SYSTEM IDENTIFICATION
&
PARAMETER ESTIMATION
OF A
MOTORCYCLE SUSPENSION
SYSTEM

By

Jim Ledwidge, B.A.I., B.A.

A thesis submitted to

Dublin City University

for the degree of

Master of Engineering

School of Electronic Engineering

DUBLIN CITY UNIVERSITY

August 1995

Abstract

The use of modern estimation theory to track the time-varying dynamics of a motorcycle suspension system is investigated. It is shown using software simulations and a test rig comprised of an analogue computer that a modified recursive least squares algorithm which incorporates a fault detection scheme is a suitable approach.

The fault detection scheme incorporated in the modified algorithm detects system changes by monitoring the statistical properties of the estimators prediction error sequence.

A mass, spring and damper model is chosen as a simple model with which to represent the dynamics of the suspension system. Initial estimates of the suspension parameters are found from static tests performed on a sample "Formula 1" front suspension unit. These estimates are incorporated into a software simulation and this simulation is used as a platform to examine the performance of existing parameter estimation techniques and also to develop the modified least squares algorithm.

An analogue computer is built as a hardware representation of the system dynamics. The modified algorithm is fine tuned using equivalent force and displacement data from this computer. The modified algorithm is used to track the parameters of this computer.

The resulting parameter estimates indicate that the modified algorithm yields estimates which have superior variance and convergence properties than those of estimates obtained using more conventional parameter estimation schemes.

Declaration

I hereby certify that this material, which I now submit for assement on the programme of study leading to the award of Master of Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ____________________

Date: 4-9-95

Acknowledgements

I wish to express my gratitude to my supervisor Dr. Aengus Murray for his much needed advice and encouragement throughout this project. A special thank you must go to Mr. Jim Dowling for taking on the onerous and I'm sure tedious business of examining this thesis at such a late stage.

I must thank Tom O' Kane of *Team Kenny Roberts* for his technical assistance and his insight into the world of Grand Prix motorcycling. I must also thank Ian Hooper of the mechanical engineering workshop for his help with some of the mechanical aspects of this work.

Many thanks to Conor, Liam, Paul, Dave and all of the technical staff in electronic engineering, also Liam and Prash from the Mechanical Engineering department.

Thanks to my fellow postgrads in the Power Electronics Laboratory for their kind support, suggestions and wit. It was my good fortune to work with such a fine group of lads. I would like to thank them and also Morgan, Liam *et al* for their support in times of extreme inebriation.

I would like to thank Power Electronics Ireland Ltd., a division of Forbairt for their financial backing and finally my family whose encouragement contributed greatly to my further education.

CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 SYSTEM ANALYSIS AND MODELLING

2.1 Introduction 5

2.2 System description 5

2.2.1 Stiffness 5

2.2.2 Damping 6

2.2.3 Suspension geometry 7

2.3 The suspension problem 8

2.3.1 The conflicting requirements of high speed motorcycles 8

2.3.1 Bad engineering design 9

2.4 System modelling 11

2.5 Summary 14

CHAPTER 3 SYSTEM IDENTIFICATION & PARAMETER ESTIMATION

3.1 Introduction 15

3.2 System identification 15

3.3 Classical identification 17

3.3.1 Time domain methods 17

3.3.2 Frequency domain methods 19

3.4 Parametric identification 19

3.4.1 System modelling 20

3.4.2 Parameter estimaton 24

3.5 The least squares estimate 25

3.6 The forgetting factor algorithm 29

3.7 The Kalman filter 31

3.7.1 State estimation 31

3.7.2 The Kalman filter as a parameter estimator 35

3.8 Fault detection 38

3.9 Summary 41

CHAPTER 4 SYSTEM SIMULATION

4.1 Introduction 42

4.2 System simulation 42

4.3 Model estimation 43

4.4 System tracking 46

4.4.1 Parameter tracking using the R.L.S. algorithm 46

4.4.2 Parameter estimation using the alternative tracking scheme 52

4.4.3 Parameter estimation using an improved alternative algorithm 54

4.5 Summary 57

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Introduction 59

5.2 An electrical analogue 59

5.3 Implementation of the electrical analogue 62

5.4 Identification/Estimation of the electrical analogue 63

5.5 Tracking the electrical analogue 65

5.6 Algorithm performance 66

5.7 Summary 67

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary and conclusions 71

6.2 Recommendations 72

6.2.1 Modifications to the current work 72

6.2.1.1 Modelling 72

6.2.1.2 Fault detection 73

6.2.2 Further research 73

6.2.2.1 Test rigs and on-site testing 73

BIBLIOGRAPHY

APPENDIX A

A.1 Derivation of the least squares estimate

A.2 Derivation of the least squares recursoin

APPENDIX B

B.1 Circuit diagram for the analgue computer

APPENDIX C

C.1 proposed test rig design

Chapter 1

Introduction

A terrain vehicle suspension system is a mechanical, energy dissipating device that lies between the chassis of the vehicle and the ground. Its purpose is firstly to restrict the influence of road irregularities upon passenger comfort and secondly to improve the "handling" ability of the vehicle.

Suspension systems vary in shape and size, these being attributes that are vehicle/purpose dependent. This is nowhere more apparent than in the class of two-wheeled vehicles. For example a child's bicycle must rely on the wheels and forks as an intermediary between itself and the ground whereas a large motorcycle uses lightweight alloy wheels and shock-absorbers. It is this latter form of suspension that was under investigation in work described in this thesis.

It is well known that the suspension set-up of a Grand Prix motorcycle is critical, especially during cornering manoeuvres. The nose-dive [1] experienced when entering a corner and the wobble effect [1] experienced when exiting, stand as visible testament to this.

In the case of such a high performance vehicle the solution to this so-called *suspension problem* [1-3] is non-trivial. The difficulty arises because the optimal set-up of the suspension system for straight-line motion is not necessarily the optimal set-up for cornering manoeuvres. Until recently the tuning of motorcycle suspensions has amounted to varying the set-up between different grades of *soft* and *stiff* [1] attributes that are heavily dependant upon rider feedback rather than scientific observation. This attitude may be acceptable to those who advocate a ride by the seat of your pants philosophy but in the competitive world of Grand Prix motorcycling, with ever increasing speeds and hence risk, a more scientific approach is both desirable and inevitable.

Similar conflicting requirements have been encountered in the suspension systems of Formula 1 (F. 1) motor racing cars. However unlike the motorcycle racing fraternity the

F. 1 establishment has had the significantly large financial backing required to investigate possible solutions to what is one of a large number of problems encountered in vehicle design.

The first in-depth investigation into these suspensions was enacted as a direct result of the introduction of ground effects aerodynamics [3-6] by the Lotus team in the 1978/79 motor-racing season. A ground effects vehicle is shaped so as to maximise the downforce acting on the vehicle resulting in better cornering performance than a non-ground-effects competitor. It soon became apparent that the vehicle had to be constrained to operate at an optimal ride-height [3], [5] in order for this new technology to work effectively.

Because there are large variations in downforce experienced over different tracks (average 1.5kN at Monaco to 32kN at Ricard) suspension systems, in general, became very stiff, minimising chassis displacement thus maintaining the optimal ride height criterion. A side-effect of this was that the vehicle was now susceptible to bumps in the road making it uncomfortable and difficult to handle. It didn't take long for engineers to realise that the best suspension system for such a vehicle would be able to change its dynamics to cater for the large downforces encountered and yet give comfortable ride and handling ability.

This idea gave birth to the concept of active suspension systems. The dynamics of such systems would be controlled by an on-board micro-computer that would determine the optimal set-up from various measurements and induce this set-up on-line with the aid of various hydraulic/mechanical actuators. With this in mind the Lotus team devised a prototype active suspension for the Lotus Turbo Esprit, a production car [3], [7]. This prototype was so successful (an increase of 10% in cornering speeds in a car that had previously exhibited excellent cornering/handling ability) that the system was adapted for the Lotus F. 1 motor racing vehicle.

The subsequent success of active suspension systems as a driver aid has led to their being banned from F.1 motor racing. However there is no reason to suggest that the wealth of knowledge surrounding these systems in four wheeled vehicles [2-5], [7-17] cannot be applied to their single track counterparts.

One of the more popular arrangements of active suspension systems implements self-tuning control (S.T.C), a form of adaptive control, of the suspension unit [10], [12], [22]. This type of control consists of three tasks,

(i) Measurement of suitable suspension data.
(ii) Estimation of the time-varying suspension parameters
(ii) Construction of the correct actuator signals from the measured data, estimated parameters and a given control law.

This thesis details the work carried out in the development of the second task in the S.T.C. algorithm i.e. the estimation and tracking of the time varying suspension parameters. Existing parameter estimation schemes which are appropriate for slowly varying dynamics are discussed. However the objective of the work in this thesis was to produce a scheme to track rapidly varying suspension dynamics. The parameters which describe these dynamics exhibit large and sudden variations. An alternative algorithm which caters for these so-called "jumping parameters" was developed.

The development of this algorithm commenced with the modelling of the suspension unit in a simple parametric form. This model was then simulated in software. This simulation was used as a platform upon which to develop the alternative parameter tracking scheme. An analogue computer was constructed as an electrical representation of the suspension dynamics. The validity of the tracking scheme was verified using data from this analogue computer.

Thesis Structure

The thesis is divided into six chapters. Chapter 1, the introduction, is given as an overview of the problems encountered in suspension systems.

Chapter 2 gives a detailed description of the mechanical structure of modern motorcycle suspension systems. The suspension problem mentioned earlier is discussed in the light of this structure. A simple model for the suspension system is introduced.

A brief summary of system identification and parameter estimation theory is given in chapter 3. The tracking of system parameters is discussed and an alternative tracking algorithm, developed as part of this work is presented.

Chapter 4 introduces a simulation of the suspension system. This is used as a platform upon which to modify the tracking scheme developed in chapter 3.

In chapter 5 the construction of an analogue computer as an electrical analogue for a motorcycle suspension system is detailed. The applicability of the alternative algorithm with respect to suspension systems is verified using data from this electrical analogue. The performance of this algorithm is compared with that of more conventional parameter tracking schemes.

Finally chapter 6 summarises the research work, details any conclusions reached and gives recommendations for any further work relating to this project.

Chapter 2

SYSTEM ANALYSIS AND MODELLING

2.1 INTRODUCTION

This chapter gives a detailed description of the mechanical design of a motorcycle shock absorber/suspension unit. Carrying on in the vein of chapter 1 the suspension problem is reviewed in light of this design and the suspension requirements of a high speed motorcycle. Finally a simple model of the suspension unit is proposed.

2.2 SYSTEM DESCRIPTION

So that an understanding of the suspension problem can be gained it is necessary to give a brief outline of the construction of the system. For simplicity only the "front end" suspension of the bike will be considered.

There are three major properties that influence the characteristics of a suspension system [1]. These are :

(i) Stiffness,
(ii) Damping,
(iii) Suspension geometry.

2.2.1 Stiffness

Stiffness is the basic suspension element and as such was the sole component of the first suspension systems. The stiffness contribution can be achieved in many different ways. Early vehicles used rocker type mechanisms (still visible on rail carriages and trucks today). Leaf springs and torsion bars have been used in the past. The elastic properties of rubber and the compressibility of air or other gases can also be exploited to introduce stiffness. However all of these methods have been superseded (at least in single track vehicles) by coil springs made of steel or, in more expensive/sophisticated models, titanium (see Fig. 2.2).

Coil springs are cleaner than any of the pneumatic struts available and have the advantage of compactness that rockers do not possess. It is also possible to wind these springs to give variable rates of compression if desired, an advantage they have over other solutions. Because of these properties coil springs are used universally in motorcycle suspension design.

2.2.2 Damping

The purpose of damping is to prevent uncontrollable oscillation of the system about its equilibrium point. Dampers work by absorbing the energy of the displaced system, resulting in a smoother ride. The first dampers used friction to dissipate the vehicles' kinetic energy. However these dampers possessed an inherent amount of stiction. As such the suspension motion was hindered. Since a damper requires relative displacement to work efficiently suspensions using this kind of damper became insensitive to the smaller road irregularities. This problem was later tackled by the introduction of hydraulic dampers (see Fig 2.1). They damped the system oscillations but with increased sensitivity and less stiction.

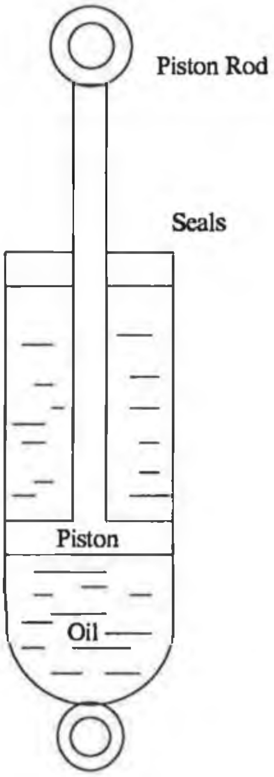


Fig. 2.1 A simple two-way hydraulic suspension damper

There are two forms of hydraulic damping, viscous and hydrodynamic. Viscous damping is caused by the shearing of the damping medium (in this case oil) whereas hydrodynamic damping is due to the mass transfer of the damping medium. Viscous damping, like stiffness, makes a linear contribution to the system dynamics, it being proportional to the velocity, on the other hand hydrodynamic damping exhibits non-linear behaviour as it is proportional to the square of the velocity. Although hydrodynamic damping is mathematically less attractive than its viscous counterpart it is more flexible. It is possible to obtain uni-directional-damping or varying degrees of damping by utilising the hydrodynamic option.

The simplicity of viscous damping and the flexibility of hydrodynamic damping means that they are both in wide use. It is common practice for Grand Prix motorcycle engineers to incorporate both methods of damping in the suspension systems of these motorcycles.

2.2.3 Suspension geometry

The front end suspension of a 500 c.c. Grand Prix motorcycle consists of two telescopic shock absorbers, one on either side of the front wheel, supported by the wheel itself and by the chassis via a metal bracket.

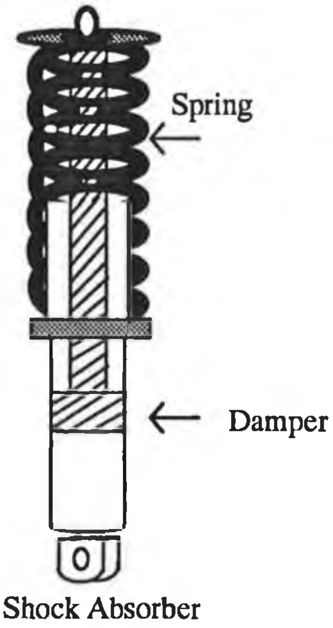


Fig. 2.2. A shock absorber profile.

Each shock absorber contains a coil. For compactness the damping rod follows the axis of the spring helix (see Fig. 2.2). In most cases the damping and stiffness are adjustable from an outside source. Fig 2.3 shows a sample telescopic shock absorber.

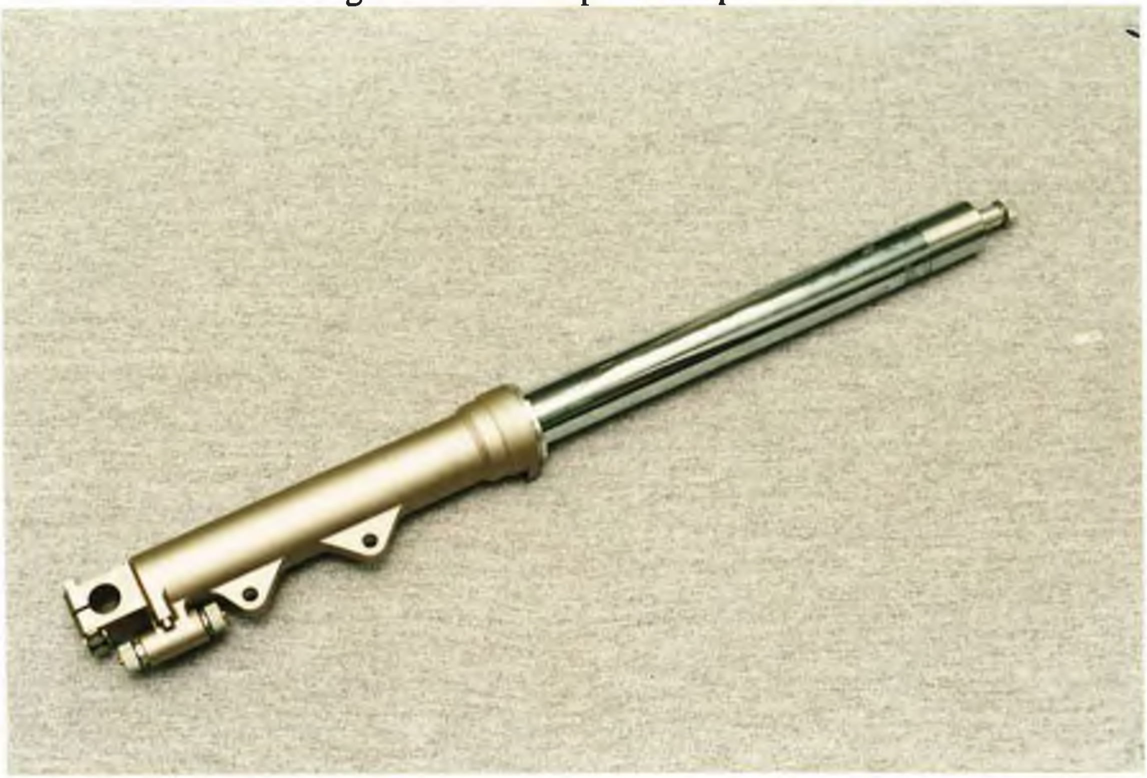


Fig 2.3 A telescopic shock absorber.

2.3 THE SUSPENSION PROBLEM

The suspension problem is caused partly by the conflicting requirements of any high speed vehicle and partly by market demands which force design engineers to keep faith in inherently bad design.

2.3.1 The conflicting requirements of high speed motorcycles

If race tracks consisted solely of flat, even, straight stretches of road the outcome of each race would be determined by engine power alone. Fortunately this is not the case. Because of this it is not good enough for a motorcycle to possess just good straight line speed but it must be able to corner in a quick and safe manner.

When a motorcycle is in straight line motion the main function of the suspension system is to isolate the unsprung mass i.e. the chassis and rider from any road irregularities. This is accomplished by using a soft set-up with low stiffness. When the motorcycle encounters a bump in the road the suspension is compressed and the rider notices little or no discomfort.

The situation changes however when the rider undertakes a cornering manoeuvre. When cornering it is essential that both tyres remain in contact with the track. This is difficult since the rider will "lean into" the corner to counteract the centrifugal force acting on the motorcycle. In doing this the size of the "contact patch", the area of the tyre which is in contact with the road, is decreased. This scenario becomes even more complicated when the rider needs to decelerate at slower corners. When he applies the brakes the weight of the motorcycle shifts forward meaning that there is more weight on the front suspension than on the rear. Because of this the front tyre has better contact with the surface than the rear tyre. When the motorcycle exits a corner good contact of the rear wheel with the ground is essential if the bike is to remain stable under acceleration. In accordance with this a stiff suspension which limits wheel movement is required.

The conflicting needs of straight line motion and cornering manoeuvres imposed on the suspension set-up cause a real problem. Suggested solutions involve the inclusion of preload, different damping for the up/down stroke of the suspension and variable spring rates at different positions of the suspension. Unfortunately these solutions treat the symptoms and not the problem itself. The resulting suspension set-ups are ad-hoc and sub-optimal.

2.3.2. Bad engineering design

The modern telescopic shock absorber comprises an aluminium slider fitted over a chromium plated steel stanchion tube (see Fig. 2.3). The obvious aesthetic appeal of these "shocks" has contributed to their domination in the motorcycle suspension market. However their popularity is not scientifically justified. Upon close examination the flawed design of these shocks becomes apparent.

Firstly when the shock absorber is fully extended there is little overlap between the slider and the stanchion (see Fig 2.4).

Because of this, extra unnecessary stresses and strains are applied to the suspension Secondly if the front suspension is considered as a whole it's clear that either slider is quite free to move independently from the other. This results in a twisting of the suspension column.

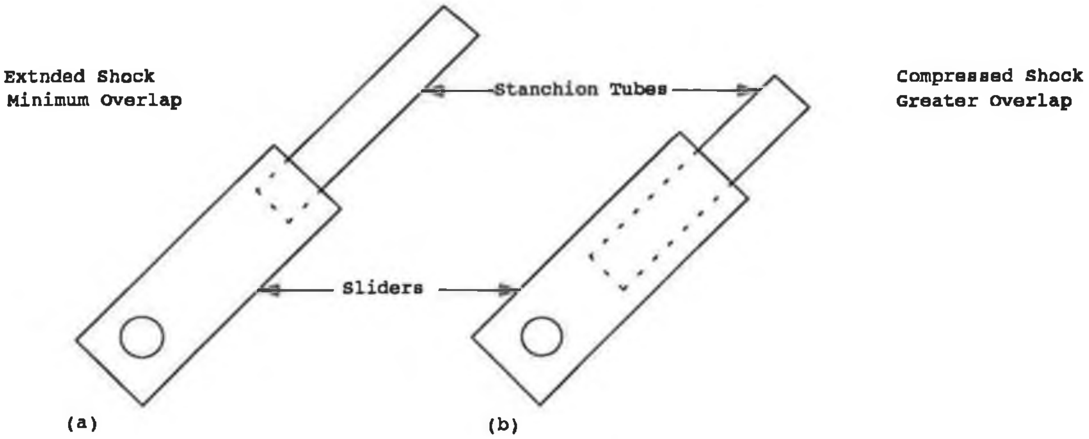


Fig 2.4 (a) Extended shock absorber, (b) compressed shock absorber.

These flaws alone lead to a suspension which is flexible in almost all directions a situation which affects lateral stability [1]. Furthermore because of the rake angle [1] (see Fig. 2.5) the front suspension is susceptible to fore and aft forces resulting in stiction.

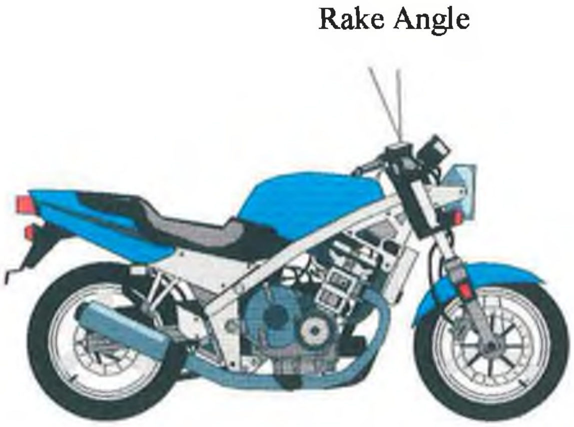


Fig. 2.5 Illustration of the rake angle.

Finally it is the nature of telescopic shock absorbers which causes the characteristic nose -dive under breaking mentioned in chapter 1. In this situation the rider is thrown forward under his own weight and has to struggle to stay upright.

2.4 SYSTEM MODELLING

Any attempt to solve the suspension problem outlined above must have as its starting point a model with which to represent the system. The model must be simple and yet represent as much of the system dynamics as possible. The purpose of such a model is twofold. Firstly, the dynamic behaviour of the actual system can be simulated leading to a better understanding of the issues at hand; and secondly, it can be used as a development platform for the testing of any algorithms to be used on the actual system.

A physical system is usually describable in terms of a mathematical relationship between the system's input and its output. This is the case with the suspension system of any vehicle. The front end suspension of a motorcycle has two constituent parts, the shock absorbers and the tyre (see Fig. 2.5). Physically speaking these can be represented diagrammatically as in figure 2.6[1]. The suspension is modelled as a two degree of freedom (2 d.o.f.) model [2], [9]. This model is sometimes referred to in the literature as a "Quarter Car Model" [9] or supposedly in the case of a motorcycle a "Half Bike Model".

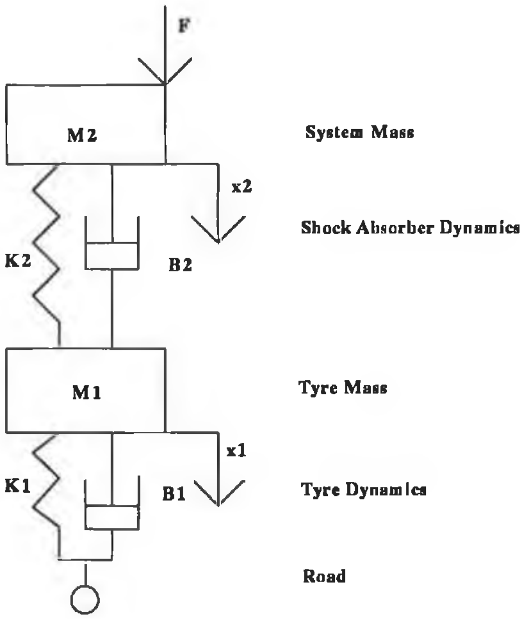


Fig 2.6 Two degree of freedom model for the front suspension

[1] Taken from [8-9].

The system mass represents both the vehicle and rider mass apportioned to the front of the bike. The force F acting on the system mass M2 is composed of three constituent forces,

(i) Weight Shift: When the motorcycle is in straight line motion its weight is apportioned equally between the front and rear wheels. However as the motorcycle decelerates under braking the weight shifts forward because of its own momentum onto the front suspension. This increases the displacement of the front suspension while at the same time decreasing rear suspension displacement. The converse occurs under acceleration. All of the weight shifts towards the rear of the motorcycle resulting in full compression of the rear suspension. In some cases this weight shift may be so pronounced that the front wheel of the motorcycle lifts from the ground resulting in the so-called "wheelie".

(ii) G-forces: When any vehicle corners at high speed it experiences what are commonly referred to as g-forces. These forces are caused by the centrifugal acceleration of the vehicle around the circle inscribing the corner. These forces have a component along the axis of the suspension and as such must be accounted for in the modelling of the suspension.

(iii) Braking forces: Under braking the suspension experiences forces transmitted from the ground in reaction to the braking of the vehicle. These also have a component along the suspension axis.

Displacements x2 and x1 are suspension and tyre deflections respectively. The elements K2 and K1 represent the shock absorber and tyre stiffness respectively. The tyres of any vehicle operate at extremely high pressures and are difficult to compress. On the other hand the ease of compression of shock absorbers is quite visible even when the rider mounts the motorcycle. This difference is attributable to the large difference in stiffness between the shock absorber spring and the tyre and air stiffness. Because of this it is usually accepted that the tyre stiffness is as much as 1000 times that of the shock absorber stiffness.

Mathematically speaking the system is described by application of Newtonian mechanics to the 2 d.o.f. model in Fig. 2.3. This yields Equ^{s} 2.1 and 2.2.

$$M_2 \ddot{x}_2 = F - B_2(\dot{x}_2 - \dot{x}_1) - K_2(x_2 - x_1)$$ Equ. 2.1

$$M_1 \ddot{x}_1 = -B_1 \dot{x}_1 - K_1 x_1 + B_2(\dot{x}_2 - \dot{x}_1) + K_2(x_2 - x_1)$$ Equ. 2.2

This model contains the most important suspension dynamics. However it would be more convenient if the model description were reduced to one equation. This can be achieved by omitting the tyre dynamics. This is justified since,

(i) It was mentioned earlier that it is usually accepted that the tyre stiffness is as much as 1000 times that of shock absorber stiffness. Keeping in mind that the maximum displacement of a front suspension is less than 20 centimetres the amount of the overall deflection which is due to tyre deflection alone is negligible.

(ii) Modern grand prix motorcycles utilise lightweight alloy wheels. In comparison to the weight of the rest of the motorcycle plus that of the rider this element is also negligible.

The system model is thus reduced to the single degree of freedom model [15] depicted in Fig. 2.7.

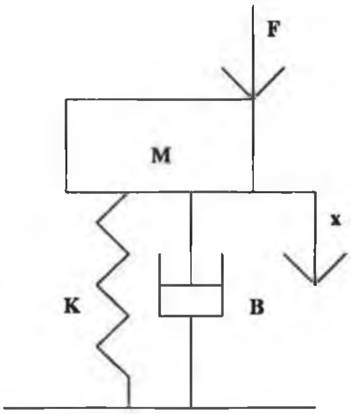


Fig 2.7 Single degree of freedom model of the shock absorber.

This simple Mass-Spring Damper (M.S.D) [19] model is described by the following equation,

$$M\ddot{x} + B\dot{x} + Kx = F \qquad \text{Equ. 2.3}$$

Equ. 2.3 assumes that the system parameters are time-invariant and that the system is linear. This is known not to be the case. Without trying to re-complicate the system model the time-invariance of the suspension parameters should be accounted for.

This results in Equ. 2.4,

$$M(t)\ddot{x} + B(t)\dot{x} + K(t)x = F(t) \qquad \text{Equ. 2.4}$$

It is this model that will be used as development tool in the following chapters.

2. 5 SUMMARY

This chapter has outlined the physical structure of suspension systems. The suspension problem has been discussed in detail justifying an investigation into this aspect of motorcycle engineering. Finally a model for the front suspension unit has been proposed. This model takes into account the time-varying aspects of the system.

Chapter 3

SYSTEM IDENTIFICATION AND PARAMETER ESTIMATION.

"..measurements and observations are nothing more than approximations to the truth, the same must be said of all calculations resting upon them, and the highest aim of all computations made concerning concrete phenomena must be to approximate as nearly as possible to the truth."

Karl Friedrich Gauss (1809)

3.1 INTRODUCTION

This chapter gives some background to system identification and parameter estimation theory. The concepts of parameter tracking and fault detection are introduced. Finally an estimation scheme is developed to the track time-varying dynamics of the front suspension unit of a motorcycle.

3.2 SYSTEM IDENTIFICATION

System identification is the process of constructing mathematical models of dynamical systems from observations and prior knowledge [23], [42]. The theory itself has its roots among the hard sciences. One of the first identification experiments was carried out by Halley in 1704 when he realised that comet sightings in the years 1531, 1607 and 1682 related to a single object. In an attempt to further understand the behaviour of this object he calculated the parameters of its orbit using Newtonian mechanics and gravitational theory (the prior knowledge) and predicted its return in 1758 [21-22]. Unfortunately Halley met his demise in 1742 and was unable to celebrate the return of the comet which was to carry his name into history.

It is clear from this, the earliest of examples, that modelling is a subset of system identification. In Halley's case the model structure was provided by the cumulative work of Gallileo, Kepler and Newton. Halley combined this wealth of background knowledge with his collection of observations (the sightings) and deduced an exact system model. He then used the prediction properties of this identified model to test its own validity.

The basis of a system identification experiment is the set of system inputs and outputs. Reasoning in this fashion leads to the obvious conclusion that identifying the system (Fig. 3.1) amounts to collecting corresponding input/output data and determining in some scientific manner a mathematical relationship between the two. This mathematical relationship will be representative of the system dynamics.

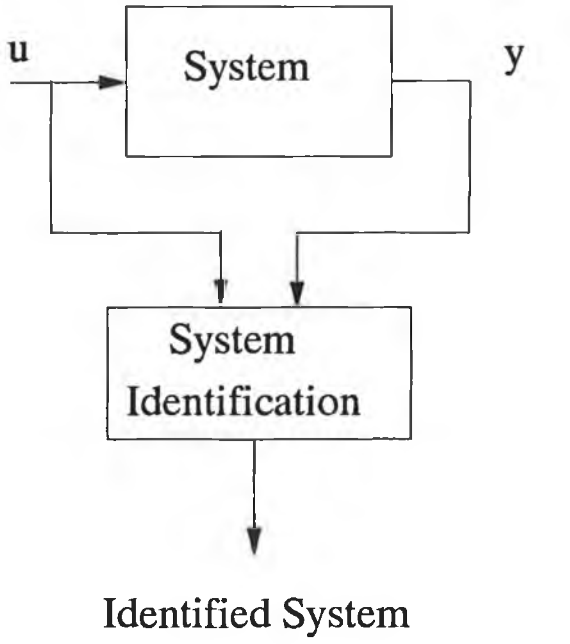


Fig. 3.1 The identification format

At this point questions arise as to whether there is only one possible structure for this relationship which describes the identified system and whether this structure can be determined prior to the identification experiment. The answer to the first question is that there are usually a number of mathematical structures which lend themselves to a particular identification experiment.

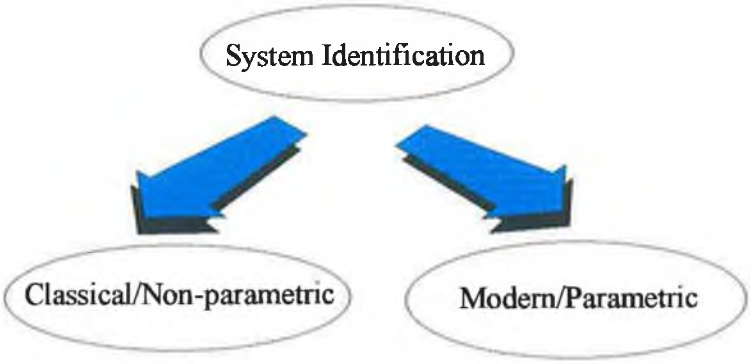


Fig. 3.2 Two classes of identification

These structures are either open-ended leading to identification by classical or non-parametric means or closed resulting in the use of modern estimation techniques [23], [24], [26], [40-41] (see Fig. 3.2).

3.3 CLASSICAL IDENTIFICATION

Non-parametric methods of identification refer to those methods which do not employ a finite dimensional parameter vector in the search for the best description of the system [24]. As such identification by these means results in the assignment of impulse responses (using time-domain methods) or frequency responses (using frequency domain methods) to the system under investigation.

3.3.1 Time domain methods

It is known from systems theory [25] that the input-output dependency of a system can be characterised by its impulse response h(t). Classical time-domain analysis centres around the approximation of this response. Once this is known the system output y(t) is given by the following convolution integral

$$y(t) = \int_0^t h(t-\tau)u(\tau)d\tau, \quad t \geq 0 \qquad \text{Equ. 3.1}$$

where $u(\tau)$ is the system input at time τ

Measurements of the system input/output data are usually taken at discrete moments in time, thus it is necessary to deal with the discrete formulation of Equ. 3.1,

$$y(k) = \sum_{j=0}^{\infty} h(j)u(k-j) \qquad \text{Equ. 3.2}$$

k in Equ. 3.2 indicates the sample instant. To identify the system impulse response one may consider simply "dividing across" by the system input in Equ. 3.2. However noise considerations dictate that an alternative method be used. The literature [23-24], [26] recommends that we try to "correlate out" the noise component in the data. Calculation of the input-output cross-correlation function, $r_{uy}(k)$ yields the Wiener-Hopf [27] equation,

$$r_{uy}(k) = \sum_{j=-N}^{\infty} h(j) r_{uu}(k-j) \qquad \text{Equ. 3.3}$$

where $r_{uu}(k)$ is the auto-correlation function of the input sequence.

If the input chosen is a white noise signal then Equ. 3.3 reduces to

$$r_{uy}(k) = h(k)\sigma_n^2 \qquad \text{Equ. 3.4}$$

σ_n^2 being the variance of the input white noise signal which is unity.

Calculation of the system impulse response is then trivial.

Rather than calculating the entire system impulse response it is more common in industry to examine the system step response and from this determine the system dynamcis. A more detailed discussion of this topic is given in section 3.4.1.

3.3.2 Frequency domain methods

In some cases it may be convenient to deal with the frequency domain representation of the system. The identification experiment must now alter to meet this new requirement and it is the frequency domain representation of the impulse response, the system frequency response, which is now approximated.

If the system operates within a certain band of frequencies then a simple solution to the problem is achieved by applying various sinusoids from within this frequency range to the system. Given the system input U and the system output Y, the discrete system transfer function $H(z^{-1})$ within the area of interest can then be approximated as

$$H(z^{-1}) = \frac{Y(z^{-1})}{U(z^{-1})} \qquad \text{Equ. 3.5}$$

where z^{-1} is the backward shift operator.

This method may be susceptible to noise. However it is possible to correlate this out as mentioned before. Failing this, other more powerful methods such as spectral analysis are available [24], [26].

These non-parametric methods are tried and tested. However they are inappropriate for the current work. Firstly they yield open-ended solutions which may not be amenable to modern control/estimation techniques. Secondly they seldom incorporate any prior knowledge about the system which may be available. Therefore this work will concentrate on parametric methods of identification.

3.4 PARAMETRIC IDENTIFICATION

Parametric identification is both physically and mathematically more appealing than any of the classical methods mentioned above. The reasons for this lie in the closed nature of the resulting solutions and the ability to take advantage of a priori information.

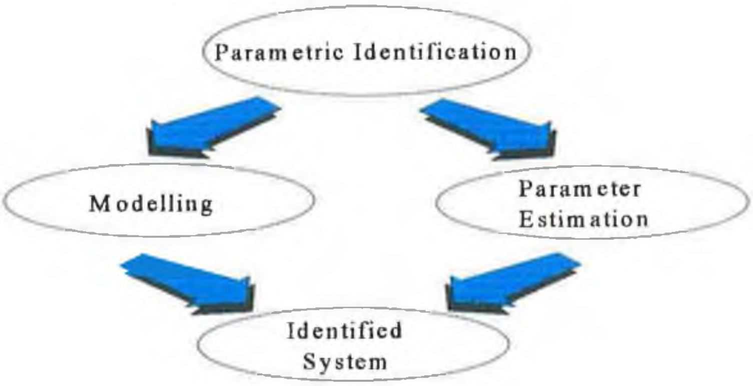


Fig. 3.3 Parametric identification

Parametric Identification is made up of two distinct sub-tasks:

(i) System Modelling,
(ii) Parameter Estimation.

3.4.1 System modelling:

Modelling with respect to the suspension unit has already been discussed in section 2.4. In a more general sense modelling refers to the representation of the system dynamics by either graphical or mathematical means. In some cases the structure of the model (its order) will be obtained through comparison with similar systems; otherwise the application of physical laws such as Newton's laws of motion and Kirchoff's current and voltage laws will determine the form of the model.

Attempts by Cohen and Coon [28] and Ziegler and Nichols [29] to tune the parameters of Proportional Integral Derivative (P.I.D.) [30], [53] controllers concentrated on what was then referred to as the *system signature*, now referred to as the system step response. It became apparent that the signature of a large number of processes possessed certain similarities. Each had a certain d.c. gain, rise time and settling time. As a result both Ziegler and Nichols and Cohen and Coon deduced standard models which represented the system dynamics in terms of these parameters. For example Cohen & Coon observed that the response, $G_{prc}(s)$, of a large proportion of industrial processes had a sigmoidal shape which could be approximated by a 1^{st} order system with a deadtime response given by

$$G_{prc}(s) \approx \frac{Ke^{-t_d s}}{\tau s + 1}$$ Equ. 3.6

where

K = the process d.c. gain,

t_d = the process dead time,

τ = the process time constant.

Once the system step response was obtained a graphical examination of it yielded approximate values for the constants in Equ. 3.6. Ziegler and Nichols devised similar models which could be parameterized using similar methods.

The graphical methods outlined above yield good results for process control and are simple to implement. However this simplicity means that the model may lack certain nuances of the system dynamics. These models are also continuous in nature and as such are not compatible with modern control/estimation techniques. These latter techniques implement algorithms which operate in discrete time so it is necessary to obtain a discrete formulation of the system dynamics.

The dynamics of many systems can be described by their continuous time differential equations. The discrete counterpart of these equations is the difference equation. This is obtained from the differential equation by using any of a number of discrete transformations [30]. These transformations will normally yield the following form of discrete-time difference equation [24].

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + \dots\dots\dots a_{na} y(k-n_a) =$$

$$b_0 u(k-1) + \dots\dots + b_{nb} u(k-n_b) + e(k) + c_1 e(k-1) + \dots\dots + c_{nc} e(k-n_c)$$ Equ. 3.7

or

$Ay(k) = Bu(k\text{-}1) + Ce(k)$

$$A = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots\dots + a_{na} z^{-na}$$

$$B = b_0 + b_1 z^{-1} + \dots\dots\dots + b_{nb} z^{-nb}$$

$$C = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots\dots\dots + c_{nc} z^{-nc}$$ Equ. 3.8

This is known as an Auto-Regressive, Moving Average with eXogenous input or ARMAX model [24]. The variable k refers to the current sample point. The variables u(k-1),.........,u(k-n_b) are lagged values of the discrete system input while the variables y(k-1),.....,y(k-n_a) are lagged values of the system output. The sequence e(k-1),....,e(k-n_c) is the sequence of system error or noise samples. The noise source, e, is assumed white. However in some cases the system noise is coloured. This is accounted for by assuming the white noise e is filtered through a discrete filter whose coefficients are given as (c_1,.....,c_{nc}). The coefficients of this filter along with the coefficients of the system input (b_0,.......b_{nb}) and the coefficients of the system output (a_1.......a_{na}) make up the system parameter vector θ.

If an estimate of the parameter vector ($\hat{\theta}$) is available the systems current output, $\hat{y}(k|\theta)$, given this estimate can be predicted using the predictor, [24]

$$\hat{y}(k|\theta) = \varphi^T(k)\hat{\theta} \qquad \text{Equ. 3.9}$$

where

$$\hat{\theta} = [\hat{a}_1, \hat{a}_2, \ldots\ldots, \hat{a}_{na}, \hat{b}_1 \ldots\ldots \hat{b}_{nb}, \hat{c}_1, \ldots\ldots, \hat{c}_{nc}]^T$$

$$\varphi(k) = [-y(k-1), \ldots -y(k-n_a), u(k-1), \ldots\ldots u(k-n_b), e(k-1) \ldots\ldots e(k-n_c)]$$

Since the variables e(k-1).....e(k-n_c) are unmeasurable they are approximated as

$$e(k) = y(k) - \hat{y}(k) \qquad \text{Equ 3.9a}$$

where y(k) is a measurement of the system output and $\hat{y}(k)$ is the estimate of this measurement given by Equ. 3.9. Since this estimate of e(k) is θ dependant e(k) should be written as e(k,θ). Since φ is dependant on e it is also dependant on θ. Thus there is a non-linear relationship between φ and θ. Equ. 3.9 must be rewritten as

$$\hat{y}(k|\theta) = \varphi^T(k,\theta)\hat{\theta} \qquad \text{Equ. 3.9b}$$

Because of this non-linearity Equ 3.9b is known as a pseudolinear regression.

If it is assumed that the statistics of the noise entering the equation are white then Equ. 3.8 reduces to

$$Ay(k) = Bu(k-1) + e(k) \qquad \text{Equ. 3.10}$$

This is known as an Auto-Regressive with eXogenous input or ARX model structure.

As with an ARMAX model a predictor can be formed once an estimate of the parameter vector is available. This predictor, known as a linear regression, is given as

$$\hat{y}(k|\theta) = \varphi^T(k)\hat{\theta} \qquad \text{Equ. 3.11}$$

where

$$\hat{\theta} = [\hat{a}_1, \hat{a}_2, \ldots. \hat{a}_{na}, \hat{b}_0, \hat{b}_1, \ldots.. \hat{b}_{nb}]^T$$

and

$$\varphi(k) = [-y(k-1), -y(k-2), \ldots..-y(k-n_a), u(k-1), u(k-2), \ldots\ldots..u(k-n_b)]$$

The discrete nature of this model makes it amenable to modern estimation techniques. It is also simple and yet can accommodate large systems by the simple addition of extra 'a' and/or 'b' parameters.

The modelling aspect of parametric identification has now been treated. Modelling the system using the ARMAX/ARX structure(s) reduces the identification problem to one of estimating the optimal value of θ, the parameter vector. This is the realm of parameter estimation.

3.4.2. Parameter estimation:

Parameter estimation is the process of determining in some optimal fashion the value of θ in equations 3.10/11. Optimality is in some senses user definable and because of this there is a plethora of estimation algorithms, all of which claim to be optimal.

The most complete of these is the Bayes estimator [23-24], [31-32]. This is an all-encompassing estimator which relies upon the statistical properties of the system to provide the best estimate of θ. Given that P denotes the probability density of a function, this approach considers the parameters as random variables with a-priori (before measurements are taken) probability densities $P(\theta)$ [33] before the identification experiment. Measurements are made and the Maximum-A-Posterior (MAP) estimates, $\hat{\theta}_{MAP}$, are calculated as those which maximise the probability $P(\theta|y)$. $P(\theta|y)$ is the probability of θ given that the measurement y has been taken. Thus,

$$\hat{\theta}_{MAP} = \arg(\hat{\theta}) \max P(\hat{\theta}|y) \qquad \text{Equ. 3.12}$$

where

$$P(\hat{\theta}|y) = \frac{P(y|\hat{\theta})xP(\hat{\theta})}{P(y)} \qquad \text{Equ. 3.13}$$

Equation 3.13 is known as Bayes rule [23-24], [33], hence the term Bayesian estimation. P(y) is the probability density function of the measurement y. $P(y|\hat{\theta})$ and $P(\hat{\theta})$ are the probability density functions of y given $\hat{\theta}$ and $\hat{\theta}$ respectively.

The necessary provision of a prior probability distribution is both the chief strength and weakness of the Bayesian estimator. Any algorithm which incorporates prior information is welcome but it is unrealistic to think that a prior distribution will be available in all situations. This problem can of course be surmounted by ignoring the prior probability distribution. This leads to the Maximum Likelihood (M.L.) estimator [34-38].

Once the measurement data are known the posterior probability distribution function, $P(y|\hat{\theta})$, becomes a function of the parameter vector. The value of $\hat{\theta}$ which maximises this function, also known as the likelihood function, is the maximum likelihood estimate of θ. In most cases it is numerically difficult to maximise the likelihood function. In such scenarios it is simpler to minimise the negative log of its inverse, ***the log loss function*** (LLF) [23], [26], [35-37]. Conceptually speaking the loss function can be seen as an error in the estimate of θ. The M.L. estimate is therefore optimal in the sense that it minimises this loss or error function. The M.L. estimate $\hat{\theta}_{ML}$ (for N measurements) is given as

$$\hat{\theta}_{ML} = \arg(\hat{\theta}) \min \mathrm{LLF}(\hat{\theta}) \qquad \text{Equ. 3.14}$$

where

$$\mathrm{LLF}(\hat{\theta}|y) = -\log P(y|\hat{\theta}) \qquad = -\sum_{i=1}^{N} \log P(y(i)|y(1{:}i-1,\hat{\theta}) \qquad \text{Equ. 3.15}$$

The M.A.P. and M.L. estimators are in wide use especially in the aeronautical industry but their reliance upon statistical inference (some would say mathematical complexity) ranks them in second place behind the most popular estimator which is the least squares estimator.

3.5 THE LEAST SQUARES ESTIMATE

The theory of least squares estimation was first devised in 1795 by the then 18 year old Karl Gauss. Following in Halley's footsteps, he was investigating the motion of the planets and comets using telescopic measurements. Knowing that the motion of astral bodies can be characterised by the formulation of six parameters he set about the determination of these parameters. Halley had achieved this for the motion of "his" comet so there was nothing new in what Gauss was attempting. The novelty lay in the method he used and it has since gained great acclaim among mathematicians and engineers alike.

Gauss devised the theory of least squares under the assumption that all observations are subject to error. He defined the least squares estimates as "*the most probable*". He later expounded that the most probable estimates are those that minimise the sum of the squares of the errors between the actual system output y(k) and the predicted output $\varphi^T(k)\theta$ given the model, i.e. the mean squared error. Given the linear regression for the ARX model in Equ. 3.10 the mean squared error $V_N(\theta,\varphi^N)$

where $\varphi^N = [y(1),u(1),y(2),u(2),.....y(N),u(N)]$

is given by

$$V_N(\theta,\varphi^N) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}[y(k) - \varphi^T(k)\theta]^2 \qquad \text{Equ. 3.16}$$

The unique feature of this, the least squares criterion is that it is quadratic in nature. Therefore it can be minimised analytically. Once minimised (derivation listed in Appendix A.1) the resulting estimate of the parameter vector θ is given as [24],

$$\hat{\theta}_{LS} = \arg\ \min V_N(\theta,\varphi^N) = \left[\frac{1}{N}\sum_{k-1}^{N}\varphi(k)\varphi^T(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)y(k) \qquad \text{Equ. 3.17}$$

The situation changes however when the system noise is coloured. In this scenario the linear regression of Equ. 3.10 is no longer applicable. In this situation the system output is predicted using the pseudo linear regression of Equ. 3.8. Estimation of the parameter vector now using the L.S. estimator results in biased estimates. The problem may be overcome by estimating the parameters of the filter C in the ARMAX representation of Equ. 3.8. Unfortunately this requires knowledge of e(k), e(k-1) quantities which as mentioned before are unmeasurable. The problem is by-passed by approximating these variables as the estimator prediction errors (or residuals) [20]. When L.S. estimation is applied in such circumstances it is referred to as extended least squares (or approximate maximum likelihood) estimation.

It has been stated previously that the least squares estimator is the most popular. However the algorithm described thus far is a batch estimator meaning that it processes the system data off-line and en-masse yielding one estimate of the parameter vector θ. Self-tuning control systems operate in real time and as such require on-line or recursive estimation of the system parameters. Naturally enough the most popular recursive estimator is a recursive form of the most popular batch estimator i.e. the recursive least squares (R.L.S.) estimator.

The R.L.S. algorithm is formulated as

$$\hat{\theta}(k+1) = \hat{\theta}(k) + P(k+1)\varphi(k+1)[y(k+1) - \varphi^T(k+1)\theta(k)] \qquad \text{Equ. 3.19}$$

$$P(k+1) = P(k)\left[I_m - \frac{\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k+1)}\right] \qquad \text{Equ. 3.20}$$

The mechanics of this algorithm are easily explained (the derivation of this recursion is listed in Appendix A.2.). The current parameter estimate $\hat{\theta}(k+1)$ is given as the previous estimate, $\hat{\theta}(k)$, plus a correction which is proportional to the prediction error $\varepsilon(k) = y(k+1) - \varphi^T(k+1)\hat{\theta}(k)$. This correction takes place in a fashion which not only minimises the least squares criterion, but under Gaussian noise conditions, the current parameter estimate covariance P(k+1) [23]. Suitable selection of the applied input sequence [45] leads to a final estimate whose covariance approaches the Cramer Rao lower bound [23-24] which is the minimum for the estimator. It is also of interest that under these noise conditions the least squares estimates are equivalent to the maximum likelihood estimates referred to earlier.

As it stands Equ. 3.20 is prone to numerically instability. It is normal practice to implement this equation using one of the many factorisation algorithms. The most prevalent among these is Bierman's U-D factorisation [43]. This is basically a square root factorisation without the square roots. It is assumed from now on that this factorisation is used in the parameter estimate covariance equation (Equ. 3.20) whenever the R.L.S. or one of its derivatives is referred to.

As an example consider the estimation of the following simple deterministic system.

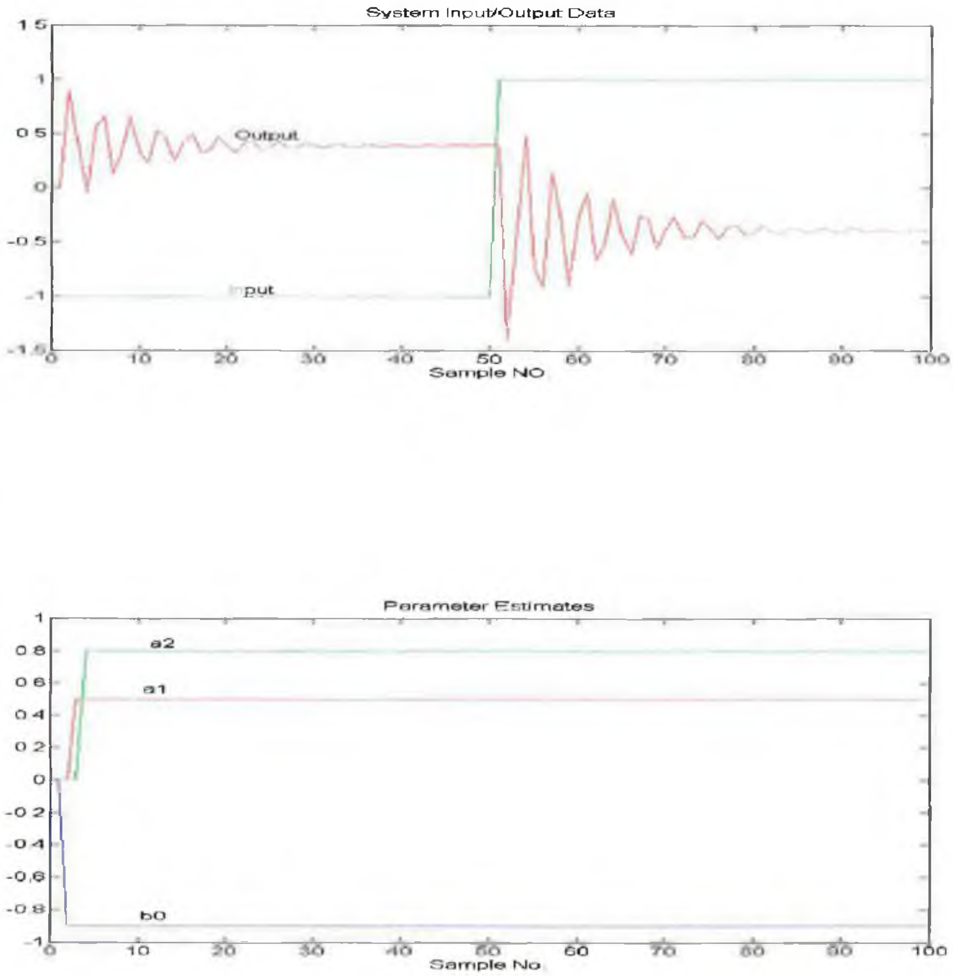
$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_0 u(k-1)$$

$$a_1 = 0.5, \qquad a_2 = 0.8, \qquad b_0 = -0.9,$$

A 1 Hz. square wave was applied to the above system. The recursive least squares algorithm was applied to the resulting input/output data. The data and the resulting estimates are depicted in Fig. 3.5 (overleaf).

As seen in Fig. 3.5 the R.L.S. algorithm performs well in this case. The convergence/variance properties of the estimates are determined by the system noise. Since the above system is deterministic these estimates are quick to converge and have excellent variance properties. It is to be expected that the condition of these properties will deteriorate under non-ideal conditions.

Unfortunately when the system dynamics in question are time-varying (such as in the motorcycle case) the estimator becomes sub-optimal and cannot track these changes. Although the ordinary R.L.S. algorithm does possess a certain innate tracking capability its rate of convergence is too slow, so much so that the system dynamics may have altered several times before the estimator reconverges.



(a)

(b)

Fig. 3.5 (a) Identification data, (b) Parameter estimates using R.L.S.

The problem arises because once the estimator has converged initially, the P matrix becomes extremely small conveying a high confidence in the current estimates. This in turn means that the estimator gain is small, rendering the effect of new measurements minute in comparison to the effect of old estimates. In short the estimator has a long memory.

It seems natural from the previous discussion that attempts to modify the R.L.S algorithm to increase its tracking capability centre around the parameter estimate covariance update equation, i.e. equation 3.20. The literature [23-24], [26] recommends two modifications to the ordinary R.L.S.. These are discussed below.

3.6 THE FORGETTING FACTOR ALGORITHM.

As mentioned previously low values of the P matrix lead to unwarranted confidence in the parameter estimates at points in time when the system is changing and these parameters are varying. These current parameter estimates are based on old data. It is possible to counteract this by decreasing the memory size of the estimator. Once this task is performed the estimator will put less weight on old measurements making it more alert to system changes. To achieve this a forgetting factor λ is inserted into Equ. 3.20 as follows,

$$P(k+1) = \lambda^{-1}P(k)\left[I_m - \frac{\varphi(k+1)\varphi^T(k+1)P(k)}{\lambda + \varphi^T(k+1)P(k)\varphi(k+1)}\right] \qquad \text{Equ. 3.21}$$

λ is usually chosen less than one. However care must be taken in its choice, as a value which is too small will lead to estimates which are susceptible to noise.

Consider again the system,

$$y(k) = -a_1y(k-1) - a_2y(k-2) + b_0u(k-1)$$

but now let the system parameters vary as follows

$$a_1 = 0.5, \qquad a_2 = 0.8,$$

$$b_0 = -0.9, \qquad k < 200;$$
$$b_0 = -0.5, \qquad 200<k<400;$$
$$b_0 = -0.9, \qquad 400<k<800;$$

A 1 Hz. square wave (see Fig 3.5a.) was applied to this system and its system parameters were identified using the forgetting factor algorithm as outlined above. λ was set to 0.97. The resulting parameter estimates are depicted in Fig. 3.6.

The estimates in Fig. 3.6 illustrate that the forgetting factor algorithm works insofar as it is capable of tracking system parameters. However it is of note that the overall variance of the parameters has increased and the tracking of the parameters is rather sluggish.

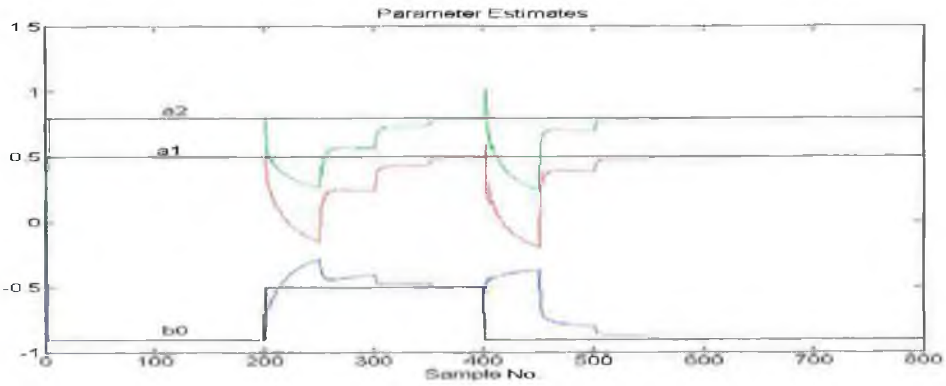


Fig. 3.6 Tracking of the system parameters using the forgetting factor algorithm.

The main failing of this algorithm is that it is prone to a condition known as estimator windup or covariance blow-up [20] (similar to integral windup experienced in some controllers). Under normal operating conditions λ will act to increase the size of P(k+1) while the data vector φ(k+1) will act to decrease it, thus keeping a fine balance between alertness to parameter changes and convergence of the estimates to their correct values. However when the input signal decreases significantly in amplitude, below a level were it ceases to excite the system, the data vector φ(k+1) holds no new information. At this point the bracketed term in Equ. 3.21 will have little or no effect on the covariance matrix and $P(k+1) \cong \lambda^{-1}P(k)$. Because $\lambda<1$ the covariance matrix P(k+1) increases in size beyond control, hence the term covariance blow-up. This has the effect of rendering the parameters unreliable and biased.

3.6 THE KALMAN FILTER

The forgetting factor algorithm outlined above is as already stated the most popular method used to track time-varying parameters. However in certain circumstances this algorithm is non-ideal. One of its chief weaknesses is the underlying fact that the inclusion of a forgetting factor restricts the user to the assumption that all of the system parameters vary in exactly the same manner and at the same rate. The estimates which result when this algorithm is applied to systems whose parameters vary independently and at different rates are therefore suboptimal. In this scenario, given prior information about expected rates of fluctuation in the system parameters, the ideal tracking algorithm would incorporate this knowledge. The flexibility offered by such an algorithm would greatly enhance not only the accuracy of the resulting parameter estimates but also the rate of convergence to new parameter values after the system had altered. This flexibility is an inherent feature of the Kalman filter.

The Kalman filter was developed by Rudy Kalman in 1960 [46]. To gain an understanding of the Kalman filter an insight into the concept of *state space* and *state estimators* must be obtained.

3.7.1. State estimation.

Although the concept of state space is not new it is only since the advent of modern control theory in the early 1960$^{'s}$ that this form of system description has gained notoriety. The theory of state space is based on the fact that, given a set of variables (known as state variables) at a certain point in time, say t_0, together with the knowledge of the system input at this time and thereafter, the behaviour of the system can be determined at any time after t_0. Any given system can be described by the following equations [24], [53]

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t) + w(t) \qquad \text{Equ. 3.22a}$$

$$y(t) = C(\theta)x(t) + v(t) \qquad \text{Equ. 3.22b}$$

Equ. 3.22a is a matrix differential equation and is known as the state equation. The system state vector x(t) is comprised of all the system state variables. In most cases these variables are physical quantities such as velocities and accelerations in the case of mechanical systems and voltages and currents in the case of electrical systems. However it is possible to describe the system using abstract state variables which may have no physical significance whatsoever.

A is the state matrix and B the input matrix. Both of these matrices are functions of the system parameter vector θ and as such these matrices are time-varying if the system parameters are non-stationary. The variable w(t) is a white noise sequence known as process noise. This takes account of unmodelled disturbances acting on the system.

Equ. 3.22b is the measurement equation. The system output y(t) is a function of the state vector x(t). The state vector is coupled to the system output by the vector C(θ). The variable v(t) is a white noise sequence known as measurement noise.

The Kalman filter was originally presented as an alternative solution to the so-called *Wiener problem*[1] [27]. This problem dealt with two issues,

(i) The prediction of random signals,

(ii) The separation of random signals from noise.

With reference to the state space description given in Equ.^{s} 3.22a/b the Wiener problem reduces to the following,

> We are given the state vector x(t) and noise v(t), but can only measure the signal y(t). Given these facts what can we infer about x(t) ?

Before Kalman it was accepted that this problem should be tackled by filtering the measurement data y(t) using what was referred to as a *Wiener filter*[1] [27]. Invariably the nature of this filter was unknown and academics spent their time formulating methods to approximate the best filter for a given scenario.

[1] Wiener filtering originated as a result of the formulation of the Wiener-Hopf equation. (see section 3.3)

Kalman had some misgivings about Wiener filtering. Firstly the optimal filter was specified by its impulse response. This meant that construction of this filter was a complicated task. Secondly numerical determination of this optimal impulse response was often quite involved and poorly suited to machine computation. Thirdly and perhaps most importantly the Wiener filter was developed under the assumption that the underlying signal and noise processes were stationary. Since this was not the case in many real applications the Wiener filter had a limited area of applicability.

The foundation of Kalman's misgivings lay in the system description used to develop Wiener filtering, the system impulse response. Recognising this Kalman developed his own filtering theory around the state space system description of Equ.^{s} 3.22a/b. However Kalman's work was developed during an era when digital computation was becoming increasingly popular. Recognising this Kalman's theory was originally developed for the discrete version of Equ.^{s} 3.22a/b.

The discrete versions of these Equ^{s} are,

$$x(k+1) = \Phi x(k) + \Gamma u(k) + w(k) \qquad \text{Equ. 3.23a}$$

$$y(k) = Hx(k) + v(k) \qquad \text{Equ. 3.24b}$$

As in Equ^{s} 3.22a/b x and y represent the system state vector and the system output vector respectively. However in this case x and y are referenced by k a measure of discrete time and not t a measure of continuous time. Φ is the state transition matrix and Γ is the input matrix. H couples the state vector to the system output y(k) and once again w(k) and v(k) represent white process and measurement noise respectively.

As stated above, in the context of state space the Wiener problem reduces to one of inference about the state vector x(k) (or x(t) in the continuous case) given the measurement vector y(k).

This problem can be broken down into three different areas,

(i) *Smoothing*:	Finding x(k) given y(k+1),y(k),........y(0)
(ii) *Filtering*	Finding x(k) given y(k), y(k-1),........y(0)
(iii) *Prediction*	Finding x(k) given y(k-1), y(k-2),.....y(0)

Realising that (i) - (iii) are inherently linked Kalman referred to these problems under the collective term of *state estimation*. He formalised his estimator under the constraints that the resulting state estimates should be linear functions of the measurement data and that these estimates should be obtained in an optimal fashion which involved the minimisation of a loss function[2]. The loss function that he chose was the square of the error between the actual state value and its current estimated value.

Working under these constraints Kalman derived the following recursive state estimator.

$$\hat{x}(k+1) = \hat{x}(k) + K(k+1)\varepsilon(k+1) \qquad \text{Equ. 3.23a}$$

$$\varepsilon(k+1) = y(k) - \varphi^T(k+1)\hat{x}(k) \qquad \text{Equ. 3.23b}$$

$$K(k+1) = P(k+1)\varphi(k+1) = \frac{P(k)\varphi(k+1)}{1+\varphi^T(k+1)P(k)\varphi(k+1)} \qquad \text{Equ. 3.23c}$$

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k)} + R \qquad \text{Equ. 3.23d}$$

In Equ. 3.23a $\hat{x}(k+1)$ is the new estimate of the state vector x(k) given the previous estimate $\hat{x}(k)$ and the *innovation* or prediction error ε(k+1). The innovation is multiplied by a factor K(k+1) known as the Kalman gain. This controls the effect new measurements have on the current state estimate. P(k+1) is the covariance of the current

[2] This is similar to the log loss function used in maximum likelihood parameter estimation (see section 3.4.2

estimates and is an indication of the reliability of these estimates. R is the covariance of the system process noise w(k).

The estimates given by Equ. 3.23a are optimal in the sense that the mean squared error between the estimated state and the actual state is minimised. However under the assumption of Gaussian noise conditions the Kalman filter also minimises the mean squared error between the actual system output and the predicted output given by $\varphi^T(k+1)\hat{x}(k)$ [23]. This is the same criterion used in the R.L.S. algorithm for parameter estimation.

3.7.2 The Kalman filter as a parameter estimator.

In the development of the R.L.S. estimator in section 3.5 the underlying model was given as

$$y(k) = \varphi^T(k)\theta + e(k) \qquad \text{Equ. 3.24}$$

This system can be described in state space form as

$$x(k+1) = x(k) \qquad \text{Equ. 3.24a}$$

$$y(k) = \varphi^T(k)x(k) + e(k) \qquad \text{Equ. 3.24b}$$

where the state vector x(k) is given as

$$x(k) = (a_1, a_2, \ldots\ldots a_{na}, b_1, \ldots\ldots\ldots b_{na})^T = \theta \qquad \text{Equ. 3.25}$$

Since Φ is taken as the identity matrix and there is no state noise, the system is time-invariant. The optimal state estimate for this system can be computed using the Kalman filter equations (Equ^s. 3.23a-d). Since the states of this system are the systems parameters the Kalman filter is in fact operating as a parameter estimator. Moreover since process noise has been ignored in this system (and hence R=0) it can be shown that the Kalman filter reduces to Equ^s. 3.19/20, the R.L.S. algorithm. Thus in the case of Equ^s. 3.24a/b under Gaussian noise conditions and assuming there is no process noise the Kalman filter and R.L.S. estimator are equivalent.

At first sight this may seem like a trivial piece of information. However in the current context this statement is highly important. The Kalman filter was originally developed to estimate system states which vary with time. The above discussion leads to the obvious conclusion that there is no reason to suggest that time-varying parameters cannot be tracked using this algorithm. In order to achieve this goal it is necessary to remove the condition that there be no process noise, yielding

$$x(k+1) = x(k) + w(k) \qquad E(w(k)) = R \qquad \text{Equ. 3.26}$$

This means that the system states (parameters) are no longer time-invariant. Giving up this assumption means that only the parameter covariance update equation in the Kalman filter algorithm is altered giving,

$$P(k+1) = P(k) - \frac{P(k)\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k)} + R \qquad \text{Equ. 3.27}$$

Equ. 3.27 is now equivalent to the state estimate covariance update equation, Equ. 3.23d.

Norton [23] suggests that implementing the Kalman filter in this fashion as a parameter tracking scheme is akin to explicitly modelling the parameter variations. This yields

$$\theta(k+1) = \theta(k) + w(k) \qquad E(w(k)) = R \qquad \text{Equ. 3.28}$$

which is equivalent to Equ. 3.26.

However in this context w is not seen as process noise but an explicit model of the parameter variation known as a *random walk* model. R is chosen as a positive definite matrix, each element along the diagonal indicating how much the corresponding parameter in the parameter vector θ varies. This alters only the covariance update equation of the R.L.S. estimator resulting in,

$$P(k+1) = P(k)\left[I - \frac{\varphi(k+1)\varphi^T(k+1)P(k)}{1+\varphi^T(k+1)P(k)\varphi(k+1)}\right] + R \qquad \text{Equ. 3.29}$$

Equ. 3.29 is equivalent to Equ. 3.27 thus maintaining the equivalence of the Kalman filter and the R.L.S. estimator.

Since w is under the control of the user this algorithm allows the user to indicate when certain parameters vary more rapidly than others. This is in contrast with the forgetting factor algorithm which as stated previously assumes that all of the parameters vary in the same manner.

Consider again the system tracked in section 3.6. Replace the forgetting factor mechanism with the Kalman filter (or random walk) and let R equal the identity matrix. The resulting estimates are illustrated in Fig. 3.7.

It is interesting to note that the Kalman filter responds quicker than the forgetting factor algorithm (see Fig. 3.6) to changes in system parameters. However it is notable that there is a severe degradation in the variance properties of the resulting estimates. This degradation occurs because implementation of the Kalman filter necessarily means that

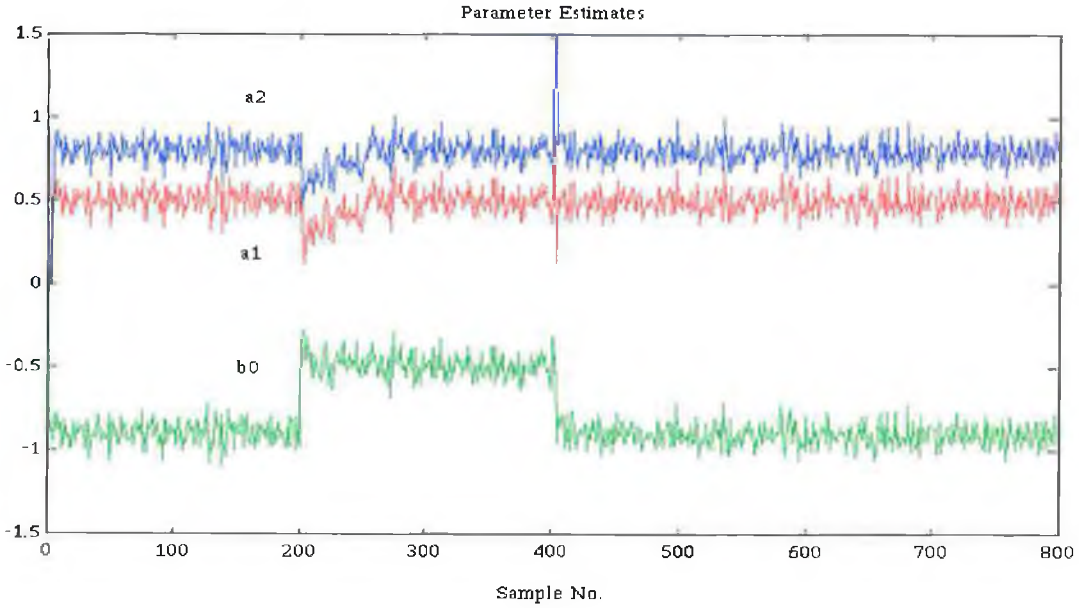


Fig. 3.7 Tracking the system parameters using the Kalman filter.

the estimator is never confident of the reliability of the current estimates. This happens because the minimum value obtainable by the parameter estimate covariance matrix P, is R.

Although both the forgetting factor algorithm and Kalman filter track time varying parameters, they do so in a rather passive manner. They both assume that the system is changing constantly even when it is not. Clearly it would be desirable to increase the size of the matrix P at instances of time variation only. This assumes knowledge of these occurrences which is unrealistic. It does however seem plausible that these occurrences are detectable. Algorithms which detect the occurrence of a parameter change are known as *fault detection* algorithms.

3.8 FAULT DETECTION

The physical nature of most control systems means that they are susceptible to failure. In order to maintain continuity of the control protocol it is necessary to be able to detect these failures so that counteracting measures can be taken. This is the realm of fault detection.

The fault detection procedure consists of three tasks, *alarm*, *isolation* and *estimation* [47]. The alarm task is concerned with deciding whether a fault has occurred or not. Isolation is the determination of the source of the failure and estimation is the process of determining the extent of the failure. A typical example would be the on-board control system in an aeroplane. If a system sensor or actuator fails the control system must be alerted to enable it to make corrections (such as an actuator replacement) so as to avoid the obvious disastrous consequences which may occur.

In more recent times fault detection algorithms have found application in the field of parameter estimation or more specifically parameter tracking. [48-49]. It is quite clear from the arguments outlined above that any attempt to track the parameters of the motorcycle suspension system would only benefit from the inclusion of a fault detection algorithm. It must be noted that whenever a fault is referred to in relation to parameter estimation it is not implied that there is something amiss with the system, rather that it is changing.

The literature [47-52] indicates that there are many forms of fault detection scheme. Perhaps the simplest of all are referred to as "failure sensitive filters". These consist of estimation schemes which have been altered to make them more alert. The main problem with the original filters is that they learn the system too well and give little weight to new system measurements. The forgetting factor and random walk modifications to the

It is visible from Fig.3.8 that this simple detection algorithm which monitors the estimator error magnitude alone works quite well. It tracks the system parameters with better variance properties than those produced by the Kalman filter and reconverges to the correct parameter estimates faster than the forgetting factor algorithm. These properties are due to the active nature of this method as opposed to the passivity of the methods illustrated in sections 3.6 and 3.7.

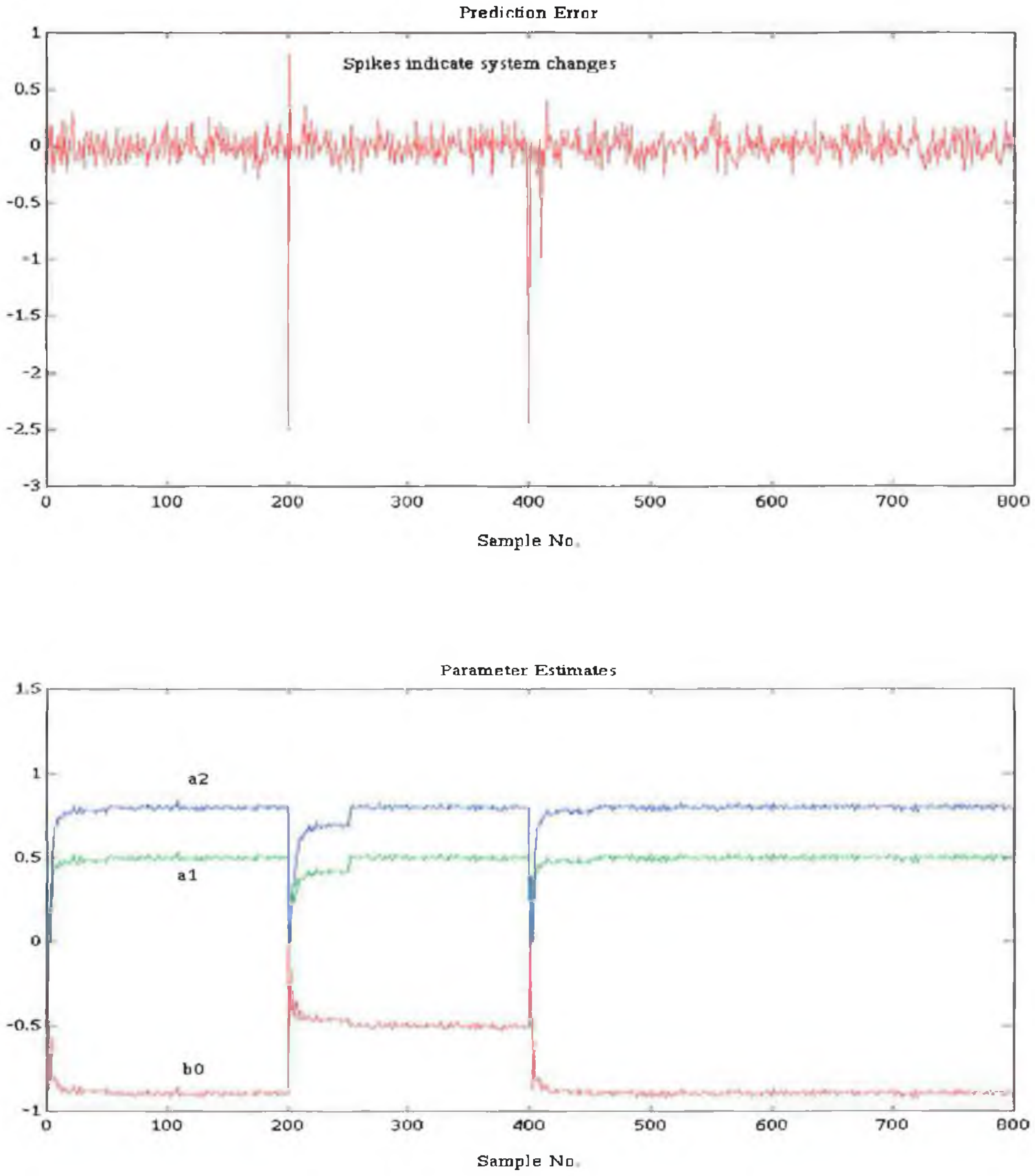


Fig. 3.8 Estimation using simple fault detection.

3.9 SUMMARY

The identification of dynamic systems has been discussed. An emphasis has been placed on time-varying systems and methods to cope with this time-variance within the system identification field have been outlined. The method of implementing fault detection algorithms in parallel with the estimation scheme has been presented as an alternative to the recommended methods and it has been shown that this type of scheme produces better variance/convergence properties than the afore mentioned methods. This method will be altered in the next chapter to cope with the specific problems encountered in the motorcycle suspension problem.

Chapter 4

SYSTEM SIMULATION.

4.1 INTRODUCTION.

In this chapter a simulation of the M.S.D. model of the suspension unit (chapter 2) is implemented. Data taken from this model is used to develop the alternative parameter tracking scheme introduced in chapter 3. The performance of this algorithm is then compared with that of conventional tracking schemes.

4.2 SYSTEM SIMULATION.

The M.S.D. model presented in chapter 2 contains three parameters of interest: mass, stiffness and damping. It was necessary to obtain approximate values for these parameters to simulate the dynamics of the suspension unit.

Static load tests were performed on the shock absorber depicted in Fig. 2.3. These tests amounted to placing a series of weights onto the shock absorber and measuring the resulting displacement. Given Hooke's law [56],

$$F = kx \qquad \text{Equ 4.1}$$

where F is the applied force, x the resulting displacement and k the spring stiffness, the shock absorber static spring stiffness was easily calculated. The system mass was calculated by weighing the shock absorber and a suitable damping ratio was assumed[1]

These tests yielded the following parameter values,

(i) Mass	M = 2kg,
(ii) Stiffness	K = 12.25kN/m,
(iii) Damping	B = 900Ns/m.

[1] Damping ratio obtained after consultation with Aengus Murray and Tom' O Kane of Team Kenny Roberts.

Insertion of these values into Equ. 2.3 leads to a differential equation description of the shock absorber dynamics,

$$2\ddot{x}+900\dot{x}+12250x=F \qquad \text{Equ. 4.2}$$

As noted in Chapter 3 modern estimation techniques require a discrete version of this continuous time representation. With this in mind equation 4.2 was discritised using a zero order hold [30]. The system sample rate was 100Hz.

This resulted in the following ARX model of the discrete shock absorber dynamics.

$$y(k)-0.8816y(k-1)+0.0112y(k-2)=$$
$$1x10^{-5}[0.8383u(k-1)+0.2197u(k-2)] \qquad \text{Equ. 4.3}$$

In Equ. 4.2 u is the system input, in this case samples of the applied force, and y is the system output, samples of the resulting displacement.

4.3 MODEL ESTIMATION.

The main requirement of any identification scheme is the system input/output information. As such detailed care and attention must be given to the selection of an appropriate system input and the necessary "pre treatment" of the collected data.

The question must be asked what is an appropriate input signal ? The answer to this may seem trivial and in cases where the choice of input is limited to the system operating conditions this may be so. However when this is not the case much debate arises concerning the best choice of input. It is important that the selected input excite all modes of the system dynamics. An input which succeeds in doing this is said to be persistently exciting and is obeying the condition of "persistent excitation" [23], [26], [45]. This condition is closely related to the concept of informative data sets which recommends that the input/output data be informative enough to be able to distinguish between two models within the same model set [24]. Theoretically speaking the optimal input is a white noise sequence though in most cases the physical constraints imposed by the system exclude this type of signal from consideration. The most common signal used in industry is the Pseudo Random Binary Sequence (PRBS). This signal is widely used in on-line applications. Wellstead & Zarrop [20] recommend a square wave input of frequency approximately 0.16 times the system bandwidth as a suitable choice.

The criterion laid out above suggests that a 1 Hz. square wave of sufficient magnitude will excite the shock absorber dynamics adequately. The stroke of the shock absorber depicted in Fig. 2.1 is 12cm. Given that the stiffness of the spring is 12.25kN/m it would take a force of 1.47 kN to fully compress the shock absorber. Applying a square wave of this magnitude and of frequency 1 Hz to the ARX model of the shock absorber (Equ. 4.2) results in the data given in Fig. 4.1

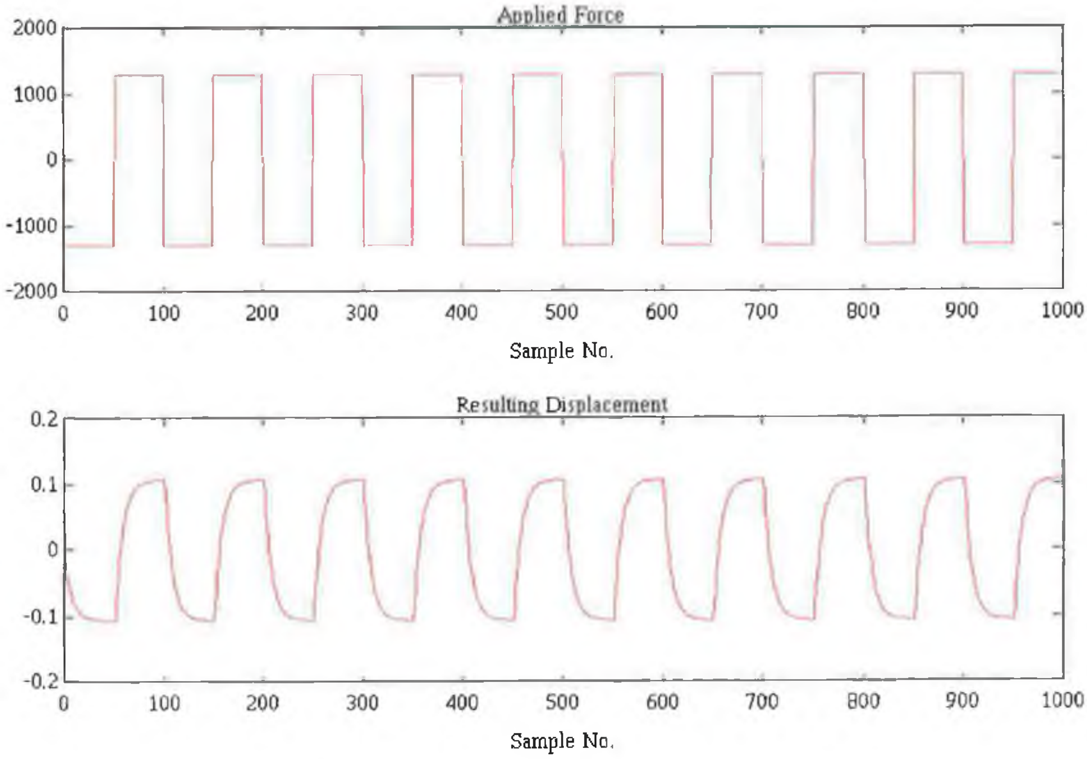


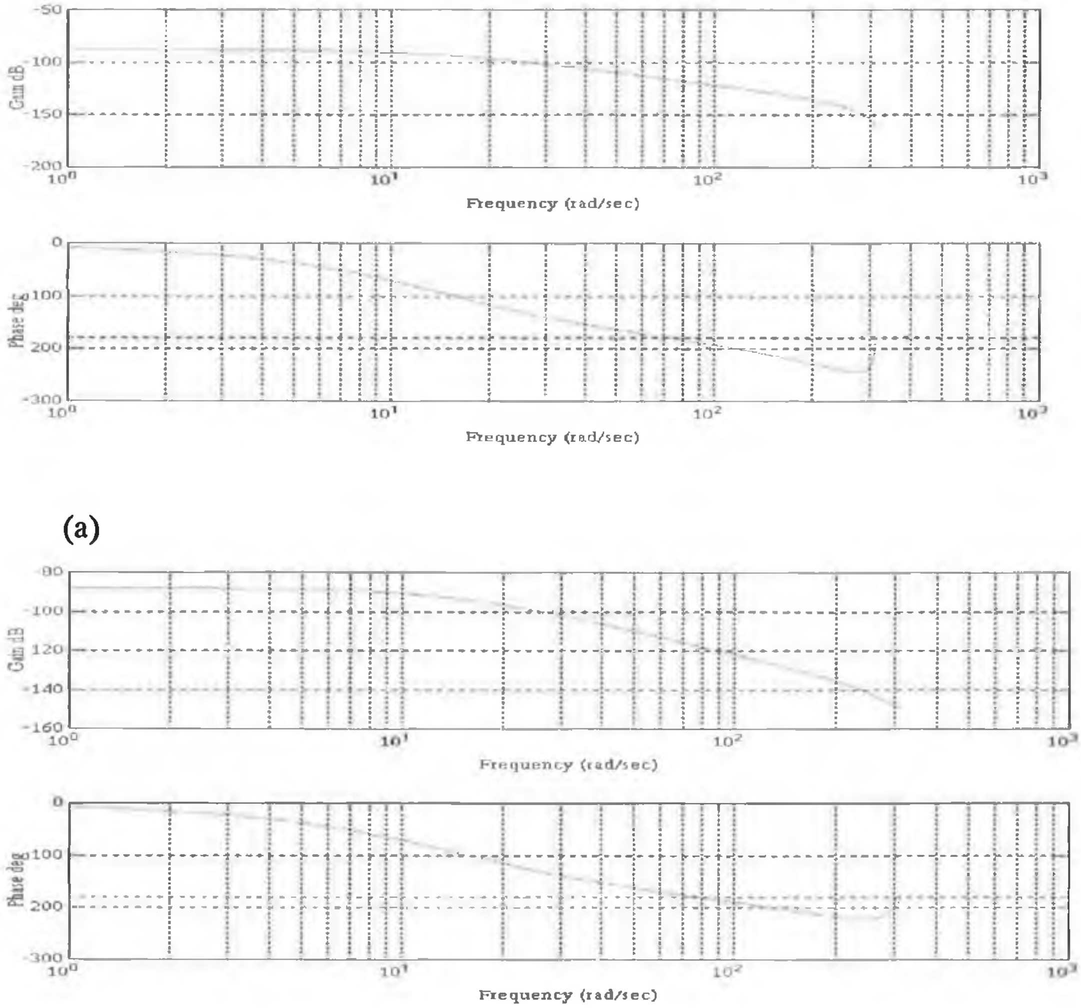
Fig 4.1 Simulated suspension data.

The simulation data depicted in Fig. 4.1 needs to be conditioned before it can be used to identify the ARX model of Equ. 4.3. Under normal circumstances this amounts to,

(i) Filtering: Most data is corrupted by sensor noise and outside disturbances. Application of a low-order, low-pass Butterwoth filter will clean the collected data sufficiently for the next stage in the conditioning process.

(ii) Detrending: The structures of the models used for identification correspond to dynamic models (which express output variations as a function of input variations around an operating point). It is therefore necessary, for correct identification, to eliminate the DC components (corresponding to the operating point) from the input/output data.

(iii) Scaling: If there are differences in the magnitudes of the input/output data (as is the case here) then either one or the other must be scaled (usually the smaller of the two) to achieve magnitude parity. If this task is not performed the estimator will perform sluggishly.



(a)

(b)

Fig 4.2 Transfer Function (a) Actual System, (b) Estimated System.

Since the input/output data depicted in Fig. 4.1 came from a noise free simulation step(i) was ignored. The chosen excitation signal has no DC level therefore step (ii) was ignored. There is a large magnitude difference between the system input and output. To compensate for this the output data was multiplied by a factor of 5. A batch least squares estimator as described in chapter 3 was applied to this data. The estimated system bode plot and actual system bode plot are given in Fig. 4.2.

Closer comparison of the actual system parameters and the estimated system parameters is achieved by examining the actual and estimated transfer functions,

$$G(z)_{\text{actual}} = \frac{0.8383e-5z+0.2197e-5}{z^2-0.8816z+0.0112} \qquad \text{Equ. 4.4}$$

$$G(z)_{\text{estimated}} = \frac{0.8382e-5z+0.2222e-5}{z^2-0.8794z+0.0093} \qquad \text{Equ. 4.5}$$

Any differences between the actual and estimated system which may not be apparent on comparison of their Bode plots becomes evident when their transfer functions are examined. However any differences between these are negligible. The batch least squares estimator performed well in this case.

4.4 SYSTEM TRACKING

In the previous section the time-invariant system model for the shock absorber was presented. However as stated previously (chapter 2) the composition of the suspension system renders it time-varying. This necessarily means that efforts should be made to track these system variations. The batch least squares estimator illustrated in use above is by its non-recursive nature incapable of giving current estimates of the system parameters let alone tracking them.

4.4.1 Parameter tracking using the R.L.S. algorithm

An obvious solution to the former problem is to implement a recursive estimator as outlined in section 3.5. Before discussing the estimation/tracking of a time-varying suspension it will prove helpful to illustrate the estimation of the time-invariant suspension unit using the R.L.S. algorithm (section 3.5). Rather than restricting this discussion to a single shock absorber the full suspension comprising two shock absorbers and approximately half of the weight of the motorcycle was considered.

The transfer function for this system is given as

$$G(s) = \frac{1}{Ms^2 + Bs + K} \qquad \text{Equ. 4.6}$$

In this instance M represents approximately half of the mass of the motorcycle plus the weight of both shock absorbers which constitute the full front suspension. B and K are the overall system stiffness and damping respectively. These are calculated by simply summing the stiffness and damping contributions of each shock absorber.

Given additional data[1] concerning the weight of the motorcycle and parameter values obtained during static load tests (section 4.2) the full front suspension transfer function is given as,

$$G(s) = \frac{1}{115s^2 + 3021s + 24500} \qquad \text{Equ. 4.7}$$

or in discrete form,

$$G(z) = \frac{-0.3650x10^{-5}z + 0.3984x10^{-5}}{z^2 - 1.7503z + 0.769} \qquad \text{Equ. 4.8}$$

Since the full front suspension system consists of two shock absorbers it requires twice as much force to displace the suspension fully. Therefore a square wave twice that in magnitude but of equal frequency to the input applied to the single shock absorber model in section 4.2, was applied to the full front suspension model.

With this in mind the simulated data given in Fig. 4.1 was utilised by a R.L.S. estimator to estimate the simulated model parameters. The estimator was initialised with a zero initial condition on the parameter estimate vector $\hat{\theta}$ and a diagonal P matrix with the value 10 along the main diagonal. The resulting parameter estimates are illustrated in figure 4.3 (overleaf).

The data depicted in Fig. 4.3 was conditioned as outlined in section 4.2 and a recursive least squares estimator was applied to the resulting pre-treated data. The estimator was

[1] Additional data obtained from T. O' Kane of *Team Kenny Roberts*.

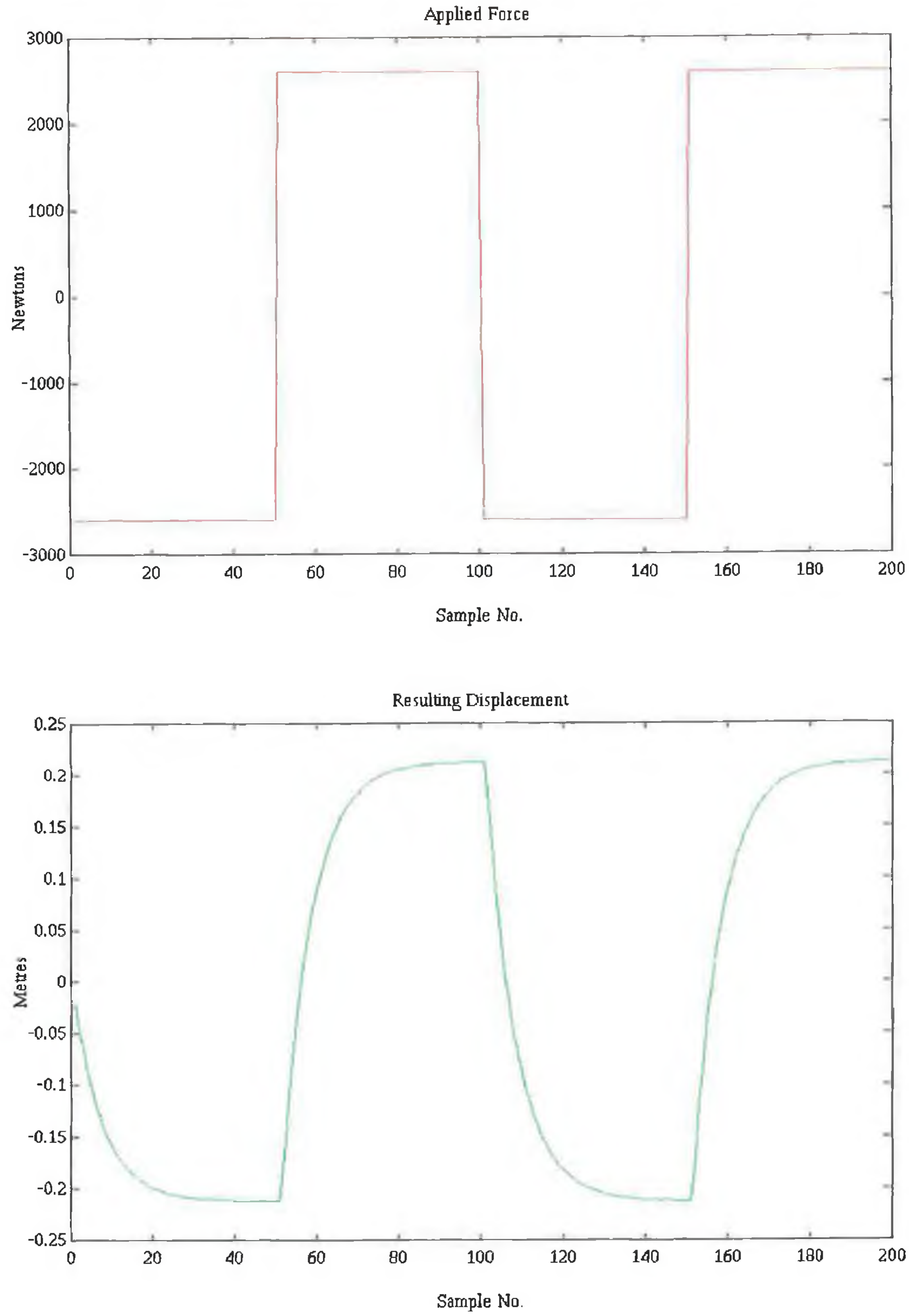


Fig. 4.3 Simulated full front suspension data.

initialised with a zero initial condition on the parameter estimate vector $\hat{\theta}$ and a diagonal P matrix with the value 10 along the main diagonal. The resulting parameter estimates are illustrated in Fig. 4.4 (note the b-parameters are scaled up by a factor of

5. This is caused by the scaling of the system output during the pre-treatment phase of the identification experiment).

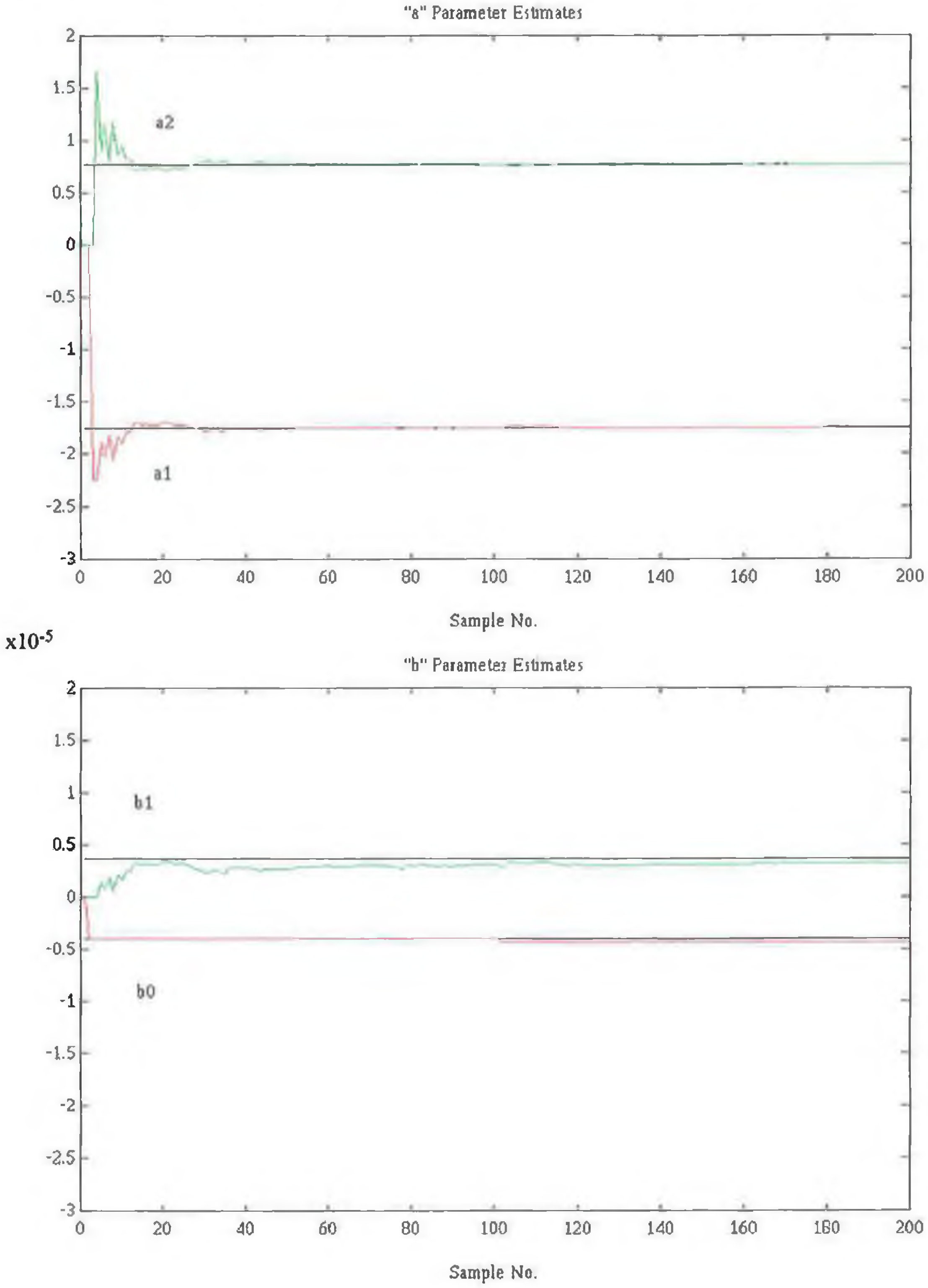


Fig. 4.4 Parameter estimates using ordinary R.L.S.

These plots indicate that the estimates converge within 30 samples. are consistent (meaning the estimates converge to the actual parameter values) [23], [24], [26] and unbiased. As stated previously the R.L.S. estimator works well when the system is stationary. However under non-stationary conditions it is usual to find a severe degradation in the performance of the estimator (see chapter 3); specifically the estimates become biased. To quantify the performance of the R.L.S. estimator when confronted by time-variance, the following time-varying full suspension unit model was considered

$$G(s) = \frac{1}{Ms^2 + Bs + K}$$

M = system mass

B = system damping

K = system stiffness

ζ = damping ratio

$0 \le t \le 200$	M = 115kg	B = 3021Ns/m	K = 24500N/m	$\zeta = 0.9$
$201 \le t \le 400$			K = 18000N/m	$\zeta = 1.04$
$401 \le t \le 500$		B = 3453Ns/m		$\zeta = 1.2$
$501 \le t \le 600$		B=2123Ns/m	K = 20000N/m	$\zeta = 0.69$
$601 \le t \le 800$	M = 30kg			$\zeta = 1.37$

This set of figures[1] represents a suspension system which undergoes some very severe and instantaneous changes. This scenario is a severe test for any tracking scheme. The parameter estimates obtained using the ordinary R.L.S. algorithm are shown in Fig. 4.5 (overleaf).

Once again the estimator quickly converges to the correct parameter values, however once the system reaches its initial change at the 200th. sample the estimates become biased. The result is a group of estimates of no real value.

[1] Obtained through consultation with Aengus Murray and Tom O' Kane of Team Kenny Roberts.

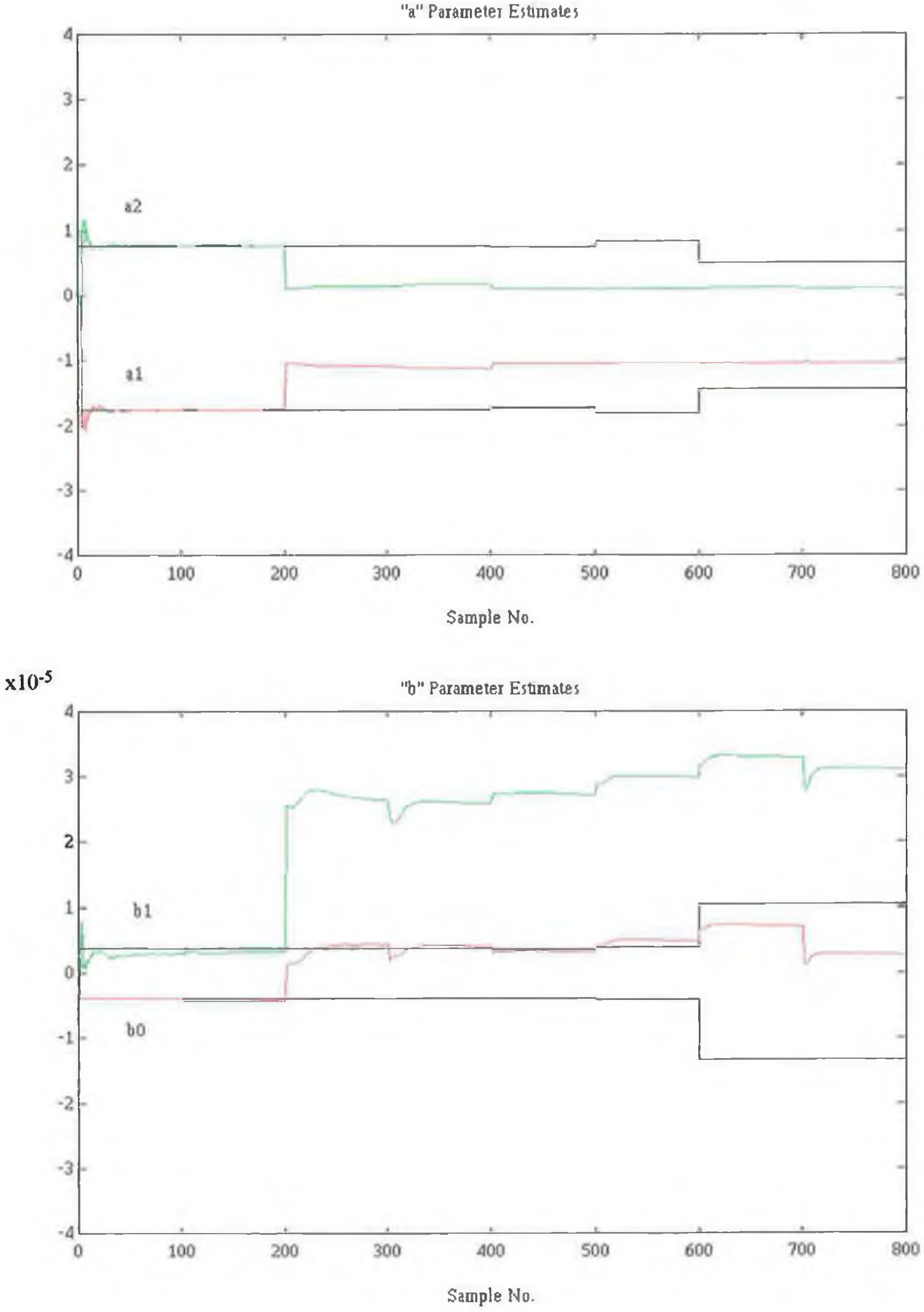


Fig 4.4 "Tracking" using the ordinary R.L.S. estimator.

4.4.2 Parameter estimation using the alternative tracking scheme.

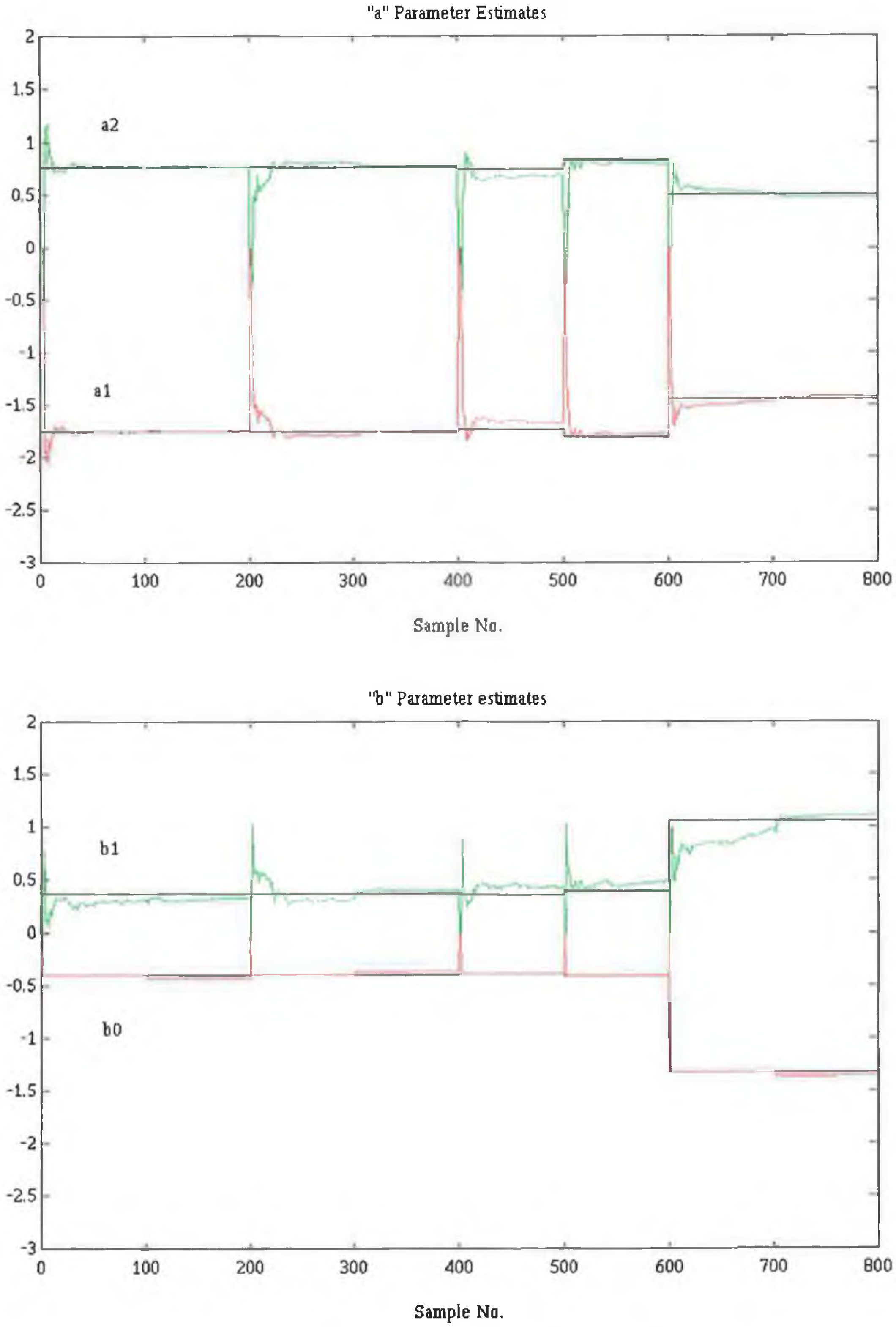
The recommended adjustments to the ordinary R.L.S. algorithm, i.e. the forgetting factor modification and Kalman filter ,were discussed in sections 3.6 and 3.7 respectively. It was concluded that both of these algorithms yielded estimates with poorer variance properties for stationary systems and that they both tackled the time-variance issue in a rather passive manner assuming that the system was constantly changing.

Having recognised the passivity of these algorithms an alternative algorithm which incorporated a fault detection scheme was introduced. This algorithm monitored the prediction error sequence as a means to detect when the system dynamics had altered and was tested using a simple system simulation. The resulting estimate sequences possessed better variance and convergence properties than those obtained using the more common alternatives to the R.L.S. algorithm.

To ascertain the validity of this algorithm with respect to a motorcycle suspension system, the prediction error based scheme was applied to the time-varying system outlined above. The parameter estimate vector $\hat{\theta}$ was initialised to zero while P was set as a diagonal matrix with 10 along the main diagonal. This value of P assigns equal uncertainty to all of the initial parameter estimates. The magnitude of relative uncertainty (10 in this case) reflects that although these estimates are not accurate, they are not outlandish. A ceiling of 1500 units was placed on the prediction error sequence. Once the error went above this ceiling the estimator was reset. The resulting parameter estimates are depicted in Fig. 4.6 (overleaf).

Fig. 4.6 illustrates that the prediction error based fault detection/estimation scheme works well when applied to the time-varying suspension model. The system faults are recognised almost immediately and once the estimator has been reset the estimates reconverge within a matter of samples.

The key to this algorithm is the prediction error sequence, depicted in Fig. 4.7 which triggers the estimate into a reset mode. This fault detection scheme uses the magnitude of this sequence as a figure of merit for the current parameter estimates. Once the magnitude of the error surpasses the ceiling provided by the user, the detection scheme realises that the system is changing and that, not only are the previous parameter estimates incorrect, but the P matrix is conveying too much confidence in these old estimates. At this point the estimator is reset. This effectively means that the R.L.S. estimator is estimating the parameters of what it now sees as a "new system". Once again the parameter estimate vector is set to zero and the P



"a" Parameter Estimates

"b" Parameter estimates

Fig 4.6 Estimates using the prediction error tracking mechanism.

matrix has 10 along the main diagonal reflecting low confidence in the initial estimates. In theory the estimates should converge to their true values as quickly as if the estimation process had just begun.

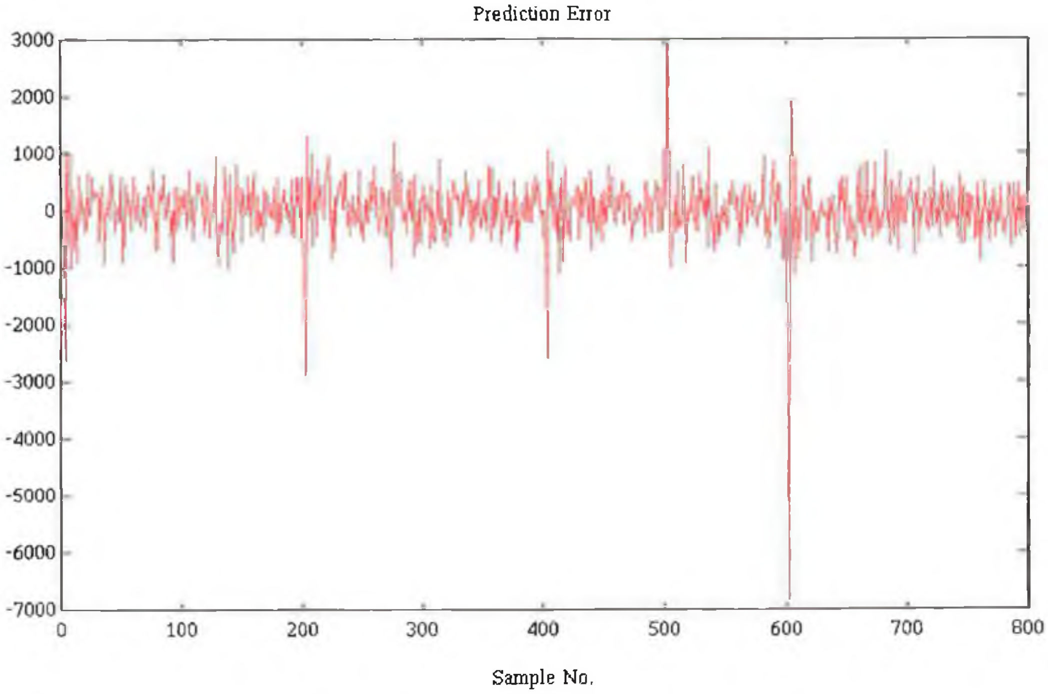


Fig. 4.7 The prediction error sequence for the time-varying shock absorber.

4.4.3 Parameter estimation using an improved alternative algorithm.

The alternative tracking scheme performs well in tracking the time-varying suspension system. However this algorithm is not without its drawbacks.

(i) There are visible spikes in the estimate sequence and these are undesirable in control situations. These spikes are caused by the resetting of the estimator to initial conditions once a fault has been detected. Since the initial condition for each parameter estimate sequence is zero it seems obvious that there would be a spike. To remove this anomaly from the estimate sequence a simple solution is to just leave the estiamte values as they are and simply increase the size of P, the parameter estimate covariance matrix. This is akin to modelling the system parameters as random walks but only in the vacinity of the system changes.

(ii) The prediction error is highly susceptible to outliers and system noise. It would therefore seem prudent to implement a more robust triggering mechanism than the simple prediction error sequence. Under normal operating conditions this sequence is white with zero mean; however when the system dynamics change so too will the error statistics. In the simplest scenario the mean of the prediction error would be monitored for large fluctuations about zero. The occurrence of a fault would then be determined by the occurrence of such deviations from the steady state mean. However, examination of the mean error, for the time-varying system listed above, as plotted in Fig. 4.8 rules this out.

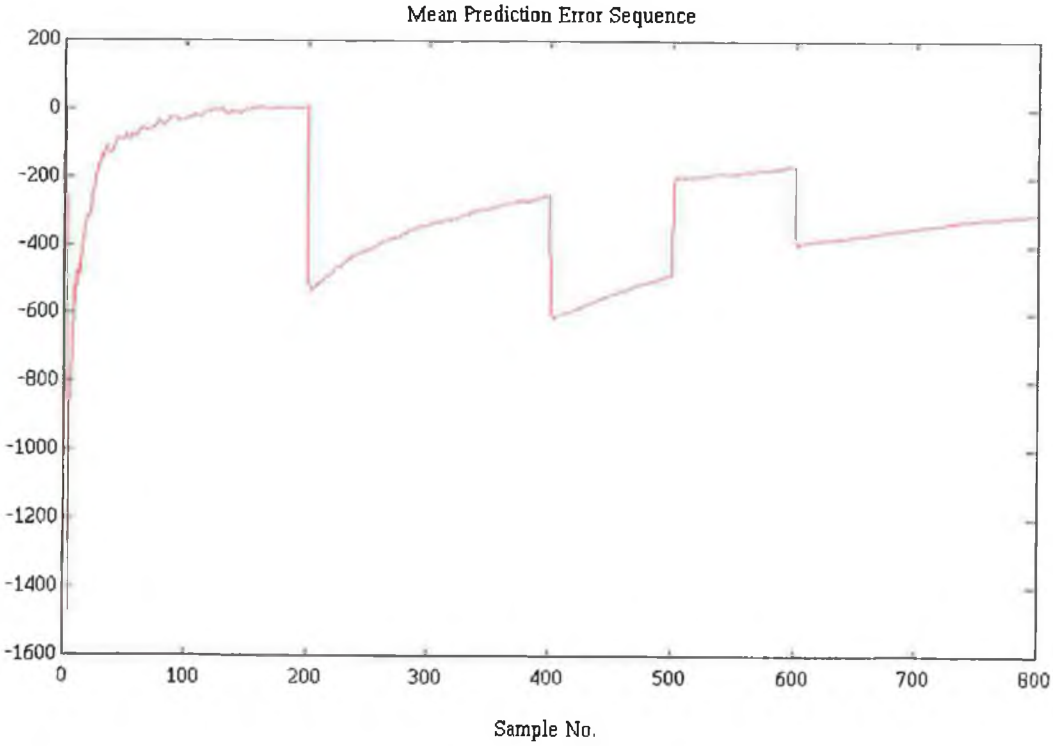


Fig. 4.8 The mean prediction error as an alternative triggering function ?

Such is the magnitude of the disturbance caused by system faults that the mean error never actually reconverges to its steady state value i.e. zero. In fact it would have difficulty reaching a level above the ceiling which determines when a fault has occurred. The fault detection scheme would interpret this as evidence that the system is constantly changing. This means that the parameter estimate covariance is constantly being reset to its initial value. The resulting algorithm would at best yield estimates equivalent to those produced by the Kalman filter.

However this problem is not unsurmountable. Under normal circumstances the mean prediction error is calculated as a running average by the fault detection scheme. The fault detection scheme relies upon the fact that when the system changes the mean between detectable faults to 0.04s. However this can be counteracted by increasing the system sample rate. In summary this algorithm works as follows

(i) Initialise the parameter estimate vector and the covariance matrix P.

(ii) Set the mean prediction error ceiling.

(iii) Start estimation.

(iv) If a fault is detected reset P and set the mean error to zero. After four samples, calculate the mean as normal.

The algorithm outlined above was applied to the system data used in previous attempts to track the suspension dynamics. $\hat{\theta}$ and P were initialised as in section 4.4.2. The ceiling for the triggering function was chosen as 300 units. (In practical situations initialisation of the triggering ceiling is aided by an examination of the innovation sequence for a particular identification experiment using R.L.S. A rule of thumb is to assume a triggering ceiling equal in magnitude to the maximum value of the innovation. A optimum ceiling is then found through suitable "tweaking"). Once the magnitude of this function rose above this value the mean function was forced to zero (thus removing control from the fault detection algorithm) and P was increased (note the estimator was not reset). After a period of 4 samples the mean function was calculated as normal (control given back to the fault detection algorithm). Fig. 4.9 depicts this "mean" triggering function (as discussed above) which was used to trigger the ordinary R.L.S. estimator into a tracking mode. Of note is the crispness of this function compared to the noisy prediction error sequence depicted in Fig. 4.7 which was used as a triggering function in earlier attempts to track the system dynamics. Because of this "crispiness" the rate of false alarms is negligible further increasing the overall robustness of the algorithm.

The parameter estimates for this simulation are illustrated in Fig. 4.10. Of note in these plots are the excellent variance properties of these estimates. The low variance properties are attributable to the fact that the estimator treats the system as time-varying only when it *is* actually time-varying. Also of note is the lack of spikes in the estimate sequences which is due wholly to leaving the estimates as they are instead of resetting the estimator as before.

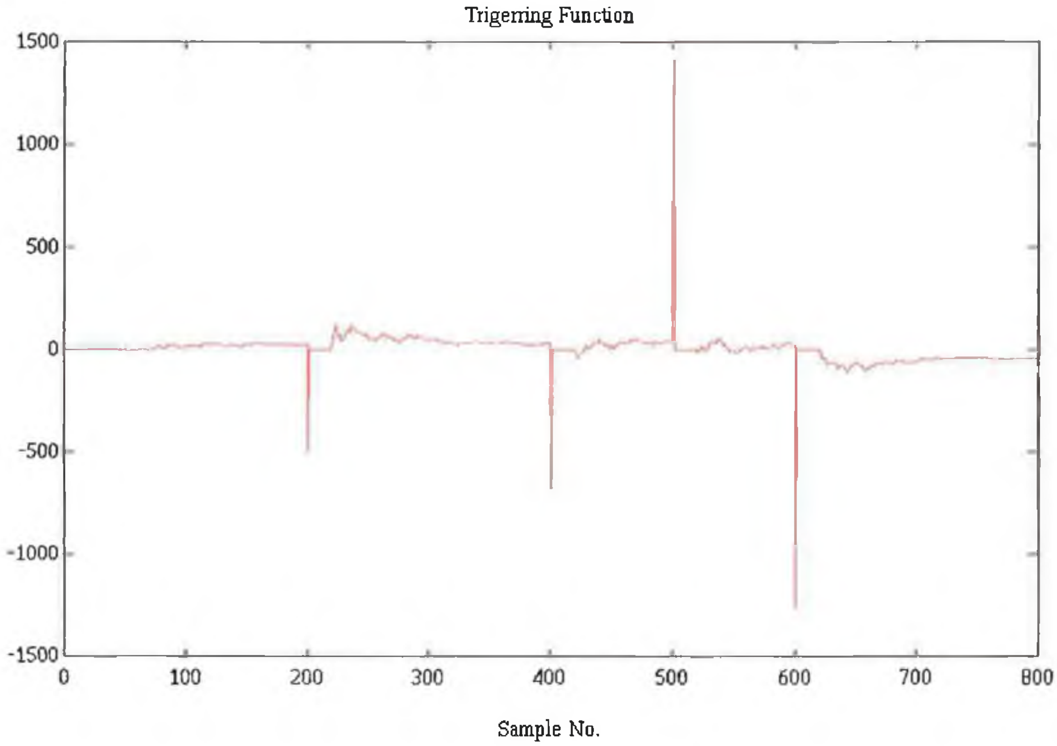


Fig. 4.9 The mean between faults as a triggering function !

4.5 SUMMARY.

Both the time-invariant and time-varying suspension units were simulated. The time-invariant system model was identified/estimated successfully using both batch and recursive forms of the least squares algorithm. The tracking capability of the R.L.S. estimator proved insufficient when confronted with the time-varying system. The fault detection scheme proposed in chapter 3 was preferred to the forgetting factor modification and the Kalman filter as it possesses better variance and convergence properties when implemented on a time-varying example. This algorithm was modified to make it more robust with respect to bad data and outliers. The resulting estimate sequences possessed better variance and convergence properties.

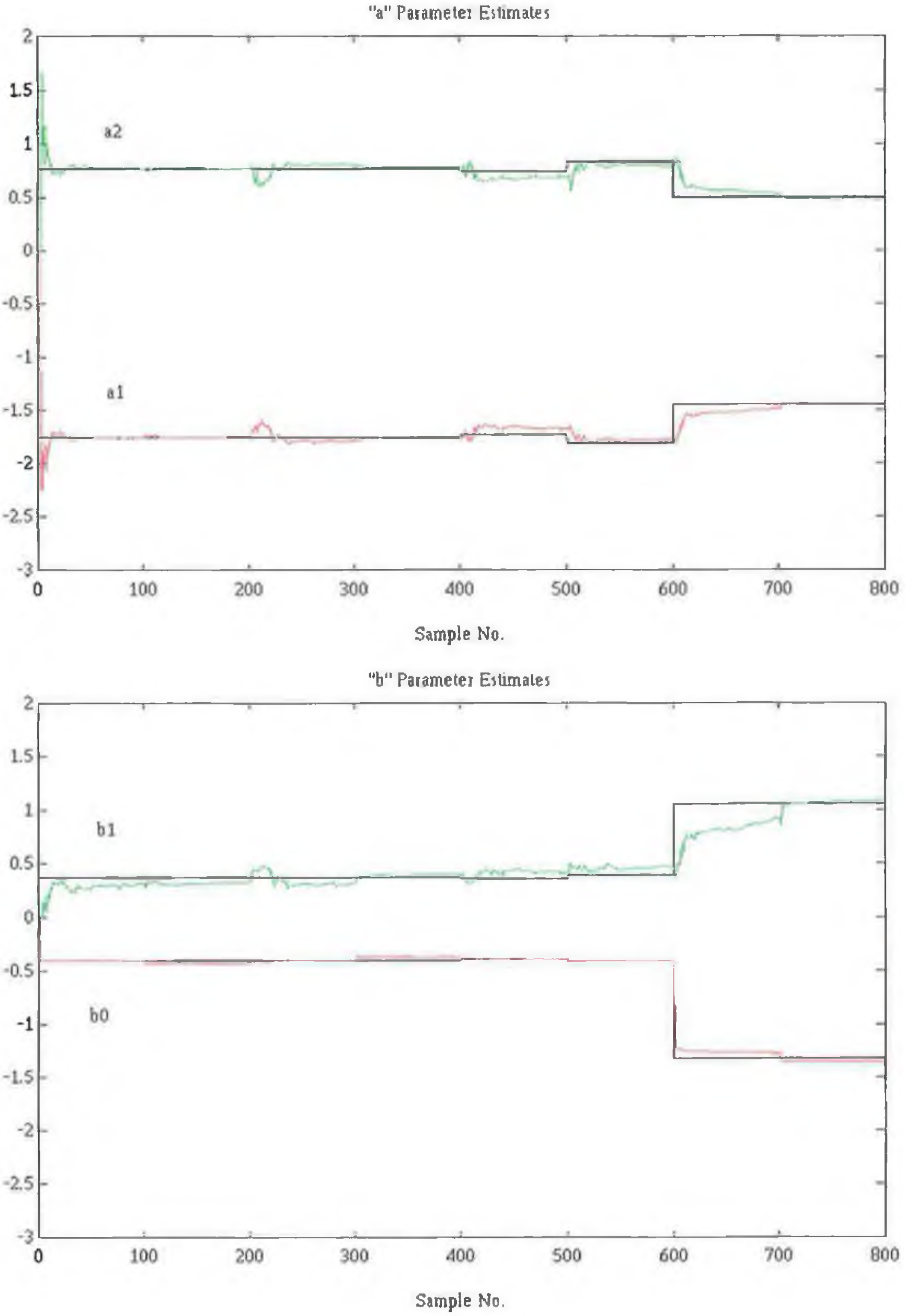


Fig. 4.10 Parameter estimates using the mean between faults triggering function.

Chapter 5.

SYSTEM IMPLEMENTATION.

5.1 INTRODUCTION.

The simulation and estimation/tracking of the time-varying suspension was undertaken in chapter 4. Resource constraints prevent the application of the resulting estimation techniques to an actual suspension system. However this chapter details the construction of an electrical analogue of the suspension system. The modified tracking algorithm presented in chapter 4 is applied to data taken from this analogue. The performance of this algorithm is evaluated in relation to the performance of the more popular tracking schemes.

5.2 AN ELECTRICAL ANALOGUE.

It is a common occurrence in the control industry to have limited access if any to the system under consideration. In cases where there is relevant information available concerning the system, such as its order and its physical components, it is possible to construct an analogous system. This occurrence is most common when confronted with mechanical systems. There is a wealth of background knowledge relating mechanical systems to analogous hydraulic and electrical systems. For brevity's sake this work will concern itself specifically with an electrical analogue of the suspension unit under investigation.

The principles of analogy lie in the fact that different physical systems can be described by similar if not the same mathematical model. If two systems can be described by the same differential equation they are said to be analogies of each other [54]. As an example the following mechanical and electrical differential equations were considered.

$$f = M\frac{dv}{dt} \qquad \text{Equ. 5.1}$$

$$i = C\frac{de}{dt} \qquad \text{Equ. 5.2}$$

Equ. 5.1 is Newtons second law of motion. This relates the acceleration, a, of a mass, M, to the force f applied to this mass. Equ. 5.2 relates the capacitance, C of a capacitor to the voltage across its terminals, e, and the current i. These equations are quite similar. As such the capacitor described by Equ. 5.2 is an anlogue of the mass described by Equ. 5.1. The force acting on the mass M is analogous to the current i flowing through the capacitor C and hence this analogy is referred to as a force-current analogy [54]. This analogy was used to design an electrical analogue of the M.S.D. model (section 2.4) of the suspension.

Mechanical systems are usually represented diagramatically as free body diagrams. The M.S.D. representation of the suspension unit is an example of this. However as with electrical systems it is possible to illustrate this system using a network of the relevant physical devices. Fig. 5.1 depicts the mechanical network for the simple M.S.D. model.

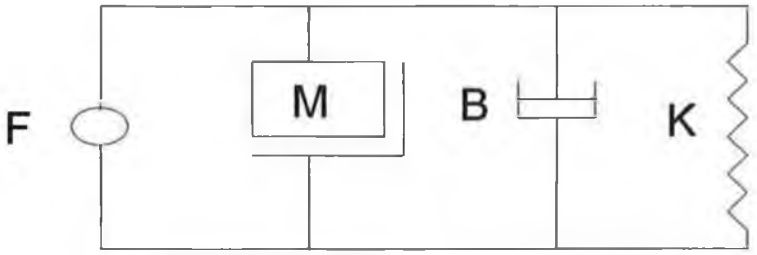


Fig. 5.1 Mechanical network for the M.S.D. model.

An analogous electrical system can be implemented using the force current analogy outlined above .This analogous electrical system consists of the electrical circuit of Fig. 5.2. Application of Kirchoffs current and voltage laws yield equations which describe not only the action of the electrical circuit but also that of the mechanical network of Fig. 5.1.

Unfortunately it proved impossible to implement the circuit depicted in Fig. 5.2 as the equivalent capacitance necessary to represent the system mass is of an impractical value. This is not to say that the system does not have an electrical analogue. Moreover the system can be represented using an analogue computer.

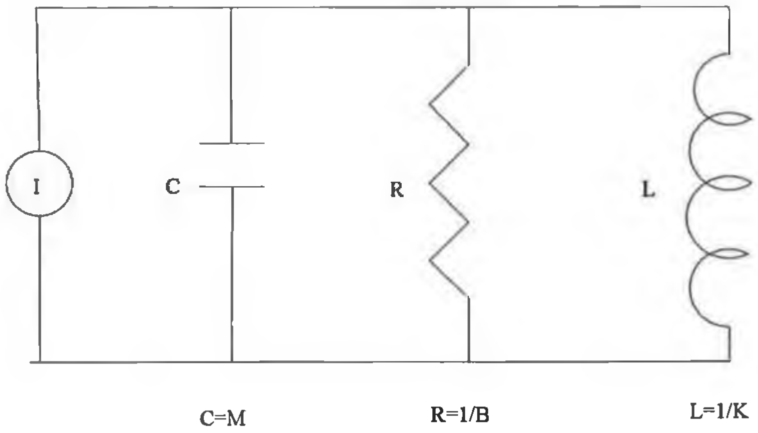


Fig 5.2 An electrical analogue of the M.S.D model.

The basis of an analogue computer is the system differential equation. Once this is known a series of integrators connected together can be implemented to reconstruct not only the system input/output data but also the intermediate states of velocity and acceleration. From chapter 2 the suspension unit is describable using the following differential equation,

$$M\ddot{x} + B\dot{x} + Kx = F \qquad \text{Equ. 5.2}$$

This equation can be rewritten as

$$\ddot{x} = \frac{1}{M}[F - B\dot{x} - Kx] \qquad \text{Equ. 5.3}$$

The design procedure for the analogue computer proceeded as follows. Firstly it was assumed that the highest derivative in the systems differential equation, i.e. the acceleration, was available. This was then integrated to yield all of the system states. These were then added together in the correct proportion together with the system input as in Equ. 5.3 . This yielded the system acceleration which is what we started with. A block diagram illustrating the behaviour of this computer is shown in Fig 5.3.

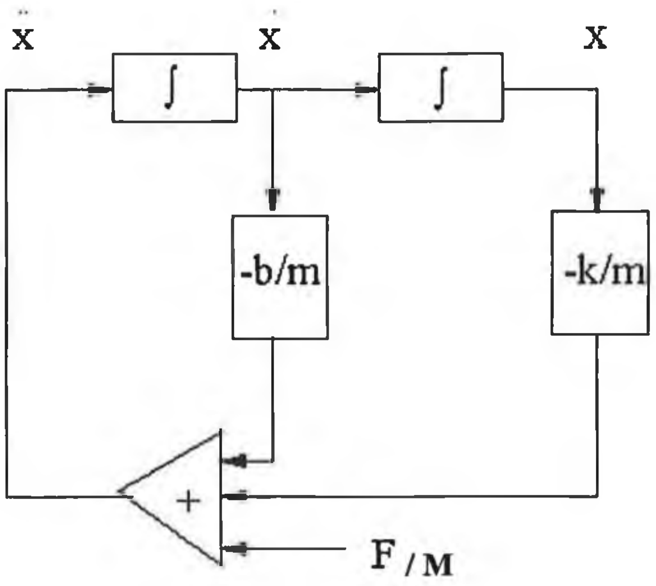


Fig. 5.3 The analogue computer representation of the shock absorber.

The analogue computer depicted in Fig. 5.3 was constructed as an electrical analogy for the motorcycle shock absorber system. The circuit layout is listed in Appendix. B1. The construction of the computer was quite simple. The integrators were constructed from operational amplifiers and capacitors whereas the adder and the gains were constructed again using operational amplifiers and resistors.

5.3 IMPLEMENTATION OF THE ELECTRICAL ANALOGUE.

The circuit depicted in Appendix B1 runs at 200 times the pace of the actual system so all the data must be frequency scaled if parity is to be gained with the simulations described in earlier chapters. The circuit input lies within the 0-10V range and as such both it and the circuit output must be scaled in magnitude to obtain data which is consistent with the suspension system.

Data was collected from the circuit via a Data Translation analog to digital interface card [55]. This card has four A/D channels with a throughput of 100 kHz at 12 bits per channel. The analog to digital converter implements a "simultaneous sample and hold" for each A/D channel ensuring that there are no lags between sampled input and output. Data from the card was accessed by the Hypersignal [56] software package. Since the electrical analog is operating 200 times faster than the simulated system it is sampled at 20 kHz. This is equivalent to the sampling rate of 100 Hz. used by the simulation. Figure 5.4 illustrates the simulation output and the analogue output (scaled) when a 1 Hz square wave is applied to the simulated model (ie. 200 Hz. to the analogue).

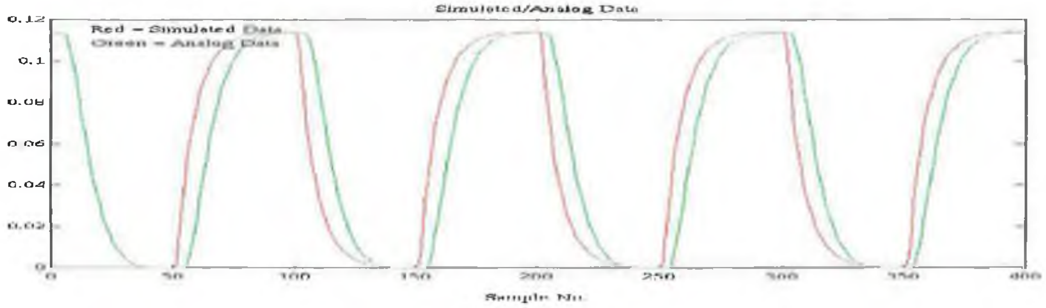


Fig. 5.4 A comparison of the simulated system output and the electrical analogue output.

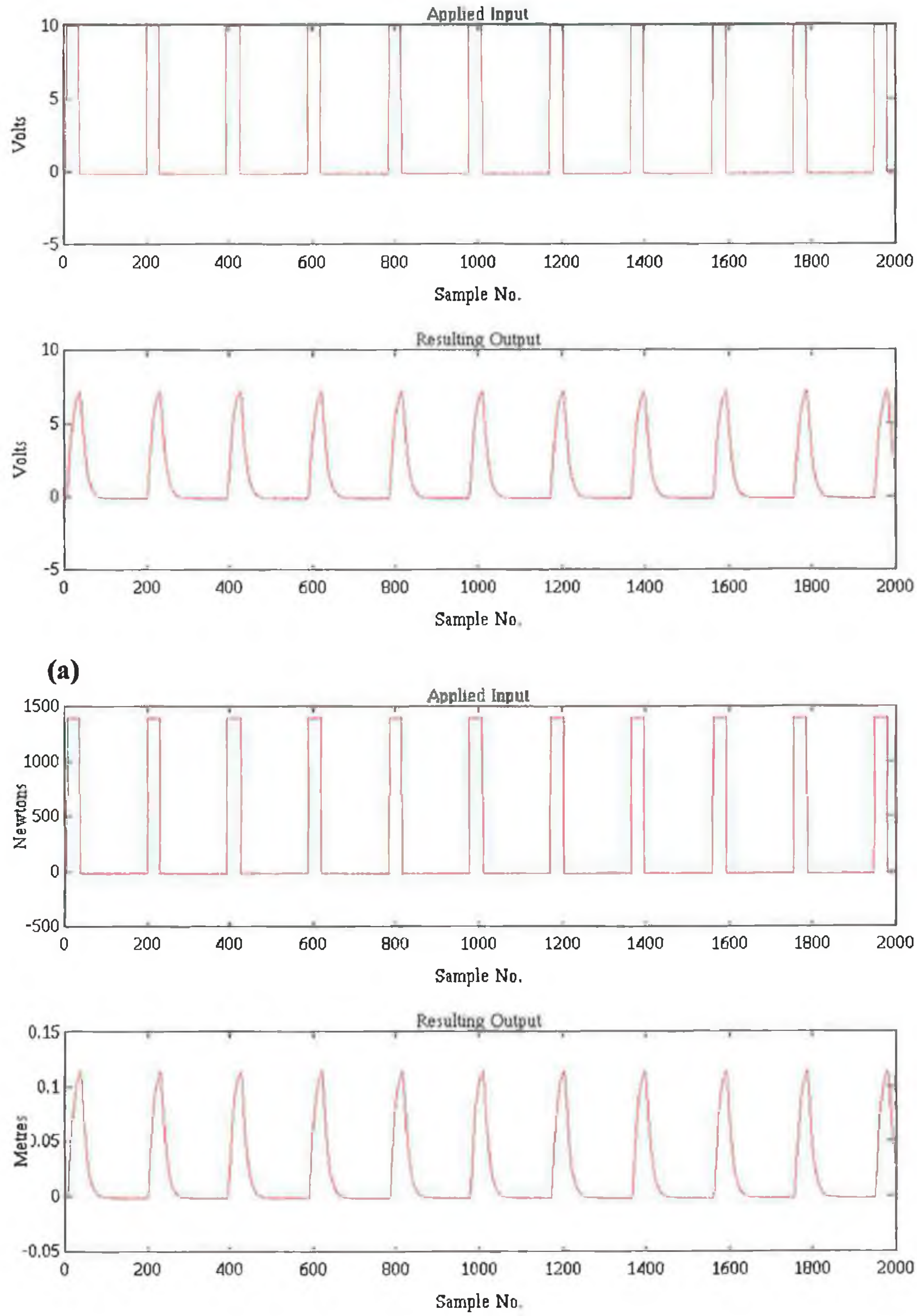
5.4 IDENTIFICATION/ESTIMATION OF THE ELECTRICAL ANALOGUE

At this stage the system electrical analogue was fully operational. Figure 5.5a illustrates some input/output data collected from the circuit as detailed in section 5.3. (Note in this section a 100Hz. wave is applied to the analogue computer). The applied input is a pulse waveform of magnitude 0-10V. This data is scaled to yield data which is consistent with data from the system simulation of section 4.2.

The first 1000 samples of this data were used to estimate the parameters of the suspension equivalent. The algorithm used was a batch least squares estimator. The resulting estimated transfer function is

$$G(z)_{\text{estimated}} = \frac{0.1817\text{x}10^{-5}z + 0.0766\text{x}10^{-5}}{z^2 - 1.5628z + 0.0766} \qquad \text{Equ. 5.4}$$

At this point it was necessary to verify that this transfer function represented the dynamics of the analogue computer (and hence the suspension equivalent). The covariance of the parameter estimate vector $\hat{\theta}$ suggested that these estimates were accurate (each parameter estimate had a variance less than 0.1). As a further test the prediction properties of $G(z)_{\text{estimated}}$ were examined. $G(z)_{\text{estimated}}$ was simulated in software. The second 1000 samples of the input data taken from the analog computer (plotted in Fig 5.5) were used by this simulation in an attempt to reproduce the second 1000 samples of output. The output resulting from the simulation of $G(z)_{\text{estimated}}$ is plotted with the actual analogu computer output in Fig. 5.6.



(a)

(b)

Fig. 5.5 (a) Circuit data, (b) scaled circuit/suspension data.

The plots in Fig. 5.6 indicate that there is good agreement between the output from the analogue computer and the simulation which was carried out using its estimated transfer function. This is a good indication that $G(z)_{estimated}$ is reliable.

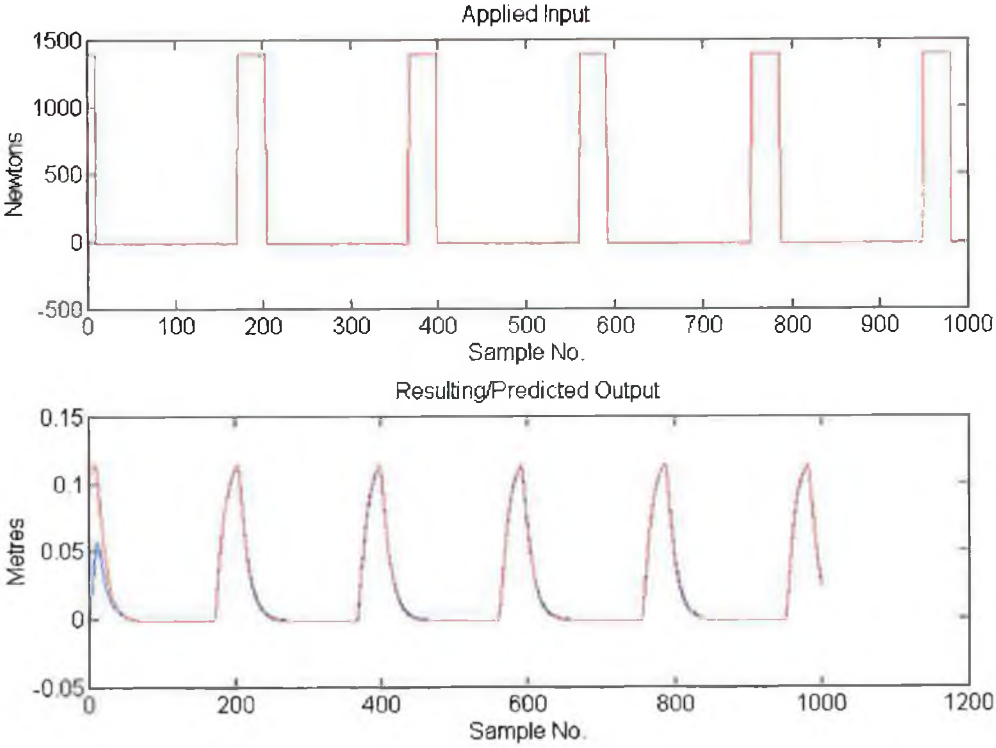


Fig. 5.6 Prediction properties of the estimated analogous transfer function.

5.5 TRACKING THE ELECTRICAL ANALOGUE

The stationary electrical system has now been identified. Following the route of the work detailed in chapter 4 it was necessary to track the changing parameters of the non-stationary circuit. The "non-stationarity" of the circuit is achieved by "tweaking" the various pots in the circuit illustration of Appendix B1. Application of a 100Hz. (1 Hz. on actual suspension system) square wave to the non-stationary circuit results in the data depicted in Fig. 5.7. It is visible from the data that both the system parameters and the input level change at various points.

The data plotted in Fig. 5.7 was fed into the tracking mechanism adopted for the simulation of chapter 4. For comparison purposes a batch least squares estimator was also implemented to give an indication of parameter values between system faults(see chapter 4). A ceiling of 500 units was placed on the triggering function as the point beyond which a fault is assumed to have occurred. It must be noted that the fault detection algorithm was switched off for the first 50 samples so as to help convergence to steady state values. The tracked parameters are plotted against these batch estimates in Fig. 5.8. Note the estimates are adjusted to reflect the dynamics of the actual suspension system and that the b parameters are offset to make them distinguishable.

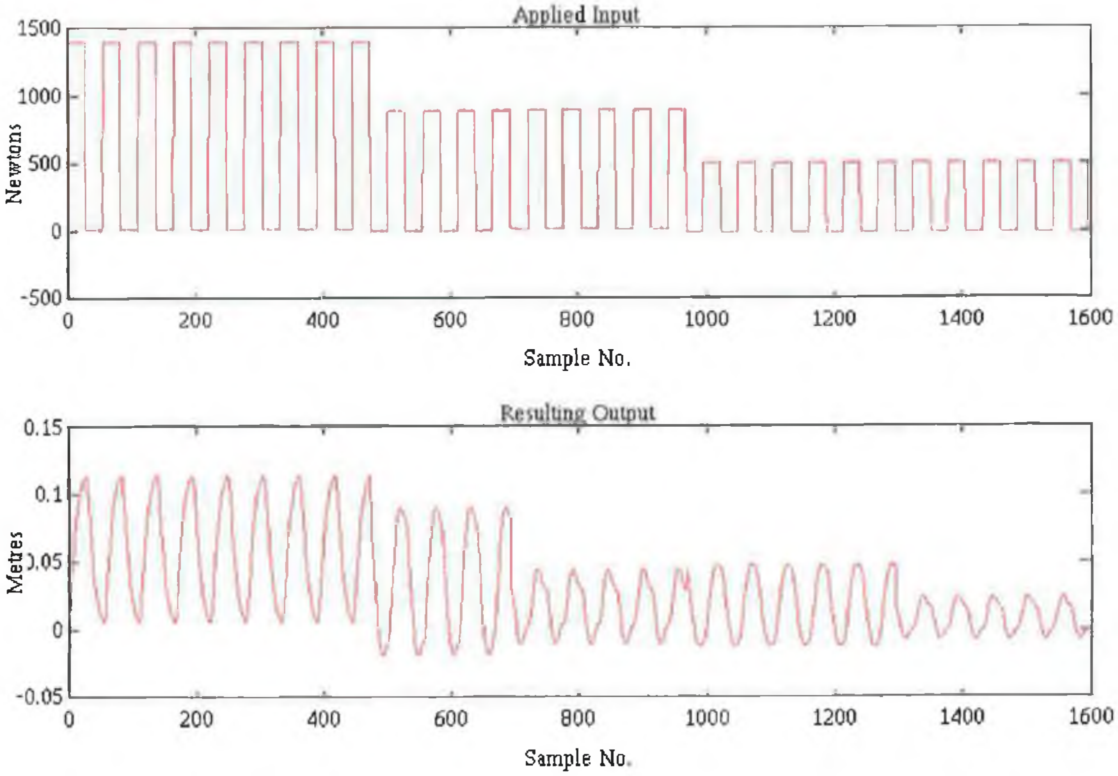


Fig. 5.7 Input/Output data from the time-varying electrical analogy.

5.6 ALGORITHM PERFORMANCE

To categorise the performance of the alternative tracking algorithm outlined in chapter 4 and implemented in section 5.6 the resulting parameter estimates must be compared with those obtained using the more traditional schemes, namely the forgetting factor and random walk algorithms. Fig. 5.9 illustrates the estimates obtained using the same data as that in the preceding section (only the b parameters shown). This time however the conventional algorithms have been used. It is seen from Fig. 5.9a that although the forgetting factor algorithm does track the system well ($\lambda = 0.97$) the variance of the resulting estimates is extremely poor in comparison to those obtained using the alternative algorithm depicted in Fig. 5.8 (page 68). The estimates provided by the random walk algorithm (R = identity matrix) are extremely poor.

The alternative algorithm outperforms both of the traditional parameter tracking mechanisms. This is in part due to the triggering function, depicted in Fig. 5.10a (page 70) and the passivity of the conventional algorithms. The prediction quality of the resulting estimates is depicted in Fig. 5.10b. Again the error between the system output and the predicted output is negligible.

It is of note that the presence of "spiking" has not been totally eradicated as in the simulations presented in chapter 4. However their presence in the parameter estimate sequence has been curtailed.

5.7 SUMMARY

An electrical analogue of the suspension unit was constructed. The parameters of this system were identified using a batch least squares estimator. The parameters of this analogue were varied and attempts to track theses parameter variations indicated that the algorithm proposed in earlier chapters which incorporates a fault detection mechanism out-performs the more traditional forgetting factor and random walk approaches.

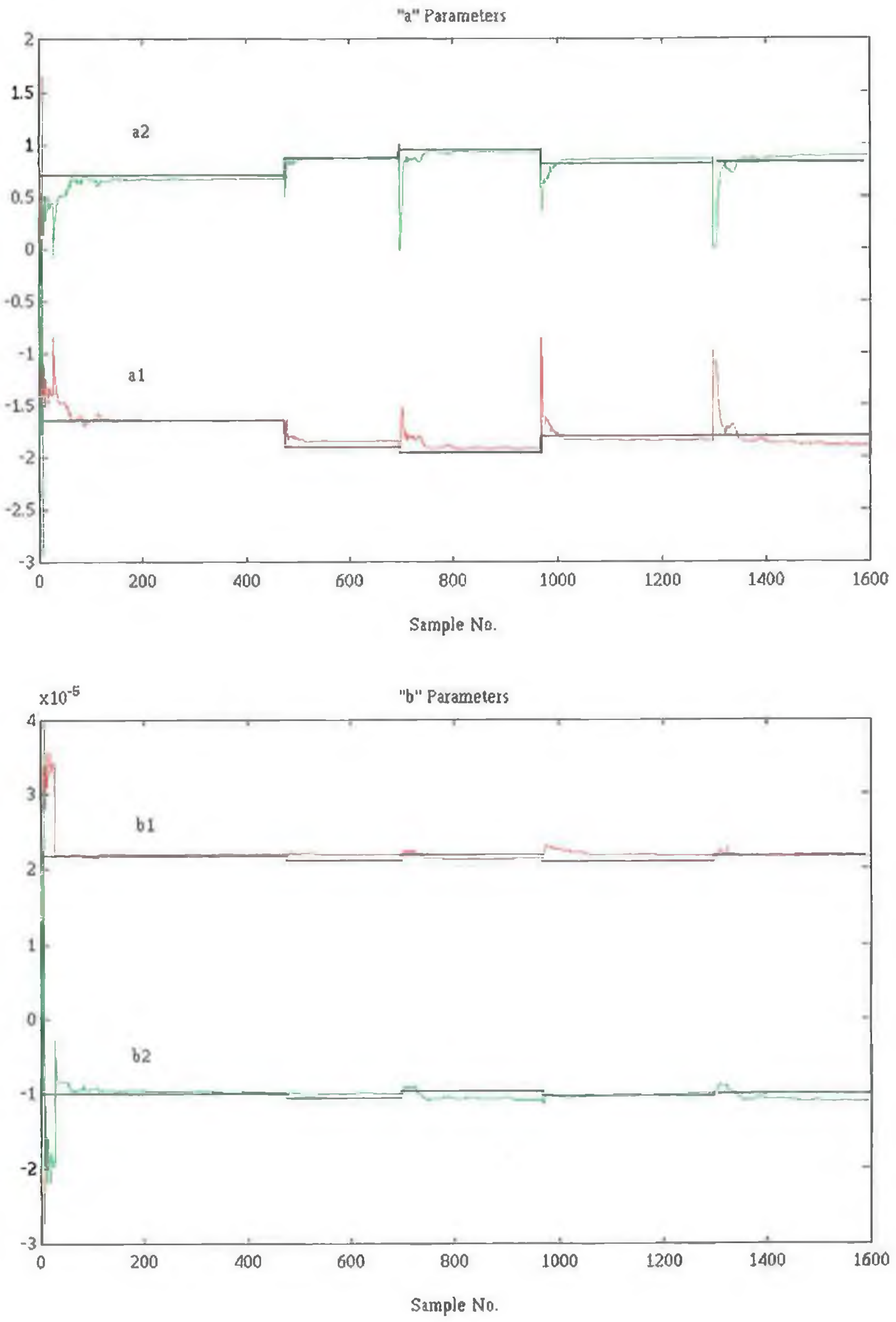
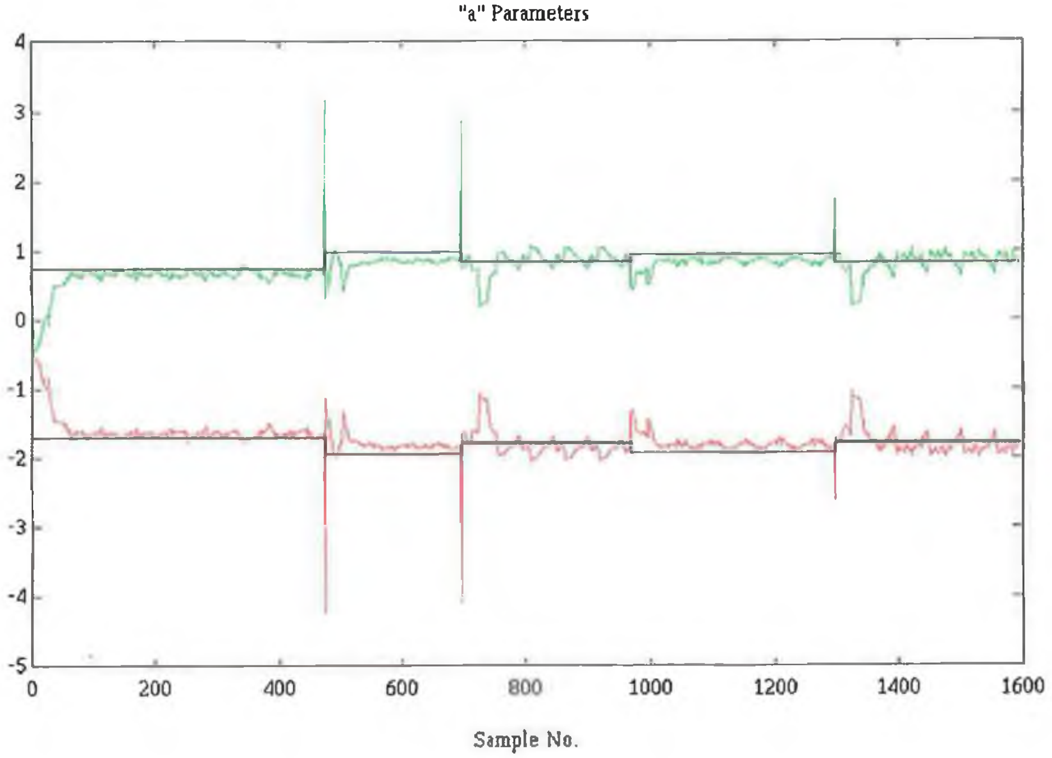
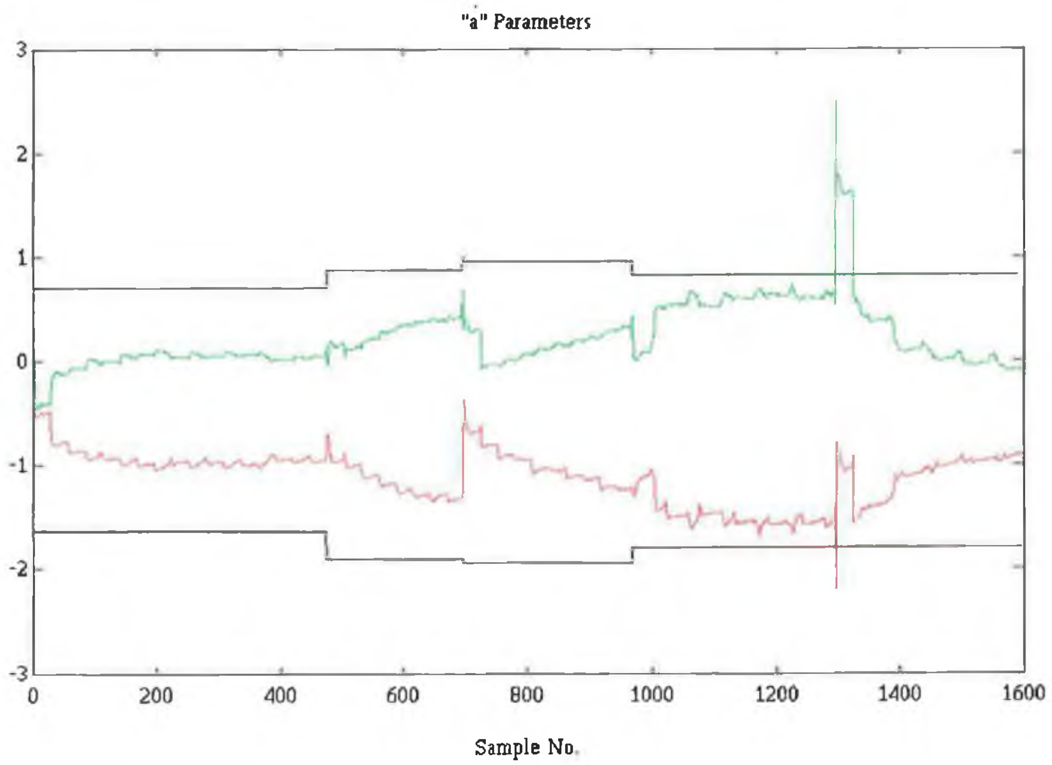


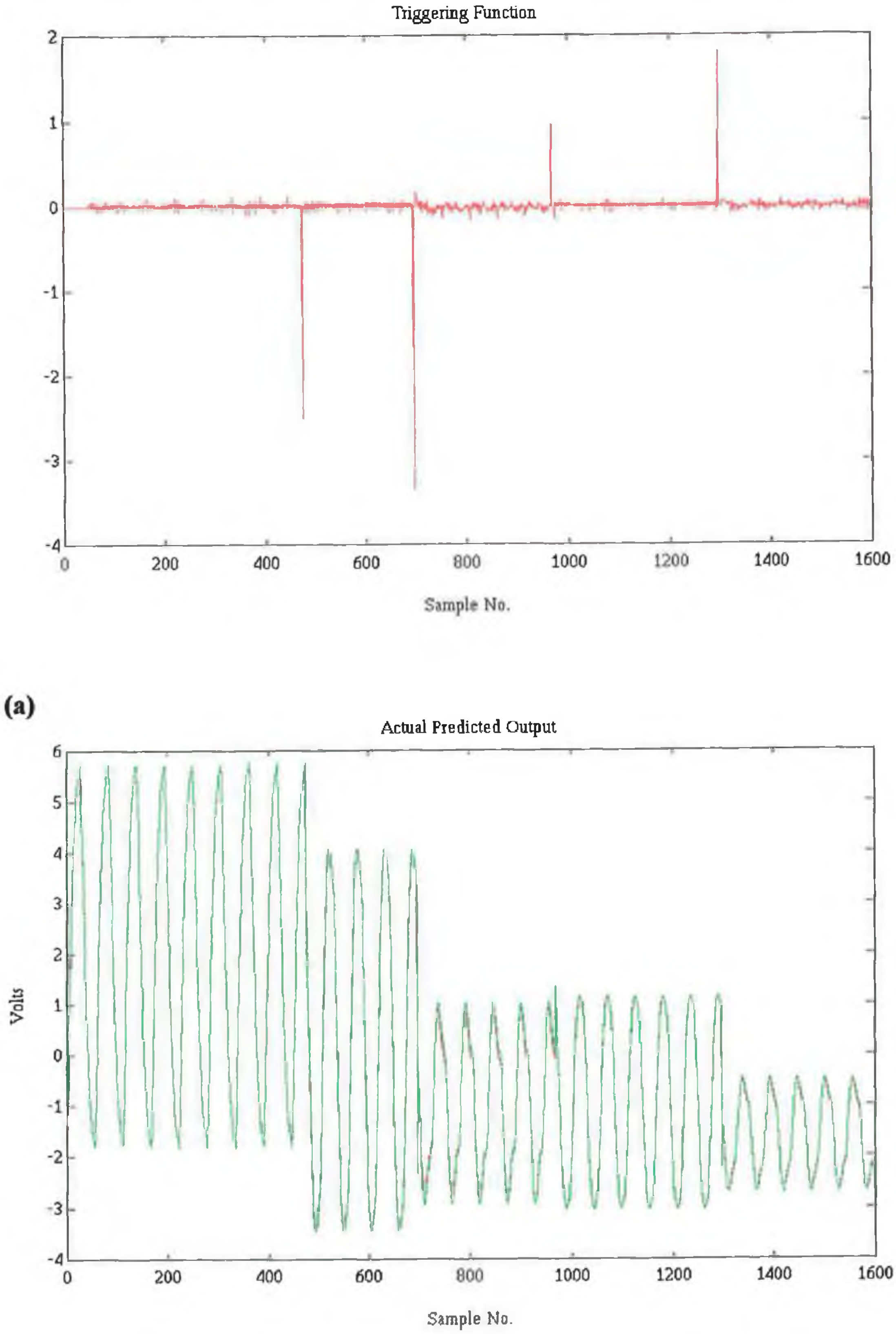
Fig. 5.8 Parameter estimates for the time-varying analogous system.



(a)



(b)
Fig. 5.9. Parameter estimates obtained using (a) forgetting factor, (b) Kalman filter.



(a)

(b)

Fig 5.10 (a) The triggering sequence (b) Actual and predicted analogous system output.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary and Conclusions.

This thesis has outlined the development of a parameter tracking algorithm applicable to motorcycle suspension systems. The development of this algorithm is based on the following phases:

- System Description
- Estimation Techniques
- System Simulation
- System Implementation

The research initially focused on the target application, motorcycle suspension systems. The problems encountered in tuning these systems were outlined and solutions to these problems were proposed. It was concluded that any attempt to solve the so-called "suspension problem" must have as its starting point an accurate model of the system dynamics. Since the suspension dynamics are non-stationary a time-varying parametric model of the suspension system, specifically the M.S.D. model, was presented.

The parameter estimation process was then approached. The most popular parameter tracking algorithms were discussed. It was concluded that these parameter estimation schemes were too passive in their approach and that the resulting parameter estimates suffered an unwarranted degradation because of this. An alternative tracking scheme which incorporated a fault detection algorithm was proposed. It was shown that this estimator yielded estimates with better variance properties than those resulting from the application of the traditional parameter tracking schemes.

To test the performance of this alternative algorithm with respect to suspension systems, the suspension system was simulated using the M.S.D. model developed earlier. Data produced by this simulation was used to verify the tracking capability of the alternative algorithm. This algorithm was altered to remove "spikes" from the parameter estimate sequences and to robustify it against outliers and bad data. This was achieved by triggering the fault detection scheme with a modified mean prediction error function.

Funding constraints meant that the alternative parameter tracking algorithm could not be implemented on an actual motorcycle suspension system. An analogue computer was constructed as an electrical analogy for the suspension system. Data was taken from this computer and its parameters were tracked using this data with the alternative tracking scheme. A comparison of the resulting parameter estimates with estimates produced by the more conventional estimation schemes indicated that the alternative tracking algorithm yielded higher quality estimates.

6.2 Recommendations

6.2.1 Modifications to the current work

6.2.1.1 Modelling

One of the most striking things about the system identification literature is its concentration on single-input single-output systems (s.i.s.o.). Only a small proportion of the recognised texts deal with multi-input multi-output systems. The reason is not mere faintheartedness but the substantial new problems that arise when these systems are considered. Because of this, most multi-input multi-output (m.i.m.o.) systems, if possible, are reduced to s.i.s.o. systems. In this fashion the modelling of the suspension system has been reduced to a single degree of freedom model thus neglecting the tyre dynamics. These dynamics are not entirely negligible and as such should not be ignored if work in this area is to progress.

It is also known that the suspension system may comprise some non-linear components. The issue of system non-linearities has not been dealt with in this thesis. As such it is recommended that the tracking algorithm be extended to cater for non-linear dynamics.

6.2.1.2 Fault detection

It was stated earlier in chapter 3 that fault detection comprises three separate tasks, alarm, isolation and estimation. This work has concerned itself solely with the alarm task in the fault detection protocol. Because of this when a fault occurs (a fault in this context being a change in the system parameters) the alternative algorithm knows only that it has occurred. It doesn't concern itself with the location of the fault (which parameters have changed) or the extent of the fault (by how much the parameters have altered). The inclusion of these secondary tasks in the fault detection scheme would improve the variance and convergence properties of the existing algorithm. However it remains to be seen if the resulting computational burden warrants their inclusion.

6.2.2 Further Research

6.2.2.1 Test rigs and on-site testing.

The work in this thesis has been developed, tested and implemented with the use of digital simulations and an electrical analogue of the actual suspension system. If work in this area is to progress a more knowledgeable understanding of the requirements of the target platform must be attained. With this in mind it is recommended, that in such a scenario, a test rig comprising the complete front suspension, wheel inclusive, be constructed. A recommended design is provided in Appendix C.1. This proposed test-rig would require the following instrumentation/components,

(i) Load Cell : To measure the force imparted to the suspension. This should preferably be mounted in the shock absorber.

(ii) Displacement sensor: To measure the resulting displacement of the shock absorber after a force has been applied.

(iii) Actuator : This would comprise a ballscrew driven by a D.C. servo. Forces would be imparted to the front wheel via a mounting block.

(iv) Surrogate. The purpose of the surrogate is to simulate the remainder of the bike.

Testing on this proposed test-rig should then be followed up with on-site testing.

Bibliography

[1] T. Foale, V Willoughby, *Motorcycle Chassis Design, the Theory and Practice*, London, Osprey, 1984.

[2] T. Yoshimura, N Ananthanarayana, "Stochastic Opimal Control of Vehicle Suspension with Preview on an Irregular Surface, *Int. J. of Systems Science*, Vol. 22, No. 9, pp. 1599-1611, Sept. 1991.

[3] P. G. Wright, D. A. Williams, "The Application of Active Suspension to High Performance Road Vehicles", *Proc. of Int. Congress Transport Election*, Deaton, MI, USA, pp. 333-338, Oct. 1986.

[4] P. G. Wright, "The Influence of Aerodynamics on Design of Formula One Racing Cars", *Int. J. of Vehicle Design*, Vol. 3, No. 4, pp. 383-397, 1982.

[5] J. Dominy, D. N. Bulman, "An Active Suspension for a Formula One Grand Prix Racing Car", ASME *J. of Dynamic Systems, Measurement and Control*, Vol. 107 No. 73, pp. 73-78, March 1985.

[6] L. D. Metz, "Aerodynamic Requirements at the Indianapolis Motor Speedway", *A.I.A.A. J. Guid. Control and Dyn.*, Vol. 8, No. 4, pp. 530-532, July-August 1985.

[7] Lous International, "Vehicle Active Suspension System-Control of Vehicle Dynamics by On-Board Computers", *Des. Eng.*, pp. 36-37, July 1984.

[8] A. G. Thompson, " An Active Suspension with Optimal Linear State Feedback", *Vehicle Syst. Design*, Vol. 5, pp. 187-203, 1976.

[9] D. Hrovat, "Optimal Active Suspension Structures for Quarter-car Vehicle Models", *Automatica*, Vol. 26, No. 5, pp. 845-860, 1990.

[10] M. Sunwoo, K. C. Cheok, "An application of Explicit Self-Tuning Controller to Vehicle Active Suspension Systems", *Proc. of the 29th. Conf. on Decision and Control*, Honolulu, Hawaii, USA, Vol. 4, pp. 2251-2257, Dec. 1990.

[11] M. Sunwoo, K. C. Cheok, N. J. Huang, "Model Reference Adaptive Control for Vehicle Active Suspension Systems", *I.E.E.E. Trans. on Industrial Electronics*, Vol. 38, No.3, pp. 217-222, June 1991.

[12] M. Sunwoo, K. C. Cheok, "Investigation Of Adaptive Control Approaches for Vehicle Active Suspension Systems", *Proc. of the 1991 American Control Conf.* Boston, Mass., USA, Vol. 2, pp. 1542-1547, 1991.

[13] K. C. Cheok, N. J. Huang, "Lyapunov Stability Analysis for Self-Learning Neural Model with Application to Semi-Active Suspension Control System", *Proc. I.E.E.E. Int. Symp. on Intell. Control*, pp. 326-331, 1989.

[14] D. Metz, J. Maddock, "Optimal Ride Height and Pitch Control for Championship Race Cars", *Automatica*, Vol. 22, No. 5, pp. 509-520, 1986.

[15] S. Narayanan, G. V. Raju, "Stochastic Optimal Control of Non-Stationary Response of a Single-Degree-Of-Freedom Vehicle Model", *J. of Sound and Vibration*, Vol. 141, No.3 pp. 449-463, 1990.

[16] A. W. Burton, "Active Vibration Control in Automotive Chassis Systems", *Computing and Control Engineering*, pp. 225-232, Oct. 1993.

[17] M. A. Salman, A. Y. Lee, N. M. Boustany, "Reduced Order Design of Active Suspension Control", *Proc. of the 27th. Conf. on Decision and Control*, Austin, Texas, USA, pp 1038-1043, 1988.

[18] B. Clancy, *A Motor Simulation of a Shock-Absorber*, A Masters Thesis, Dublin City University, 1994.

[19] J. J. D'Azzo, C.H. Houpis, *Feedback Control System Analysis and Synthesis*, McGraw Hill Kogakusha, Ltd., 1960.

[20] P. E. Wellstead, M. B. Zarrop, *Self-Tuning Systems, Control and Signal Processing*, Wiley and Sons, 1991.

[21] I. Asimov, *The Universe*, 3$^{rd.}$ ed. Penguin, London, 1983.

[22] P. Murdin, A. Allen, *Catalogue of the Universe*, Cambridge University Press, London and New York, 1979.

[23] J. P. Norton, *An Introduction to Identification*, Academic Press, 1988.

[24] L. Ljung, *System Identification-theory for the user*, PTR Prentice Hall, 1987.

[25] L. Balmer, *Signals and Systems-an introduction*, Prentice Hall, 1991.

[26] T. Soderstrom, P. Stoica, *System Identification*, Prentice Hall, 1989.

[27] N. Wiener, *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, Wiley, New York, 1949.

[28] G. H. Cohen, G. A. Coon, Theoretical Considerations of Retarded Control, *Trans. A.S.M.E.*, Vol. 75, pp. 827-834, 1953.

[29] J. G. Ziegler, N. B. Nichols, Optimum Settings for Automatic Controllers, *Trans. A.S.M.E.*, Vol. 64, pp. 759-768, 1942.

[30] G. F. Franklin, J. D. Powell, M. L. Workman, *Digital Control of Dynamic Systems*, Addison Wesley, 1990.

[31] J. Pilz, *Bayesian Estimation and Experimental Design in Linear Models*, J. Wiley, 1991.

[32] J. Schoukens, R. Pintelon, *Identification of Linear Systems*, Pergamon Press, 1991.

[33] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3$^{rd.}$ Ed. McGraw-Hill, Inc. 1991.

[34] R. L. Kashyap, Maximum Liklihood Identification of Stochastic Linear Systems, *I.E.E.E. Trans. on Automatic Control*, Vol. 15, No. 1, Feb. 1970.

[35] K. W. Iliff, R. W. Maine, More Than You May Want to Know About Maximum Liklihood Estimation, *A.I.A.A.* paper 84-2070.

[36] K. W. Iliff, R. W. Maine, Important Factors in the Maximum Liklihood Analysis of Flight Test Maneuvers, *NASA* Technical Paper 1459, 1979.

[37] K. W. Iliff, R. W. Maine, *State Space Identification Toolbox Users Manual,* The Mathworks Inc., 1990.

[38] A. P. Sage, C. D. Wakefield, Maximum Liklihood Identification of TIme-Varying and Random System Parameters, *Int. J. Control*, Vol. 16, No. 1, pp. 81-100, 1972.

[39] H. Anton, *Elementary Linear Algebra*, 5th. Ed., Wiley and Sons, 1987.

[40] K. J. Astrom, P. Eykhoff, "System Identification-A Survey", *Automatica*, Vol. 12, pp. 123-162, 1971

[41] L. Ljung, K. Glover, "Frequency Domain Versus Time Domain Methods in System Identification", *Automatica*, Vol. 17, No. 1, pp. 71-86, 1981.

[42] P. E. Wellstead, "Non-Parametric Methods of System Identification", *Automatica,* Vol. 17, No.1, pp. 55-69, 1981.

[43] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.

[44] H. W. Sorenson, "Least Squares Estimation: from Gauss to Kalman, *I.E.E.E. Spectrum*, pp.63-68, July, 1970.

[45] R. K. Mehra, "Optimal Input Signals for Parameter Estimation in Dynamic Systems-Survey and New Results", *I.E.E.E. Trans. on Automatic Control*, Vol. AC-19, No. 6, December, 1974.

[46] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Trans. A.S.M.E Ser. D. J. of Basic Engineering*, 82, pp. 35-45.

[47] A. S. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems", *Automatica*, Vol. 12, pp. 601-611, 1976.

[48] T. Hagglund, "Adaptive Control of Systems Subject to Large Parameter Changes", *I.F.A.C. $9^{th.}$ World Conf.*, Hungary, 1984.

[49] R. K. Mehra, J. Peschon, "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems", *Automatica*, Vol. 7, pp. 637-640, 1970.

[50] J. J. Deyst, J. C. Deckert, "RCS Jet Failure Detection for the Space Schuttle", *Automatica*, Vol. 12, pp. 601-611, 1976.

[51] R. N. Clark, D. C. Fosth, V. M. Walton, "Detecting Instrument Malfunctions in Control Systems" *I.E.E.E. Trans. Aerospace Elect. Syst.*, Vol. AES-11, No.4, July, 1975.

[52] P. Sanyal, "Bayes' Descision Rule for Rapid Detection and Adaptive Estimation Scheme with Space Applications", *I.E.E.E. Trans. Aut. Control*, Vol. 19, pp. 228-231, Jan. 1974.

[53] K. Ogato, *Modern Control Engineering*, Prentice Hall International, Inc., Englewood Cliffs, N.J., U.S.A.

[54] " Simulink A Program for Simulating Dynamic Systems", *The Mathworks, Inc.*, 1990.

[55] I. J. Nagrath, M. Gopal, *Systems Modelling and Analysis*, Tata McGraw Hill Ltd. 1982.

[56] *User Manual for the DT2821 Series*, Data Translation, Inc. 1990.

[57] *Hypersignal Users Manual*, Hyperception Inc. February 1991.

Appendix A.1.

Derivation of the Least Squares Estimate [23-24], [26]

Given the system,

$$y(t) = \varphi^T(t)\theta + e(t) \qquad \text{Equ. A.1.1}$$

the least squares criterion minimises the mean squared error between the system output and the model output. The mean squared criterion is given as,

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[y(t) - \varphi^T(t)\theta]^2 \qquad \text{Equ. A.1.2}$$

Collect all measurements to give,

$$V(\theta) = \frac{1}{2}[Y - \phi\theta]^T[Y - \phi\theta]$$

$$= \frac{1}{2}[\theta^T\phi^T\phi\theta - \theta^T\phi^T Y - Y^T\phi\theta + Y^T Y] \qquad \text{Equ. A.1.3}$$

Hence,

$$V(\theta) = \frac{1}{2}[\theta - (\phi^T\phi)^{-1}\phi^T Y]^T(\phi^T\phi)[\theta - (\phi^T\phi)^{-1}\phi^T Y]$$

$$+\frac{1}{2}[Y^T Y - Y^T\phi(\phi^T\phi)^{-1}\phi^T Y] \qquad \text{Equ. A.1.4}$$

The second term is independent of θ. Since $\phi^T\phi$ is assumed positive definite, the first term is always greater than or equal to zero. Thus $V(\theta)$ can be minimised by setting the first term to zero.

This yields the least squares estimate as

$$\hat{\theta} = (\phi^T\phi)^{-1}\phi^T Y \qquad \text{Equ. A.1.5}$$

or

$$\hat{\theta}_N^{LS} = \arg\ \min V_N(\theta, Z^N) = \left[\frac{1}{N}\sum_{t-1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t) \qquad \text{Equ. A.1.6}$$

The minimum value of the cost function is thus given by

$$\min_{\theta} V(\theta) = V(\hat{\theta}) = \frac{1}{2}[Y^T Y - Y^T\phi(\phi^T\phi)^{-1}\phi^T Y] \qquad \text{Equ. A.1.7}$$

Appendix A.2.

Derivation of the Least Squares Recursion [23-24], [26].

The batch least squares estimate is given as,

$$\hat{\theta}(t) = (\phi^T(t)\phi(t))^{-1}\phi^T(t)Y(t) \qquad \text{Equ. A.2.1}$$

This is based on t measurments. Now assume one more measurement is made. This yields,

$$\hat{\theta}(t+1) = (\phi^T(t+1)\phi(t+1))^{-1}\phi^T(t+1)Y(t+1) \qquad \text{Equ. A.2.2}$$

Now

$$\phi^T(t+1)\phi(t+1) = [\phi^T(t)\varphi(t+1)][\phi(t)\varphi^T(t+1)] \qquad \text{Equ. A.2.3}$$

Thus given $\phi(t+1)$ it is easy to update the old matrix of correlations $\phi^T(t)\phi(t)$ to obtain the new matrix $\phi^T(t+1)\phi(t+1)$. However it is necessary to update the inverse of $\phi^T(t)\phi(t)$ without requiring a matrix inversion at each time step. It is also necessary to update the term $\phi^T(t+1)Y(t+1)$. We have,

$$\phi^T(t+1)Y(t+1) = [\phi^T(t)\ \varphi(t+1)][Y(t)\ y(t+1)]^T$$

$$= \phi^T(t)Y(t) + \varphi(t+1)y(t+1) \qquad \text{Equ. A.2.4}$$

Now introduce the following shorthand notation,

$$P(t) = [\phi^T(t)\phi(t)]^{-1} \qquad \text{Equ. A.2.5}$$

$$B(t) = \phi^T(t)Y(t) \qquad \text{Equ. A.2.6}$$

Thus we have

$$\hat{\theta}(t+1) = P(t+1)B(t+1) \qquad \text{Equ. A.2.7}$$

$$\hat{\theta}(t) = P(t)B(t) \qquad \text{Equ. A.2.8}$$

Also

$$P^{-1}(t+1) = P^{-1}(t) + \varphi(t+1)\varphi^{T}(t+1) \qquad \text{Equ. A.2.9}$$

$$B(t+1) = B(t) + \varphi(t+1)y(t+1) \qquad \text{Equ. A.2.10}$$

The B update equation is a direct update involving no matrix inversions. To accomplish this for the P update equation requires the matrix inversion lemma[24]. This states that,

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad \text{Equ. A.2.11}$$

Assigning A=$P^{-1}(t)$, C=1, B=$\varphi(t+1)$ and D=$\varphi^{T}(t+1)$ gives

$$P(t+1) = P(t)[I_m - \varphi(t+1)(1+\varphi^{T}(t+1)P(t)\varphi(t+1))^{-1}\varphi^{T}(t+1)P(t)]$$

Equ. A.2.12

Thus we have a recursion for the P matrix which doesn't involve a matrix inversion. Now given that

$$\varepsilon(t+1) = y(t+1) - \varphi^{T}(t+1)\hat{\theta}(t) \qquad \text{Equ. A.2.13}$$

we can substitute this into equation A.2.10.

$$B(t+1) = B(t) + \varphi(t+1)\varphi^{T}(t+1)\hat{\theta}(t) + \varphi(t+1)\varepsilon(t+1) \qquad \text{Equ. A.2.14}$$

This gives

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\varphi(t+1)\varepsilon(t+1) \qquad \text{Equ. A.2.15}$$

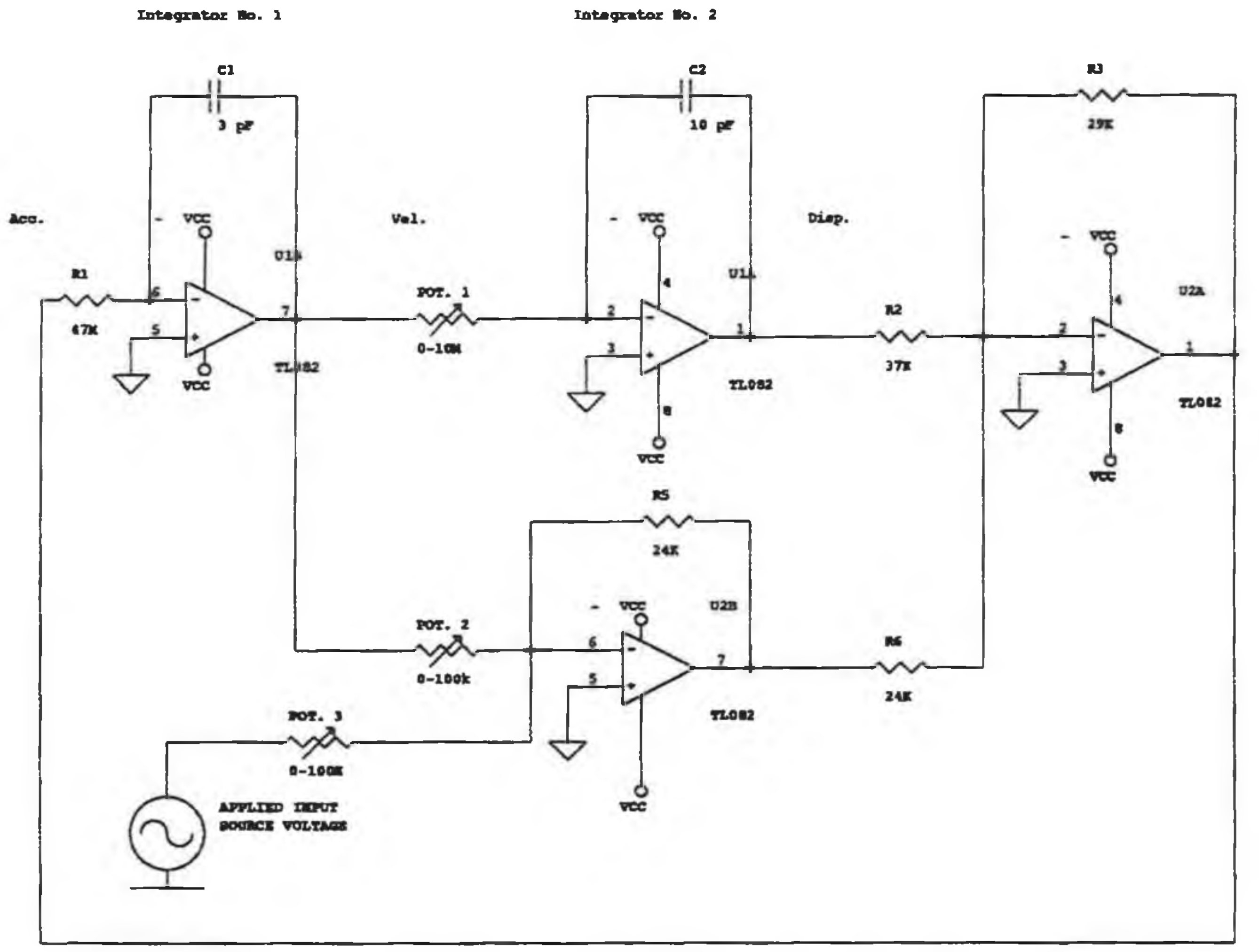
Summarising, the recursive least squares estimator is composed of the following equations,

$$P(t+1) = P(t)\left[I_m - \frac{\varphi(t+1)\varphi^T(t+1)P(t)}{1+\varphi^T(t+1)P(t)\varphi(t+1)} \right]$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\varphi(t+1)[y(t+1) - \varphi^T\hat{\theta}(t)].$$

Appendix B.1.

Circuit Diagram for the Analogue Computer

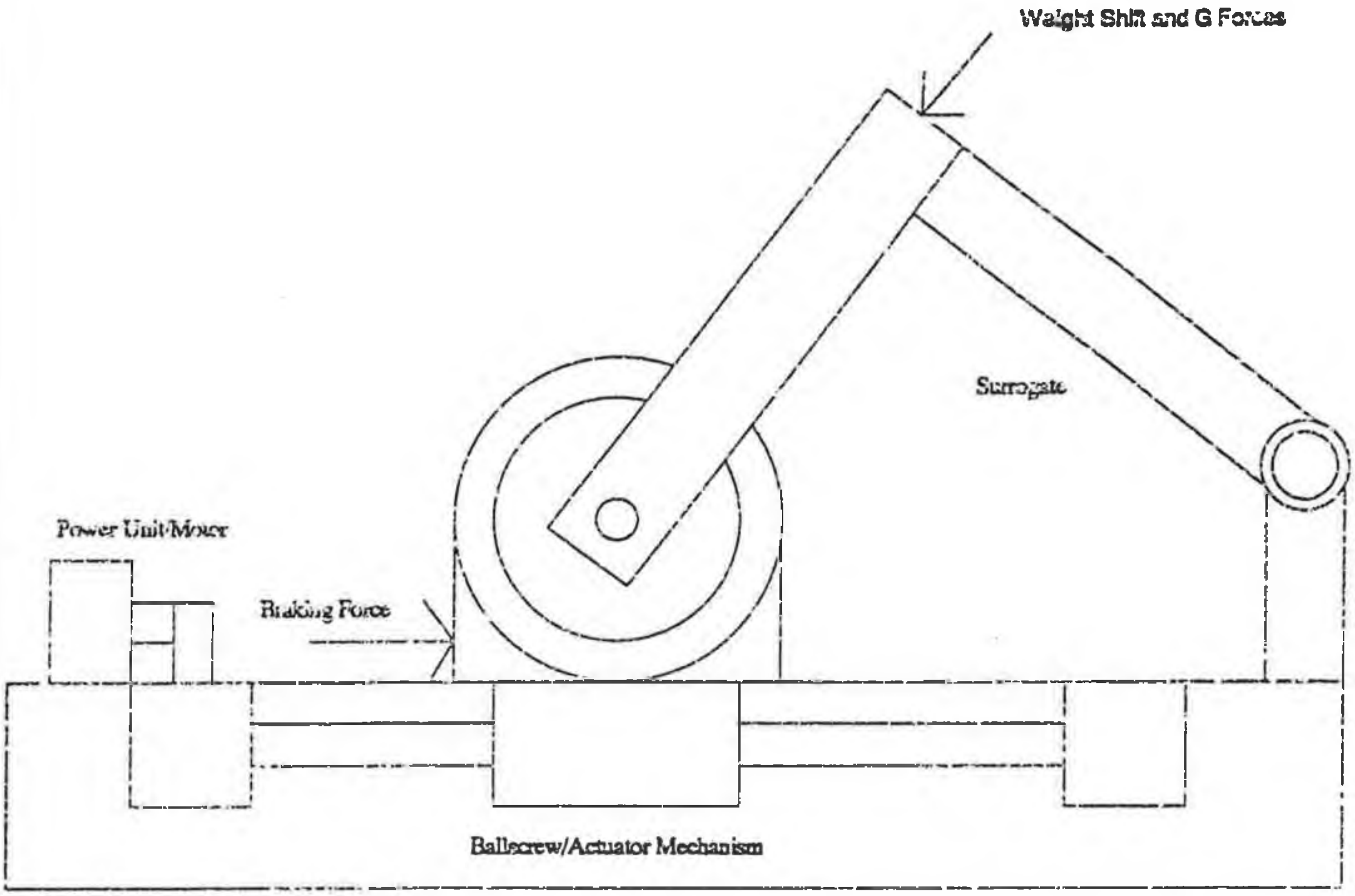


Analog Computer Equivalent Of Shock Absorber.

Title Analogue Computer			
Size A	Document Number 1		REV
Date: February 9, 1995		Sheet 1 of 1	

Appendix C.1.

Proposed Test Rig Design.


Weight Shift and G Forces
Surrogate
Power Unit/Motor
Braking Force
Ballscrew/Actuator Mechanism