begin{bmatrix} 1 & -c_{10_1} & -c_{20_1} \\ -c_{10_2} & 1 & -c_{20_2} \\ -c_{10_3} & -c_{20_3} & 1 \end{bmatrix} + \begin{bmatrix} a_{1_1} & -c_{11_1} & -c_{21_1} \\ -c_{11_2} & a_{1_2} & -c_{21_2} \\ -c_{11_3} & -c_{21_3} & a_{1_3} \end{bmatrix} q^{-1} + \right.$$

$$\left. \begin{bmatrix} a_{1_1} & -c_{12_1} & -c_{22_1} \\ -c_{12_2} & a_{1_2} & -c_{22_2} \\ -c_{12_3} & -c_{22_3} & a_{1_3} \end{bmatrix} q^{-2} \right\} \left\{ \begin{matrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{matrix} \right\} =$$

$$\left\{ \begin{bmatrix} b_{10_1} & b_{20_1} & b_{30_1} \\ b_{10_2} & b_{20_2} & b_{30_2} \\ b_{10_3} & b_{20_3} & b_{30_3} \end{bmatrix} + \begin{bmatrix} b_{11_1} & b_{21_1} & b_{31_1} \\ b_{11_2} & b_{21_2} & b_{31_2} \\ b_{11_3} & b_{21_3} & b_{31_3} \end{bmatrix} q^{-1} + \begin{bmatrix} b_{12_1} & b_{22_1} & b_{32_1} \\ b_{12_2} & b_{22_2} & b_{32_2} \\ b_{12_3} & b_{22_3} & b_{32_3} \end{bmatrix} q^{-2} \right\} \left\{ \begin{matrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{matrix} \right\} \tag{C.4}$$

Using the z-transform, this may be written as

$$[A_0 + A_1 z^{-1} + A_2 z^{-2}]\mathbf{Y}(z) = [B_0 + B_1 z^{-1} + B_2 z^{-2}]\mathbf{U}(z) \tag{C.5}$$

from which

$$[A_0 z + A_1 + A_2 z^{-1}]\mathbf{Y}(z) = [B_0 z + B_1 + B_2 z^{-1}]\mathbf{U}(z) \tag{C.6}$$

and

$$[A_0 z^2 + A_1 z + A_2]\mathbf{Y}(z) = [B_0 z^2 + B_1 z^1 + B_2]\mathbf{U}(z) \tag{C.7}$$

We may now define state variables as follows:

$$\mathbf{X}_1(z) = A_0 \mathbf{Y}(z) - B_0 \mathbf{U}(z) \tag{C.8}$$

$$\mathbf{X}_2(z) = A_0 z \mathbf{Y}(z) + A_1 \mathbf{Y}(z) - B_0 z \mathbf{U}(z) - B_1 \mathbf{U}(z) \tag{C.9}$$

where the state vectors $\mathbf{X}_i(z)$ each consist of a vector with as many elements as there are inputs.

From the above two equations we may now write

$$z\mathbf{X}_1(z) = zA_0 \mathbf{Y}(z) - zB_0 \mathbf{U}(z) \tag{C.10}$$

$$z\mathbf{X}_2(z) = A_0 z^2 \mathbf{Y}(z) + A_1 z \mathbf{Y}(z) - B_0 z^2 \mathbf{U}(z) - B_1 z \mathbf{U}(z) \tag{C.11}$$

we also see that

$$\mathbf{X}_2(z) = z\mathbf{X}_1 + A_1 \mathbf{Y}(z) - B_1 \mathbf{U}(z) \tag{C.12}$$

$$0 = z\mathbf{X}_2 + A_2 \mathbf{Y}(z) - B_2 \mathbf{U}(z) \tag{C.13}$$

and from (C.8) we solve for $Y(z)$

$$\mathbf{Y}(z) = A_0^{-1}\mathbf{X}_1(z) + A_0^{-1} B_0 \mathbf{U}(z) \tag{C.14}$$

which we may substitute into the two state variable equations giving

$$
\begin{aligned}
\mathbf{X}_2(z) &= z\mathbf{X}_1 + A_1A_0^{-1}\mathbf{X}_1(z) + A_1A_0^{-1}B_0\mathbf{U}(z) - B_1\mathbf{U}(z) & \text{(C.15)} \\
0 &= z\mathbf{X}_2 + A_2A_0^{-1}\mathbf{X}_1(z) + A_2A_0^{-1}B_0\mathbf{U}(z) - B_2\mathbf{U}(z) & \text{(C.16)}
\end{aligned}
$$

or

$$
\begin{aligned}
\mathbf{X}_2(z) &= z\mathbf{X}_1 + A_1A_0^{-1}\mathbf{X}_1(z) + A_1A_0^{-1}B_0\mathbf{U}(z) - B_1\mathbf{U}(z) & \text{(C.17)} \\
0 &= z\mathbf{X}_2 + A_2A_0^{-1}\mathbf{X}_1(z) + A_2A_0^{-1}B_0\mathbf{U}(z) - B_2\mathbf{U}(z) & \text{(C.18)}
\end{aligned}
$$

Taking the inverse z-transform and writing in matrix form, the state equation becomes

$$
\left[\begin{array}{c} \mathbf{x_1}(k+1) \\ \mathbf{x_2}(k+1) \end{array}\right] = \left[\begin{array}{cc} -A_1A_0^{-1} & I \\ -A_2A_0^{-1} & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{x_1}(k) \\ \mathbf{x_2}(k) \end{array}\right] + \left[\begin{array}{c} B_1 - A_1A_0{-}1B_0 \\ B_2 - A_2A_0{-}1B_0 \end{array}\right] \mathbf{u}(k) \tag{C.19}
$$

with output equation

$$
\mathbf{y}(k) = \left[\begin{array}{cccc} A_0^{-1}\cdot & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{x}_1(k) \\ \mathbf{x}_2(k) \end{array}\right] + \left[\begin{array}{c} A_0^{-1}B_0 \end{array}\right] \mathbf{u}(k) \tag{C.20}
$$

The above formulation is of course only possible if the inverse of A_0 exists. In general however this formulation would be preferred over the one given in the previous chapter since one would expect no delays between the various outputs, and would want to be able to incorporate the no-delay terms $c_{10_1}, c_{20_1}, ...c_{(p-1)0_1}$ in the model formulation.

Appendix D

Linear Quadratic Optimal Servo-controller

In some instances it is not possible to invert the system either to a forward or to a reversed inverse, due to the system exhibiting an unstable inverse in both cases. The only possible solution is then a synthesis of the model using a linear quadratic optimal servo-controller. In multivariable systems this is one of the only successful methods of obtaining the system inputs from the desired responses. This presentation is based on a similar LQ controller in Ogata [1987]. Certain deviations from Ogata's presentation have been made: firstly provision has been made for the direct transmission matrix, and secondly the application of the LQ controller in this specific manner, especially as regards the system delays, is considered novel.

We shall henceforth in this section refer to the model of the dynamic system as the plant, which we assume to be described by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{u}(k) && \text{(D.1)} \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) && \text{(D.2)} \end{aligned}$$

Note that we are dealing with a system with m inputs and m outputs, and all symbols therefore represent vectors or matrices. We add an integrator to the plant, and also use full state feedback.

The integrator model is given by the following two equivalent equations:

$$\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{r}(k) - \mathbf{y}(k) \qquad \text{(D.3)}$$

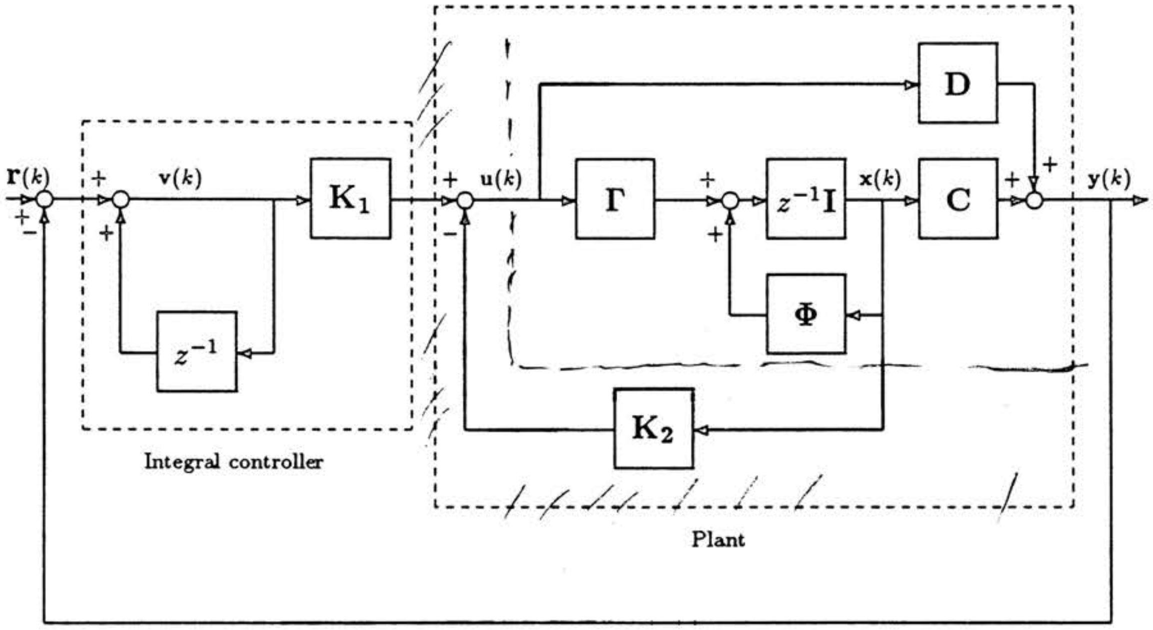


Figure D.1: Block diagram representation of optimal servo-controller

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \mathbf{r}(k+1) - \mathbf{y}(k+1) \tag{D.4}$$

The control vector for the plant is given by:

$$\mathbf{u}(k) = \mathbf{K_1 v}(k) - \mathbf{K_2 x}(k) \tag{D.5}$$

The purpose is to find the gain matrices $\mathbf{K_1}$ and $\mathbf{K_2}$ by minimizing a cost function or performance index which we shall denote by J .

From (D.4), we find by substituting (D.1) and (D.2):

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{v}(k) + \mathbf{r}(k+1) - \mathbf{C}[\mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k)] - \mathbf{Du}(k+1) \\ &= -\mathbf{C\Phi x}(k) + \mathbf{v}(k) - \mathbf{C\Gamma u}(k) - \mathbf{Du}(k+1) + \mathbf{r}(k+1) \end{aligned} \tag{D.6}$$

and from (D.5)

$$\begin{aligned} \mathbf{u}(k+1) &= -\mathbf{K}_2\mathbf{x}(k+1) + \mathbf{K}_1\mathbf{v}(k+1) \\ &= -\mathbf{K}_2[\mathbf{\Phi x}(k) + \mathbf{\Gamma u}(k)] + \mathbf{K}_1[-\mathbf{C\Phi x}(k) + \mathbf{v}(k) - \mathbf{C\Gamma u}(k) \\ &\quad -\mathbf{Du}(k+1) + \mathbf{r}(k+1)] \end{aligned}$$

or

$$\begin{aligned} [\mathbf{I}_m + \mathbf{K}_1\mathbf{D}]\mathbf{u}(k+1) &= [-\mathbf{K}_2\mathbf{\Phi} - \mathbf{K}_1\mathbf{C\Phi}]\mathbf{x}(k) + [-\mathbf{K}_2\mathbf{\Gamma} - \mathbf{K}_1\mathbf{C\Gamma}]\mathbf{u}(k) \\ &\quad + \mathbf{K}_1\mathbf{v}(k) + \mathbf{K}_1\mathbf{r}(k+1) \end{aligned} \tag{D.7}$$

from (D.5) we get

$$\mathbf{K}_1\mathbf{v}(k) = \mathbf{u}(k) + \mathbf{K}_2\mathbf{x}(k) \tag{D.8}$$

Substitute (D.8) into (D.7)

$$\begin{aligned} \mathbf{u}(k+1) &= [\mathbf{I}_m + \mathbf{K}_1\mathbf{D}]^{-1}[\mathbf{K}_2 - \mathbf{K_2\Phi} - \mathbf{K}_1\mathbf{C\Phi}]\mathbf{x}(k) \\ &+ [\mathbf{I}_m + \mathbf{K}_1\mathbf{D}]^{-1}[\mathbf{I}_m - \mathbf{K}_2\mathbf{\Gamma} - \mathbf{K}_1\mathbf{C\Gamma}]\mathbf{u}(k) \\ &+ [\mathbf{I}_m + \mathbf{K}_1\mathbf{D}]^{-1}\mathbf{K}_1\mathbf{r}(k+1) \end{aligned} \tag{D.9}$$

where $\mathbf{I}_m$ is a unit matrix of dimension m. We may now define a new state vector

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix}$$

Using the following three definitions:

$$\begin{aligned} \mathbf{L} &= (\mathbf{I}_m + \mathbf{K}_1\mathbf{D})^{-1}(\mathbf{K}_2 - \mathbf{K}_2\mathbf{\Phi} - \mathbf{K}_1\mathbf{C}\mathbf{\Phi}) \\ \mathbf{M} &= (\mathbf{I}_m + \mathbf{K}_1\mathbf{D})^{-1}(\mathbf{I}_m - \mathbf{K}_2\mathbf{\Gamma} - \mathbf{K}_1\mathbf{C}\mathbf{\Gamma}) \\ \mathbf{N} &= (\mathbf{I}_m + \mathbf{K}_1\mathbf{D})^{-1}\mathbf{K}_1 \end{aligned} \tag{D.10}$$

we define a new state equation describing the complete closed loop system from (D.1) and (D.9)

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{N} \end{bmatrix} \mathbf{r}(k+1) \tag{D.11}$$

and output equation

$$\begin{bmatrix} \mathbf{y}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} \tag{D.12}$$

Assume a constant step input vector $\mathbf{r}(k) = \mathbf{r}$. Then (D.11) becomes

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{N}\mathbf{r} \end{bmatrix} \tag{D.13}$$

From (D.3) follows $\mathbf{v}(k) = \mathbf{v}(k-1) + \mathbf{r}(k) - \mathbf{y}(k)$.
Now as $k \to \infty$, $\mathbf{v}(\infty) \to 0$, it follows that $\mathbf{r}(\infty) \to \mathbf{r}$ constant.

$$\mathbf{v}(\infty) = \mathbf{v}(\infty) + \mathbf{r} - \mathbf{y}(\infty) \Longrightarrow \mathbf{y}(\infty) \to \mathbf{r}$$

We therefore get no steady state error. Equation (D.13) subsequently becomes

$$\begin{bmatrix} \mathbf{x}(\infty) \\ \mathbf{u}(\infty) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}(\infty) \\ \mathbf{u}(\infty) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{Nr} \end{bmatrix} \tag{D.14}$$

Define error vectors as

$$\mathbf{x}_e(k) = \mathbf{x}(k) - \mathbf{x}(\infty) \tag{D.15}$$

$$\mathbf{u}_e(k) = \mathbf{u}(k) - \mathbf{u}(\infty) \tag{D.16}$$

Subtracting (D.14) from (D.13) gives

$$\begin{bmatrix} \mathbf{x}_e(k+1) \\ \mathbf{u}_e(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} \tag{D.17}$$

which we modify to read

$$\begin{bmatrix} \mathbf{x}_e(k+1) \\ \mathbf{u}_e(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \mathbf{w}(k) \tag{D.18}$$

where

$$\begin{bmatrix} \mathbf{w}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} \tag{D.19}$$

If we define

$$\xi(k) = \begin{bmatrix} \mathbf{x}_e(k) \\ \mathbf{u}_e(k) \end{bmatrix} \tag{D.20}$$

$$\hat{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{D.21}$$

$$\hat{\mathbf{\Gamma}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \tag{D.22}$$

and

$$\hat{\mathbf{K}} = -\begin{bmatrix} \mathbf{L} & \mathbf{M} \end{bmatrix} \tag{D.23}$$

Then we may write a new state equation

$$\xi(k+1) = \hat{\mathbf{\Phi}}\xi(k) + \hat{\mathbf{\Gamma}}\mathbf{w}(k) \tag{D.24}$$

with control vector from (D.19) given by

$$\mathbf{w}(k) = -\hat{\mathbf{K}}\xi(k) \tag{D.25}$$

We now want to determine the optimal state feedback gain of our new system

$$\begin{aligned} \xi(k+1) &= \hat{\mathbf{\Phi}}\xi(k) + \hat{\mathbf{\Gamma}}\mathbf{w}(k) \\ &= [\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}]\xi(k) \end{aligned} \tag{D.26}$$

with control vector

$$\mathbf{w}(k) = -\hat{\mathbf{K}}\xi(k) \tag{D.27}$$

The state feedback gain matrix is given by $\hat{\mathbf{K}}$
We therefore wish to minimize the performance index

$$J = \frac{1}{2}\sum_{k=0}^{\infty}[\xi^T(k)\hat{\mathbf{Q}}\xi(k) + \mathbf{w}^T(k)\mathbf{R}\mathbf{w}(k)] \tag{D.28}$$

by the proper choice of $\hat{\mathbf{K}}$. By substituting (D.27) into (D.28) we get

$$\begin{aligned} J &= \frac{1}{2}\sum_{k=0}^{\infty}[\xi^T(k)\hat{\mathbf{Q}}\xi(k) + \xi^T(k)\hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}}\xi(k)] \\ &= \frac{1}{2}\sum_{k=0}^{\infty}[\xi^T(k)(\hat{\mathbf{Q}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}})\xi(k)] \end{aligned} \tag{D.29}$$

Following the Lyapunov approach, we set

$$\xi^T(k)(\hat{\mathbf{Q}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}})\xi(k) = -[\xi^T(k+1)\hat{\mathbf{P}}\xi(k+1) - \xi^T(k)\hat{\mathbf{P}}\xi(k)] \quad \text{(D.30)}$$

Lyapunov's second method says that for a stable $[\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}]$ matrix there exists a positive definite matrix $\hat{\mathbf{P}}$ which satisfies equation (D.30). Using (D.26), (D.30) may be rewritten as follows:

$$\xi(k+1) = [\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}]\xi(k)$$

Using $(AB)^T = B^T A^T$,

$$\xi^T(k+1) = \xi^T(k)[\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}]^T$$

Now (D.30) becomes

$$\xi^T(k)(\hat{\mathbf{Q}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}})\xi(k) = -[\xi^T(k)(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}})\xi(k) - \xi^T(k)\hat{\mathbf{P}}\xi(k)]$$

or

$$\xi^T(k)(\hat{\mathbf{Q}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}})\xi(k) = -\xi^T(k)[(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}) - \hat{\mathbf{P}}]\xi(k) \quad \text{(D.31)}$$

Comparing both sides of (D.31), which must hold true for all $\xi(k)$, we require that

$$\hat{\mathbf{Q}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}} = -(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}) + \hat{\mathbf{P}} \quad \text{(D.32)}$$

This may also be written as

$$\hat{\mathbf{Q}} - \hat{\mathbf{P}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}} + (\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}(\hat{\mathbf{\Phi}} - \hat{\mathbf{\Gamma}}\hat{\mathbf{K}}) = \underline{0}$$

$$\hat{\mathbf{Q}} - \hat{\mathbf{P}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}} + \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} + (\hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}(\hat{\mathbf{\Gamma}}\hat{\mathbf{K}}) - (\hat{\mathbf{\Gamma}}\hat{\mathbf{K}})^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}}\hat{\mathbf{K}} = \underline{0}$$

$$\hat{\mathbf{Q}} + \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} - \hat{\mathbf{P}} + \hat{\mathbf{K}}^T\mathbf{R}\hat{\mathbf{K}} + \hat{\mathbf{K}}^T(\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}})\hat{\mathbf{K}} - \hat{\mathbf{K}}^T\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} - \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}}\hat{\mathbf{K}} = \underline{0}$$

$$\underbrace{\hat{\mathbf{Q}} + \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} - \hat{\mathbf{P}}}_{\hat{A}} + \hat{\mathbf{K}}^T\underbrace{(\mathbf{R} + \mathbf{\Gamma}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}})}_{\hat{B}}\hat{\mathbf{K}} - (\hat{\mathbf{K}}^T\underbrace{\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}}}_{\hat{C}} + \underbrace{\hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}}}_{\hat{D}}\hat{\mathbf{K}}) = \underline{0} \quad \text{(D.33)}$$

Defining $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ as indicated above, the left hand side of the above equation must be minimized with respect to $\hat{\mathbf{K}}$. Equation (D.33) may be written as

$$\hat{A} + \hat{\mathbf{K}}^T\hat{B}\hat{\mathbf{K}} - (\hat{\mathbf{K}}^T\hat{C} + \hat{D}\hat{\mathbf{K}}) = \underline{\mathbf{0}} \tag{D.34}$$

which may also be written as

$$\hat{A} + [\hat{B}^{\frac{1}{2}}\hat{\mathbf{K}} - \hat{B}^{-\frac{1}{2}}\hat{C}]^T[\hat{B}^{\frac{1}{2}}\hat{\mathbf{K}} - \hat{B}^{-\frac{1}{2}}\hat{C}] - \hat{D}\hat{B}^{-1}\hat{C} = \underline{\mathbf{0}} \tag{D.35}$$

Since the braced terms are always non-negative, the minimum occurs when they are zero, or

$$\hat{B}^{\frac{1}{2}}\hat{\mathbf{K}} = \hat{B}^{-\frac{1}{2}}\hat{C}$$

or

$$\hat{\mathbf{K}} = \hat{B}^{-1}\hat{C} \tag{D.36}$$

Substituting (D.36) into (D.34) gives

$$\begin{aligned}
& \hat{A} + (\hat{B}^{-1}\hat{C})^T\hat{B}(\hat{B}^{-1}\hat{C}) - (\hat{B}^{-1}\hat{C})^T\hat{C} - \hat{D}(\hat{B}^{-1}\hat{C}) = \underline{\mathbf{0}} \\
& \hat{A} + (\hat{B}^{-1}\hat{C})^T\hat{C} - (\hat{B}^{-1}\hat{C})^T\hat{C} - \hat{D}(\hat{B}^{-1}\hat{C}) = \underline{\mathbf{0}} \\
\Longrightarrow \quad & \hat{A} - \hat{D}\hat{B}^{-1}\hat{C} = \underline{\mathbf{0}}
\end{aligned} \tag{D.37}$$

Back substituting the definitions of $\hat{A}, \hat{B}, \hat{C}, \hat{D}$, (D.36) becomes

$$\hat{\mathbf{K}} = (\mathbf{R} + \hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}})^{-1}\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} \tag{D.38}$$

and (D.37) becomes

$$\hat{\mathbf{Q}} + \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} - \hat{\mathbf{P}} - \hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}}(\mathbf{R} + \hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}})^{-1}\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} = \underline{\mathbf{0}} \tag{D.39}$$

The above equation is known as the discrete matrix Ricatti equation. Solution of the Ricatti equation gives $\hat{\mathbf{P}}$, which may be used in (D.38) to obtain the optimal state feedback gain matrix $\hat{\mathbf{K}}$.

However from equation (D.21) we see that $\hat{\mathbf{\Phi}}$ is singular, and the above form of the Ricatti equation cannot be solved directly. We therefore require a different formulation for the Ricatti equation.

For $\hat{\mathbf{\Phi}}$ singular, we proceed as follows: Define $\hat{\mathbf{P}}$ as

$$\hat{\mathbf{P}} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix} \tag{D.40}$$

Note that $\mathbf{P}_{21} = \mathbf{P}_{12}^T$. Now

$$\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{D.41}$$

and from (D.21) and (D.22) we have

$$\hat{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ 0 & 0 \end{bmatrix} \qquad \hat{\mathbf{\Phi}}^T = \begin{bmatrix} \mathbf{\Phi}^T & \mathbf{0} \\ \mathbf{\Gamma}^T & \mathbf{0} \end{bmatrix} \tag{D.42}$$

$$\hat{\mathbf{\Gamma}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_m \end{bmatrix} \qquad \hat{\mathbf{\Gamma}}^T = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix} \tag{D.43}$$

Second Term

The second term of the Ricatti equation (D.39) becomes

$$\hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi}^T & \mathbf{0} \\ \mathbf{\Gamma}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{\Phi}^T\mathbf{P}_{11}\mathbf{\Phi} & \mathbf{\Phi}^T\mathbf{P}_{11}\mathbf{\Gamma} \\ \mathbf{\Gamma}^T\mathbf{P}_{11}\mathbf{\Phi} & \mathbf{\Gamma}^T\mathbf{P}_{11}\mathbf{\Gamma} \end{bmatrix} \tag{D.44}$$

Fourth Term

The fourth term of equation (D.39) becomes

$$\hat{\mathbf{\Phi}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}}(\mathbf{R} + \hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Gamma}})^{-1}\hat{\mathbf{\Gamma}}^T\hat{\mathbf{P}}\hat{\mathbf{\Phi}}$$

$$= \begin{bmatrix} \mathbf{\Phi}^T & 0 \\ \mathbf{\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{I}_m \end{bmatrix} \left(\mathbf{R} + \begin{bmatrix} 0 & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{I}_m \end{bmatrix} \right)^{-1} *$$

$$\begin{bmatrix} 0 & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{\Phi}^T \mathbf{P}_{12}(\mathbf{R} + \mathbf{P}_{22})^{-1} \\ \mathbf{\Gamma}^T \mathbf{P}_{12}(\mathbf{R} + \mathbf{P}_{22})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{12}^T \mathbf{\Phi} & \mathbf{P}_{12}^T \mathbf{\Gamma} \end{bmatrix} \tag{D.45}$$

Defining ¶ as

$$¶ = \mathbf{P}_{12}(\mathbf{R} + \mathbf{P}_{22})^{-1} \mathbf{P}_{12}^T \tag{D.46}$$

the fourth term of (D.39) simplifies to

$$= \begin{bmatrix} \mathbf{\Phi}^T ¶ \mathbf{\Phi} & \mathbf{\Phi}^T ¶ \mathbf{\Gamma} \\ \mathbf{\Gamma}^T ¶ \mathbf{\Phi} & \mathbf{\Gamma}^T ¶ \mathbf{\Gamma} \end{bmatrix} \tag{D.47}$$

Ricatti Equation

Having found alternative expressions for the four terms of equation (D.39), we may substitude these expressions – (D.41), (D.44), (D.40), (D.47), into the Ricatti equation (D.39)

$$\begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}^T \mathbf{P}_{11} \mathbf{\Phi} & \mathbf{\Phi}^T \mathbf{P}_{11} \mathbf{\Gamma} \\ \mathbf{\Gamma}^T \mathbf{P}_{11} \mathbf{\Phi} & \mathbf{\Gamma}^T \mathbf{P}_{11} \mathbf{\Gamma} \end{bmatrix} - \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix}$$

$$- \begin{bmatrix} \mathbf{\Phi}^T ¶ \mathbf{\Phi} & \mathbf{\Phi}^T ¶ \mathbf{\Gamma} \\ \mathbf{\Gamma}^T ¶ \mathbf{\Phi} & \mathbf{\Gamma}^T ¶ \mathbf{\Gamma} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Comparing matrix elements gives the following three equations:

$$\begin{aligned}
\mathbf{Q} + \mathbf{\Phi}^T\mathbf{P}_{11}\mathbf{\Phi} - \mathbf{P}_{11} - \mathbf{\Phi}^T\P\mathbf{\Phi} &= \mathbf{0} \\
\mathbf{\Phi}^T\mathbf{P}_{11}\mathbf{\Gamma} - \mathbf{P}_{12} - \mathbf{\Phi}^T\P\mathbf{\Gamma} &= \mathbf{0} \\
\mathbf{\Gamma}^T\mathbf{P}_{11}\mathbf{\Gamma} - \mathbf{P}_{22} - \mathbf{\Gamma}^T\P\mathbf{\Gamma} &= \mathbf{0}
\end{aligned}$$

Defining

$$\begin{aligned}
\mathbf{P} &= \mathbf{P}_{11} - \P \\
&= \mathbf{P}_{11} - \mathbf{P}_{12}(\mathbf{R} + \mathbf{P}_{22})^{-1}\mathbf{P}_{12}^T, \qquad \text{(D.48)}
\end{aligned}$$

the above three equations become

$$\begin{aligned}
\mathbf{P}_{11} &= \mathbf{Q} + \mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} \qquad \text{(D.49)} \\
\mathbf{P}_{12} &= \mathbf{\Phi}^T\mathbf{P}\mathbf{\Gamma} \qquad \text{(D.50)} \\
\mathbf{P}_{22} &= \mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma} \qquad \text{(D.51)}
\end{aligned}$$

Substituting these matrix elements into the definition of **P** from (D.48), gives

$$\mathbf{Q} + \mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{\Phi}^T\mathbf{P}\mathbf{\Gamma}(\mathbf{R} + \mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma})^{-1}(\mathbf{\Phi}^T\mathbf{P}\mathbf{\Gamma})^T = \mathbf{P}$$

or

$$\mathbf{Q} + \mathbf{\Phi}^T\mathbf{P}\mathbf{\Phi} - \mathbf{\Phi}^T\mathbf{P}\mathbf{\Gamma}(\mathbf{R} + \mathbf{\Gamma}^T\mathbf{P}\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{P}\mathbf{\Phi} = \mathbf{P} \qquad \text{(D.52)}$$

which is the final Ricatti equation to solve for **P** .

Optimal Gain Matrices $\mathbf{K}_1$ and $\mathbf{K}_2$

Having obtained the optimal state feedback gain matrix $\hat{\mathbf{K}}$ through solution of the Ricatti equation from (D.38), we can use $\hat{\mathbf{K}}$ to determine $\mathbf{K}_1$ and $\mathbf{K}_2$ as follows:
Recall the following definitions from (D.10)

$$\begin{aligned}
\mathbf{L} &= (\mathbf{I}_m + \mathbf{K}_1\mathbf{D})^{-1}(\mathbf{K}_2 - \mathbf{K}_2\mathbf{\Phi} - \mathbf{K}_1\mathbf{C}\mathbf{\Phi}) \\
\mathbf{M} &= (\mathbf{I}_m + \mathbf{K}_1\mathbf{D})^{-1}(\mathbf{I}_m - \mathbf{K}_2\mathbf{\Gamma} - \mathbf{K}_1\mathbf{C}\mathbf{\Gamma})
\end{aligned}$$

Substituting the above two equations into the optimal state feedback gain matrix (D.23)

$$\hat{\mathbf{K}} = -\begin{bmatrix} \mathbf{L} & \mathbf{M} \end{bmatrix}$$

we may write

$$\begin{aligned}(\mathbf{I}_m + \mathbf{K}_1\mathbf{D})\hat{\mathbf{K}} &= \begin{bmatrix} (\mathbf{K}_2\mathbf{\Phi} - \mathbf{K}_2 + \mathbf{K}_1\mathbf{C}\mathbf{\Phi}) & (\mathbf{K}_2\mathbf{\Gamma} + \mathbf{K}_1\mathbf{C}\mathbf{\Gamma} - \mathbf{I}_m) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_2 & \mathbf{K}_1 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi} - \mathbf{I}_n & \mathbf{\Gamma} \\ \mathbf{C}\mathbf{\Phi} & \mathbf{C}\mathbf{\Gamma} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\mathbf{I}_m \end{bmatrix}\end{aligned}$$

(Note that $\mathbf{I}_m$ is a unit matrix of dimension m, which is the number of inputs or outputs, whereas $\mathbf{I}_n$ is a unit matrix of dimension n, which is the number of states.)

$$\mathbf{I}_m\hat{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_2 & \mathbf{K}_1 \end{bmatrix} \begin{bmatrix} \mathbf{\Phi} - \mathbf{I}_n & \mathbf{\Gamma} \\ \mathbf{C}\mathbf{\Phi} & \mathbf{C}\mathbf{\Gamma} \end{bmatrix} - \mathbf{K}_1\mathbf{D}\hat{\mathbf{K}} + \begin{bmatrix} \mathbf{0} & -\mathbf{I}_m \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{K}_2 & \mathbf{K}_1 \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{\Phi} - \mathbf{I}_n & \mathbf{\Gamma} \\ \mathbf{C}\mathbf{\Phi} & \mathbf{C}\mathbf{\Gamma} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{D}\hat{\mathbf{K}} \end{bmatrix} \right\} = \mathbf{I}_m\hat{\mathbf{K}} + \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix}$$

or

$$\begin{bmatrix} \mathbf{K}_2 & \mathbf{K}_1 \end{bmatrix} = \left(\mathbf{I}_m\hat{\mathbf{K}} + \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix} \right) \left\{ \begin{bmatrix} \mathbf{\Phi} - \mathbf{I}_n & \mathbf{\Gamma} \\ \mathbf{C}\mathbf{\Phi} & \mathbf{C}\mathbf{\Gamma} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{D}\hat{\mathbf{K}} \end{bmatrix} \right\}^{-1} \quad \text{(D.53)}$$

Steps in Applying the Optimal Servo-system

Having derived the necessary equations for the optimal servo-controller, we shall briefly describe how it is put to use in obtaining the system inputs from the desired system responses.

- Step 1: Solve for $\mathbf{P}$ from the Ricatti equation (D.52) for the plant only

- Step 2: Calculate the elements $\mathbf{P}_{11}, \mathbf{P}_{12}, \mathbf{P}_{22}$, from (D.49) through (D.51)

- Step 3: Set up matrix $\hat{\mathbf{P}}$ as in (D.40)

- Step 4: Set up matrices $\hat{\mathbf{\Phi}}$ and $\hat{\mathbf{\Gamma}}$ as in (D.21) and (D.22)

- Step 5: Determine the optimal state feedback gain matrix $\hat{\mathbf{K}}$ for the system $\xi(k+1) = \hat{\mathbf{\Phi}}\xi(k) + \hat{\mathbf{\Gamma}}\mathbf{w}(k)$; $\mathbf{w}(k) = -\hat{\mathbf{K}}\xi(k)$ from(D.38)

- Step 6: Calculate $\mathbf{K}_1$ and $\mathbf{K}_2$ from $\hat{\mathbf{K}}$ as in (D.53)

- Step 7: Calculate the system inputs $\mathbf{u}(k)$ by simulation of the system (D.11) using $\mathbf{r}(k)$ as input

Appendix E

Test rig photographs

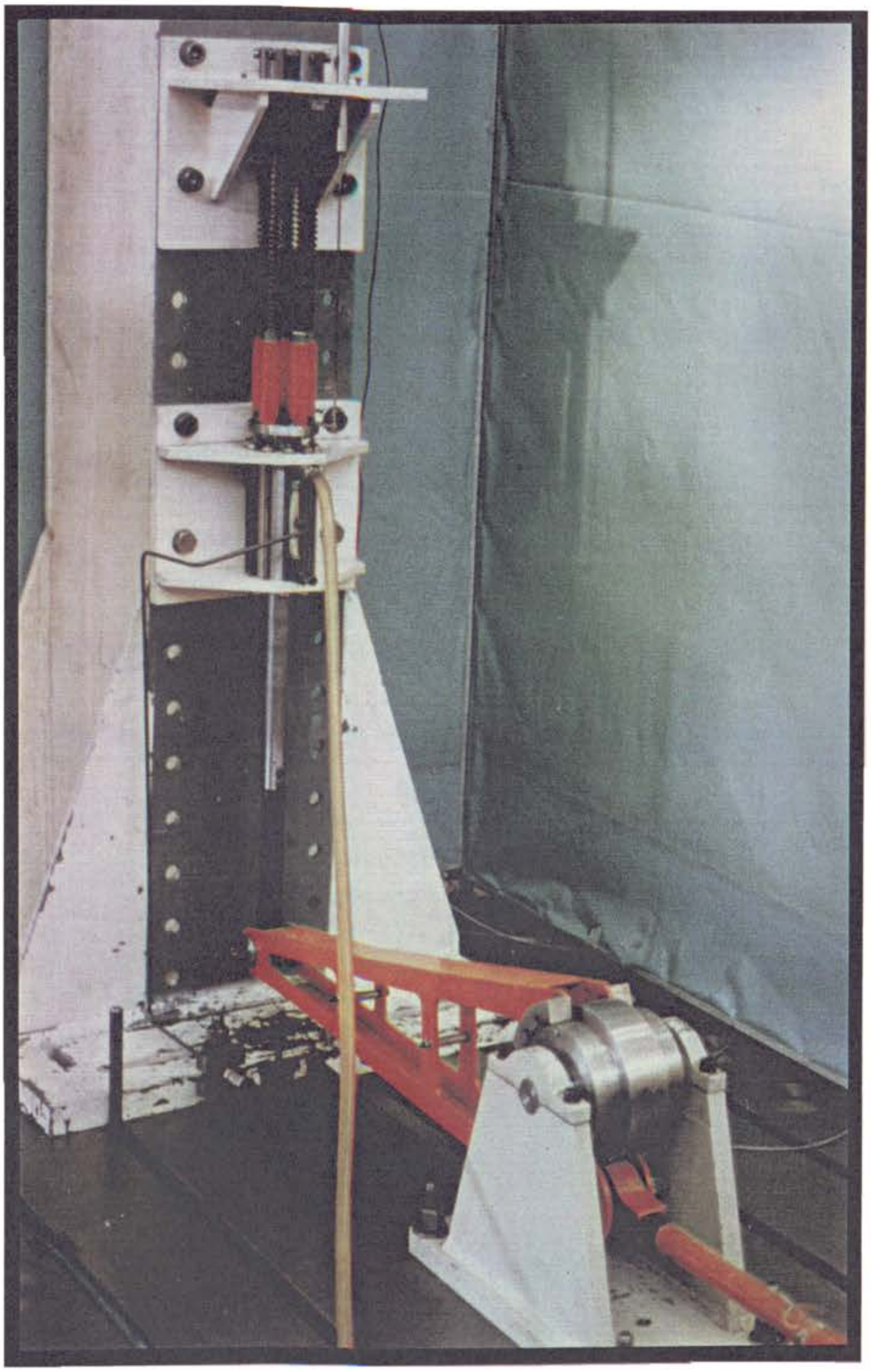


Figure E.1: High speed spring loading simulator global view (Case study No.1)

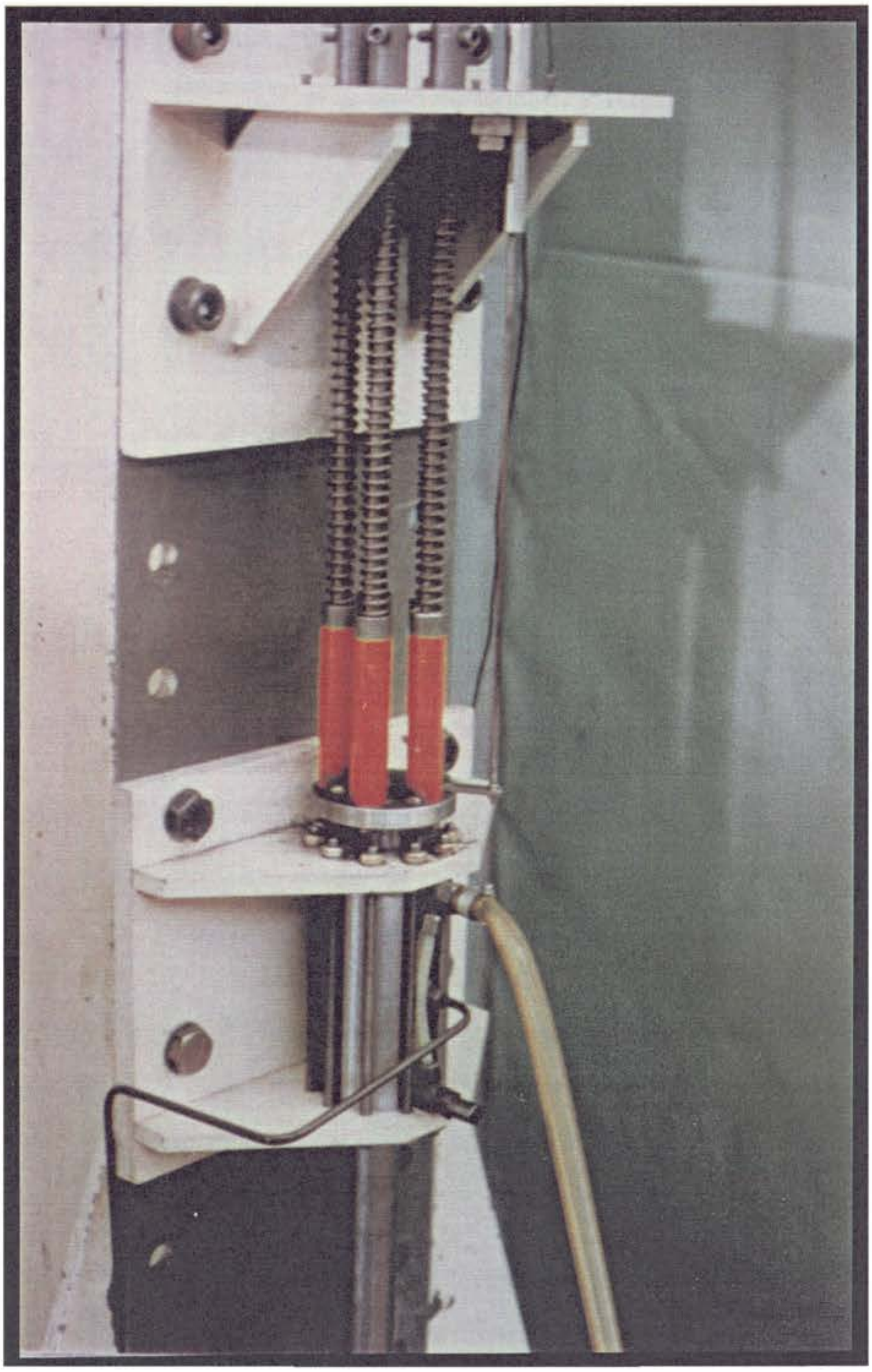


Figure E.2: Detail view on high speed spring loading simulator (Case study No.1)

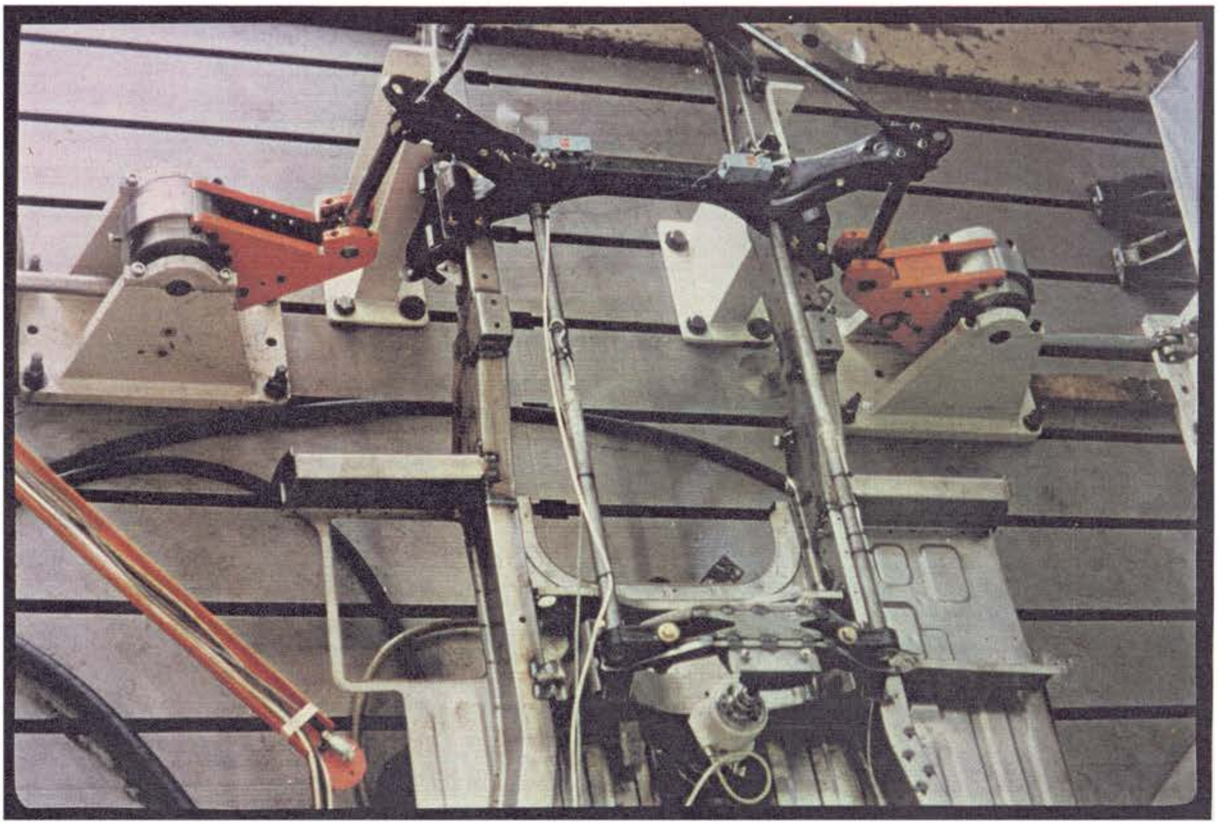


Figure E.3: Three actuator vehicle suspension rig (Case study No.2)

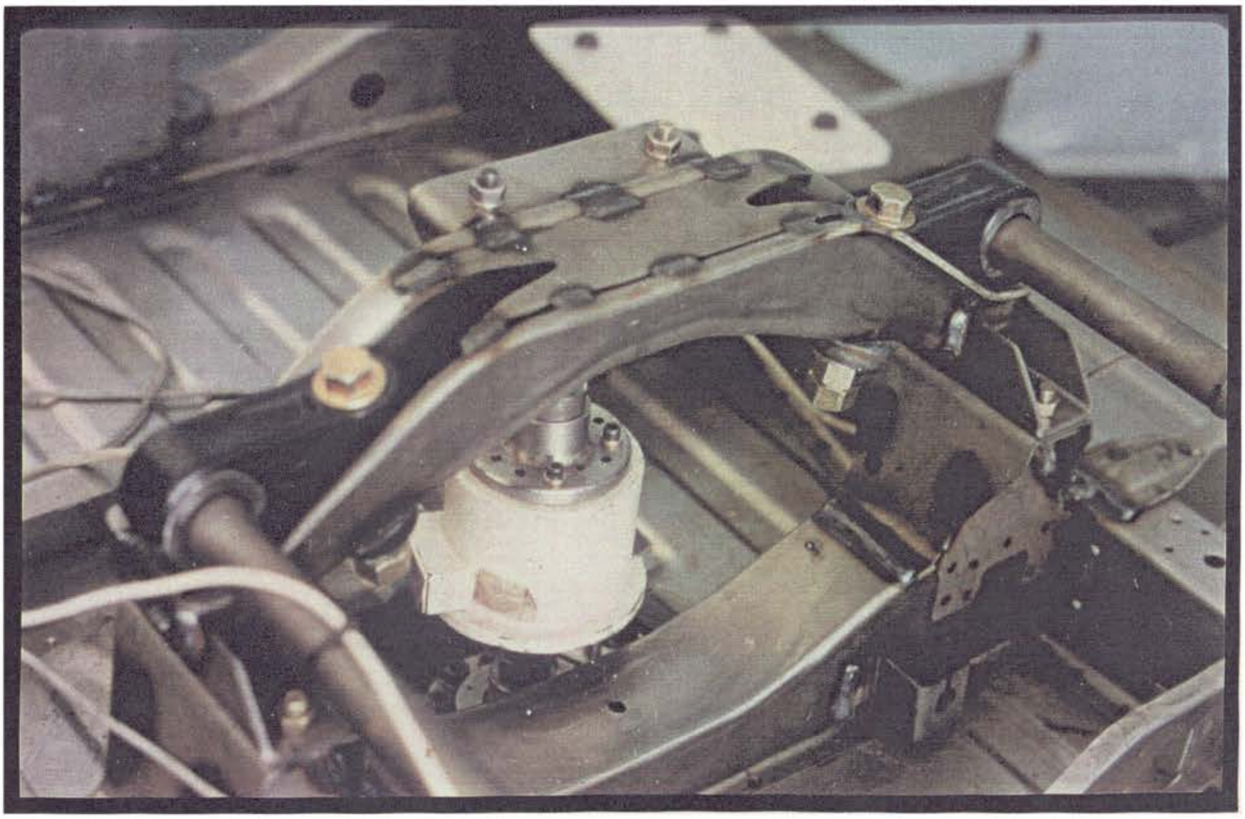


Figure E.4: Detail view on crossmember (Case study No.2)

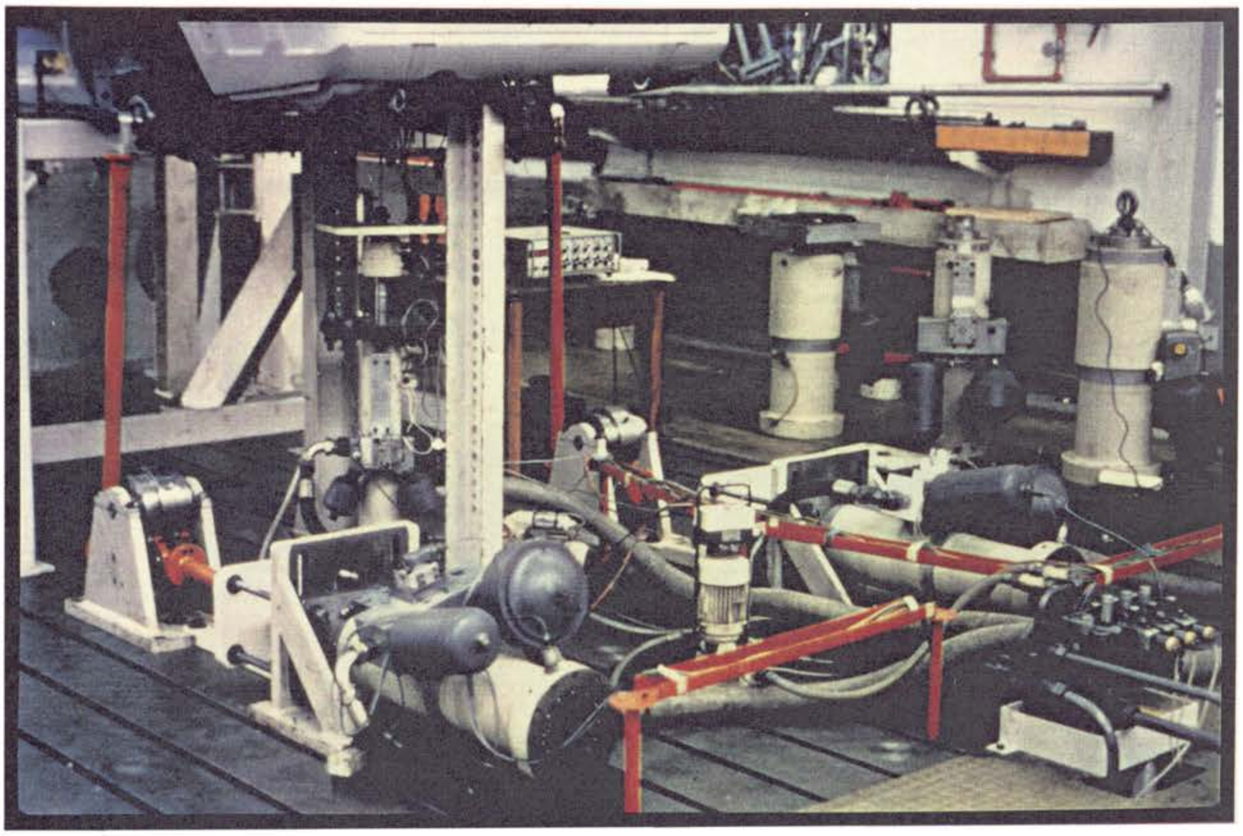


Figure E.5: Four actuator passenger vehicle rig (Case study No.3)

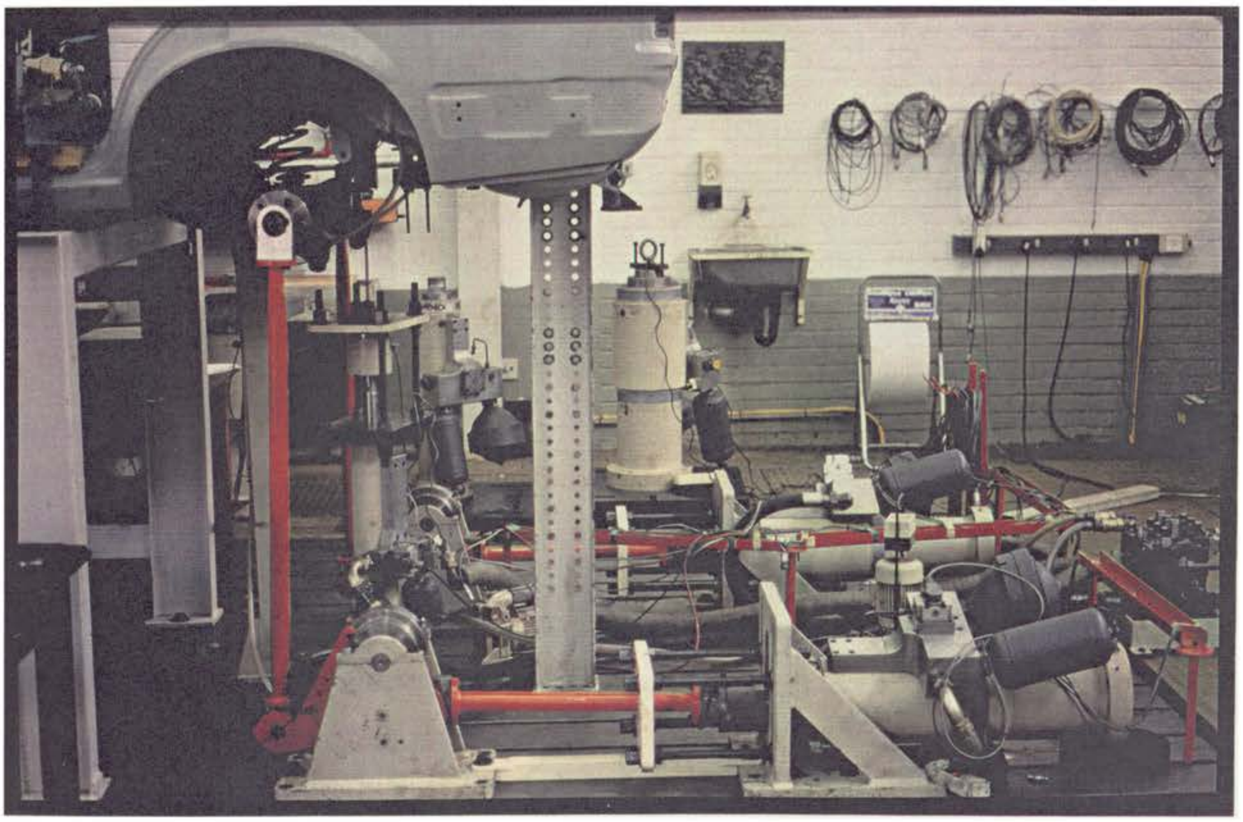


Figure E.6: Side view on passenger vehicle rig (Case study No.3)

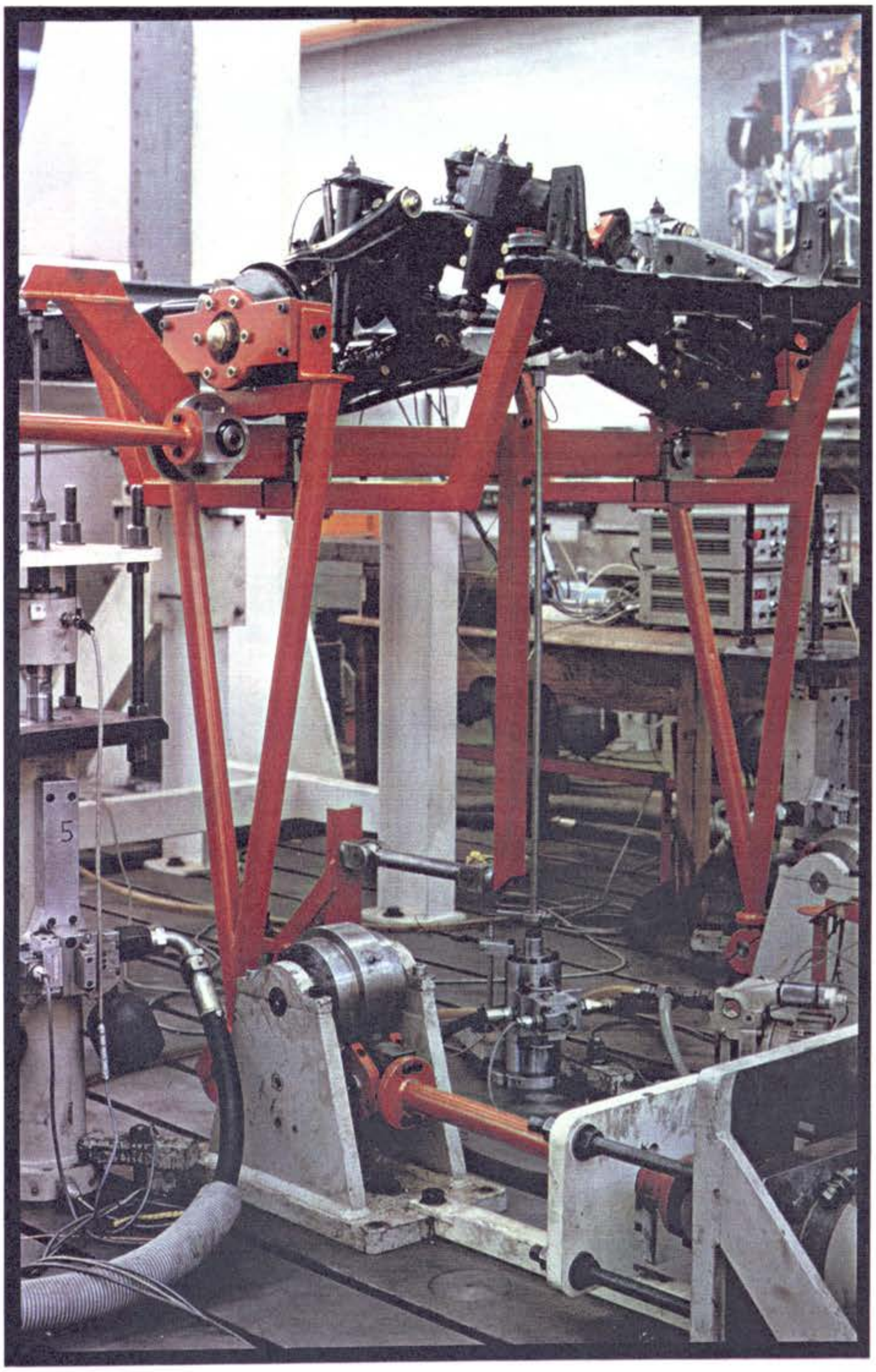


Figure E.7: Pick-up truck chassis rig - front view (Case study No.4)

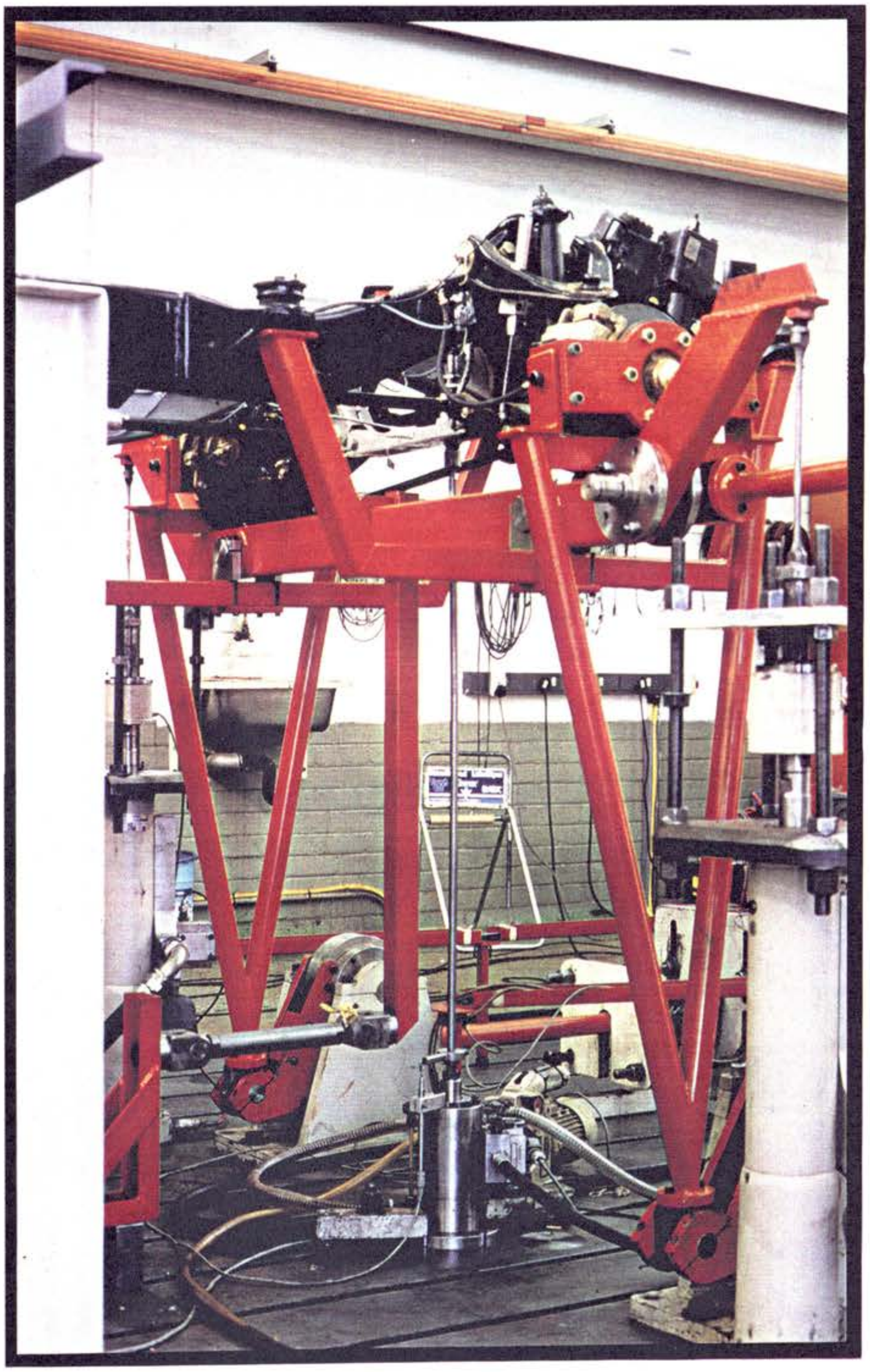


Figure E.8: View from right rear side on chassis rig (Case study No.4)

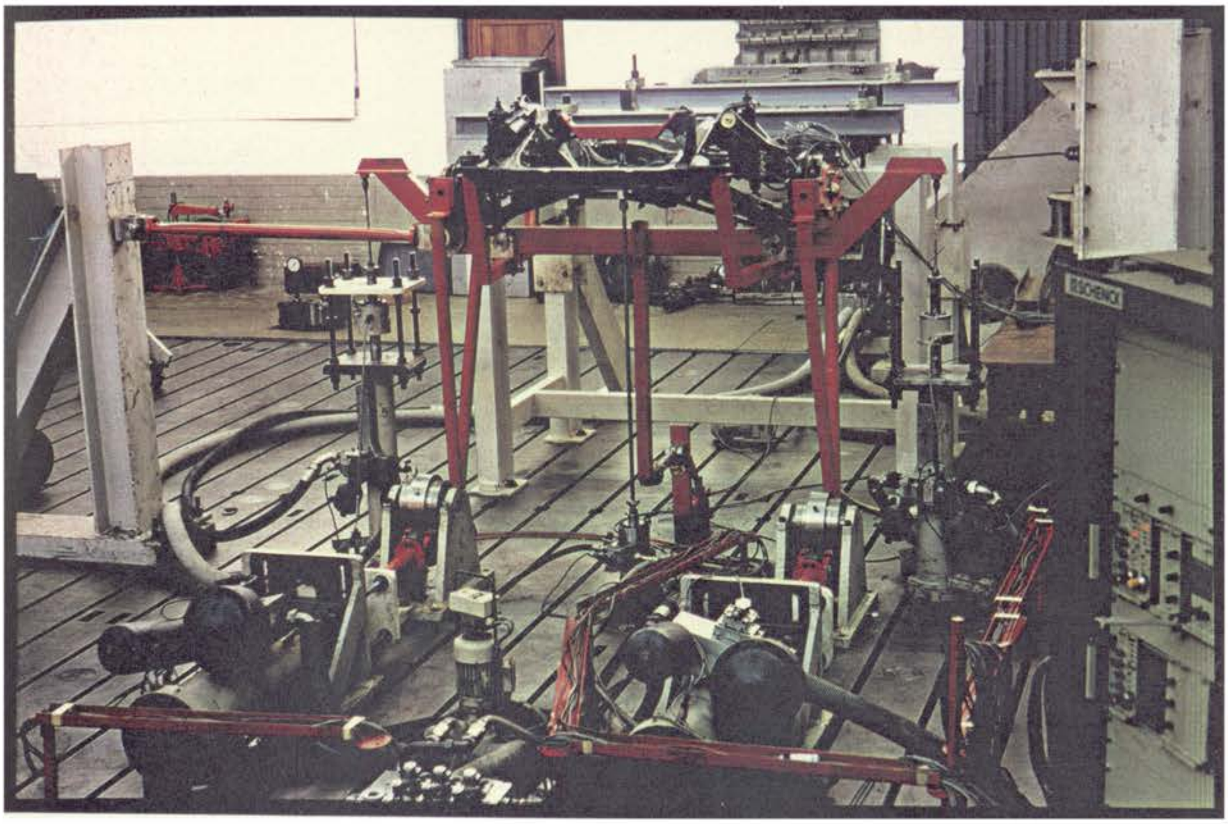


Figure E.9: Global view on chassis rig (Case study No.4)

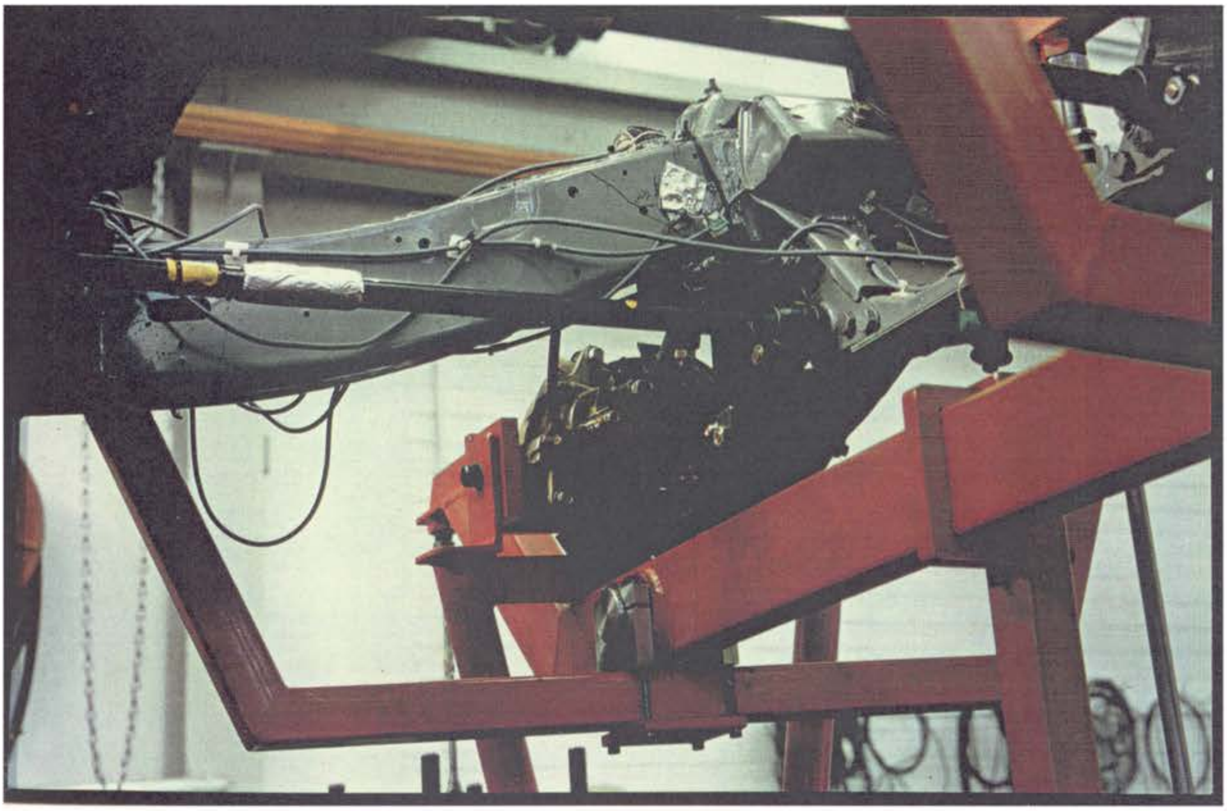


Figure E.10: Detail view on torsion bar and chassis beam strain gauges (Case study No.4)