

Cranfield Institute of Technology

Department of Electronic System Design

M.Phil Thesis

Academic year 1989-90

Vehicle active suspension system design using
strategies based on state estimation by Kalman
filters

A. A. Shakir

Supervisor: J. Borrie

June 1990

This thesis is submitted in partial fulfilment of
the requirements for the degree of Master of
Philosophy

ProQuest Number: 10832224

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10832224

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by Cranfield University.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

In the name of God
most gracious most merciful

TO MY PARENTS

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my supervisor, Mr. J. Borrie, for his outstanding supervision and guidance, and for his generosity in arranging sufficient time for consultation and supervision throughout this work.

I acknowledge the support of the Iraqi Government who granted this study.

Finally, I would like to thank my wife for her patience, help and encouragement.

A A SHAKIR
June 1990
Cranfield

ABSTRACT

The suspension system of a ground vehicle moving on or off road can be improved by so called "active control" using control strategies which require feedback of all or some of the system states. A stochastic state estimator, or "Kalman filter", can be used to estimate these states.

Various Kalman filters suited to this problem are developed, and optimal control strategies suited to these Kalman filters are specified and assessed. It turns out that there are basic problems of unobservability and uncontrollability inherent in the behaviour of such suspension systems, and original solutions to these problems are suggested and explored.

A laboratory equipment is designed and constructed to reveal the behaviour of hydraulic actuators in this context, and this is used to ensure that the strategies suggested are realistic. An Intel 8086 microprocessor based circuit in which a practical Kalman filter can be realized is designed. A considerable number of computer simulations are also used to test and assess the estimation and control strategies. All these represent original work by the author.

CONTENTS

	Page
1.0 INTRODUCTION	
1.1 Introduction	1-1
1.2 Objectives of This Project	1-3
1.3 Thesis Outline	1-5
2.0 THE NATURE AND REQUIREMENTS OF VEHICLE SUSPENSION SYSTEMS	
2.1 Introduction	2-1
2.2 Human Response to Vibration	2-3
2.3 Vehicle Suspension System Models	2-4
2.4 Types of Vehicle Suspension System	2-7
3.0 ROAD SURFACES	
3.1 Introduction	3-1
3.2 The Road As a Homogeneous and Isotropic Random Process	3-1
3.2.1 Homogeneity	3-2
3.2.2 Isotropy	3-2
3.3 The Spectral Density of a Road	3-6
3.4 Computer Simulation of a Random Road Disturbances	3-9
3.5 Pseudo Random Signal Generation	3-12
3.5.1 Evenly Distributed Random Numbers	3-12
3.5.2 Random Numbers with Gaussian Distribution	3-13
3.6 Generating Gaussian White Noise on the Computer	3-13
3.7 Generating Colored Noise on the Computer	3-16
3.8 Simulating Various Noises	3-21
4.0 STATE ESTIMATION AND OPTIMAL FILTERING WITH LINEAR SYSTEM MODELS	
4.1 Introduction	4-1
4.2 The Asymptotic State Estimator	4-2
4.2.1 Choice of Estimator Gain Matrix M	4-4
4.3 Linear Stochastic Models	4-5
4.4 Kalman Filter for Discrete Time Systems	4-7
4.4.1 The Basic Kalman Filter	4-7
4.4.2 The Split Form of Kalman Filter	4-11

4.5	Covariance Factorisation Filter	4-11
4.5.1	UD Method of Kalman Filter	4-12
4.6	Factors Influencing the Choice of a Suitable microprocessor	4-16
5.0	8086-MICROPROCESSOR HARDWARE AND SOFTWARE	
5.1	Introduction	5-1
5.2	The 8086 CPU Board (board 1)	5-5
5.2.1	Clock Generator	5-5
5.2.2	The Microprocessor (CPU)	5-5
5.2.3	System Buses	5-6
5.2.4	RAMs & EPROMs Memory Chips.	5-6
5.2.5	I/O Decode	5-7
5.2.6	Interrupt Controller and Interval Timer	5-7
5.3	The Analog Board (board 2)	5-9
5.3.1	The A/D and D/A converters	5-9
5.4	The Keyboard and Display Board (board 3)	5-11
5.5	The Software Operating System	5-11
5.5.1	Data Input	5-16
5.5.2	Decimal to Binary Data Conversion	5-18
5.5.3	Binary to Decimal Data Conversion	5-20
5.5.4	Programming the Sample Time	5-22
6.0	A 2 DOF QUARTER CAR MODEL	
6.1	Introduction	6-1
6.2	The Quarter Car Model	6-1
6.3	The Open Loop System Response	6-4
6.4	Closed Loop System Response	6-5
6.5	Digital Simulation	6-12
6.6	More Developed Quarter Car Models	6-26
6.7	Conclusion	6-29
7.0	ACTIVE SUSPENSION VEHICLE EXPERIMENTAL MODEL	
7.1	Introduction	
7.2	System States Equations	7-7
7.3	System Optimal Control	7-17
7.4	The State Estimation Problem	7-35

7.4.1	Approximate Kalman Filter	7-35
7.4.2	Augmented Outputs	7-47
7.4.3	Limited State Estimation	7-56
7.4.4	Reduced Order Kalman Filter	7-62
7.5	Optimal Control Strategies	7-63
7.5.1	Control Strategy Based on Approximate KF	7-63
7.5.2	Control Strategy Based on Augmented Outputs	7-66
7.5.3	Control Strategy Based on Limited State Estimation	7-68
7.6	Conclusion	7-70

SUMMARY, CONCLUSION AND SUGGESTION FOR FURTHER WORK

8.1	Introduction	8-1
8.2	Summary of Work to Date	8-1
8.3	Conclusions	8-2
8.4	Further Work	8-3

APPENDICES:

A	CONTINUOUS AND DISCRETE TIME DYNAMIC SYSTEMS REPRESENTATION
B	PARTS LIST FOR MICROPROCESSOR HARDWARE
C	THE 8086 μ P AND THE PERIPHERALS AS USED IN THIS PROJECT
D	THE STATE ESTIMATOR, SUB-OPTIMAL KALMAN FILTER MICROPROCESSOR SOFTWARE
E	STATE ESTIMATOR EXPERIMENTAL RESULTS
F	LINEAR STOCHASTIC OPTIMAL CONTROL THEORY USING PARTIAL STATE FEEDBACK
G	NON-LINEARITIES OF A HYDRAULIC SYSTEM
H	SPECIFICATIONS OF THE HYDRAULIC EQUIPMENT

LIST OF NOTATIONS

S	Continuous time system.
$\sum$	Discrete time system.
$\hat{\sum}$	Estimated discrete time system.
$u(t)$	Continuous time input vector.
$u(n)$	Discrete time input vector.
$y(t)$	Continuous time output (measurement) vector.
$y(n)$	Discrete time output (measurement) vector.
$x(t)$	Continuous time states vector.
$\tilde{x}(t)$	Continuous time error states vector.
$\hat{x}(t)$	Estimated continuous time state vector.
$x(n)$	Discrete time state vector.
$\hat{x}(n)$	Estimated discrete time states vector.
$\tilde{x}(n)$	Discrete time error states vector.
$x(0)$	Initial state vector.
$\hat{x}^-(n)$	Time update estimated state vector.
$\hat{x}^+(n)$	Measurement update estimated state vector.
T	Sampling time.
$A(t)$	State matrix (continuous time).
$B(t)$	Input matrix (continuous time).
$C(t)$	output matrix (continuous time).
$D(t)$	Direct matrix (continuous time) zero in this case.
$\phi(n)$	State matrix (discrete time).
$\psi(n)$	Input matrix (discrete time).
$C(n)$	output matrix (discrete time).
$D(n)$	Direct matrix (discrete time) zero in this case.
K_F	Kalman Filter optimal gain matrix.
M	State Estimator gain matrix.
$w(t)$	Process noise vector.
$v(t)$	Measurement noise vector.

$R(n)$	Measurement noise Covariance matrix (continuous time).
$Q(n)$	Process noise Covariance matrix (continuous time).
$R_d(n)$	Measurement noise Covariance matrix (discrete time).
$Q_d(n)$	Process noise Covariance matrix (discrete time).
$G(t)$	Noise input matrix (continuous time).
$G(n)$	Noise input matrix (discrete time).
$G(s)$	Continuous time transfer function.
$S(\eta)$	Power spectral density.
$R(\tau)$	Auto-correlation.
$P(n)$	Error covariance matrix.
$U(n), D(n)$	Covariance factorisations matrices (Kalman filter).

ABBREVIATIONS AND GREEK LETTERS

CPU	Central Processing Unit.
PCT	Programmable Counter Timer.
PIC	Programmable Interrupt Controller.
CLK	Clock.
RAM	Random Access Memory.
EPROM	Erasable Programmable Read Only Memory.
A.S.E.	Asymptotic State Estimator.
K.F.	Kalman Filter.
UD.K.F.	UD method of Kalman Filter.
C.K.F.	Conventional Kalman Filter.
OLHP	Open Left Half Plane.
PDF	Probability Density Function.
s	Second.
μ	Micro-.
σ	White noise Standard Deviation.
ε, E	Mean value.
β	Brownian motion noise vector.
ω	Natural frequency (rad/sec).
$E(.)$	Expectation.
$VAR = \sigma^2$	Variance.
$S.D. = \sigma$	Standard of deviation.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

Ground vehicles, which are supported by the ground, as contrasted with aircraft and marine craft, may be broadly classified as guided and nonguided. Guided ground vehicles are constrained to move along a fixed path (guide way), such as railway and levitated vehicles. Nonguided ground vehicles can move, by choice, in various directions on the ground, such as road and off-road vehicles.

A vehicle which runs on the ground employs a suspension system to separate the passenger or cargo container from auxiliary bodies, some of which are in contact with the ground. The traditional suspension system serves as a low pass filter which is designed to compromise between ride comfort and safe handling, specified as follows.

- 1- The suspension must reduce the transmission of the vibrations from the road to the body/cargo and thus ensure *"ride comfort"*.
- 2- The suspension must prevent the wheels from leaving the ground during cornering, braking or accelerating to ensure the transfer of tractive forces to the ground and thus ensure *"safe handling"*.

The first requirement implies a "soft" suspension with a low cut off frequency which isolates the vehicle body from ground induced vibrations. The latter implies a "stiff" suspension to keep the wheels/tyres in contact with the ground. Increased stiffness implies an increase in transmission bandwidth and a larger range of frequencies are therefore transmitted to the vehicle body,

which is in conflict with the first requirement. A compromise must therefore be made.

This leads to the idea of "active" suspension with characteristics which can realize this compromise efficiently, or even be altered to suit the terrain being negotiated. Hydraulic actuators and microprocessors are evidently suitable for such active suspension systems, and this project seeks to apply these to the problem.

Although microprocessors are cheap nowadays, active suspensions systems still have limited application, and vehicles with active suspensions are complex and expensive. Intensive research is still required before this kind of suspension becomes widely used.

To date, the most obvious areas in which such active suspensions seem feasible are:-

** Commercial Vehicles:*

- Active systems can increase safety by achieving better road holding (or ground holding for agricultural vehicles). Comfortable ride characteristics would also allow longer periods of operation due to reduction in operator fatigue.

** Military Vehicles:*

- In this context cross country speed is limited by the ability of the crew to withstand vibration, the modern fighting vehicle being capable of going faster over rough terrain than a human can stand. The use of active suspension might allow an increase in speed. The strategic value of such an improvement would probably be considerable.

** Passenger Vehicle:*

- Active systems can increase both ride comfort and safety, by ensuring that the wheels remain on the ground during maneuvers such as braking and cornering. At the smaller end of the scale (ie cars) the cost is still too high to be widely adopted, but larger vehicles (buses, etc) might benefit considerably.

* *Racing Cars:*

- Racing cars which enhance performance through the use of aerodynamic effects can be height regulated using active suspension systems, producing stable, controllable vehicles capable of braking and cornering with greater efficiency. The economics of this are thought to be attractive and several such systems are believed currently to be under development.

1.2 OBJECTIVES OF THIS PROJECT:

Any effective control strategy must use state feedback and this requires measurement of the states. In active suspension systems, these are not generally accessible; in particular the absolute position of the cargo container, or "sprung mass", above mean ground level can not easily be measured. This must be somehow be found by processing the output of an accelerometer attached to the sprung mass, and possibly the outputs of other transducers, most notably a linear potentiometer monitoring the relative displacement of sprung mass and axle, or "unsprung mass".

This difficulty can be solved if a proper state estimator is designed. A Kalman filter seems suitable in this situation as the available outputs are corrupted by measurement noise and the system is disturbed by process noise. Naturally, different ground and speed conditions will affect the Kalman filter design, and, with microprocessor realization, the optimal Kalman filter gains can be updated according to ground type and vehicle speed.

There is a considerable body of literature on this subject and many control strategies have been suggested ranging from simple classical systems to adaptive optimal control strategies based on stochastic analysis. A useful review is provided by ref. 1.

Relatively few of the published papers present practical results, the majority being satisfied with simulations, and, in simulation

at least, the modern control strategies appear to yield better results than the classical.

A seminal paper on optimal control applied to this subject is by Thompson [2], who presents good simulated results, but whose strategy depends on all the system states being accessible. This is not so, since the height of the sprung mass above the mean ground level is difficult, if not impossible, to measure. Thompson specifies carefully chosen cost functions to avoid this problem, but still requires a transducer to measure the sprung mass height above the actual (not mean) ground level.

Further work reported [3,4], conforms that optimal control strategies, some using partial state feedback yield good simulated results, but all require the system states to be accessible. Some work has been done on applying Kalman filters to active suspensions [5,6], but this seems to be for complex systems involving flexible vehicles, compensation for body height transducer delays, etc, and this work is associated with elaborate control schemes.

Clearly, if a relatively simple Kalman filter could be designed to construct accurate state estimates using simple transducer signals as data, optimal control strategies would become practically attractive. The most obvious transducers are an accelerometer monitoring sprung mass acceleration, and a potentiometer monitoring suspension deflection.

The objectives of this project are to design simple Kalman filters suited to this problem and to design optimal control strategies suited to any peculiarities of these Kalman filters.

It is proposed that the strategies be developed using a laboratory equipment employing hydraulic actuators, so that any results obtained should have practical meaning. Computer simulation is also, of course, to be used.

1.3 THESIS OUTLINE:

This thesis can be outlined as follows:

Chapter 2: Recognized comfort criteria and the general features of suspension systems are briefly reviewed.

Chapter 3: Road surfaces are studied and it is concluded that a satisfactory model is a stationary Gaussian noise stochastic process, for constant vehicle speed. Typical roads are represented by a special form of PSD, that means the road disturbance is not white noise but "colored" noise. To simulate typical roads for this work, specialized digital filters are designed of which cases are realized for good and poor roads.

Chapter 4: The algorithms of state estimators and various types of Kalman filter are reviewed.

Chapter 5: The electronic hardware required for on-line Kalman filter realization based on an 8086 microprocessor and suitable for 4 wheel vehicle is presented. The necessary software in 8086 assembly language is also described.

Chapter 6: A typical simple model representing a 2-DOF quarter car is considered. The objective of this chapter is to illustrate how an optimal control strategy is realized, and to compare simple active and passive suspension systems assuming all the system states are available. More developed models (based on practical experience described in Chapter 7) are also presented.

Chapter 7: An experimental equipment representing a 2-DOF quarter car is developed and modeled, and a basic optimal control strategy is considered. A basic problem of unobservability is exposed, and structures of different Kalman filters are proposed which overcome this. A basic problem of uncontrollability is also exposed and optimal control strategies which deal with this and suit the Kalman filters are proposed. These ideas are tried out in simulation and on the laboratory system.

Chapter 8: Conclusions are presented and suggestions for further work are discussed.

This thesis also includes the following appendices:

Appendix A: The algorithms used for system conversion from continuous to discrete time form.

Appendix B: Parts list for the 8086 microprocessor hardware rig.

Appendix C: The operation of the main components used by the 8086 microprocessor rig.

Appendix D: The 8086 microprocessor software structure.

Appendix E: Some results of testing the 8086 microprocessor state estimator algorithm on an analog computer model are shown.

Appendix F: The partial state feedback method for one optimal control strategy used in Chapter 7 is described.

Appendix G: The main non-linearities of the hydraulic system used in this work are considered.

Appendix H: The specifications of the hydraulic equipment discussed in Chapter 7.

From the outline above, it can be seen that the work covers an extensive field. While the practicality main objectives has been demonstrated as described in Chapters 7 and 8, it has not been possible in the time available to tie all the strands together. In particular, the microprocessor hardware has not been applied to the experimental equipment, nor have the strategies been thoroughly tested using the Kalman filters. Further work is thus indicated, but it is claimed that useful basic ideas have been established.

REFERENCES:

- [1] R. S. Sharp and D. A. Crolla, "Road Vehicle Suspension System Design - a review", Vehicle System Dynamics, 1987.
- [2] A. G. Thompson, " An Active Suspension with Optimal Linear State Feedback", Vehicle System Dynamics, 1976.
- [3] D. A. Wilson, R. S. Sharp, and S. A. Hassan, "The Application of Linear Optimal Control Theory to the Design of Active Automotive Suspension", Vehicle System Dynamics, 1986.
- [4] A. Hac, " Suspension Optimization of a 2-DOF Vehicle Model Using a Stochastic Optimal Control Technique", Journal of Sound and Vibration 1985.
- [5] A. Hac, " Stochastic Optimal Control of Vehicle with Elastic Body and Active Suspension", Vehicle System Dynamics, 1986.
- [6] F. Fruhauf, R. Kasper, J. Luckel, "Design of an Active Suspension System for a Passenger Vehicle Model Using Input Processes with Time Delays, Proc. 9th IAVSD Symposium on the Dynamics of Vehicle on Roads and on Railway Tracks, 1985.
- [7] T. Yoshimura, "An Active Vertical Suspension for Track/Vehicle Systems", Journal of Sound and Vibration, 1986.

CHAPTER 2

THE NATURE AND REQUIREMENTS OF VEHICLE SUSPENSION SYSTEMS

2.1 INTRODUCTION:

The behaviour of a ground vehicle represents the results of interaction between the driver, the vehicle and the environment, as illustrated in Fig. 2.1, so in the end we can determine the suspension system demands of "safety" and the "comfort" of the vehicle for different conditions of driving. Some understanding of the behaviour of the human driver, the characteristics of the vehicle, and the physical and geometric properties of the ground are, therefore, necessary to the design and evaluation of ground vehicle systems.

Many workers in this field are trying to find the optimal parameters of the suspension system to achieve "optimal" riding and handling properties [1,2,3,4]. It may be said that there is no universally accepted set of criteria which can be taken to measure these.

One of the major factors in the study of ground vehicles is certainly riding "*comfort*". This is concerned with the sensation or *feel* of the passenger in the vehicle. In this work we concentrate on this feature, at the expense of road holding which is not considered in any depth.

Ride comfort problems mainly arise from vibration of the vehicle, which are induced by surface irregularities, aerodynamic forces and vibrations of the engine and transmission. Usually surface irregularity is the major source of vibration.

In the next paragraph we consider human response to vibration.

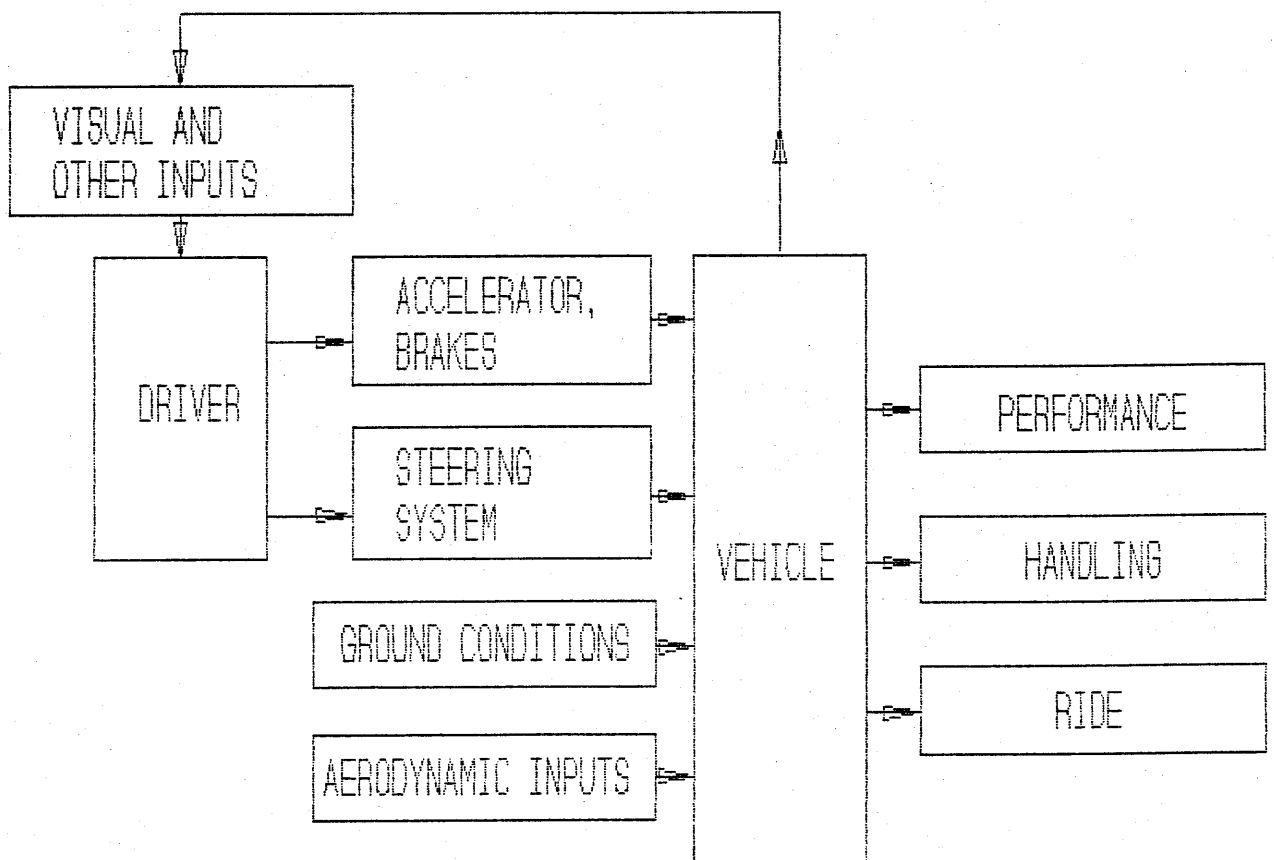


Fig 2.1 The Driver-Vehicle-Ground System

2.2 HUMAN RESPONSE TO VIBRATION:

In general, passenger comfort boundaries are difficult to determine. Various methods for assessing human response to vibration have been developed [1]. They include subjective ride assessments, shake table tests, ride simulator experiments and ride measurements in vehicle [1].

In general these methods attempt to correlate the response of the test subjects in qualitative terms, such as "uncomfortable" and "extremely uncomfortable", with vibrational variables such as displacement, velocity, acceleration and jerk over the frequency range of interest.

At present there is little consensus on how to evaluate human response to whole body vibration. An International Standard (ISO) [12] has been published which uses rms acceleration. Various levels are proposed with time durations at each level. This guide is recommended for the evaluation of vibrational environments in transport vehicles as well as in industry, and defines three distinct limits for whole-body vibration in the frequency range 1-80 Hz.

Another method which is widely accepted as more promising than the ISO method has been proposed by Lee & Pradko in 1966 [2]. This is based on the assumption that comfort is a function of power absorbed by the human body. The measure of comfort is defined as the sum of the power absorbed by the body from all the components of imposed vibration. This method has been confirmed as useful by extensive testing.

The total absorbed power in frequency domain is:

$$P_{ab} = \sum_{i=0}^N K_i \cdot A_i^2(\text{rms})$$

where

A_i^2 rms = mean square acceleration at frequency 'i'.

K_i = conversion constant, which is frequency and body orientation dependent.

The power absorbed depends on the physical attributes of the person. A muscular, trim person generally absorbs less power than an obese person of the same weight when subjected to same vibrational input.

2.3 VEHICLE SUSPENSION SYSTEM MODELS:

To study the behaviour of vehicle suspension system, various models have been developed. For a passenger car with independent front suspension, a seven-degree-of-freedom model as indicated in Fig.(2.2) may be used. In this model, the pitch, bounce, and roll of the vehicle body, and each wheel bounce are taken into consideration.

For a military tank as shown in Fig.(2.3), a fifteen degree of freedom model may be used, which includes the pitch, bounce, and roll of the vehicle body and the bounce of each wheel, these being typically twelve in number.

To construct such a model, the equations of motion for each mass are formulated, or Lagrangian mechanics are used, and a set of differential equations is formulated. Digital computer simulations are usually employed to determine the responses, when the excitation of the system is known, by solving these equations. The models become very complex as the degrees of freedom of the system increase.

The mass of the vehicle body is usually referred to as the sprung mass, whereas the mass of the axle with its associated components is referred to as the unsprung mass. To obtain a qualitative insight into the functions of the suspension, particularly the effects on the sprung and unsprung masses of vehicle vibration, a linear model with two degrees of freedom as indicated in Fig.(2.4) is usually used. On the other hand to reach a better understanding of the pitch and bounce vibration of the vehicle body, a linear two-degree-of-freedom model as indicated in Fig.(2.5) can be used. Such linear models can not of course be used if the tyres, springs and the valves of the hydraulic actuators (in case of active suspension) are severely non-linear, but this is generally not the case, and the models considered in this work are linear.

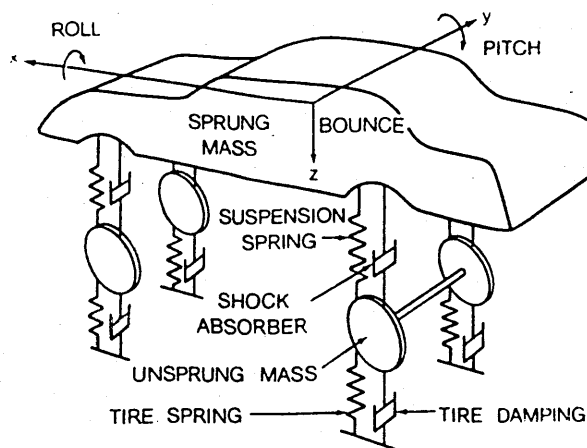


Fig. 2. 2 seven-degrees-of-freedom ride model for passenger cars.

Vehicle Ride Models

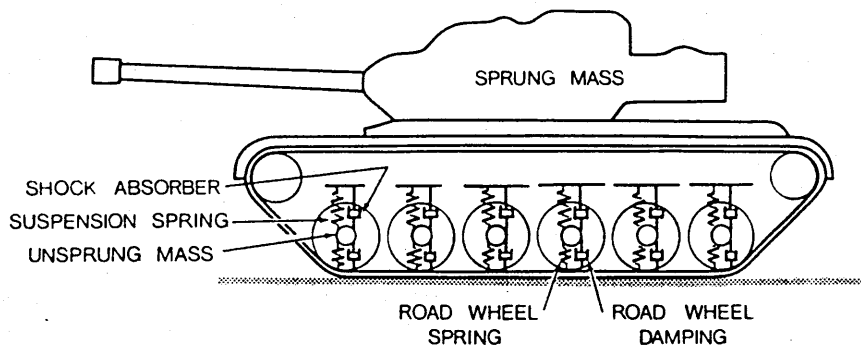


Fig2. 3A ride model for a military vehicle.

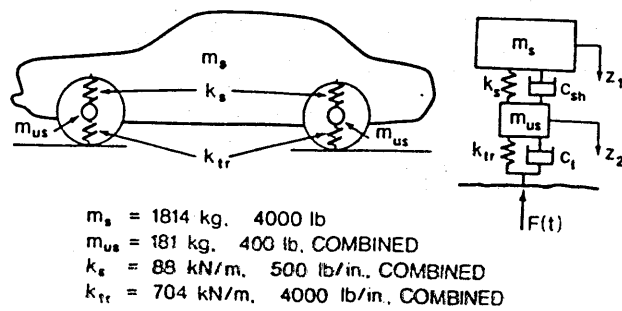


Fig. 2. 4 two-degrees-of-freedom ride model for the sprung and unsprung mass.

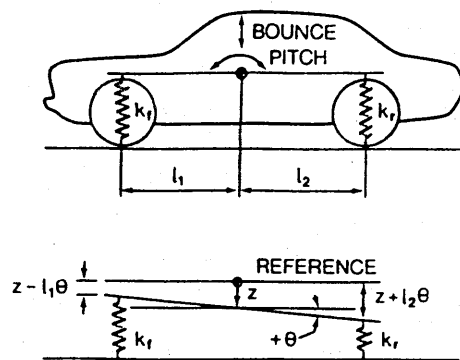


Fig 2 . 5A two-degrees-of-freedom ride model for pitch and bounce of the sprung mass.

2.4 TYPES OF VEHICLE SUSPENSION SYSTEM:

Various types of suspension system are commonly used as follows [7]:

- ▶ Passive systems; These contain only springs and dampers with the possible addition of self leveling systems which typically involve delays of several seconds and are designed to compensate for variations in static load, (Fig 2.6a).
- ▶ Active systems; the springs and dampers of passive systems are replaced by or connected in combination with hydraulic actuators. Various transducers provide feedback signals to these actuators which apply the force required according to some control law. This control law may use information obtained from anywhere in the system, and the determination of a control law which gives good performance is clearly of great importance, (Fig 2.6c).
- ▶ Semi-active systems; are closely related to the active systems just described, the main difference being that the actuators are replaced by controllable dampers, (Fig 2.6b).

Because of the limitations of passive suspensions, there are currently a number of activities to develop active suspensions [5-11], using various sensing devices and feedback controllers (Figs. 2.6, 2.7). They also require a suitable power source, usually hydraulic.

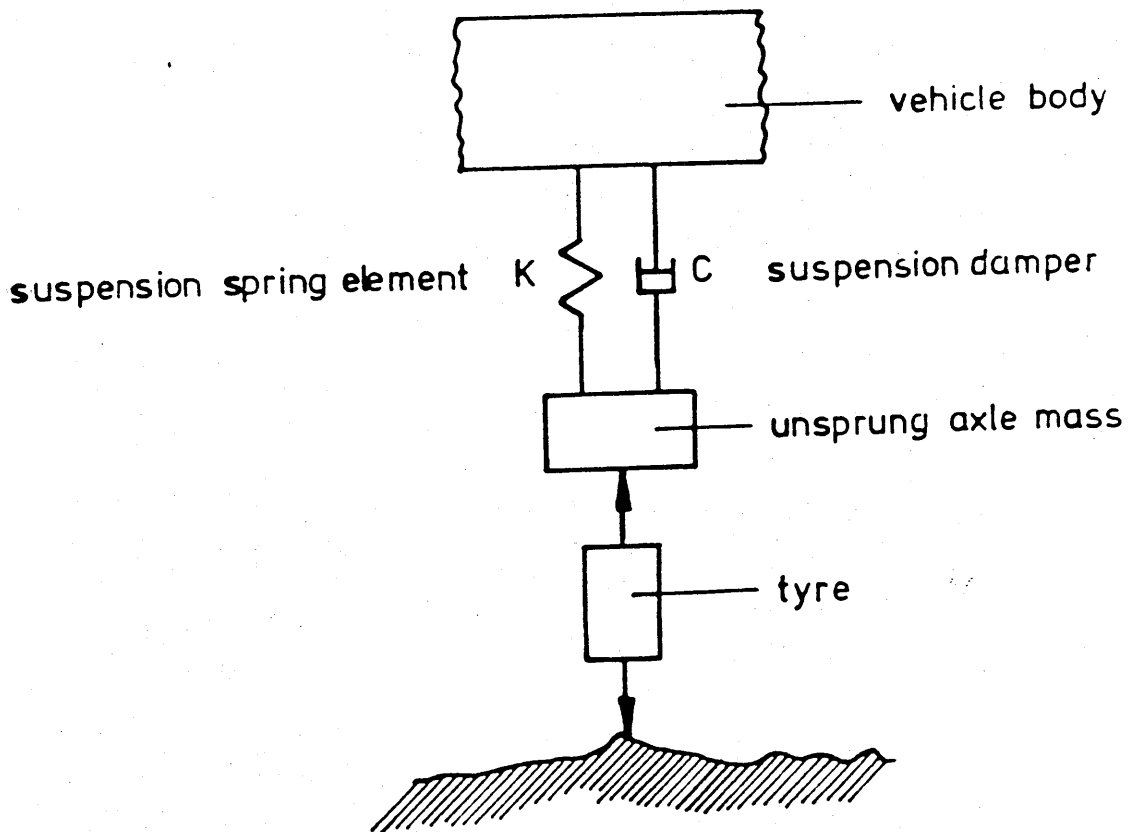


Fig 2.6a Passive suspension system

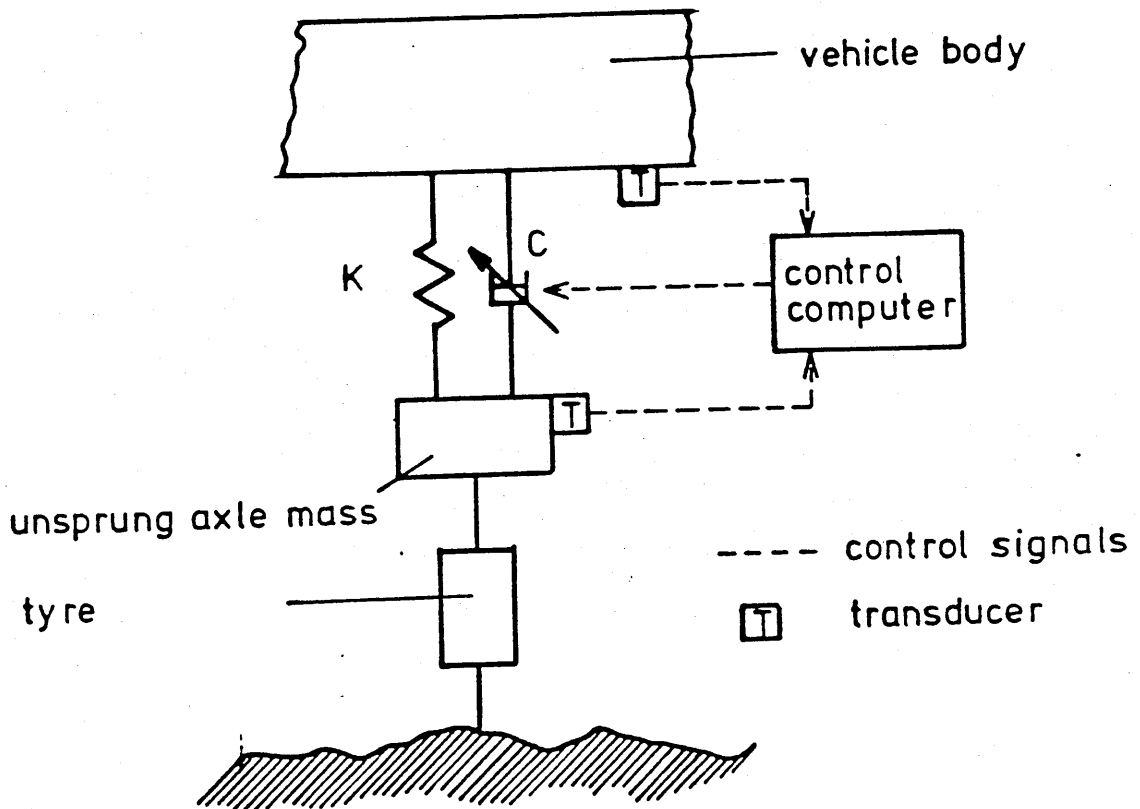


Fig 2.6b Semi- active suspension system

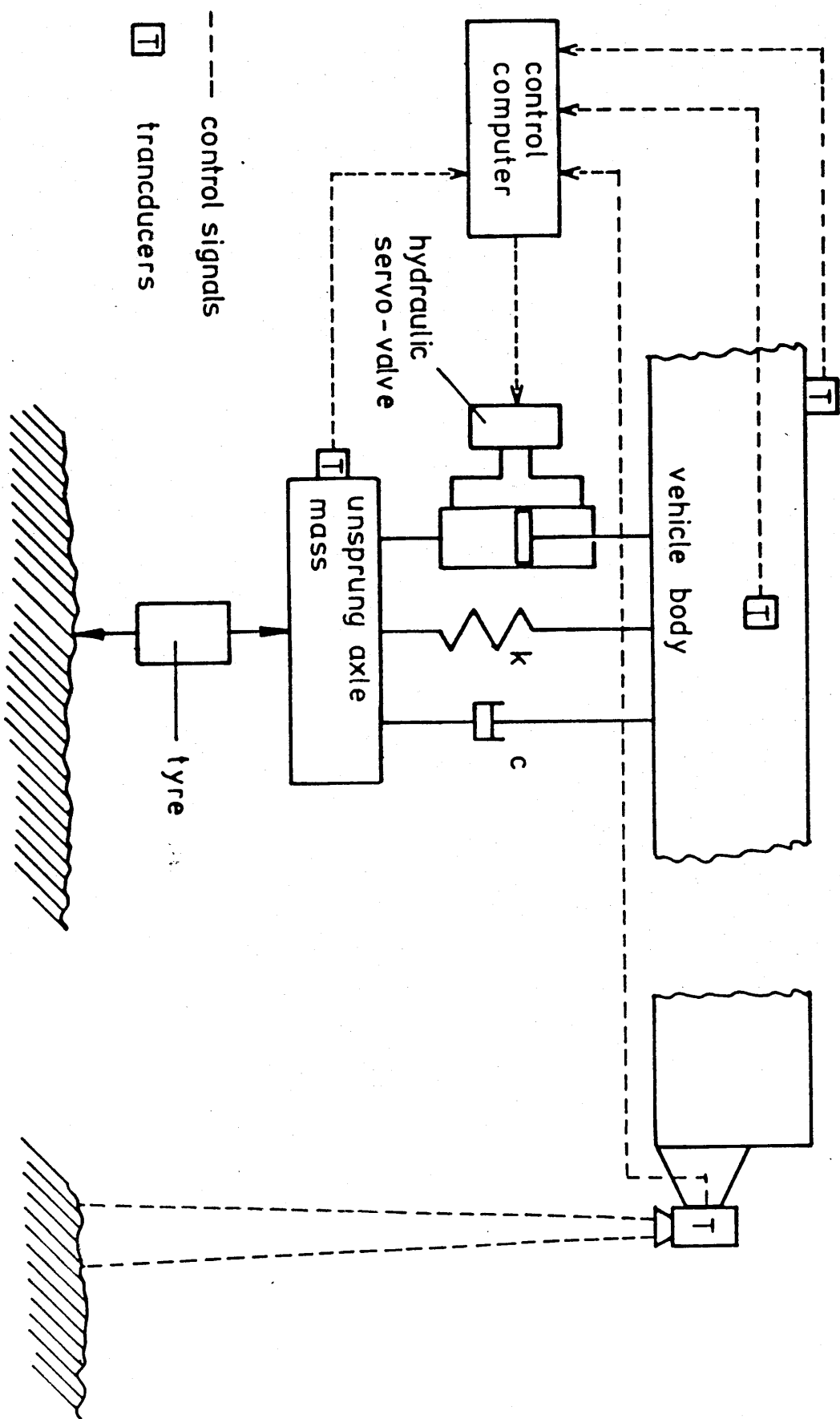


Fig 2.6c Schematic of fully active vehicle suspension with ground preview

In this thesis, only active suspension system is considered.

The advantages of active hydraulic suspension can be summarized as follows:

- small sprung mass dynamic deflections under transient excitation.
- suspension characteristics maintained regardless of loading.
- high speed of response.
- improved ground holding and handling.
- possibility of adaptive control to suit various ground conditions.

An active suspension system is shown in block diagram form in Figure 2.7.

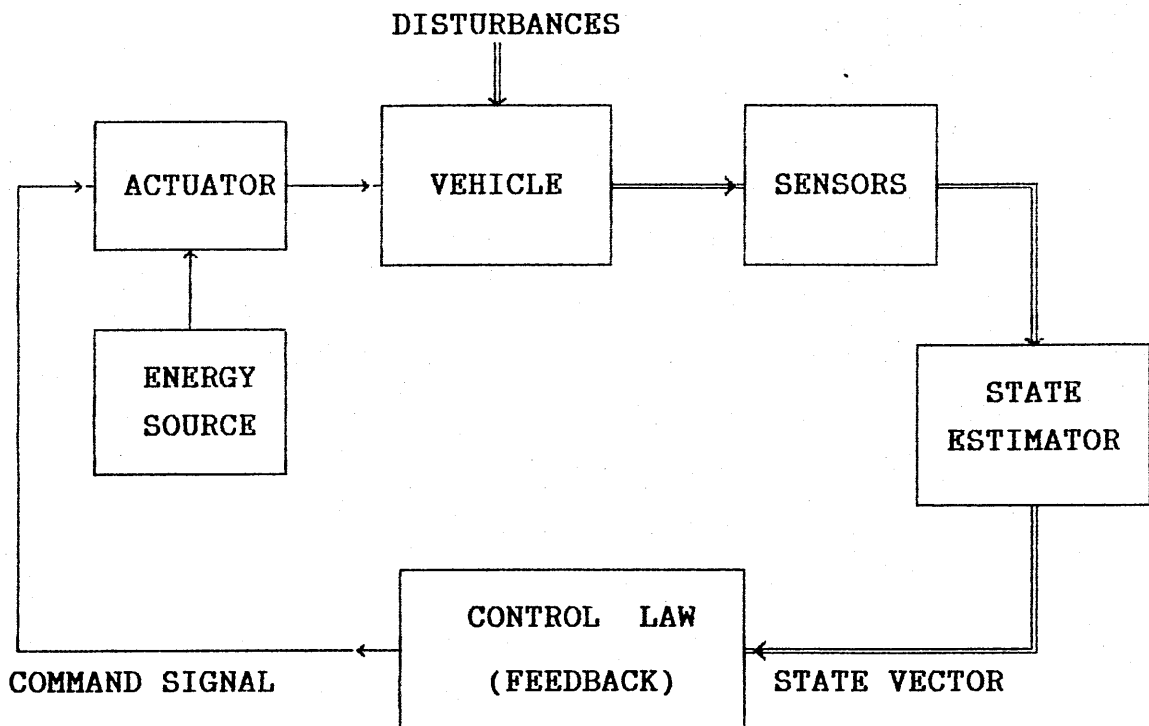


Fig. 2.7 Main Components of an Active Suspension System.

The above advantages of active suspensions are derived from two basic features:

1. Control forces can be generated which do not depend upon energy previously stored by the suspension.
2. The control law can generate forces which are functions of many variables, some of them remotely measured.

Some disadvantages of active systems are their complexity, their installation and maintenance costs and the consequently reduced robustness and reliability. However there are many efforts to overcome these drawbacks.

REFERENCES:

- [1] B. D. Van Deusen, "Human Response to vehicle Vibration", SAE Transactions, Vol. 77 paper 680090, 1969.
- [2] F. Pradko, R. A. Lee, and V. Kaluza, "Theory of human vibration response", ASME Paper No.66-WA/BHF-15, December 1966.
- [3] J. Y. Wong, "The theory of ground vehicles", John Wiley & Sons, 1978.

- [4] L. F. Stikeleather, G. O. Hall, and A. O. Radke, "A study of vehicle vibration spectra as related to seating dynamics", SAE Transaction, Vol.81, Paper 720001, 1973.
- [5] J. Dominy, and D. N. Bulman; "An active suspension for a formula I Grand Prix car", Trans. A.S.M.E. Journal of Dynamic Systems, Measurement and Control, 107, 1985, 73-78.
- [6] D. Metz & J. Maddock; "Optimal Ride Height and Pitch Control for Championship Race Cars", Automatica, Vol.22, No.5, pp. 509-520, 1986.
- [7] R. S. Sharp & D. A. Crolla ; "Road Vehicle Suspension system Design - a Review", Vehicle System Dynamics, No.16 (1987) pp.167-192.
- [8] L. F. Strikeleather, G. O. Hall, and A. O. Radke; "A Study of Vehicle Vibration Spectra as Related to Seating Dynamics", Bostrom Div. ,Universal Oil Products Co. ,Paper no. 720001.
- [9] A. Hac; "Suspension Optimization of a 2-DOF Vehicle Model Using A Stochastic Optimal Control Technique", Journal of Sound and Vibration (1985), 100(3), 343-357.
- [10] R. S. SHARP; "Automotive Product Design" lecture notes 1987, University of Leeds.
- [11] Wili Kortum, "Introduction to System -Dynamics of Ground Vehicles". J. of Vehicle System Design, 1987.
- [12] MIRA Research Report 1979/1 "The Subjective Response Of Occupants To The Noise Inside Vehicles".

CHAPTER 3

ROAD SURFACES

3.1 INTRODUCTION :

It is commonly assumed that a track over rough ground may be regarded as a sample from a stationary random process [1,2] and so, for a vehicle travelling at constant velocity, its response is a stationary random process assuming any initial transient to have died away. In practice, the vehicle motion is usually non-stationary, due either to variation in its velocity or to inhomogeneity in the surface profile, or both [3], but for purpose of analysis, these difficulties are not usually considered.

The statistical distribution of the ground height can be considered to assume a zero mean Gaussian probability distribution, and there is empirical evidence to suggest that this is reasonable [4-7]. This means that a complete statistical description can be obtained from the autocorrelation or power spectral density of the stationary random process. Together with a knowledge of the traverse velocity and (linear) dynamic properties of the vehicle, this allow a response analysis of the system to be undertaken.

Some properties of roads and the modelling of their characteristics are considered in this chapter. For military and agricultural vehicles, the properties of rough ground must also be explored, but this is beyond the scope of the present work.

3.2 THE ROAD AS A HOMOGENEOUS AND ISOTROPIC RANDOM PROCESS :

Two important properties, homogeneity and isotropy are considered here.

3.2.1 HOMOGENEITY :

Homogeneous random surfaces have statistical properties which are independent of co-ordinate translations [4]. Thus if the random variable $z(x,y)$ is homogeneous, where z represents vertical displacement (Fig. 3.2), it follows that:

- (a) The expected value of z , $E[z(x,y)]$, is independent of x,y . Without loss of generality, it is assumed to be zero throughout this analysis, i.e. $E[z(x,y)] = 0$.
- (b) The two dimensional correlation function $R(x_1,y_1;x_2,y_2)$ depends only on the differences $\Delta_1 = (x_2-x_1)$, $\Delta_2 = (y_2-y_1)$ and not on the individual values of x_1,y_1,x_2,y_2 .

Because of homogeneity the autocorrelation function of the profile along any straight line is identical to that taken along any other parallel straight line.

The autocorrelations from homogeneity of the surface, equal:

$$\begin{aligned} E(z(x,y).z(x+\Delta_1,y)) &= E(z(x,y+\Delta_2).z(x+\Delta_1,y+\Delta_2)) \\ \text{which is also equal to;} \\ E(z(x,y).z(x-\Delta_1,y)) &= E(z(x,y+\Delta_2).z(x-\Delta_1,y+\Delta_2)) \\ &= R_a(\Delta_1) \quad \text{say} \quad \dots (3.1) \end{aligned}$$

i.e. from the definition of the homogeneous surface this is equal to the one-dimensional autocorrelation $R(\Delta_1)$ along any line parallel to the longitudinal track (x -axis in this case).

3.2.2 ISOTROPY :

A homogeneous Gaussian random surface is said to be *isotropic* when the cross correlation function $R(\Delta_1,\Delta_2)$ is independent of the direction of the vector separating the two points (x,y) and $(x+\Delta_1,y+\Delta_2)$. The statistics of the plotted surface of $R(\Delta_1,\Delta_2)$ will therefore have circular symmetry about the origin. In other words, statistical properties of isotropic random surfaces are independent of co-ordinate rotations [4,7].

The cross correlations of an isotropic surface equal:

$$\begin{aligned} E(z(x,y).z(x+\Delta 1,y+\Delta 2)) &= E(z(x,y).z(x-\Delta 1,y-\Delta 2)) \\ &= R_x(\rho) \quad \text{say} \quad \dots (3.2) \end{aligned}$$

where;

$$\rho = \sqrt{(\Delta 1^2 + \Delta 2^2)}$$

An auto-spectrum (also called "power spectral density") $S_x(\eta)$ and cross-spectrum $S_{xy}(\eta)$ (where η is the wave number of the surface in cycle/m), are defined as the Fourier transforms of the auto and cross correlations (Eqns 3.1 and 3.2) respectively [5]. These are calculated using various techniques [5,6].

Two parallel profiles separated by a distance " $\Delta 2$ " are considered, as shown in Figs. 3.3 and 3.4 in analyzing the behaviour of four wheeled vehicle. In this thesis, however, we are concerned mainly with quarter car models and the cross correlation/spectrum terms are not relevant. However, further development of the research to deal with more elaborate models would require consideration of these.

As a conclusion, typical road surface are usually considered as homogeneous and isotropic two dimensional Gaussian random process. Complete description of any such process is provided by a single autocorrelation or spectral density function evaluated along any longitudinal track, and thus a single direct spectral density function provides a road surface description. This hypothesis is justified by agreement between theoretical and experimental results presented in the literature [1,2,4,7].

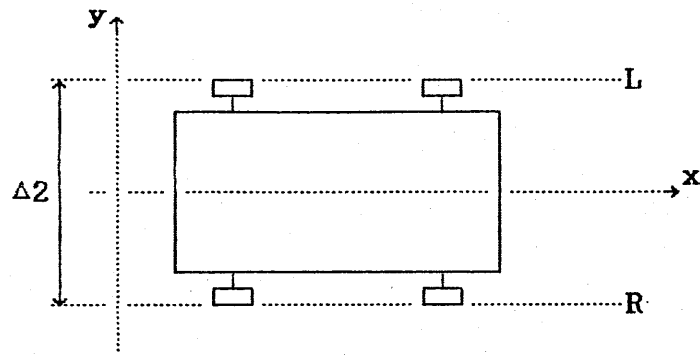
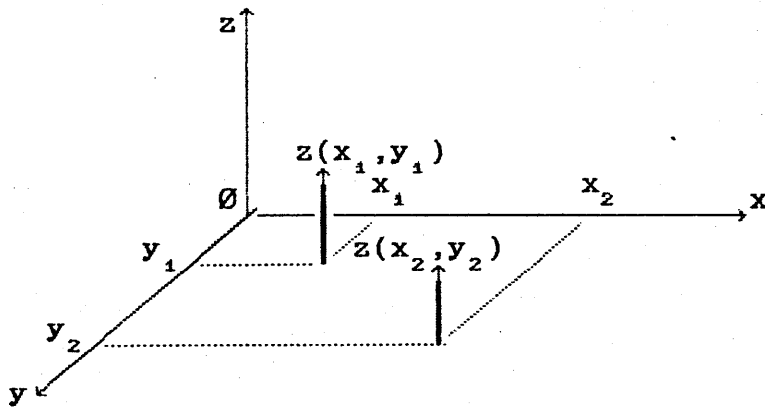


Fig. 3.1 Four-wheel vehicle layout



where;

$$x_2 = x_1 + \Delta 1$$

$$y_2 = y_1 + \Delta 2$$

Fig. 3.2 Height (z) of surface in the x - y plane

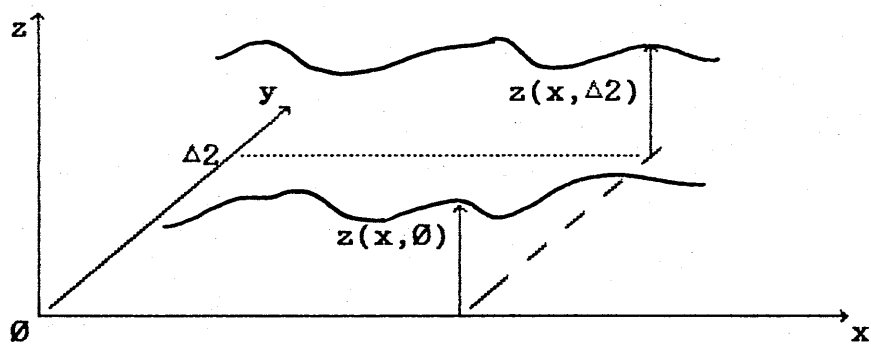


Fig 3.3 Specification of homogeneous road surface
(two tracks)

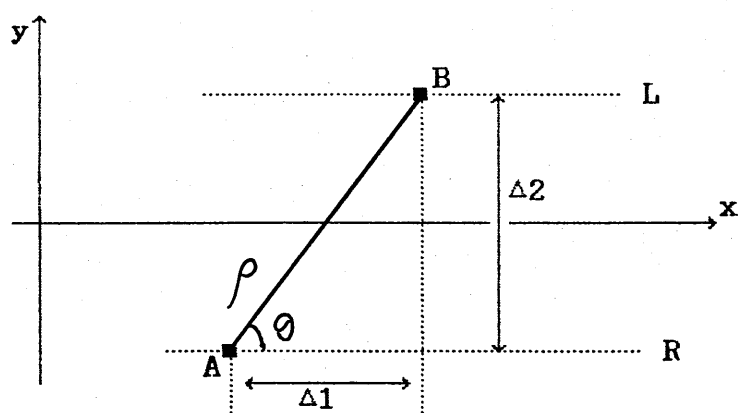


Fig. 3.4 Two-dimensional road surface parameters

3.3 THE SPECTRAL DENSITY OF A ROAD SURFACE:

Having assumed that the road surface is a homogeneous and isotropic stochastic process with zero mean Gaussian amplitude distribution, it is necessary to specify its spectral density.

It is suggested by Dodds [1] and Kamash & Robson [7], that the one sided spectral density $S(\eta)$ can be approximated by a simple analytical form for different types of road using the following formula:

$$S(\eta) = \begin{cases} S(\eta_0) \cdot (\eta/\eta_0)^{-w_1} & \eta \leq \eta_0 \\ S(\eta_0) \cdot (\eta/\eta_0)^{-w_2} & \eta \geq \eta_0 \end{cases} \quad \dots (3.3)$$

where;

$S(\eta_0)$ is a "roughness coefficient" (value of the spectral density at the discontinuity frequency η_0), η_0 is a datum wave number, w_1 and w_2 are positive constants which are functions of the road surface characteristics.

From tests carried out by MIRA [8], η_0 is found:

$$\eta_0 = 1/2\pi \quad (\text{cycle/m})$$

$S(\eta_0)$ provides a basic roughness coefficient, and a measure of the comparative roughness of two roads is given by the square root of the ratio of their respective $S(\eta_0)$ values.

For a given $S(\eta_0)$ a high value of w_1 indicates a road which increases in roughness at longer wavelengths. Conversely a road with high w_2 suggests a road which increases in roughness at shorter wavelengths.[1]

The variance of the process is given by;

$$\sigma^2 = 2 \int_0^{+\infty} S(\eta) d\eta \quad \dots (3.4)$$

Although Eqn.(3.3) defines road roughness by using a simple mathematical expression and is ideal for computational or design purposes, it does not allow easy mathematical analysis, as $S(\eta)$ becomes infinite as $\eta \rightarrow 0$. This inconvenience can be overcome in various ways. In ref.[1] it is suggested to filter the road spectrum with a high pass filter with transfer function:

$$G(j\eta) = \frac{j\lambda 2\pi\eta}{1 + j\lambda 2\pi\eta} \quad \dots (3.5)$$

where λ is chosen to be a length greater than any wave length of interest (wave length = $1/\eta$).

The road spectral density, Eqn 3.3, becomes:

$$S(\eta) = \begin{cases} \frac{[(2\pi)^2 \eta^2 \lambda^2]}{[1 + \lambda^2 (2\pi)^2 \eta^2]} S(\eta_0) \cdot (\eta/\eta_0)^{-w_1} & \eta \leq \eta_0 \\ \frac{[(2\pi)^2 \eta^2 \lambda^2]}{[1 + \lambda^2 (2\pi)^2 \eta^2]} S(\eta_0) \cdot (\eta/\eta_0)^{-w_2} & \eta \geq \eta_0 \end{cases} \quad \dots (3.6)$$

Using the above roughness coefficients, road surfaces can be classified. From extensive data gathered by MIRA[8] the parameters given in Table 3.1 have been published for typical road types.

A wave number range of 0.01 - 10 cycle/m is typically considered [15,16]. This range corresponds to a frequency range $0.5\text{Hz} \leq f \leq 50\text{Hz}$ for vehicle velocities in the range 5-50 m/s (11-110 mile/h).

The wave length " λ " in Eqn. 3.5 is chosen as twice the maximum wave length of interest, i.e. $\lambda = 2/0.01 = 200\text{m}$, and the profile spectral density (Eqn. 3.6) is assumed zero for higher or lower wave numbers of interest [15].

Having found the profile spectral density, and assuming a car traverses a surface at a constant speed, U m/s, the excitation frequency, f Hz, and the wavenumber, η , are related by [16]:

$$f = U\eta \quad \dots (3.7)$$

Road class	$S(\eta_o)$ $\times 10^{-6} \frac{\text{m}}{\text{cyc}}$	W_1		W_2	
		Mean	S. dev.	Mean	S. dev.
►Motor way					
• Very good	2 - 8	1.945	0.464	1.360	0.221
• Good	8 - 32				
►Principal rd					
• Very good	2 - 8	2.05	0.487	1.440	0.266
• Good	8 - 32				
• Average	32 - 128				
• Poor	128 - 512				
►Minor roads					
• Average	32 - 128	2.28	0.534	1.428	0.263
• Poor	128 - 512				
• Very poor	512 - 2048				

Table 3.1
Classification of roads based on road spectra presented by MIRA [8]
 $\eta_o = 1/2\pi$ cycle/m

3.4 COMPUTER SIMULATION OF A RANDOM ROAD DISTURBANCES:

For a certain road profile (of a type discussed in the previous paragraph), and assuming the vehicle is travelling at a constant speed, we simulate the random noise applied to the vehicle suspension using the ISIS simulation package on the VAX 11-750 computer.

In this thesis typical roads disturbances simulated by the computer have the properties as shown in Table 3.2 in which two cases are considered (cf. Table 3.1), good principal road and very poor minor (paved) road, so, any road type will be in between. The power spectral density of these roads are shown in Figs. (3.5) and (3.6).

Road class	$S(\eta_o)$ $\times 10^{-6} \frac{m^3}{cyc}$	w_1 Mean	w_2 Mean	σ^2 (mm ²)	σ (mm)
Good principal road (motor way)	13.8	1.945	1.360	9.0	3.0
Very poor minor road	2048	2.28	1.428	300.0	17.32

Table 3.2
Classification of roads used in this thesis

To simulate this process by the computer, a suitable filter must be designed using white noise input as discussed in the next paragraphs.

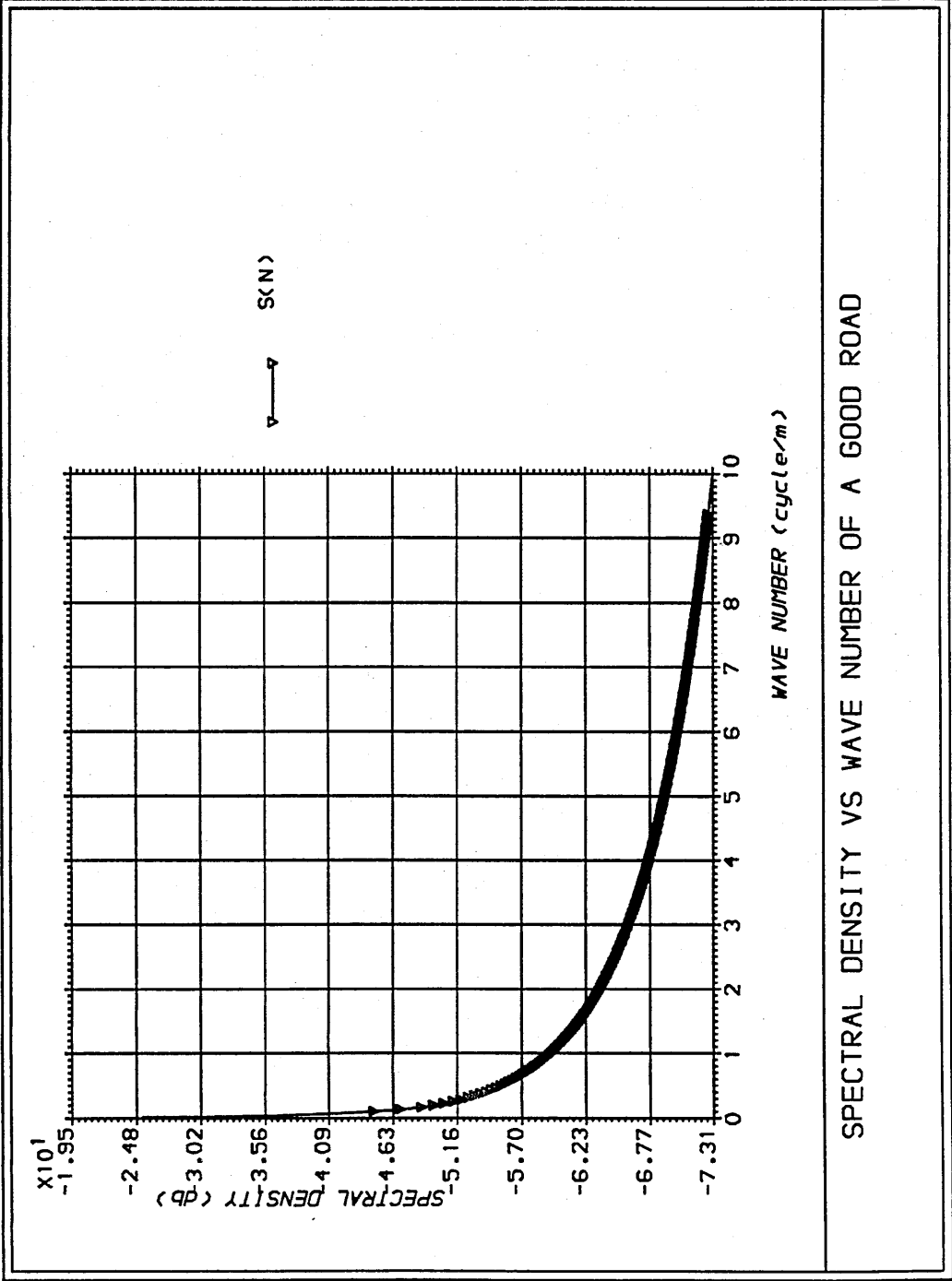


Fig. 3.5 Power spectral density of a good road

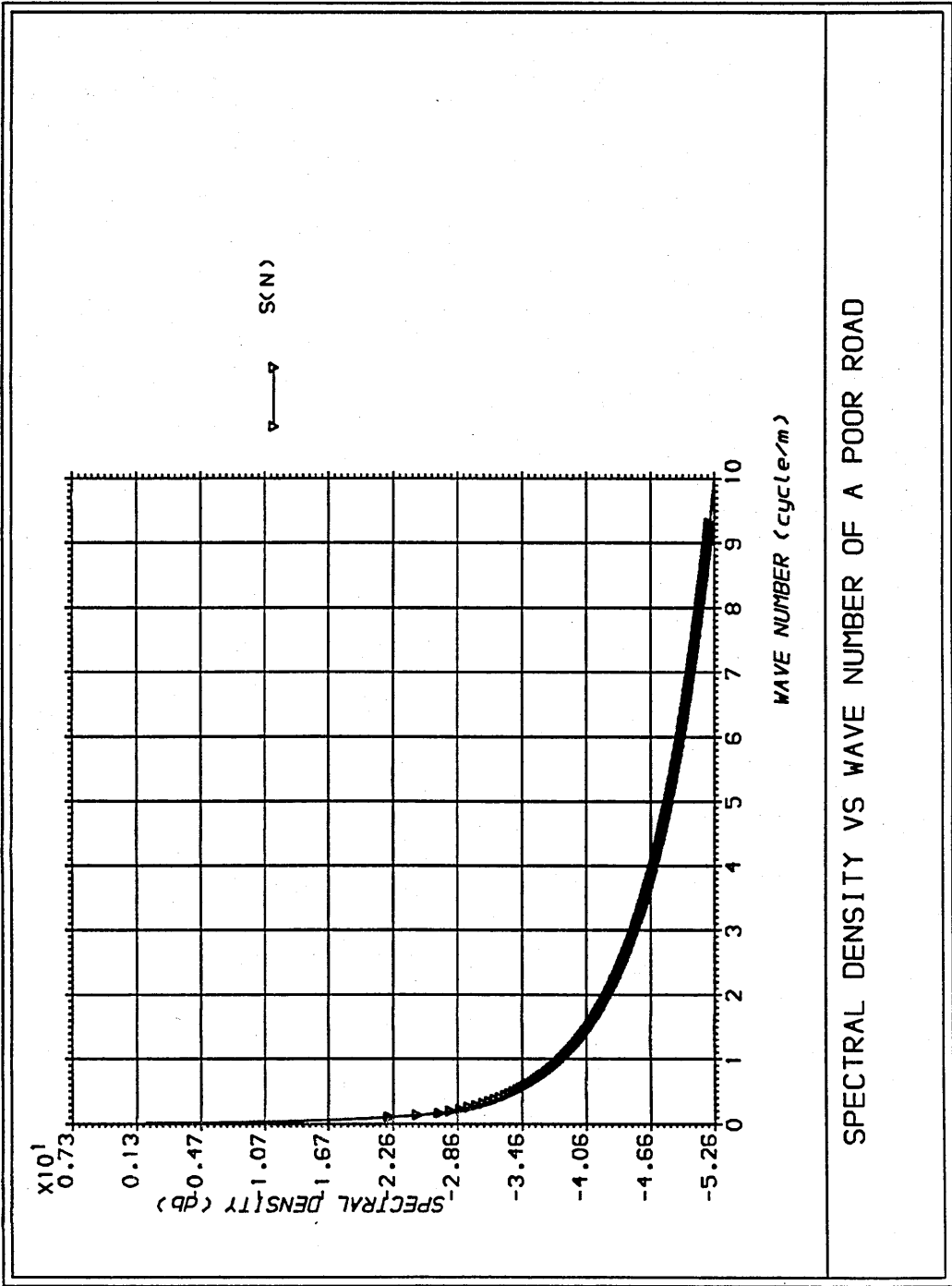


Fig. 3.6 Power spectral density of a poor road

3.5 PSEUDO-RANDOM SIGNAL GENERATION :

Pseudo-random signals of various kinds can easily be generated by the computer. The simplest way to generate random numbers with Gaussian distribution is to use the central limit theorem (which states that; if a random process consists of the sum of a large number of independent processes, the distribution of the overall process is Gaussian, or normal) [9,10].

In other words, if a series of samples, of size N , are taken from a population $\{(r)\}$ (not necessary normal), then the sample means tends to the normal distribution as N increases. Thus, from an independent, uniformly distributed number (r) , an independent, Gaussianly distributed random number can be generated by taking the mean of samples of, say, 10 to 20 elements from that population.

3.5.1 EVENLY DISTRIBUTED RANDOM NUMBERS :

Evenly (uniformly) distributed random numbers can easily be generated by the multiplicative congruential formula [10]:

$$r(n+1) = [\beta r(n) + \alpha] \text{ modulo } \mu^p \quad \dots (3.8)$$

where μ is the radix used,
 p is the word length,
 α, β are constants.

The numbers are evenly distributed between 0 and μ^p . To obtain an even random distribution $r'(n)$ with limits $[a, b]$ the following formula is used:

$$r'(n) = r(n) \cdot (b-a) / \mu^p + a \quad \dots (3.9)$$

in this case the mean and variance of $r'(n)$ are:

$$E[r'(n)] = (b-a)/2 \quad \dots (3.10)$$

$$\text{var}[r'(n)] = (b-a)^2/12 \quad \dots (3.11)$$

We use the case of evenly distributed random numbers between 0 - 1 (the typical limit used in the NAG library routines). So, Eqns. (3.10) & (3.11) become:

$$E[r'(n)] = 0.5 \quad \dots (3.12)$$

$$\text{var}[r'(n)] = 1/12 \quad \dots (3.13)$$

3.5.2 RANDOM NUMBERS WITH GAUSSIAN DISTRIBUTION :

A sequence of uniformly distributed numbers $r'(n)$ can conveniently be converted to a set of Gaussian distributed random numbers by a formula which depends on the central limit theorem: [10]

$$Z(n) = \frac{1}{N} \sum_{i=0}^{N-1} r'(nN - i) \quad \dots (3.14)$$

N is theoretically large, but N between 10-20, typically, gives a sufficiently good approximation to a Gaussian distribution. In this case;

$$E[Z(n)] = E[r'(n)] \quad \dots (3.15)$$

$$\text{var}(Z(n)) = 1/N [\text{var}(r'(n))] \quad \dots (3.16)$$

3.6 GENERATING GAUSSIAN WHITE NOISE ON THE COMPUTER :

Equations 3.8 & 3.9 generate a uniformly distributed sequence between 0 and 1 (Fig. 3.7), and from this sequence a white noise with Gaussian distribution is obtained (Eqn. 3.14). Colored noise of any power spectral density can then be generated using an appropriate filter.

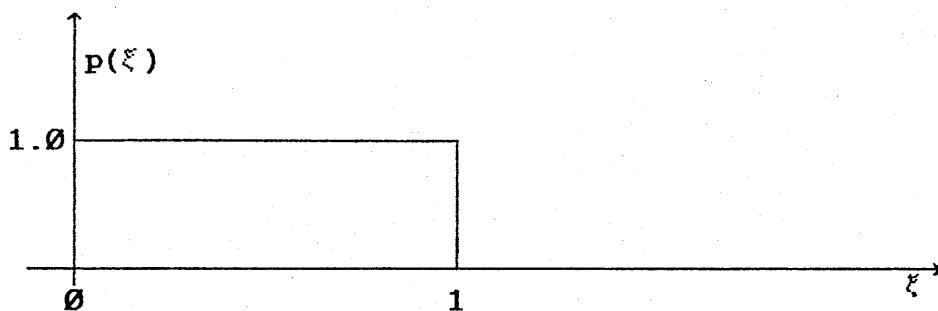


Fig. (3.7) PDF of a Uniformly Distributed Sequence.

The algorithm producing a sequence of white noise $Z(n)$ with zero mean and variance $1/12N$ for a chosen value of $N = 20$ is:

$$\text{var}(Z(n)) = \frac{1}{12N} = \frac{1}{240} \quad \dots (3.17)$$

$$Z(n)_{[E=0.0]} = \frac{1}{20} \sum_{i=0}^{19} r'(nN-i) - 0.5 \quad \dots (3.18)$$

Fig.(3.8) shows a plot of white noise sample generated using Eqns. 3.17 and 3.18.

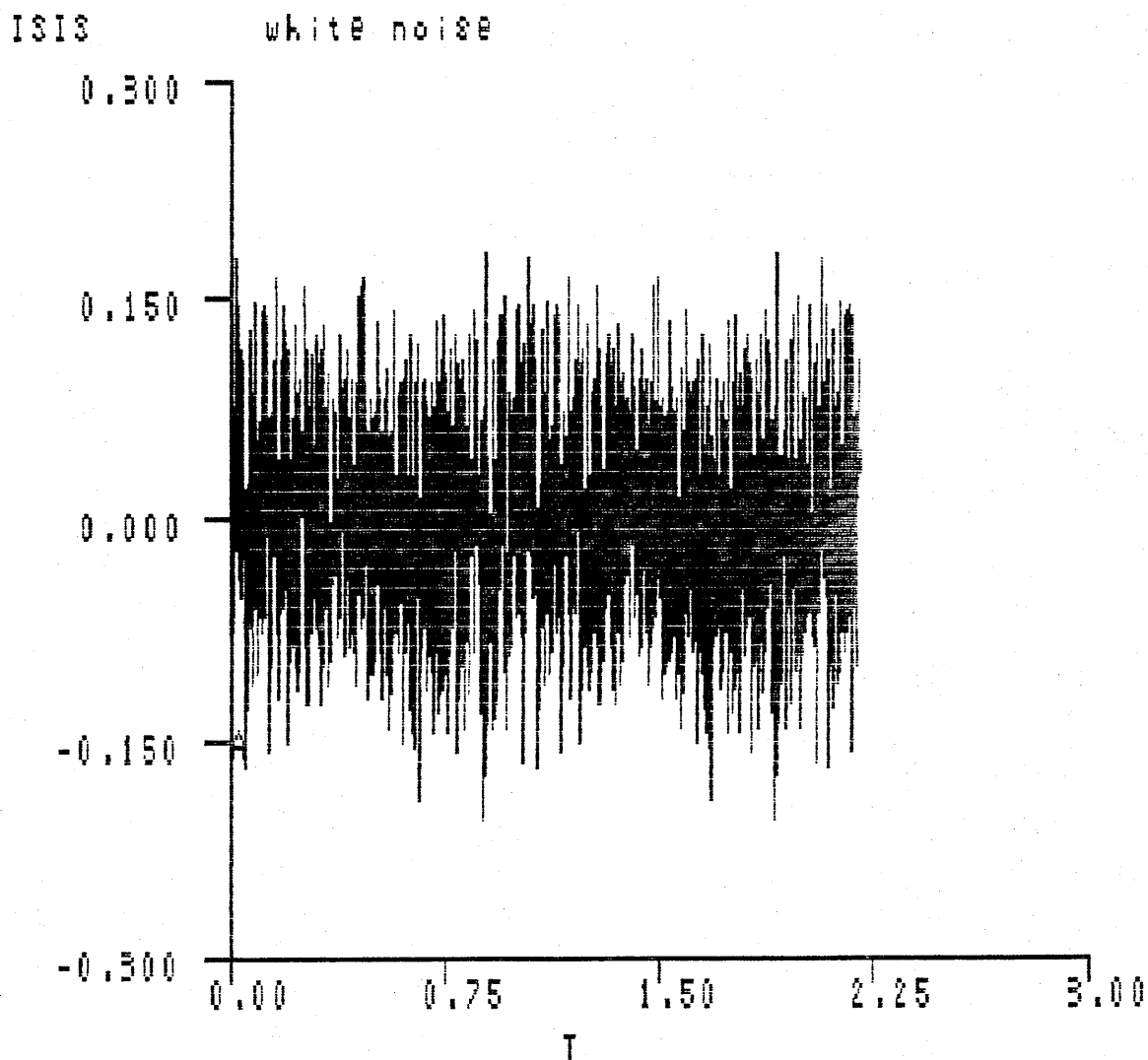


Fig. 3.8 A White Noise Sample

3.7 GENERATING COLORED NOISE ON THE COMPUTER :

To generate a random process which has a power spectral density as discussed in Section 3.3, a digital filter must be used to give the required PSD, as indicated in Fig. 3.9.

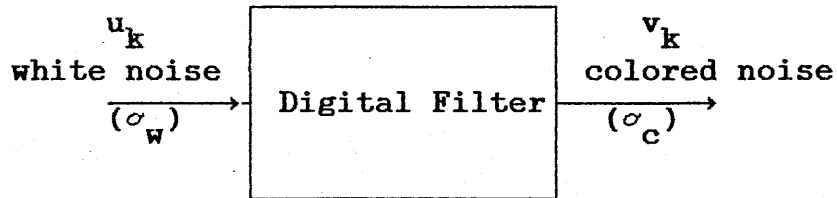


Fig. 3.9 Colored Noise Generation by Computer

A Finite Impulse Response (FIR) filter is considered in this work [17], which can be represented as Fig 3.10.

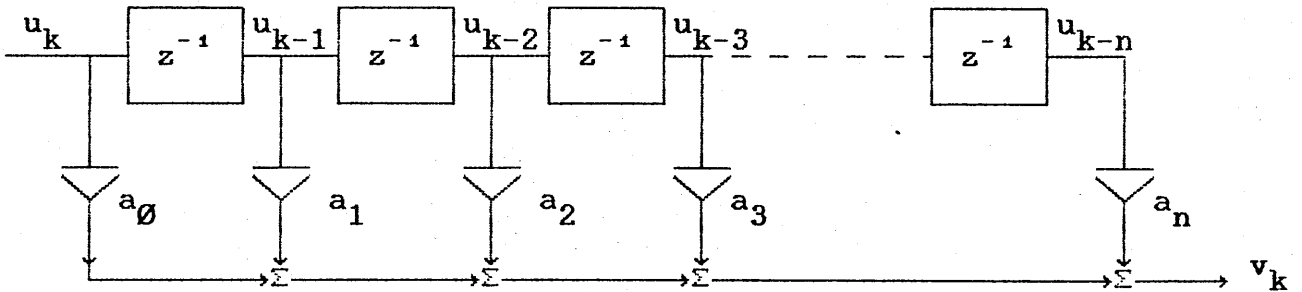


Fig. 3.10 Finite Impulse Response Digital Filter

The transfer function of this filter is:

$$H(z) = \sum_{i=0}^n a_i z^{-i} \quad \dots (3.19)$$

The output is:

$$v_k = \sum_{i=0}^n a_i u_{k-i} \quad \dots (3.20)$$

The impulse response will be v_k for;

$\{u_k\} = \{1, 0, 0, \dots, 0\}$, namely;

$\{v_k\} = \{a_0, a_1, a_2, \dots, a_{n-1}, a_n, 0, 0, \dots\}$

and must clearly be finite, hence the title finite impulse response (FIR). Because there is no feedback, this class of filters is also known as *non-recursive*.

To find the filter coefficients ($a_0, a_1, \dots, a_n$), an Inverse Fourier Transform is taken of the specified power spectral density (Eqn. 3.6). This is then windowed, to reduce the filter ripples, using a Hamming window which has a transfer function as [17]:

$$D_{rh} = \frac{1}{2}(1 + \cos \frac{\pi r}{n}), \quad r=0,1,2, \dots, n \quad \dots (3.21)$$

This yields the filter coefficients.

A program, COLFIL, has been designed which does all this neatly.

Two 8th order filters are designed in this way for the two cases considered in Sect. 3.4 (i.e. good and poor road).

The calculated coefficients by the COLFIL program are:

(1) Good road:

$$\begin{aligned} a_0 &= 5.136 \times 10^{-2} \\ a_1 &= 3.916 \times 10^{-2} \\ a_2 &= 3.531 \times 10^{-2} \\ a_3 &= 3.433 \times 10^{-2} \\ a_4 &= 3.433 \times 10^{-2} \\ a_5 &= 3.531 \times 10^{-2} \\ a_6 &= 3.916 \times 10^{-2} \\ a_7 &= 5.136 \times 10^{-2} \end{aligned}$$

(2) Poor road:

$$\begin{aligned} a_0 &= 1.050 \\ a_1 &= 0.812 \\ a_2 &= 0.733 \\ a_3 &= 0.712 \\ a_4 &= 0.712 \\ a_5 &= 0.733 \\ a_6 &= 0.812 \\ a_7 &= 1.050 \end{aligned}$$

Finally, to realize the filter, convolution of the filter coefficients with a white noise sample gives the colored noise output.

The package MSPECT available on the PDP-11 computer in the signal processing lab (DESD) is used in this work. Fig 3.11 shows the first filter operation as follows;

Fig 3.11a : A white noise sample.

Fig 3.11b : The colored noise filter output.

Fig 3.11c : The PSD of the colored noise filter output.

Fig 3.12 shows the second filter operation, as described in Fig 3.11.

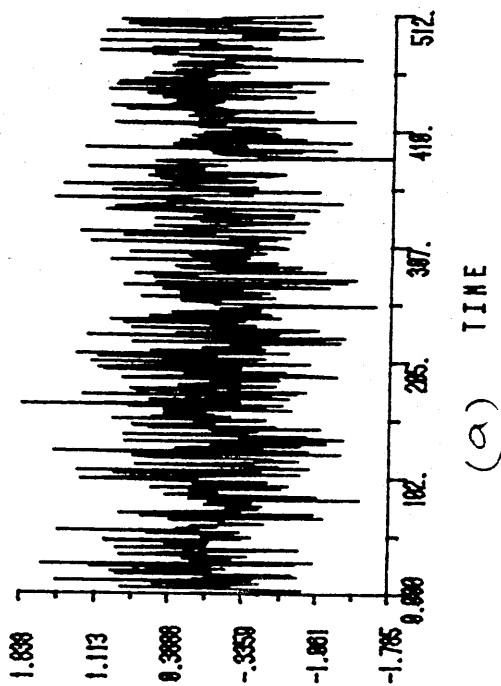
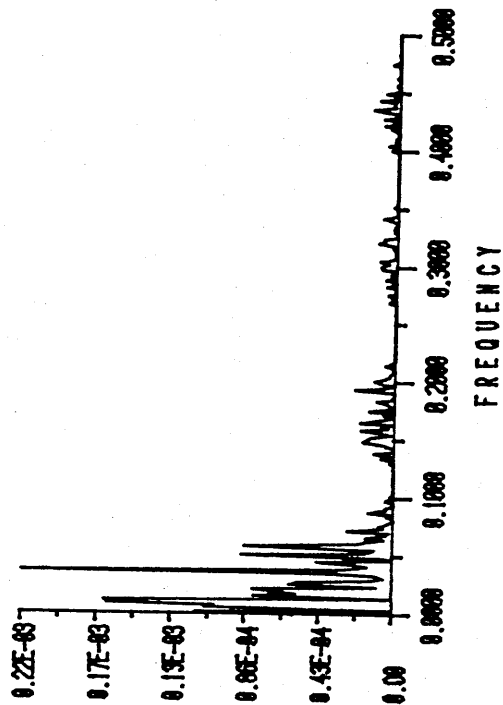
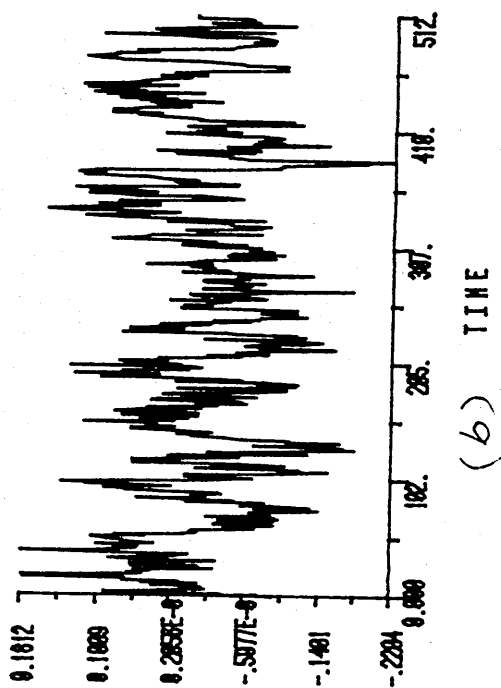


Fig 3.11 Filter Operation (Good Road)

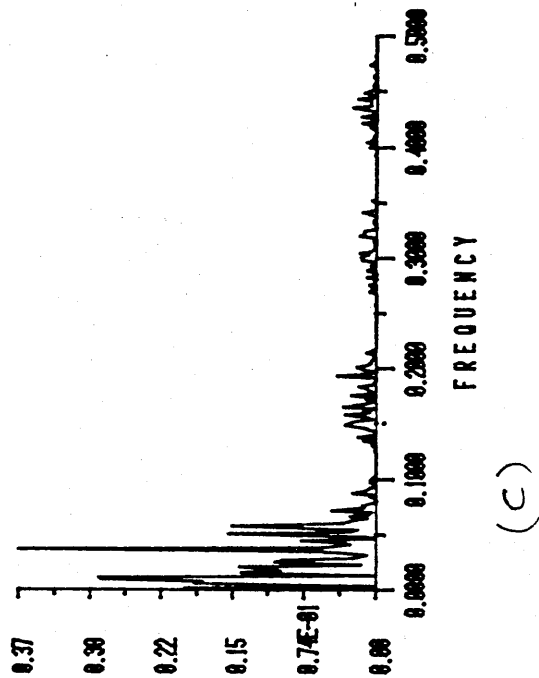
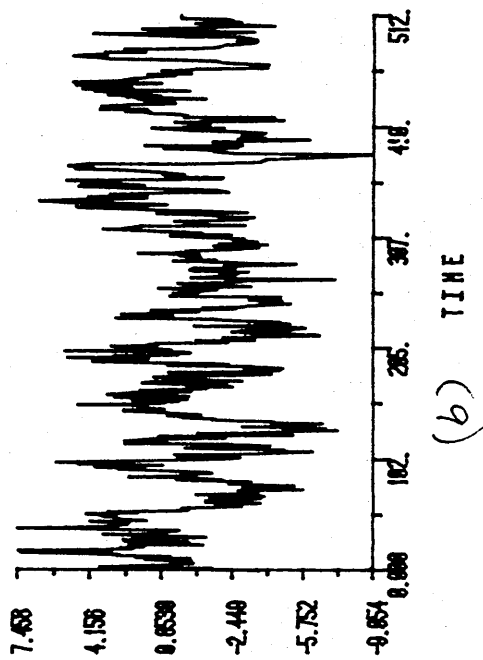
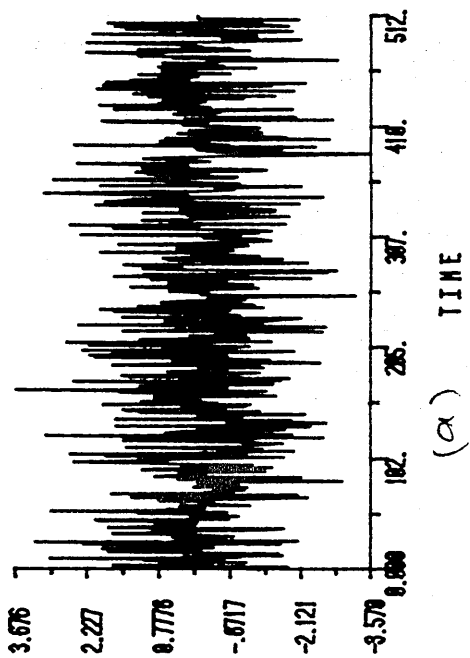


Fig 3.12 Filter Operation (Poor Road)

3.8 SIMULATING VARIOUS NOISES:

The ISIS is used for simulating various road types discussed so far. The white noise sample signal (Sect. 3.6, using Eqns. 3.17 & 3.18) is shown in Fig. 3.8.

To achieve the right variance for the selected road type (according to Table 3.2), the filter output (see Fig. 3.9) must be multiplied by the right scale factor as follows:

let;

σ_w^2 = variance of the white noise input.

σ_w = standard deviation of the white noise input.

σ_c^2 = variance of the colored noise output.

σ_c = standard deviation of the colored noise output.

F_g = filter gain (or attenuation).

To find (F_g) for the filters designed in Sect. 3.7, the variances of the input and output of the data sample to/from the filter are calculated using the following formula [18]:

$$\sigma_w^2 = E(u_k^2) - (E(u_k))^2 \quad \dots (3.22)$$

$$\sigma_v^2 = E(v_k^2) - (E(v_k))^2$$

and the filter gain is the standard deviation of the output divided by the standard deviation of the input, as shown in the following results;

Case I: Good road test.

$$\text{Filter output } \sigma_c^2 = 0.495 \times 10^{-2}, \quad \sigma_c = 0.7035 \times 10^{-1}$$

$$\text{Filter input } \sigma_w^2 = 0.401, \quad \sigma_w = 0.633$$

$$\text{Filter gain } F_g = \frac{\sigma_c}{\sigma_w} = 0.7035 \times 10^{-1} / 0.633 = 0.1111$$

Case II: Poor road test.

$$\text{Filter output } \sigma_C^2 = 2.104, \quad \sigma_C = 1.451$$

$$\text{Filter input } \sigma_W^2 = 0.401, \quad \sigma_W = 0.633$$

$$\text{Filter gain } F_g = 1.451/0.633 = 2.292$$

To get the right variances from the ISIS for the two cases considered in this work, the scale factor will be as follows:

Using the random number generator RAND(R) available in the ISIS for $N=20$ (Eqns. 3.17, 3.18), $\sigma_W^2 = 1/240$, $\sigma_W = 0.06455$ so for;

$$\text{Good road, } \sigma_C^2 = 9\text{mm}^2, \quad \sigma_C = 3\text{mm (see Table 3.2)}$$

$$\text{scale factor} = \frac{3}{0.1111 \times 0.06455} = 418.282$$

$$\text{Poor road, } \sigma_C^2 = 300\text{mm}^2, \quad \sigma_C = 17.32\text{mm}$$

$$\text{scale factor} = \frac{17.32}{2.292 \times 0.06455} = 117.0677$$

Fig. 3.13 simulates a good road, while Fig. 3.14 simulates a poor road for a constant vehicle speed. Various speeds can be chosen in the run time of the ISIS by changing the sampling rate (number of output communication interval NOCI), but we will consider one constant vehicle speed (NOCI=10 in this work).

These generated signals are used as the road disturbances in simulations of the vehicle's suspension system as discussed in Chapters 6 & 7.

REFERENCES :

- [1] C. J. Dodds and J. D. Robson, "The Description of Road Surface Roughness", Journal of Sound and Vibration (1973) vol.31 no.2 175-183.
- [2] C. J. Dodds and J. D. Robson, "Stochastic Road Inputs and Vehicle Response", Vehicle System Dynamics, vol.5 (1975/76), pp.1-13.
- [3] R. F. Harrison, "The Non-stationary Response of Vehicles on Rough Grounds", Institute of Sound and Vibration Research, Faculty of Engineering and applied science, University of Southampton, Ph.D. Thesis sept.1983.
- [4] A. N. Heath, " Application of The Isotropic Road Roughness Assumption", Journal of Sound and Vibration (1987) 115(1), pp.131-144.
- [5] J. S. Bendat, and A. G. Piesol, "Random Data: Analysis and Measurement Procedures", Wiley-Interscience, New York 1971.
- [6] J. W. Cooley, P. A. W. Lewis & P. D. Welch, "The Fast Fourier Transform Algorithm", J. of Sound and Vibration vol.12, 1970 315-338.
- [7] K. M. A. Kamash and J. D. Robson, "Implication of Isotropy In Random Surfaces", Journal of Sound and Vibration (1977) vol.54(1), 131-145.
- [8] R. P. LaBarre, R. T. Forbes and S. Andrew "Motor Industry Research Association Report" No,1970/5. The measurement and analysis of road surface roughness ,(1969).
- [9] K. Kafadar, "Gaussian White-Noise Generation for Digital Signal Synthesis", IEEE Trans. on Instrumentation and Measurement, Vol. IM_35, No.4, Dec.1986.

- [10] J. A. Borrie, "Modern Control Systems, A Manual of Design Methods", Prentice/Hall International 1986.
- [11] VAX NAG_library manual NAGFLIB:1443/Ø:MK6:MAY77
- [12] Dr. Russell Lecture notes, SESD, Cranfield Institute of Technology, 1987.
- [13] Yassine Saeed Al-Masri, "Adaptive State Estimation Compared With Proportional Navigation for Bearing Only Missile Guidance", Phd thesis, CIT August 1986.
- [14] A. Hac, "Suspension Optimization of a 2-DOF vehicle Model Using a Stochastic Optimal Control Technique", Journal of Sound and Vibration (1985) 100(3).
- [15] K. M. A. Kamash and J. D. Robson, "The Application of Isotropy in Road Surface Modelling", Journal of Sound and Vibration (1978) 57(1),89-100
- [16] R. S. Sharp and D. A. Corolla, " Road Vehicle Suspension System Design - a review", Vehicle System Dynamics, 16(1987), 167-192.
- [17] Signal Processing MSc course Lecture Notes (1986-7), DESD, CIT.
- [18] Stochastic Systems MSc course Lecture notes (1986-7), DESD, CIT.

CHAPTER 4

STATE ESTIMATION AND OPTIMAL FILTERING WITH LINEAR SYSTEM MODELS

4.1 INTRODUCTION:

In many practical processes it is not possible to apply modern feedback control laws directly, either because all the state variables are not accessible for direct measurement or because the number of measuring devices is limited [5]. In the case of ground vehicles, as considered in this thesis, this problem does appear because all the vehicle states, in particular the height above the mean ground level, are not available. In this situation, the states of the system are estimated by a "state estimator" (Fig.4.1) where $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are process and measurement noise respectively, $\mathbf{x}(t)$ is the system states vector, $\mathbf{y}(t)$ is the system output (measurements) vector and $\mathbf{u}(t)$ is the system input vector.

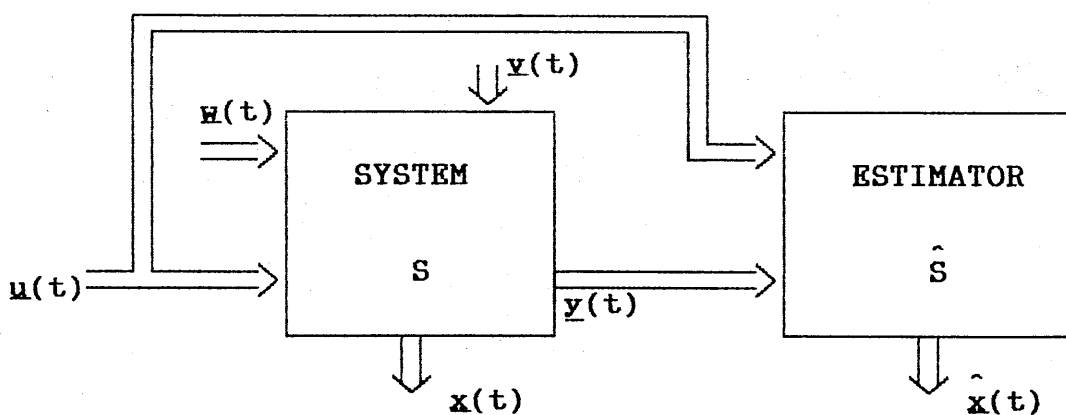


Fig. 4.1 Block Diagram of a System and an Estimator.

A Kalman filter is a state estimator designed to estimate the states of a stochastic system using the system input and output as data, in the presence of process and measurement noise (Fig. 4.1) [2]. Various forms of Kalman filter exist; the basic optimal form applies to linear time variant systems while suboptimal forms apply to linear time invariant (LTI) systems and various extended forms have been successfully designed for non linear systems.[2]

We are suggesting here that the forms of Kalman filter best suited to active vehicle suspension using a microprocessor are:

- (1) The sub-optimal Kalman filter, which is an asymptotic state estimator (for LTI systems) after calculating the optimal gain matrix off-line, using the conventional Kalman filter algorithm shown in the next paragraphs of this chapter.
- (2) The optimal Kalman filter, with updated gain matrix to deal with changing road conditions.
- (3) To avoid a matrix inversion in type (2), the UD method of Kalman filter is suitable for on-line use of a microprocessor to calculate this gain matrix, and this is proposed.

4.2 THE ASYMPTOTIC STATE ESTIMATOR:

An observable (LTI) system (S) can be modeled by the m-dimensional dynamical equations:

S : (a continuous time system)

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \quad \dots (4.1)$$

$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) \quad \dots (4.2)$$

also for a discrete system :

$\sum$: (a discrete time system)

$$\underline{x}(n+1) = \phi \underline{x}(n) + \psi \underline{u}(n) \quad \dots (4.3)$$

$$\underline{y}(n) = C \underline{x}(n) + D \underline{u}(n) \quad \dots (4.4)$$

where:

A, ϕ = (mxm) state matrix, or state transition matrix

B, ψ = (mxs) input matrix

C = (rxm) output matrix

D = (rxs) direct matrix

m is the number of states (system order)

s is the number of inputs

r is the number of outputs

The estimator equations are [5];

For $\hat{\underline{S}}$:

$$\dot{\hat{\underline{x}}}(t) = A \hat{\underline{x}}(t) + B \underline{u}(t) + M \left[\underline{y}(t) - C \hat{\underline{x}}(t) - D \underline{u}(t) \right] \quad \dots (4.5)$$

For $\hat{\underline{\Sigma}}$:

$$\hat{\underline{x}}(n+1) = \phi \hat{\underline{x}}(n) + \psi \underline{u}(n) + M \left[\underline{y}(n) - C \hat{\underline{x}}(n) - D \underline{u}(n) \right] \quad \dots (4.6)$$

$$\hat{\underline{y}}(n) = C \hat{\underline{x}}(n) + D \underline{u}(n) \quad \dots (4.7)$$

where $M(m \times r)$ is an estimator gain matrix chosen to drive the estimator states $\hat{\underline{x}}(n)$ into coincidence with the system states $\underline{x}(n)$.

Because the computer is of discrete time nature, the second form of the estimator (Eqns. 4.6 & 4.7) is considered through out this thesis for discrete systems represented by Eqns. 4.3 & 4.4.

The continuous time form represented by Equations 4.1 & 4.2 are used for the simulation on the VAX computer (using ISIS continuous/discrete time computer simulation packages) to compare the actual and the estimated states.

Considering the discrete case, and defining the error in the states as:

$$\tilde{\underline{x}}(n) = \underline{x}(n) - \hat{\underline{x}}(n) \quad \dots (4.8)$$

then the estimator is said to be an Asymptotic State Estimator if;

$$\lim_{n \rightarrow \infty} \tilde{\underline{x}}(n) \rightarrow 0 \quad \dots (4.9)$$

where $\underline{u}(n)$ is constant.

4.2.1 CHOICE OF ESTIMATOR GAIN MATRIX M :

Consider the behaviour of the error $\tilde{\underline{x}}$;

$$\begin{aligned} \tilde{\underline{x}}(n) &= \underline{x}(n) - \hat{\underline{x}}(n) \\ &= \phi \underline{x}(n) + \psi \underline{u}(n) - \{ \phi \hat{\underline{x}}(n) - \psi \underline{u}(n) + M(C \hat{\underline{x}}(n) + D \underline{u}(n)) - (C \hat{\underline{x}}(n) + D \underline{u}(n)) \} \\ &= (\phi - MC) \hat{\underline{x}}(n) \quad \dots (4.10) \end{aligned}$$

Equation (4.10) is asymptotically stable if and only if the eigen values of $(\phi - MC)$ are inside the unit circle of the z-plane.

In order for M to be able to affect all the eigen values of $(\phi - MC)$, (ϕ, C) must be an observable pair, that is;

$$\text{rank } [C^T \ \phi^T C^T \ \phi^{2T} C^T \ \dots \ \phi^{(m-1)T} C^T] = m \quad \dots (4.11)$$

The estimator gain M may be designed to ensure that the eigen values of $(\phi - MC)$ correspond to a rapid well behaved decay of any estimator error [2]. In general, the nearer to the z-plane origin we place these eigenvalues, the faster the response, but this has the effect of corrupting the estimated states with noise if measurement noise is present, which is the case in this work.

Eigenvalues closer to the z-plane unit circle remove more measurement noise at a cost of slower estimator response, particularly to process noise. This leads to a compromise on the value of M to give an optimal solution. The larger the measurement noise variance, the closer the eigen values must be to the unit circle, while the smaller this variance the nearer the origin we can place them, and vice versa for the process noise variance. An "optimal" choice of M, given system and noise characteristics, is provided by the Kalman filter [1,2].

The Kalman Filter is a stochastic state estimator which estimates the states of an observable stochastic system in such away that it minimizes a measure of the expected error between the estimated and the actual state(s). It thus allows the adverse effects of process noise and measurement noise on state estimation to be optimally reduced provided these noises are uncorrelated, white and Gaussian, and their variances are known.

4.3 LINEAR STOCHASTIC MODELS:

Consider a linear continuous time system, and let the state process $\underline{x}(t)$ of the system model satisfy the linear stochastic equation:

$$d\underline{x}(t) = A(t)\underline{x}(t)dt + B(t)\underline{u}(t)dt + G(t)d\underline{\beta} \quad \dots (4.12a)$$

or, in the less rigorous white noise notation;

$$\dot{\underline{x}}(t) = A(t)\underline{x}(t) + B(t)\underline{u}(t) + G(t)\underline{w}(t) \quad \dots (4.12b)$$

where $\underline{x}(t)$ is an m-vector state process, one sample of which would generate a state time history. $\underline{u}(t)$ is an s-vector of control input functions. $A(t)$ is an m-by-m system dynamic matrix, $B(t)$ is an m-by-s deterministic input matrix, and $G(t)$ is an m-by-m noise input matrix. $\underline{\beta}(t)$ in (4.12a) is an m-vector Brownian motion with statistics:

$$E\{\underline{\beta}(t)\} = \emptyset$$

$$E\{[\underline{\beta}(t) - \underline{\beta}(t')][\underline{\beta}(t) - \underline{\beta}(t')]^T\} = \int_t^t Q(\tau) d\tau$$

With $Q(t)$ an m -by- m matrix of piecewise continuous functions such that $Q(t)$ is symmetric and positive semidefinite. This implies that at least some states are influenced by process noise. Thus, in (4.12b) which is the form used in this thesis, the $\underline{w}(t)$ is m -vector white Gaussian noise with statistics;

$$E\{\underline{w}(t)\} = \emptyset$$

$$E\{\underline{w}(t) \cdot \underline{w}(t')^T\} = Q(t) \delta(t - t')$$

with the same description of $Q(t)$ as just given. The state differential Equation (4.12b) is propagated forward from the initial condition $\underline{x}(t_0)$. The whole system model can be described by Equations (4.13 a&b) as follows:

$$\dot{\underline{x}}(t) = A(t)\underline{x}(t) + B(t)\underline{u}(t) + G(t)\underline{w}(t) \quad \dots (4.13a)$$

$$\underline{y}(t) = C(t)\underline{x}(t) + D(t)\underline{u}(t) + \underline{v}(t) \quad \dots (4.13b)$$

For an LTI system, the matrices A , B , C , D are constant matrices. The direct matrix D is usually zero when the output vector $\underline{y}(t)$ does not depend directly on the input vector $\underline{u}(t)$.

The measurement noise vector $\underline{v}(t)$ is an r -dimensional white Gaussian noise with statistics :

$$E\{\underline{v}(t)\} = \emptyset$$

$$E\{\underline{v}(t) \cdot \underline{v}(t')^T\} = R(t) \delta(t - t')$$

Here $R(t)$ is an r -by- r , symmetric, positive definite matrix. Positive definiteness of $R(t)$ implies that all components of the measurement vector are corrupted by noise. [1]

Throughout this thesis, as a microprocessor is used, we will use the discrete equivalent of equations (4.13) as follows:

$$\underline{x}(n+1) = \phi(n)\underline{x}(n) + \psi(n)\underline{u}(n) + G(n)\underline{w}(n) \quad \dots (4.14a)$$

$$\underline{y}(n) = C(n)\underline{x}(n) + D(n)\underline{u}(n) + \underline{v}(n) \quad \dots (4.14b)$$

Again, for (LTI) systems (which is the case dealt in this work), the matrices ϕ, ψ, G, C & D are constant.

4.4 KALMAN FILTERING FOR DISCRETE TIME SYSTEMS :

4.4.1 THE BASIC KALMAN FILTER:

We consider a discrete system $\sum$ which is completely controllable (to be used later in the optimal feedback control problem) and completely observable and has state equations (4.14).

Assuming knowledge of ϕ, ψ, G, C, D and the variances of $\underline{w}$ and $\underline{v}$, it is required to find the "best" estimate (in a mean square sense) of the present value of $\underline{x}$ given $\underline{u}$ and $\underline{y}$.

More precisely to find $\hat{\underline{x}}(n)$ where $\hat{\underline{x}}(n)$ is such that;
 $E[(\underline{x}(n) - \hat{\underline{x}}(n))^T (\underline{x}(n) - \hat{\underline{x}}(n))]$ is minimum.

The error covariance matrix P is defined as:

$$P(n) = E[(\underline{x}(n) - \hat{\underline{x}}(n))(\underline{x}(n) - \hat{\underline{x}}(n))^T] \quad \dots (4.15)$$

The problem is to find $\hat{\underline{x}}(n)$ which minimizes trace $P(n)$.

Let

$$E(\underline{w}) = E(\underline{v}) = \emptyset \quad (\underline{w} \text{ \& \; } \underline{v} \text{ have zero mean value})$$

$$E[\underline{w}(i)\underline{v}(j)^T] = \emptyset \quad (\underline{w} \text{ \& \; } \underline{v} \text{ uncorrelated with each other})$$

$$E[\underline{w}(i)\underline{w}(j)^T] = Q_d \delta_{ij} \quad (Q_d \text{ is positive semi definite})$$

$$E[\underline{v}(i)\underline{v}(j)^T] = R_d \delta_{ij} \quad (R_d \text{ is positive definite})$$

where δ is Kronecker delta function defined as:

$$\delta_{ij} = 1 \text{ for } i=j$$

$$\delta_{ij} = 0 \text{ for } i \neq j$$

The corresponding state estimator system to that used in Kalman filtering is:

$\hat{\Sigma}$:

$$\hat{\mathbf{x}}(n+1) = \phi(n)\hat{\mathbf{x}}(n) + K_f(n)\{\mathbf{y}(n) - C(n)\hat{\mathbf{x}}(n) - D(n)\mathbf{u}(n)\} + \psi(n)\mathbf{u}(n) \quad \dots (4.16)$$

$$\hat{\mathbf{y}}(n) = C(n)\hat{\mathbf{x}}(n) + D(n)\mathbf{u}(n) \quad \dots (4.17)$$

where $K_f(n)$ is the optimal Kalman filter gain matrix which is calculated from [1,2];

$$K_f(n) = \phi(n)P(n)C(n)^T[R_d(n) + C(n)P(n)C(n)^T]^{-1} \quad \dots (4.18)$$

$$P(n+1) = [\phi(n) - K_f(n)C(n)]P(n)\phi(n)^T + G(n)Q_d(n)G(n)^T \quad \dots (4.19)$$

where $P(0)$ is chosen arbitrarily.

The discrete algorithm above is suitable to the discrete nature of the digital computer. The basic relations, which are of a recursive nature, are given by Eqns. (4.18), (4.16), (4.19) in that order repeated for;

$n = 0, 1, 2, \dots, \infty$.

This algorithm has the following properties; [2]

1. The filter gain $K_f(n)$ is affected by $Q_d(n)$ and $R_d(n)$ which are basically measures of the amplitudes of noise affecting the system. The algorithm fails if $Q_d = 0$ or $R_d = 0$.
2. Provided the system Σ is LTI, the filter gain settles down after an initial transient to values of $K_f(n)$ independent of $P(0)$, the elements of which may in principle be set to any value (typically zero). In this case the constant K_f may be calculated off line and simply used in Eqn (4.16), which then becomes a state estimator (cf Eqn 4.6).

The structure of the optimal filter, and the linear system with which it is associated, is shown in Fig 4.2.

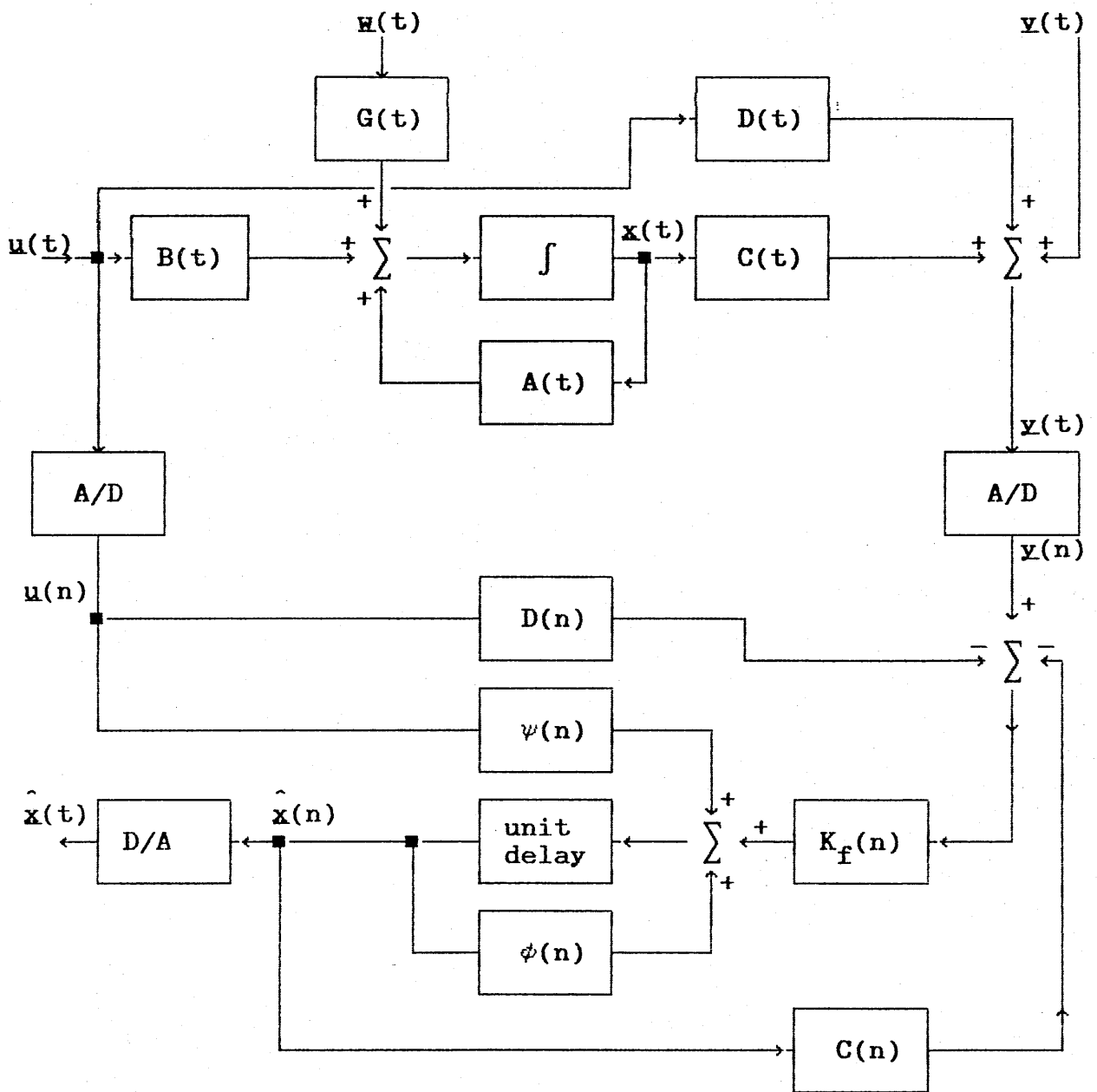


Fig. 4.2 Block Diagram of Optimal Kalman Filter.

4.4.2 THE SPLIT FORM OF KALMAN FILTER :

In a "split" form of the algorithm, which is commonly used, the optimal state estimate is propagated from time (n) to time (n+1) by the relation [1];

$$\hat{\mathbf{x}}(n+1^-) = \phi(n)\hat{\mathbf{x}}(n) + \psi(n)\mathbf{u}(n) \quad \dots (4.20)$$

$$\mathbf{P}(n+1^-) = \phi(n)\mathbf{P}(n^+)\phi(n)^T + \mathbf{G}(n)\mathbf{Q}_d(n)\mathbf{G}(n)^T \quad \dots (4.21)$$

At measurement time (n+1) when the estimate $\hat{\mathbf{x}}(n+1^-)$ is available, the measurement $\mathbf{y}(n+1)$ becomes available. The estimate is updated to $\hat{\mathbf{x}}(n+1^+)$ by defining the Kalman filter gain matrix $\mathbf{K}_f(n+1)$ as:

$$\mathbf{K}_f(n+1) = \mathbf{P}(n+1^-)\mathbf{C}(n)^T \left[\mathbf{R}_d(n) + \mathbf{C}(n)\mathbf{P}(n+1^-)\mathbf{C}(n)^T \right]^{-1} \quad \dots (4.22)$$

$$\hat{\mathbf{x}}(n+1^+) = \hat{\mathbf{x}}(n+1^-) + \mathbf{K}_f(n+1) \left[\mathbf{y}(n+1) - \mathbf{C}(n)\hat{\mathbf{x}}(n+1^-) - \mathbf{D}(n)\mathbf{u}(n+1) \right] \quad (4.23)$$

$$\mathbf{P}(n+1^+) = \mathbf{P}(n+1^-) - \mathbf{K}_f(n+1)\mathbf{C}(n)\mathbf{P}(n+1^-) \quad \dots (4.24)$$

The initial conditions for the recursion are $\hat{\mathbf{x}}(0)^+$, $\mathbf{P}(0)^+$, usually chosen arbitrarily.

Eqns (4.22), (4.23), (4.24) are usually called measurement update sequence, realized immediately after a sampling instant, while Eqns (4.20), (4.21) are the time update sequence, realized at any time before the next sampling instant.

4.5 COVARIANCE FACTORISATION FILTER :

It has been found [1,8] that on-line (optimal) Kalman filtering presents several problems, for example:

- a- Significant computational burden because of a high data rate.
- b- Round-off errors inherent in any recursive algorithm.
- c- Ill-conditioned problems may result if the covariance matrix $\mathbf{P}(n)$ becomes non-positive semi-definite. This can occur due to numerical computation errors and can lead to filter divergence.

In particular, the matrix inversion of Eqn (4.18) or Eqn. (4.22) can cause these problems.

In order to resolve these problems, more attractive algorithms have been developed which do the same task as the conventional Kalman filter. Two method is based on "factorising" the covariance matrix [1]. These are the "Covariance square root filter", and the "UD method". Both are developed from the split Kalman filter algorithm of Sect 4.4.2. In the first case the covariance matrix $P(n)$ is factorised into its "roots" ($P = \sqrt{P} \cdot \sqrt{P}^T$). This method was developed by "Potter" in 1964 for space navigation applications [1], using the Cholesky lower or upper triangular square root method for factorising the covariance matrix. In the second method, the covariance matrix $P(n)$ is factorised, again using a Cholesky decomposition method, into a unit upper triangular matrix U and a diagonal matrix D , so that $P = U \cdot D \cdot U^T$ [1].

In this work the UD covariance factorising method is proposed, and this is described in Section 4.5.1.

4.5.1 UD METHOD OF KALMAN FILTERING :

Referring to Eqns (4.21) and (4.24), this method expresses the covariance before and after measurement incorporation as;

$$P(n^-) = U(n^-)D(n^-)U^T(n^-) \quad \dots (4.25)$$

$$P(n^+) = U(n^+)D(n^+)U^T(n^+) \quad \dots (4.26)$$

Where the U matrices are upper triangular and unity (with ones along the diagonal) and the D matrices are diagonal.

Considering a given P as m -by- m symmetric, positive semi-definite matrix, a unit upper triangular factor U and diagonal factor D such that $P = U \cdot D \cdot U^T$ can always be generated. Although such U and D matrices are not unique, one such defined pair can be generated through an algorithm closely related to the backward running Cholesky decomposition algorithm [1].

First for the m 'th column :

$$D_{mm} = P_{mm}$$

$$U_{im} = \begin{cases} 1 & i=m \\ P_{im}/D_{mm} & i=m-1, m-2, \dots, 1 \end{cases} \quad \dots (4.27)a$$

Then for the remaining columns, for $j = m-1, m-2, \dots, 1$

$$D_{jj} = P_{jj} - \sum_{k=j+1}^m D_{kk} \cdot U_{jk}^2$$

$$U_{ij} = \begin{cases} 0 & i > j \\ 1 & i = j \\ [P_{ij} - \sum_{k=j+1}^m D_{kk} \cdot U_{ik} \cdot U_{jk}] / D_{jj} & i=j-1, j-2, \dots, 1 \end{cases} \quad \dots (4.27)b$$

This generates the U-D factors of $P(0)$, and the Kalman filter algorithm can be started. It proceeds as follows.

(a) The Measurement Update Sequence:

The scalar measurement update for the U-D covariance factorisation filter can now be specified. At sampling n , $U(n^-)$ and $D(n^-)$ are available from a previous time propagation (to be discussed). Using the measurement value $y(n)$ and the known 1-by- m $C(n)$, the following can be calculated:[1]

$$\underline{f} = U^T(n^-) \underline{C}^T(n)$$

$$V_j = D_{jj}(n^-) f_j \quad j=1, 2, \dots, m$$

$$a_o = r_{ii} \quad i=1, 2, \dots, r \quad \dots (4.28)$$

where r_{ii} is the i 'th element in the measurement noise covariance matrix R_d , which is assumed to be diagonal.

Then for $k = 1, 2, \dots, m$

$$a_k = a_{k-1} + f_k v_k$$

$$D_{kk}(n^+) = D_{kk}(n^-) a_{k-1}/a_k$$

$$b_k \leftarrow v_k$$

$$p_k = -f_k/a_{k-1}$$

$$\left. \begin{aligned} U_{jk}(n^+) &= U_{jk}(n^-) + b_j p_k \\ b_j &\leftarrow b_j + U_{jk}(n^-) v_k \end{aligned} \right\} \quad j = 1, 2, \dots, (k-1)$$

... (4.29)

In Eqn. (4.29) $\leftarrow$ denotes replacement.

After m iterations (Eqn(4.29)), $U(n^+)$ and $D(n^+)$ have been computed, and the filter gain $\bar{K}_f(n)$ is calculated in terms of the m -vector b made up of components $b_1, b_2, \dots, b_m$ computed in the last iteration of (4.29). The state updated as;

$$\bar{K}_f(n) = b / a_m$$

$$\underline{x}(n^+) = \underline{x}(n^-) + \underline{K}(n)[\underline{y}(n) - \underline{C}(n).\underline{x}(n^-)] \quad \dots (4.30)$$

Vector measurement updates are performed component by component "r" times to complete the whole sequence at each iteration. A transformation of variables [1], if R_d is not originally diagonal, is required, but in this work R_d is assumed diagonal which is usually the case.

(b) The Time Update Sequence:

The time propagation of the U-D factors employs a generalized Gram-Schmidt orthogonalization to preserve numerical accuracy while attaining high computation efficiency [3].

Given the covariance time propagation relation;

$$P(n+1^-) = \phi(n+1, n).P(n^+).\phi^T(n+1, n) + G(n).Q_d(n).G^T(n) \quad (4.31)$$

and the U-D factors of $P(n^+)$, we desire the factors $U(n+1^-)$ and $D(n+1^-)$ such that $[U(n+1^-).D(n+1^-).U^T(n+1^-)]$ equals the right hand side of Eqn (4.26). Q_d is assumed diagonal, since, given the m-by-m matrix $[G(n).Q_d(n).G^T(n)]$, (4.27) can be used to generate $G(n)$ as its U-factors and $Q_d(n)$ as its D-factors.

An m-by-(m+s) matrix $\bar{Y}(n+1^-)$ and an (m+s)-by-(m+s) diagonal matrix $\bar{D}(n+1^-)$ are defined as

$$\bar{Y}(n+1^-) = [\phi(n+1,n).U(n^+) \ ; \ G(n)] \quad \dots (4.32a)$$

$$\bar{D}(n+1^-) = \begin{bmatrix} D(n^+) & ; & \emptyset \\ \emptyset & ; & Q_d(n) \end{bmatrix} \quad \dots (4.32b)$$

The actual computational algorithm is the modified weighted Gram Schmidt (MWGS)[1] method. Thus the time propagation relations are to compute $\bar{Y}(n+1^-)$ and $\bar{D}(n+1^-)$ as in Eqn.(4.32), and initialize m-vectors, each of dimension (m+s), through

$$\begin{bmatrix} a_1 & a_2 & ; & \dots & ; & a_m \end{bmatrix} = \bar{Y}(n+1^-)$$

and then to iterate on the following relations for $k=m, m-1, \dots, 1$

$$c_k = \bar{D}(n+1^-).a_k$$

$$D_{kk}(n+1^-) = a_k.c_k$$

$$d_k = c_k / D_{kk}(n+1^-)$$

$$\begin{cases} U_{jk}(n+1^-) = a_j^T d_k \\ a_j \leftarrow a_j - U_{jk}(n+1^-).a_k \end{cases} \quad j = 1, 2, \dots, k-1 \quad \dots (4.33)$$

The time update state estimate is then given by;

$$\hat{x}(n+1^-) = \phi(n).\hat{x}(n^+) + \psi(n)u(n) \quad \dots (4.34)$$

4.6 FACTORS INFLUENCING THE CHOICE OF A SUITABLE MICROPROCESSOR:

To implement the Kalman filter algorithm in a microprocessor, the following points should be considered [7]:

1. Input/Output:

Since the Kalman filter is a real-time algorithm often used in direct control applications, the microprocessor must be able to communicate effectively with the system. The transfer of data must be sufficiently fast with respect to the speed of computation to ensure that input and output do not take a significant portion of the iteration time.

2. Calculations:

A relatively high speed of operation is necessary. In order to secure this, hardware multiplication and division facilities are advantageous, since performing considerable number of multiplications and particularly divisions can limit the Kalman filter iteration rate.

3. Memory:

The microprocessor should be able to communicate with memory in a versatile way to enable data to be processed rapidly without the need for much swapping in and out of workspace or accumulator.

These points can be concluded as follows;

1. The microprocessor software must be written in a form which suits the real-time application.
2. To achieve fast algorithm execution, maths co-processor could be used to do the mathematical operations.
3. A suitable choice of the microprocessor (CPU) and the peripherals (converters, decoders, ...etc) must be made.

In Chapter 5 we consider these in more detail.

REFERENCES :

- [1] MAYBECK, P S. "Stochastic Models, Estimation, and Control", Vol's. 1,2&3, Academic Press 1979
- [2] BORRIE, J. A. "Modern Control Systems", Prentice/Hall International, 1986.
- [3] Schmidt, S. F. "Computational Technique in Kalman Filtering, Theory and Application of Kalman Filtering", AGARD 139, Chapter-3, London 1970.
- [4] Yazdanian, M. "Implementation of a Kalman Filter using Z8000 Microprocessor", MSc thesis, Sept. 1982 SESD, CIT.
- [5] Chi-Tsong Chen, "Linear System Theory and Design", Holt-Saunders International Edition, 1984.
- [6] Bierman, G. J. " A Comparison of Discrete Linear Algorithms", IEEE Trans. of Aerospace and Electronic Systems, Vol. AES-9, No.1, January 1973.
- [7] Page, M. "Microprocessor Implementation of the Kalman Filter", Microelectronics Journal 1979.
- [8] Yassine Saeed Al-Masri, "Adaptive State Estimation Compared With Proportional Navigation for Bearing Only Missile Guidance", Phd thesis, CIT August 1986.

CHAPTER 5

8086-MICROPROCESSOR HARDWARE AND SOFTWARE

5.1 INTRODUCTION :

An electronic circuit based on an 8086 Intel[®] 16-bit microprocessor was designed to run the Kalman filter algorithms described in Chapter 4, and the control strategies in Chapter 7.

The circuit is on two boards (and an auxiliary board) and consists of an 8086 microprocessor, an A/D converter with 8 multiplexed inputs, D/A converter with 8 multiplexed outputs, a key board, a display and associated support hardware. A schematic is shown in Fig. (5.1).

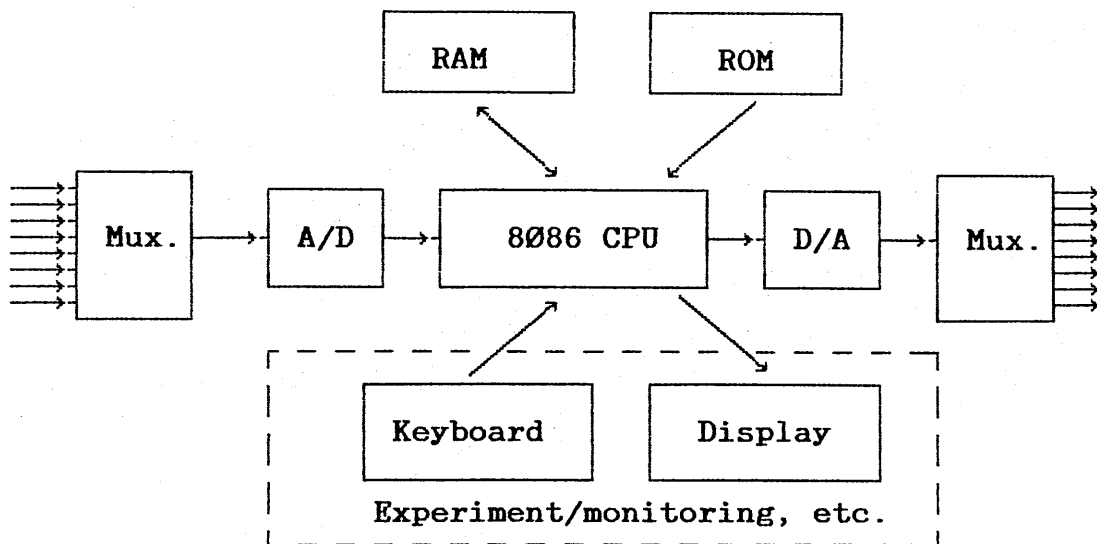


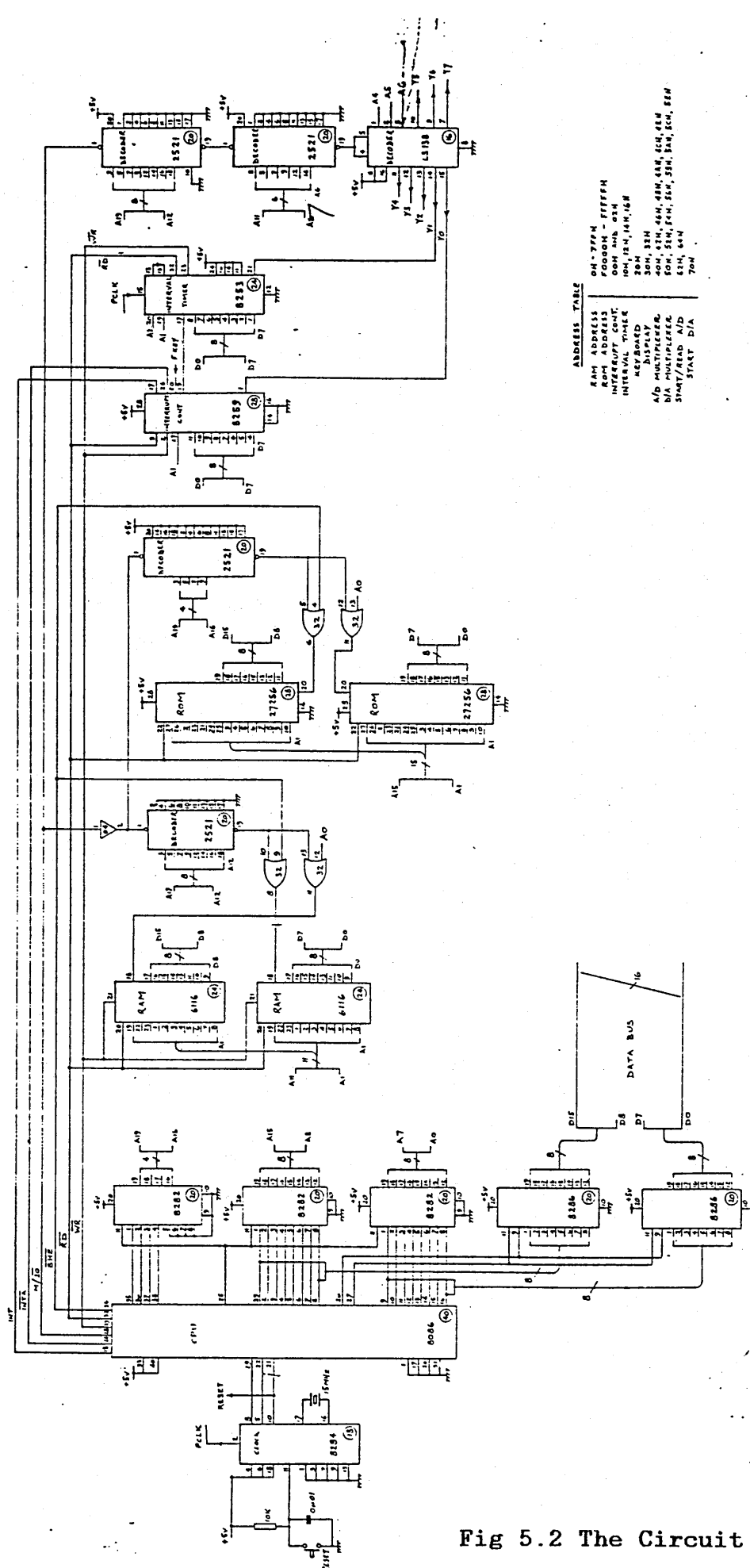
Fig. 5.1 Schematic of Equipment

The components are mounted on three boards which can be described briefly as follows:

- Board 1; includes the microprocessor, memory (4K Bytes RAMs and 64 K Bytes EPROM), clock generator, Programmable Counter Timer (PCT), Programmable Interrupt Controller (PIC), and a number of digital components (decoders, gates,...etc.).
- Board 2; includes all the analogue components which are treated as peripherals of the microprocessor (A/D & D/A converters, multiplexers, operational amplifiers,...etc.).
- Board 3; includes a 2x20-character DOT-MATRIX LCD-display, and 4x4 keypad with the controlling chips. This board is for experimental and monitoring purposes.

A stabilized power supply unit is used to supply +5 volts for the digital board, and +15, -15 volts for the analogue board.

Fig. 5.2 shows the circuit diagram of Board 1 and Fig.5.3 shows the circuit diagram of Boards 2 and 3. Parts list of the components used are shown in Appendix-B. The boards are considered in some detail in the following sections.



ADDRESS TABLE

ADDRESS	DESCRIPTION
0000H - 0FFFH	RAM ADDRESS
1000H - 1FFFH	ROM ADDRESS
2000H - 2FFFH	INTERF. CONT.
3000H - 3FFFH	INT. TIME
4000H - 4FFFH	KEYBOARD
5000H - 5FFFH	A/D CONVERTER
6000H - 6FFFH	D/A CONVERTER
7000H - 7FFFH	START/READ A/D
8000H - 8FFFH	START/READ D/A

Fig 5.2 The Circuit Diagram of Board 1.

THIRD ANGLE PROJECTION				TITLE: CPU BOARD			
ITEM	PART NO.	DESCRIPTION	QTY	MATL.	SPEC.	REMARKS	
GENERAL TOLERANCE ON DIMENSIONS							
MACHINED							
UNMACHINED							
OTHER DIMENSIONS AS STATED							
WELD WIRE SPACING THIS SIDE							
MACHINE WIRE BOWING THIS SIDE							
USED ON ONE							
CRANFIELD INSTITUTE OF TECHNOLOGY							
CRANFIELD							
DRAWING NO. 1							
SHEET 1 OF 1							

5.2 THE 8086 CPU BOARD (BOARD 1) FIG 5.2:

This board holds the digital devices and the associated circuitry.

5.2.1 CLOCK GENERATOR:

The clock generator circuit is an Intel[®] 8284 clock generator. This incorporates a crystal oscillator which operates at 12 MHz.

The clock generator divides the crystal frequency by three to produce a 4 MHz (MOS level) signal required by the CPU. Additionally, the clock generator performs a further divide by two output called PCLK (peripheral clock) which is primary TTL level clock signal used by the remainder of the circuits.

5.2.2 THE MICROPROCESSOR (CPU) :

The heart of the board is the 8086 microprocessor chip. This chip has 40 pins where 20 of them are used as an Address Bus of which 16 are multiplexed with a 16 bit Data Bus. The Bus decoders 8282 and 8286 are used to de-multiplex these. In this thesis, because we are using one (main) processor and one system Bus, we used the minimum mode configuration where the $\overline{MN}/\overline{MX}$ (8086 chip pin 33) input is held high.

The 8086 Microprocessor is one of the 16 bit Microprocessors family introduced by the Intel[®] company. The device can address up to 1 M-bytes of memory and/or Input/Output ports (using the first addressable 64 K byte).

The CPU has a powerful array of 16 bit registers allowing a versatile order code and rapid operation.

5.2.3 SYSTEM BUSES :

To service the system Buses, i.e. the 20 bit Address Bus and 16 bit Data Bus, the 8282 and 8286 chips are used.

The 8282 8-bit INPUT/OUTPUT latches are unidirectional 8-bit latch buffers with three-state outputs; when not selected their outputs are floated.

The 8286 8 bit Bi-directional Bus Transceivers are used to buffer bi-directional lines on a system Bus. When not selected, their inputs and outputs are floated.

5.2.4 RAM^S & EPROM^S MEMORY CHIPS:

These are;

- Two static RAMs (6116), each chip contains 2 K-byte of Random Access Memory (access time 150ns maximum).
- Two EPROMs (ultra-violet-light Erasable Programmable Read Only Memory "27C256Q") of 32 K-byte per chip (access time 200ns maximum).
- Memory Address Decoders (2521) for accessing the RAMs and EPROMs, via the address and data buses.

Since the 8086 CPU requires 16-bit word memory access, RAM and EPROM devices are used in pairs. One device is connected to the low-order Data (D0-D7) highway, and the other device is connected to the high-order Data highway (D8-D15). A decoding circuit uses $\overline{BHE}$ (from the CPU) and A0 (from the Address Bus) for memory block selection as shown in Table 1.

Decode input		Byte Accessed
A0	$\overline{\text{BHE}}$	
0	0	Both Bytes (D0-D15)
1	0	High Byte (D8 - D15)
0	1	Low Byte (D0 - D7)
1	1	None

Table 1 Memory Byte Decoding

The microprocessor internal sequence sorts out the details of the addressing operations.

5.2.5 I/O DECODE :

The peripherals (i.e. the I/O ports) have even addresses only. The $\text{M}/\overline{\text{IO}}$ CPU signal indicates whether a memory location or an Input/Output port is to be serviced. The decoding circuitry for the peripheral devices consists of two 8 bit decoders (2521), using (A7-A19) on the Address Bus, and (LS138) which uses the lower bits (A4,A5,A6).

All the peripherals (i.e. A/D & D/A converters, multiplexers, keyboard, display,...) are addressed within the lower 64 K bytes.

5.2.6 INTERRUPT CONTROLLER AND INTERVAL TIMER:

The CPU is interrupted by an interrupt controller chip (8259A) which processes interrupts from the relevant peripherals, in this case the keyboard, and the interval timer. The interval timer chip (8253) can be set by software to generates interrupts at any intervals required.

Table 2 shows the peripheral devices and their addresses as used in this project.

Enabled LS138 output	Device selected	Lower Address Bus							selected port hex code
		A6	A5	A4	A3	A2	A1	A0	
Y ₀	8259A	0	0	0	x	x	0	0	A00 = 00H (OUT) A01 = 02H (OUT)
Y ₁	8253	0	0	1	x	0	0	0	counter#0 = 10H (OUT) counter#1 = 12H (OUT) counter#2 = 14H (OUT) controlWord= 16H (OUT)
Y ₂	key- Board	0	1	0	0	0	0	0	= 20H (IN)
Y ₃	Display	0	1	1	0	0	0	0	= 30H Instr. code (IN) = 32H Char. code (IN)
Y ₄	A/D mux	1	0	0	0	0	0	0	CH#0 = 40H (IN) CH#1 = 42H (IN) CH#2 = 44H (IN) CH#3 = 46H (IN) CH#4 = 48H (IN) CH#5 = 4AH (IN) CH#6 = 4CH (IN) CH#7 = 4EH (IN)
Y ₅	D/A mux	1	0	1	0	0	0	0	CH#0 = 50H (OUT) CH#1 = 52H (OUT) CH#2 = 54H (OUT) CH#3 = 56H (OUT) CH#4 = 58H (OUT) CH#5 = 5AH (OUT) CH#6 = 5CH (OUT) CH#7 = 5EH (OUT)
Y ₆	A/D	1	1	0	x	0	1	0	READ-AD = 62H (IN) START = 64H (IN)
Y ₇	D/A	1	1	1	x	x	x	0	START = 70H (OUT)

Table 2 I/O Ports Addresses

5.3 THE ANALOGUE BOARD (BOARD 2) FIG 5.3:

This board carries the analogue devices. These are :

- ▶ An A/D Reference voltage generator (584).
- ▶ A 10 bit analogue to digital converter (AD 7570).
- ▶ A 10 bit digital to analogue converter (AD 7522).
- ▶ Two 8-way multiplexers (DG 508).
- ▶ Various operational amplifiers (311 and 741). They work with the D/A and A/D converters and as buffers on 8 input and 8 output analogue channels.
- ▶ All required passive components (resistors, capacitors, diodes,...).

Fig. 5.3 shows the circuit diagram. The parts list is shown in Appendix B.

5.3.1 THE A/D AND D/A CONVERTERS :

The sampling of the analogue input signals and binary coding of the result is done by the AD7570 Analogue to Digital Converter and its associated circuitry. The Digital to Analogue Converter AD7522 has the opposite job, it converts the binary numbers calculated by the microprocessor to analogue voltages.

The input(s) and output(s) to and from the plant under control can make up a total of up to 8 channels of analogue inputs, and 8 channels of analogue outputs. A quadruple latch (7475), circuit is used to maintain a selected multiplexer channel as indicated in Fig 5.4.

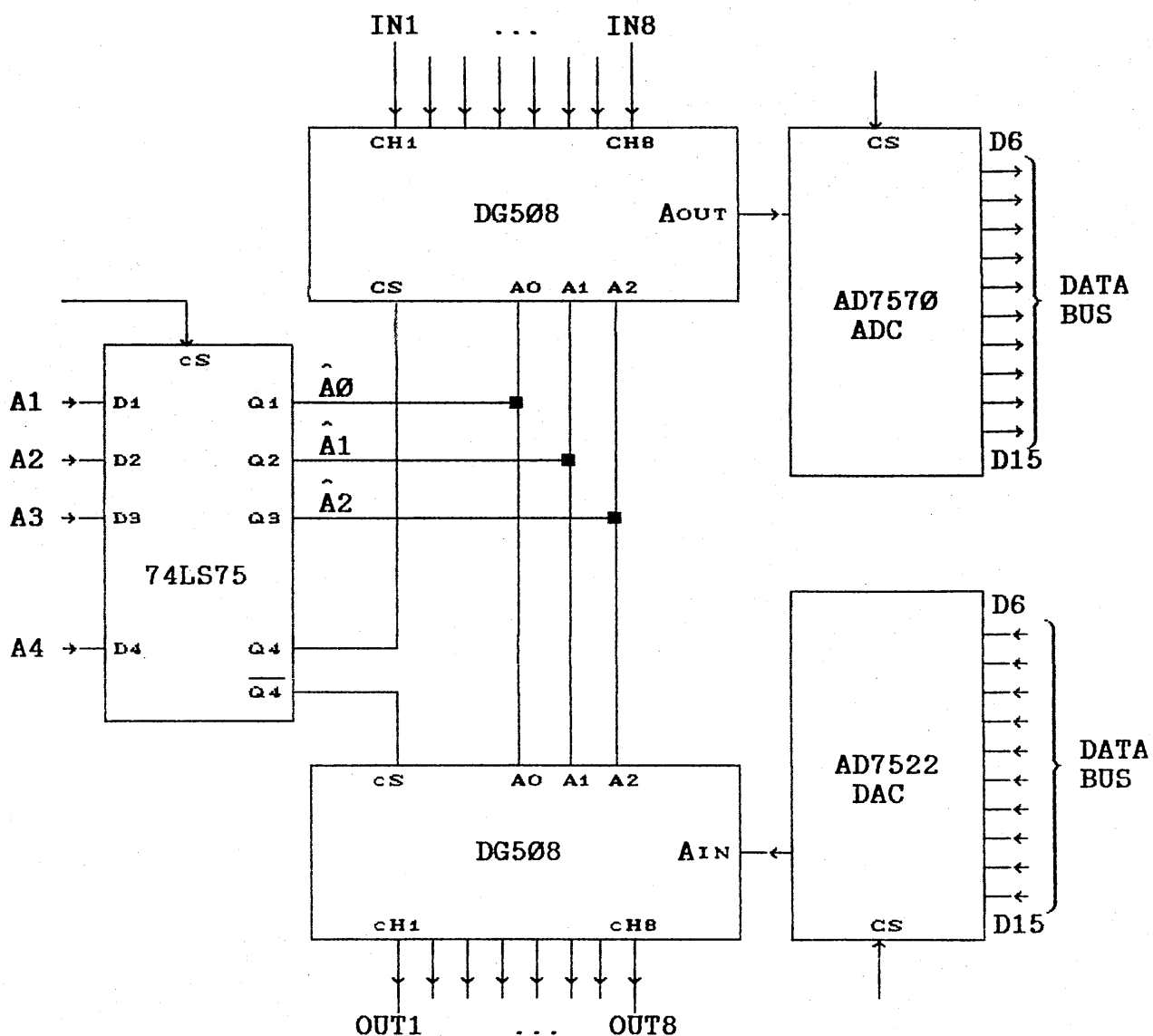


Fig. 5.4 Multiplexers Converters Circuit

The Analogue to Digital Converter AD7570 and the Digital to Analogue Converter AD7522 are 10-bit monolithic devices. A 10-bit data in Bipolar mode of operation gives a resolution of 20 mv using $V_{ref} = 10V$.

The "AD7570" A/D uses successive approximation technique for conversion at a clock frequency of 550 KHz (using external RC circuit, 39K, 100pF, as shown in Fig 5.3).

5.4 THE KEYBOARD AND DISPLAY BOARD (BOARD 3) FIG 5.3:

This board contains;

- ▶ A 4x4 key-board (labelled 0 to 9 and A to F) and its associated chip (74C992), together with a 3 state buffer (74LS244).

When a key is pressed, an interrupt is generated, and a 4 bit binary code, corresponding to the key pressed, is applied to the Data Bus when an IN statement is executed by the CPU.

- ▶ A 2x20 character (5x7 dot matrix) LCD display and its driving chip (74LS74).

The display used in this project is the Dot Matrix LCD "EPSON" EX-20027AR, which has high-contrast and wide-viewing angle, and has a "Character Generator" and "Data RAM" on board.

This board is intended for experimentation and development work and will not appear in the final system.

5.5 THE SOFTWARE OPERATING SYSTEM :

Each time the supply power is switched ON the operating system will :-

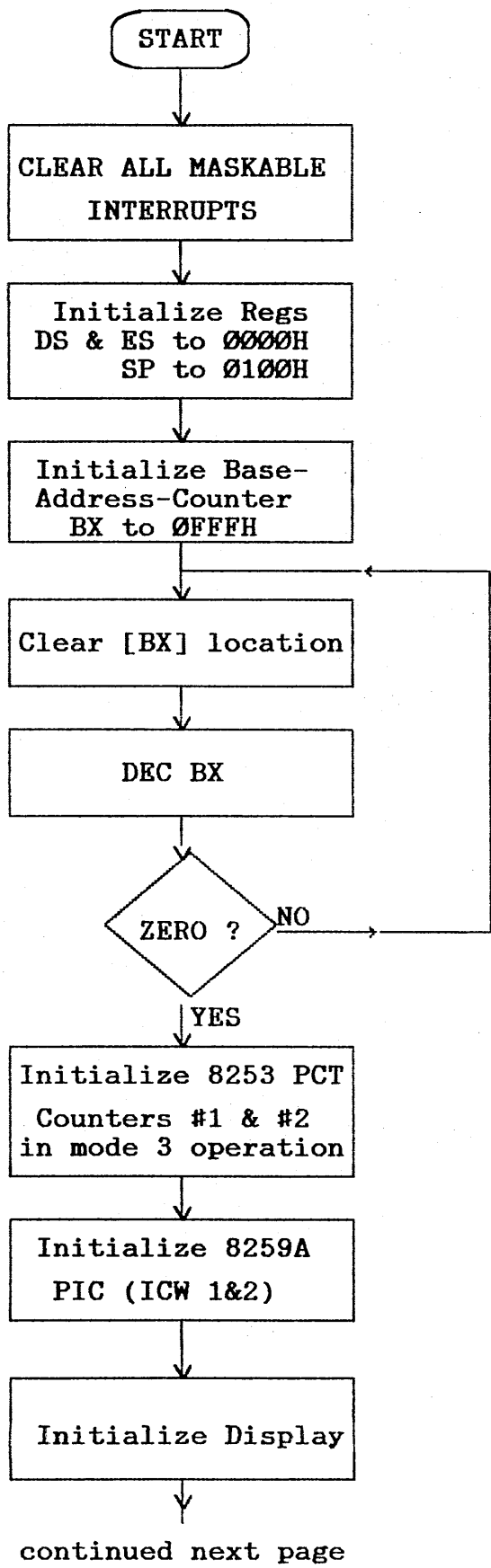
- Disable all maskable interrupts.
- Clear RAM.
- Initialize the 8253 counter/timer.
- Initialize the 8259A interrupt controller.
- Enable interrupts.
- Loop and wait for interrupts.

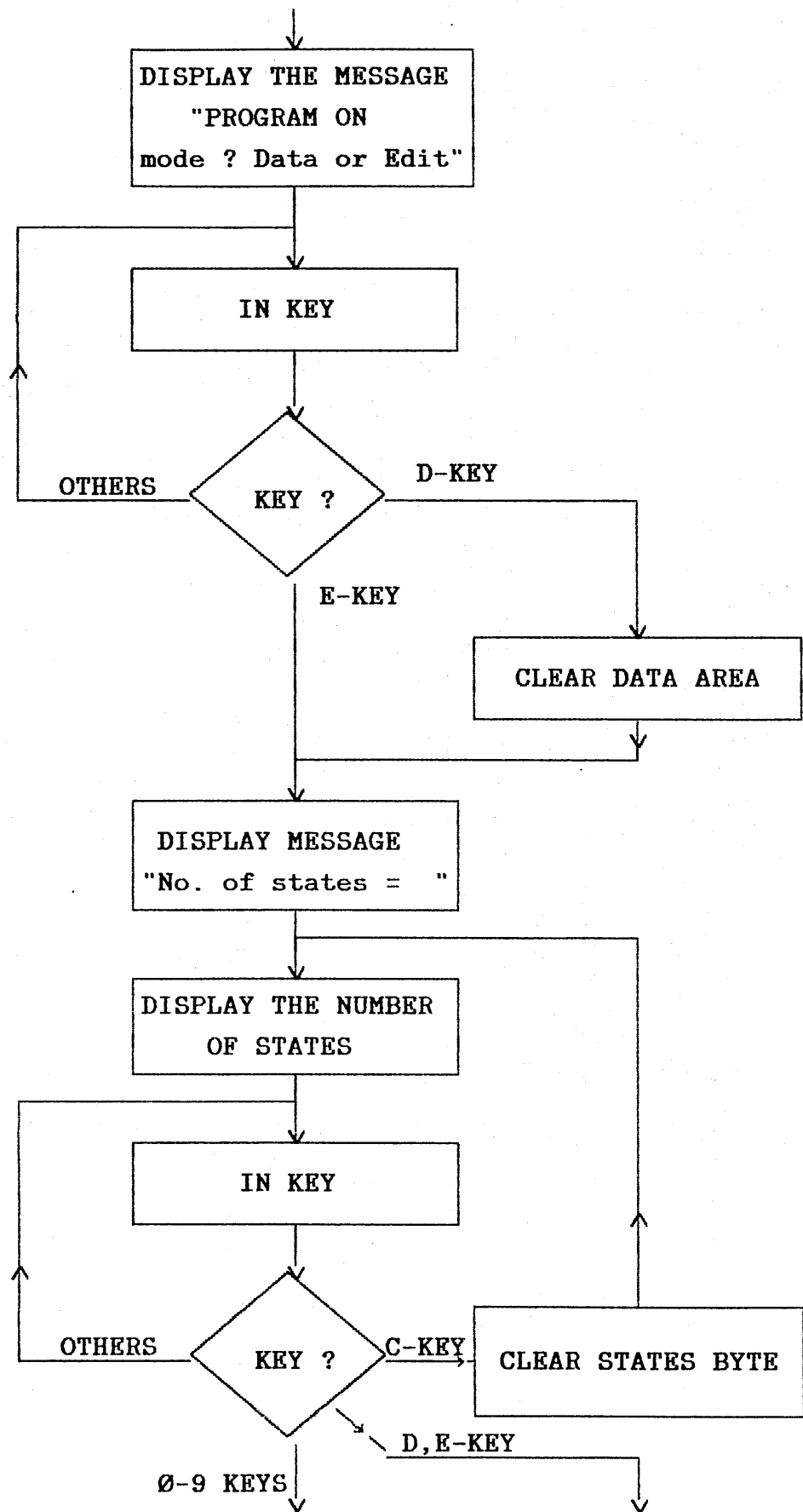
In the experimental phase of the equipment, the routine will also;

- Reset the display.
- Display a starting message, requesting data.

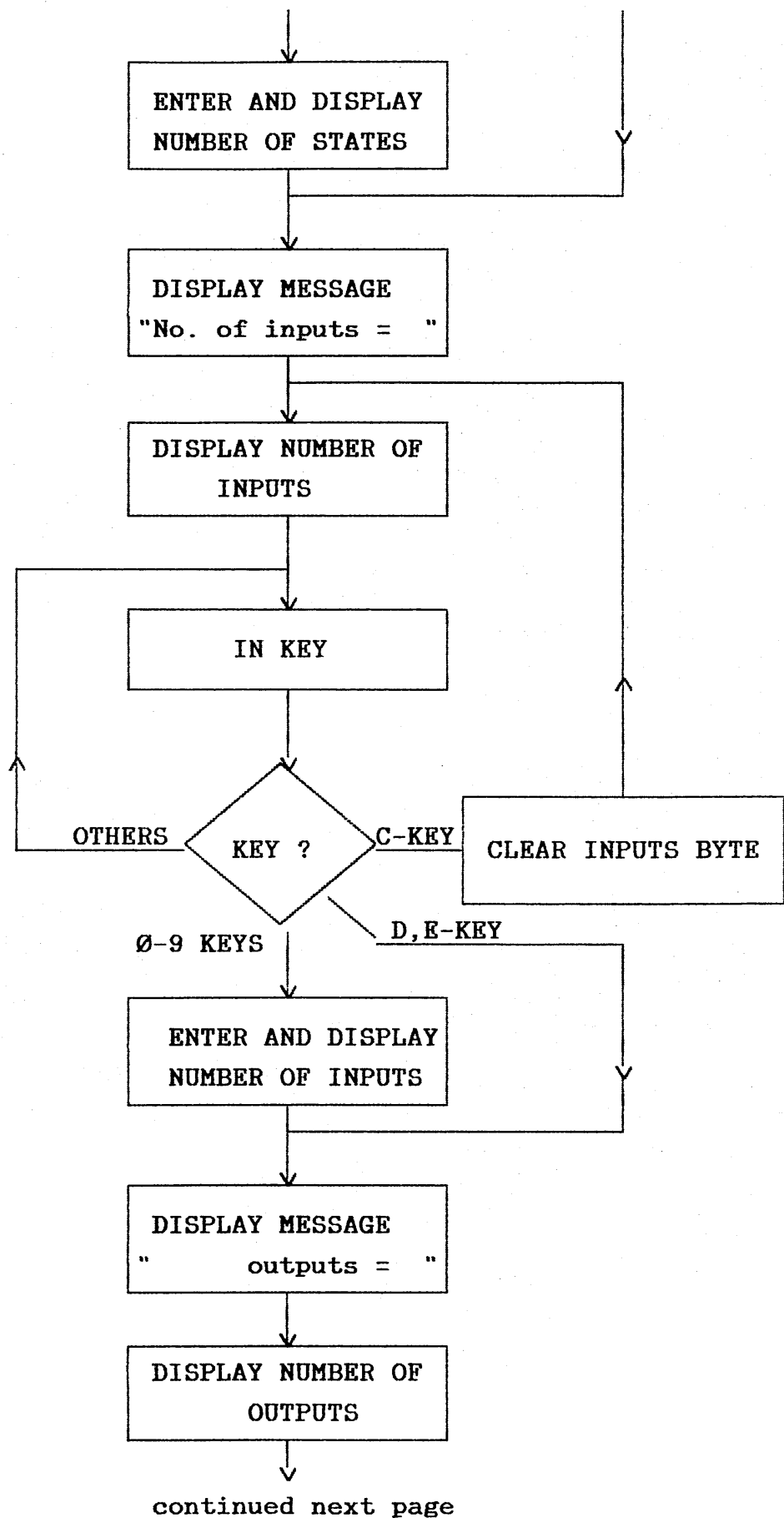
This sequence is illustrated in Fig 5.5.

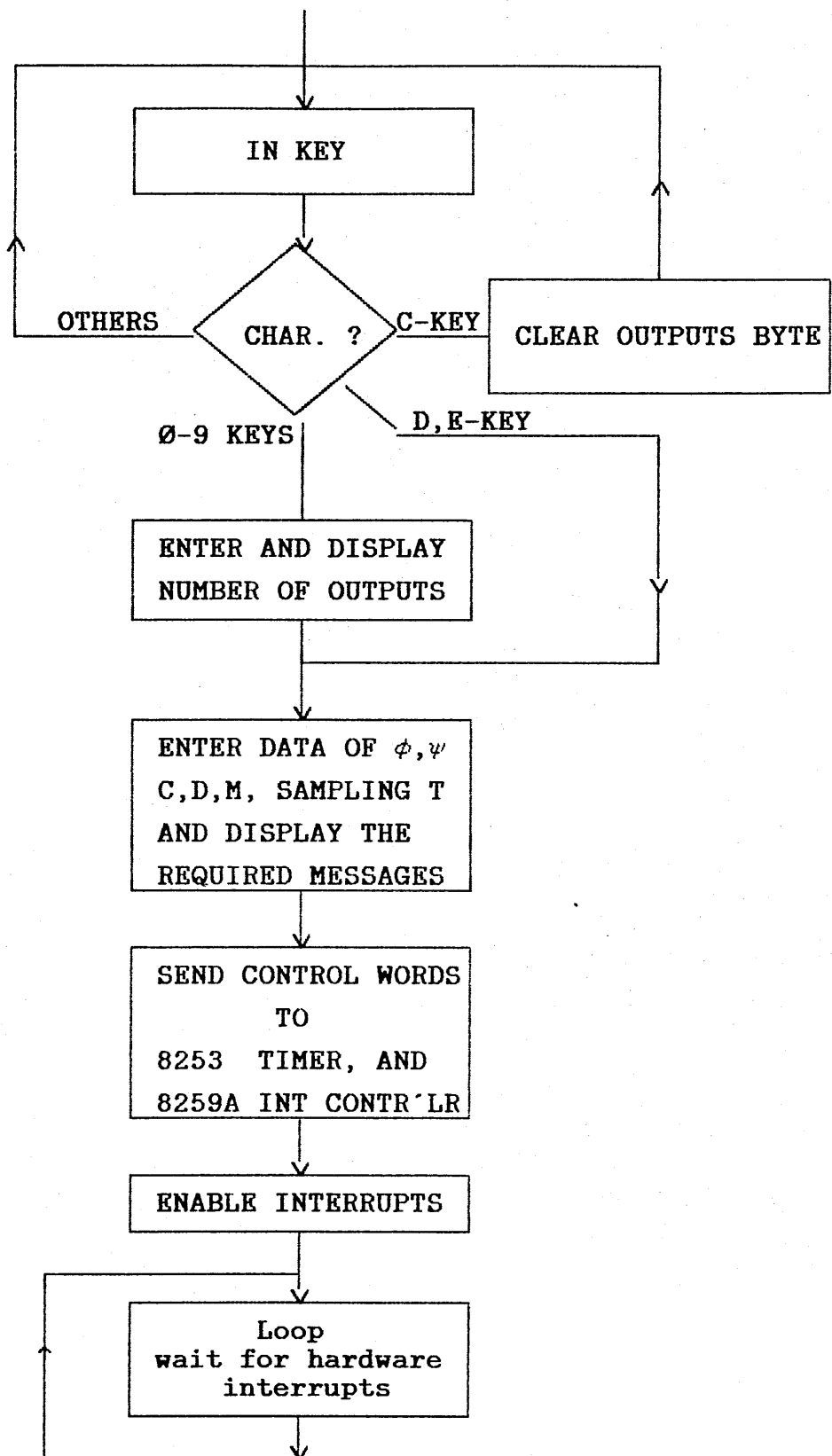
Fig 5.5 8086 System Initialization:



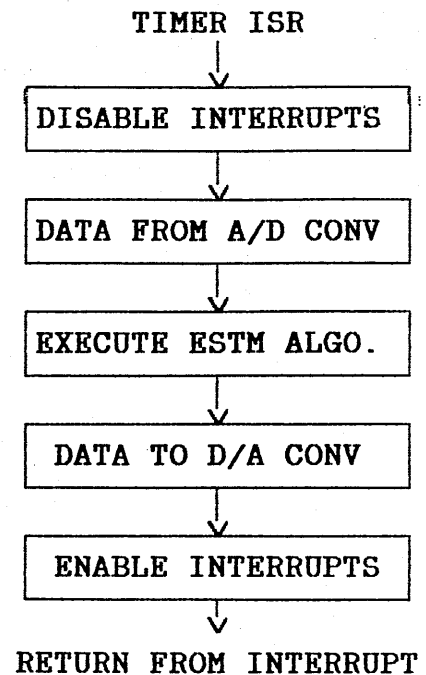
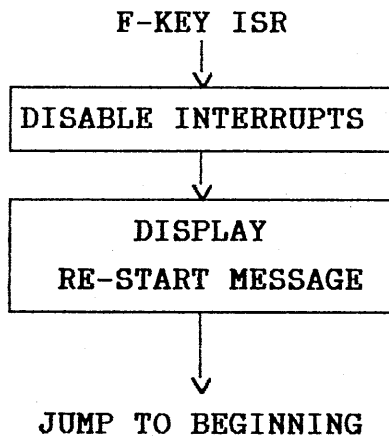


continued next page





The Interrupt Service Routines:



5.5.1 DATA INPUT :

The data required by the program are as follows :-

- ▶ NUMBER OF STATES.
- ▶ NUMBER OF INPUTS.
- ▶ NUMBER OF OUTPUTS.
- ▶ MATRIX ϕ .
- ▶ MATRIX ψ .
- ▶ MATRIX C.
- ▶ MATRIX D.
- ▶ MATRIX M.
- ▶ SAMPLING TIME.

The user is prompted to specify the number of states, number of inputs and number of outputs of the system in use before entering the elements of the matrices.

The data describing the various matrices required by the estimator are entered via the keyboard. They are stored as arrays of (16-bit "word") binary numbers and addressed by pointers. The first *WORD* of each array at the pointer represents the matrix size (dimension), i.e. number of rows "m" and number of columns "n", as shown in Fig 5.6.

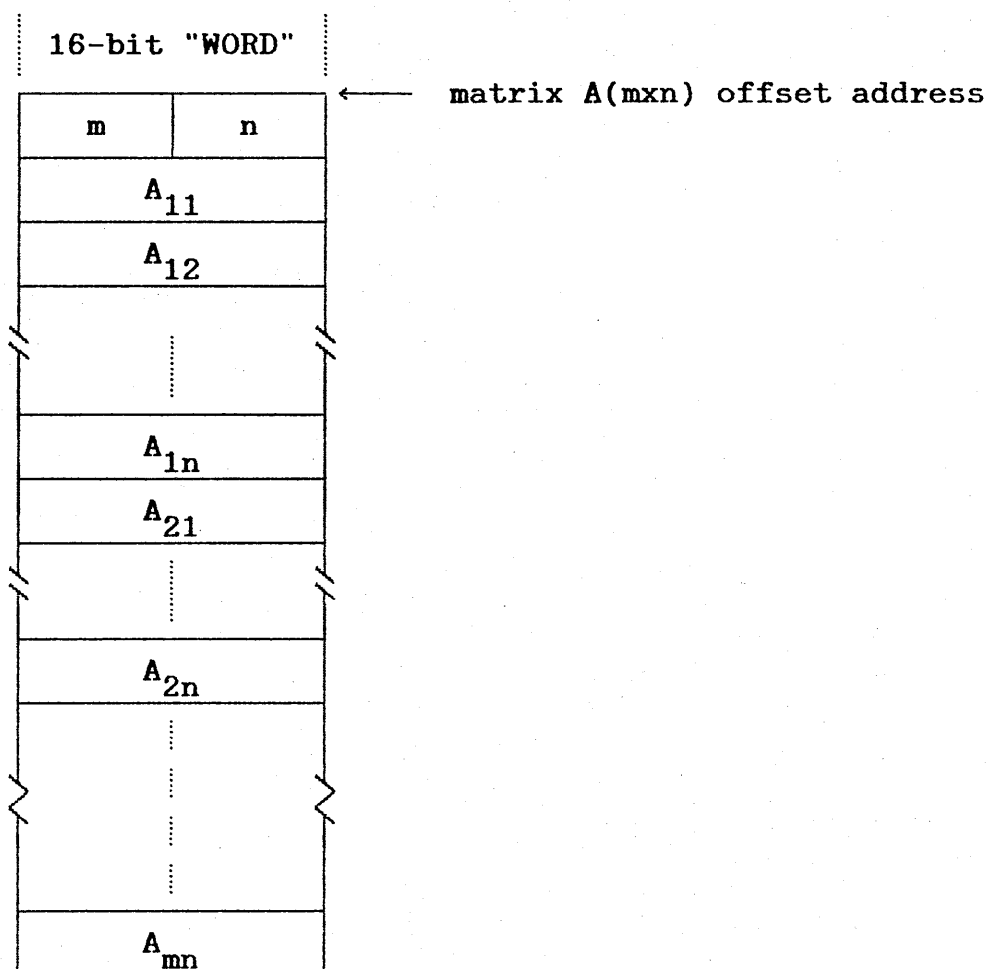


Fig. 5.6 Matrix Format in the Memory

Each matrix element is displayed before being saved, including the minus sign and decimal points. If an element is entered incorrectly it is possible to change it by pressing the keyboard C-KEY (CORRECTION) before the element is saved by pressing D-KEY (DATA) or E-KEY (ENTER). Each element comprises 4 decimal digits and sign (A-KEY), these digits are 0 - 9 only. F-key can be pressed any time to re-start the program again, allowing data editing.

The range 00.00 to 99.99 only is allowed so the system accepts only numbers with a decimal point after two entries and only allows two entries after the decimal point.

The program, when it starts, can work in either of two "MODES":

- Data mode.
- Edit mode.

In DATA mode all the data previously saved in RAM are erased waiting for a new system operation.

The EDIT mode is used for checking the data saved in the RAM which can be modified or corrected.

To reset the program, the user can use SYSTEM RESET by pressing the reset push button, in this case all the RAM area is cleared and the program re-starts, or by pressing F-KEY on the keyboard in which case the program re-starts but the contents of the RAM are not changed.

A number of subroutines are used by the program, and these are now briefly described.

5.5.2 DECIMAL TO BINARY DATA CONVERSION :

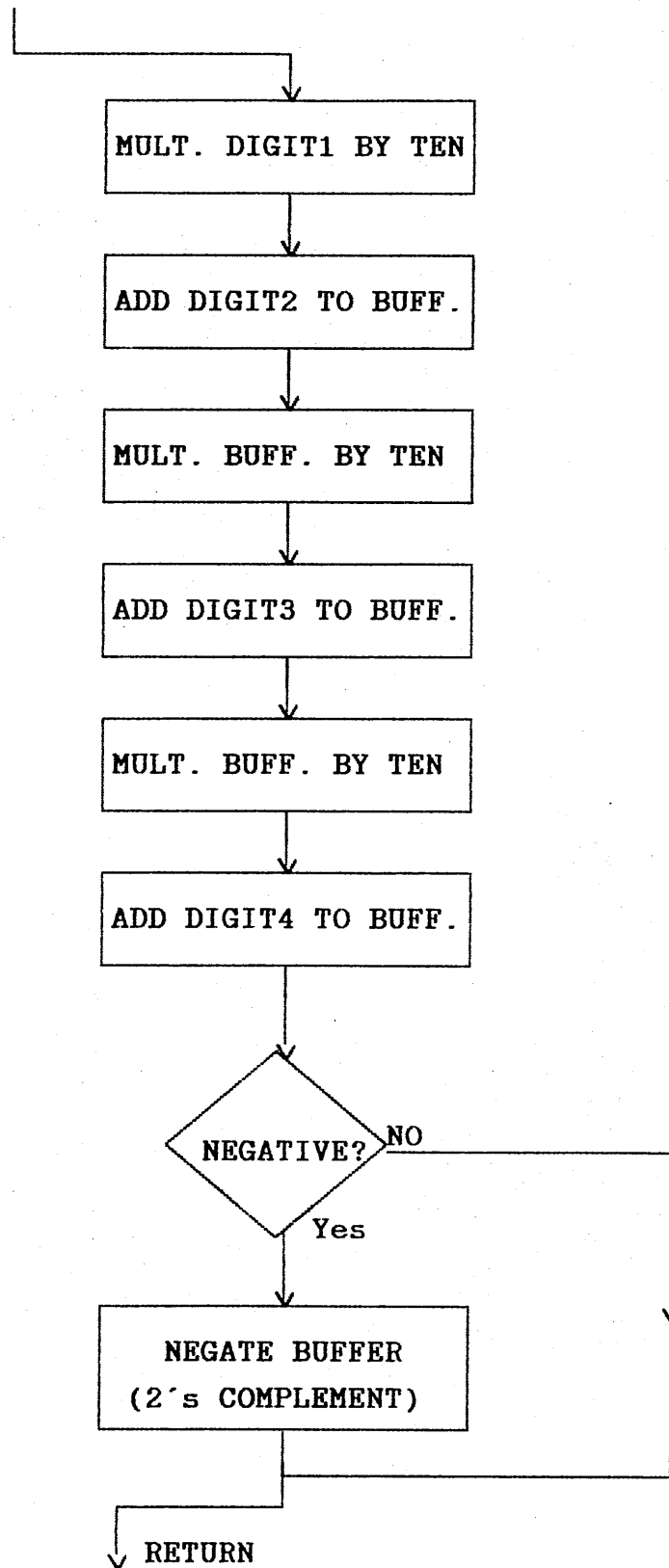
The numerical data entered from the keyboard are in signed decimal format. The subroutine CONV (see source file) is used to change them into appropriate 16-bit scaled 2's complement binary format.

The routine operation is:-

The tens digit (DIGIT1, or the first keyboard entry) is multiplied by 0AH (the Hex equivalent of 10), add the units digit (DIGIT2). This is multiplied by 0AH again, add to the first decimal digit (DIGIT3). After one more multiplication by 0AH the second decimal digit (DIGIT4) is added. Finally, the minus sign is checked, if present the 2's complement of the binary number is calculated then saved at the appropriate memory location.

Fig 5.7 shows the flow chart of this subroutine.

Fig 5.7 The Subroutine CONV (Decimal to Binary Conversion):

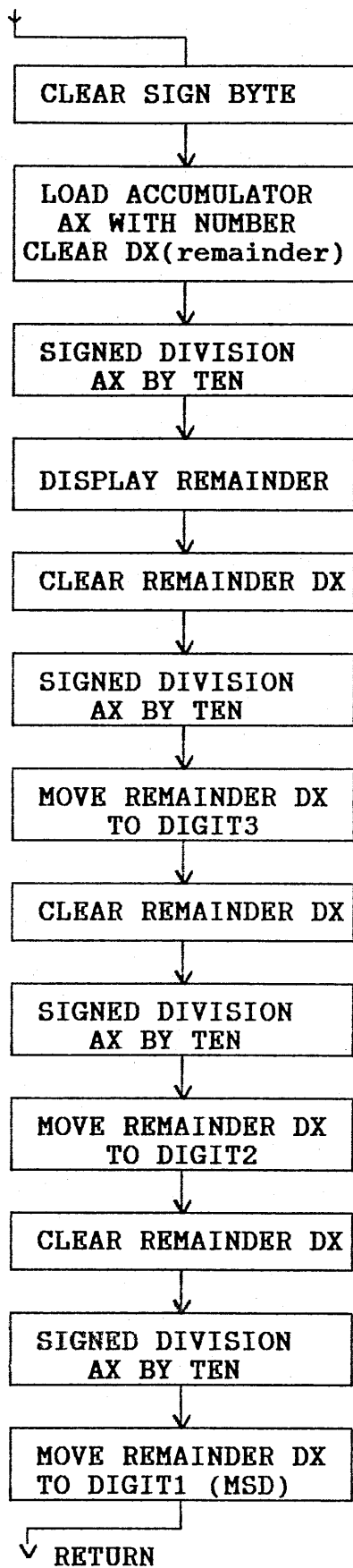


5.5.3. BINARY TO DECIMAL DATA CONVERSION:

During data checking (i.e. using EDIT mode after interrupting the system by the keyboard F-KEY), the data saved in the memory from a previous run are in a 2's complement format. To display the decimal number (in ASCII code) the inverse of the procedure explained above should be followed. This unscales the element number and convert it to decimal form (sign plus 4 digits).

The flow chart in Fig 5.8 is for UNCONV subroutine to do these calculations.

Fig 5.8 The Subroutine UNCONV (Binary to Decimal Conversion):



5.5.4 PROGRAMMING THE SAMPLE TIME:

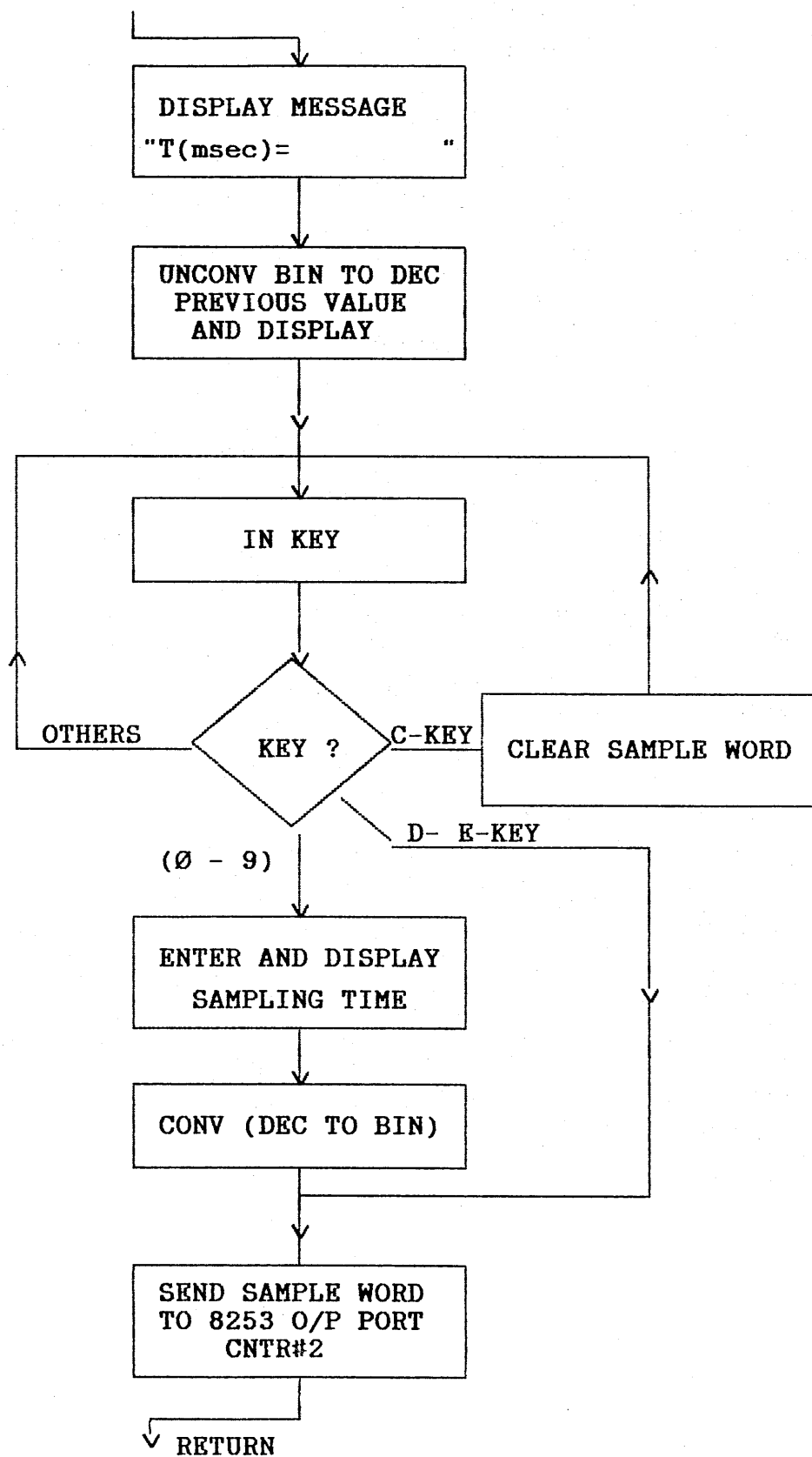
The 8253 Programmable Counter Timer chip is used to interrupt the CPU each sampling time T , requesting an Interrupt Service Routine ISR (contains the estimator algorithm in this case).

Two counters CNTR#1 and CNTR#2 are used in cascaded mode 3 operation (square wave output). Counter CNTR#1 is programmed at a fixed value of 1 msec time interval during the initialization process of the 8253 PCT by the operating system. Counter CNTR#2 is then programmed according to the operator instructions.

To program CNTR#2 for the required sampling period (minimum 1 msec), the user enters the decimal number from the keyboard in 4 digits, when prompted on the display.

The flow chart in Fig 5.9 shows the procedure of programming the 8253 PCT.

Fig 5.9 The Subroutine INIT_8253 (Programming 8253 PCT chip):



When the elements of all the matrices are entered satisfactorily, the program enables Interrupt Request 1&2 (IR1 for keyboard "F-KEY" and IR2 "timer") by sending Output Control Word (OCW2) to the Programmable Interrupt Controller device 8259A PIC.

At this stage the CPU waits in a base load program for an interrupt from a peripheral (the keyboard or the timer). Each time the CPU receives an interrupt signal from the timer at sampling interval T , the Interrupt Service Routine which includes the estimator algorithm is executed.

Appendix-C gives more details about the software and the hardware discussed in this chapter.

The program described above was developed on the 8086 microprocessor rig as described above, and this was separately developed for another project. The resulting circuit and program can now be used for the sub-optimal filters described later in this thesis. A photograph of the rig and the developed version is shown in Fig 5.10.

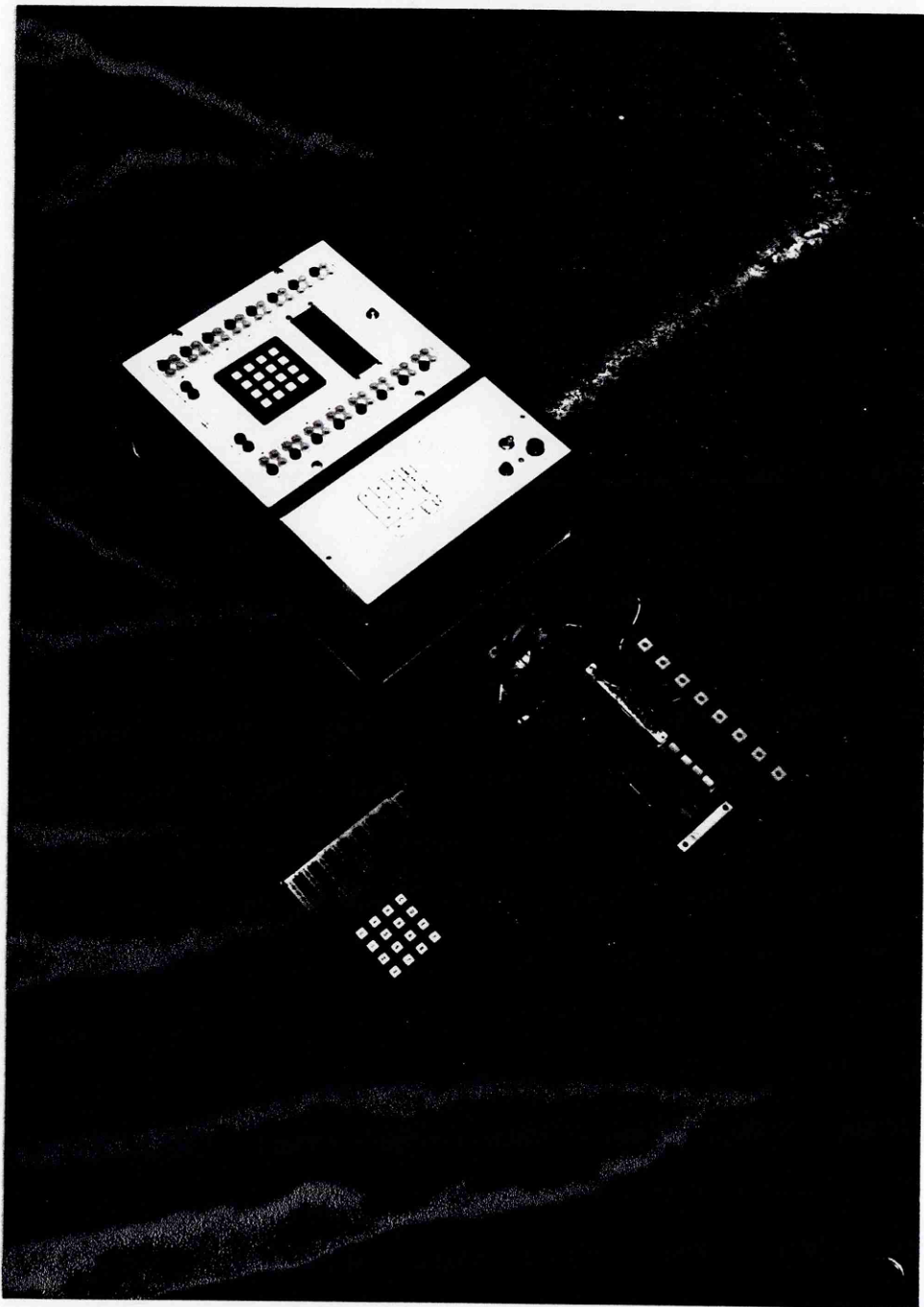
However, it was found that the inconvenience of repeatedly burning ROMs could be avoided by using an LSI11 microcomputer available in this department, and to save time, this was used during much of the work.

The on-board software of the LSI11 provides facilities for editing, compiling and running the program straight away. The programs (for the three forms of the Kalman filter described in Chapter 4) are written in MACRO-11 assembly language which is fully compatible with the PDP11 macro-assembler.

The sub-optimal Kalman filter (or the state estimator) using the 8086 microprocessor described in this chapter was tested on the analog computer (see Appendix D), and the other forms, the split and the UD methods, (also the sub-optimal form) were designed and tested using the LSI11. Unfortunately, there has been insufficient time to test the Kalman filters on the hydraulic equipment described in Chapter 7.

We are confident that the software can all be realized without real trouble using the 8086 configuration described above.

Fig. 5.10 Photograph of the Rig and the Developed Version.



REFERENCES:

- [1] A. Osborne & G. Kane," Osborne 16-Bit Microprocessor Handbook", McGraw-Hill 1981.
- [2] V. K. Bansal "Design of Microprocessor Based Systems", Wiley Eastern, 1985.
- [3] INTEL 8086, DESD Student Notes.
- [4] The 8086 Family User's Manual.

CHAPTER 6

A 2 DOF QUARTER CAR MODEL

6.1 INTRODUCTION:

In Sects 6.2, 6.3 of this chapter, the traditional single wheel station, or "quarter car model" (refs 1-6), is considered. A basic optimal control strategy suited to it is examined in Sects 6.4, 6.5.

Important features of the quarter car model are that it includes a proper representation of the problem of controlling wheel load variations and contains suspension system forces which are applied between wheel mass and body mass (sprung and unsprung masses). It is the simplest model which has these features and possesses advantages over more complex models in terms of [1]:

- (a) being described by relatively few design parameters,
- (b) having only a single input, leading to ease of application of optimal control laws,
- (c) ease of understanding of the relationships between design and performance.

Disadvantages of this model are that it does not take proper account of the hydraulic unit behaviour, or of the system behaviour when the wheel leaves the ground, or of non-linearities. Some of these problems are tackled in Sect 6.6.

6.2 THE QUARTER CAR MODEL:

A traditional, simple linear 2-DOF quarter car model [6] is shown in Fig. 6.1. In the model rigid bodies of masses m_1 (Kg) and m_2 (Kg) representing vehicle body (the sprung mass) and axle (the unsprung mass), perform only motion described by vertical displacements x_b and x_a respectively.

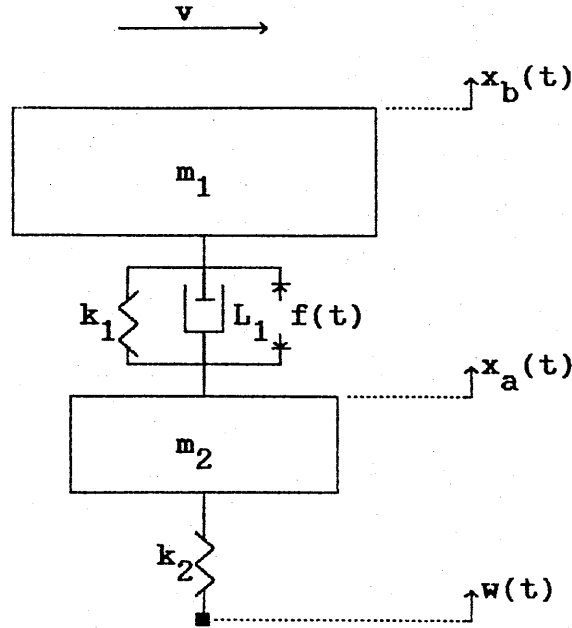


Fig. 6.1 2-DOF Quarter Car Model

Passive spring and damping elements of the suspension are modeled as a linear spring element (stiffness $k_1 \text{ Nm}^{-1}$) and linear damper (damping constant $L_1 \text{ Nsm}^{-1}$). $f(t)$ (N) denotes the active suspension force (if applied). $k_2 \text{ (Nm}^{-1}\text{)}$ is the tyre stiffness, while the tyre damping is neglected. $w(t)$ (m) denotes road irregularities. The vehicle is moving horizontally with constant velocity $v \text{ (ms}^{-1}\text{)}$.

The equations of vertical motion of the system considered are;

$$m_1 \ddot{x}_b + L_1 (\dot{x}_b - \dot{x}_a) + k_1 (x_b - x_a) - f = 0, \quad \dots (6.1a)$$

$$m_2 \ddot{x}_a + L_1 (\dot{x}_a - \dot{x}_b) + k_1 (x_a - x_b) + k_2 (x_a - w) + f = 0 \quad \dots (6.1b)$$

By dividing the first equation by m_1 , the second by m_2 and introducing the state space notation;

$$\begin{aligned}
x_1 &= x_b \\
\dot{x}_2 &= \dot{x}_1 = \dot{x}_b \\
x_3 &= x_a \\
\dot{x}_4 &= \dot{x}_3 = \dot{x}_a \\
u &= f
\end{aligned}
\quad \dots (6.2)$$

a system of first order differential equations is obtained;

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{k_1}{m_1}x_1 - \frac{L_1}{m_1}x_2 + \frac{k_1}{m_1}x_3 + \frac{L_1}{m_1}x_4 + \frac{1}{m_1}u \\
\dot{x}_3 &= x_4
\end{aligned}
\quad \dots (6.3)$$

$$\dot{x}_4 = \frac{k_1}{m_2}x_1 + \frac{L_1}{m_2}x_2 - \frac{k_1+k_2}{m_2}x_3 - \frac{L_1}{m_2}x_4 + \frac{k_2}{m_2}u - \frac{1}{m_2}u$$

This can be written in matrix form as;

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{G}w
\quad \dots (6.4)$$

where;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \emptyset & 1 & \emptyset & \emptyset \\ -k_1/m_1 & -L_1/m_1 & k_1/m_1 & L_1/m_1 \\ \emptyset & \emptyset & \emptyset & 1 \\ k_1/m_2 & L_1/m_2 & -(k_1+k_2)/m_2 & -L_1/m_2 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \emptyset \\ 1/m_1 \\ \emptyset \\ -1/m_2 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \\ k_2/m_2 \end{bmatrix}
\quad \dots (6.5)$$

For a typical vehicle, the following numerical values are used [1]:

$$\begin{aligned}
m_1 &= 500 \text{ Kg} \\
m_2 &= 100 \text{ Kg} \\
k_1 &= 60,000 \text{ N/m} \\
k_2 &= 200,000 \text{ N/m} \\
L_1 &= 6,000 \text{ Ns/m}
\end{aligned}$$

The parameters in Eqns.(6.5) become for this case:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -120 & -12 & 120 & 12 \\ 0 & 0 & 0 & 1 \\ 600 & 60 & -2600 & -60 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.002 \\ 0 \\ -0.01 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2000 \end{bmatrix} \quad \dots (6.6)$$

This system is simulated with a view to illustrating its behaviour using a typical optimal control strategy [2,6].

6.3 THE OPEN LOOP SYSTEM RESPONSE:

The step response of the system is considered first, and the simulated results using ISIS are shown in Figs 6.2 to 6.6 using; $w = 10\text{mm}$ (0.01m). Naturally, since the system is linear, larger road disturbances produce proportionally larger step function responses.

It can be seen that the natural modes of the system are generally of a damped oscillatory nature, corresponding to the eigen values of A (Eqn 6.6):

$$\begin{aligned} &-3.975 \pm j9.934 \quad (\text{dominant}) \\ &-32.616 \pm j32.826 \quad (\text{non-dominant}) \end{aligned}$$

ISIS passive, sprung mass position

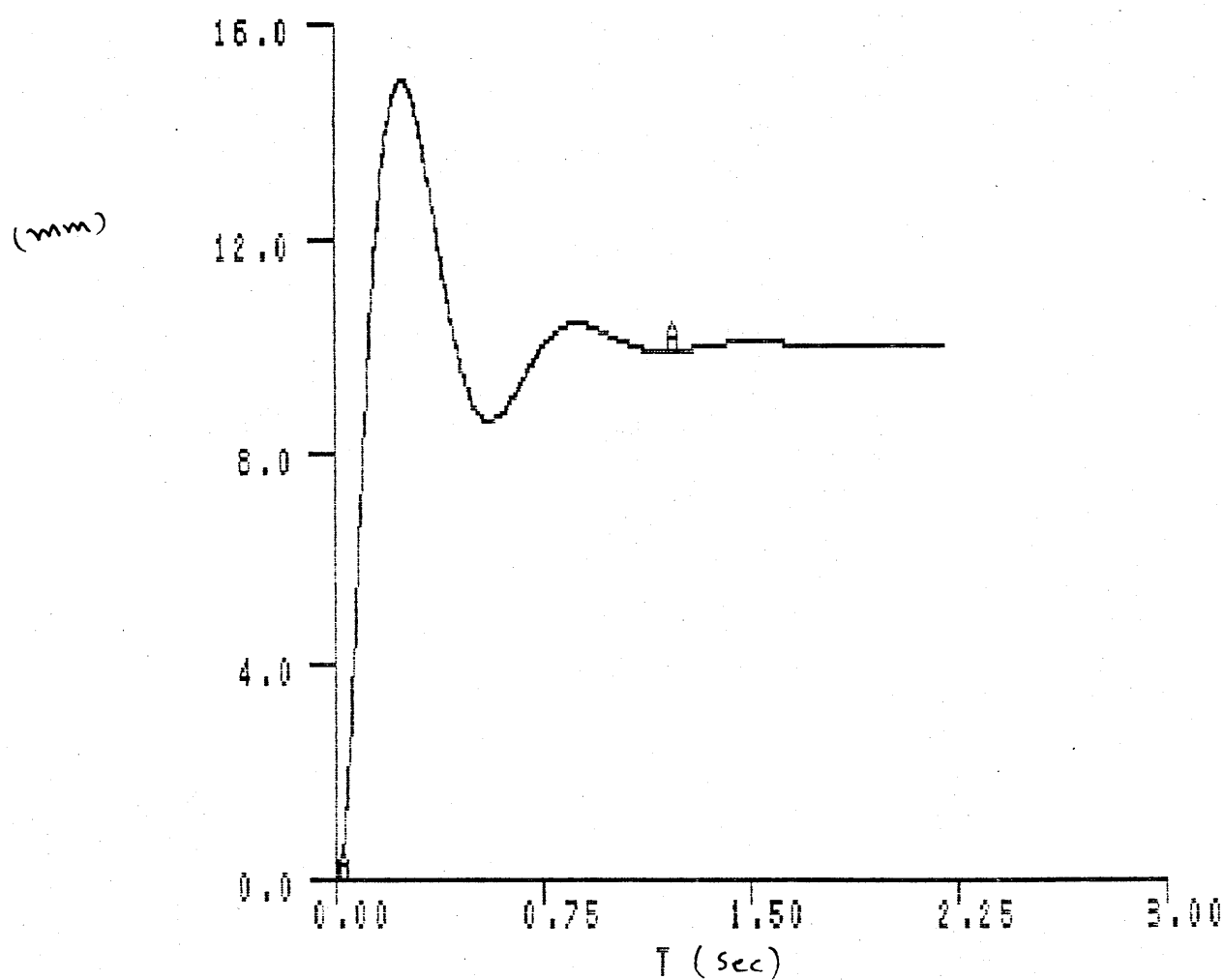


Fig 6.2 Open Loop Step Function Response (x_1)

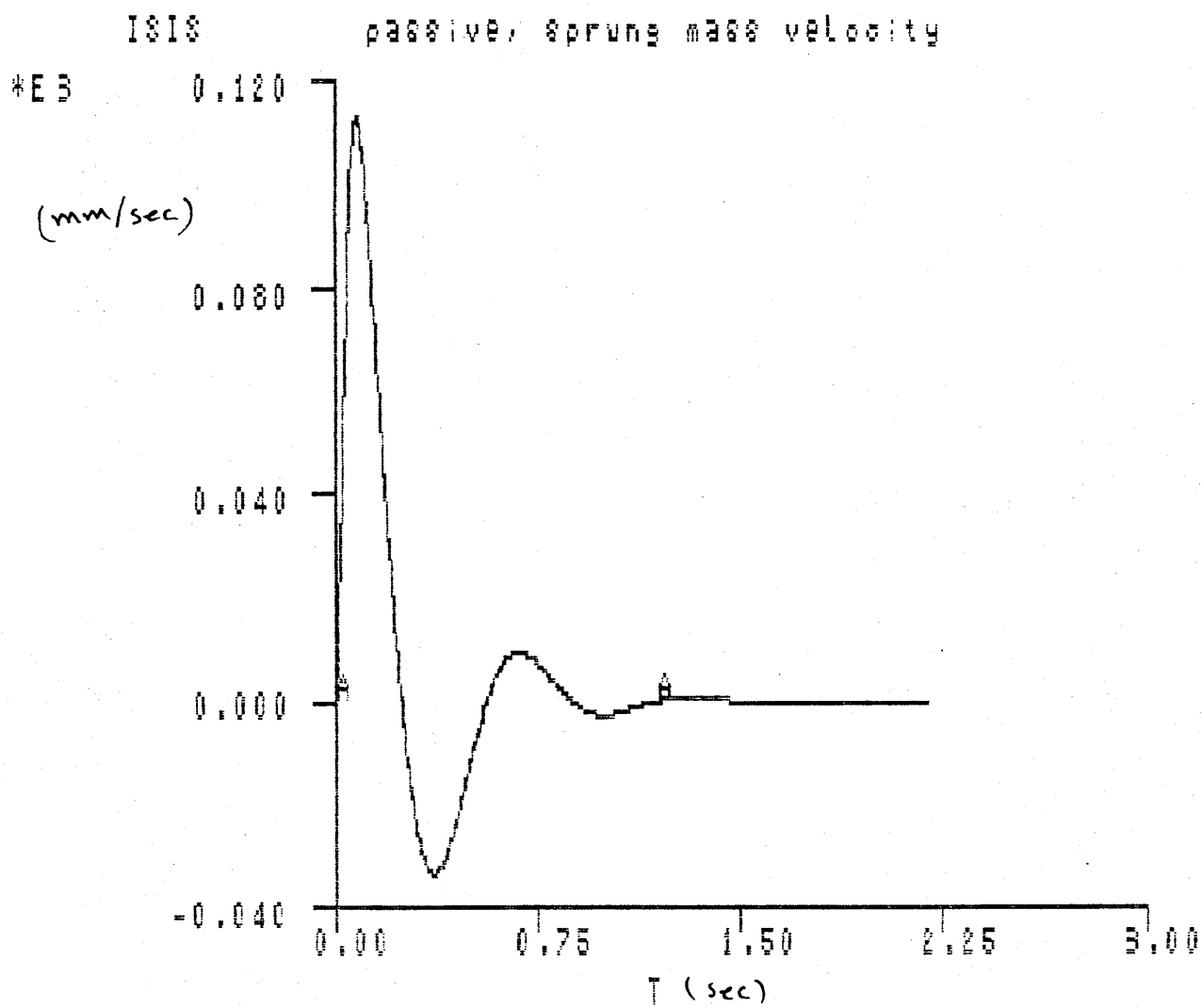


Fig 6.3 Open Loop Step Function Response (x_2)

ISIS

PASSIVE, SPRUNG MASS ACCELERATION

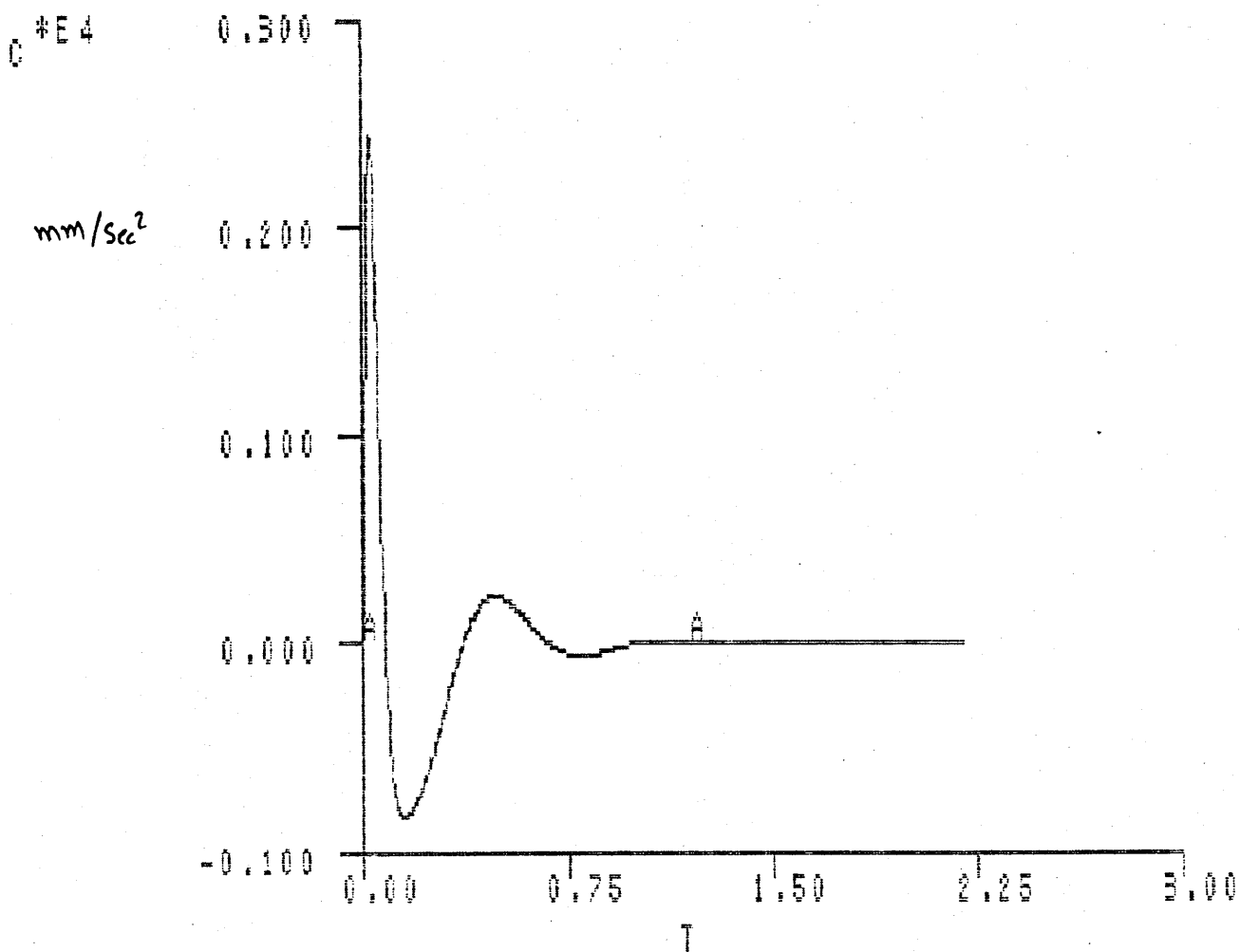


Fig 6.4 Open Loop Step Function Response ($\dot{x}_2$)

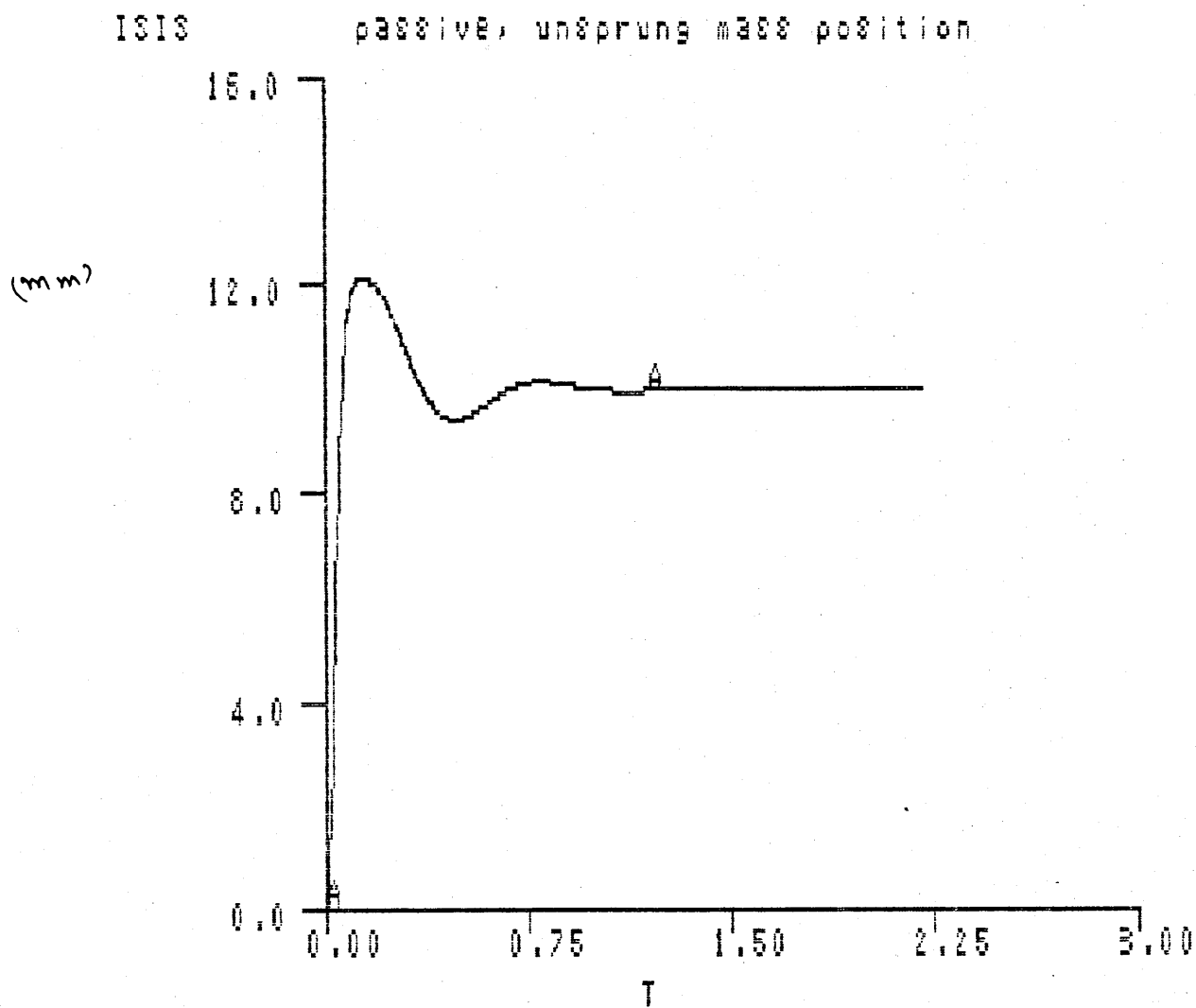


Fig 6.5 Open Loop Step Function Response (x_3)

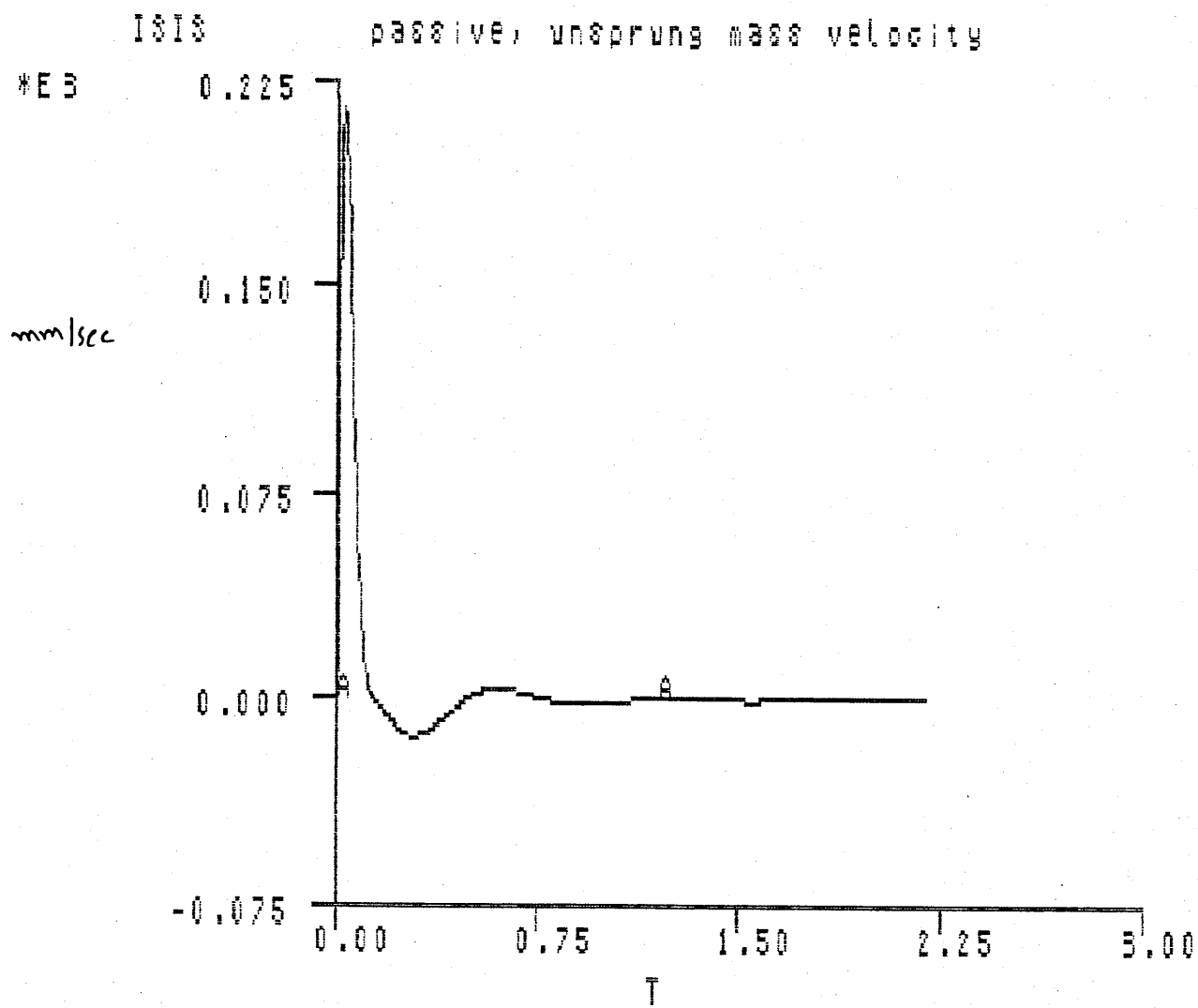


Fig 6.6 Open Loop Step Function Response (x_4)

The maximum sprung mass acceleration (Fig 6.4) is about 2.5 m/s^2 , and the transient lasts for about 1 second. This leaves considerable room for improvement. Other features of interest, such as the unsprung mass behaviour (Fig 6.5) and relative positions of sprung and unsprung mass (Figs 6.2 and 6.5) could also, hopefully, be improved by a suitable active control strategy.

6.4 CLOSED LOOP SYSTEM RESPONSE:

In many studies concerned with active suspensions a control theory approach has been used so that the problem becomes that of minimizing a weighted sum of quadratic state variables and control forces [2,3,4]. In this minimization problem, linear state feedback is used as the strategy for the active suspensions. In this work it is proposed that the states are replaced by estimates, taken as the output from a Kalman filter [6].

The variables selected as a performance index are typically suspension space (or clearance between sprung and unsprung masses), road holding (which is proportional to relative displacement between road and unsprung mass), and the acceleration of the sprung mass relating to the ride quality.

In terms of the system states, the performance index to be minimized for the derivation of optimal control u^* is given by:

$$J = \lim_{T_f \rightarrow \infty} E \int_0^{T_f} \left[x^T \bar{Q}_c x + u^T \bar{R}_c u \right] dt \quad \dots (6.7)$$

where the matrices $\bar{Q}_c$ and $\bar{R}_c$ are defined to be positive semi definite and positive definite respectively. They are of the form [2,6]:

$$\bar{Q}_c = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix}, \quad \bar{R}_c = \begin{bmatrix} \rho \end{bmatrix} \quad \dots (6.8)$$

where $\rho, q_1 \dots q_4$ are constants.

ρ is chosen to limit the control input to what is achievable in practice. $q_1 \dots q_4$ are chosen according to the states whose variance is to be minimized by the optimal control strategy.

$T_f \rightarrow \infty$ is taken to mean a time which is "long" compared with the natural dynamics of the system - in this case, about 1.5 sec (Figs 6.2 - 6.4).

From optimal control theory [8], the optimal control input $u^*(t)$ is derived as:

$$u^*(t) = -K_c \hat{x}(t) = -\bar{R}_c^{-1} B^T P_c \hat{x}(t) \quad \dots (6.9)$$

where P_c is the positive definite solution of the algebraic matrix Riccati equation;

$$P_c A + A^T P_c - P_c B \bar{R}_c^{-1} B^T P_c + \bar{Q}_c = 0 \quad \dots (6.10)$$

The VAX computer Fortran program ARE1 is used in this work for solving Equations (6.10 & 6.9) to find K_c (the optimal feedback gain vector). This gain vector is then used by the ISIS to simulate system response driven by the optimal input $u^*(t)$.

The active suspension is shown in Fig. 6.7 (cf. Fig 6.1).

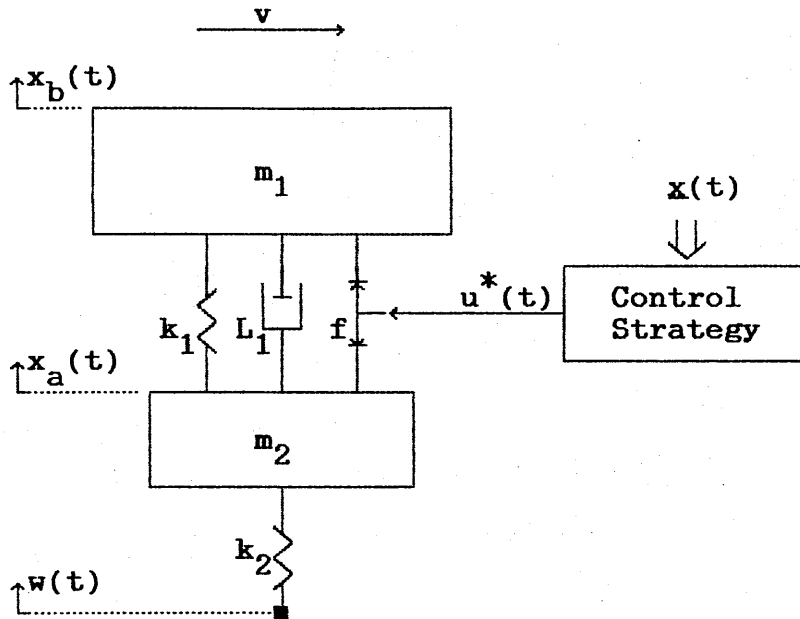


Fig. 6.7 Quarter Car Active Suspension System

6.5 DIGITAL SIMULATION:

Assuming the 4th order model of Eqns 6.4, 6.6 and that the actuator dynamics are "fast", various optimal control strategies can fairly easily be designed.

To achieve ride comfort demand, a major control requirement is to reduce the sprung mass acceleration. However, the 4th order model of Eqn. 6.4, 6.6 does not include acceleration terms and it is proposed, for illustration purposes, to try cost functions which limit the sprung mass positions x_1 , and velocity $\dot{x}_2$; and these will of course affect the acceleration $\ddot{x}_2$ via Eqns 6.3, 6.4.

Choosing $\bar{Q}_c$, $\bar{R}_c$ (Eqn. 6.8) as:

$$\bar{Q}_c = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{R}_c = \begin{bmatrix} 1.0 \times 10^{-5} \end{bmatrix} \quad \dots (6.11)$$

In this case the cost function is;

$$J = E \int_0^{\infty} (10x_1^2 + x_2^2 + 0x_3^2 + 0x_4^2 + 1.0 \times 10^{-9} u^2) dt$$

This has the effect of limiting the position x_1 more than the velocity x_2 . Solution of Eqns.(6.10 & 6.9) give:

$$K_c = \begin{bmatrix} 0.5662 \times 10^5 & 0.2853 \times 10^5 & -1.4229 \times 10^5 & 0.0184 \times 10^5 \end{bmatrix} \dots (6.12)$$

The eigen values of A matrix (Eqn 6.6) are:

$$-32.616 \pm j32.826$$

$$-3.975 \pm j9.934$$

The eigen values of $(A-BK_c)$ are:

$$-3.7743 + j0.0$$

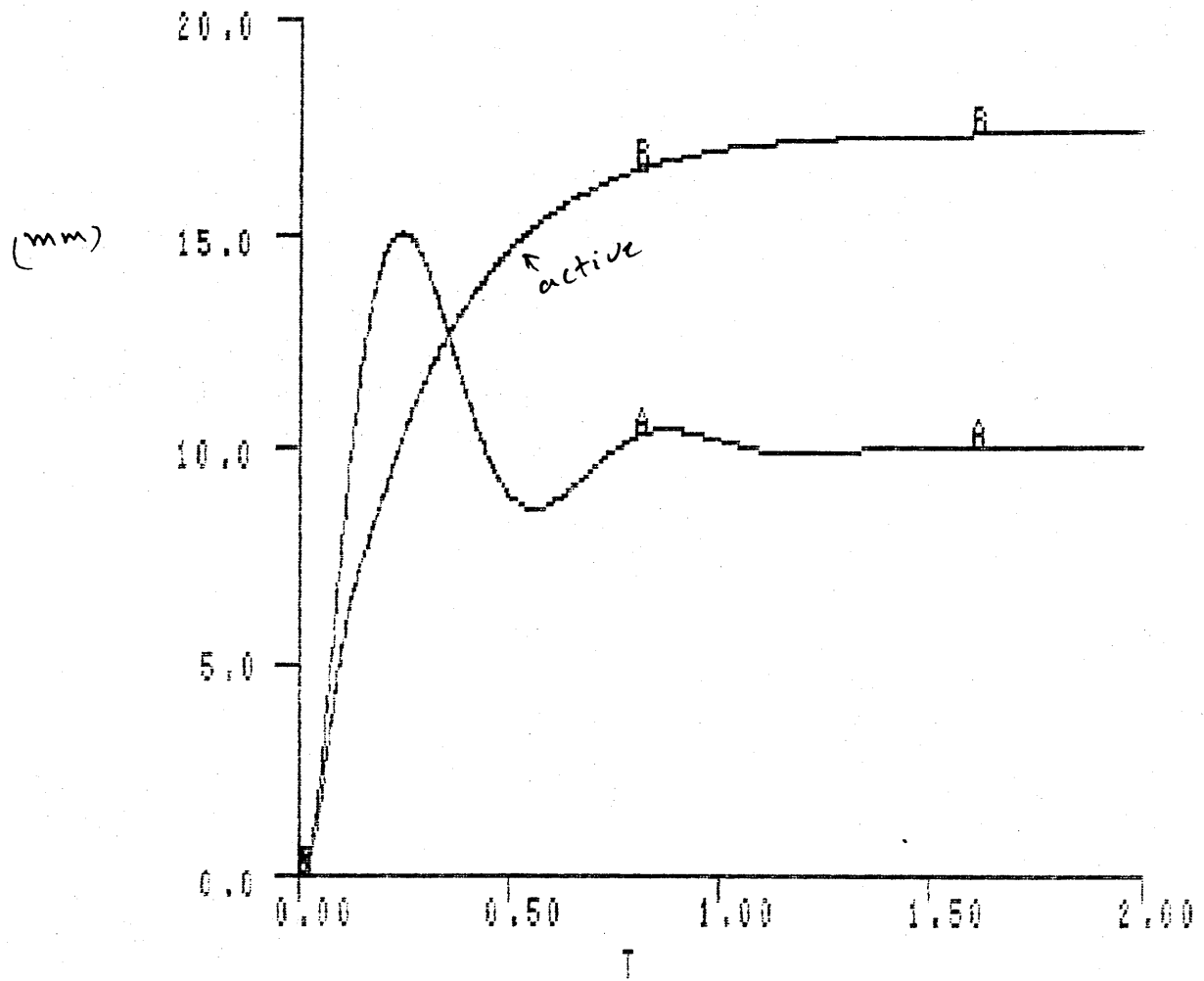
$$-14.5475 \pm j37.1$$

$$-77.8388 + j0.0$$

Figs. 6.8 to 6.14 show step response computer simulation of this optimal system compared with the passive system (for sprung mass position x_1 , velocity x_2 , acceleration $\dot{x}_2$ and unsprung mass position x_3 and velocity x_4).

1818

Passive & Active Sprung Mass Position

Fig 6.8 Open and Closed Loop Step Function Response (x_1)

1813

Passive & Active Spring Mass Velocity

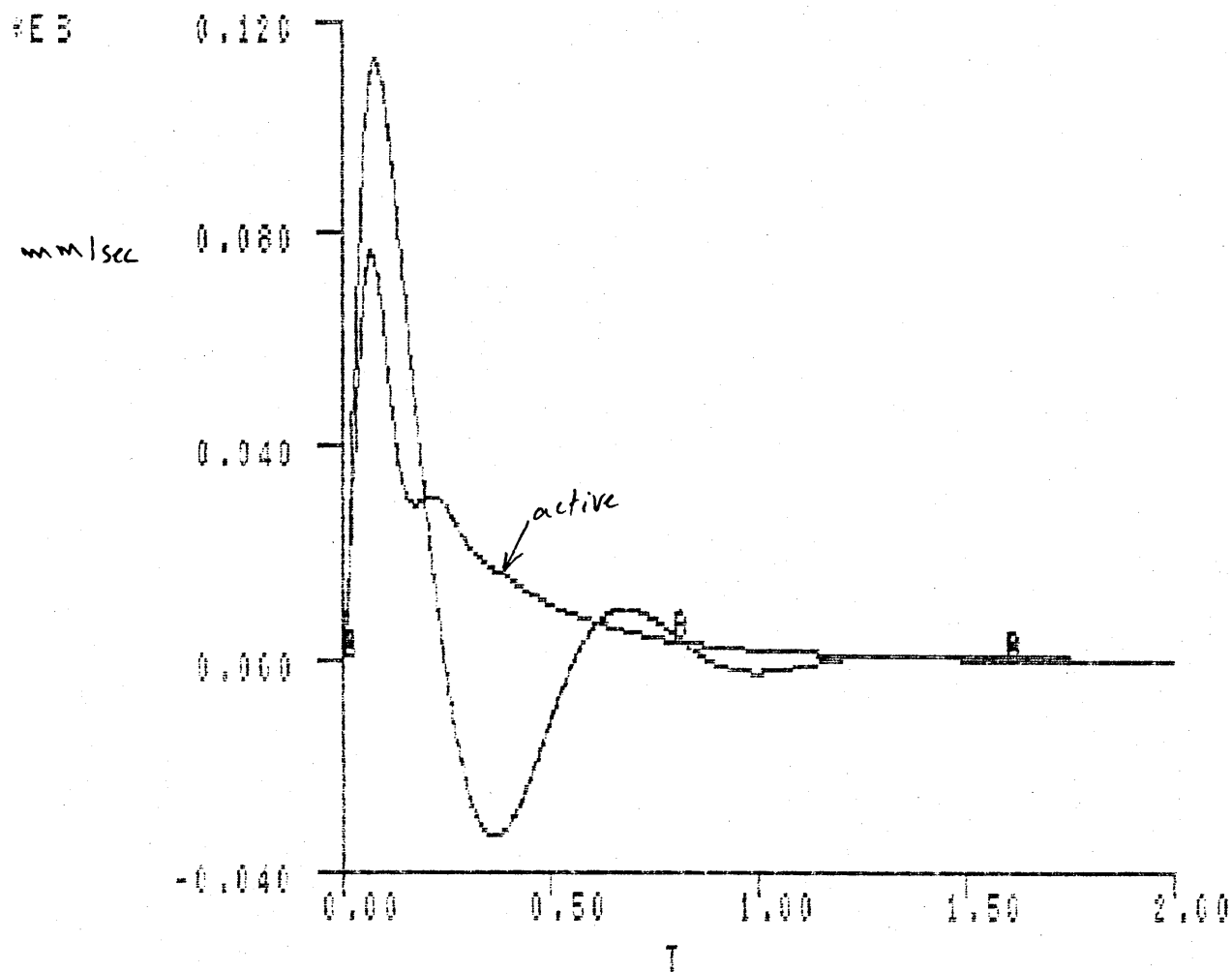


Fig 6.9 Open and Closed Loop Step Function Response (x_2)

ISIS

Passive & Active Sprung Mass Acceleratio

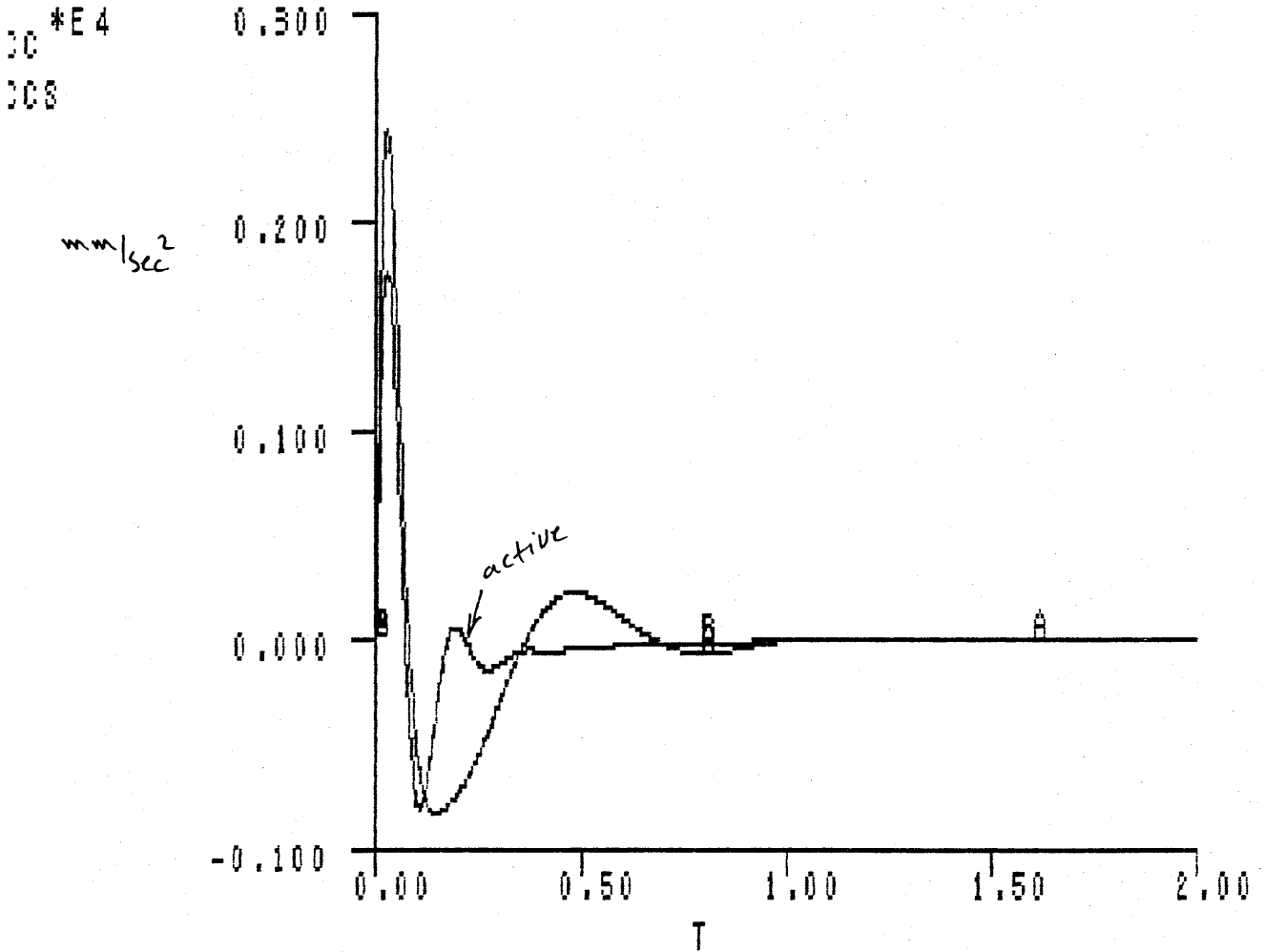
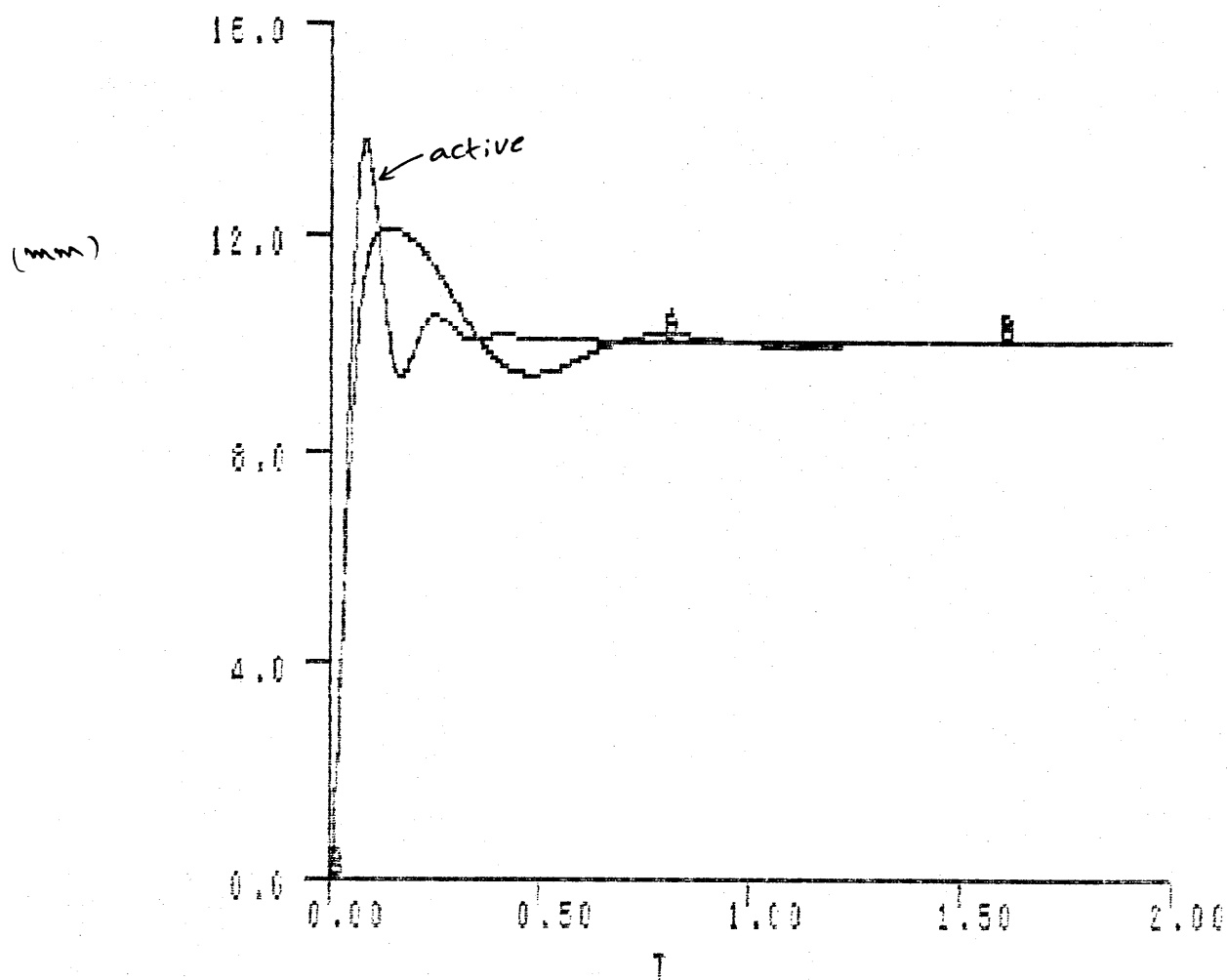


Fig 6.10 Open and Closed Loop Step Function Response ($\dot{x}_2$)

1818

Passive & Active Unsprung Mass Position

Fig 6.11 Open and Closed Loop Step Function Response (x_3)

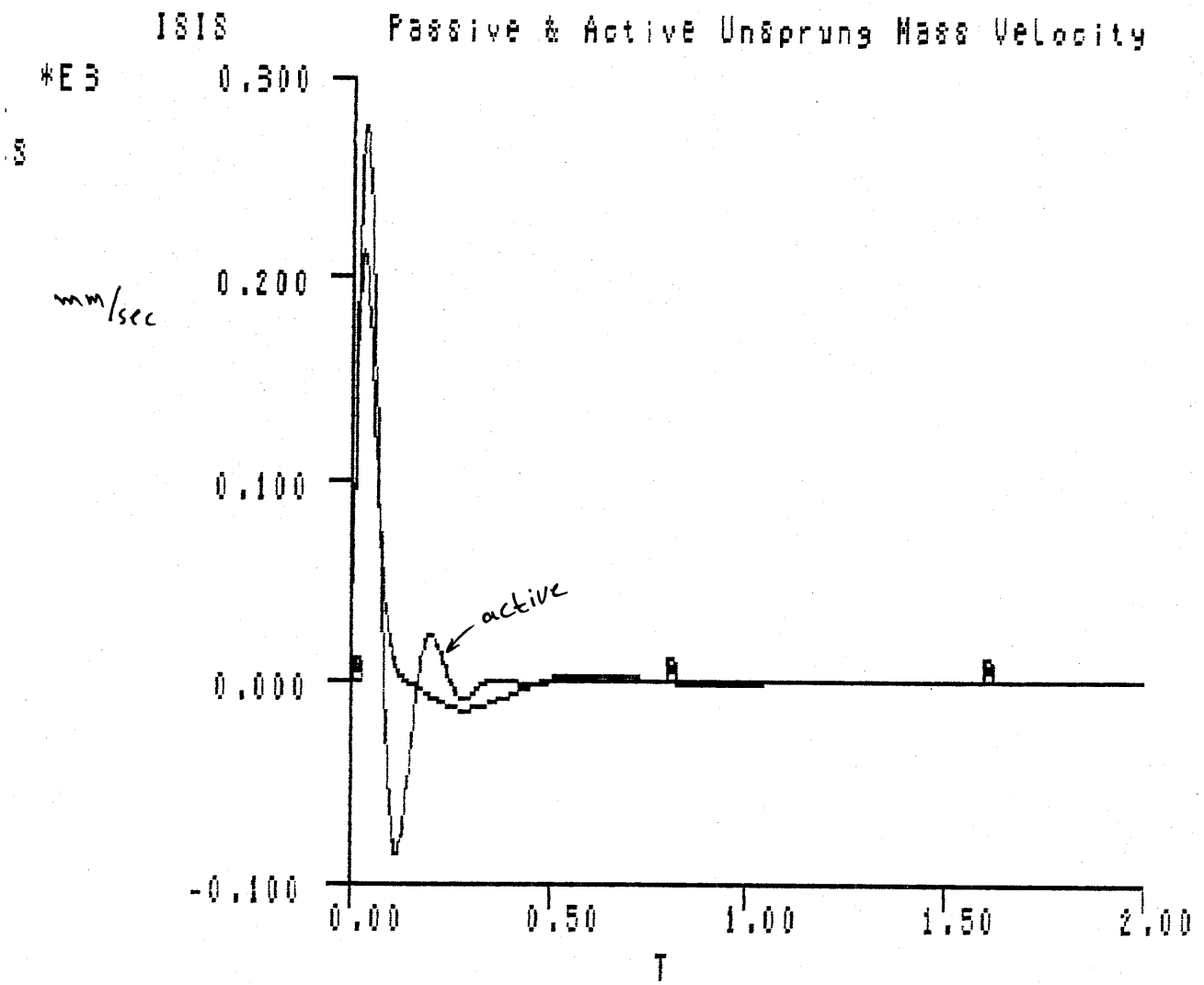


Fig 6.12 Open and Closed Loop Step Function Response (x_4)

ISIS

Passive & Active Road Holding (1)

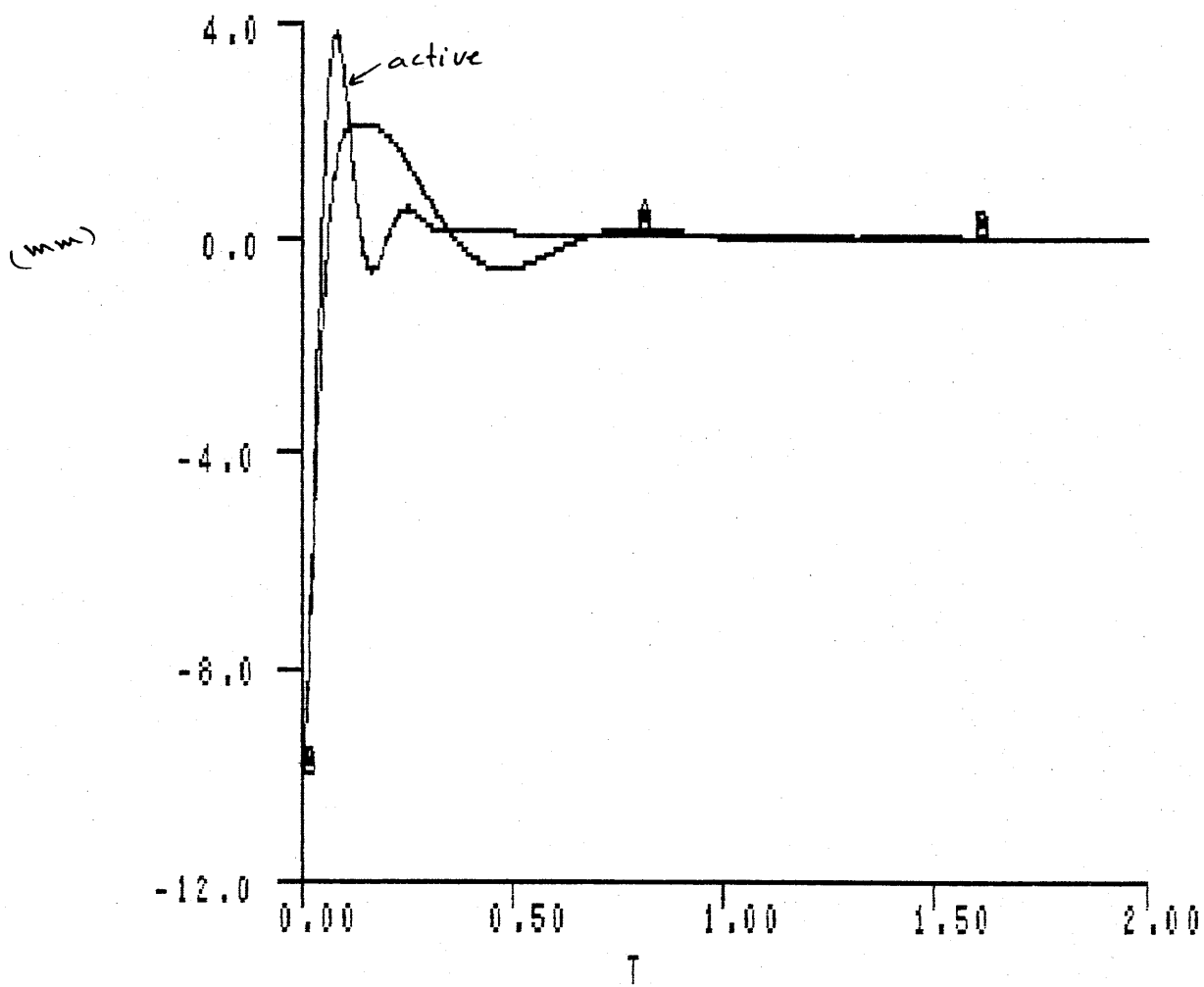


Fig 6.13 Open and Closed Loop Step Function Response (x_3-w)

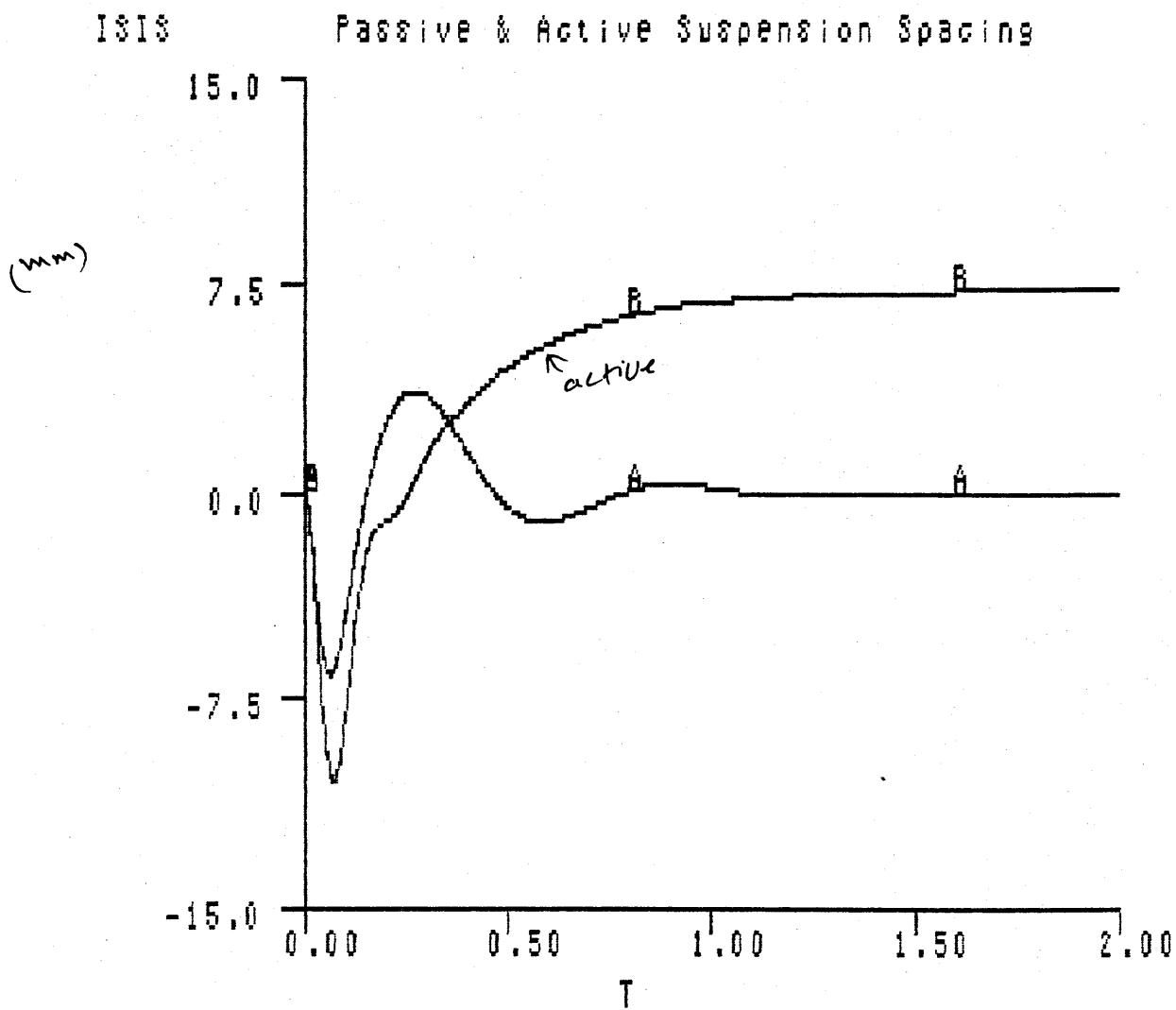


Fig 6.14 Open and Closed Loop Step Function Response ($x_1 - x_3$)

It can be seen (Fig 6.8) that the sprung mass position of the active suspension is considerably affected by the control strategy, not least in that it has a dc content of about 17mm. In a practical control strategy, this must be dealt with, and this is explored in Chapter 7.

The maximum acceleration (Fig. 6.10) is reduced from 2.5 m/s^2 to about 1.8 m/s^2 , showing some improvement. Clearly by altering the cost function (Eqn 6.8) different performances can be achieved. This is not pursued further here, since the objective in this chapter is merely to describe the application of the optimal control strategy to this problem. The road holding (x_3-w) and the suspension working space (x_1-x_3) are also affected as shown in Fig. 6.13 and Fig 6.14 respectively. The improvements here are not great, but as before, better cost functions could no doubt be found.

Figs 6.16 to 6.18 show the sprung mass position, velocity and acceleration response to random disturbance shown in Fig. 6.15, for uncontrolled and controlled systems. Again, some modest improvements are obtained from the controlled system.

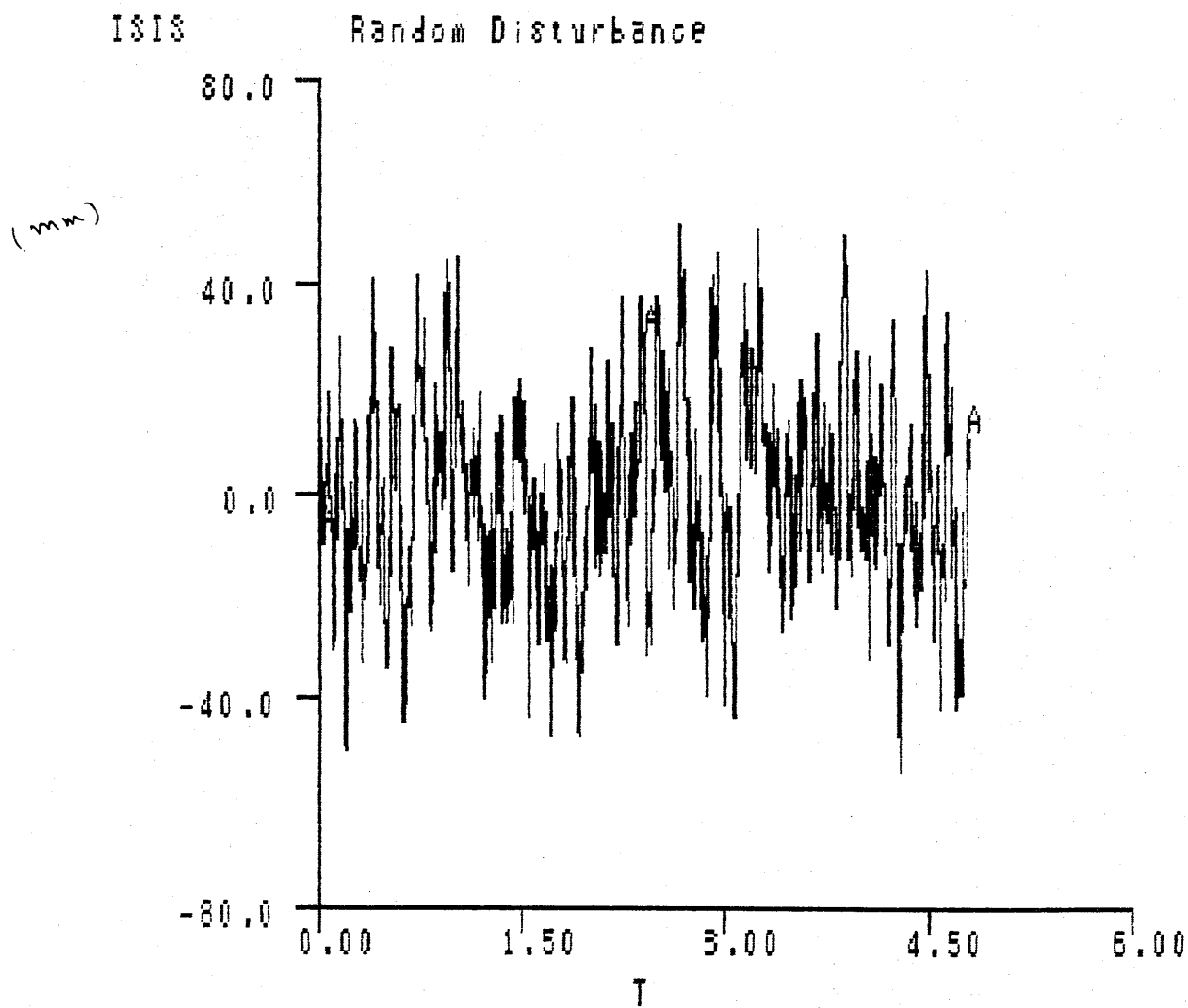


Fig 6.15 Random Disturbance (w)

IS13

Passive & Active Sprung Mass Position

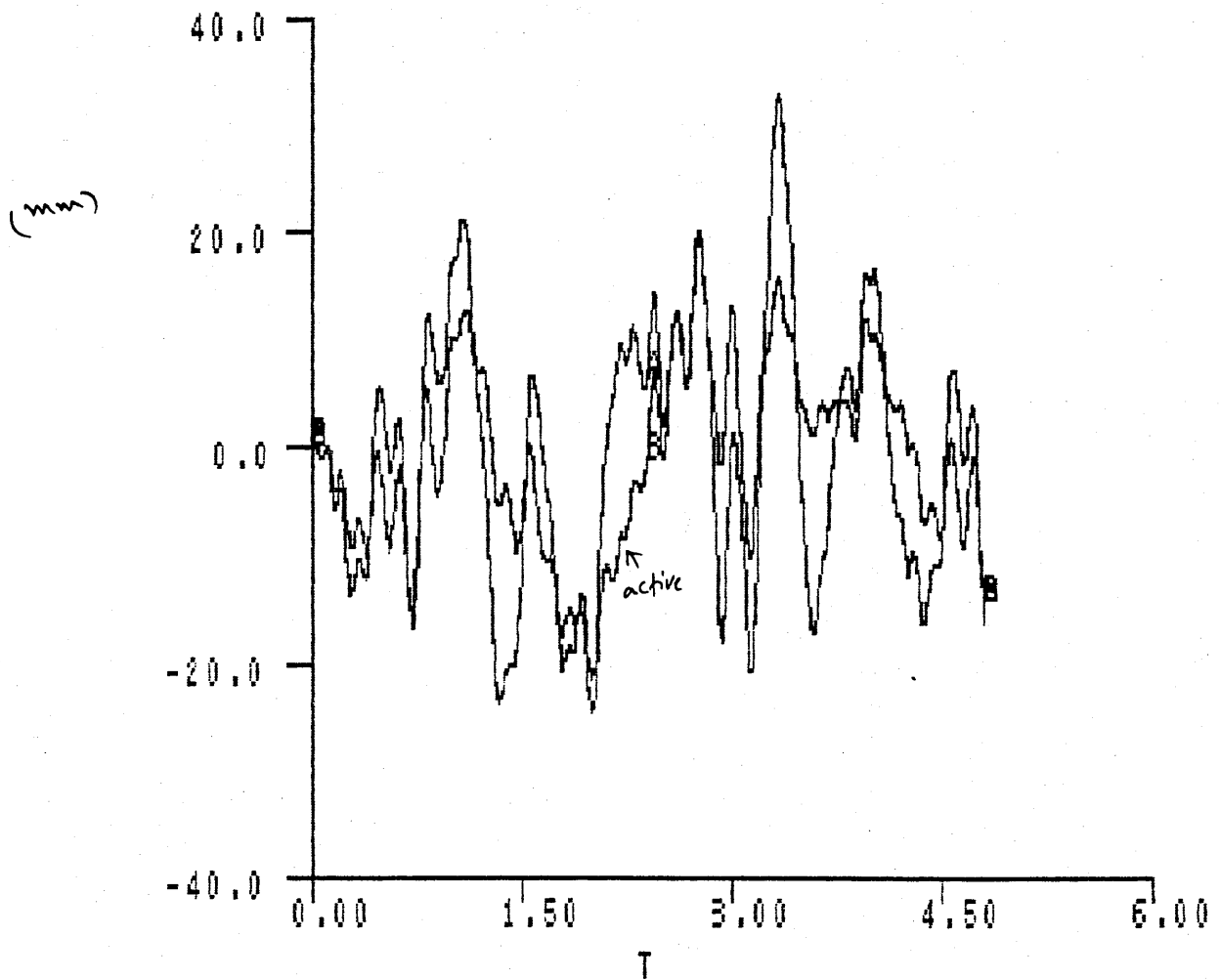


Fig 6.16 Open and Closed Loop Random Disturbance Response (x_1)

ISIS Passive & Active Sprung Mass Velocity

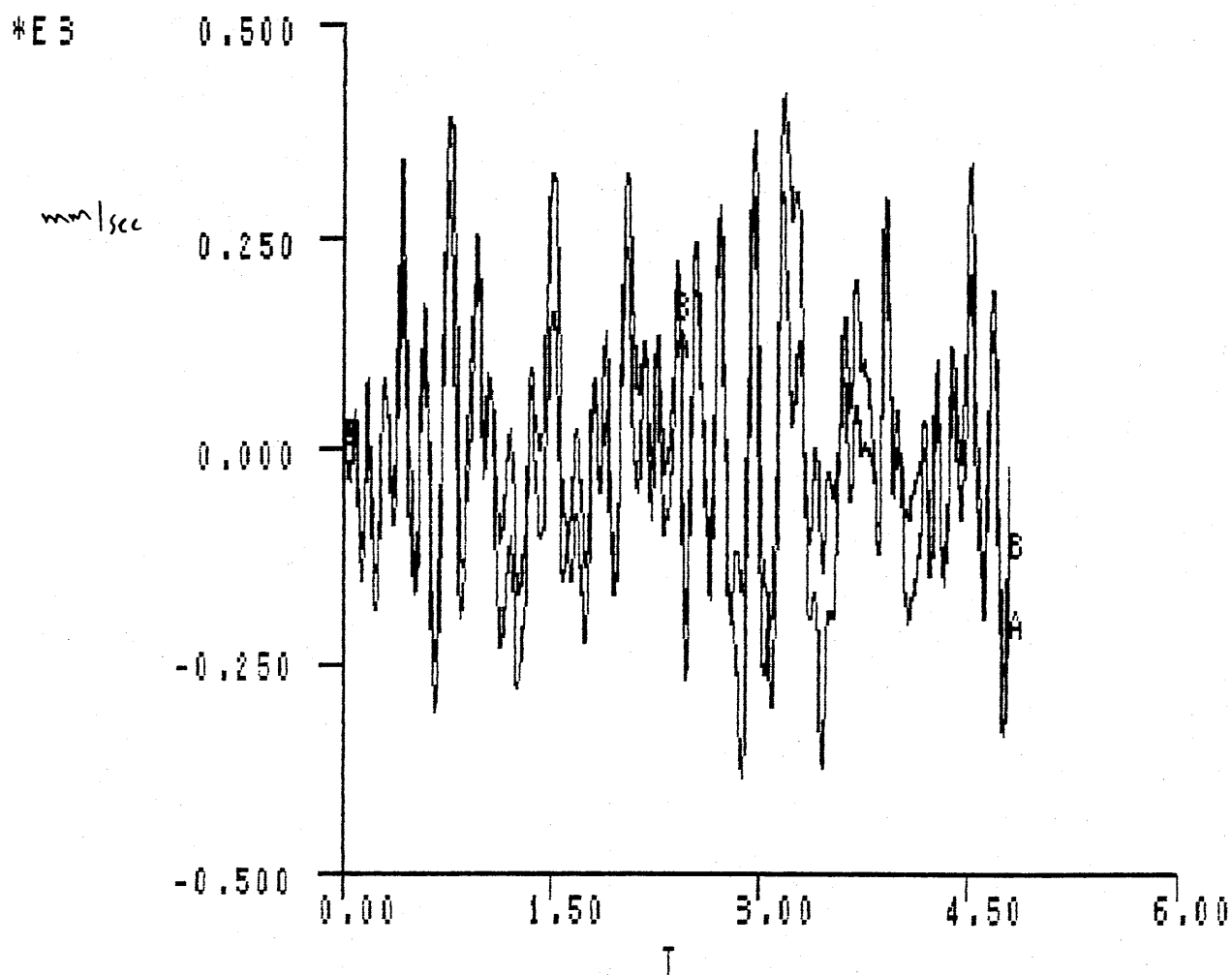


Fig 6.17 Open and Closed Loop Random Disturbance Response (x_2)

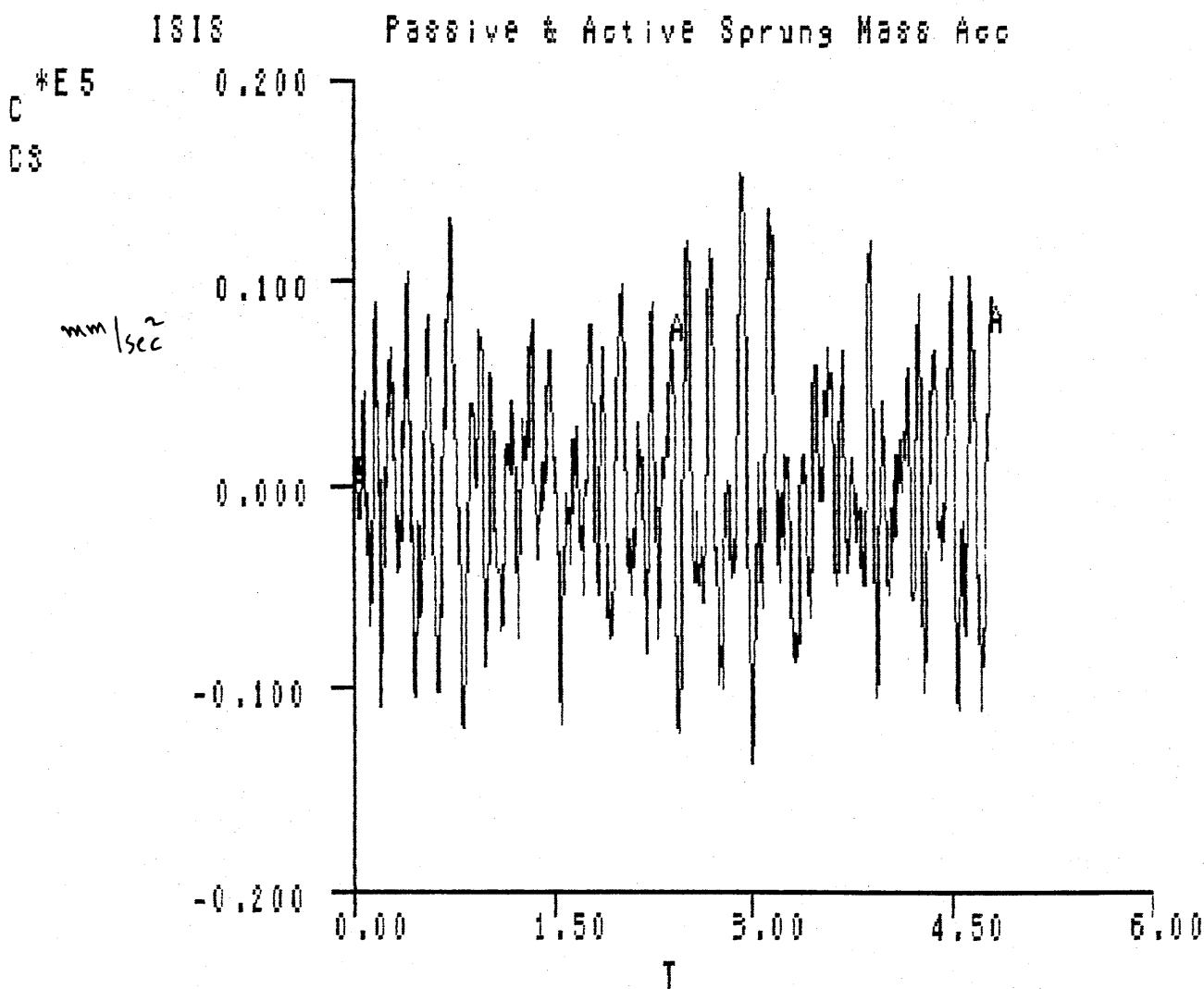


Fig 6.18 Open and Closed Loop Random Disturbance Response ($\ddot{x}_2$)

6.6 MORE DEVELOPED QUARTER CAR MODELS:

The main drawback of the model of Eqn (6.3) is that no account is taken of the behaviour of the hydraulic system providing the control force $u(t)$. The model also fails to represent the system behaviour when the tyre leaves the ground - with the result that road holding is not properly considered.

As shown in Chapter 7, the force generated by a hydraulic actuator driven by an amplifier and valve, can be modeled by an equation of the form:

$$\ddot{f} + \alpha \dot{f} + \beta f = \gamma u$$

where u is the input to the amplifier (volts).

f is the force generated by the actuator (N).

α, β, γ are constants.

Using this, the model of Eqn 6.1(a) can be augmented to account for the hydraulic system behaviour:

$$m_1 \ddot{x}_b + L_1 (\dot{x}_b - \dot{x}_a) + k_1 (x_b - x_a) - f = 0 \quad \dots (6.13)$$

$$\ddot{f} + \alpha \dot{f} + \beta f = \gamma u \quad \dots (6.14)$$

where the symbols of Sect 6.2 retain their meanings except u , which is redefined as the amplifier input.

Together with Eqn 6.1(b), Eqns (6.13), (6.14) yield the state space model (cf Eqns (6.4), (6.5)):

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -k_1/m_1 & -L_1/m_1 & 1/m_1 & 0 & k_1/m_1 & L_1/m_1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\beta & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ k_1/m_2 & L_1/m_2 & -1/m_1 & 0 & -(k_1+k_2)/m_2 & -L_1/m_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \gamma \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ k_2/m_2 \end{bmatrix} w \quad \dots (6.15)$$

where

$$x_1 = x_b$$

$$x_2 = \dot{x}_1 = \dot{x}_b$$

$$x_3 = f$$

$$x_4 = \dot{f} = \dot{x}_3$$

$$x_5 = x_a$$

$$x_6 = \dot{x}_a = \dot{x}_5$$

Eqns (6.13) to (6.15) apply to the situation where the tyre is in contact with the ground - ie $x_a \leq w$.

If this is not so, ie $x_a > w$, Eqn (6.13), (6.14), (6.1b) must be respecified:

$$m_1 \ddot{x}_b + L_1(\dot{x}_b - \dot{x}_a) + k_1(x_b - x_a) - f + m_1 g = 0 \quad \dots (6.16)$$

$$\ddot{f} + \alpha \dot{f} + \beta f = \gamma u \quad \dots (6.17)$$

$$m_2 \ddot{x}_a + L_1(\dot{x}_a - \dot{x}_b) + k_1(x_a - x_b) + f + m_2 g = 0 \quad \dots (6.18)$$

where $g = 9.81 \text{ ms}^{-2}$

Eqns (6.16) to (6.18) yield the state space model:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -k_1/m_1 & -L_1/m_1 & 1/m_1 & 0 & k_1/m_1 & L_1/m_1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\beta & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ k_1/m_2 & L_1/m_2 & -1/m_1 & 0 & -k_1/m_2 & -L_1/m_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \gamma \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} g \quad \dots (6.19)$$

The models of Eqns (6.15) and (5.19) are different in some important respects;

(a) Eqn (6.15), represents a theoretically controllable, but in practice almost uncontrollable system, since the effect of u on x_5 , x_6 is very small ($k_2 \gg k_1$, $k_2 \gg L_1$) compared with that of w . This reflects the fact that the control input has little effect on the unsprung mass behaviour compared with the ground surface, when the tyre is in contact with that surface.

(b) Eqn (6.19), represents a controllable system, though the effect of gravity (treated as "noise", g) is important on both the sprung and unsprung masses.

In this case, while there is no escaping downward acceleration of the whole system, one control objective is to force the unsprung mass downwards quickly to re-establish tyre contact with the ground.

In this thesis, tyre contact with the ground is generally assumed in the design of the Kalman filter and control strategies. However, behaviour of these must eventually be checked against the model of Eqn (6.19) to ensure satisfactory road holding performance.

Further elaboration of these models may also eventually prove necessary and some work has been done on representing the main nonlinearities of hydraulic actuators. This is outlined in Appendix-G.

These models are not pursued further here, but they must in the long term be proved valid against suitable experimental equipment, and the Kalman filter and control strategies of Chapter 7 (modified if necessary) must finally be checked using these models and that equipment.

6.7 CONCLUSION:

Comparing the step responses of the passive and the active suspension system shown in the previous paragraphs, it can be concluded that in the active suspension the body motions are improved to some extent. Work on the cost function parameters (Eqn 6.8) could no doubt yield improved performance.

The value of $\bar{R}_c$ in the cost function (Eqn. 6.7) is related to the control force available from the actuator, and is a function of the power consumption of the active suspension, while $\bar{Q}_c$ is chosen to meet the suspension system demands. A proper choice of $\bar{Q}_c$ and $\bar{R}_c$ can only be made by a combination of theoretical considerations and practical experiment.

More elaborate models, which may also be required later in the project have also been considered.

In this work an experimental model representing the main features of a quarter car model is constructed and the practical effects of various cost functions are examined. This is discussed in detail in Chapter 7. Our eventual objective is to suggest Kalman filters and associated optimal control strategies so that measurements of the sprung and the unsprung mass positions are avoided.

REFERENCES:

- [1] R. S. Sharp and D. A. Crolla, "Road Vehicle Suspension System Design - a review", Vehicle system dynamics (16) 1987.
- [2] A. G. Thompson, " An Active Suspension with Optimal Linear State Feedback", Vehicle system dynamic (5) 1976.
- [3] T. Yoshimura, " An Active Vertical Suspension For Track/Vehicle Systems", Journal of Sound and Vibration (1987) 115(3).
- [4] D. A. Wilson, R. S. Sharp, and S. A. HASSAN, "The Application of Linear Optimal Control Theory to the Design of Active Automotive Suspension", Vehicle system dynamics.
- [5] A. G. Thompson, " Optimal and Suboptimal Linear Active Suspensions for Road Vehicle", Vehicle System Dynamics, vol 13 (1984).
- [6] A. Hac, "Suspension Optimization of a 2-DOF Vehicle Model Using a Stochastic Optimal Control Technique", Journal of Sound and vibration 1985) 100(3).
- [7] P. Bark and D. Hrovat, "Application of the LOG Approach to Design of an Automotive Suspension for 3D Vehicle Model", (Advanced Suspension), International Conference 1988, vol 9.
- [8] Arthur E Bryson, Jr. Yu-Chi Ho, "Applied Optimal Control", Hemisphere Publishing Corp. 1975.

C H A P T E R 7

ACTIVE SUSPENSION VEHICLE EXPERIMENTAL MODEL

7.1 INTRODUCTION:

It was decided to build a laboratory model of a simplified quarter car active suspension system to gain insight into the operation of hydraulic servomechanisms and practical optimal control strategies in this context. It was expected that the model would also be useful in examining realizations of Kalman filters using on-line microprocessors and in pointing the way to practical combinations of Kalman filter and control strategies. No attempt was made to build a damped suspension system or to simulate tyre behaviour.

The laboratory model is scaled with respect to the quarter car model of Chapter 6 at 1:100 (rather roughly) for masses and forces, and 1:1 for displacements, velocities and accelerations.

The laboratory model is shown in block form in Fig 7.1a. photographs are shown in Figs 7.1b and 7.1c.

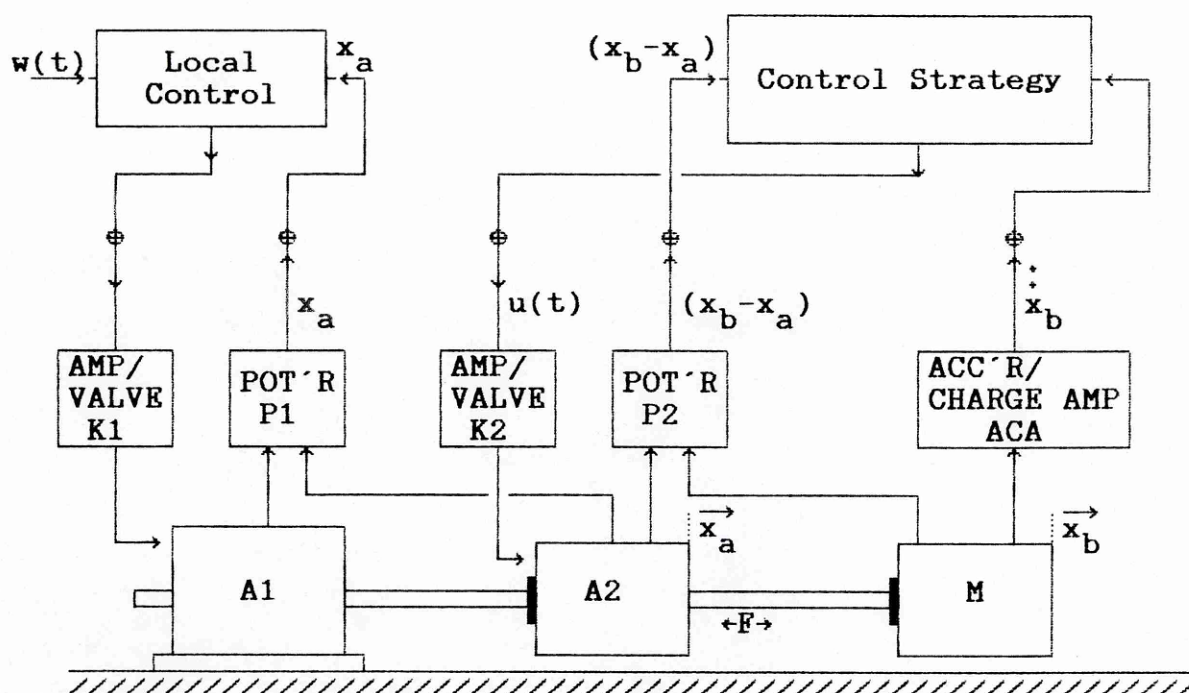


Fig. 7.1a Laboratory Quarter Car Suspension Model

Fig. 7.1b Photograph of the Laboratory Equipment

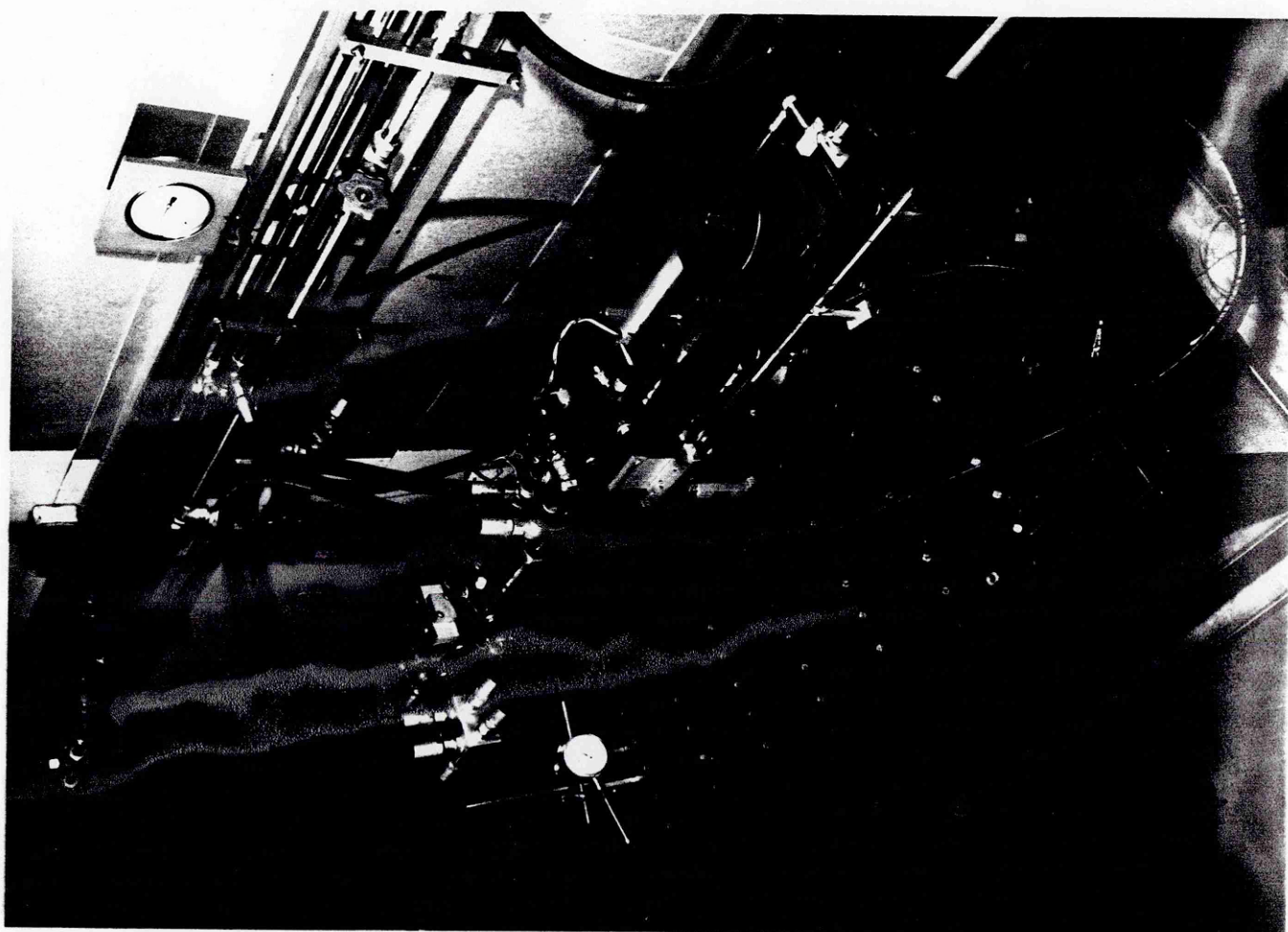
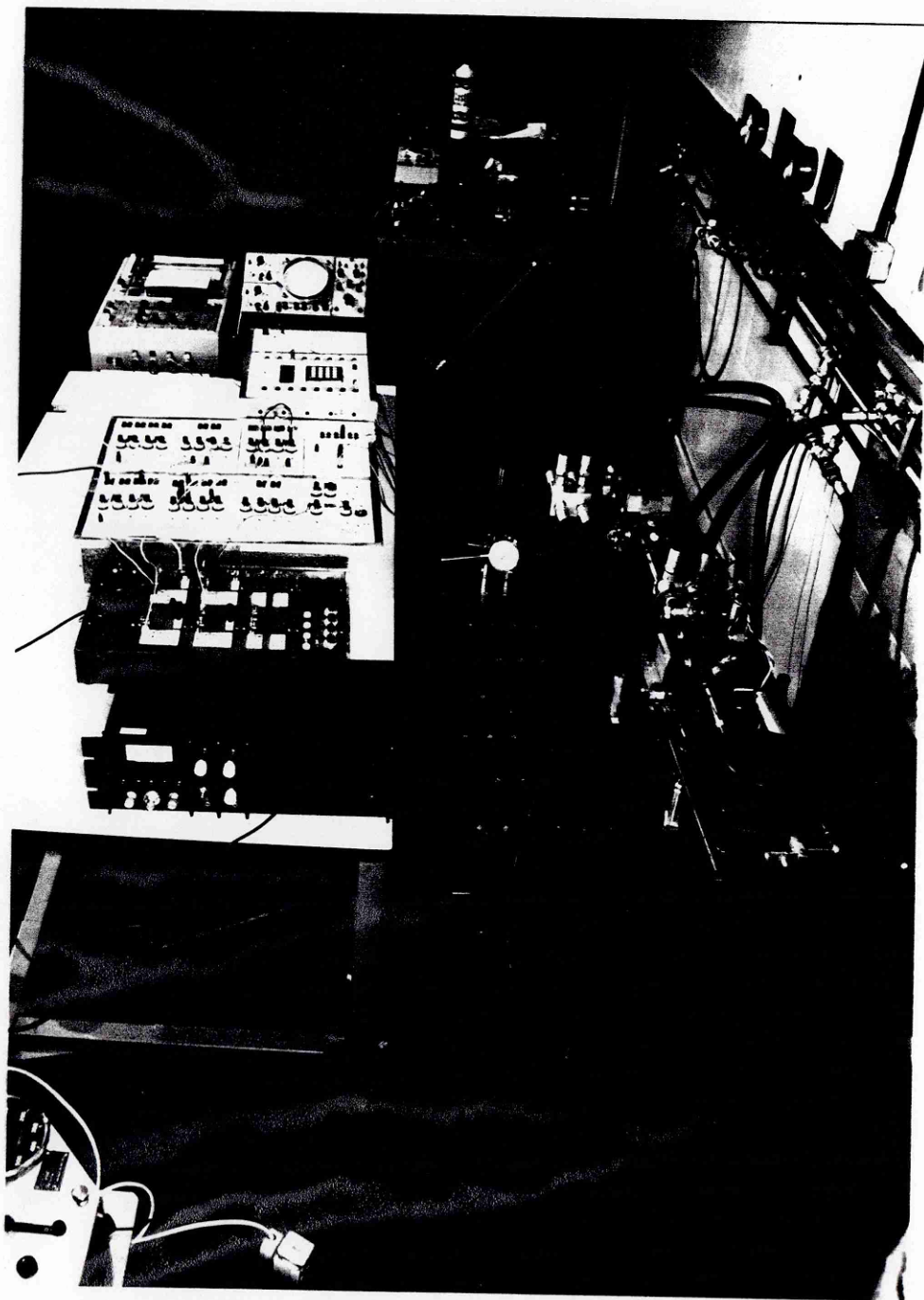


Fig. 7.1c Photograph of the Laboratory Equipment
(The Hydraulic Equipment and the Controllers)



As shown in Fig. 7.1a, two hydraulic actuators are used. The actuator A1 (double stroke, 6.0 cm double acting, 1000psi electro-hydraulic) is part of a local servo loop. Random signals (w) can be fed to this so that the actuator output represents road disturbance as seen by the unsprung mass.

The ram of A1 drives the actuator A2 (single stroke, 6.0 cm double acting, 1000psi electro-hydraulic), which represents the unsprung mass of the quarter car. The ram of A2 is attached to a mass M representing the sprung mass (5 Kg total sprung mass). Both A2 and M are mounted on horizontal slides, so that they can move horizontally without undue friction.

An accelerometer is mounted on the mass block (M) to measure its acceleration. The absolute position of the unsprung mass (chassis of A2) is monitored by a potentiometer " P_1 " (12cm, $\pm 15V$).

The position of the sprung mass (M) relative to the unsprung mass is monitored by a second potentiometer " P_2 " (20 cm, $\pm 15V$).

The displacement voltages are scaled to 1.8 volts/cm and biased using the circuit shown in Fig. 7.2.

Figs 7.1 and 7.2 indicate the electro-hydraulic equipment and its associated electronics, which includes a local control amplifier closing a local control loop incorporating the actuator A1.

It is of interest to feedback the system states according to various optimal control strategies (Sect 7.3) and this is done in an analog controller as shown in Fig. 7.3a. The states are taken from the various outputs (Figs 7.1, 7.2), or generated from these where necessary. Further processing networks are described in the text where appropriate. It is also of interest to consider various Kalman filters which are required to generate the state estimates needed by these control strategies (since all the states will not be available directly in a practical system), and this is illustrated in block form in Fig. 7.3b.

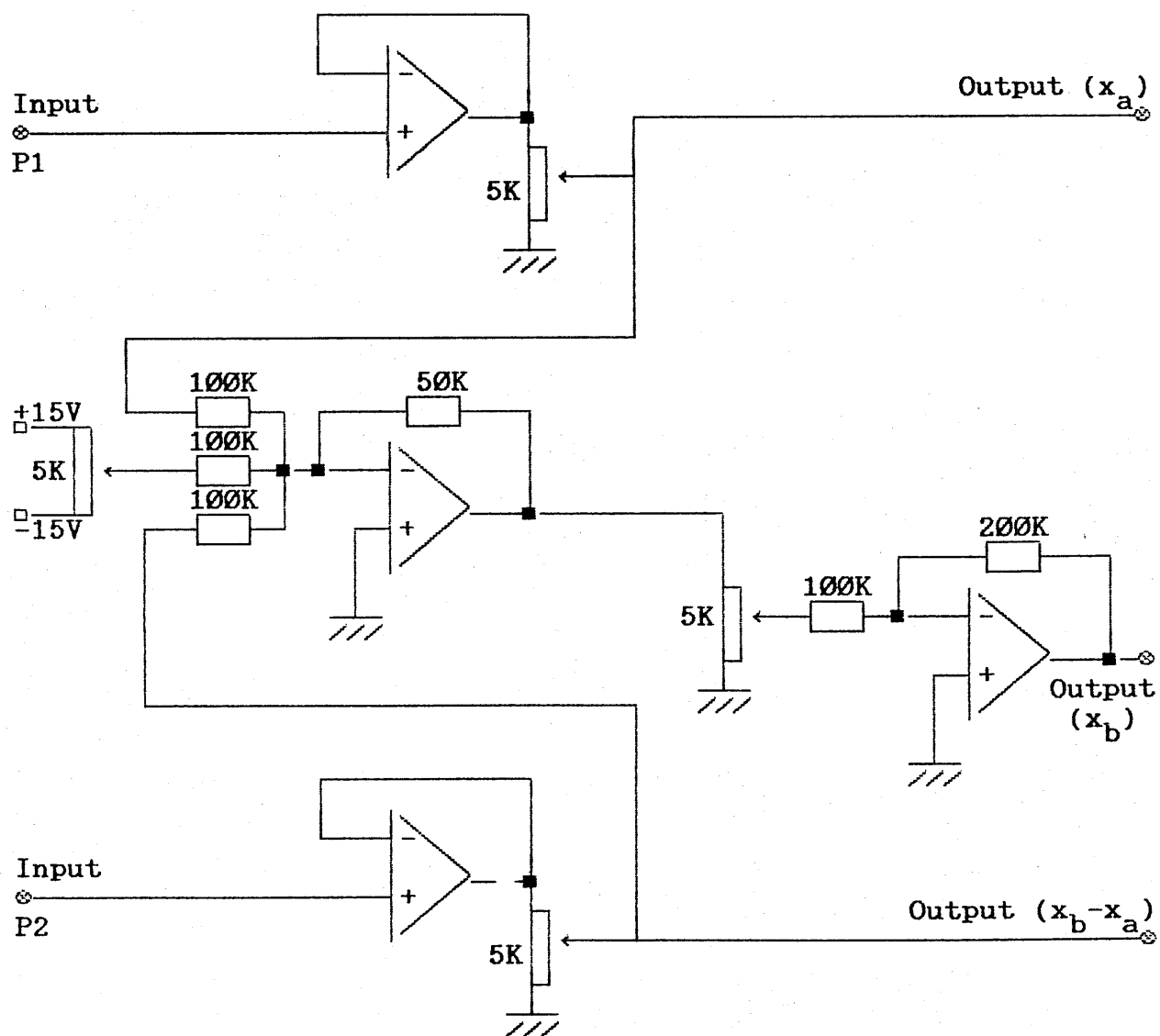


Fig. 7.2 Displacement Biasing and Scaling Network

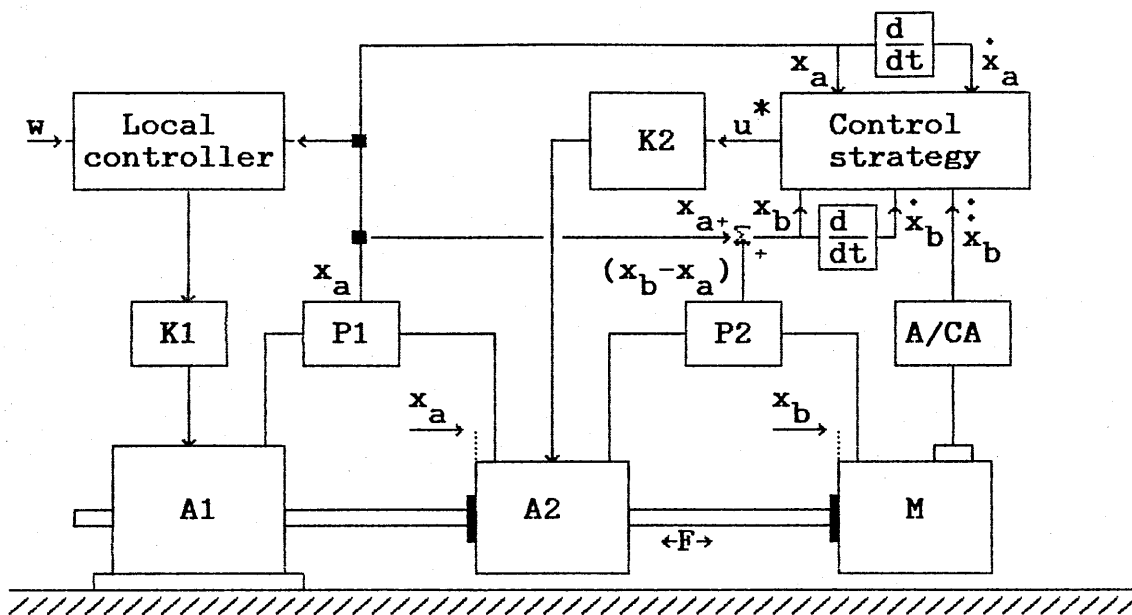


Fig. 7.3a Optimal Control Strategy Using Actual System States

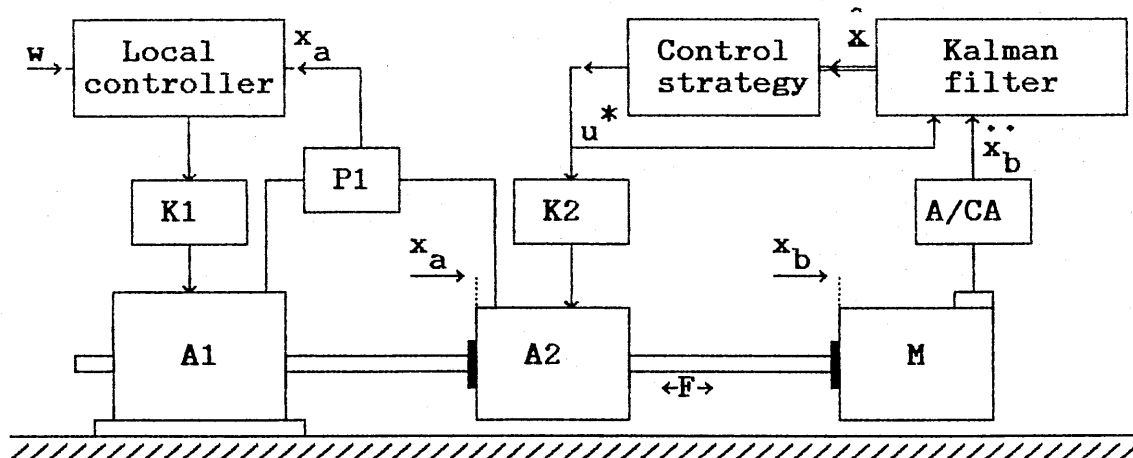


Fig. 7.3b Optimal Control Strategy with Kalman Filter

Section 7.2 describes the identification and modelling of the laboratory system, and Section 7.3 the behaviour of the system under one illustrative straightforward optimal control strategy, Section 7.4 examines some Kalman filter designs, and in Section 7.5 three practical optimal control strategies using the conclusions reached in Sections 7.3, 7.4 are examined. Some overall conclusions are drawn in Section 7.6.

7.2 SYSTEM STATES EQUATIONS:

To represent the laboratory system model in state equation form, the open loop transfer function (OLTF) of the "suspension unit" has to be found first (between $u(t)$ and $(x_b - x_a)$, Fig. 7.1a).

Finding the open loop transfer function directly is impractical, so a temporary closed loop was built as shown in Fig. 7.4.

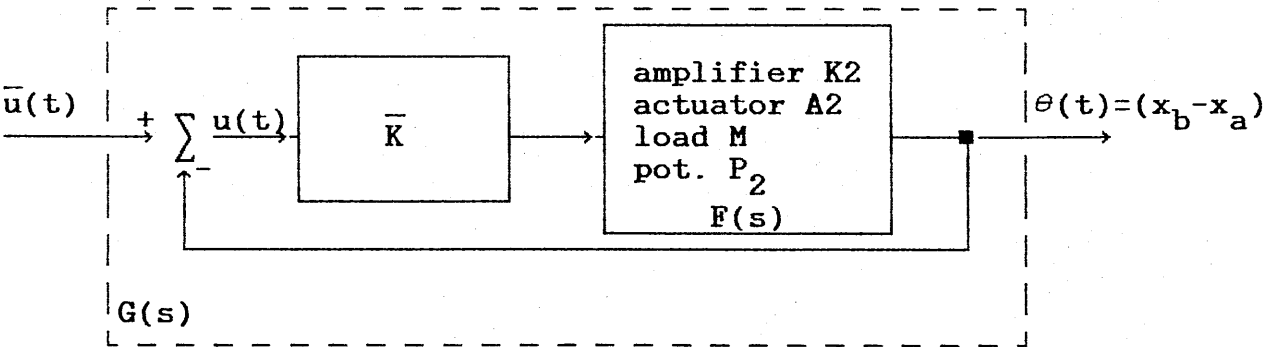


Fig. 7.4 The actuator A2 and load closed-loop.

let:

$F(s) \equiv$ the O.L.T.F.

$G(s) \equiv$ the C.L.T.F.

$\theta(t) = (x_b - x_a)$ (volts) represents the mass position.

$u(t)$ (volts) represents the amplifier input.

$\bar{u}(t)$ (volts) represents the temporary closed loop input.

$F(s)$ can then be found from $G(s)$ as [3];

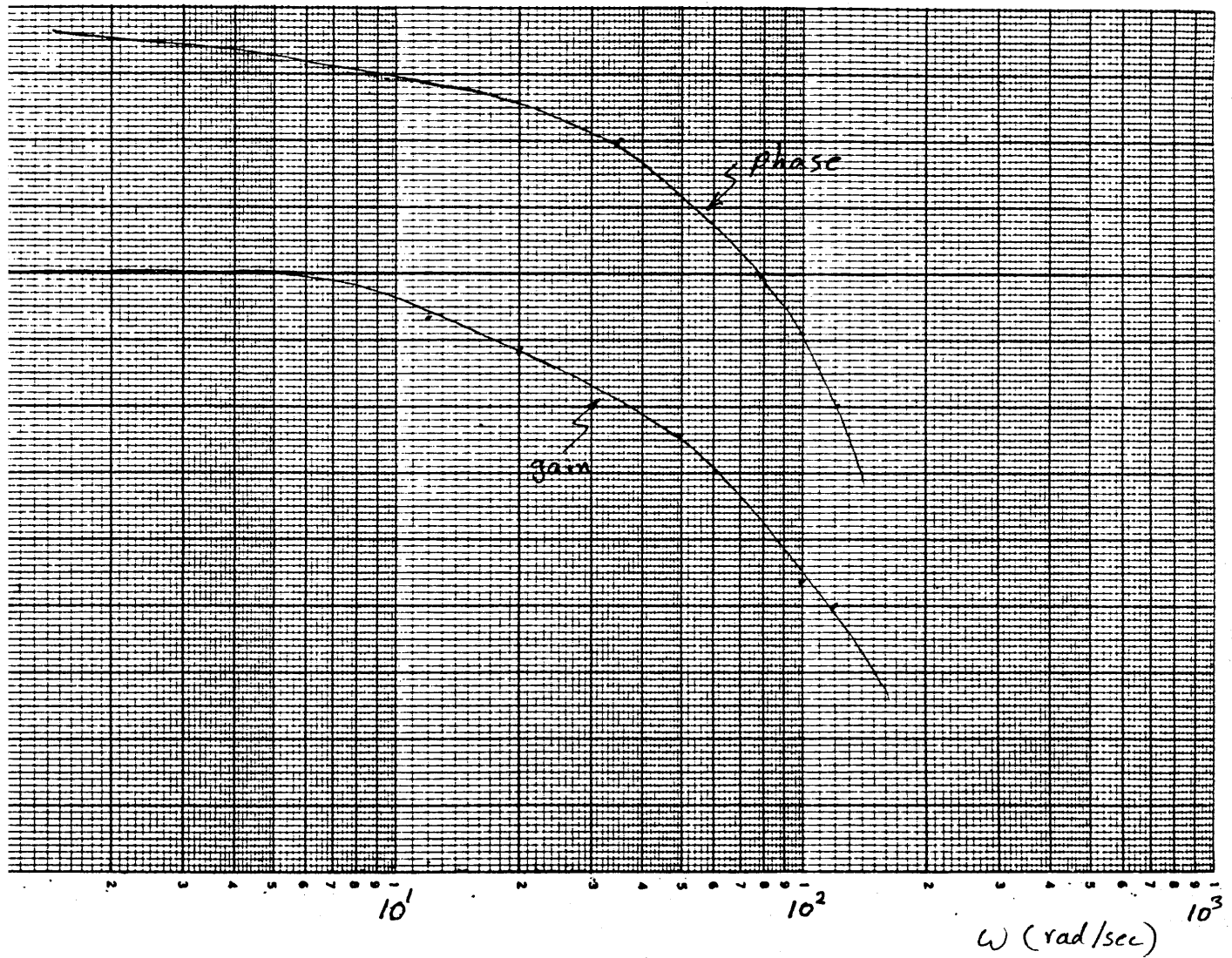
$$F(s) = \frac{1}{\bar{K}} \frac{G(s)}{1 - G(s)} \quad \dots (7.1)$$

where $\bar{K} = 0.2$, and K_2 was set to $K_2 = 10$ (Fig 7.4), after some experimentation to find a reasonably well behaved temporary closed loop.

Using the Transfer Function Analyzer (TFA) and setting $x_a = 0$ for the purpose of the experiment, the Bode plot shown in Fig. 7.5 was developed.

With the help of the CAD package on the PDP11 computer, the Bode plot for a third order model which has (nearly) similar phase and gain characteristics is shown in (Fig. 7.6).

Fig. 7.5 The System Bode Plot



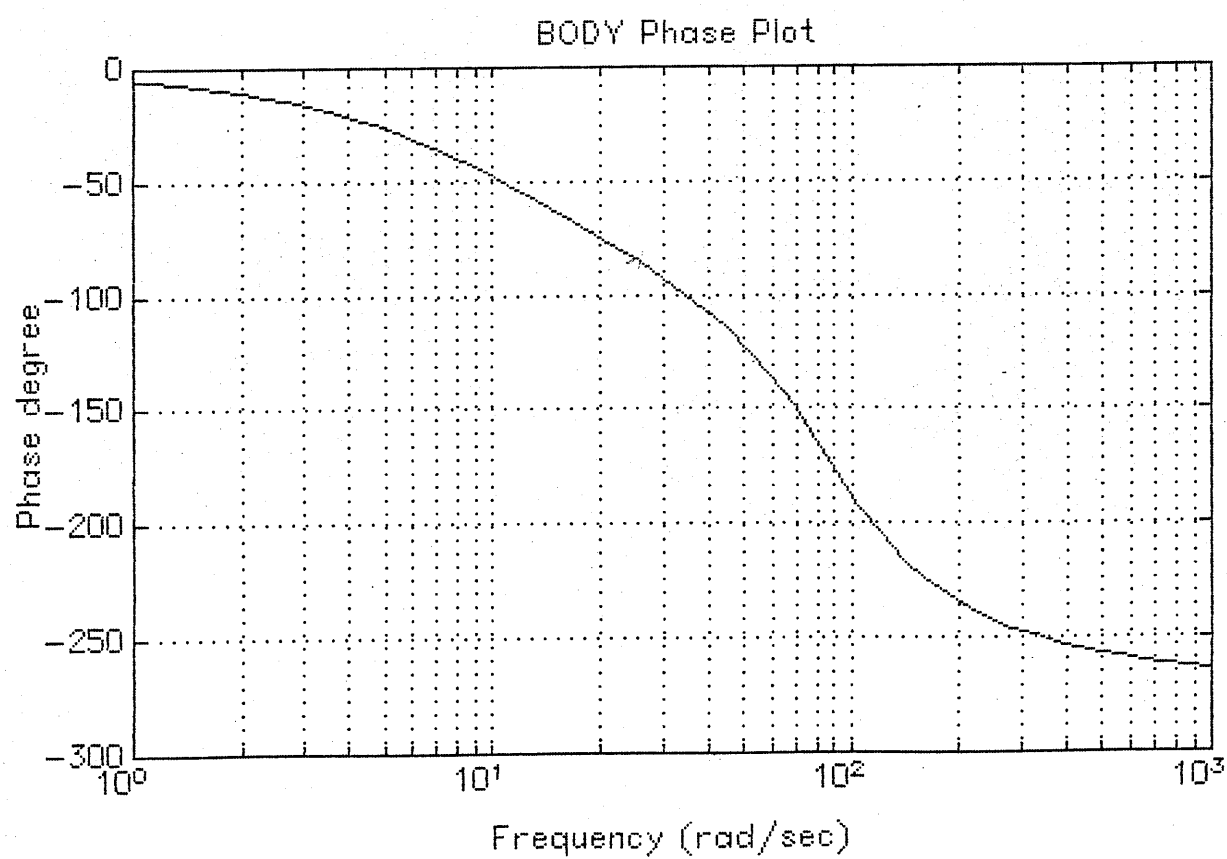
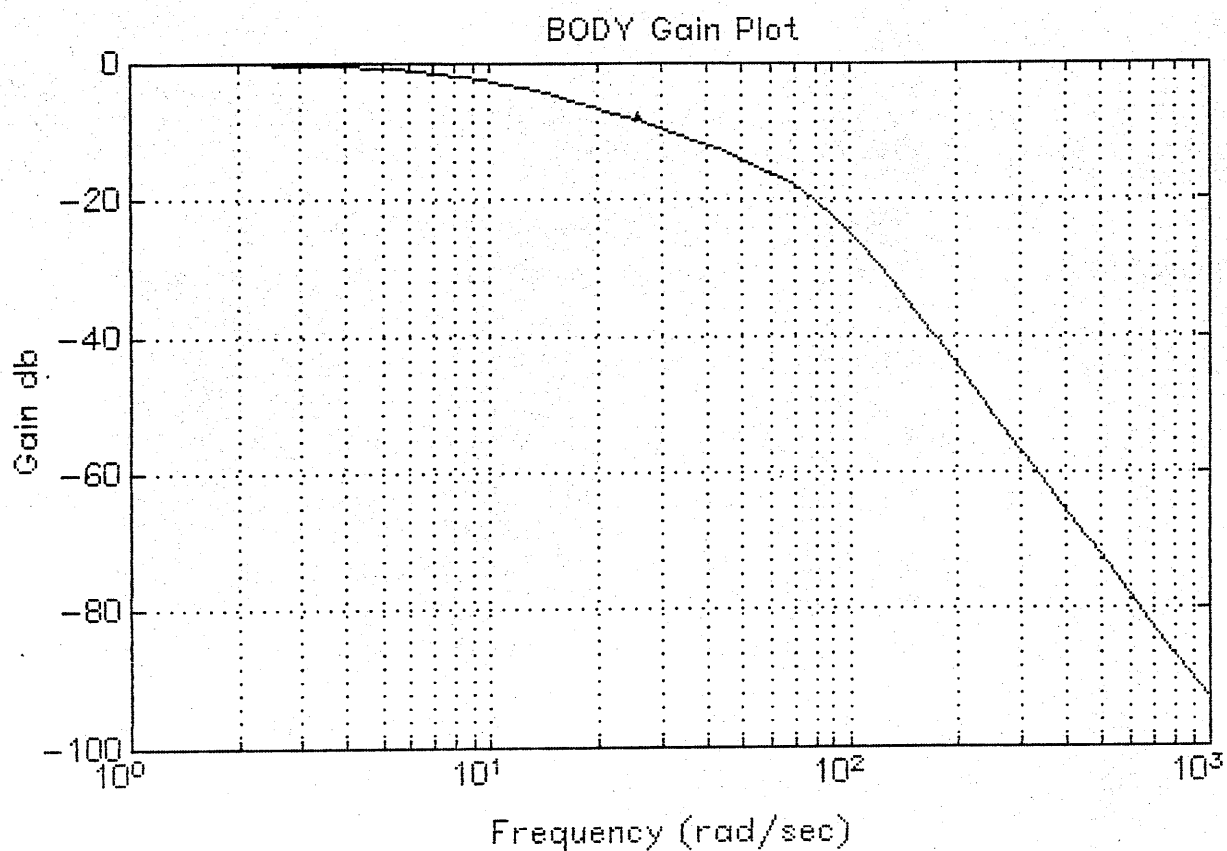
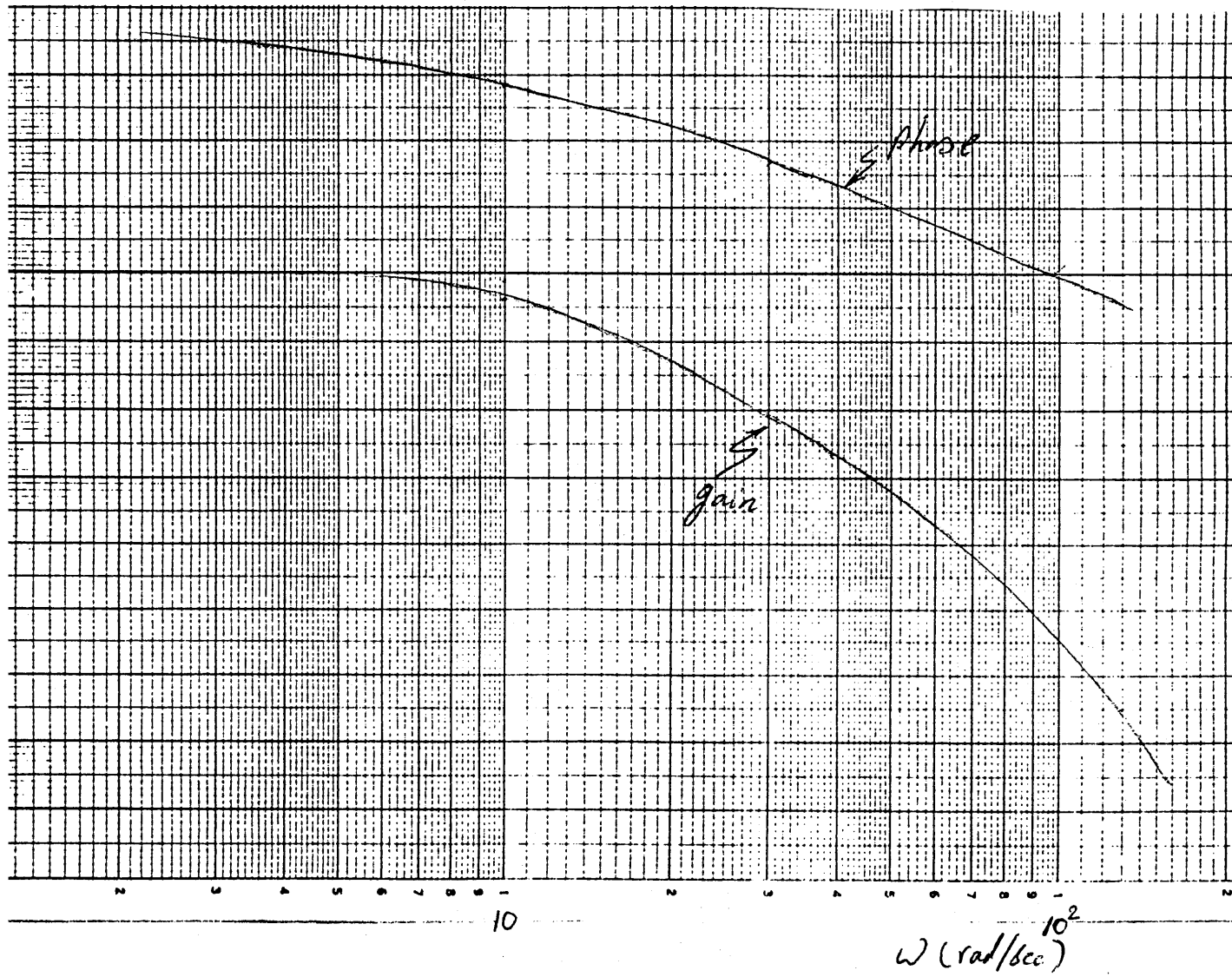


Fig. 7.6 The CAD Bode Plot of the System

Fig. 7.7 The Axle Bode Plot



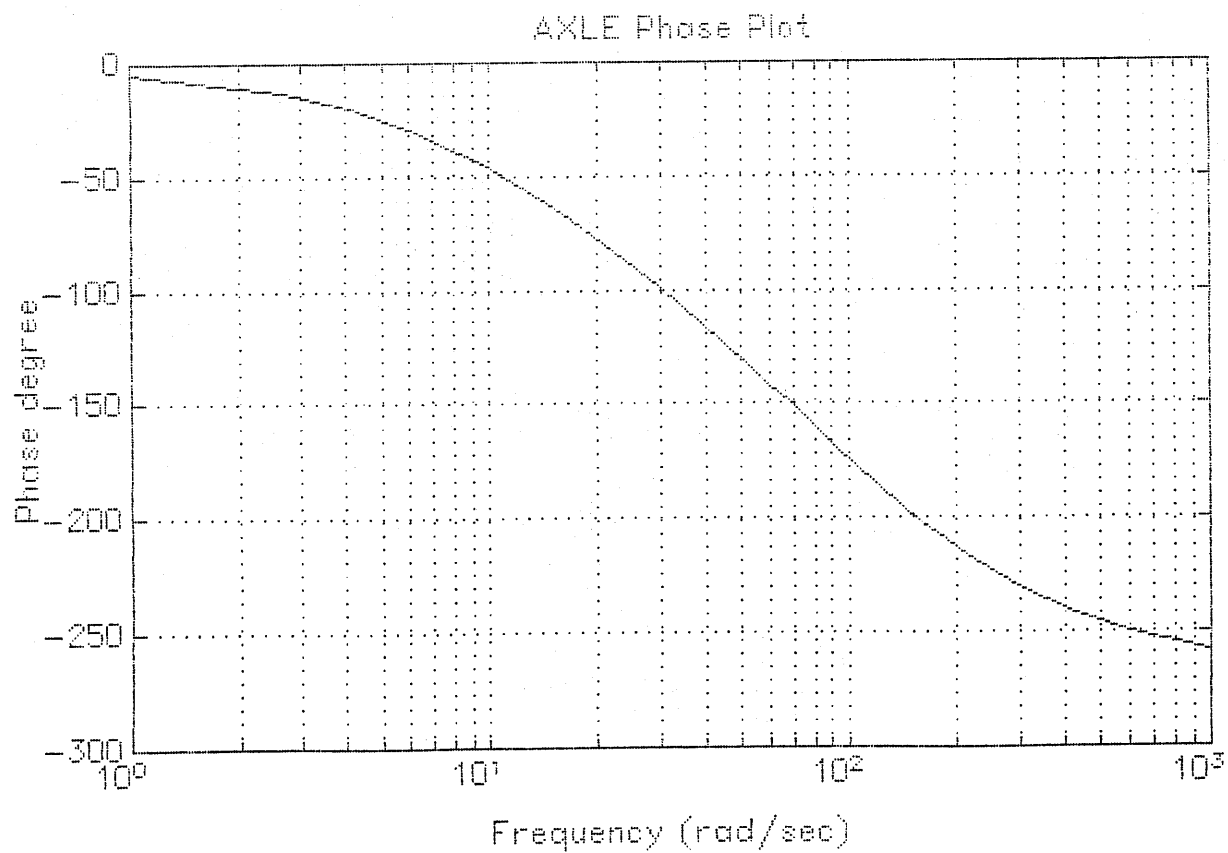
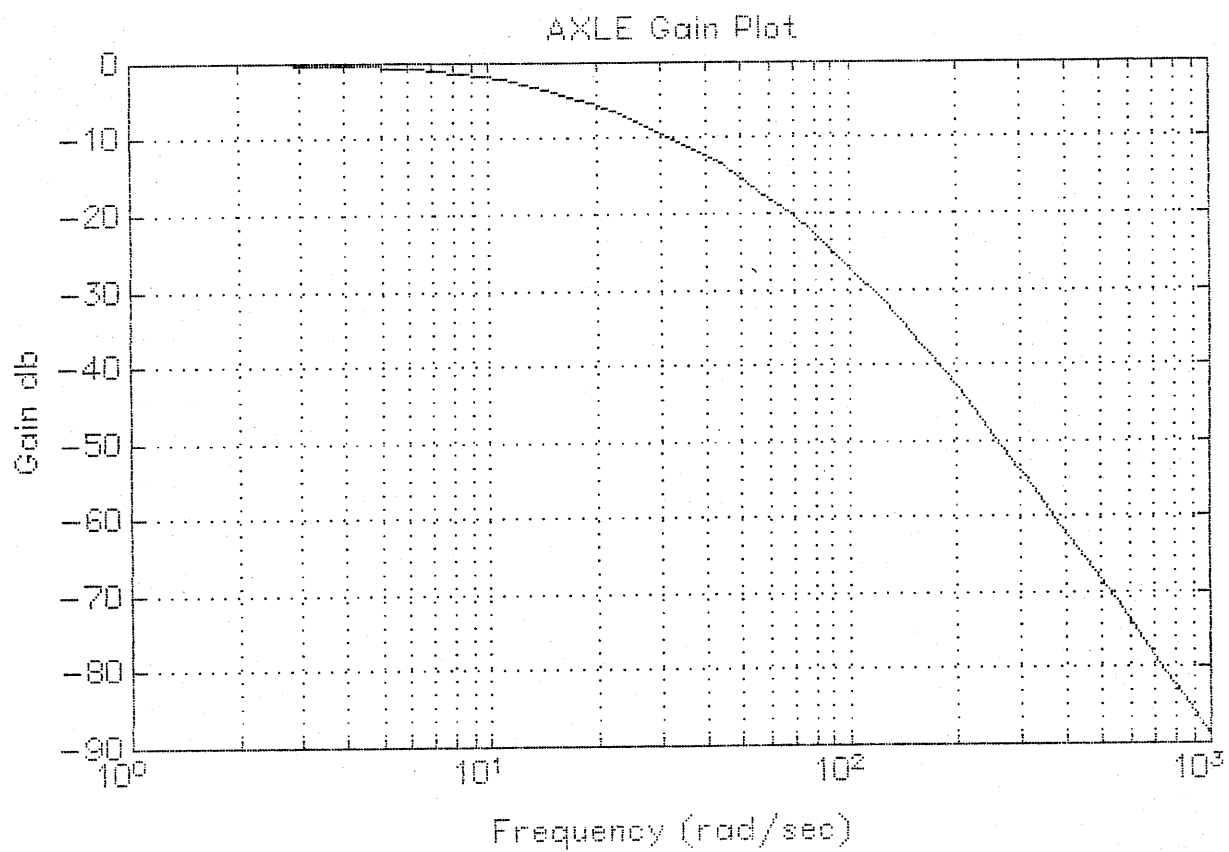


Fig. 7.8 CAD Bode Plot of the Axle

In this case, the transfer function (in Fig 7.6) is:

$$G(s) = \frac{8.88 \times 10^4}{(s+12)(s^2+100s+7400)} \quad \dots (7.2)$$

and this is taken to represent the temporary closed loop.

From Eqn. (7.1), the OLTF is:

$$F(s) = \frac{\theta(s)}{U(s)} = \frac{10}{2} \frac{8.88 \times 10^4}{s^3 + 112s^2 + 8600s} = \frac{4.44 \times 10^5}{s(s+56 \pm j73.9)} \quad \dots (7.3)$$

where $\theta(s) = \mathcal{L}\{(x_b - x_a)\}$, $x_a = 0$

$U(s) = \mathcal{L}\{u(t)\}$

Eqn. 7.3 can be written in differential equation form as:

$$\ddot{\theta} + 112\dot{\theta} + 8600\theta = 4.44 \times 10^5 u \quad \dots (7.4)$$

let

$$\left. \begin{aligned} x_1 &= \theta \\ x_2 &= 0.1\dot{x}_1 = 0.1\dot{\theta} \quad \text{or } \dot{x}_1 = 10x_2 \\ x_3 &= 0.1\dot{x}_2 = 0.01\ddot{\theta} \quad \text{or } \dot{x}_2 = 10x_3 \end{aligned} \right\} \quad \dots (7.5)$$

Substituting Eqns (7.5) in (7.4);

$$\frac{\dot{x}_3}{0.01} + \frac{112x_3}{0.01} + \frac{8600x_2}{0.1} = 4.44 \times 10^5 u$$

i.e.

$$\dot{x}_3 = 4440u - 860x_2 - 112x_3$$

The state equation model is therefore;

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ 0 & -860 & -112 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 4440 \end{bmatrix} u(t) \quad \dots (7.6)a$$

and

$$\mathbf{y}(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t) \quad \dots (7.6)b$$

Eqns (7.5) models the laboratory equipment behaviour if the actuator A1 ram is stationary. The first task is to extend the model to accommodate movement of this ram.

The transfer function of the closed loop actuator A1 following the same procedure described above using the TFA (Figs. 7.7 and 7.8 show the Bode plots and CAD results) is:

$$\frac{X_a(s)}{W(s)} = \frac{136500}{(s+15)(s+70)(s+130)}$$

which can be approximated by a second order transfer function as:

$$\frac{X_a(s)}{W(s)} = \frac{1050}{(s+15)(s+70)} = \frac{1050}{s^2 + 85s + 1050} \quad \dots (7.7)$$

i.e.

$$\ddot{x}_a + 85\dot{x}_a + 1050x_a = 1050w \quad \dots (7.8)$$

This differs in structure from Eqn (6.1b) (which it supposed to represent) in various ways, of which the most significant is that x_a is not affected by the control input, $u(t)$ ("f" in Eqns 6.1). Clearly, circuitry could be built to accommodate these differences, but it was felt that since Eqn 7.8 represents generally harsher unsprung mass behaviour than Eqn 6.1b, such complications were perhaps best avoided, at least in the first place.

Eqn. 7.4 can now be rewritten to take account of A1 as;

$$\ddot{x}_b + 112(\ddot{x}_b - \ddot{x}_a) + 8600(\dot{x}_b - \dot{x}_a) = 4.44 \times 10^5 u \quad \dots (7.9)$$

This models the behaviour of the hydraulic system, but not of course of the suspension spring modeled in Eqn 6.1(a).

let

$$\left. \begin{aligned} x_4 &= x_a \\ x_5 &= 0.1\dot{x}_a = 0.1\dot{x}_4 \quad \text{or} \quad \dot{x}_4 = 10x_5 \end{aligned} \right\} \dots (7.10)$$

From (7.9), using (7.5) and (7.10);

$$100\dot{x}_3 + 112(100x_3 - \dot{x}_a) + 8600(10x_2 - 10x_5) = 4.44 \times 10^5 u \quad (7.11)$$

from (7.8) and using (7.10);

$$\dot{x}_a = 10\dot{x}_5 = 10(105w - 105x_4 - 85x_5) \dots (7.12)$$

Substituting Eqn (7.12) in (7.11);

$$\dot{x}_3 + 112x_3 - 1176w + 1176x_4 + 952x_5 + 860x_2 - 860x_5 = 4440u$$

i.e.

$$\dot{x}_3 = -860x_2 - 112x_3 - 1176x_4 - 92x_5 + 4440u + 1176w \dots (7.13)$$

The extended model becomes;

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & -860 & -112 & -1176 & -92 \\ 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & -105 & -85 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 4440 \\ 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1176 \\ 0 \\ 105 \end{bmatrix} w(t) \dots (7.14a)$$

and

$$\mathbf{y}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \mathbf{v}(t) \dots (7.14b)$$

where;

$w(t)$ (volts) is band limited white noise of variance chosen to represent random road displacement.

$v(t)$ (volts) is band limited white noise of variance chosen to represent the accelerometer/ charge amplifier noise.

$y(t)$ (volts) is the suitably scaled output of the accelerometer charge amplifier.

The eigen values of this system are:

0.0

$-56.0 \pm j73.9$

-15.0

-70.0

On the laboratory model, the following states are directly available for measurement (Fig. 7.3a):

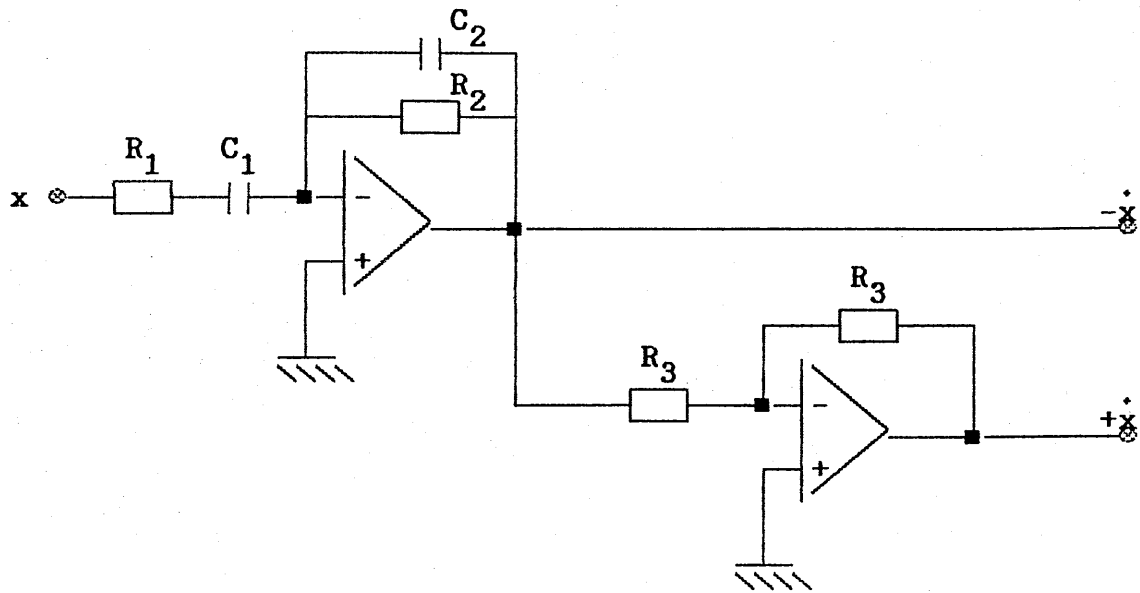
x_1 ; (x_b) as shown in Fig. 7.2.

x_3 ; $(\ddot{x}_b)$ the scaled accelerometer/charge amplifier output.

x_4 ; (x_a) as shown in Fig. 7.2.

$(x_1 - x_4)$; $(x_b - x_a)$ as shown in Fig. 7.2.

The sprung mass velocity state x_2 is indirectly generated by differentiating and filtering the position state x_1 , as indicated in Fig. 7.9. x_5 is generated similarly from x_4 , where necessary.



where

$$R_1 = 10K\Omega$$

$$R_2 = 1M\Omega$$

$$R_3 = 100K\Omega$$

$$C_1 = 0.1\mu F$$

$$C_2 = 0.01\mu F$$

Fig. 7.9 Generation of Velocity State from Position.

7.3 SYSTEM OPTIMAL CONTROL:

In this work the underlying active suspension problem is treated from the view point of optimal control theory [4], which involves state feedback as indicated in block diagram form in Fig. 7.10.

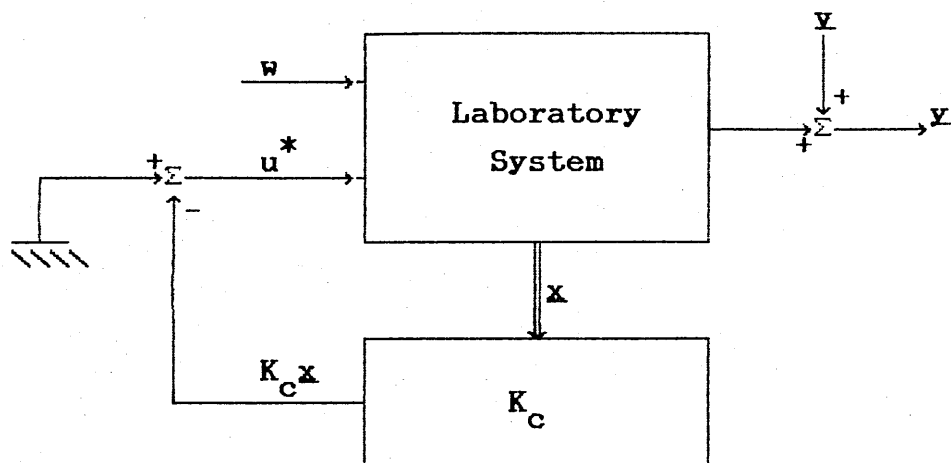


Fig. 7.10 Optimal State Feedback

The performance index to be minimized for the derivation of optimal control u^* is assumed to be of a form (6.7) which can be re-written as;

$$J = E \left[\int_0^{T_f} \left[x \bar{Q}_c x^T + u \bar{R}_c u^T \right] dt \right] \quad \dots (7.15)$$

in which $\bar{Q}_c$ and $\bar{R}_c$ are chosen to give some desirable behaviour, and K_c is then calculated by solving a Riccati equation (as discussed in Chapter 6, Section 6.4).

The term " T_f " in the cost function (Eqn 7.15) is taken to be that at which the natural dynamics of a single disturbance - say a step function - have died away. This is less than 500ms on this equipment.

To derive an optimal control for any system, the system must be controllable¹. Testing the controllability of the system described

(¹) A system is controllable if all the modes can be excited, or controlled by the input. For a system to be controllable, the rank of:

$$\begin{bmatrix} B & AB & A^2B & \dots & A^{m-1}B \end{bmatrix} = m, \text{ where } m \text{ is the number of states.}$$

by Eqns. 7.14, it was found uncontrollable reflecting a difference between the laboratory equipment (Eqns 7.8, and 7.9) and the standard quarter car model (Eqns 6.1) discussed earlier.

However, on closer inspection, it was found that x_4 and x_5 (i.e. the unsprung mass states) are the uncontrollable states and these are not of major interest in our control strategy which must concentrate on x_1 , x_2 & x_3 (the sprung mass states). These are controllable. Since no attempt here is made to control x_4 and x_5 , $\bar{Q}_C$ is chosen accordingly, to influence x_1 , x_2 and x_3 .

We consider, for example, the solution of the Riccati Equation (6.10) minimizing the cost function (7.15) for $\bar{Q}_C$ and $\bar{R}_C$ as:

$$\bar{Q}_C = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{R}_C = \begin{bmatrix} 100 \end{bmatrix} \quad \dots(7.16)$$

The corresponding control strategy will try to minimize the performance index;

$$E \int_0^{T_f} \left(10 \cdot x_1^2 + 1 \cdot x_2^2 + 1 \cdot x_3^2 + 0 \cdot x_4^2 + 0 \cdot x_5^2 + 100u^2 \right) dt,$$

attempting particularly to reduce the position variance $E(x_1^2)$.

The optimal input $u^*(t)$ is limited by the choice of $\bar{R}_C$ corresponding to Eqn. (7.15), where it can be seen that increasing $\bar{R}_C$ reduces u^* and vice versa. This is chosen so that u^* stays within the operating limits of the amplifiers, hydraulic valves and hydraulic power supply.

It should also be observed at this stage that the optimal strategy takes no account of the requirement that the mean sprung mass position must lie at the mean value of $(x_1 - x_4)$ - not at zero with respect to the frame of reference (i.e. $x_1 = 0$) - so that the suspension system can recover when the vehicle goes up a hill, or is disturbed by a "step", etc. This topic is ignored in this section, but is taken into account in Section 7.5.

Using (7.16), the program ARE1 on the VAX computer gives:

$$K_c = \begin{bmatrix} 0.316 \\ 0.145 \\ 0.081 \\ -0.179 \\ -0.015 \end{bmatrix}, \quad \dots (7.17)a$$

The eigen values of this closed loop system are:

-16.8 ± j6.2

-438.2

-15.0

-70.0

Step function responses using this strategy are shown in Figs. 7.11 to 7.13.

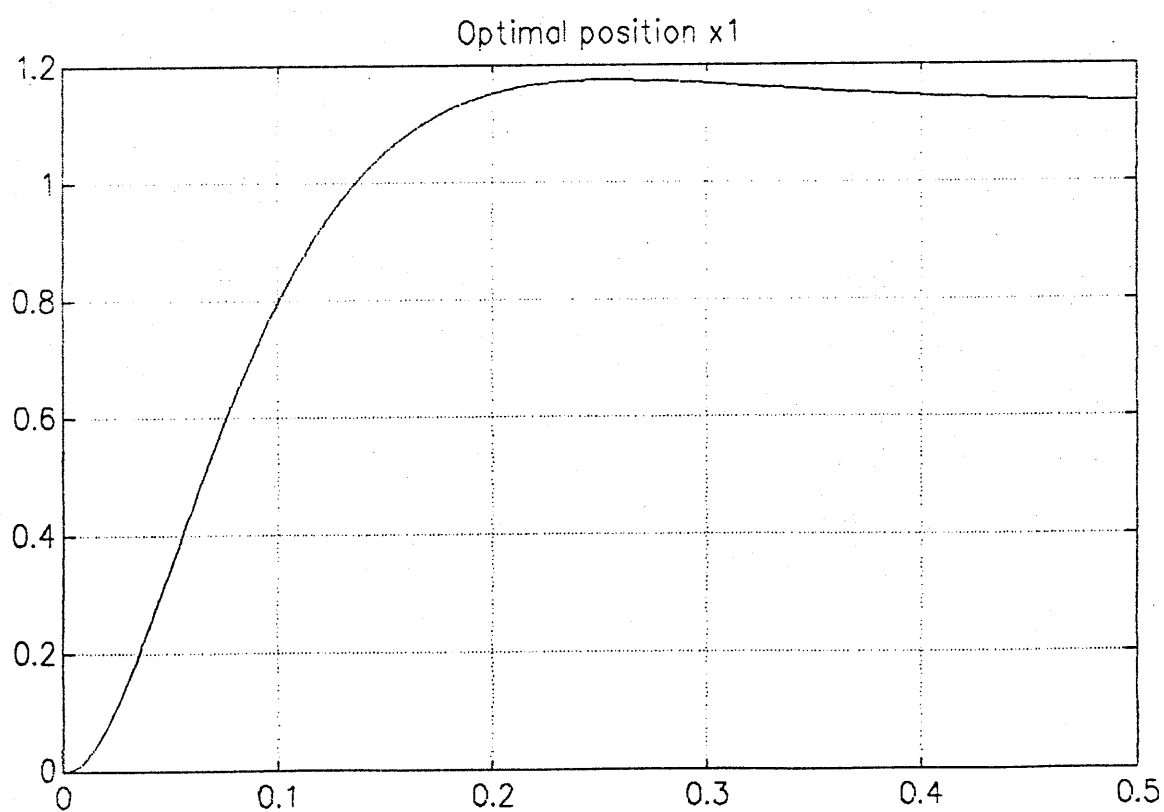
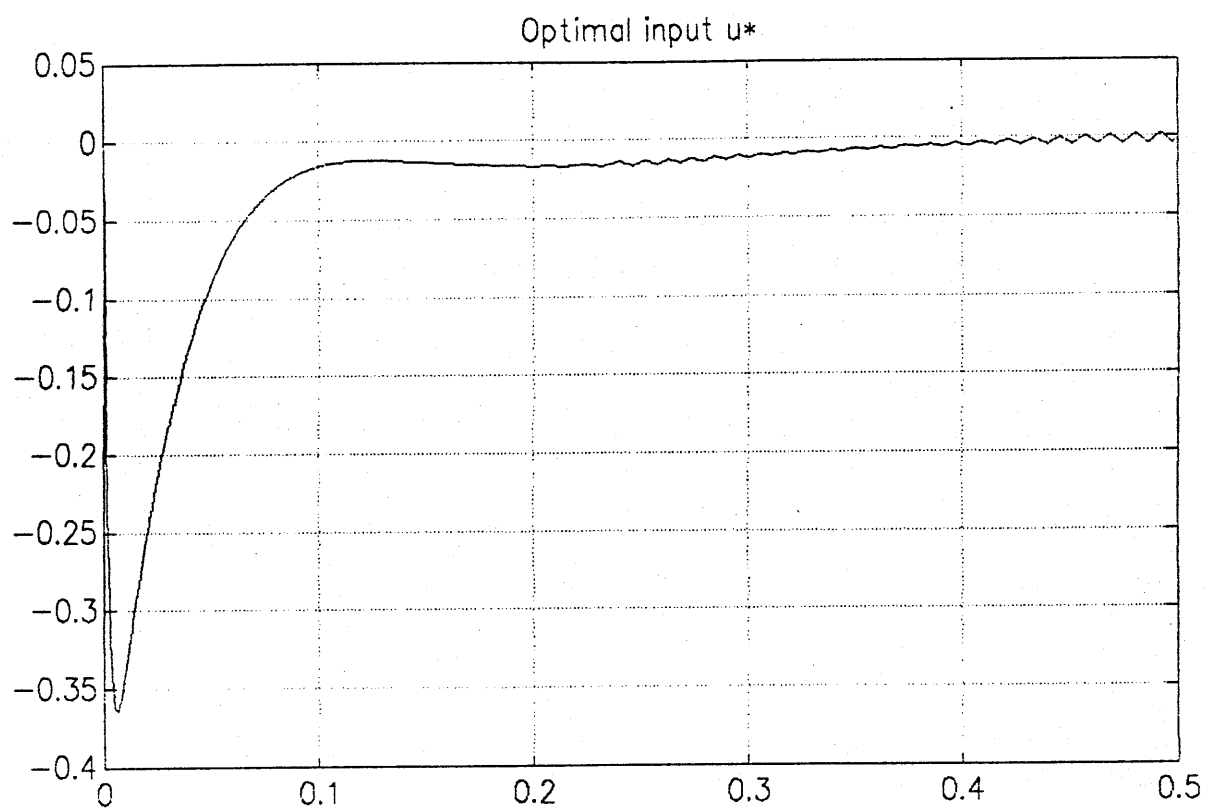


Fig. 7.11 Step Response Simulation (u^* and x_1) Using (7.17a)

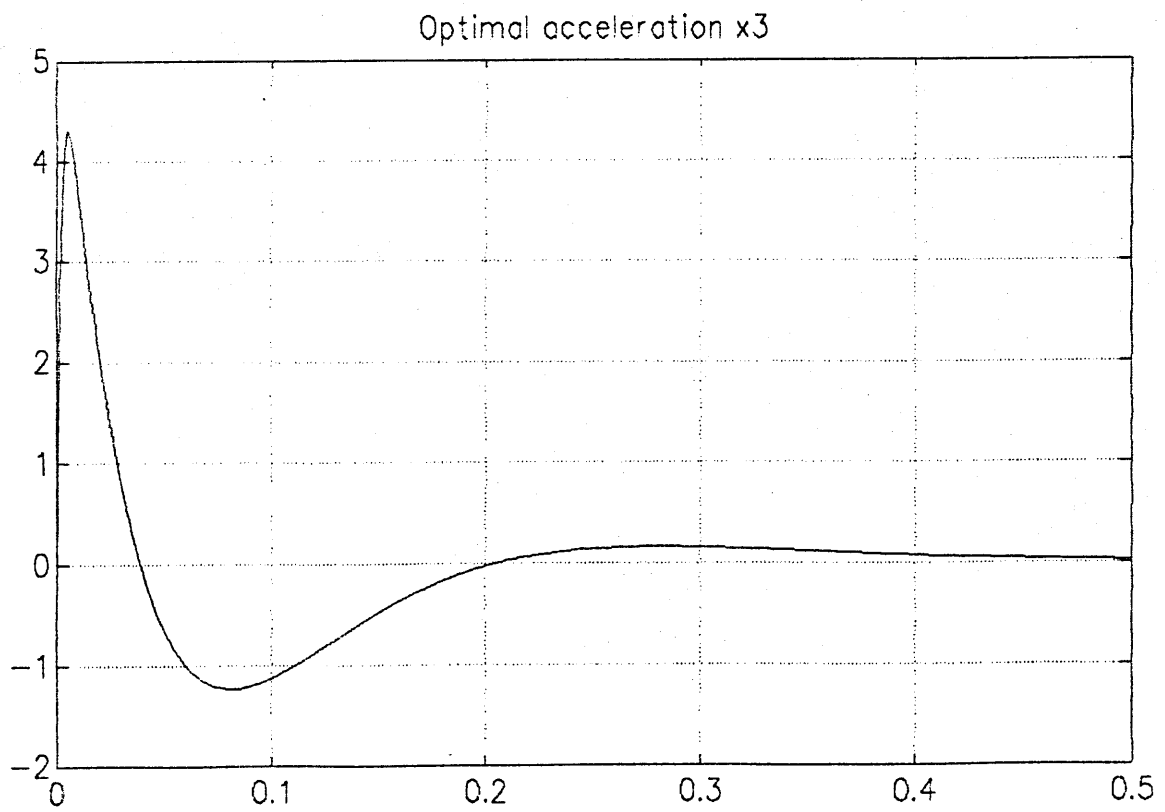
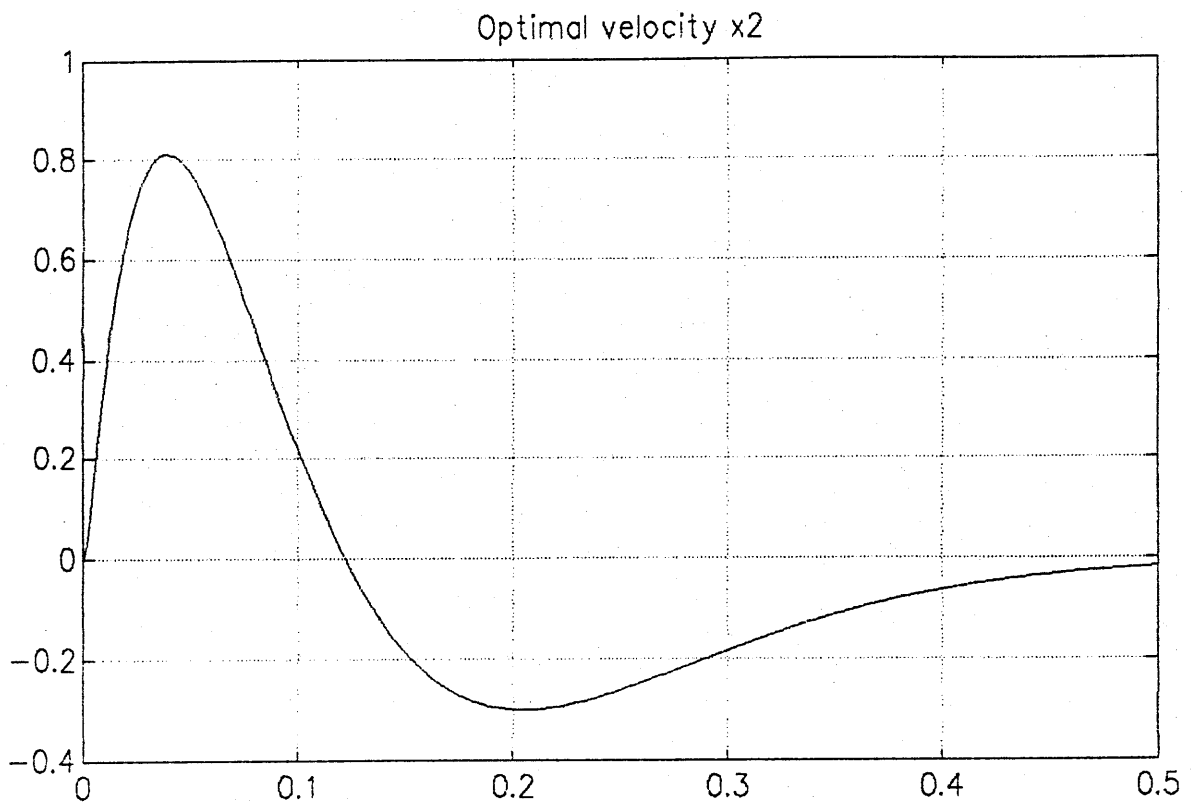


Fig. 7.12 Step Response Simulation (x_2 and x_3) Using (7.17a)

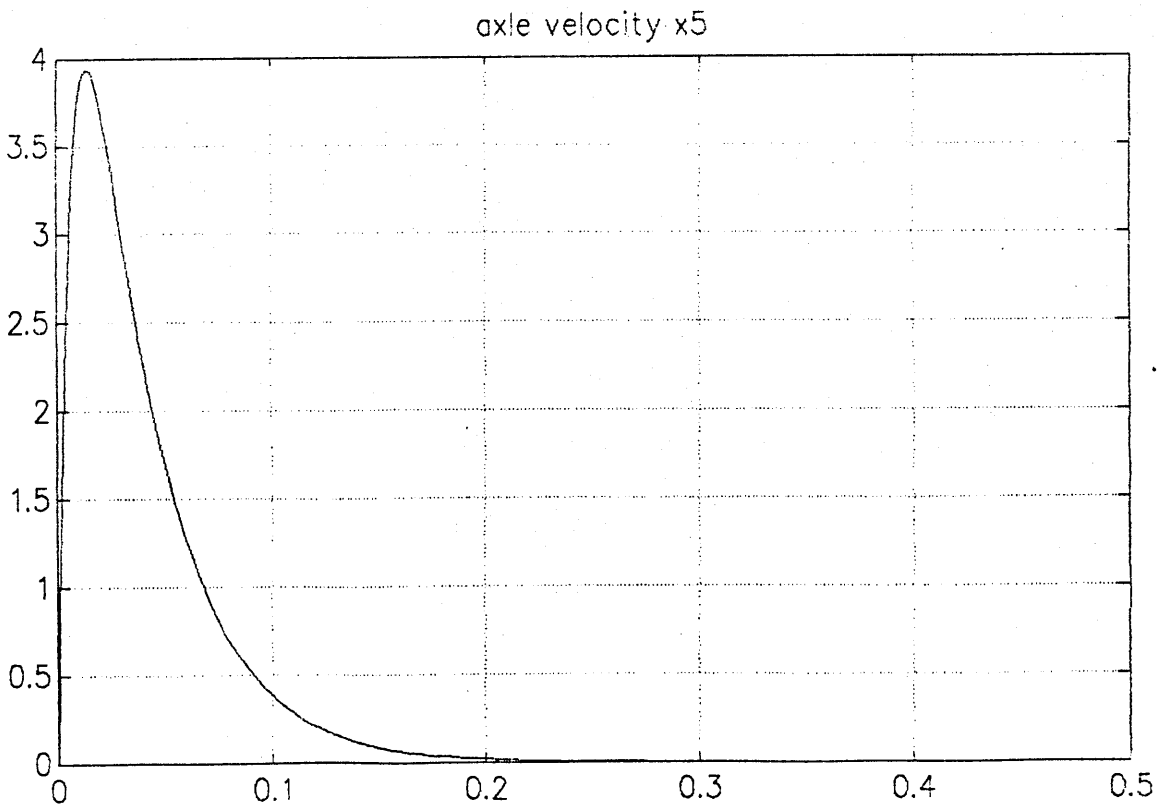
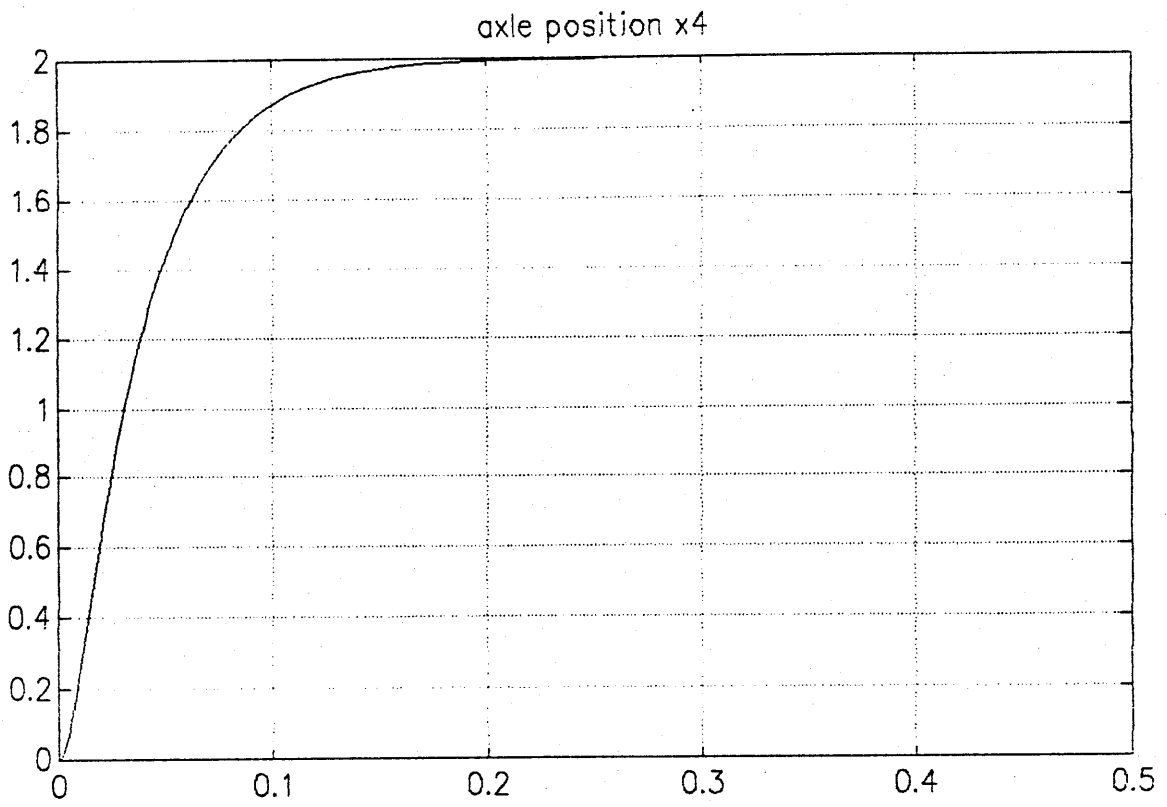


Fig. 7.13 Step Response Simulation (x_4 and x_5) Using (7.17a)

It is of interest here to consider the strategy;

$$K_c = \begin{bmatrix} 0.316 \\ 0.145 \\ 0.081 \\ 0.0 \\ 0.0 \end{bmatrix}, \quad \dots (7.17)b$$

The closed loop system in this case has exactly the same eigen values as that using the feedback of Eqn 7.17a. Examination shows that this is due to the structure of the open loop model (Eqn 7.14) which is such that any feedback of x_4 , x_5 has no effect on the closed loop system eigen values. This could be taken to mean that there is no great advantage in feeding back these states, but this will be examined in more detail in Sect 7.5.

Figs. 7.14, 7.15 & 7.16 show computer simulations for this system's response to a step input ($w=2$ volt).

These show dynamics which are very similar to those in Fig 7.11 to 7.13. This is natural since the eigen values of the system are the same, but detailed differences appear in the dc level of x_1 , and in the shape of u^* .

Applying these control strategies to the equipment, Fig. 7.17 shows the laboratory system step responses using Eqn 7.17a, and Fig 7.18 using Eqn 7.17b. It can be seen that these match the computer simulations fairly well, including the dc content of x_1 in Figs. 7.11 and 7.18. Generally, the experimental system results are less damped than the computer simulations, owing, perhaps, to non-linearities and unmodeled system noise.

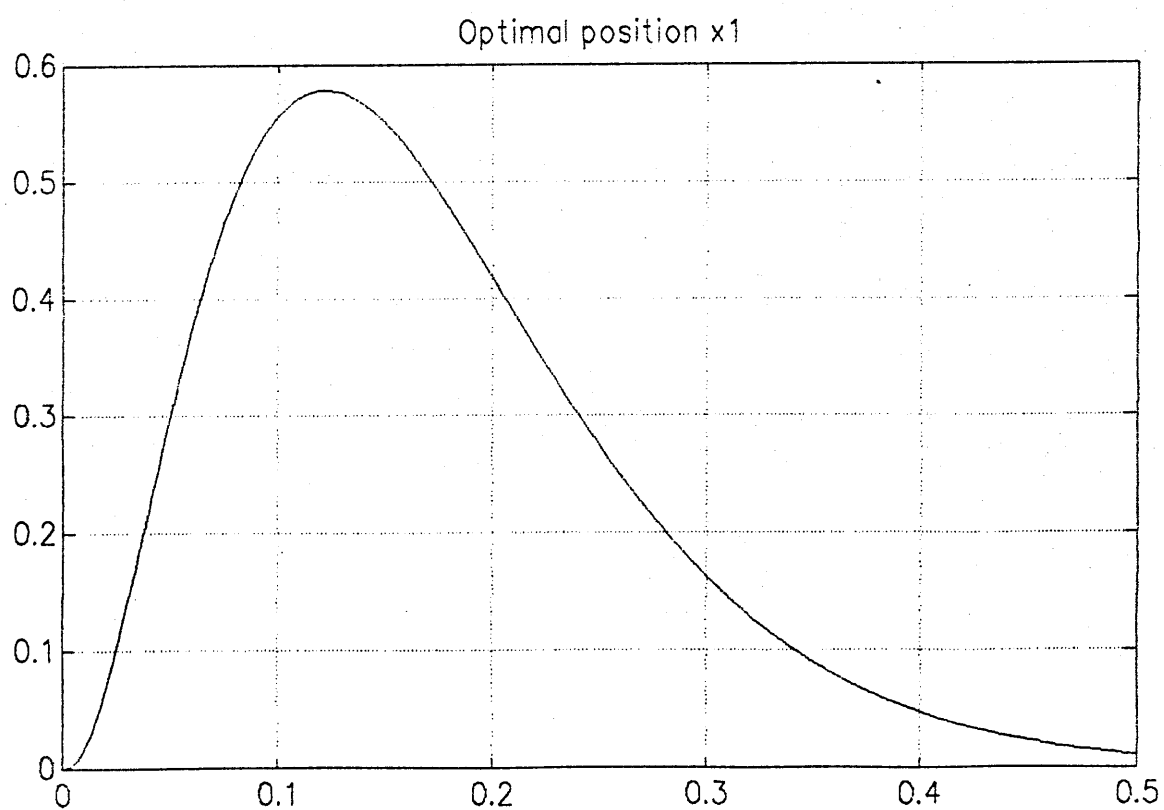
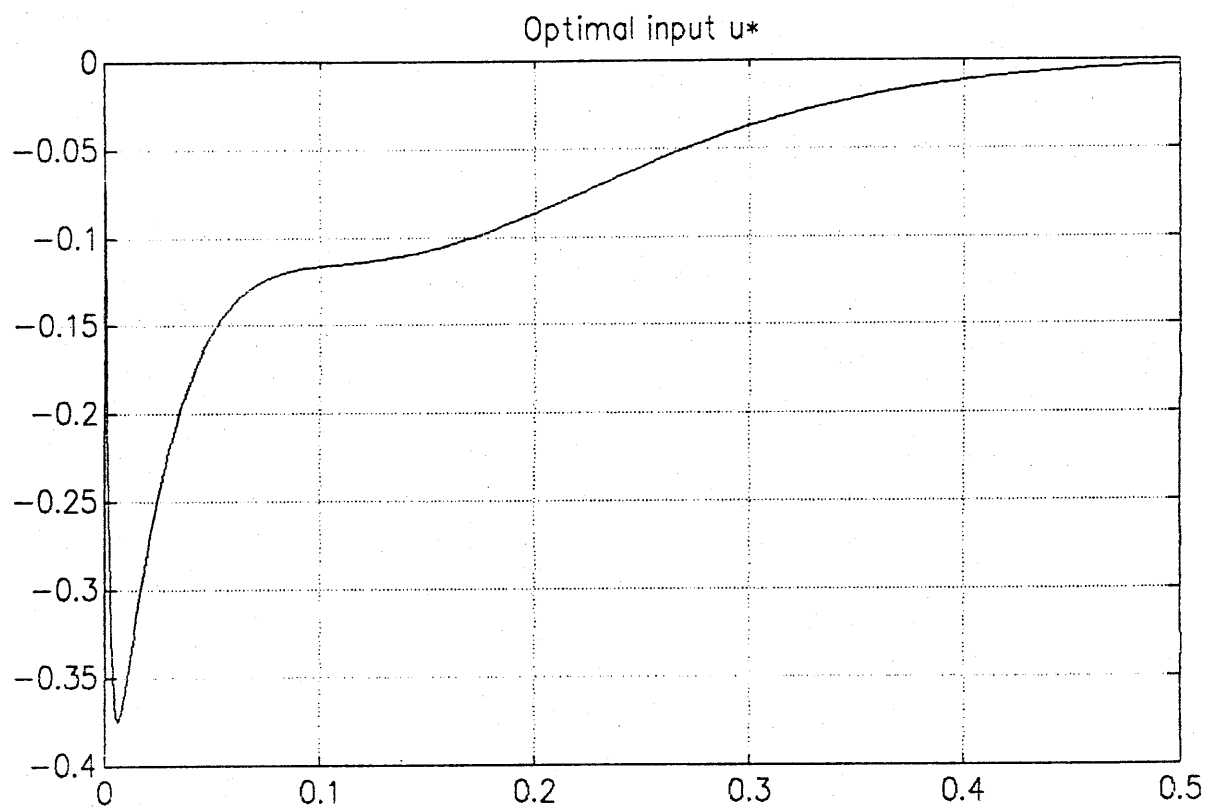


Fig. 7.14 Step Response Simulation (u^* and x_1) Using (7.17b)

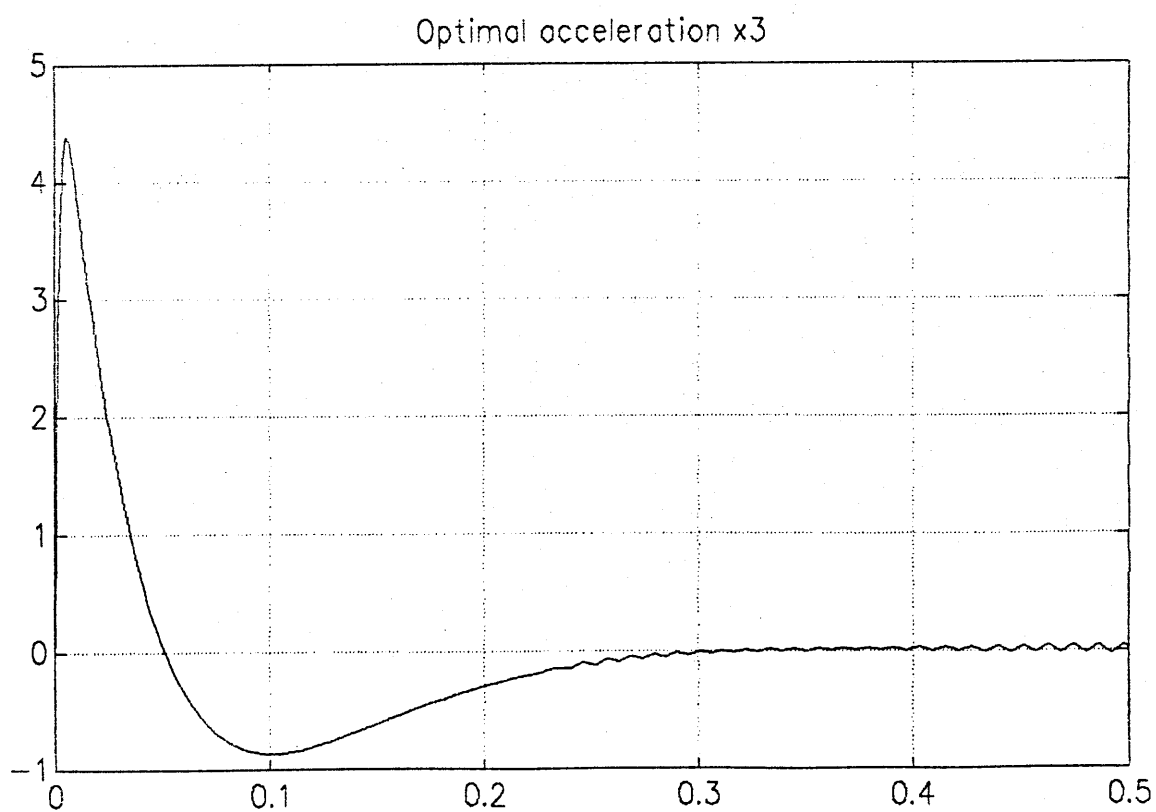
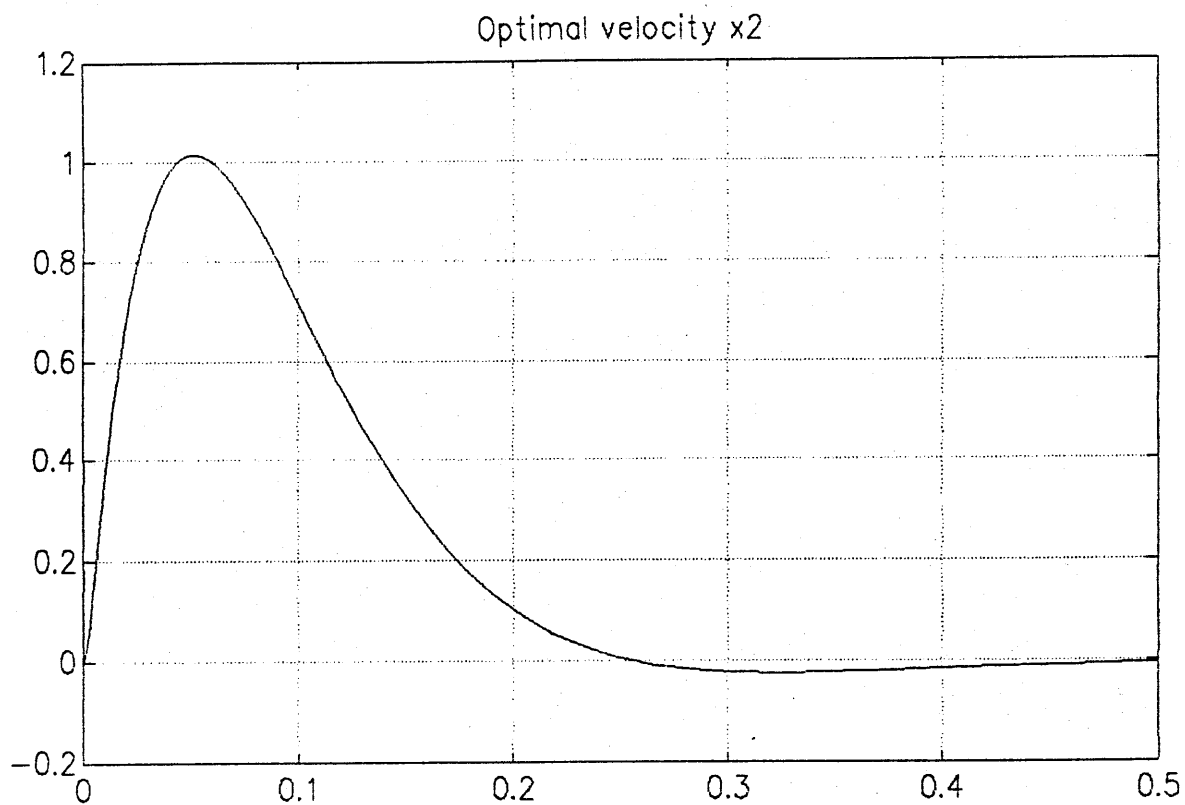


Fig. 7.15 Step Response Simulation (x_2 and x_3) Using (7.17b)

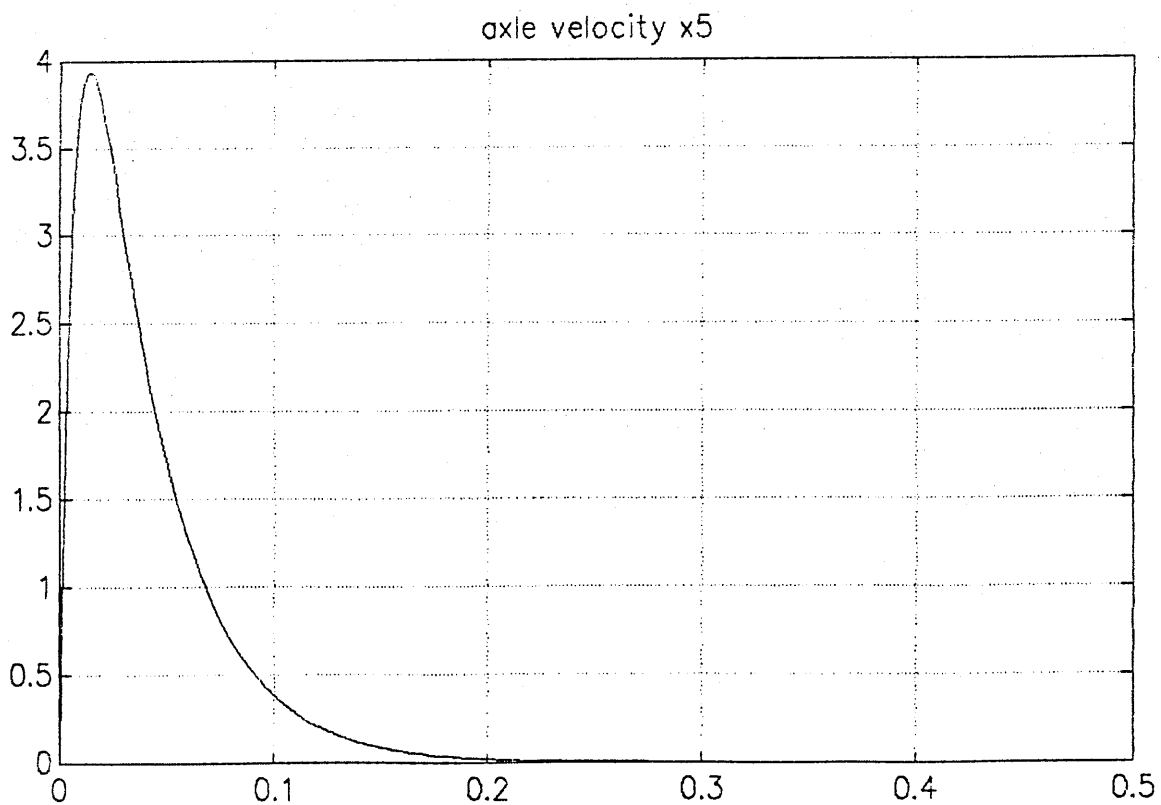
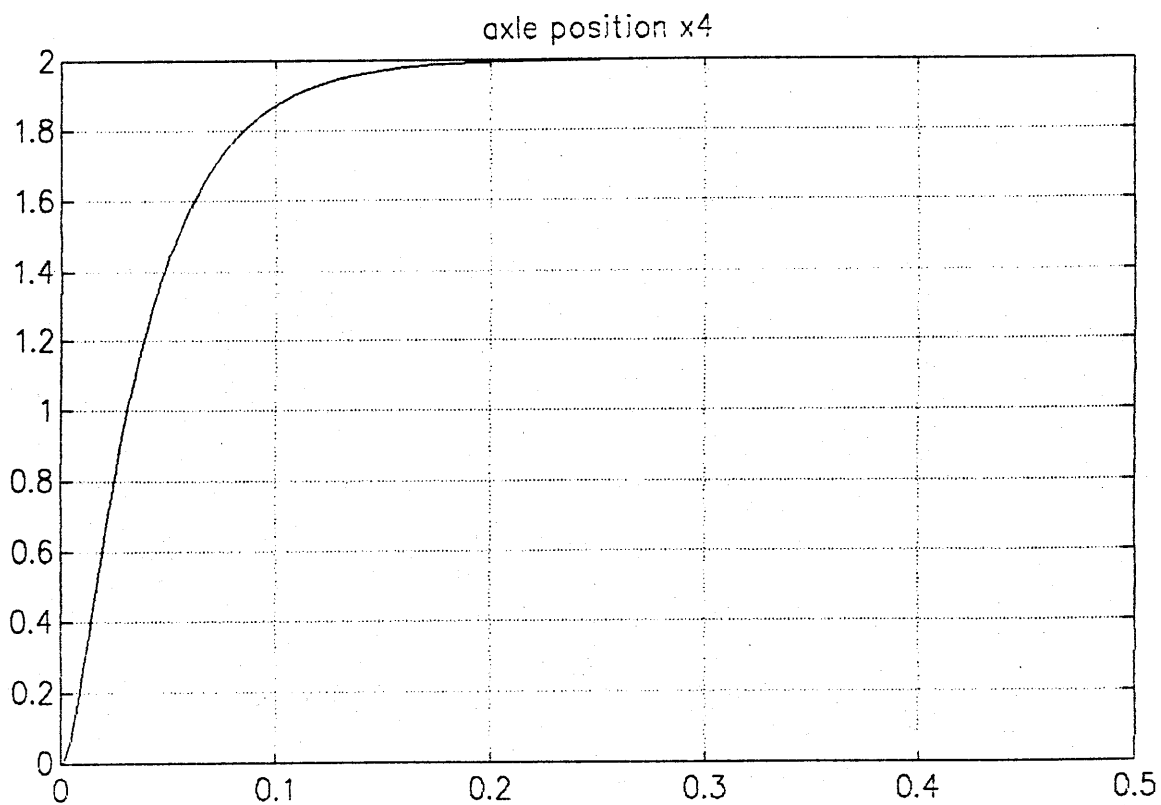


Fig. 7.16 Step Response Simulation (x_4 and x_5) Using (7.17b)

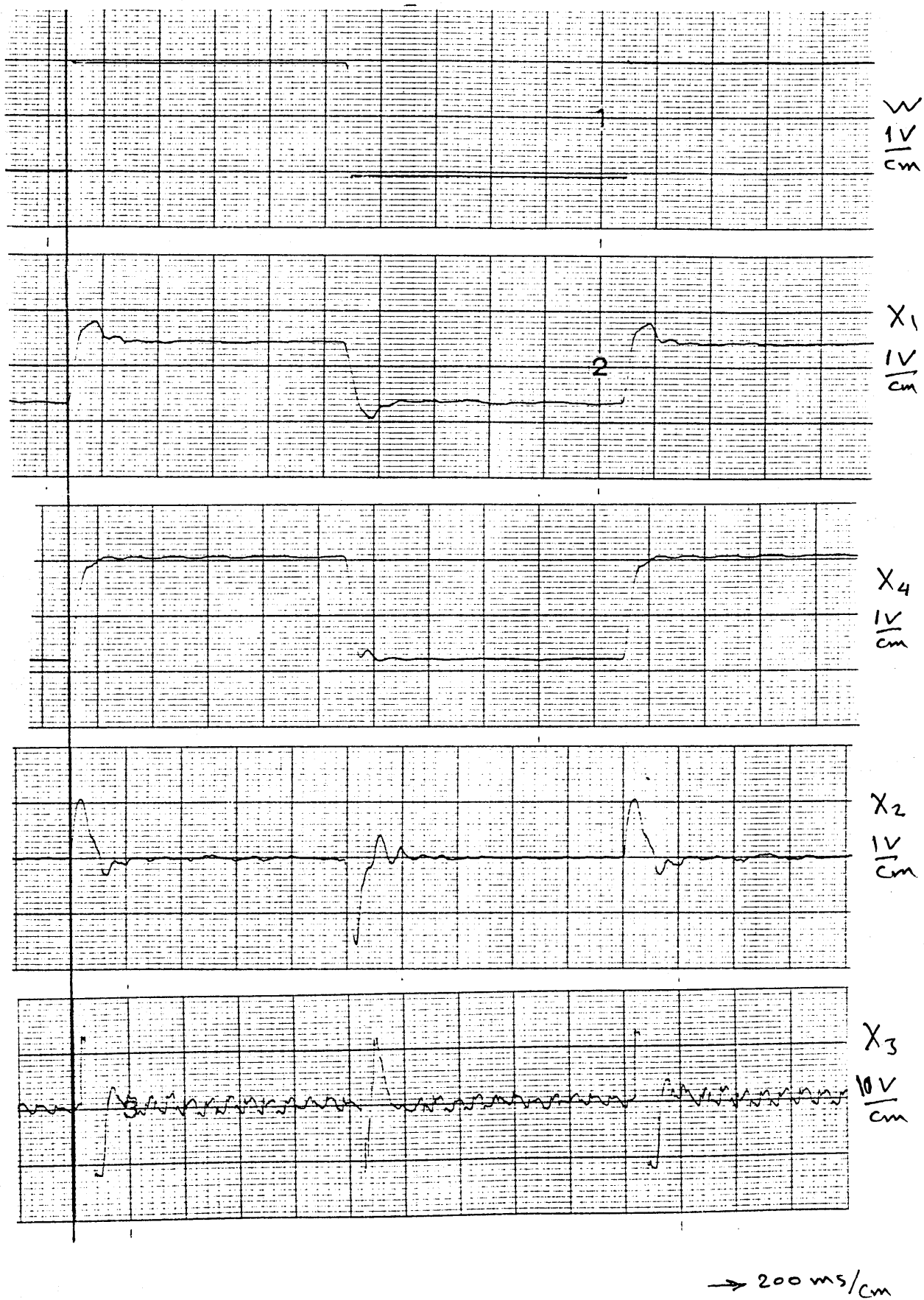


Fig. 7.17 Equipment Step Responses Using the Strategy (7.17a)

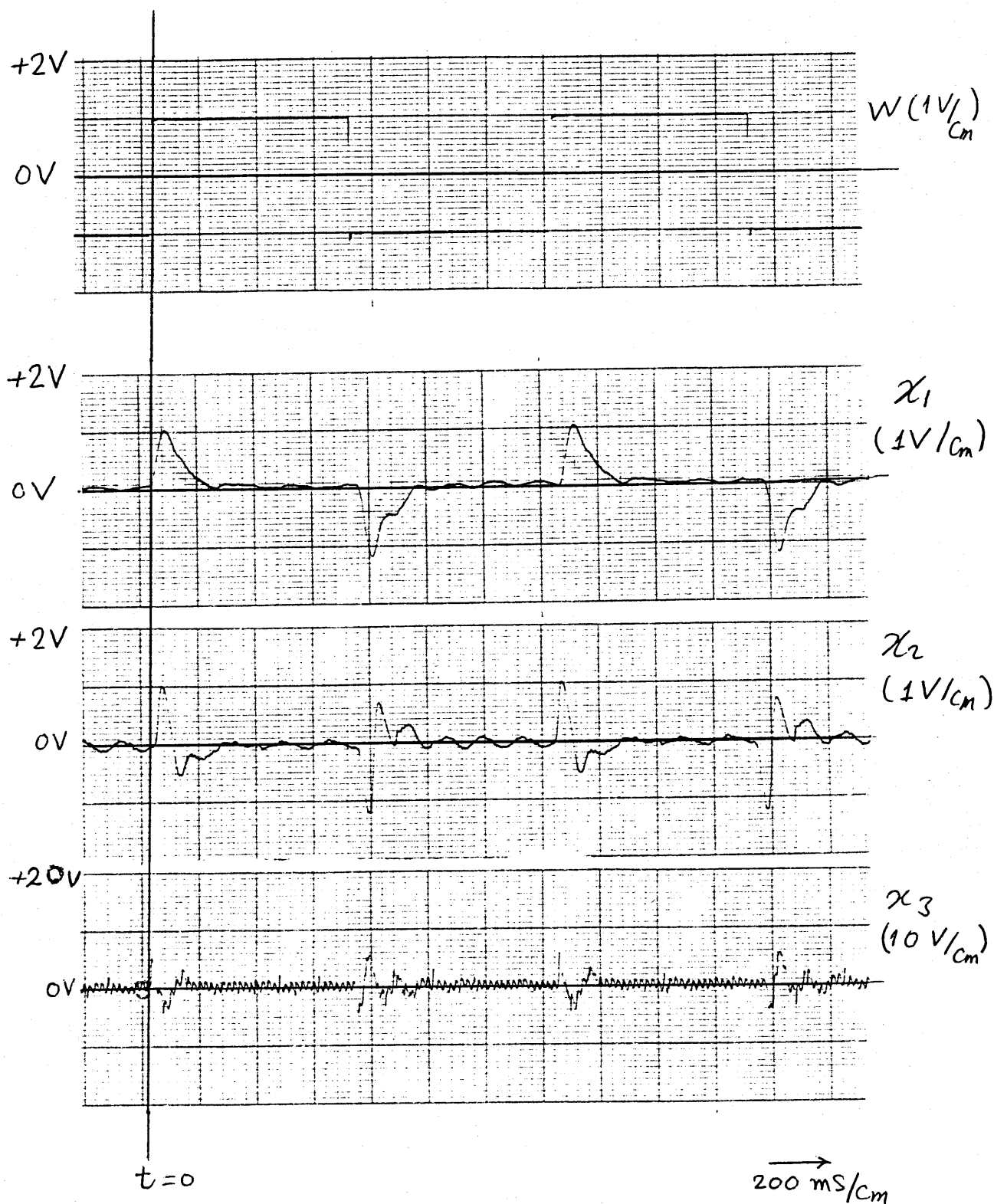


Fig. 7.18 Equipment Step Responses Using the Strategy (7.17b)

By experiment on the system, another good control strategy, this time ignoring feedback of x_4 and x_5 , was found using:

$$K_c = \begin{bmatrix} 1.70 \\ 0.70 \\ 0.12 \\ 0 \\ 0 \end{bmatrix}, \quad \dots (7.18)$$

The eigen values of this closed loop system are:

$$-33.2 \pm j14.3$$

$$-578.5$$

$$-15.0$$

$$-70.0$$

Step function responses taken from the experimental equipment using this strategy is shown in Fig 7.19 and the equivalent computer simulation results are shown in Figs 7.20, 7.21 & 7.22.

It can be seen that the computer results again match the experimental one reasonably well, from which we can conclude that the mathematical model is adequate.

We can also conclude that feedback of x_4 , x_5 does not greatly affect the closed loop eigenvalues, and so, does not have a radical effect on closed loop behaviour. This will almost be true for true quarter car model, since the axle behaviour is clearly dominated by the random road disturbances and not the control input.

Various strategies can clearly be developed, by varying $\bar{Q}_c$ and $\bar{R}_c$, and by experimentation, but they all depend on the states being available. While this is true on the laboratory equipment, it is certainly not so in the practical vehicle, nor in the quarter car model. The design of a state estimator, or Kalman filter, is therefore the next task to be undertaken and this is the subject of Section 7.4. The influence of any implementations of the Kalman filter on the practical control strategies, and relevant experiments and results are also examined in Sect. 7.5.

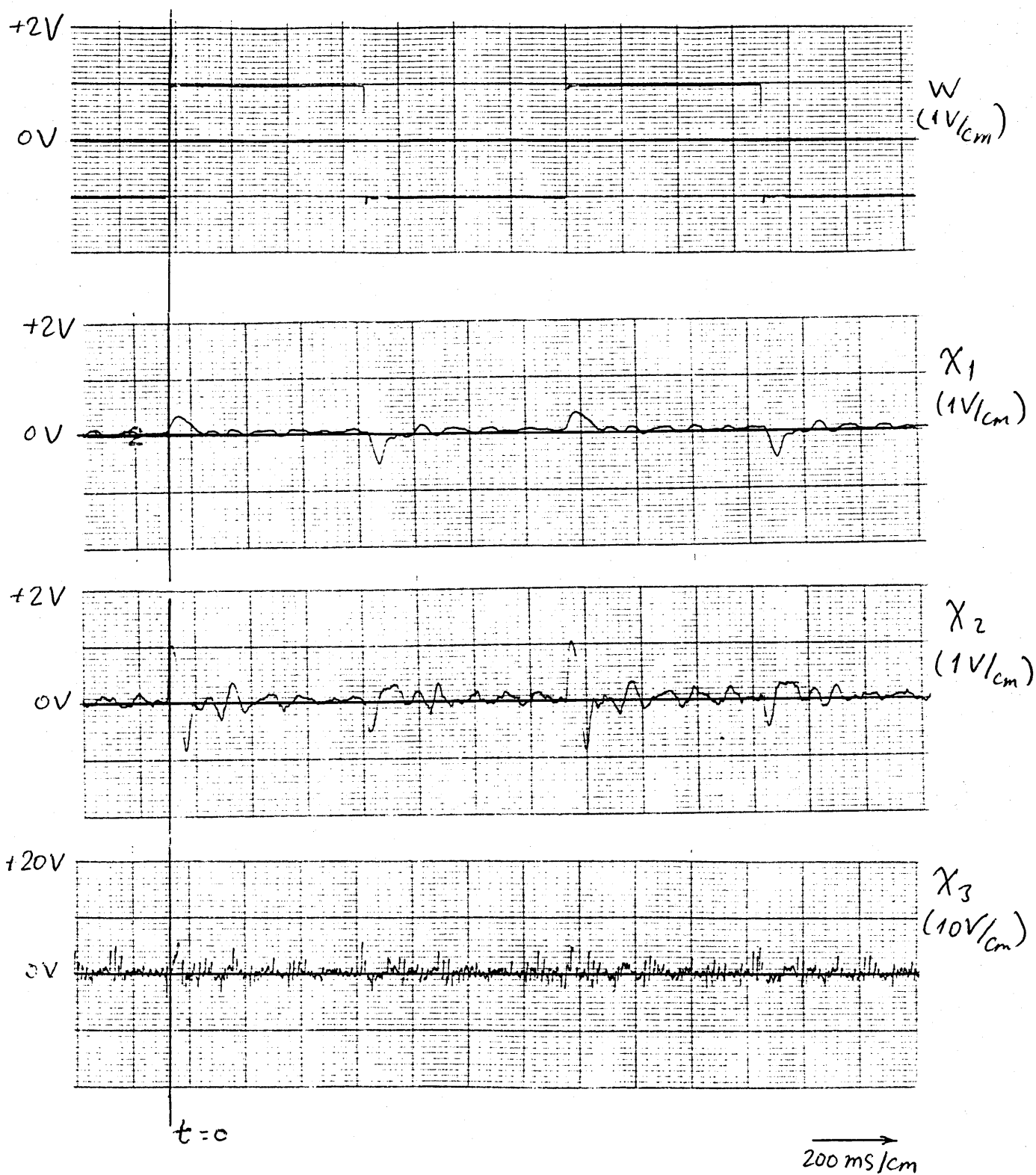


Fig. 7.19 Equipment Step Responses Using the Strategy (7.18)

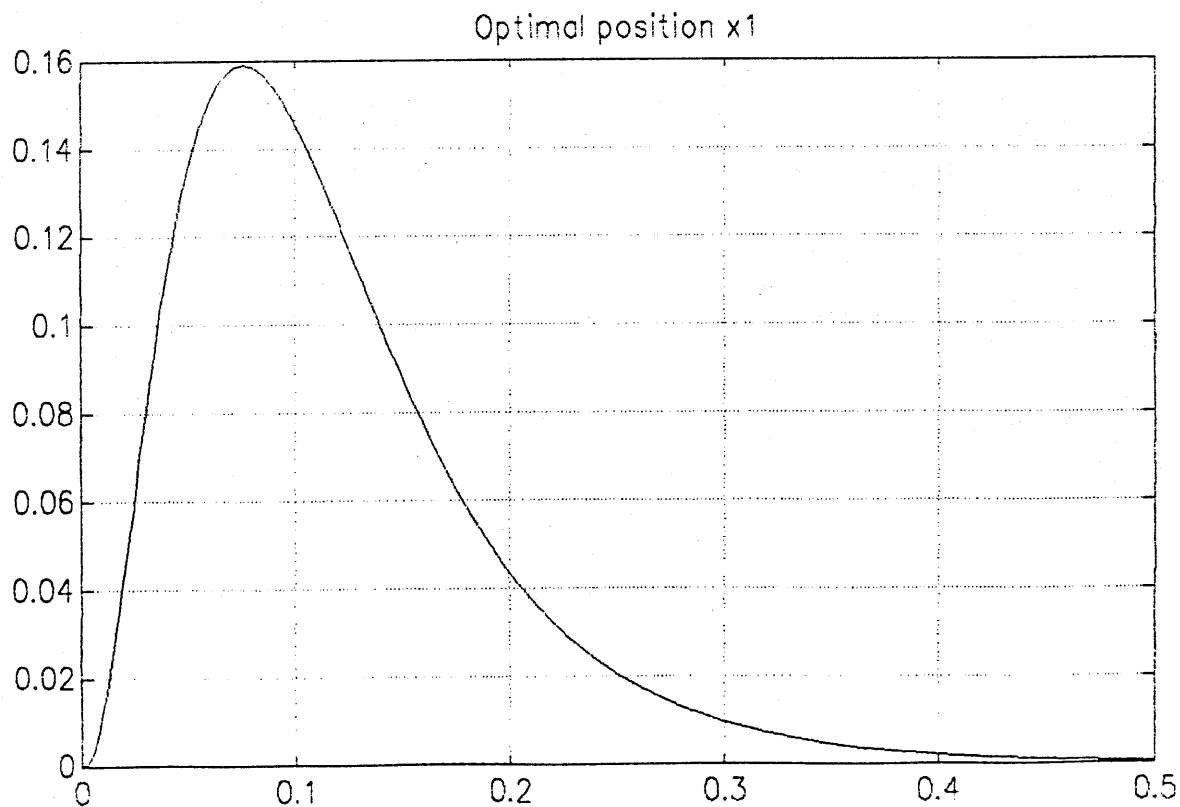
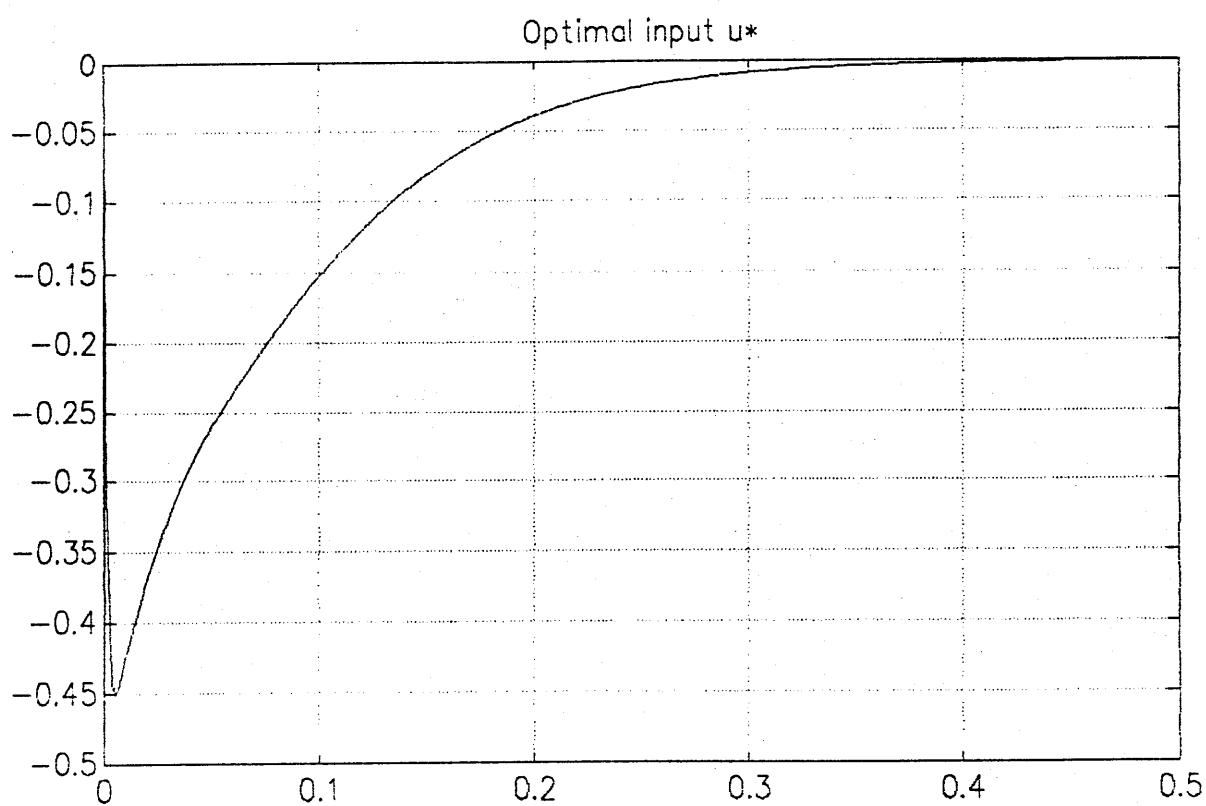


Fig. 7.20 Step Response Simulation (u^* and x_1) Using (7.18)

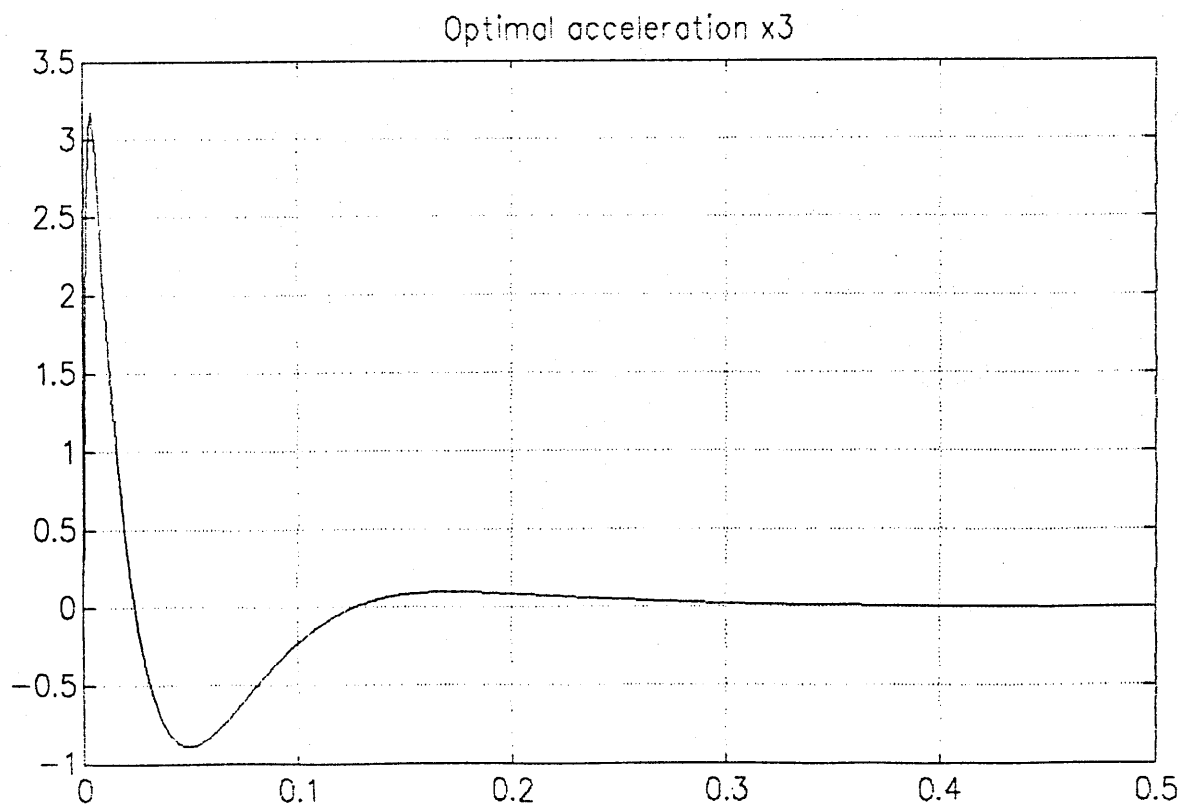
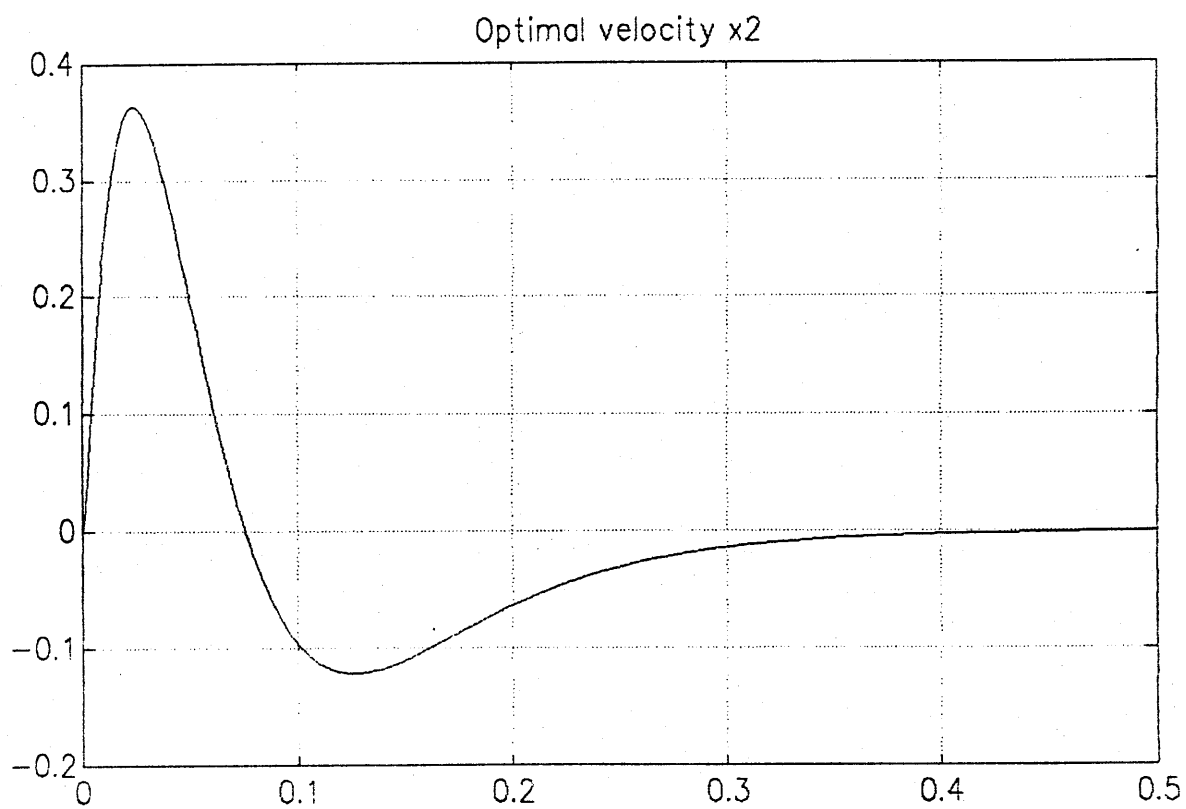


Fig. 7.21 Step Response Simulation (x_2 and x_3) Using (7.18)

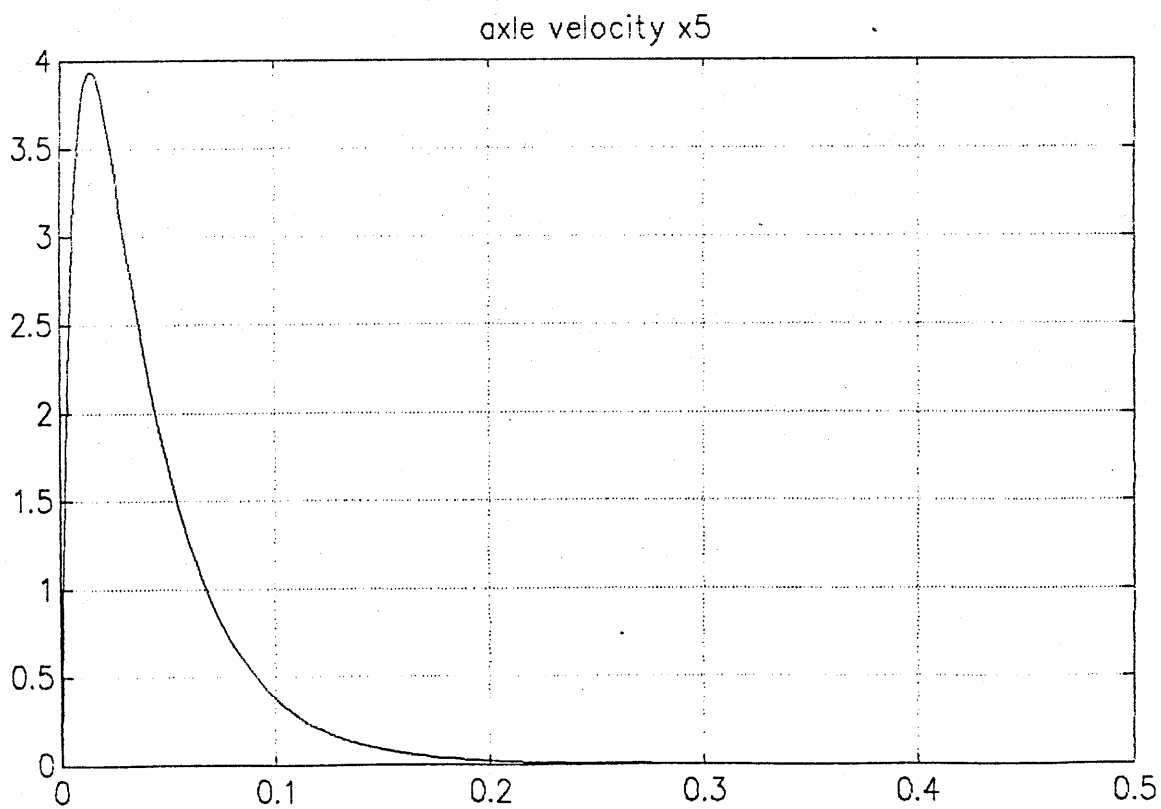
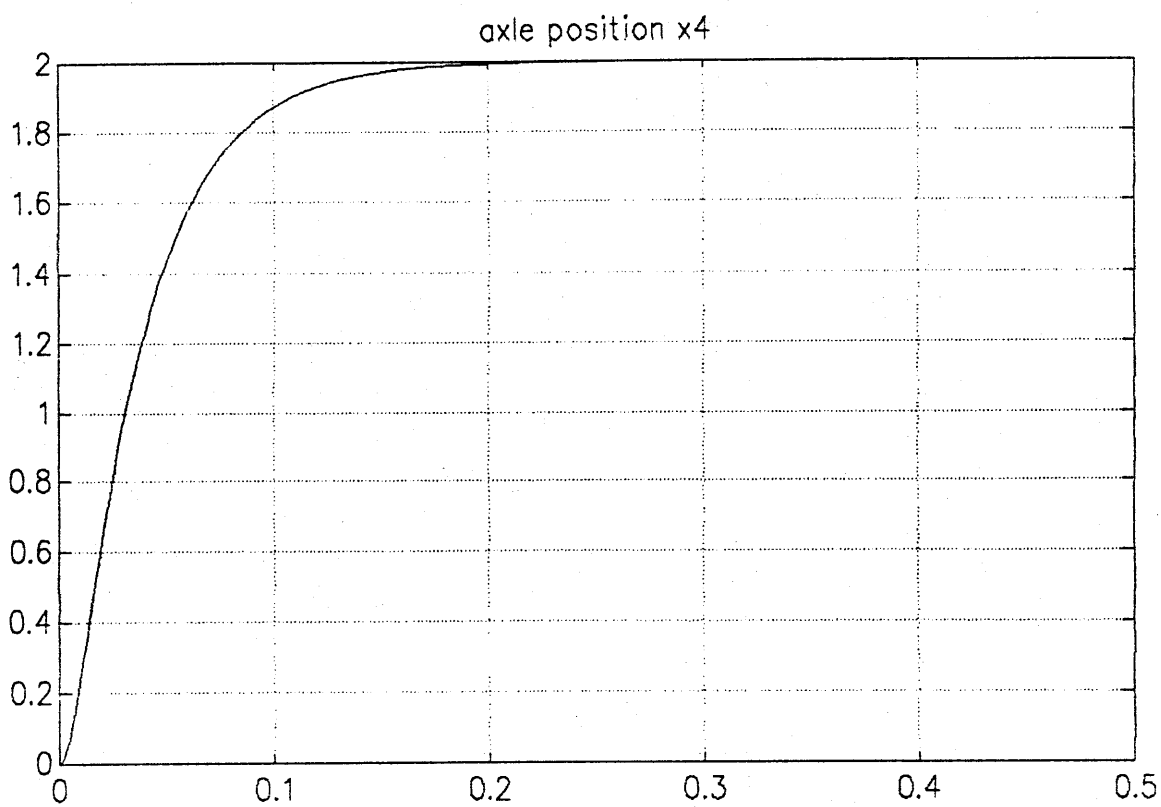


Fig. 7.22 Step Response Simulation (x_4 and x_5) Using (7.18)

7.4 THE STATE ESTIMATION PROBLEM:

To design a state estimator (Kalman filter in this case) for any system, that system must be observable². Testing the observability of the system represented by Equations (7.14), we found one unobservable mode related to the sprung mass position state, x_1 , where the sprung mass acceleration is the only output (Eqn. 7.14b).

Clearly, some way round this difficulty must be found if a Kalman filter is to form the basis of a control strategy. The underlying problem is of course that the sprung mass absolute position and velocity are not directly available - which in turn is a major reason for the presence of the Kalman filter.

Three distinct lines of attack were developed to deal with this problem, and in each case the suggested solution affects the control strategy which can be implemented.

The structure of the problem also suggests the possibility of designing reduced order Kalman filters, since it seems (Sect 7.3) there might be a good prospect in feeding back only some states. This topic is explored in Sect 7.4.4.

7.4.1 APPROXIMATE KALMAN FILTER:

The first solution involves approximating the system model - at least as far as the Kalman filter is concerned - so that the unobservable mode disappears. The model of the actuator A2 and load M (Eqn 7.3) is replaced in the Kalman filter design with;

(2) A system is observable if all the modes affect the outputs. For a system to be observable, the rank of;

$$\begin{bmatrix} C^T & (CA)^T & (CA^2)^T & (CA^3)^T & \dots & (CA^{m-1})^T \end{bmatrix} = m$$

$$F(s) = \frac{4.44 \times 10^5}{(s+s_0)(s+56 \pm j73.9)} \quad \dots (7.19)$$

i.e. as far as the Kalman filter is concerned, the actuator pole at the origin is slightly shifted to $-s_0$ in the s -plane. The actual system of course is still represented by Equation (7.14). s_0 is chosen so that the frequency content of x_1 is reasonably accurately estimated for "high" frequencies, while the low frequency behaviour will be accommodated by a separate strategy (Sect 7.5.1).

Following the procedure of Section 7.2 for $s_0=3$ (Eqn 7.19) (minimum frequency of interest is about 0.5 Hz), the system's continuous time state equations become (cf Eqn 7.14a):

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ -258 & -893.6 & -115 & -949.5 & -83.9 \\ 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & -105 & -85 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 4440 \\ 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1207.5 \\ 0 \\ 105 \end{bmatrix} w(t) \quad \dots (7.20a)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + v(t) \quad \dots (7.20b)$$

The eigen values of this system are:

-3.0
-56.0 $\pm$ j73.9
-15.0
-70.0

This is observable. In discrete form and using 5 msec sampling time, the model becomes;

$$\begin{aligned} \mathbf{x}(n+1) &= \phi \mathbf{x}(n) + \psi u(n) \\ y(n) &= C \mathbf{x}(n) \end{aligned} \quad \dots (7.21)$$

where

$$\phi = \begin{bmatrix} 0.100E+01 & 0.484E-01 & 0.102E-02 & -0.167E-02 & -0.156E-03 \\ -0.264E-01 & 0.908E+00 & 0.367E-01 & -0.954E-01 & -0.892E-02 \\ -0.946E+00 & -0.330E+01 & 0.487E+00 & -0.339E+01 & -0.327E+00 \\ 0.000E+00 & 0.000E+00 & 0.000E+00 & 0.989E+00 & 0.405E-01 \\ 0.000E+00 & 0.000E+00 & 0.000E+00 & -0.426E+00 & 0.644E+00 \end{bmatrix}$$

$$\psi = \begin{bmatrix} 0.792E-02 \\ 0.454E+00 \\ 0.163E+02 \\ 0.000E+00 \\ 0.000E+00 \end{bmatrix}$$

The control system is the cascade arrangement of the optimal filter (Kalman filter) and optimal controller shown in Fig 7.23 (cf to Fig. 7.3b);

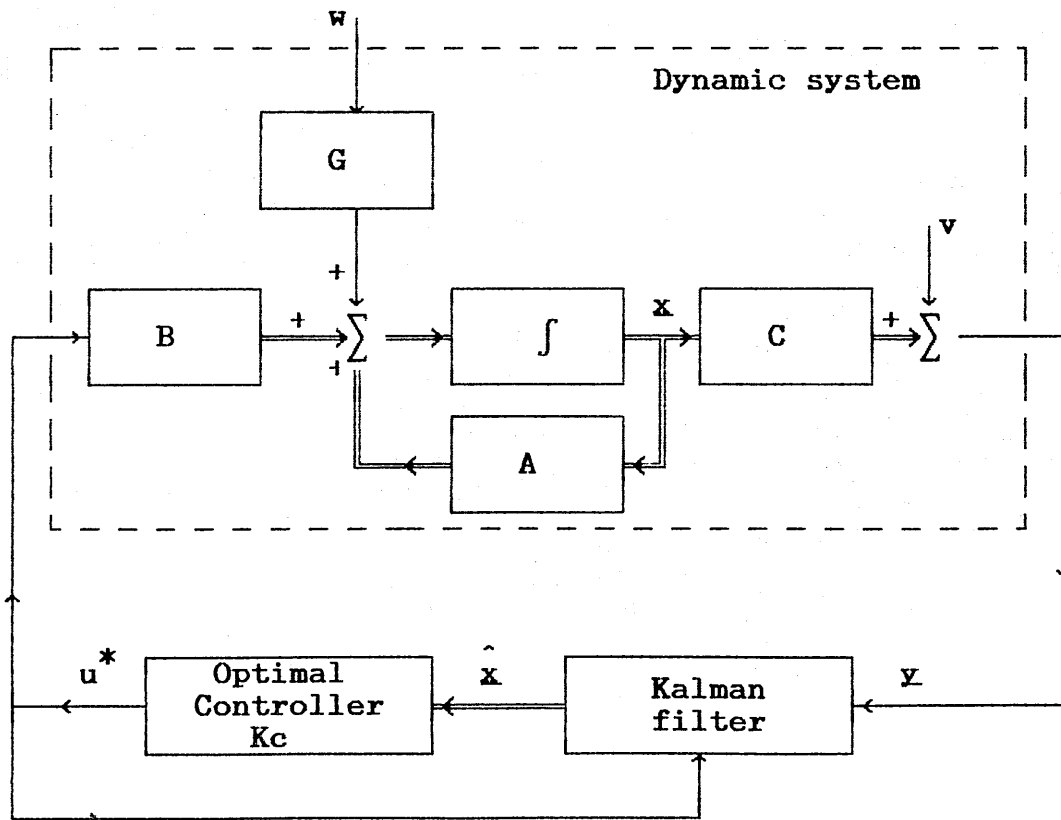


Fig. 7.23 Configuration of Active Suspension System.

Since the system is linear time invariant, a sub-optimal Kalman filter can be designed which has a constant "optimal" filter gain matrix by solving the Kalman filter algorithm (Sect. 4.4) off line [3].

The process and measurement noise variances are described by Q and R (Chapter 4), and for road variance of 3 cm^2 and a typical accelerometer/charge amplifier noise content at 1 volt^2 variance, this translates (at 1.8 v/cm , Sect 7.1) to;

$$Q = 10, \quad R = \begin{bmatrix} 1 \end{bmatrix}$$

for a sampling period $T = 0.005 \text{ sec}$, this yields;

$$Q_d = 0.05, \quad R_d = \begin{bmatrix} 1 \end{bmatrix}$$

The sub-optimal Kalman filter gain matrix and the sub-optimal error covariance matrix are as follows:

$$K_f(\omega) = \begin{bmatrix} -0.5428E-02 \\ 0.3779E-01 \\ 0.5215E+00 \\ -0.8868E-02 \\ -0.2121E-01 \end{bmatrix}$$

$$P(\omega) = \begin{bmatrix} 0.4660E+00 & 0.2471E+00 & -0.4323E+00 & -0.2654E+00 & 0.3129E+00 \\ 0.2471E+00 & 0.2858E+00 & 0.2979E-01 & -0.3471E+00 & 0.2254E+00 \\ -0.4323E+00 & 0.2979E-01 & 0.6367E+02 & -0.4762E+00 & -0.2451E+01 \\ -0.2654E+00 & -0.3471E+00 & -0.4762E+00 & 0.5063E+00 & 0.3130E-01 \\ 0.3129E+00 & 0.2254E+00 & -0.2451E+01 & 0.3130E-01 & 0.9045E+01 \end{bmatrix}$$

where;

$K_f(\omega)$ is the sub-optimal KF gain vector.

$P(\omega)$ is the sub-optimal error covariance matrix.

The open loop system response to a unit step disturbance and its estimation (sprung mass states) are shown in Figs 7.24 to 7.26. It can be seen that the Kalman filter estimations well match the actual states, although some drift is shown in Fig 7.24, this is to be expected and being low frequency should not affect any reasonable strategy.

ISIS

Open Loop, sprung mass position

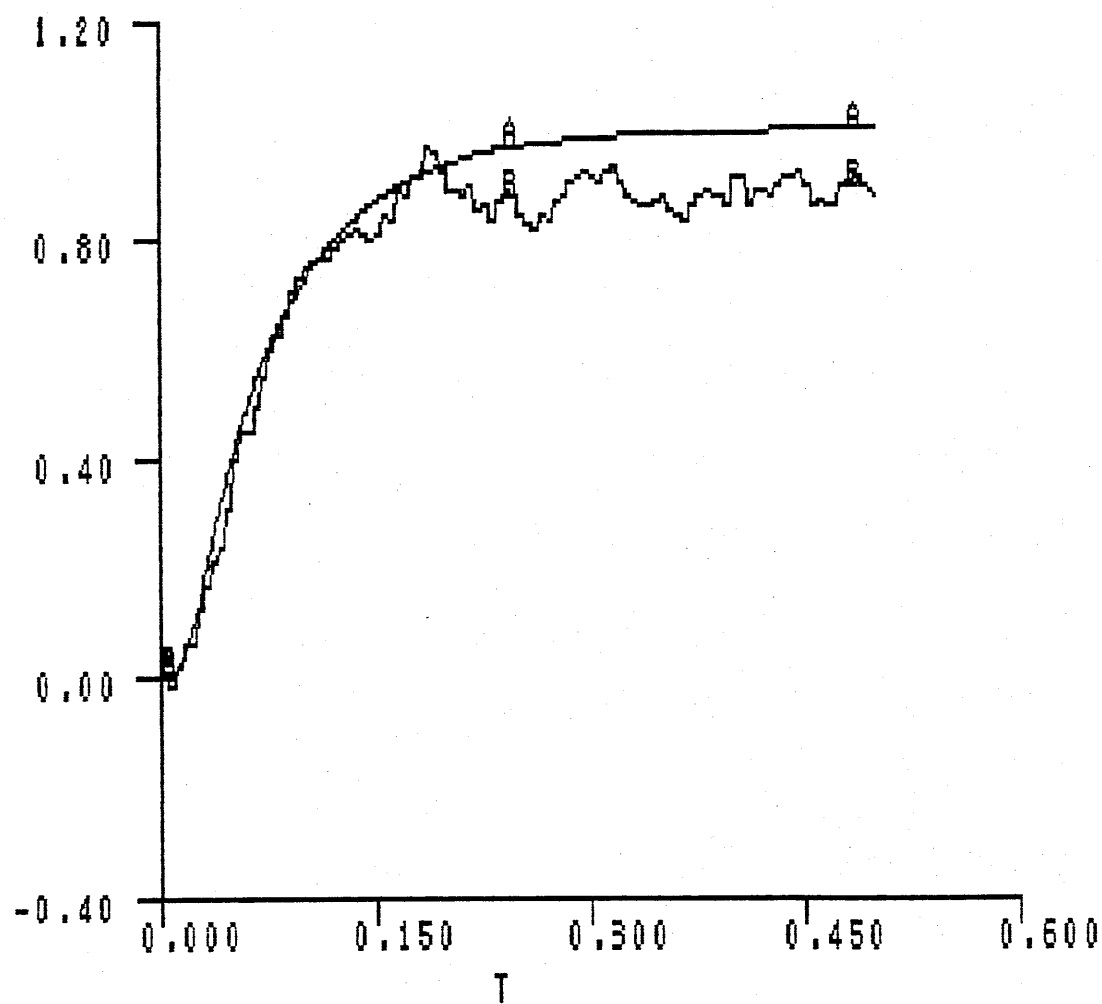


Fig 7.24 Approximate Kalman Filter (x_1 and $\hat{x}_1$)

ISIS

Open Loop, Sprung mass velocity

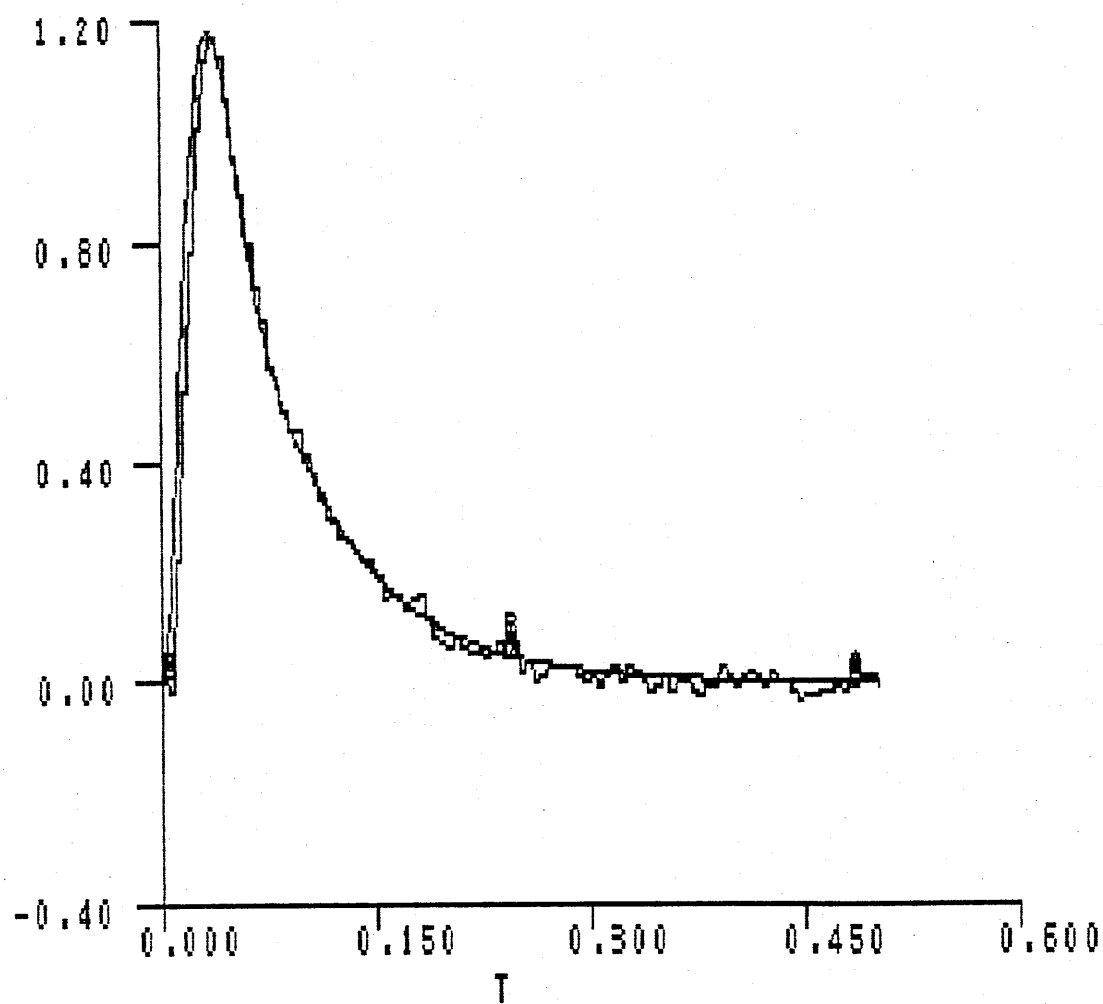


Fig 7.25 Approximate Kalman Filter (x_2 and $\hat{x}_2$)

ISIS

Open Loop, sprung mass acceleration

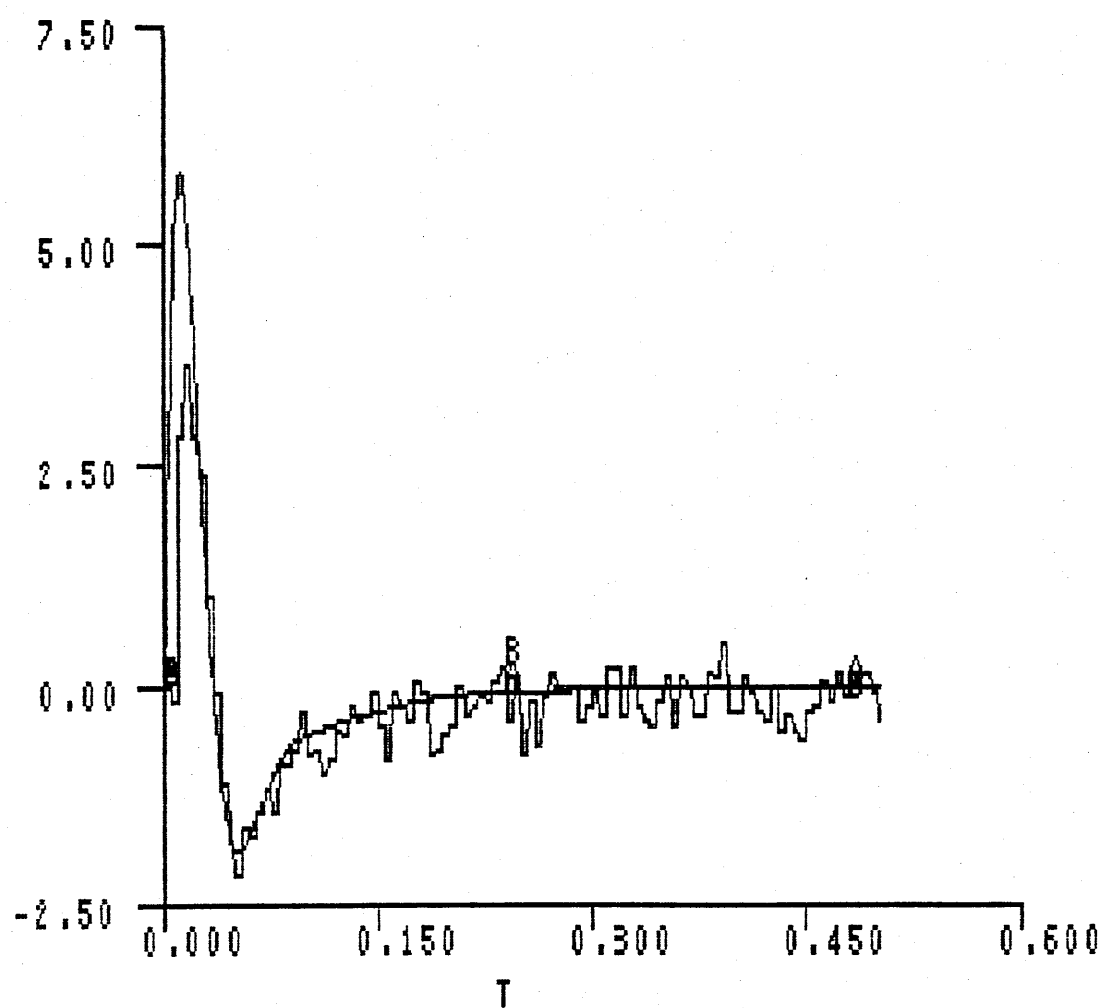


Fig 7.26 Approximate Kalman Filter (x_3 and $\hat{x}_3$)

For a random input sequence shown in Fig. 7.27, the "optimal" sprung mass position (x_1 & $\hat{x}_1$), velocity (x_2 & $\hat{x}_2$) and acceleration (x_3 & $\hat{x}_3$) states of the actual and the (discrete) estimated states are shown in Figs. 7.28 to 7.30 respectively using the control strategy (Eqn 7.18). Again, the Kalman filter estimates are all acceptably good.

A control strategy using the states estimated by this Kalman filter is considered in Sect 7.5.1.

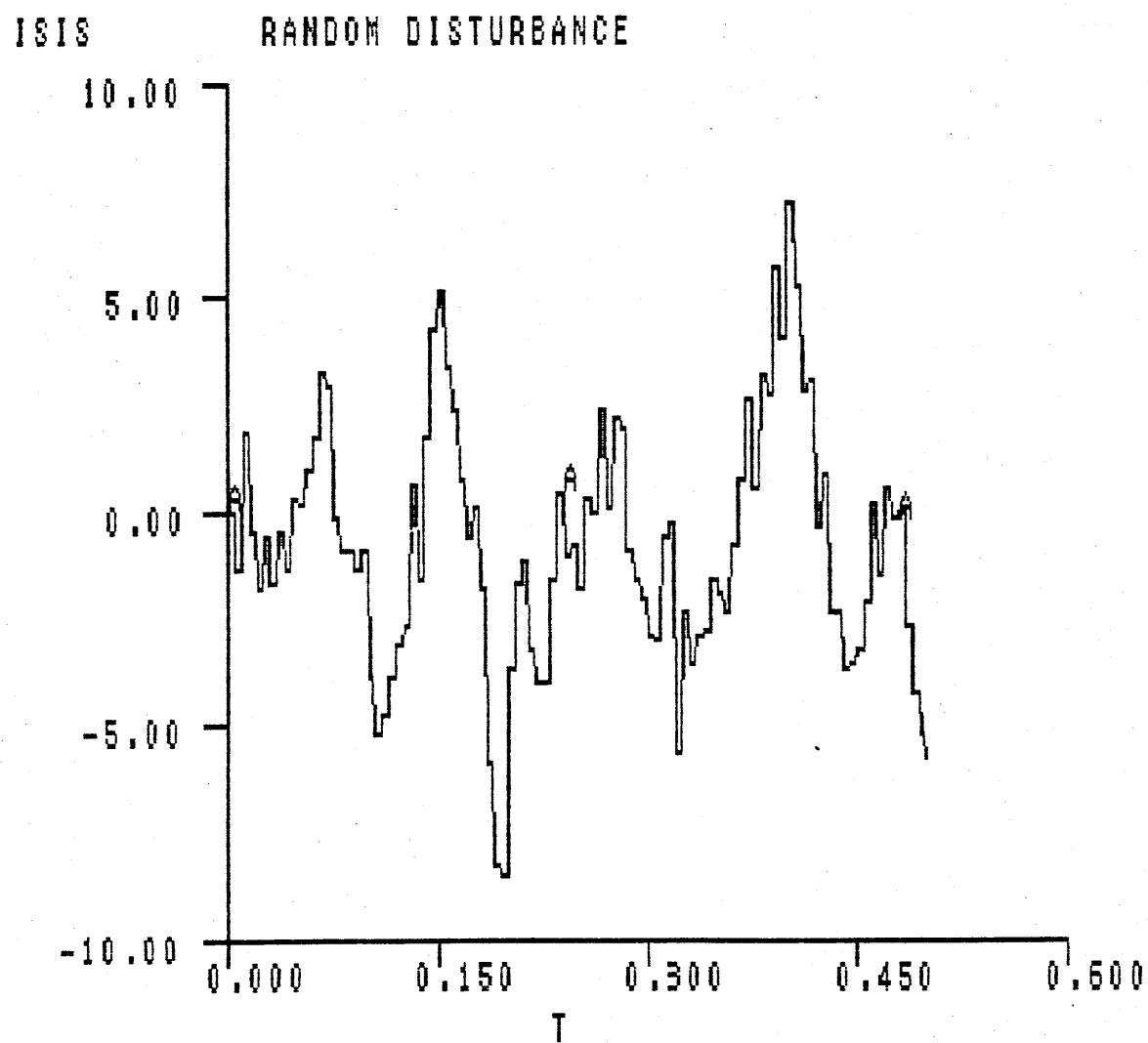


Fig 7.27 Random Disturbance

ISIS

Optimal, sprung mass position

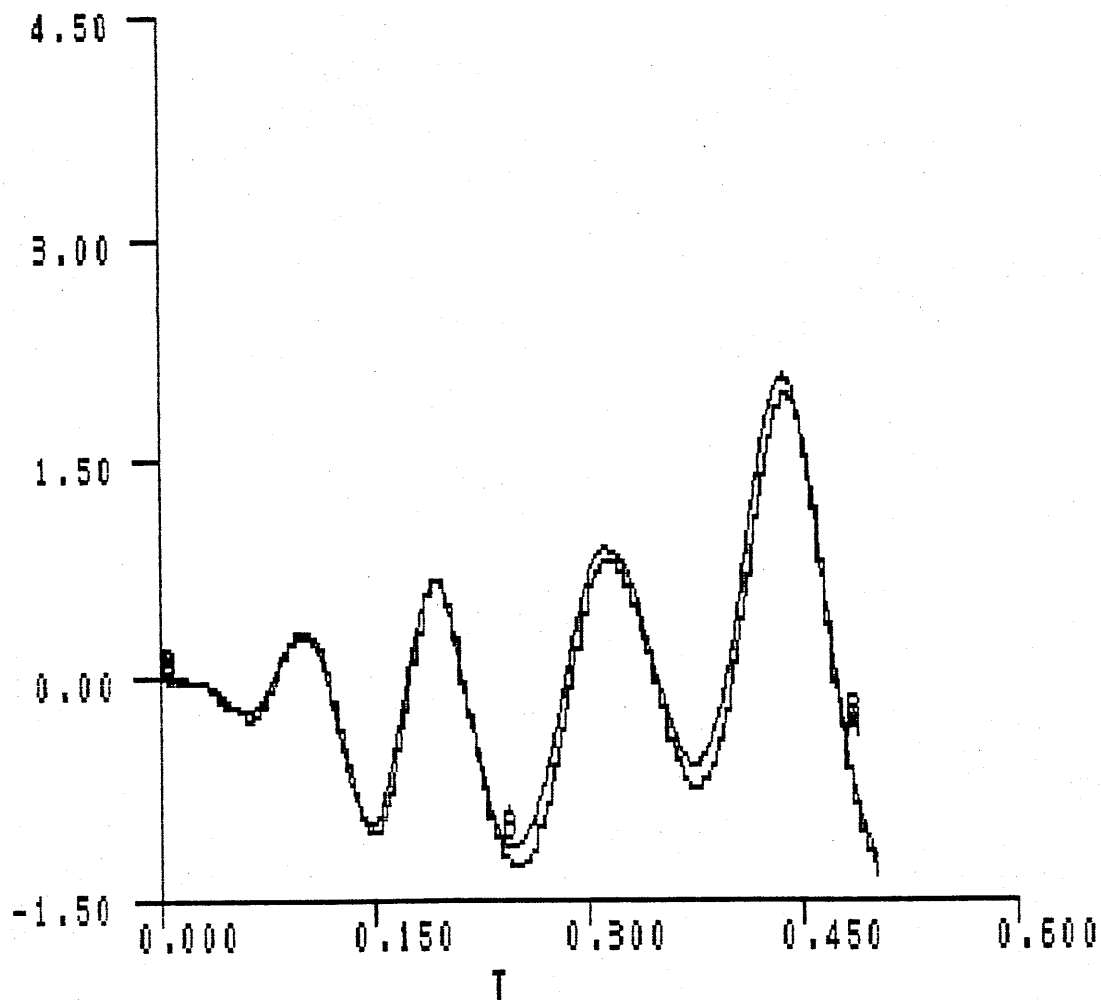


Fig 7.28 Approximate Kalman Filter (x_1 and $\hat{x}_1$)

ISIS

Optimal, sprung mass velocity

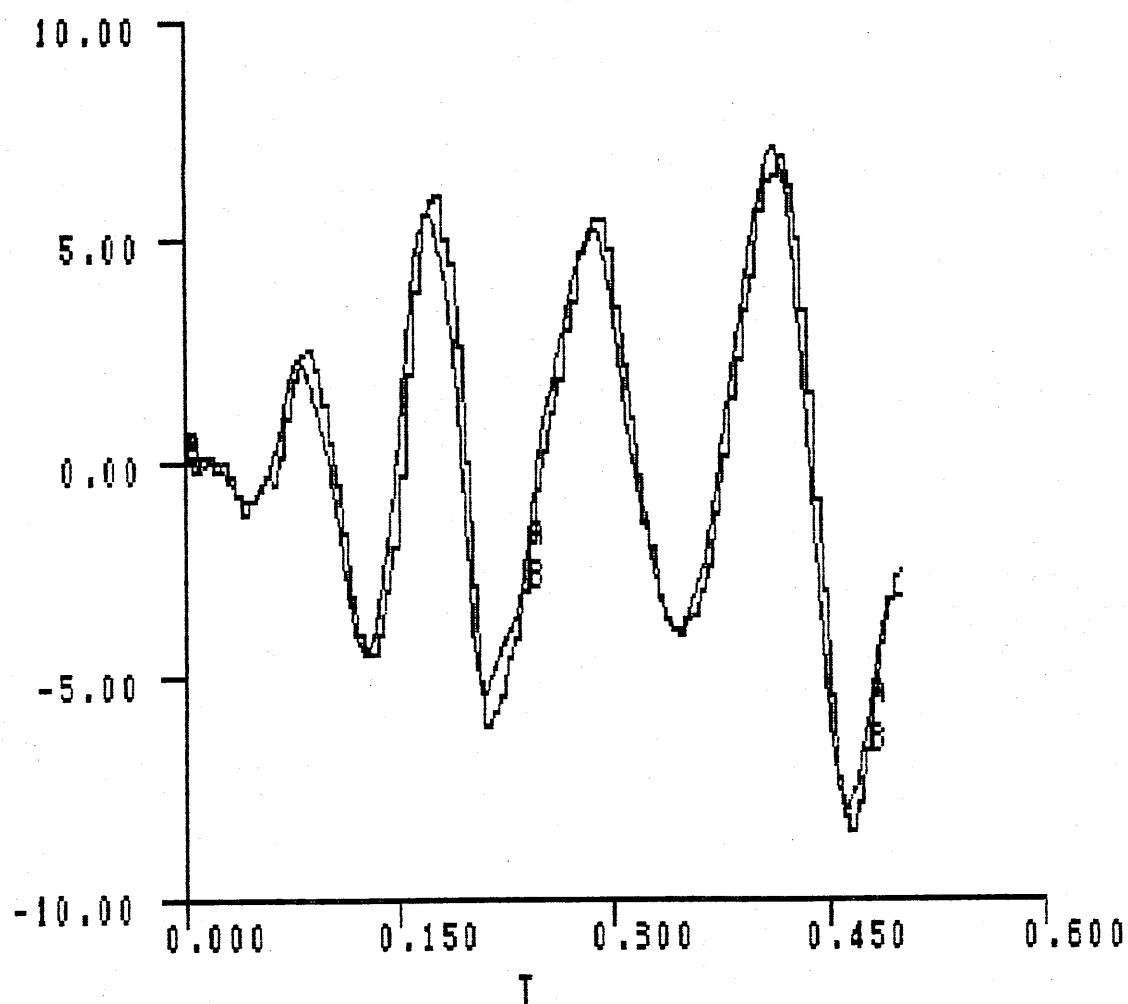


Fig 7.29 Approximate Kalman Filter (x_2 and $\hat{x}_2$)

ISIS

Optimal, sprung mass acceleration

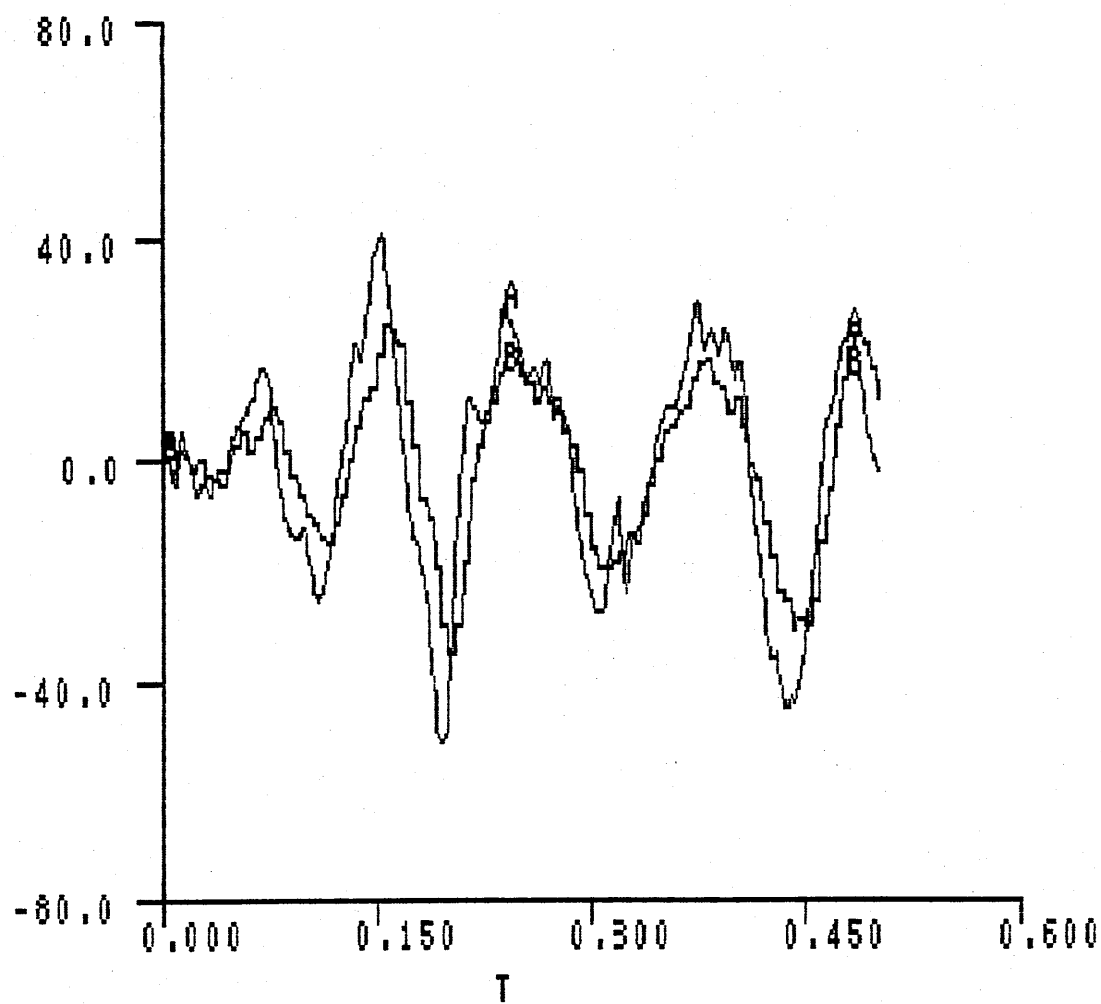


Fig 7.30 Approximate Kalman Filter (x_3 and $\hat{x}_3$)

7.4.2 AUGMENTED OUTPUTS:

A second possible strategy to overcome the system unobservability problem is to use more system outputs as Kalman filter inputs. In this case, considering the sprung-unsprung relative position ($x_1 - x_4$) as another measured system output, the system model Eqn (7.14) can be re-written as:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & -860 & -112 & -1176 & -92 \\ 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & -105 & -85 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 4440 \\ 0 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1176 \\ 0 \\ 105 \end{bmatrix} w(t) \quad (7.22a)$$

$$\mathbf{y}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \end{bmatrix} \mathbf{x}(t) + \mathbf{v}(t) \quad \dots (7.22b)$$

Testing the observability of the system represented by Eqns (7.22), it was found observable and a Kalman filter can therefore be designed.

In discrete form (Eqns 7.21) using 5 msec sampling time, using Eqns (7.22);

$$\phi = \begin{bmatrix} 0.1000\text{E}+11 & 0.485\text{E}-11 & 0.103\text{E}-22 & -0.208\text{E}-22 & -0.175\text{E}-33 \\ 0.0000\text{E}+01 & 0.912\text{E}+00 & 0.370\text{E}-11 & -0.119\text{E}+01 & -0.100\text{E}-12 \\ 0.0000\text{E}+00 & -0.318\text{E}+01 & 0.498\text{E}+00 & -0.424\text{E}+01 & -0.374\text{E}+00 \\ 0.0000\text{E}+00 & 0.0000\text{E}+00 & 0.0000\text{E}+00 & 0.989\text{E}+00 & 0.405\text{E}-11 \\ 0.0000\text{E}+00 & 0.0000\text{E}+00 & 0.0000\text{E}+00 & -0.426\text{E}+00 & 0.644\text{E}+00 \end{bmatrix}$$

$$\psi = \begin{bmatrix} 0.795\text{E}-02 \\ 0.456\text{E}+00 \\ 0.164\text{E}+02 \\ 0.0000\text{E}+00 \\ 0.0000\text{E}+00 \end{bmatrix}$$

A typical Kalman filter is designed for this system considering the case as before (Sect 7.4.1), and with a noise variance on $(x_1 - x_4)$ of 0.1 (volts)^2 ;

$$Q = 10, \quad Q_d = 0.05,$$

$$R_d = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$K_f(\omega) = \begin{bmatrix} 0.6926E-03 & 0.2476E-01 \\ 0.3777E-01 & 0.2031E+00 \\ 0.5126E+00 & 0.1785E+01 \\ -0.4585E-02 & -0.4458E+00 \\ -0.1596E-01 & -0.9446E+00 \end{bmatrix}$$

$$P(\omega) = \begin{bmatrix} 0.5958E-02 & 0.6845E-03 & -0.1763E-01 & 0.3365E-02 & -0.1931E-02 \\ 0.6845E-03 & 0.2065E-01 & 0.9625E-01 & -0.2517E-01 & -0.6819E-01 \\ -0.1763E-01 & 0.9625E-01 & 0.6119E+02 & -0.3429E+00 & -0.2245E+01 \\ 0.3365E-02 & -0.2517E-01 & -0.3429E+00 & 0.6819E-01 & 0.2946E+00 \\ -0.1931E-02 & -0.6819E-01 & -0.2245E+01 & 0.2946E+00 & 0.8416E+01 \end{bmatrix}$$

The open loop system responses to a unit step disturbance and its estimation (sprung mass states) are shown in Figs 7.31 to 7.33, and response to the noise of Fig 7.34 in Fig 7.35 to 7.37. In every case, the Kalman filter can be seen to generate reasonably good state estimates.

A control strategy assuming all the system states to be available, as estimated by this Kalman filter, is considered in Sect 7.5.2.

H

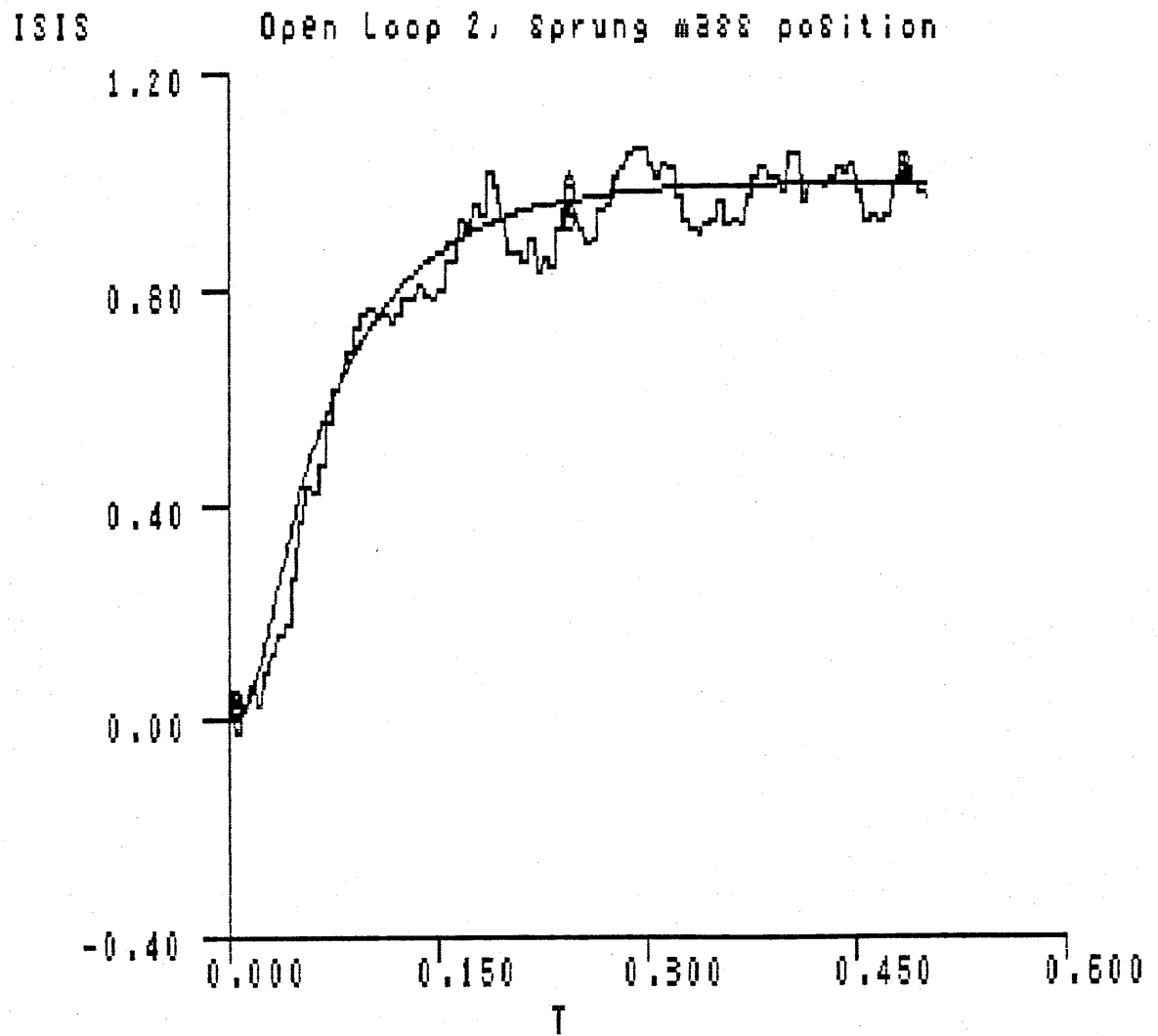


Fig 7.31 Augmented Outputs Kalman Filter (x_1 and $\hat{x}_1$)

ISIS

Open loop 2, sprung mass velocity

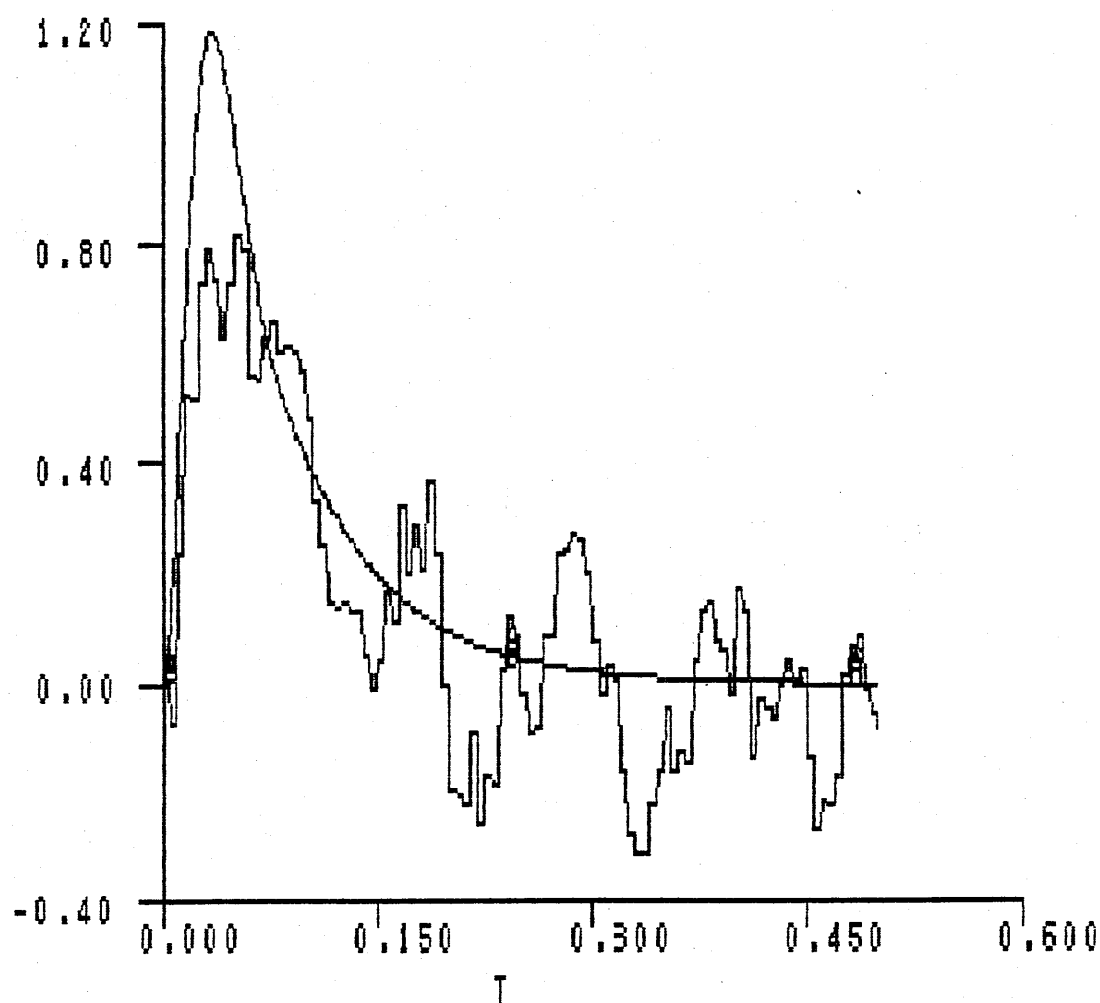


Fig 7.32 Augmented Outputs Kalman Filter (x_2 and $\hat{x}_2$)

ISIS Open Loop 2, sprung mass acceleration

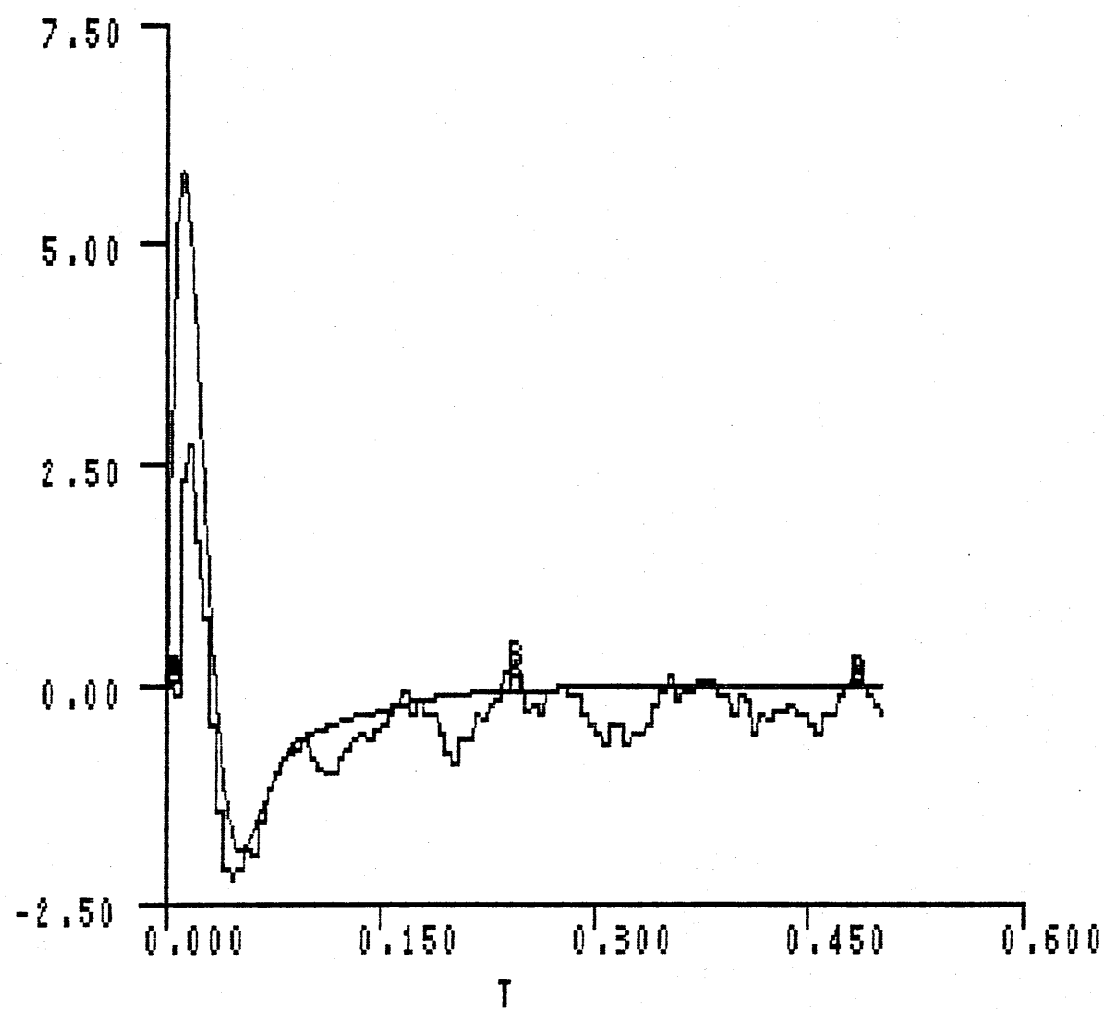


Fig 7.33 Augmented Outputs Kalman Filter (x_3 and $\hat{x}_3$)

ISIS

Optimal 2, random disturbance

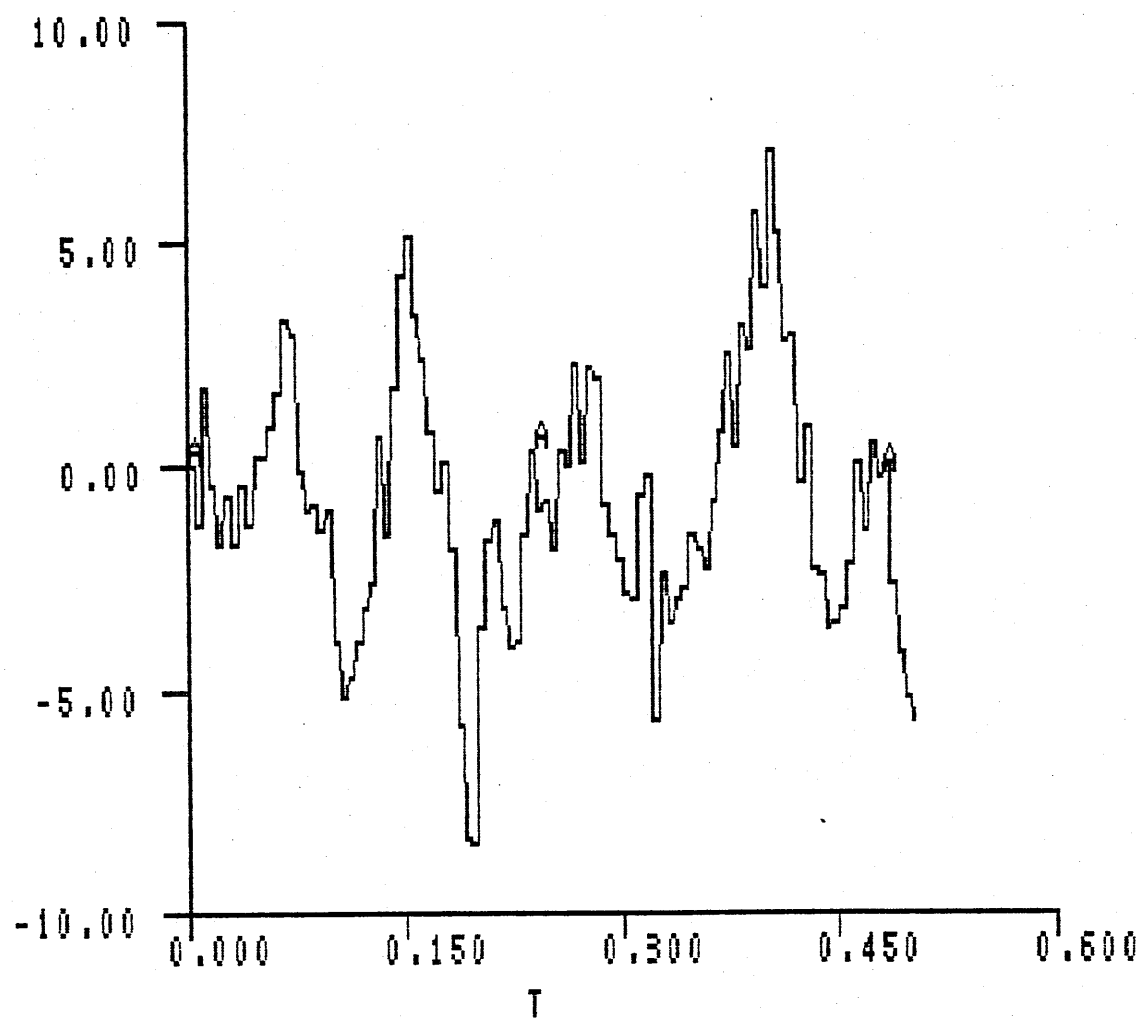


Fig 7.34 Random Disturbance

ISIS Optimal 2, sprung mass position

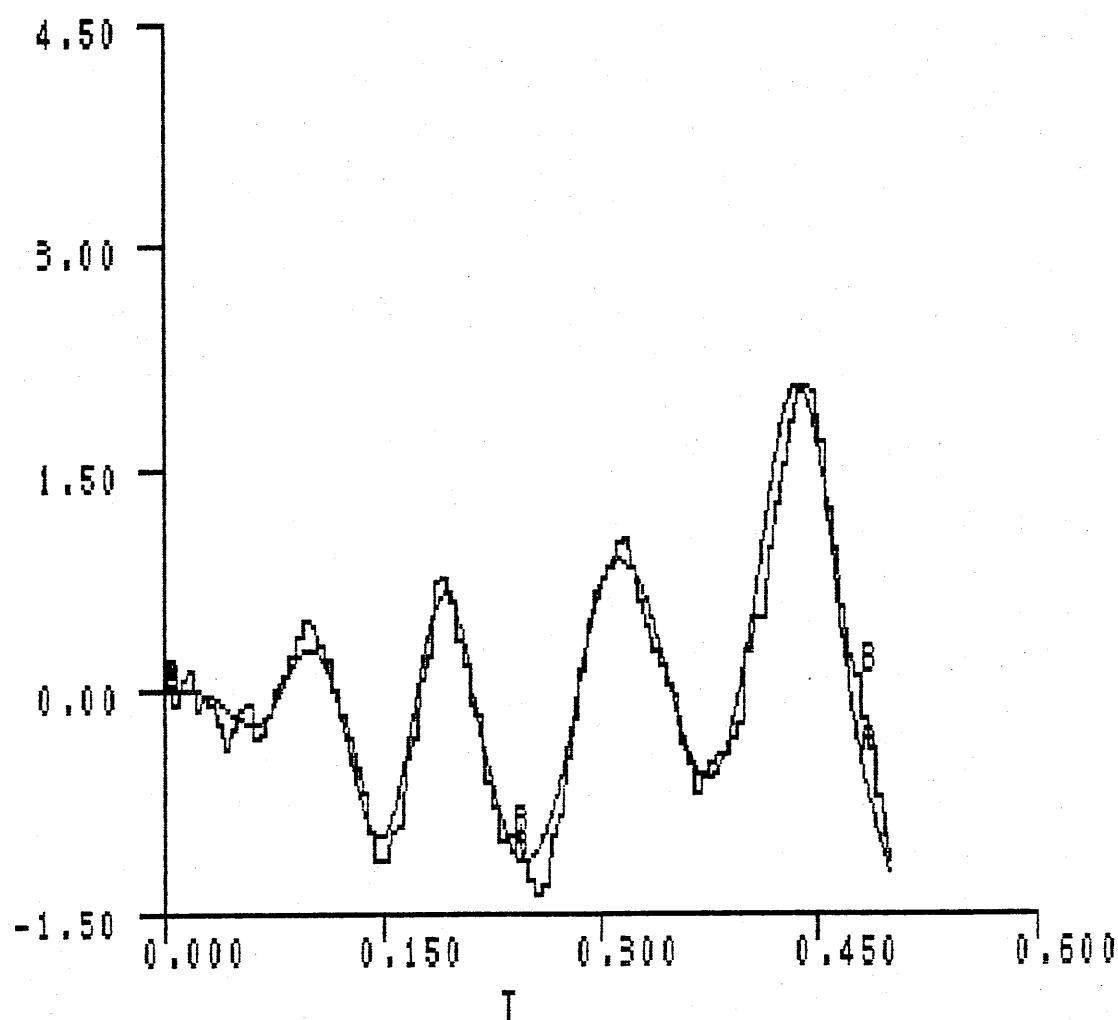


Fig 7.35 Augmented Outputs Kalman Filter (x_1 and $\hat{x}_1$)

ISIS

Optimal 2, sprung mass velocity

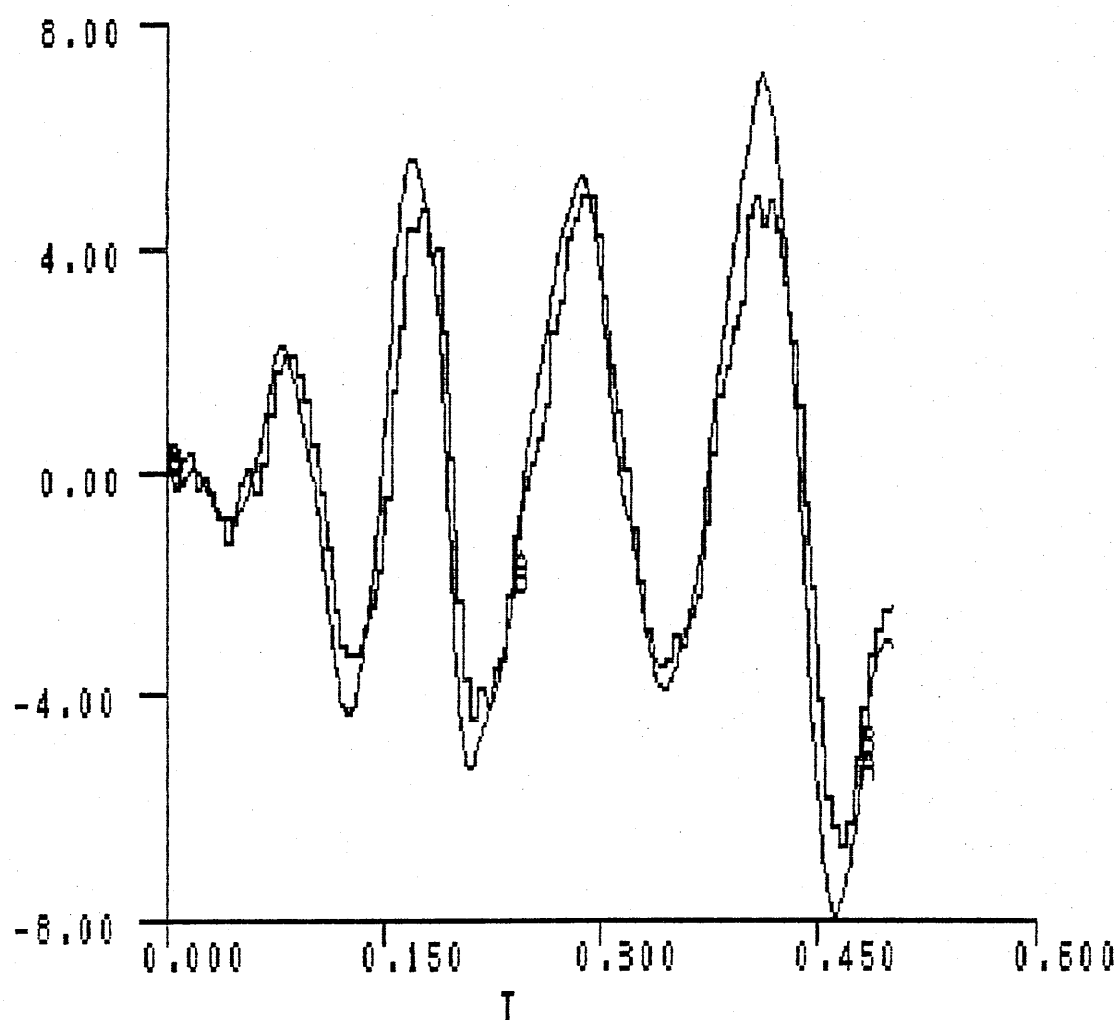


Fig 7.36 Augmented Outputs Kalman Filter (x_2 and $\hat{x}_2$)

ISIS

Optimal 2, sprung mass acceleration

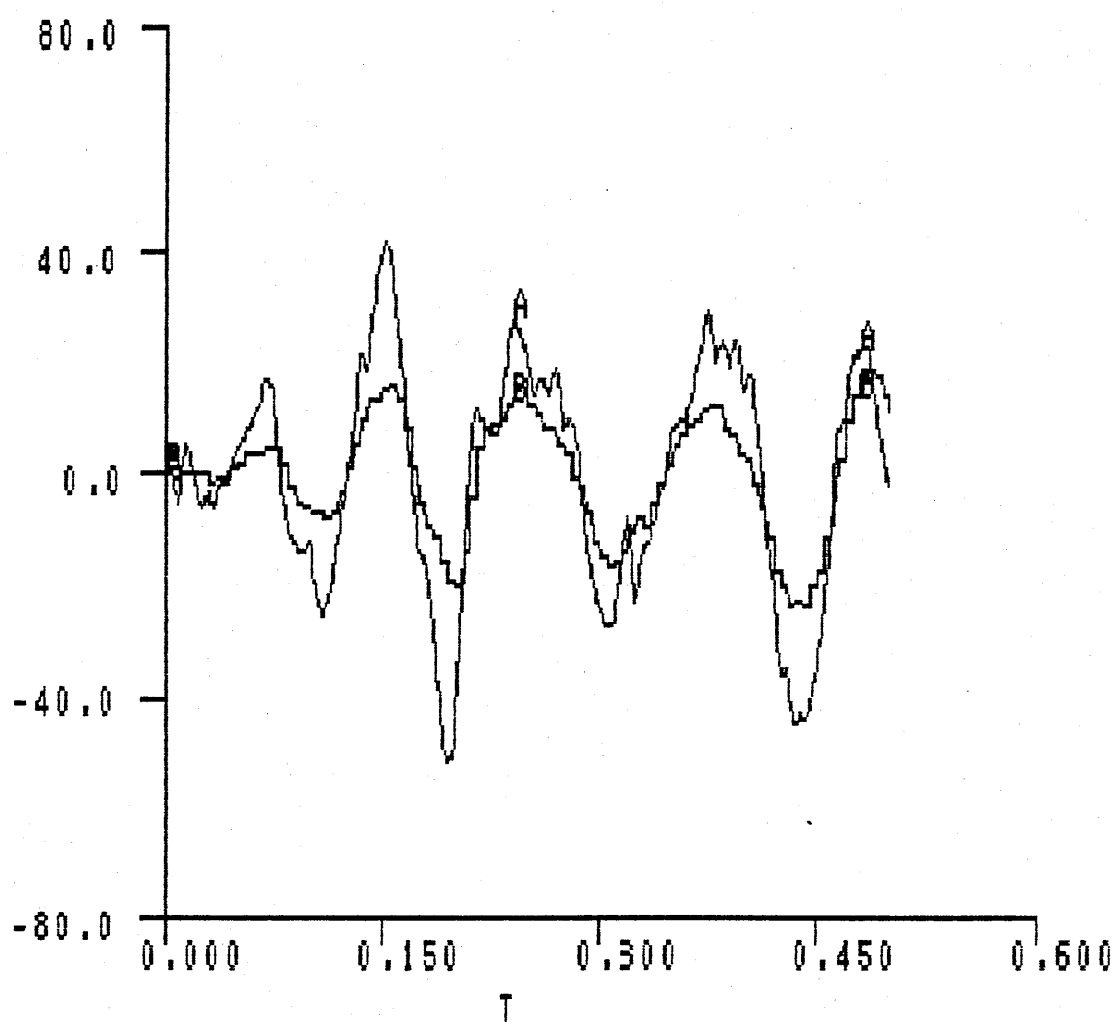


Fig 7.37 Augmented Outputs Kalman Filter (x_3 and $\hat{x}_3$)

7.4.3 LIMITED STATE ESTIMATION:

A third strategy to overcome the unobservability problem, is based on limited state estimation. Essentially, the unobservable state (x_1) is simply ignored as far as the Kalman filter and the optimal control strategy are concerned, and the problems caused by this are dealt with otherwise (Sect 7.5.3). This strategy is represented in Fig 7.38.

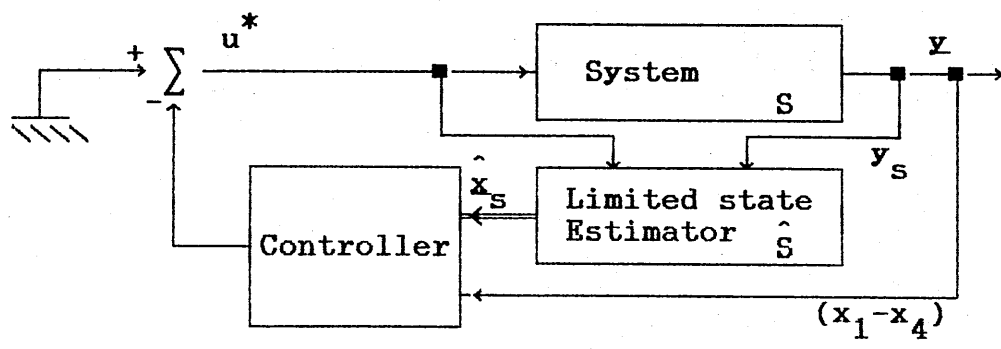


Fig. 7.38 Limited State Feedback Scheme

Recall the "A", "B" & "G" matrices in (7.14):

$$A = \begin{bmatrix} \emptyset & 1\emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & 1\emptyset & \emptyset & \emptyset \\ \emptyset & -86\emptyset & -112 & -1176 & -92 \\ \emptyset & \emptyset & \emptyset & \emptyset & 1\emptyset \\ \emptyset & \emptyset & \emptyset & -1\emptyset 5 & -85 \end{bmatrix}, \quad B = \begin{bmatrix} \emptyset \\ \emptyset \\ 444\emptyset \\ \emptyset \\ \emptyset \end{bmatrix}, \quad G = \begin{bmatrix} \emptyset \\ \emptyset \\ 1176 \\ \emptyset \\ 1\emptyset 5 \end{bmatrix}$$

The $\emptyset$'s in the 1st row and column of A allow us to consider the sub-system;

$$\dot{x}_s(t) = A_s x_s(t) + B_s u(t) + G_s w \quad \dots (7.23)a$$

$$y_s(t) = C_s x_s(t) + v \quad \dots (7.23)b$$

where;

$$x_s = \begin{bmatrix} x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T, \text{ and}$$

$$A_s = \begin{bmatrix} 0 & 10 & 0 & 0 \\ -860 & -112 & -1176 & -92 \\ 0 & 0 & 0 & 10 \\ 0 & 0 & -105 & -85 \end{bmatrix}, \quad B_s = \begin{bmatrix} 0 \\ 4440 \\ 0 \\ 0 \end{bmatrix}, \quad G_s = \begin{bmatrix} 0 \\ 1176 \\ 0 \\ 105 \end{bmatrix},$$

$$C_s = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

The eigen values of this are: $-56.0 \pm j73.9$, -15.0 , -70.0 .

In this case the 4'th order sub-system is observable and a Kalman filter can be designed.

In discrete form using 5 msec sampling time;

$$\phi_s = \begin{bmatrix} 0.9117E+00 & 0.3696E-01 & -0.1189E+00 & -0.1004E-01 \\ 0.3178E+01 & 0.4978E+00 & -0.4241E+01 & -0.3736E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.9886E+00 & 0.4055E-01 \\ 0.0000E+00 & 0.0000E+00 & -0.4258E+00 & 0.6439E+00 \end{bmatrix}$$

$$\psi_s = \begin{bmatrix} 0.4560E+00 \\ 0.1641E+02 \\ 0.0000E+00 \\ 0.0000E+00 \end{bmatrix}$$

The sub-optimal Kalman filter is used for this system considering the case as before:

$$Q = 10, \quad Q_d = 0.05, \quad R_d = \begin{bmatrix} 1 \end{bmatrix}$$

The optimal Kalman filter gain and error covariance matrices are:

$$K_f(\infty) = \begin{bmatrix} 0.3979E-01 \\ 0.5464E+00 \\ -0.1249E-01 \\ -0.2159E-01 \end{bmatrix}$$

$$P(\infty) = \begin{bmatrix} 0.5342E+00 & 0.1122E+00 & -0.4226E+00 & 0.3009E+00 \\ 0.1122E+00 & 0.6469E+02 & -0.7171E+00 & -0.2682E+01 \\ -0.4226E+00 & -0.7171E+00 & 0.4447E+00 & 0.8026E-01 \\ 0.3009E+00 & -0.2682E+01 & 0.8026E-01 & 0.8978E+01 \end{bmatrix}$$

For the random disturbance shown in Fig 7.39, Figs. 7.40 to 7.41 show the actual and estimated states of the sprung mass. The Kalman filter clearly generates acceptably good estimates of the states.

A control strategy using these states is considered in Sect 7.5.3.

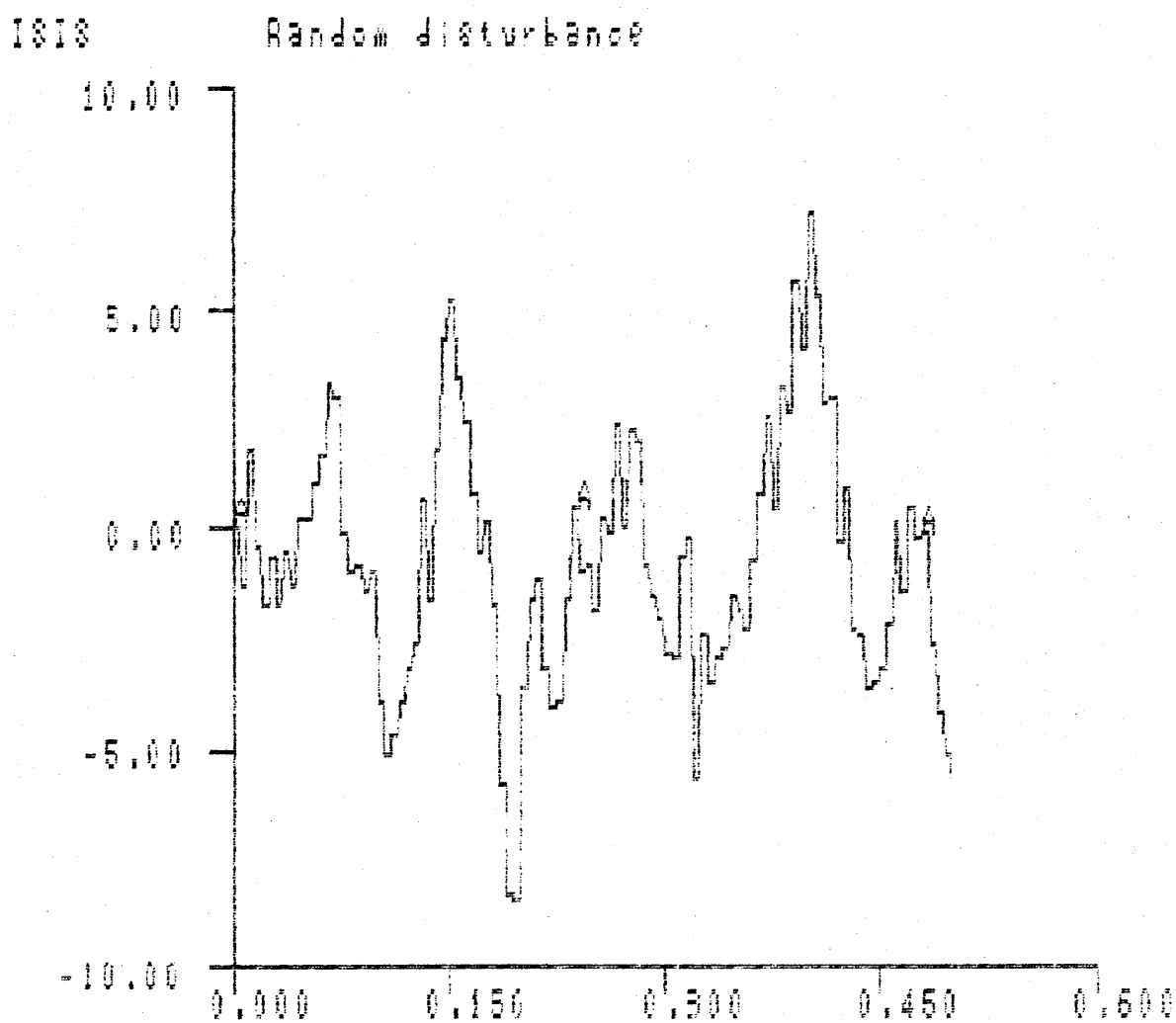


Fig 7.39 Random Disturbance

ISIS

Limited state FB optimal spring mass vel

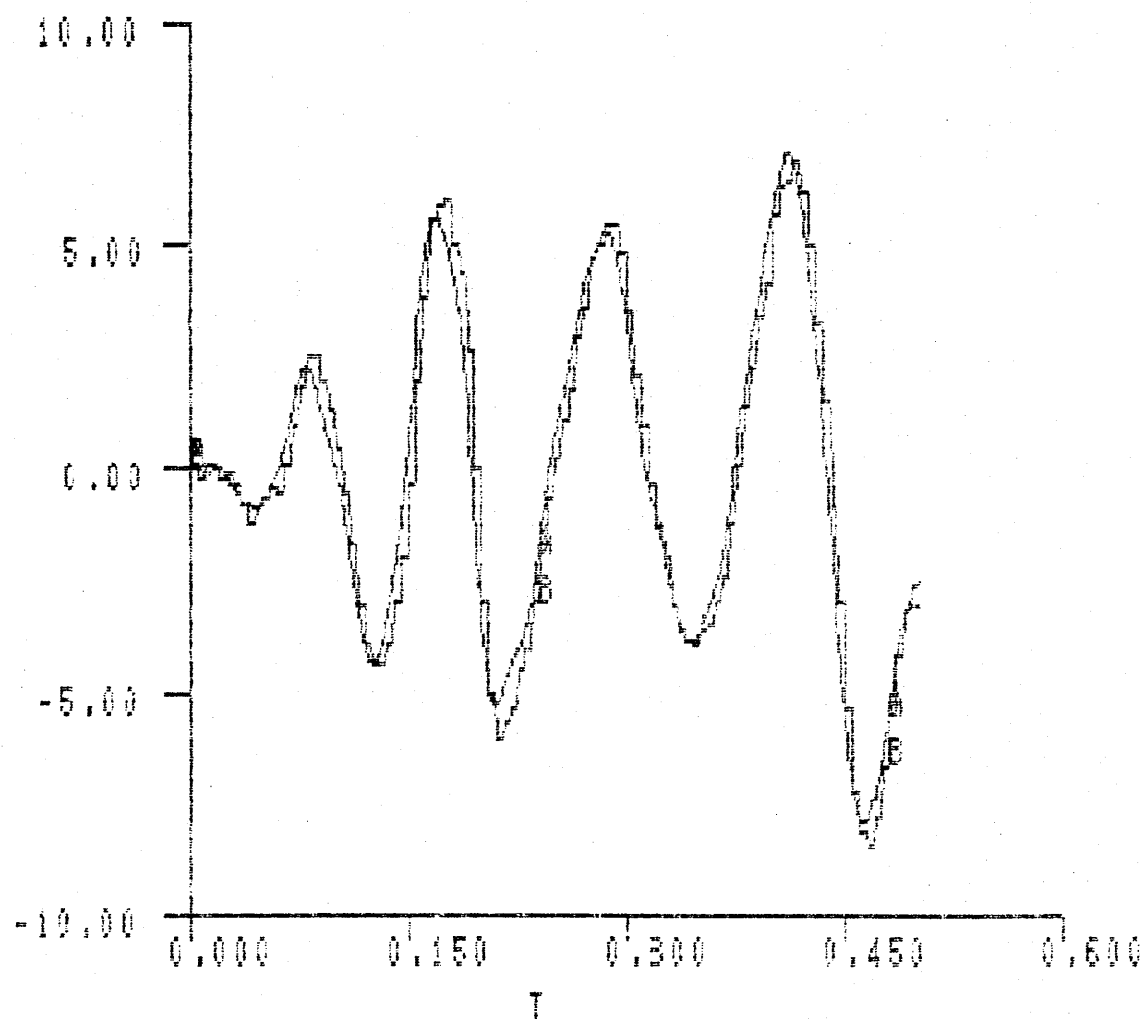


Fig 7.40 Limited States Kalman Filter (x_1 and $\hat{x}_1$)

1313

Limited state FB optimal sprung mass acc

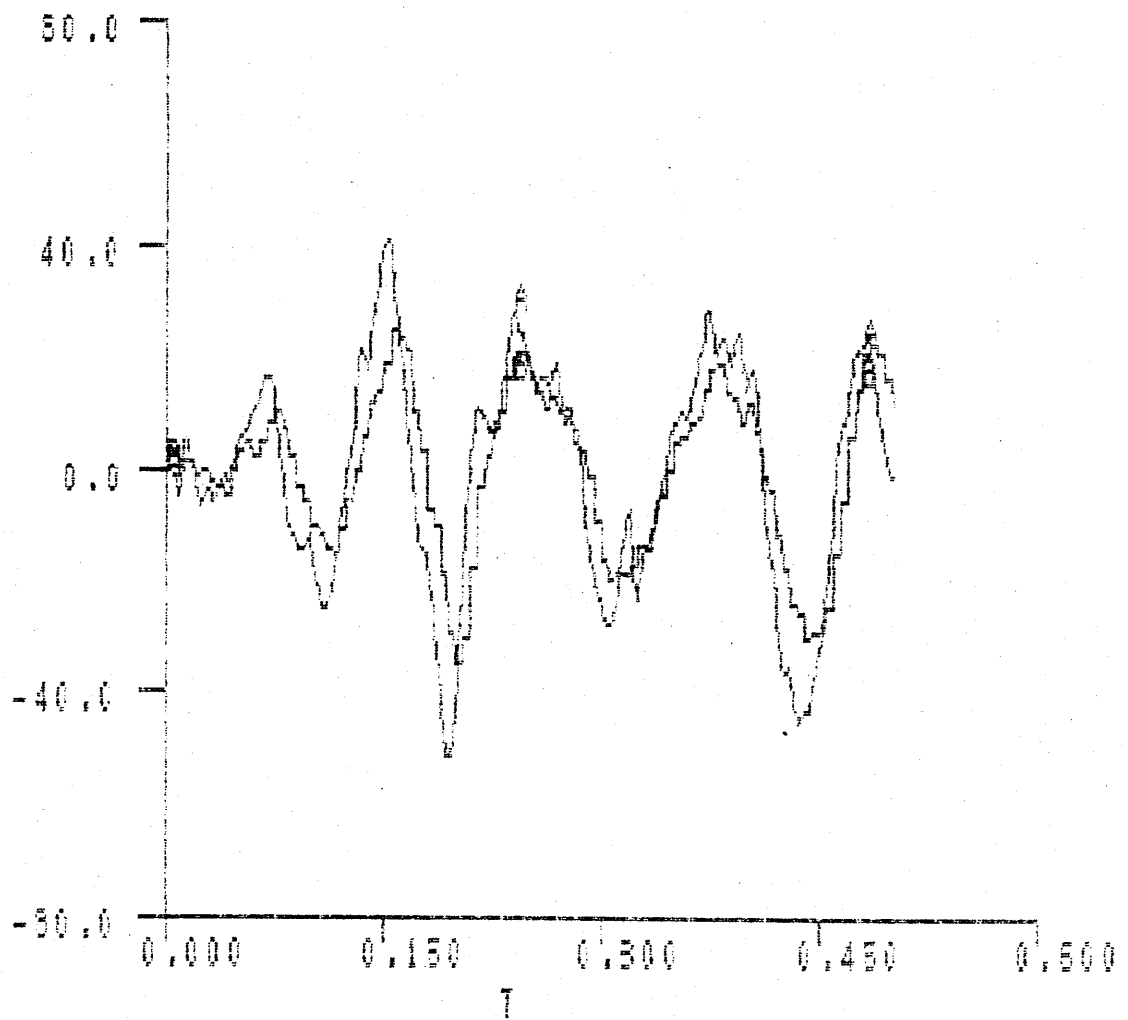


Fig 7.41 Limited States Kalman Filter (x_2 and $\hat{x}_2$)

7.4.4 REDUCED ORDER KALMAN FILTER:

The structure of the models used in Sects 7.4.1, 7.4.2 and 7.4.3 suggests that reduced order Kalman filters could be designed to estimate x_1 , x_2 and x_3 , fairly easily.

The model in Sect 7.4.1 (Eqns 7.20) could be regarded as a third order system representing (x_1, x_2, x_3) with input $u(t)$, and a non-white noise disturbance $w(t)$ corresponding to the effects of x_4 and x_5 .

$$\dot{x}(t) = \begin{bmatrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ -258 & -893.6 & -115 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 4440 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1207.5 \end{bmatrix} \bar{w}(t) \quad \dots (7.24a)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(t) + v(t) \quad \dots (7.24b)$$

The eigen values of this are: -3.0 , $-56.0 \pm j73.9$.

This gives us the opportunity to design a reduced order Kalman filter by treating $\bar{w}(t)$ as white noise.

This can not be done for the policy of Sect 7.4.2 (Eqn 7.22) since x_4 is required by the Kalman filter, and this affects x_5 .

However, the policy of Sect 7.4.3 can lead a second order Kalman filter, since x_2 , x_3 only are to be estimated. From Eqn 7.23, the reduced second order system is;

$$\dot{x}(t) = \begin{bmatrix} 0 & 10 \\ -860 & -112 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 4440 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1176 \end{bmatrix} w(t) \quad \dots (7.25a)$$

$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t) + v(t) \quad \dots (7.25b)$$

The eigen values of this are: $-56.0 \pm j73.9$.

This topic has not been developed further here.

7.5 OPTIMAL CONTROL STRATEGIES:

We now consider the various optimal control strategies which can practically be based on the Kalman filter designs of Sect 7.4. These are realized on the laboratory equipment using the relevant states directly (not generated by Kalman filter). The three strategies examined in Sects 7.5.1, 7.5.2, 7.5.3 correspond to the Kalman filters developed in Sects 7.4.1, 7.4.2 and 7.4.3 respectively.

7.5.1 CONTROL STRATEGY BASED ON APPROXIMATE KALMAN FILTER:

The Kalman filter of Sect 7.4.1 generates estimates of the system states with approximation to $\hat{x}_1$ which is accurate at "high" frequencies but not at "low" frequencies.

Any practical control strategy must also include a low frequency recovery feature so that the unsprung mass position retains a mean value $E(x_1 - x_4) = 0$. This can be achieved by measuring $(x_1 - x_4)$ with a potentiometer - which is quite practical on the laboratory equipment and in a quarter car model - filtering the "low" frequency content out of it, and using the result in the overall control strategy. This arrangement is shown in Fig 7.42.

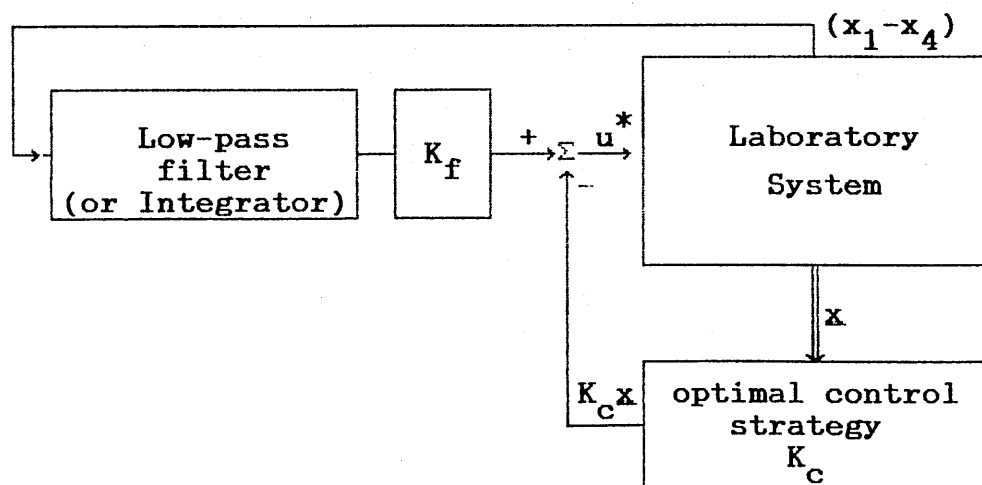


Fig. 7.42 Approximate State Control Strategy

After some experimentation, the low pass filter of Fig 7.43 was constructed.

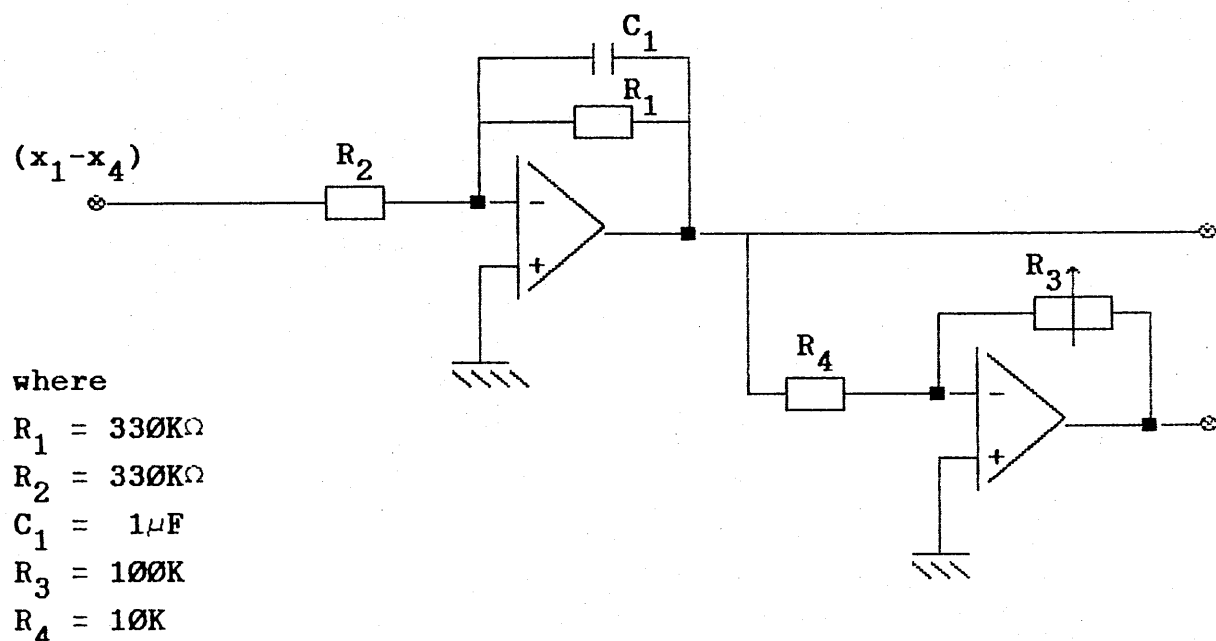


Fig. 7.43 Low Pass Filter Used in Outer F/B Loop

The effect of choosing such a filter is to treat frequency components below about 3 rad/sec as "low", and not within the authority of the optimal strategy. Those above 3 rad/sec are treated as "high" and are subject to the effect of the optimal strategy. This was confirmed by experiment to be reasonable.

The control strategy, ignoring x_4 and x_5 as explained in Sect 7.3, used here is:

$$K_c = \begin{bmatrix} 0.46 \\ 1.85 \\ 0.75 \\ 0 \\ 0 \end{bmatrix}, \quad K_f = 0.21 \quad \dots (7.24)$$

where K_c and K_f were found by experiment on the equipment to yield good results. Step function responses for the laboratory equipment are shown in Fig. 7.44.

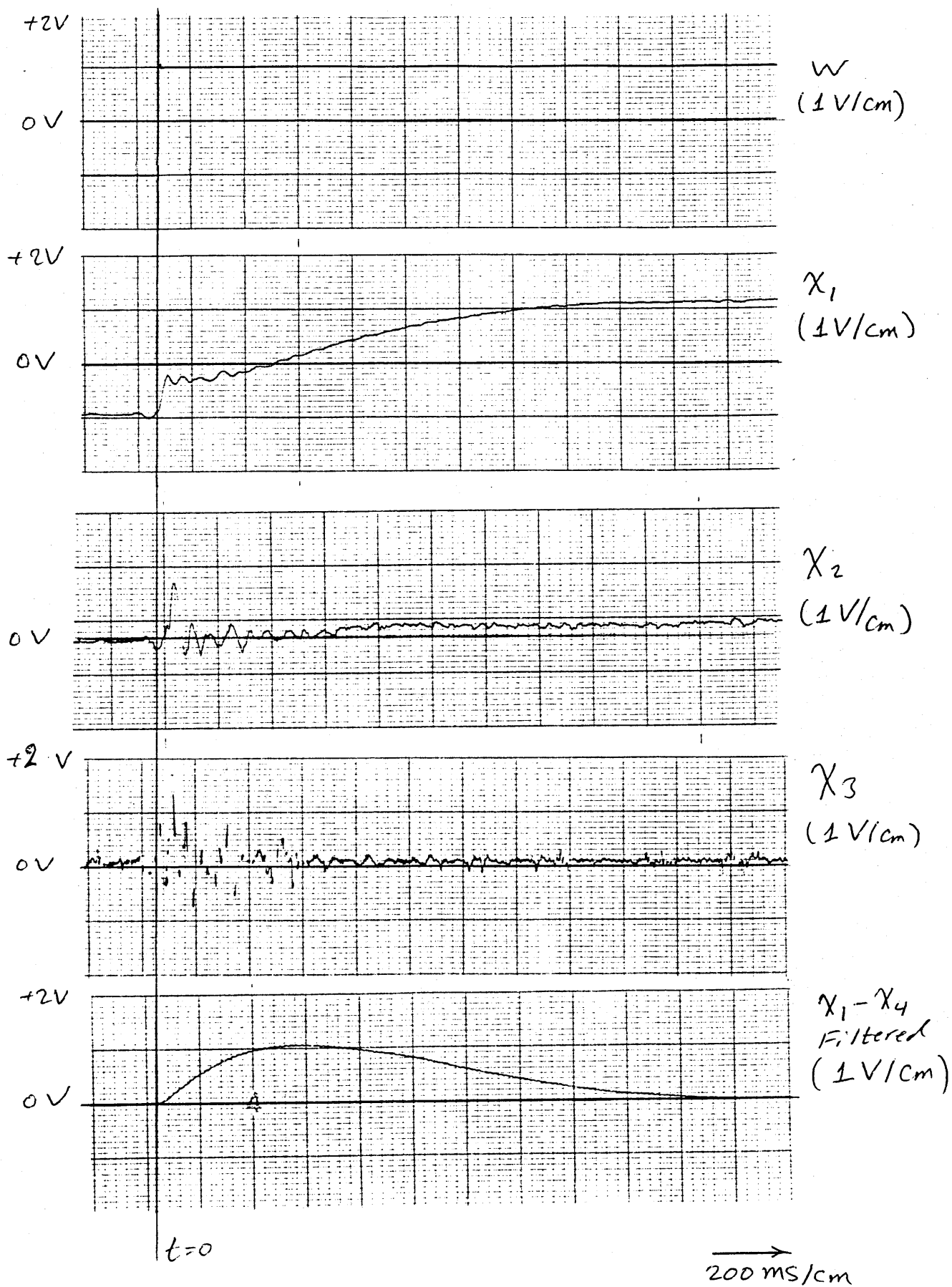


Fig. 7.44 Equipment Step Function Responses
Using Control Strategy Based on Approximate KF

It can be seen in Fig 7.44 that the sprung mass position x_1 does recover to $E(x_1 - x_4) = 0$ after about 1.6sec - which may be slightly slow - but can easily be altered by altering the feedback filter gain (Fig 7.43). The results are quite good with a maximum acceleration of 1.5v equivalent to 0.27ms^{-2} . This compares with about 28ms^{-2} for the uncontrolled system (taken from Fig 7.13).

7.5.2 CONTROL STRATEGY BASED ON AUGMENTED OUTPUTS:

In this case, the Kalman filter outputs are taken to include an estimate of x_1 , at the cost of a somewhat more complex Kalman filter using $(x_1 - x_4)$, x_3 as system outputs (Sect 7.4.2).

The idea of Sect 7.4.2, must be combined with a low frequency strategy designed to ensure that $E(x_1 - x_4) = 0$. This is shown in Fig 7.38, though in this case x_1 , and not approximation to it, is fed to the optimal control strategy, and the filter of Fig 7.42 must be replaced with an integrator with a low gain to remove the dc content of the x_1 .

Using the control strategy, found by experiment to give good results, as:

$$K_c = \begin{bmatrix} 1.47 \\ 2.67 \\ 0.85 \\ 0 \\ 0 \end{bmatrix}, \quad K_f = 0.78, \quad K_I = 6 \quad \dots (7.25)$$

where K_I is the integrator gain (volts/sec/volt).

Step function responses for this are shown in Fig 7.45.

The results shown here are promising and show a distinct reduction in the sprung mass acceleration. Compared with Fig 7.44, the maximum being equivalent to 0.18ms^{-2} .

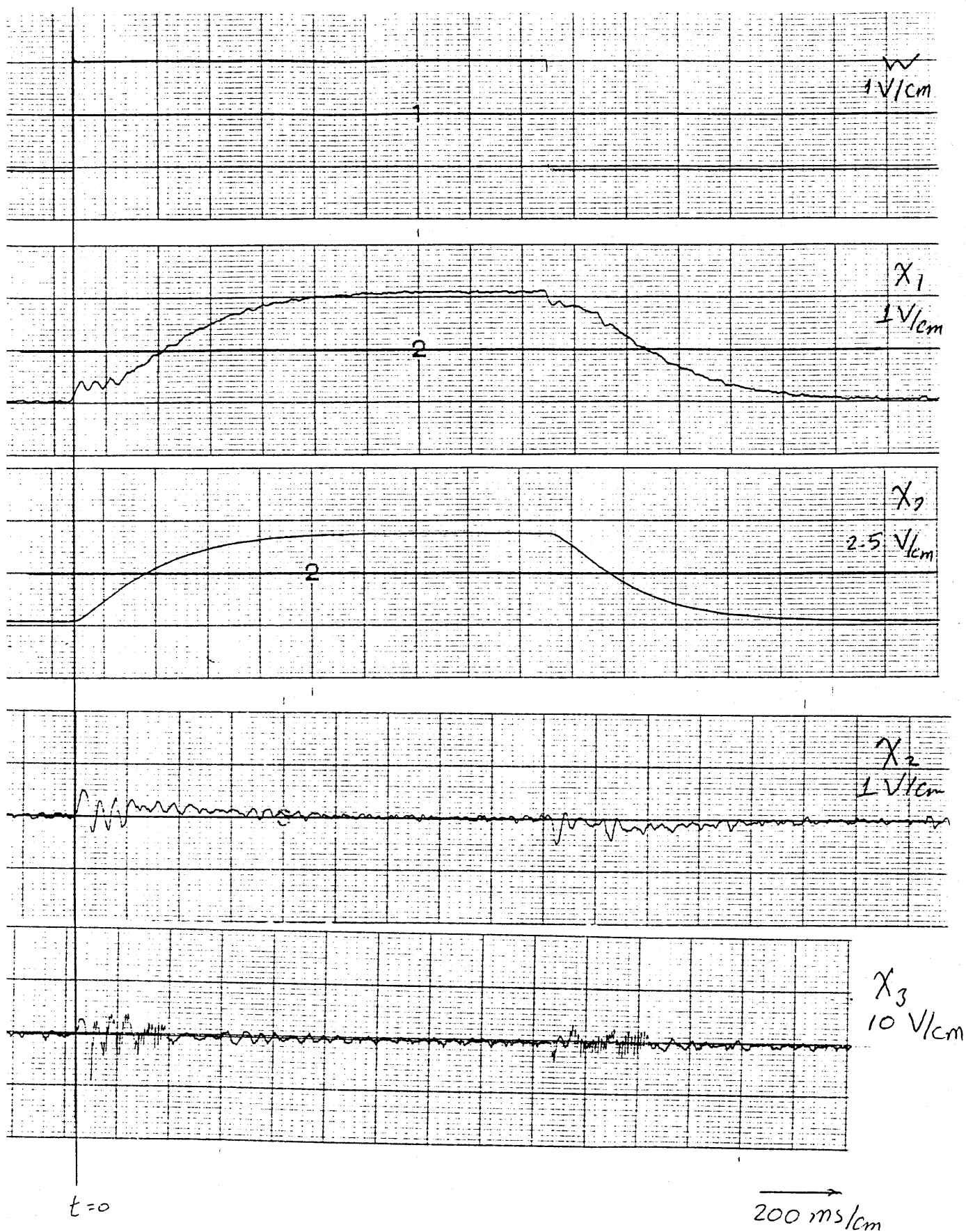


Fig. 7.45 Equipment Step Function Responses
Using Control Strategy Based on Augmented Outputs KF

7.5.3 CONTROL STRATEGY BASED ON LIMITED STATE ESTIMATION:

The reduced order Kalman filter of Sect 7.4.3 generate estimates of x_2, x_3, x_4, x_5 (not x_1), and a corresponding optimal control strategy is designed to minimize a cost function of x_2 and x_3 . As with the previous strategies, an outer low frequency loop is required to ensure that $E(x_1 - x_4) = 0$. The overall strategy is like that shown in Fig 7.42 with x replaced with $x_s = [x_2 \ x_3 \ x_4 \ x_5]^T$.

The feedback strategy is found in this case using the partial state feedback algorithm (Appendix-F). Let;

$$\bar{Q}_c = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{R}_c = \begin{bmatrix} 1 \end{bmatrix}$$

This limits the variances of the velocity and acceleration of the sprung mass, leaving the unsprung mass uncontrolled as before.

The optimal feedback gain vector is:

$$\bar{K}_c = \begin{bmatrix} k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 3.2 \\ 0.85 \end{bmatrix}, \quad K_f = 0.29 \quad \dots (7.26)$$

Fig. 7.46 shows the closed loop system responses. Again these results looks promising, with the sprung mass states well controlled, and the maximum acceleration being about 0.9ms^{-2} . This is still good compared with the uncontrolled 28ms^{-2} , but is not as good as the previous cases.

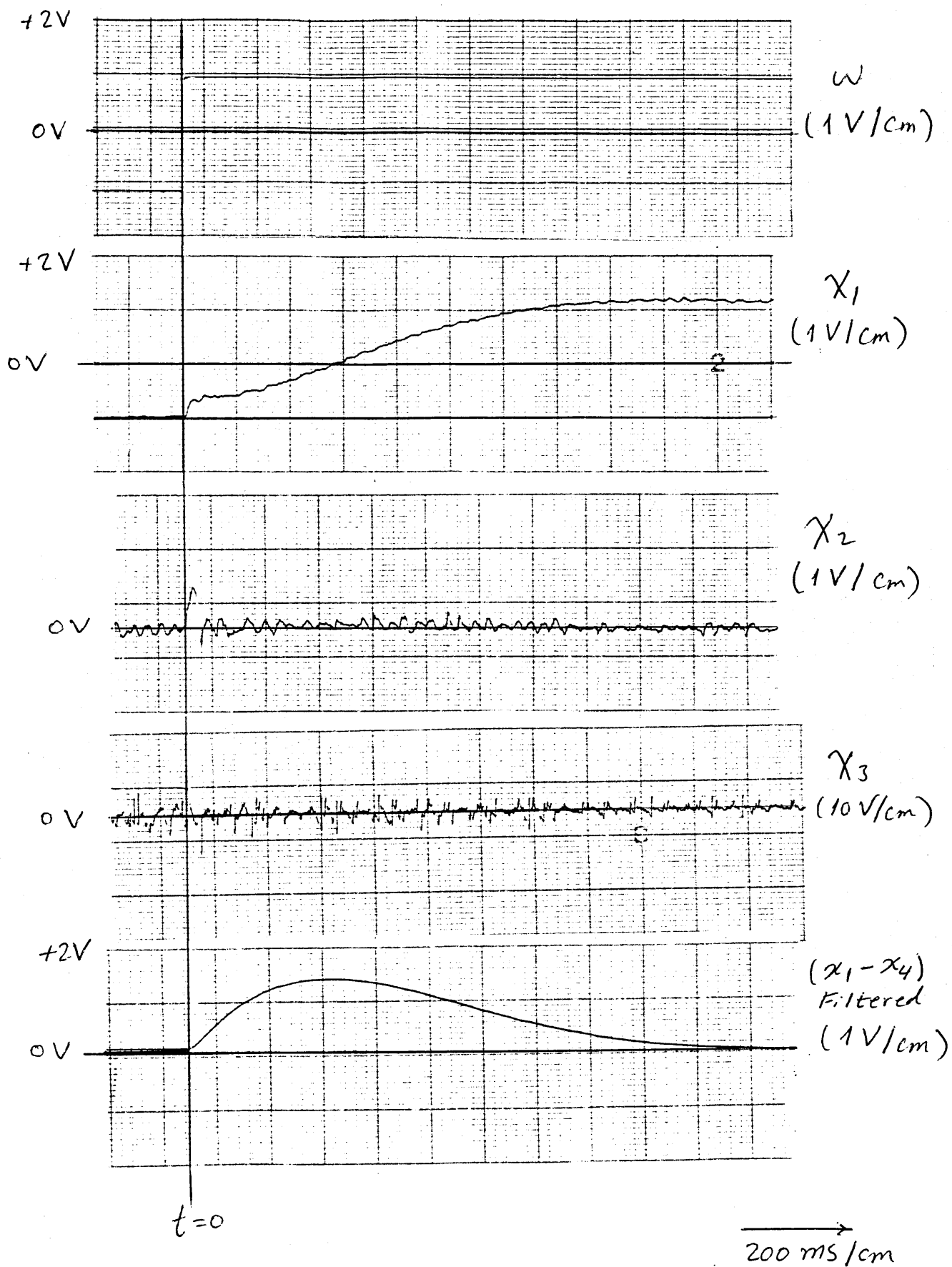


Fig. 7.46 Equipment Step Function Responses
Using Control Strategy Based on Limited States KF

7.6 CONCLUSION:

In this chapter a laboratory equipment is used as an experimental model. This is represented by a 5th order model using sprung mass acceleration as the only output. A problem of observability appears when a Kalman filter is designed to estimate the system states to be used for the optimal control strategy and different suggestions for solving this problem are examined. Sub-optimal control strategies corresponding to these strategies are also examined and compared. Measurement of the relative sprung and unsprung mass positions is found to be necessary for all control strategies. The results in all these cases look quite promising, with the best results being achieved using the most elaborate strategy.

It has proved impossible in the time available to realize all the Kalman filters and implement the entire strategies based on their estimates. This clearly is the next step as far as the laboratory equipment is concerned.

REFERENCES:

- [1] T. Yoshimura, "An Active Vertical Suspension For Track/Vehicle Systems", Journal of Sound and Vibration (1986) 106(2).
- [2] D. A. Wilson, R. S. Sharp, and S. A. Hassan, "The Application of Linear Optimal Control Theory to the Design of Active Automotive Suspension", Vehicle system dynamics (1988).
- [3] J. A. Borrie, "Modern Control Systems", Prentice/Hall International, 1986.
- [4] Yu-Chi Ho and Arthur E. Bryson, "Applied Optimal Control, The Optimization, Estimation, and Control", Hemisphere Publishing Corp 1975.
- [5] A. G. Thompson, "Optimal and Suboptimal Linear Active Suspensions for Road Vehicle", Vehicle System Dynamics, vol 13 (1984).
- [6] A. Hac, "Suspension Optimization of a 2-DOF Vehicle Model Using a Stochastic Optimal Control Technique", Journal of Sound and Vibration (1985) 100(3).

C H A P T E R 8

SUMMARY, CONCLUSION AND SUGGESTION FOR FURTHER WORK

8.1 INTRODUCTION:

In this project, a number of models and control strategies based on Kalman filters have been suggested for vehicle active suspension systems. Problems of unobservability and uncontrollability have been exposed, and solutions suggested.

A number of conclusions can now be drawn and further work specified to complete the project.

The project is summarized in Sect 8.2 and conclusions are drawn in Sect 8.3. Suggestions for further work are outlined in Sect 8.4.

8.2 SUMMARY OF WORK TO DATE:

The work to date may be summarized as follows:

(1) Some investigation of road surface modelling have been undertaken, and some filters prepared (in software form) for simulating various road surfaces.

(2) An electro-hydraulic laboratory equipment for simulating the main features of a quarter car suspension system has been designed and built. The behaviour of this has been modeled using a linear model, and this model has been realized on the computer.

(3) An electronic circuit, incorporating an 8086 microprocessor has been built. This is designed to allow the proposed Kalman filters and control strategies to be realized.

(4) Various computer models of suspension systems have been examined with a view to implementing active control strategies based on Kalman filters. Problems of unobservability and uncontrollability have been exposed.

(5) Various Kalman filter strategies which take account of the unobservability problem have been proposed and examined using computer simulations. These are sub-optimal Kalman filters, some of reduced order. However, some attention has been paid to optimal filters (with variable gains) which may be needed in some final versions of the filters.

(6) Various optimal control strategies suited to the Kalman filter outputs have been suggested and tested by computer simulation and on the laboratory equipment. Three of these seem quite promising.

(7) It has not been possible in the time available to test the Kalman filters on the laboratory equipment, or, consequently to test the control strategies using them. Nor have rigorous computer models of quarter car suspension systems with the Kalman filter designs and control strategies been tested. These are clearly the next step to the work.

8.3 CONCLUSIONS:

It can reasonably be concluded from the work:

(1) That it is feasible to construct fairly simple Kalman filters for active suspension systems, the transducers required being an accelerometer and linear potentiometer for each quarter car system. Suitable Kalman filter structures have been proposed.

(2) That it is feasible to realize reasonable simple optimal control strategies using the Kalman filter outputs, the actuators required being one hydraulic ram per quarter car, connected in parallel with a spring-damper arrangement. Suitable control strategies have been proposed.

(3) The necessary filter and control strategies can be realized in a fairly simple microprocessor circuit, incorporating an Intel 8086 microprocessor. This seems adequate for the suspension of a whole vehicle. A suitable electronic circuit has been proposed.

8.4 FURTHER WORK:

Clearly, more work is needed to bring this project to a successful conclusion; the nature of this has been revealed by the work to date. The project is seen to proceed further as follows.

(1) The Kalman filters are tested on the laboratory equipment - to confirm the simulation results.

(2) The control strategies on the laboratory equipment are tested using the Kalman filter outputs.

(3) The more elaborate quarter car computer models of Chapter 6 are developed and the Kalman filter and control strategies tested against them.

(4) Either the laboratory equipment is enhanced to suit these quarter car model features more accurately, or a new equipment, ideally using a vehicle, is constructed. This is used to test the designs, including the electronics.

(5) More elaborate ideas, involving whole vehicle suspension systems, and dealing with their non-linearities, can be tried.

A P P E N D I X A
CONTINUOUS AND DISCRETE TIME DYNAMIC SYSTEMS REPRESENTATION

As described in Chapter 4, a Linear Time Invariant dynamic system can be described in vector differential equations form as:

$$\left. \begin{aligned} \frac{dx}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \right\} \quad \dots (A.1)$$

where A, B, C, D are constant.

Throughout this thesis, we deal mainly with discrete (or sampled) dynamic systems;

$$\left. \begin{aligned} x(n+1) &= \phi x(n) + \psi u(n) \\ y(n) &= Cx(n) + Du(n) \end{aligned} \right\} \quad \dots (A.2)$$

where ϕ , ψ , C, D are constant.

A.1 SAMPLED DATA SYSTEMS:

One type of discrete system of particular interest is the sampled data system. This is a continuous time system buffered with analogue-to-digital and digital-to-analogue converters (ADCs and DACs) as shown in Fig. A.1.

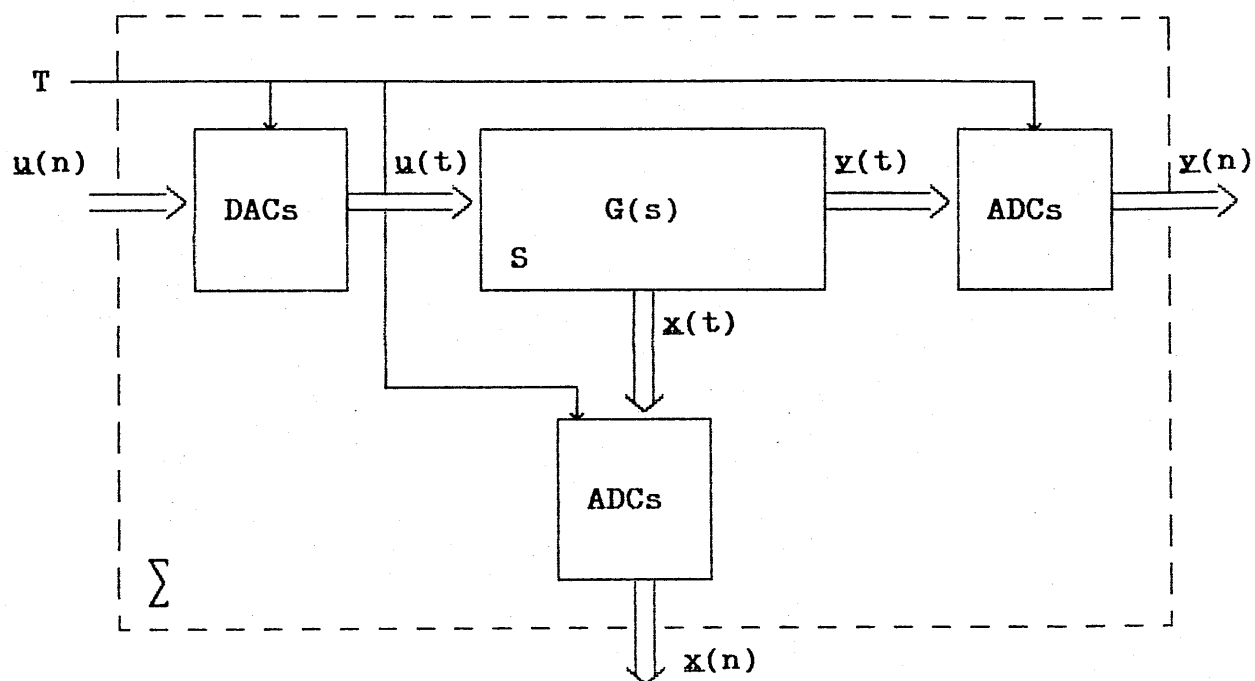


Fig. A.1 Sampled Data System.

If the state equations of S are known and the DACs and ADCs are clocked simultaneously, discrete state equations for $\sum$ may readily be found from the solution of the continuous time equations for the time $((n+1)T)$, taking $x(nT)$ as initial conditions and $u(nT)$ as the input:

$$\left. \begin{aligned} x((n+1)T) &= e^{AT} x(nT) + \int_{nT}^{(n+1)T} e^{A\tau} B u(nT) d\tau \\ y(nT) &= C x(nT) + D u(nT) \end{aligned} \right\} \dots (A.3)$$

Rewriting this in the standard form of equation (A.2), where;

$$\left. \begin{aligned} \phi &= e^{AT} \\ \psi &= \int_0^T e^{A\tau} B \, d\tau \quad (u(nT) \text{ is constant, } nT < \tau \leq (n+1)T) \end{aligned} \right\} \dots \text{ (A.4)}$$

C and D matrices are unchanged.

ϕ and ψ in Eqn (A.4) can be found either by the direct use of series expansions for the required numbers of terms depending on the accuracy needed, or using Runge-Kutta method. In this project the 4'th order Runge-Kutta method is used as described in Sect A.2.

A.2 RUNGE-KUTTA METHOD:

This method is usually used for solving differential equations based on the Taylor expansion series:

$$x((n+1)h) = x(nh) + hx(nh) + \frac{h^2}{2!} x''(nh) + \frac{h^3}{3!} x'''(nh) + \dots \quad \text{(A.5)}$$

Commonly, four terms in this series are used to derive the fourth-order Runge-Kutta algorithm;

$$k_1 = hf(x(nh), nh)$$

$$k_2 = hf(x(nh) + \frac{1}{2}k_1, nh + \frac{1}{2}h)$$

$$k_3 = hf(x(nh) + \frac{1}{2}k_2, nh + \frac{1}{2}h)$$

$$k_4 = hf(x(nh) + k_3, nh + h)$$

Then:

$$x(n+1)h = x(nh) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \dots \text{ (A.6)}$$

In the algorithm above, since $x((n+1)h)$ is found from $x(nh)$, the starting data $x(0)$ are sufficient to allow the algorithm to proceed.

In this way, ϕ and ψ matrices in Eqn. (A.4) are calculated using the 4'th order Runge-Kutta integrator algorithm (A.6), using the following differential equations:

$$\begin{aligned} d\phi/dt &= A\phi \\ d\psi/dt &= \phi B = e^{At} B \end{aligned} \quad \dots (A.7)$$

The algorithm starts from initial values for ϕ = unity matrix, and $\psi = \emptyset$, for "h" represents the calculation interval.

The Fortran program DISCR is used to calculate ϕ and ψ matrices, using the algorithm described above, given A, B matrices and T sampling time.

REFERENCES :

- [1] J. A. Borrie, "Modern Control Systems", Prentice/Hall International, 1986.

APPENDIX B

PARTS LIST FOR THE MICROPROCESSOR HARDWARE

Table B.1 - PARTS LIST FOR THE 8086 CPU CARD :

Device name	Qty	Function	Comments
▶ 8086	1	CPU	16-bit Microprocessor
▶ 8284	1	Clock generator/driver	3 clock outputs; 1) CLK, MOS level. 2) PCLK, TTL level. 3) OSC, Osc. output.
▶ 8282	3	Un-idirectional 8-bit I/O latch buffers	Address Bus Controller
▶ 8286	2	Bi-directional Bus Transceiver	Data Bus Controller
▶ 6116	2	RAM	Size 2048 words x8bits Access time 150ns max.
		Memory	
▶ NMC27C256Q /20	2	EPROM	Size 32768 x 8 CMOS 200ns access time
▶ 2521	4	decoders	High speed TTL
▶ 8259A	1	Programmable Interrupt Controller	8 Maskable Interrupts Controller
▶ 8253	1	Programmable Interval Timer	3 software programmable counters
▶ 74LS138	1	8-Line decoder	High speed Schottky TTL
▶ 74LS32	1	logic OR gates	6 gates per chip 2 input for gate

Table B.2- PARTS LIST FOR THE ANALOGUE BOARD :

Device name	Qty	Function	Comments
▶ 7570	1	Analogue to Digital Converter	CMOS 10-Bit 20 μ s Conversion time
▶ 7522	1	Digital to Analogue Converter	CMOS 10-bit
▶ DG508	2	Multiplexers	8 channels
▶ 584	1	ADC voltage reference	+10V. ref.
▶ 311	1	Comparater	Op-Amp.
▶ 741	16	Amplifiers	
▶ 74LS27	1	Logic NOR gates	3 input for gate 4 gate per chip
▶ 74LS04	1	Logic NOT gates	6 gates per chip
▶Miscellaneous		Resistors, Capacitors Diodes,...	as shown in Figs. 5.2, 5.3

Table B.3- PARTS LIST FOR THE KEYBOARD DISPLAY BOARD :

Device name	Qty	Function	Comments
▶EAX20027AR	1	2x20 Dot Matrix LCD	display
▶ Keypad	1	4x4 keys	0 -9 & A - F
▶ 74C992	1	Keyboard Driver	
▶ 74LS74	1	divide by 2	Dual-D-type Flip-Flop
▶ 74LS244	1	3 state Buffer	
▶ 74LS08	1	Logic AND gate	2 inputs for gate

A P P E N D I X C
THE 8086 μ P AND THE PERIPHERALS
AS USED IN THIS PROJECT

This appendix gives a brief description of the main devices and their operation used for the hardware discussed in Chapter 5. For full details, devices data sheets should be referred to.

C.1 8086 IN MINIMUM MODE:

In the minimum mode, as used in this project, ($MN/\overline{MX}$ pin strapped to +5V), the CPU supports small, single-processor systems that consist of a few devices and that use the system Bus rather than support the MultibusTM architecture. In the minimum mode, the CPU itself generates all Bus control signals ($DT/\overline{R}$, $\overline{DEN}$, ALE and $M/\overline{IO}$) and the command output signal ($\overline{RD}$, $\overline{WR}$ or $\overline{INTA}$).

C.2 EXTERNAL MEMORY ADDRESSING:

The 8086 CPU has a 20-bit Address Bus and is capable of accessing 1 megabyte of memory in which any two consecutive bytes can be accessed as a 16-bit data word. As shown in Figure C.1 the memory address space is physically divided into two banks of up to 512K bytes each.

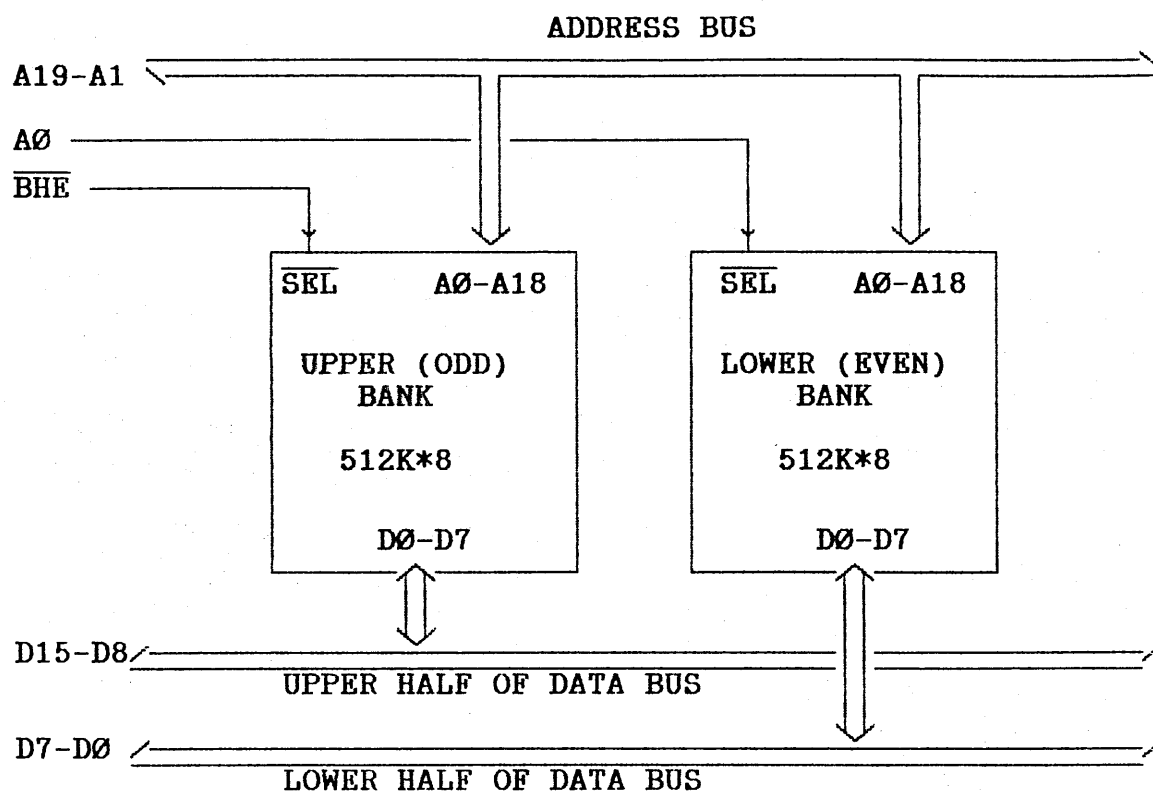


Figure C.1 8086 Memory Interface

One bank of memory is associated with the lower half of the CPU's 16-bit Data Bus (data bits D7-D0), and the other bank is associated with the upper half of the Data Bus (data bits D15-D8).

Address bits A1 through A19 are used to address a specific byte location in both the upper and lower banks, and the A0 is used in memory bank selection (i.e. in the memory decoding circuitry). The lower bank, which contains even address bytes, is selected when A0=0. The upper bank, containing odd address bytes (A0=1), is selected by a separate signal, Bus High Enable ($\overline{\text{BHE}}$) from the CPU.

Fig. C.2 shows the logic circuit design of the EPROMs. The same circuit can be applied for the RAMs case by adding $\overline{\text{WR}}$, CPU control signal for writing access to the memory.

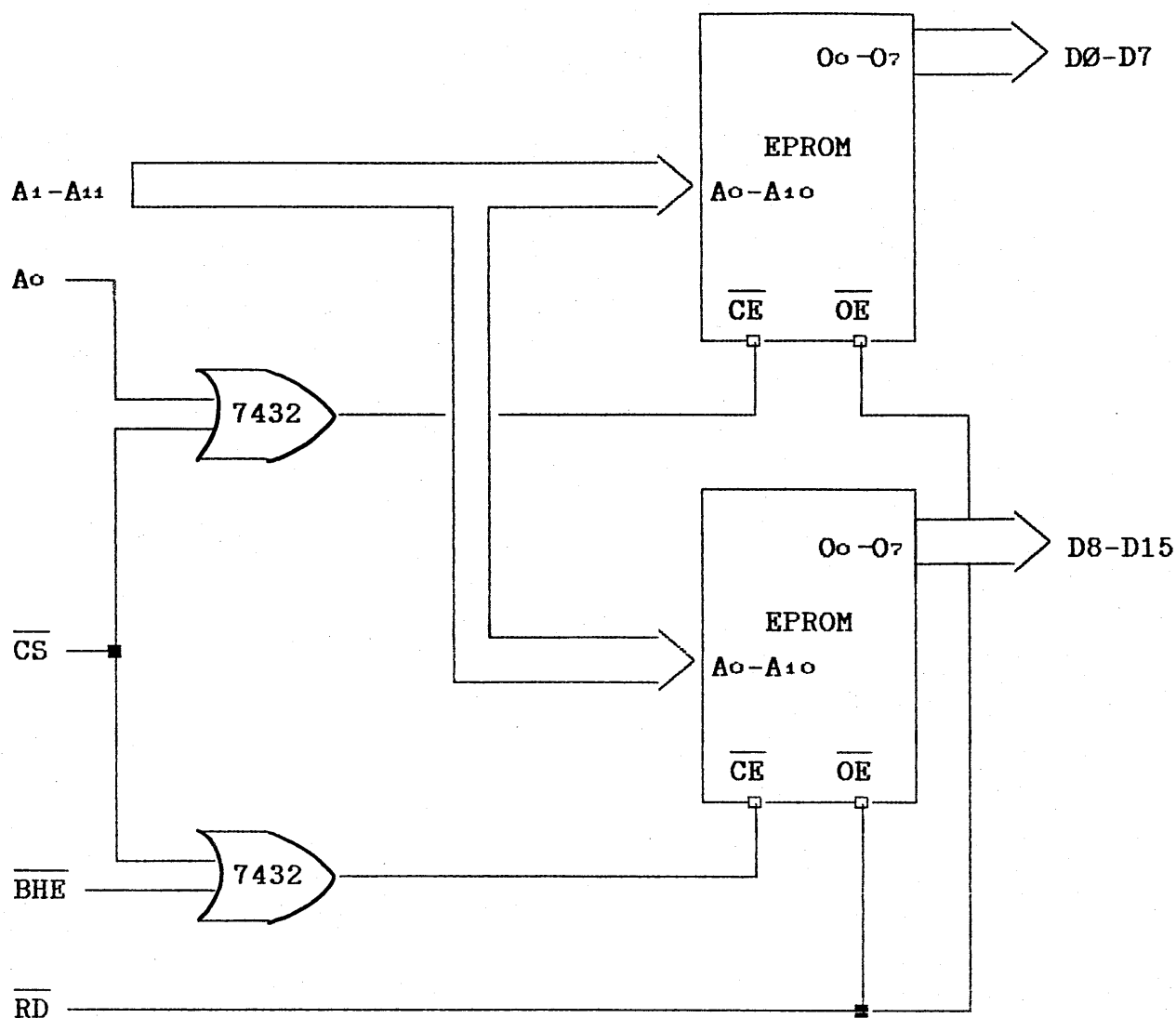


Fig C.2 EPROM Interface with the CPU

C.3 8259A PROGRAMMABLE INTERRUPT CONTROLLER :

The "8259A" IC is programmed to interrupt the CPU on the interrupt request from "8253" Timer or keyboard, etc, and also to generate interrupt vectors for the "8086" μP to indicate the start addresses of interrupt service routines. 8 interrupts can be used, of which 2 are used in our circuit.

The "8259A" generates an interrupt request signal on its "INT" pin after one or more of its IR0, IR1,...,IR7 lines are pulled high by an interrupting device after interrupt priority checks. IR0 is programmed to have the highest priority. "8259A" after requesting for interrupt, waits for negative pulse to appear on its "INTA" input pin after which it resets its registers and waits for a second negative interrupt acknowledge pulse from the CPU, then it releases an 8 bit interrupt vector code onto the Data Bus, which sets a pointer to the interrupt table in the (RAM) from which 8086 μ P can read the start address of the Interrupt Service Routine associated with the interrupting device. In this case, the IR1 is connected to OUT2 of the 8253 Timer.

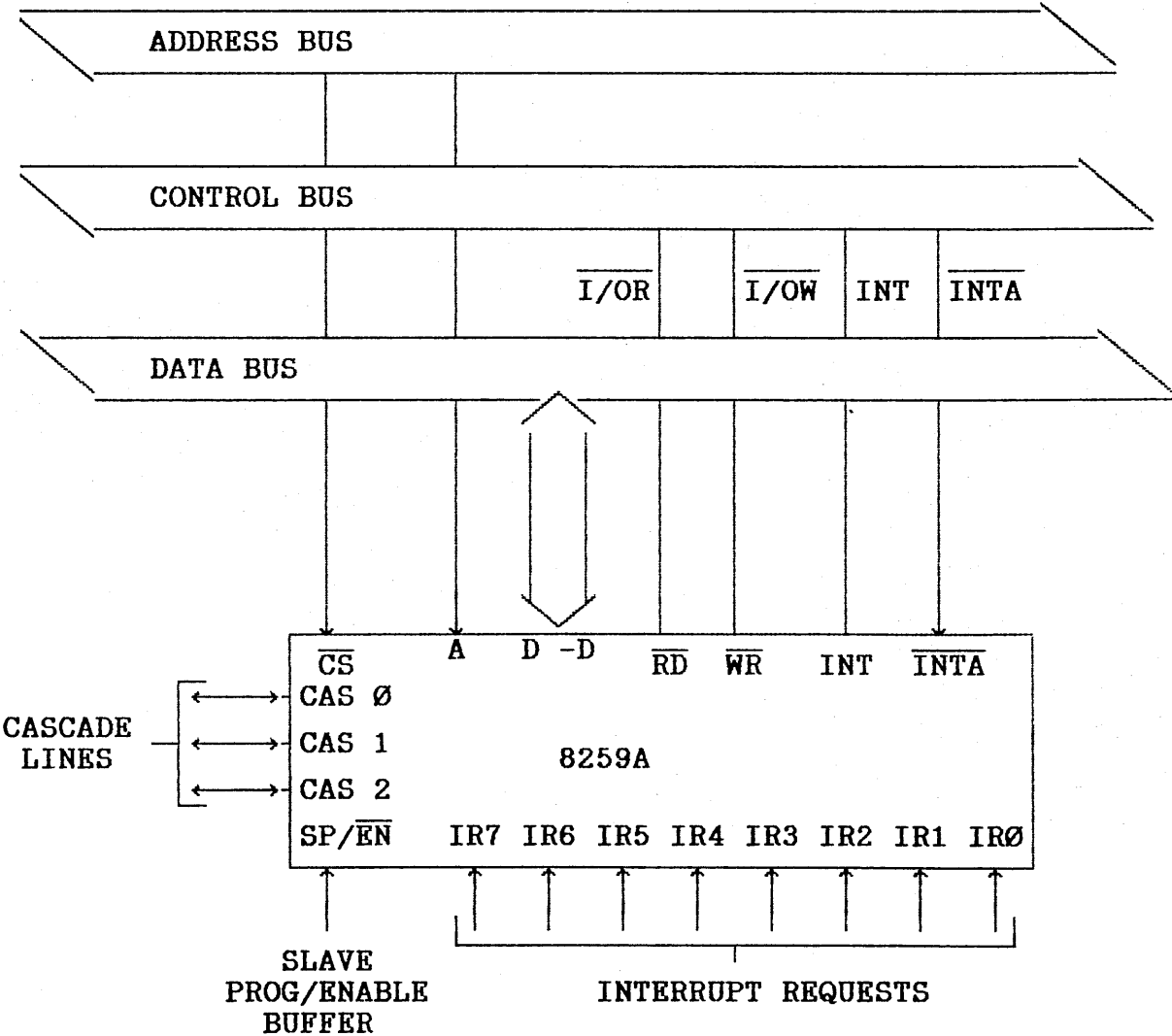


Fig. C.3 The 8259A Programmable Interrupt Controller

C.4 THE REAL TIME CLOCK 8253 :

The Intel 8253 is a Programmable Interval Counter/Timer chip. It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable. Each counter may be programmed in one of five possible modes of counting and generates a pulse at the termination of its count on its "OUT" pin.

The clock signal used is the Peripheral Clock (PCLK) from the clock generator "8284" device. The frequency of PCLK is 2MHz, so this clock can be applied directly to the CLK input signal of the "8253" device.

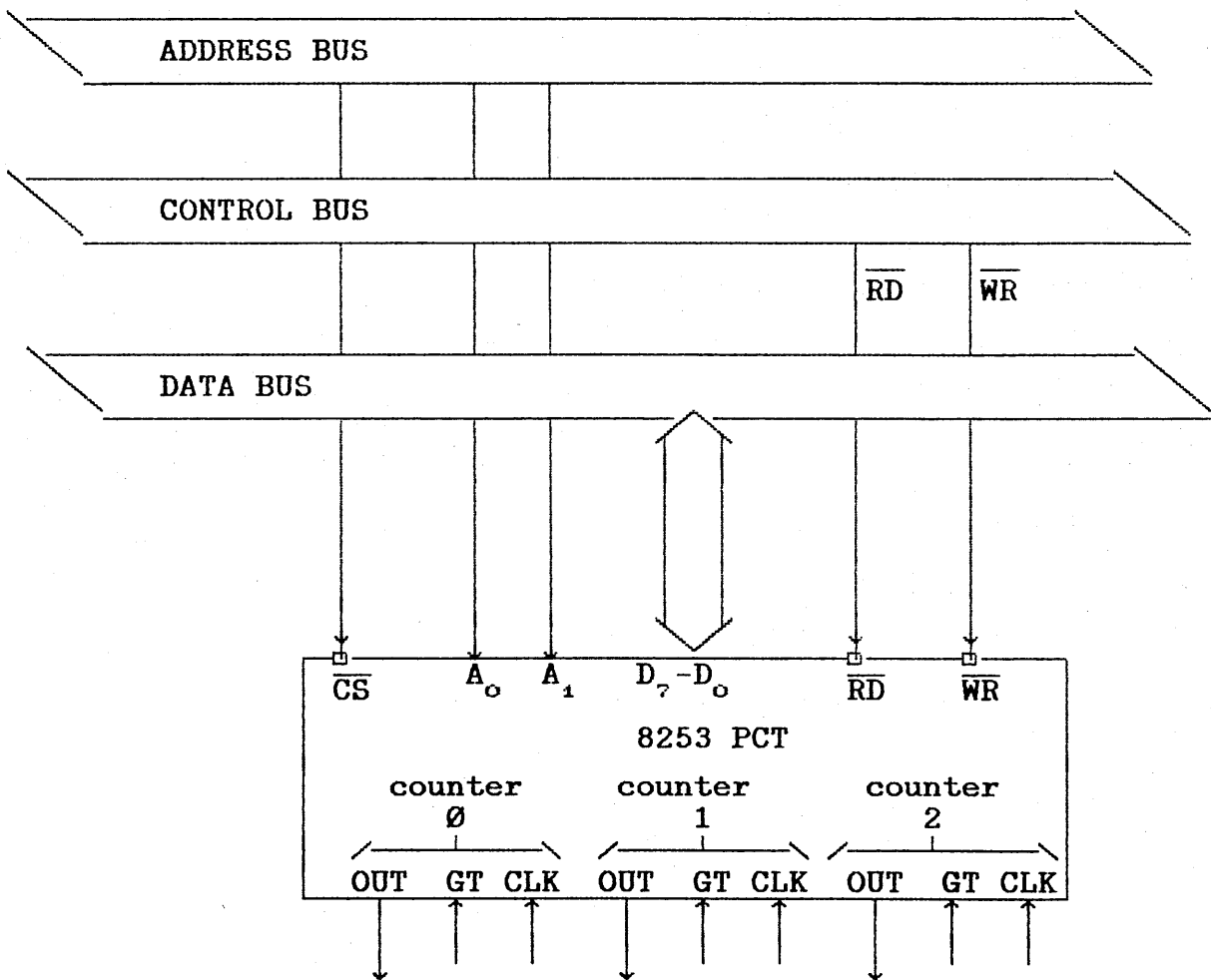


Fig. C.4 PCT 8253 System Interface

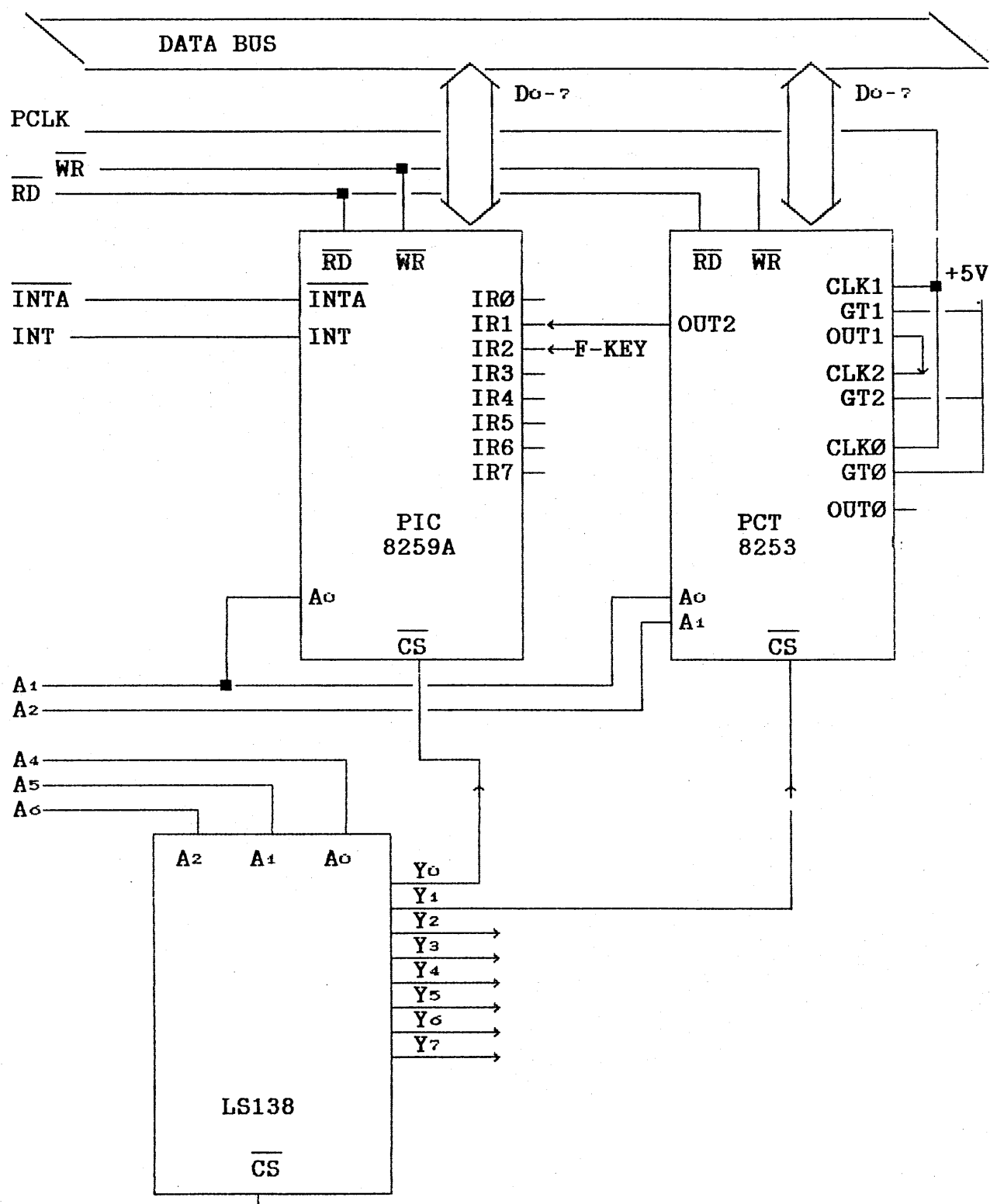


Fig. C.5 The PIC and the PCT Connection As Used in This Project

C.5 AD7570 A/D CONVERTER:

The AD7570 is a monolithic CMOS 10-bit successive approximation A/D converter requiring only an external comparator, reference and passive clocking components.

A functional diagram of AD7570 is shown in Fig C.6.

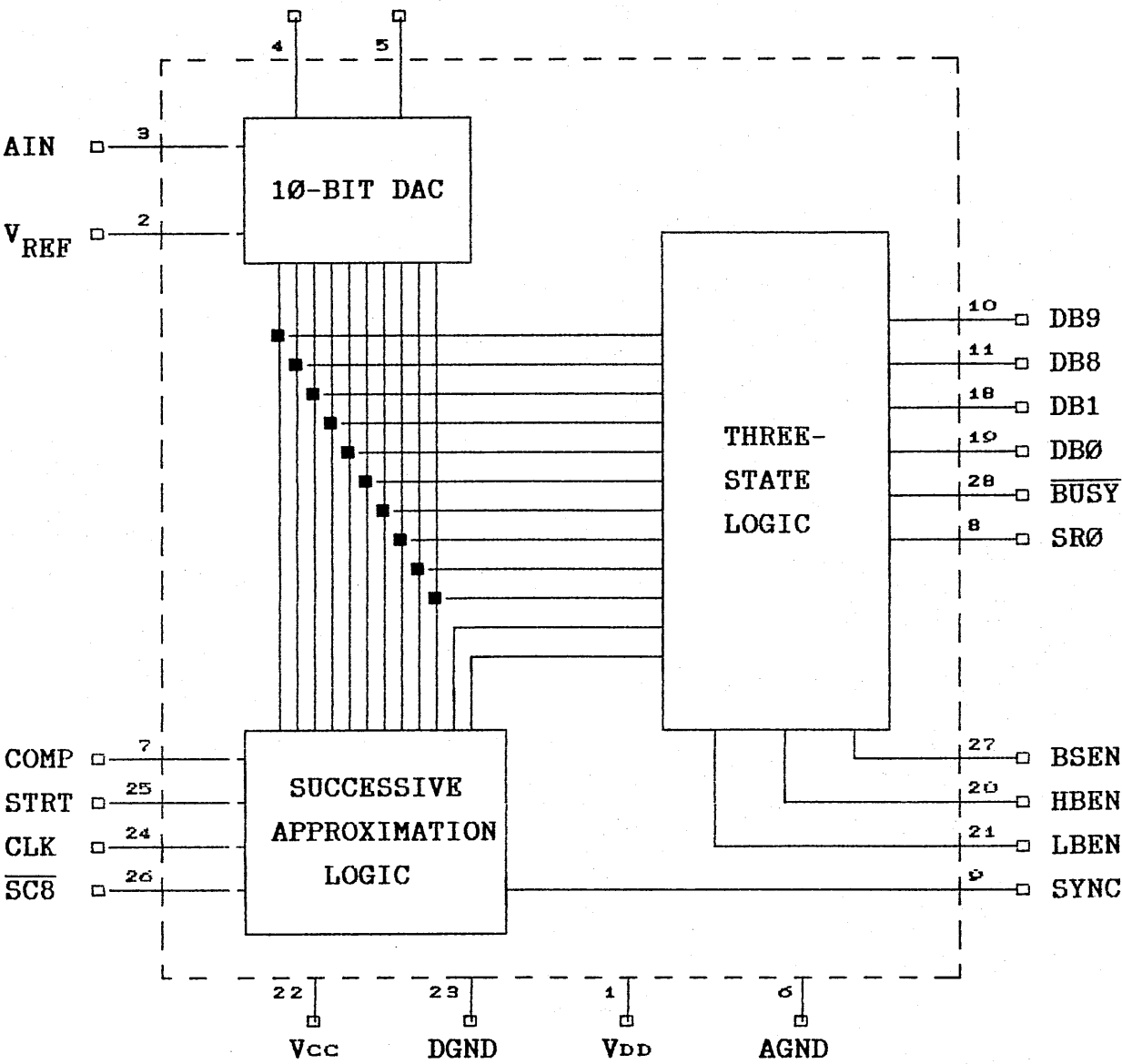


Fig. C.6 AD7570 Functional Diagram

AD7570 BIPOLAR (OFFSET BINARY) OPERATION:

Fig. C.7 shows the AD7570 configured for offset binary (modified 2's complement) operation as used in this project.

Input voltage/output codes are shown in Table C.1.

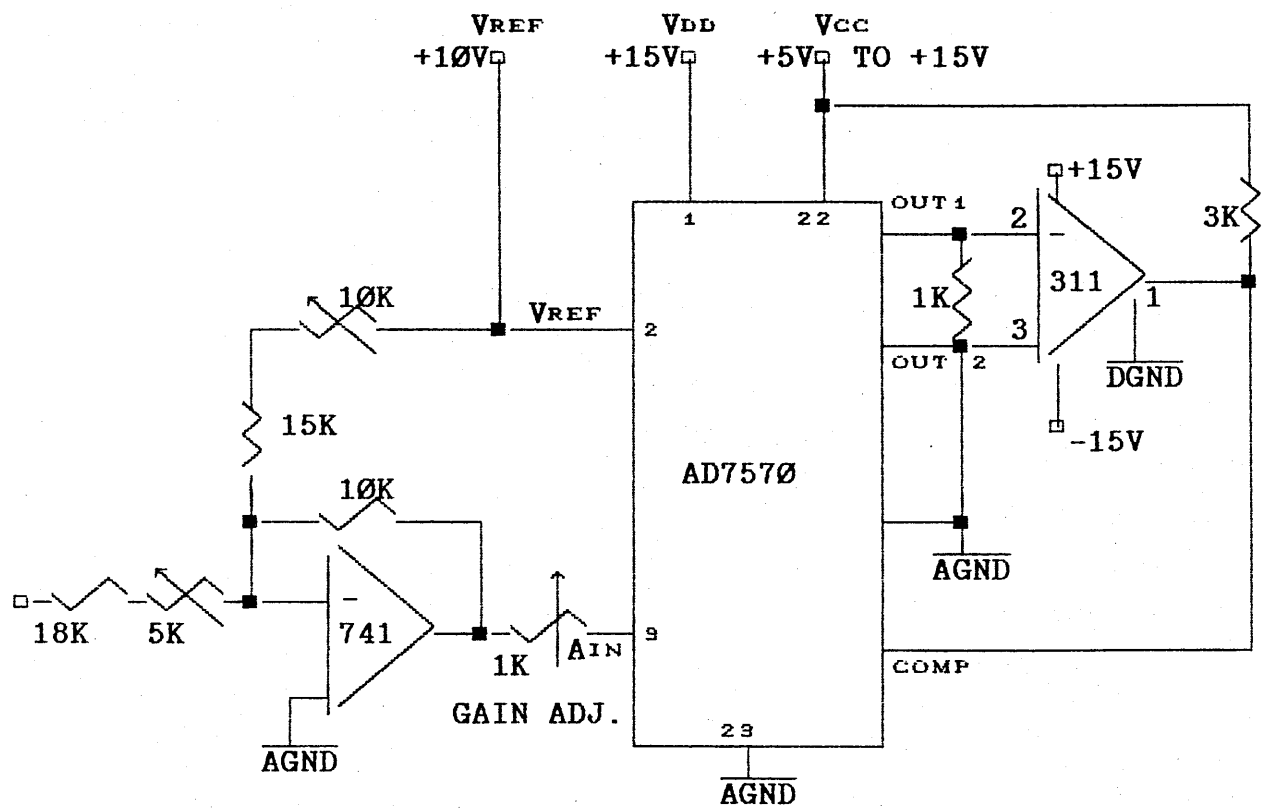


Fig. C.7 Bipolar Operation

ANALOG INPUT (AIN)	DIGITAL OUTPUT CODE	
	MSB	LSB
+(FS-1LS)	1111111111	
+(FS-2LSB)	1111111110	
+(1/2 FS)	1100000000	
+(1LSB)	1000000001	
0	1000000000	
-(1LSB)	0111111111	
-(1/2 FS)	0100000000	
-(FS-1LSB)	0000000001	
-FS	0000000000	

FS is full scale i.e. Vref

Table C.1 AD7570 Bipolar Operation

C.6 AD7522 D/A CONVERTER:

The AD7522 is a monolithic CMOS 10-bit multiplying D/A converter, with an input buffer and a holding register, allowing direct interface with microprocessors, requires the addition of only an operational amplifier and a reference voltage.

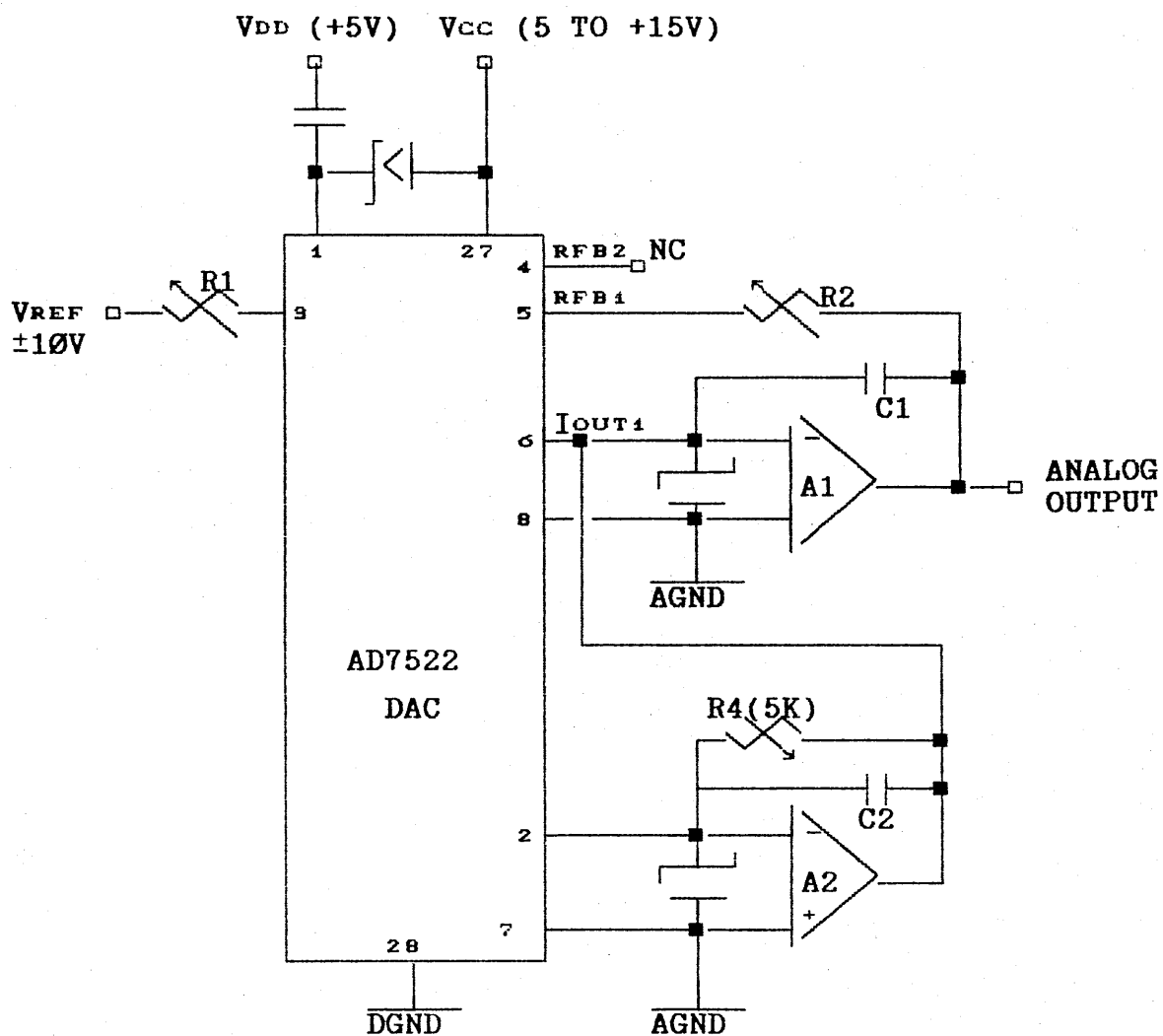


Fig. C.9 Bipolar Operation.

DIGITAL INPUT	ANALOG OUTPUT
1111111111	$-V_{REF} (1 - 2^{-9})$
1000000001	$-V_{REF} (2^{-9})$
1000000000	0
0111111111	$V_{REF} (2^{-9})$
0000000001	$V_{REF} (1 - 2^{-9})$
0000000000	V_{REF}

Table C.2 AD7522 Bipolar Code Table

A P P E N D I X D
THE STATE ESTIMATOR, SUB-OPTIMAL KALMAN FILTER
MICROPROCESSOR SOFTWARE

D.1 INTRODUCTION:

The estimator algorithm is executed each sampling time when the CPU receives an interrupt from the timer. The structure of this interrupt service routine can be summarized as follows:

- Digital input: converts the analog input signals (using the A/D converter) into a suitable digital form and saves them in the appropriate memory locations.
- Estimator algorithm: executes the estimator algorithm using matrix multiplication, addition, subtraction routines. These require matrices dimension preparation using validity checking routines.
- Digital output: outputs the calculated state estimates to the D/A converter.

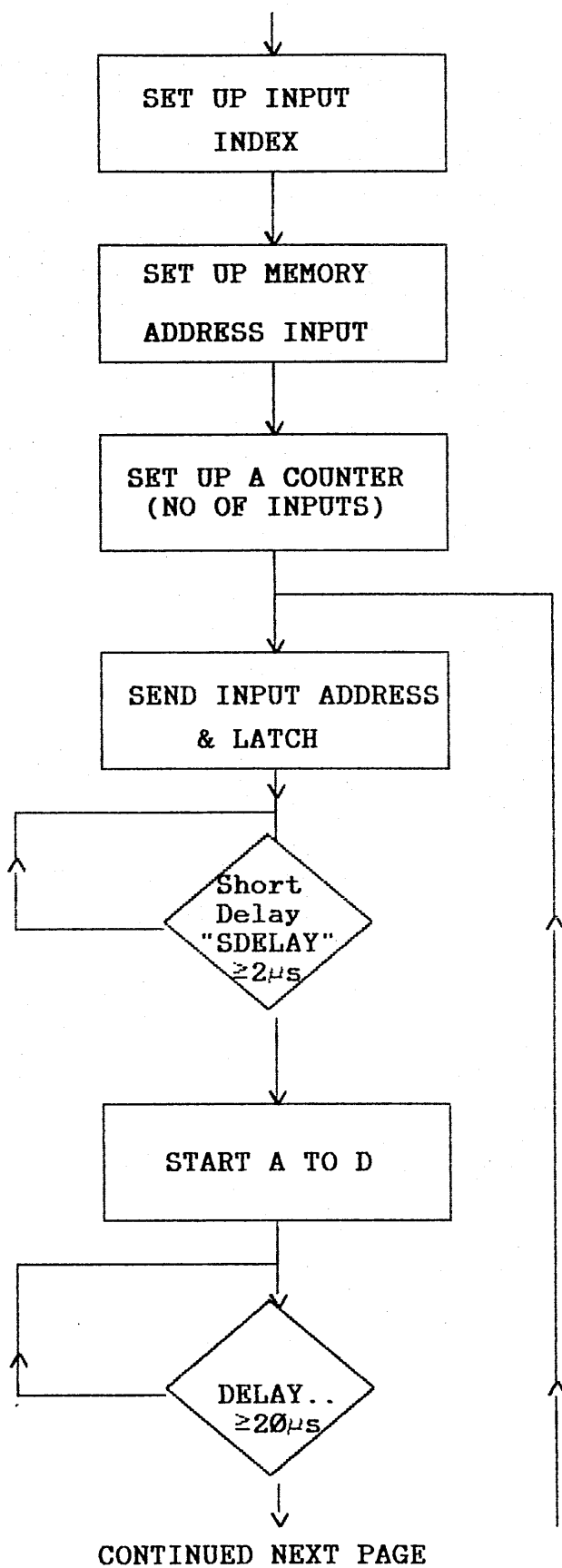
These are described in the following sections.

D.2 DIGITAL INPUT :

Digital input from each analogue input signal is achieved via the analogue to digital converter. The program first sets up the address latch to enable the multiplexer device for the first port address. Once the line has been latched in (after a few micro seconds calling a short delay routine), the A to D is started and a delay loop waits for conversion (at least 20 μ sec is needed by calling a DELAY routine). The digital value is then saved in the appropriate memory location after changing into the 2's complement form. This process is repeated for the number of inputs required.

Fig D.1 shows the flow chart of this subroutine. Fig D.2 shows the flow chart of converting binary to 2's complement notation.

Fig D.1 DIGITAL INPUT:



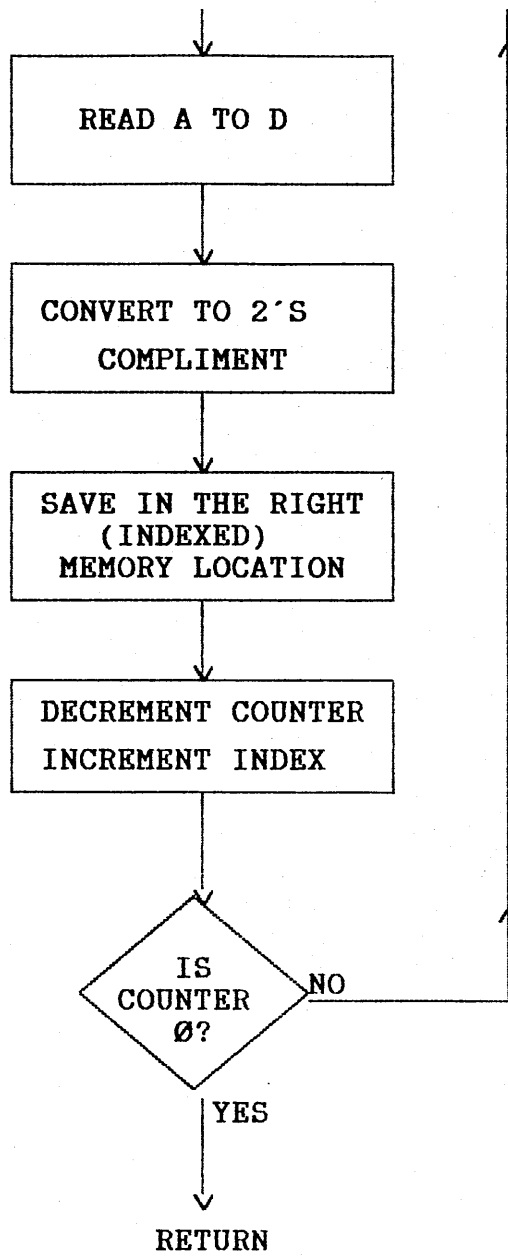
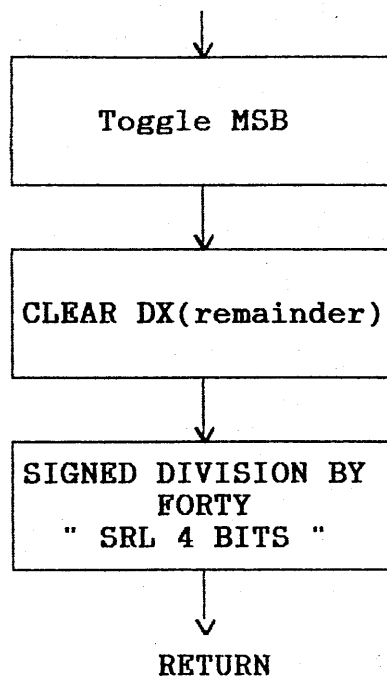


Fig D.2 CONVERTING OFFSET BINARY TO TWO'S COMPLEMENT ROUTINE:



D.3 ESTIMATOR ALGORITHM:

The algorithm required is:

$$\underline{x}(n+1) = \phi \underline{x}(n) + \psi \underline{u}(n) + M(\underline{y}(n) - C \underline{x}(n) - D \underline{u}(n)) \quad \dots (D.1)$$

All the matrices and vectors shown in the algorithm above have the appropriate size in the memory. To achieve the final result three intermediate stores are used; WORK, ANS and INTER (see source file).

Table D.1 shows the algorithm sequence.

Table D.1 The Estimator Algorithm Sequence

OPERATION	OPERATION TO DATE	SUBROUTINE REQUIRED
WORK = D U	Du	Multiply
ANS = C X	Cx	Multiply
INTER = Y - ANS	y-Cx	Subtract
ANS = INTER-WORK	y-Cx-Du	Subtract
WORK = M x ANS	M(y-Cx-Du)	Multiply
ANS = ψ U	ψu	Multiply
INTER = WORK+ANS	M(y-Cx-Du)+ ψu	Add
ANS = ϕ X	ϕx	Multiply
X = ANS + INTER	$x(n+1)=\phi x(n)+M(y-Cx-Du)+\psi u$	Add

In matrix addition, subtraction and multiplication a check is made for matrix validity (i.e. the selected matrices are of proper size for the required operation) and an error is shown if incorrect. When multiplication of elements is done, the result is re-scaled, that is, the answer is divided by the scaling factor.

An overflow check is made after each division, and the message DIVISION ERROR is displayed if an overflow occurs using the OVERFLOW Interrupt Service Routine.

D.3.1 VALIDITY CHECK FOR MULTIPLICATION :

This routine checks that the number of columns of the multiplicand is equal to the number of the multiplier. If not so an error message MATRIX SIZE ERROR is displayed and the program will halt. Otherwise, the size of the matrix containing the result is determined, to be saved in the reserved "word" for matrix definition.

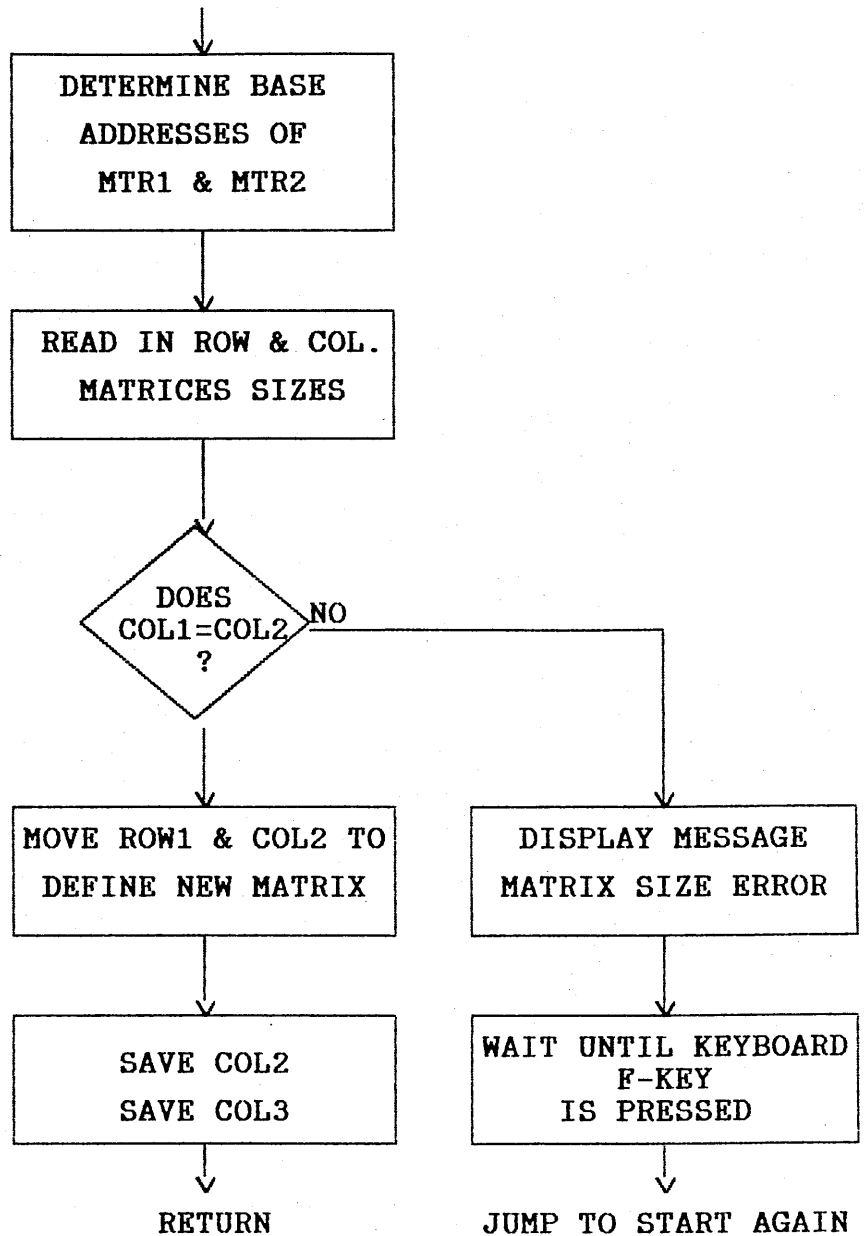
The routine also saves number of the multiplicand columns in the memory word COL2 and number of the result matrix columns in COL3, to be used by the MAMUL (matrix multiplication) routine as counters.

At entry to this routine:

MTR1 (MATRIX 1): RAM memory word contains offset address of the multiplicand matrix dimension.
MTR2 (MATRIX 2): RAM memory word contains offset address of the multiplier matrix dimension.
MTR3 (MATRIX 3): RAM memory word contains offset address of the result matrix dimension.

Fig D.3 shows the flow chart of this subroutine.

Fig D.3 VALM ROUTINE (testing matrices for valid multiplication):

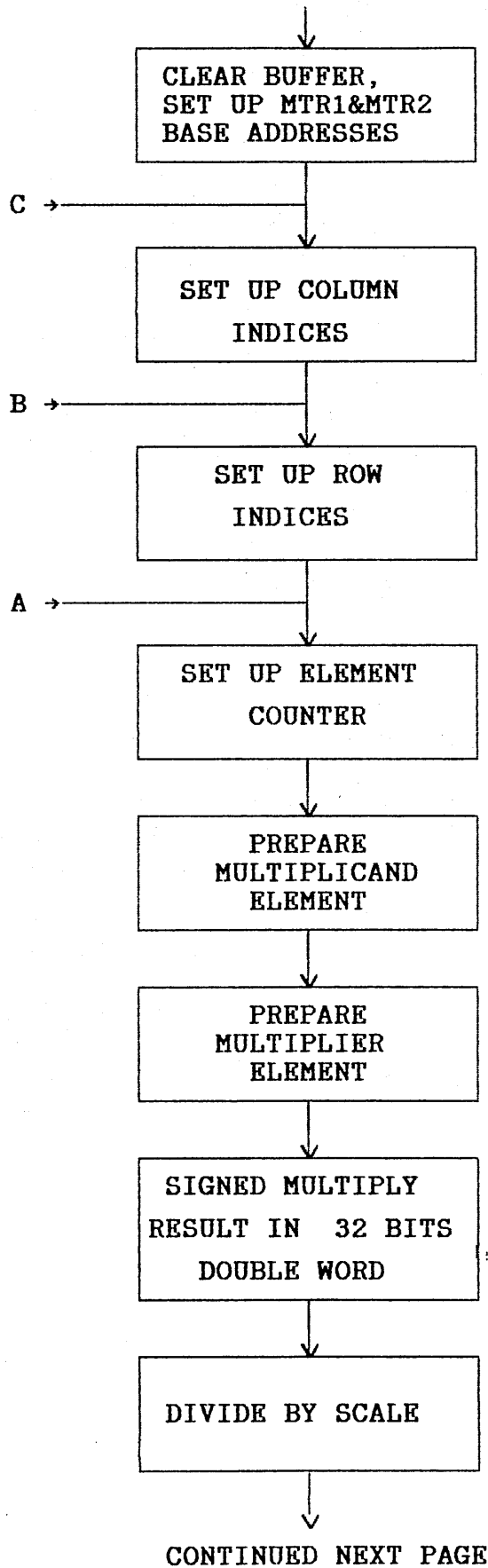


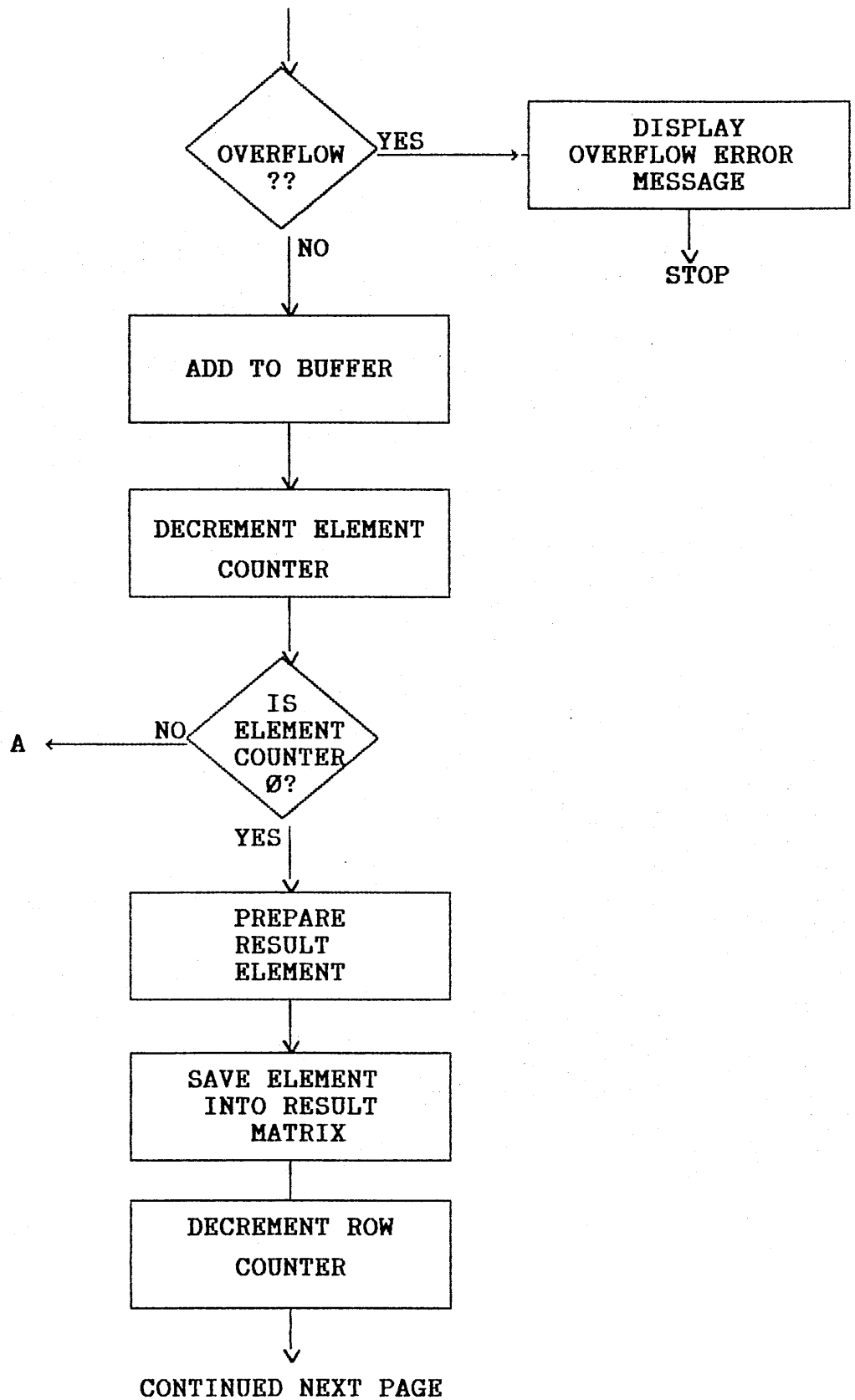
D.3.2 MATRIX MULTIPLICATION ROUTINE :

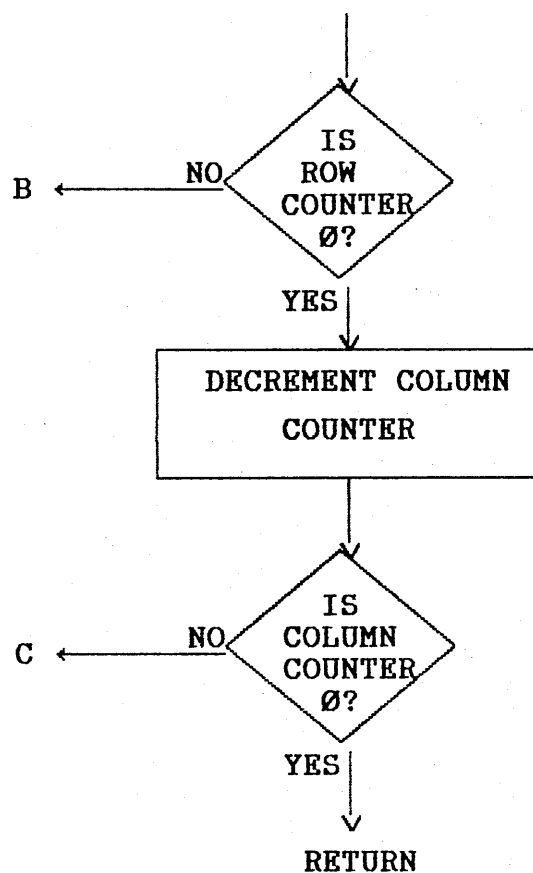
The heart of the matrix multiplication routine is the system of indexing which enables the elements to be indexed for each operation in the appropriate manner. Each element of the result matrix requires a sub-index to change multiplicand columns and multiplier rows. Following multiplication and re-scaling, these are summed into the appropriate element.

Fig D.4 shows the flow chart of this subroutine.

Fig D.4 MATRIX MULTIPLICATION ROUTINE (MAMUL):







D.3.3 ADDITION/SUBTRACTION VALIDITY:

A validity check is carried out on matrices before addition or subtraction is performed. This ensures that the matrices to be added/subtracted have both the same number of rows and the same number of columns, if not, the message **MATRIX SIZE ERROR** will be displayed on the display and the program will halt waiting an interrupt from keyboard F-KEY or system RESET.

At entry to this routine ;

MTR1 contains first matrix offset address.

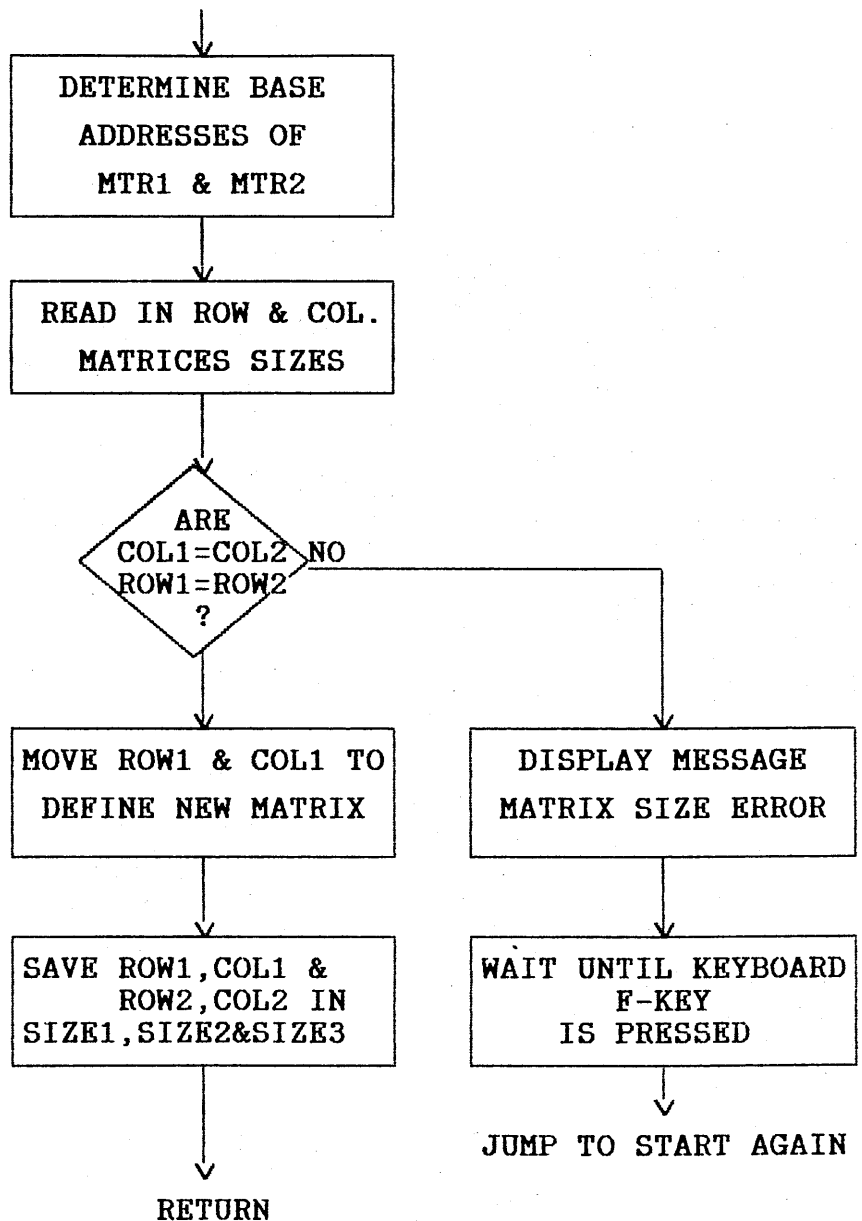
MTR2 contains second matrix offset address.

MTR3 contains result matrix offset address.

The number of rows and columns of the matrix storing the result is also recorded at this stage.

Fig D.5 shows the flow chart of this subroutine.

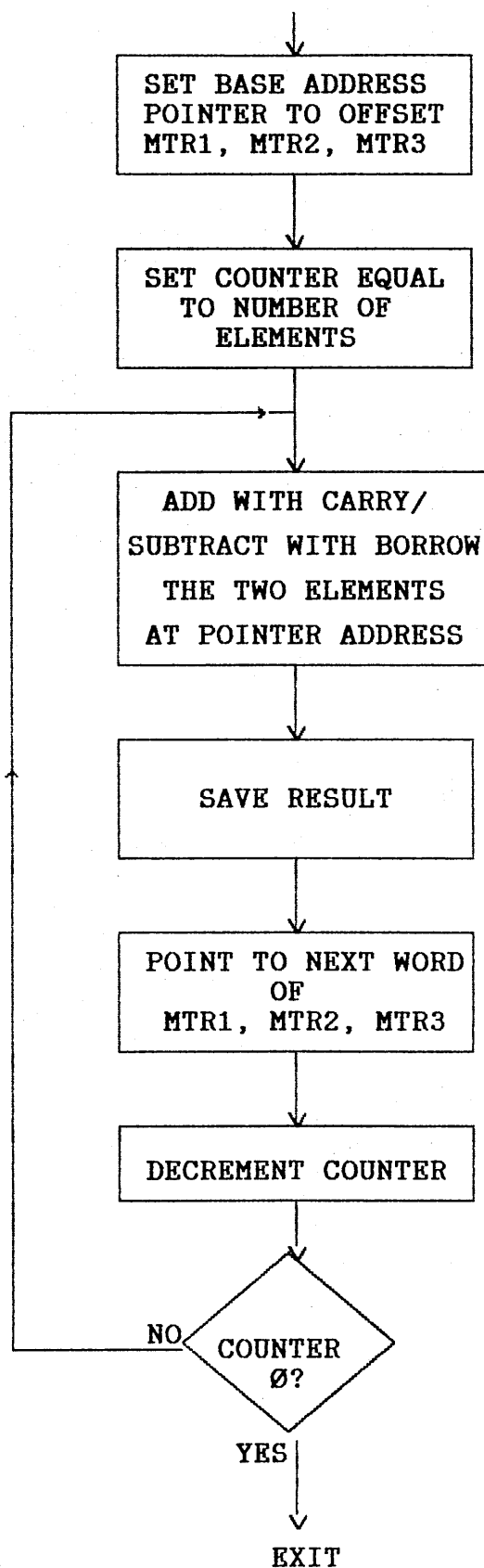
Fig D.5 VALA/VALS ROUTINE



D. 3. 4 ADDITION/SUBTRACTION ROUTINE:

This routine requires simple indexing for the source matrices (MTR1 & MTR2) and the destination matrix (MTR3) which is the result matrix. MADD (matrix addition subroutine) and MSUB (matrix subtraction subroutine) are exactly the same except in the mathematical operation ADC (Add with carry) & SBB (subtract with borrow) commands. Fig. D.6 shows the flow chart of this subroutine.

Fig D.6 MATRIX ADDITION/SUBTRACTION ROUTINE:



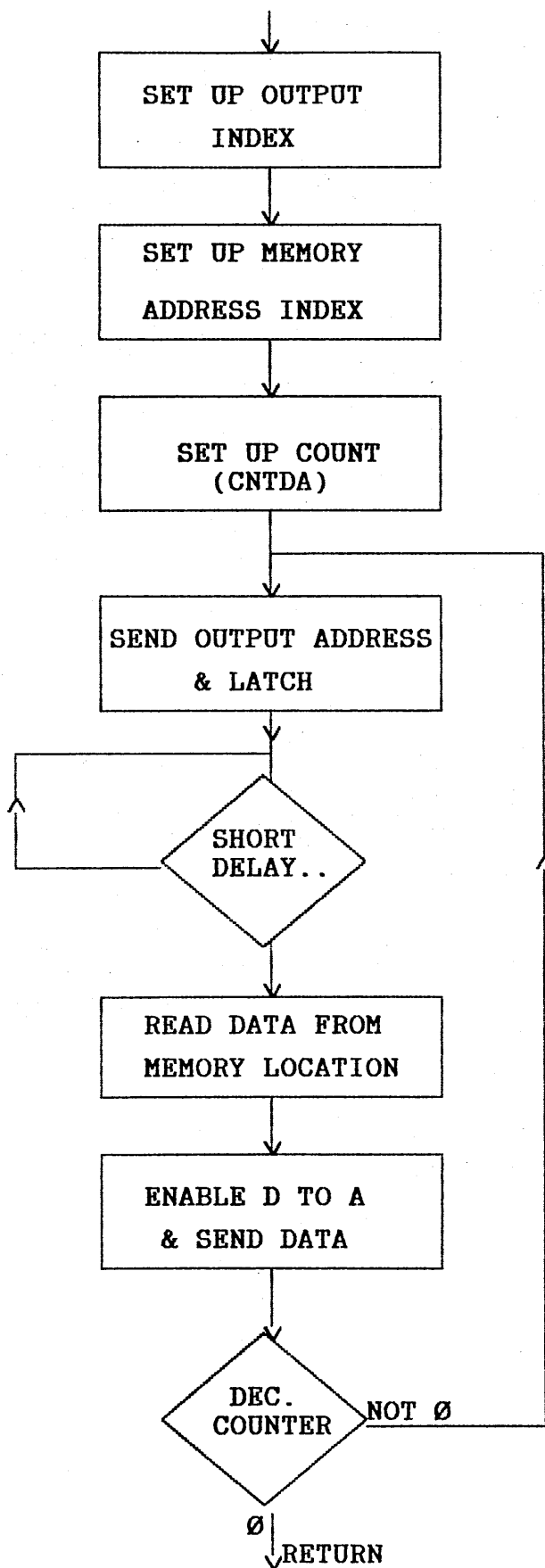
D.4 DATA OUTPUT :

This routine outputs the digital data calculated by the estimator algorithm (the estimated states) to the D/A converter and to the appropriate output port through the 8-channel multiplexer.

The output address is first set up, then latched into the interface to provide the multiplexer its address, in the same instruction the multiplexer is enabled using A4 Address bit, then the D/A is enabled to start conversion.

Fig. D.7 shows the flow chart of this subroutine.

Fig D.7 DIGITAL OUTPUT SUBROUTINE:



D.5 PROGRAM ERRORS ROUTINES:

These interrupt service routines are used by the program just in case of errors during the run time, these are:

- **OVER_FLOW:** The main program jumps to this routine in case of over flow (O.F. bit of the flag set to 1). The routine displays on the display first line "OVER FLOW ERROR" and on the second line of the display "Press F key to restart". After that the routine stays in a waiting loop until keyboard F-KEY (or SYSTEM RESET) is pressed. In such case the program restarts again.

- **DIVIDE_ERROR:** If an attempt to divide by zero anywhere in the program, this routine will be served. The routine displays the " DIVIDE ERROR " on line 1, and "Press F key to restart" on line 2 of the display. Also the routine waits F-KEY or SYSTEM RESET to restart again.

- **SIZE_ERROR:** As above but this routine is used by the main program in case of attempting to add, subtract or multiply two matrices of unsuitable sizes.

D.6 INTERRUPT TABLE:

The interrupts Table D.2 shows the Interrupt Service Routines used by this program. All the interrupt vectors are maskable i.e. can be enabled or disabled any time by the program as required.

Table D.2 The Interrupt Vectors Used by the Program

Interrupt Vector	type	Interrupt Service Routine (ISR)	Function
IV_0	0	DIVIDE_ERROR	Dividing error
IV_1	1	START	} program top
IV_2	2	START	
IV_3	3	START	
IV_4	4	OVER_FLOW	Over flow error
IV_8	8	SIZE_ERR	matrix dimension err
IV_9	9	CALC	Estimator algorithm
IV_10	10	F_KEY_ISR	F-KEY routine

APPENDIX E STATE ESTIMATOR EXPERIMENTAL RESULTS

E.1 DISCRETE STATE ESTIMATOR AND CONTROLLER:

A schematic of a discrete state estimator is shown in Fig. E.1.

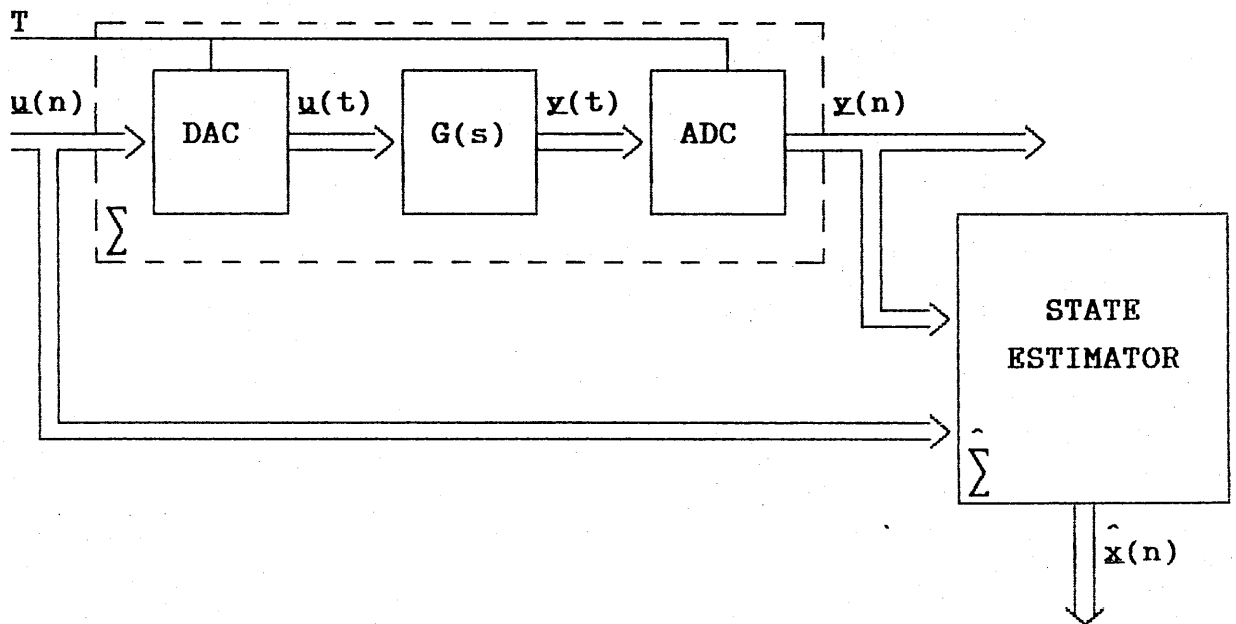


Fig. E.1 Discrete State Estimator

The behaviour a discrete system Σ is described by the dynamical equations :-

$$\left. \begin{aligned} \mathbf{x}(n+1) &= \phi \mathbf{x}(n) + \psi \mathbf{u}(n) \\ \mathbf{y}(n) &= \mathbf{C} \mathbf{x}(n) + \mathbf{D} \mathbf{u}(n) \end{aligned} \right\} \quad \dots (E.1)$$

where $\mathbf{u}(n)$, $\mathbf{x}(n)$ and $\mathbf{y}(n)$ are the system inputs, states and outputs respectively. ϕ , ψ , $\mathbf{C}$, and $\mathbf{D}$ are constant matrices (LTI systems).

The equation of the estimator is as follows:-

$$\hat{\underline{x}}(n+1) = (\phi - MC)\hat{\underline{x}}(n) + \psi \underline{u}(n) + M\underline{y}(n) - MD\underline{u}(n) \quad \dots (E.2)$$

where $\hat{\underline{x}}(n)$ represents the estimated system states and the matrix M is selected so that the eigenvalues of $(\phi - MC)$ have values in the z -plane corresponding to a rapid decay. It is easy to show that these eigenvalues govern the behaviour of the estimator "error" $[\underline{x}(n) - \hat{\underline{x}}(n)]$ and a rapid decay of this naturally ensures "good" state estimation.

The poles (or eigenvalues of ϕ) of a discrete system Σ described by Eqn (E.1), may be at undesirable locations in the z -plane. These may be shifted to desirable locations using state feedback. However, because the states $\underline{x}(n)$ are sometimes not easily available for this purpose, an estimator may be used, and the control strategy indicated in Figure E.2 is adopted.

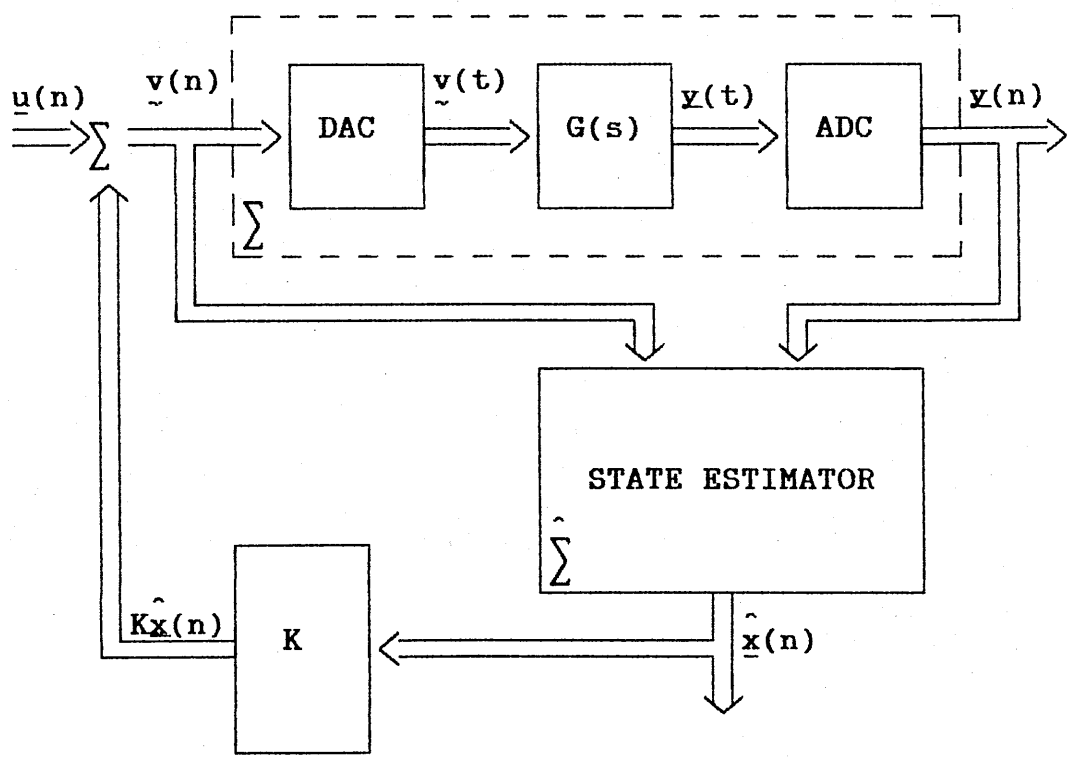


Fig. E.2 Discrete State Estimator with State Feedback.

For convenience the input to system Σ is redefined as $\tilde{v}(n)$ while the closed loop system input is designated $u(n)$. Assuming that $\hat{x}(n) = x(n)$, the dynamical equations of this arrangement are taken to be as follows;

$$x(n+1) = (\phi - \psi K)x(n) + \psi u(n) \quad \dots (E.3)$$

$$y(n) = (C - DK)x(n) + Du(n) \quad \dots (E.4)$$

The matrix K is selected so that the eigenvalues of $(\phi - \psi K)$ have values in the z -plane corresponding to "good" closed loop system behaviour.

E.2 PROCEDURE AND CALCULATIONS:

The example used to demonstrate the above theory is a third order continuous time system S with dynamical equations as follows:

$$\dot{x}(t) = \begin{bmatrix} -14 & 10 & 0 \\ -18.8 & 0 & 10 \\ -14.8 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 14.8 \end{bmatrix} u(t) \quad \dots (E.5)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t) \quad \dots (E.6)$$

The poles of this system in the s -plane are shown in Fig. E.3.

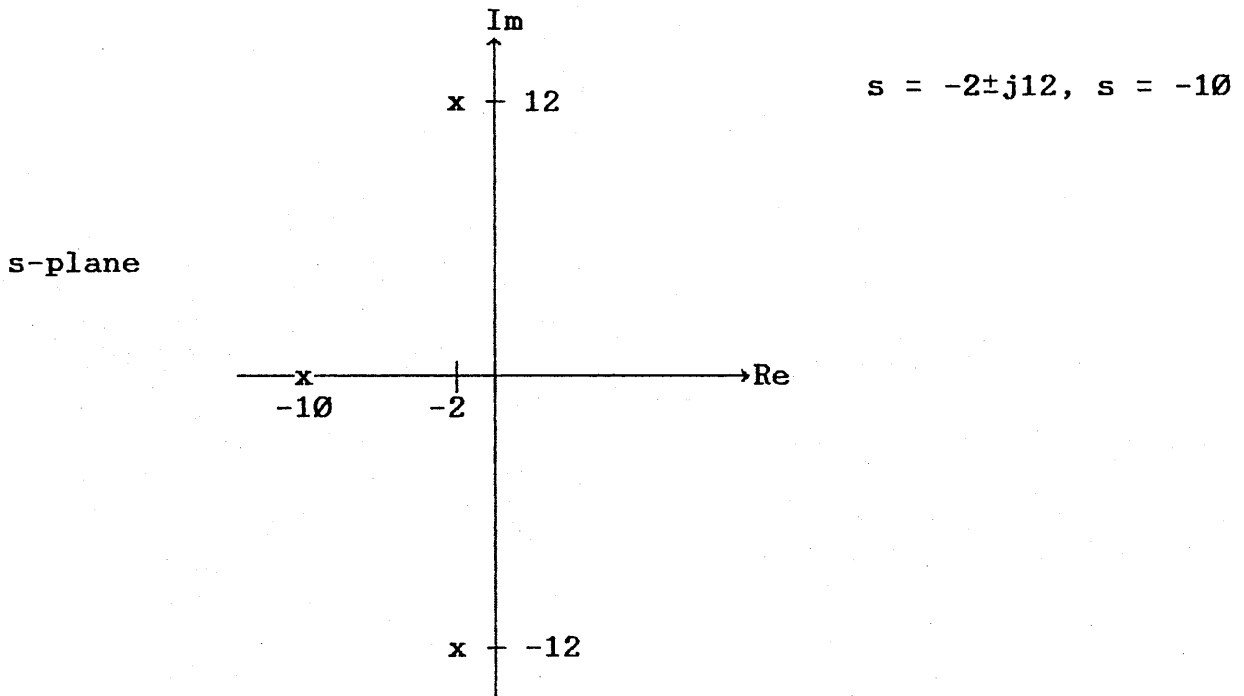


Fig. E.3 s-plane for the system S.

The system was simulated on an analogue computer and the analogue model for the system is shown in Fig. E.4. This was connected up to the analogue computer as shown in Fig. E.5.

The equivalent discrete system for a sampling period of $T = 0.02$ seconds has the dynamical equations as follows:-

$$\mathbf{x}(n+1) = \begin{bmatrix} 0.723 & 0.172 & 0.018 \\ -0.351 & 0.964 & 0.198 \\ -0.255 & -0.268 & 0.998 \end{bmatrix} \mathbf{x}(n) + \begin{bmatrix} 0.002 \\ 0.029 \\ 0.296 \end{bmatrix} u(n) \quad \dots (E.7)$$

$$\mathbf{y}(n) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}(n) \quad \dots (E.8)$$

The Eigen values of ϕ matrix for this system are at;
 $z = 0.793, \quad z = 0.95 \pm j0.32$

To design the asymptotic state estimator given by Eqn. E.2, M has to be selected so that the poles of $(\phi - MC)$ have values in the z-plane corresponding to a rapid decay. The values chosen are;

$$z = 0.22, \quad z = 0.22, \quad z = 0.6$$

M then found to be as follows:-

$$M = \begin{bmatrix} 1.647 \\ 5.592 \\ 4.568 \end{bmatrix}$$

As stated above, the pole positions indicate an oscillatory response, these can be shifted using the arrangement outlined in Fig E.2. The dynamical equations for this arrangement, as given by Equation E.3, have to be designed such that the matrix K is selected so that the poles of $(\phi - \psi K)$ have values in the z-plane corresponding to *good* closed loop system behaviour. The values chosen are as follows:-

$$z = 0.5, \quad z = 0.1, \quad z = 0.1$$

so;

$$K = \begin{bmatrix} 2.0 & 7.8 & 9.0 \end{bmatrix}$$

E.3 RESULTS AND OBSERVATIONS:

Fig. E.4 shows the analog model of the system. The system was set up on an analog computer as Fig. E.5 and the step response obtained as shown in Fig. E.6. The response is oscillatory as expected from the position of the poles.

The actual states and the states estimated by the equipment were recorded for comparison and are shown in Fig. E.7, Fig. E.8 and Fig. E.9. It can be seen that the behaviour of the estimated states $\hat{x}(n)$ and their equivalents $x(t)$ are quite well matched.

Finally, the equipment was connected as outlined by the arrangement shown in Figure E.2 which provides state feedback to obtain a response with *good* behaviour. The values designed for the feedback matrix K were entered into the equipment and the closed loop response and states were recorded as shown in Fig. E.10, Fig. E.11 and Fig. E.12. It can be seen that the poles have been shifted compared with Fig. E.7, Fig. E.8 and Fig. E.9.

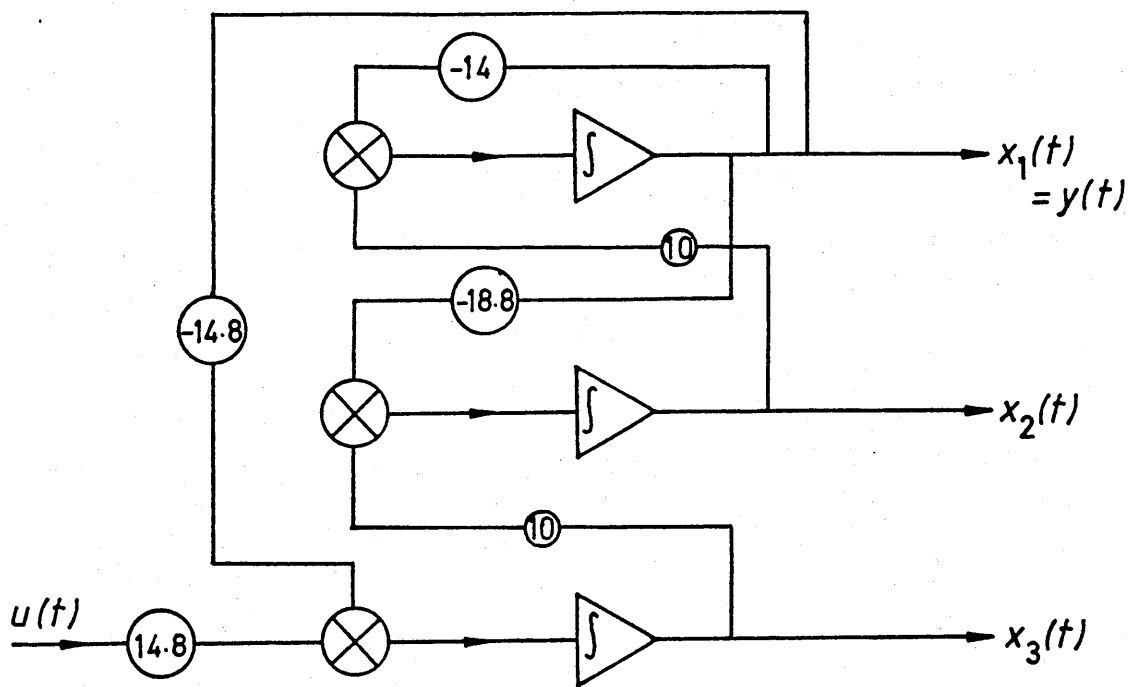


Fig. E.4 Analogue model for the system S.

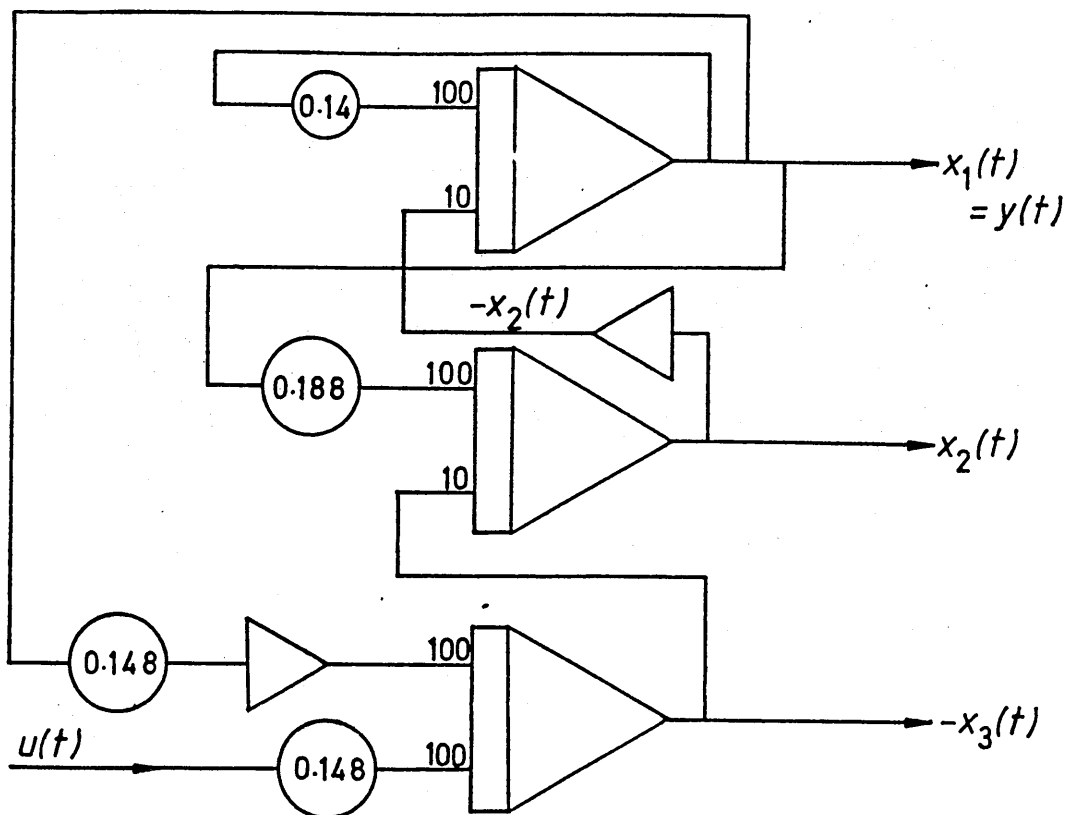


Fig. E.5 Analogue Computer Diagram

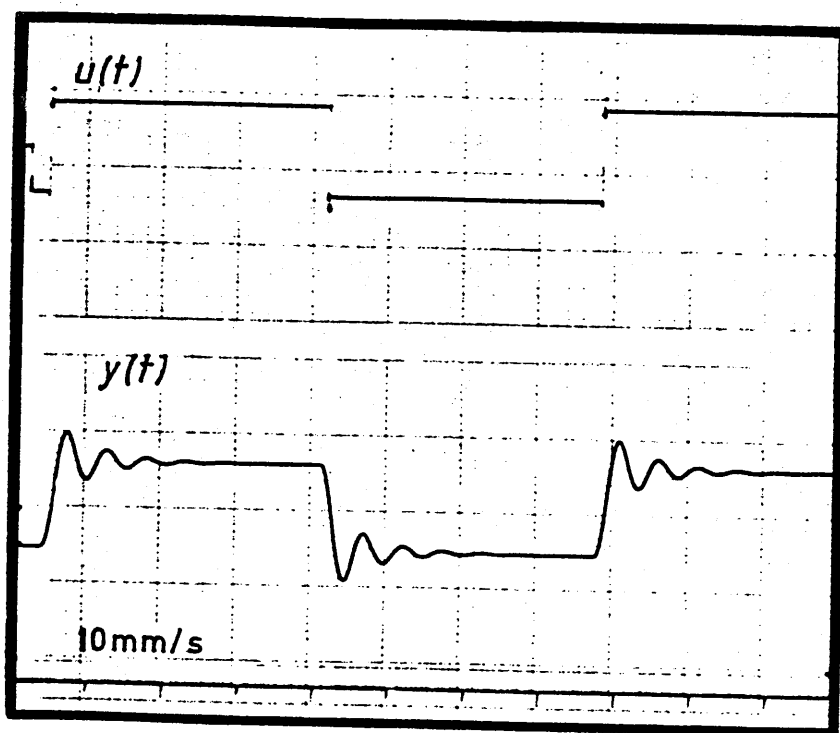


Fig. E.6 Step response of the system.

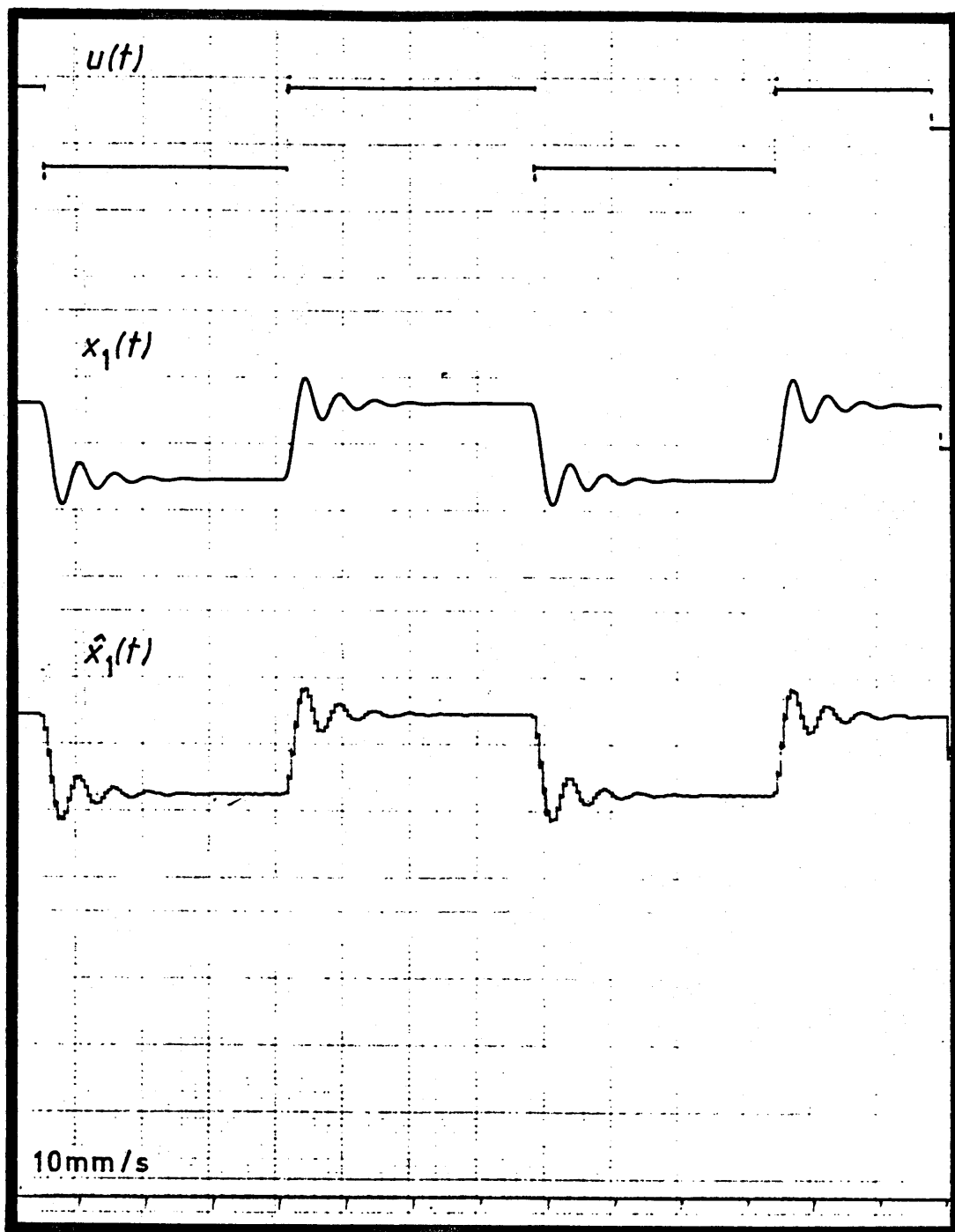


Fig. E.7 Actual and Estimated State 1 of the System

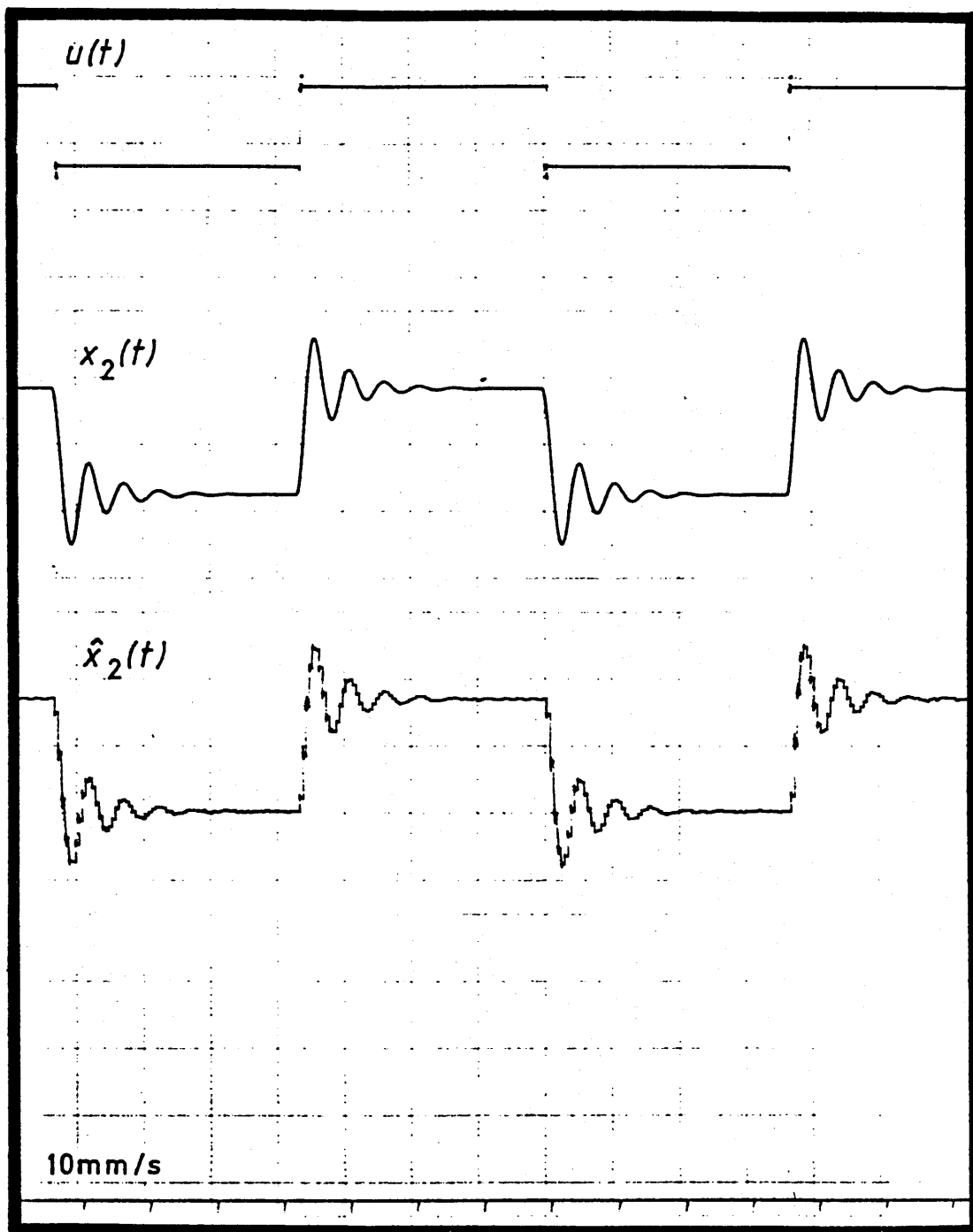


Fig. E.8 Actual and Estimated State 2 of the System.

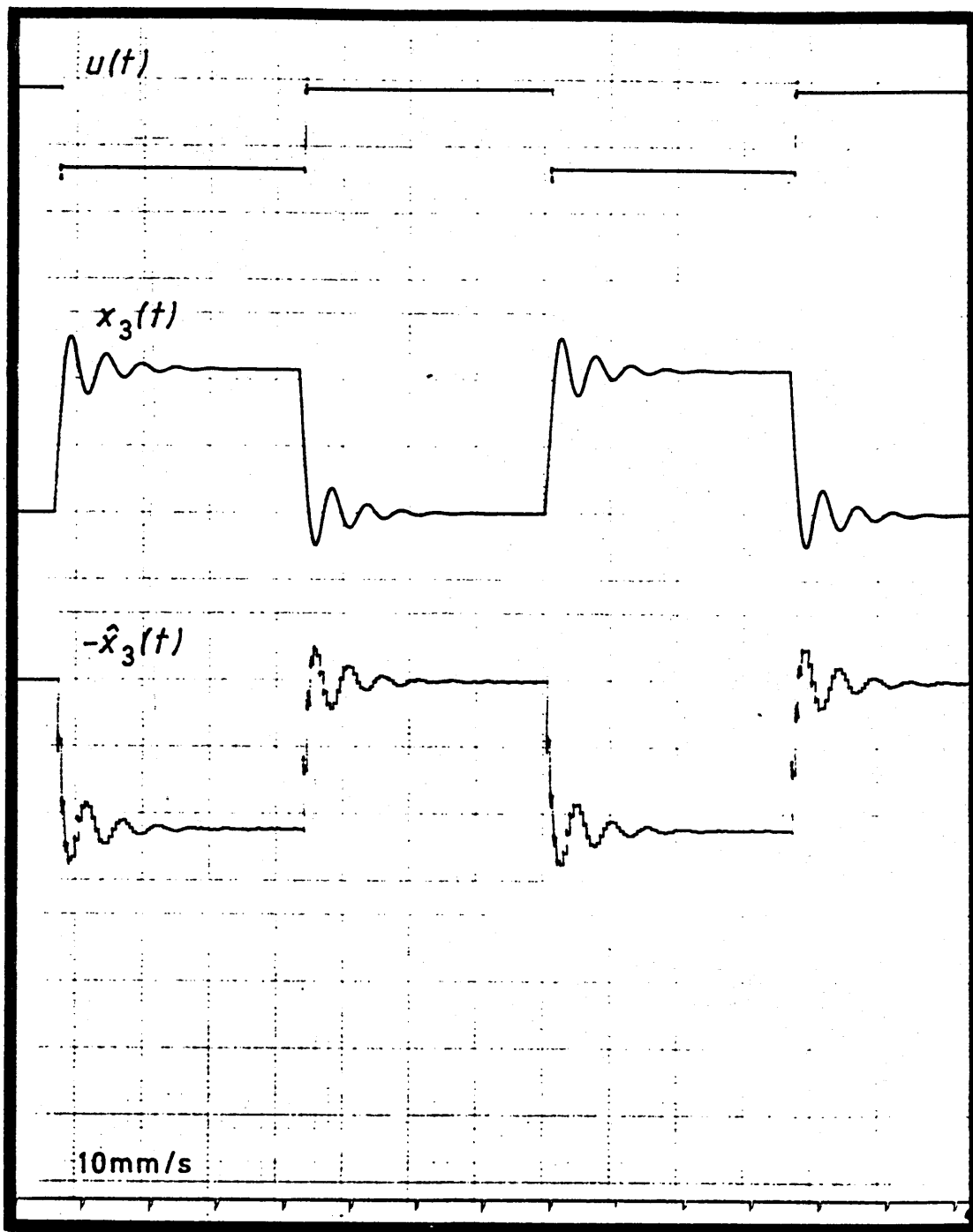


Fig. E.9 Actual and Estimated State 3 of the System.

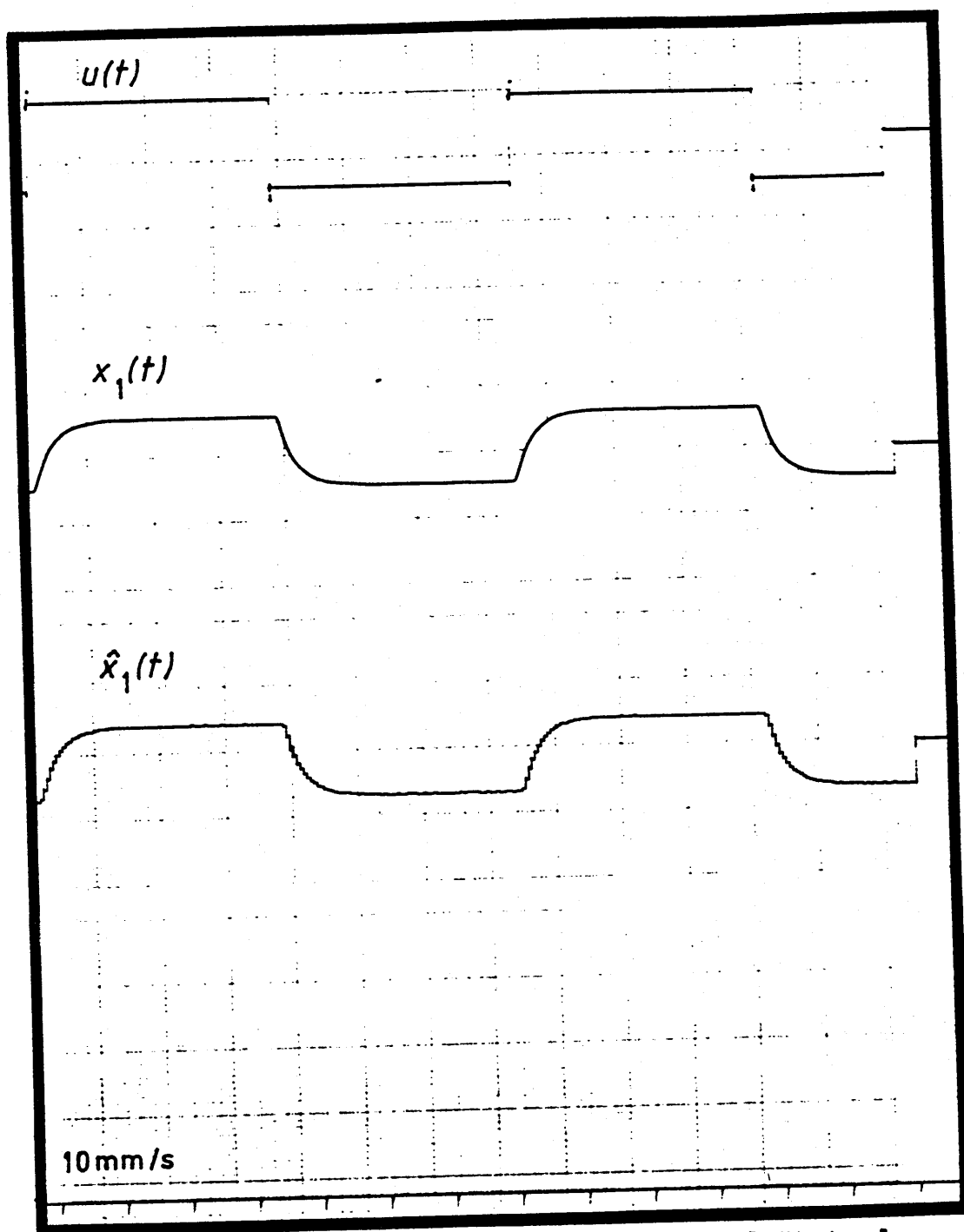


Fig. E.10 Closed Loop Actual and Estimated State 1 for the System.

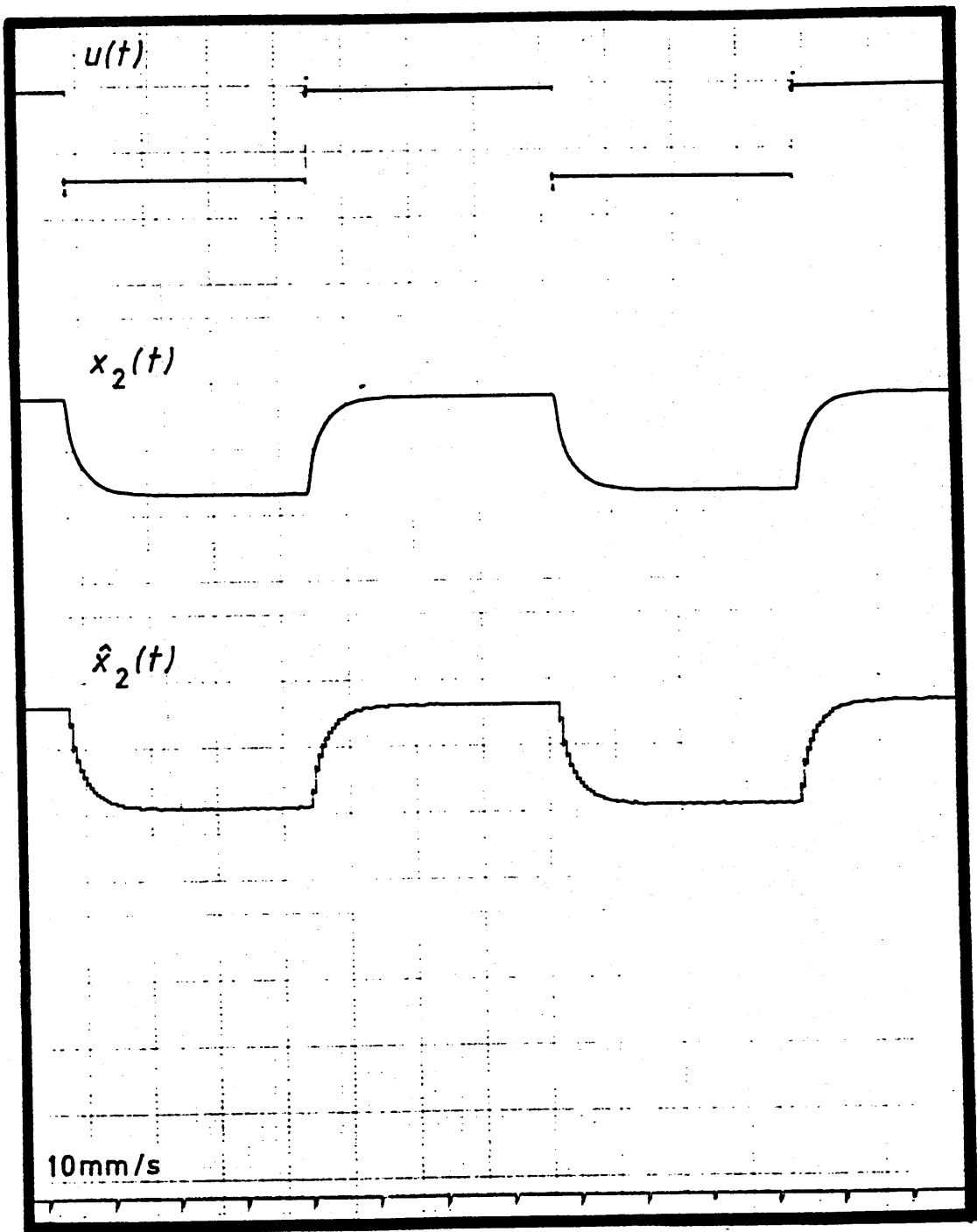


Fig. E.11 Closed Loop Actual and Estimated State 2 of the System.

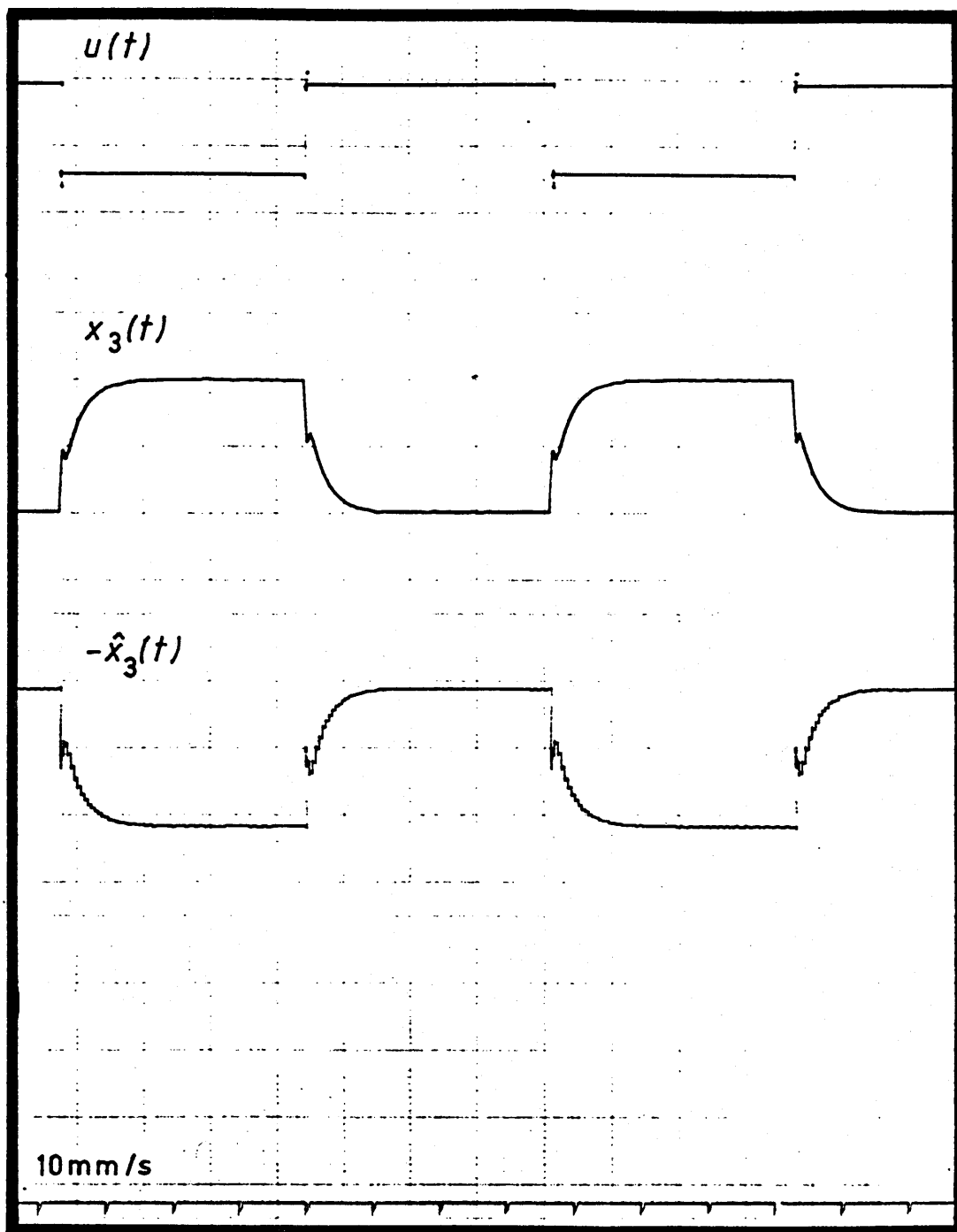


Fig. E.12 Closed Loop Actual and Estimated State 3 of the System.

These tests carried out by the microprocessor show satisfactory behaviour of the hardware and the software, and any system model can be used later on.

A P P E N D I X F

LINEAR STOCHASTIC OPTIMAL CONTROL THEORY USING PARTIAL STATE FEEDBACK

F.1 INTRODUCTION:

The aim of this Appendix is to find an optimal state feedback gain vector using partial state feedback for the system described in Chapter 7.

F.2 LINEAR OPTIMAL CONTROL THEORY:

First, consider a linear system with full state feedback represented by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \quad \dots (F.1)a$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{v} \quad \dots (F.1)b$$

with m state variables $\mathbf{x}$, s control inputs $\mathbf{u}$, r controlled variables $\mathbf{y}$ and noise disturbance inputs $\mathbf{w}$ & $\mathbf{v}$ satisfying:

$$\mathbb{E} \left[\mathbf{w}(t) \mathbf{w}^T(t+\tau) \right] = \mathbf{Q} \delta(\tau) \quad \dots (F.2)a$$

$$\mathbb{E} \left[\mathbf{v}(t) \mathbf{v}^T(t+\tau) \right] = \mathbf{R} \delta(\tau) \quad \dots (F.2)b$$

where $\mathbb{E}$ is the expectation operator, δ is the Dirac delta function and $\mathbf{Q}$ is positive semi definite and $\mathbf{R}$ is positive definite symmetric matrices.

The steady state, linear stochastic regulator problem consists of determining the control $\mathbf{u}^*$ which minimizes:

$$J = \mathbb{E} \int_0^{\infty} \left[\mathbf{x}(t) \bar{\mathbf{Q}}_c \mathbf{x}^T(t) + \mathbf{u}(t) \bar{\mathbf{R}}_c \mathbf{u}^T(t) \right] dt \quad \dots (F.3)$$

where $\bar{Q}_c$ and $\bar{R}_c$ are positive semi definite and definite symmetric matrices respectively.

It is well known [2] that the solution of this problem involves solving the algebraic Riccati equation:

$$A^T P_c + P_c A + C^T \bar{Q}_c C - P_c B \bar{R}_c^{-1} B^T P_c = 0 \quad \dots (F.4)$$

The optimal control law is given by:

$$u^* = -K_c x = -\bar{R}_c^{-1} B^T P_c x \quad \dots (F.5)$$

P_c is the "optimal" positive semi definite solution to (F.4) and the resulting closed loop system is stable.

If a system is *unobservable* it can be transformed into the form;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \quad \dots (F.10)$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \dots (F.11)$$

Where the subsystem (A_{22}, C_2) is observable, and (A_{11}) contains the unobservable modes. Eqn. (F.5) can then be used to calculate the optimal input for this subsystem.

Partial state feedback is used [1], when some of the states are missing because they can not be measured or estimated (unobservable system), using a control law as;

$$u = -Kz \quad \dots (F.12)$$

where K is a constant rxn matrix and the n measurement z defined by;

$$z = Nx \quad \dots (F.13)$$

where N is a square matrix chosen to select the available states from x .

The elements of K minimize the performance index (F.3) as before.

Substituting (F.12) and (F.13) into (F.1a) gives the following closed loop system:

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + G\tilde{w} \quad \dots (F.14)$$

where

$$\tilde{A} = A - BKN \quad \dots (F.15)$$

The performance index (F.3) may then be determined from [1]:

$$J = \text{trace} \left[W_c \left(C^T \bar{Q}_c C - N^T K^T \bar{R}_c K N \right) \right] = \text{trace} \left[W_o Q \right] \quad \dots (F.16)$$

where

$$\tilde{A}W_c + W_c \tilde{A}^T + Q = 0 \quad \dots (F.17)$$

$$\tilde{A}^T W_o + W_o \tilde{A} + C^T \bar{Q}_c C - N^T K^T \bar{R}_c K N = 0 \quad \dots (F.18)$$

The approach to choose K close to optimal is to use a gradient search technique [2]. For this purpose, the gradient of the performance index in (F.16) with respect to any element K_i ($i=1, \dots, n$) of K can be found from:

$$\begin{aligned} \frac{\partial J}{\partial K} &= \begin{bmatrix} \frac{\partial J}{\partial K_1} \\ \vdots \\ \frac{\partial J}{\partial K_n} \end{bmatrix} \\ &= 2 \left[R_c K N W_c N^T - B^T W_o W_c N^T \right] \quad \dots (F.19) \end{aligned}$$

Hence, the basic algorithm uses the following steps:

- (i) Guess an initial value for K .
- (ii) Solve (F.17) & (F.18) for W_c and W_o (the method described in Ref 5 is used in this work).
- (iii) Calculate J in (F.16) and $\frac{\partial J}{\partial K}$ in (F.19) and update K using a gradient search routine [2].
- (iv) Goto step (ii) until satisfactory convergence has been achieved.

To prove that Equation (F.19) is correct let us assume all the states are available, "N" in (F.13) becomes the (mxm) identity matrix. Equating the derivative in (F.19) to zero gives an expression for "K" which when substituted into (F.18) turns this last equation into the algebraic Riccati Equation (F.4) with $W_o = P_c$.

REFERENCES:

- [1] D. A. Wilson, R. S. Sharp, and S. A. Hassan, "The Application of Linear Optimal Control Theory to the Design of Active Automotive Suspension", Vehicle System Dynamics. 1986.
- [2] Arthur E Bryson, Jr. Yu-Chi Ho, "Applied Optimal Control", Hemisphere Publishing Corp. 1975.
- [3] A. G. Thompson, "Optimal and Suboptimal Linear Active Suspensions for Road Vehicle", Vehicle System Dynamics, vol 13 (1984).
- [4] A. Hac, "Suspension Optimization of a 2-DOF Vehicle Model Using a Stochastic Optimal Control Technique", Journal of Sound and Vibration (1985) 100(3).
- [5] E.J. Davison and F.T. Man "The numerical solution of $A'Q + QA = -C$ ", IEEE Trans. on Automatic Control, August 1968, Vol. AC-13, No.4 PP.448-449.

APPENDIX G

NON-LINEARITIES OF A HYDRAULIC SYSTEM

First consider one hydraulic actuator as shown in Fig. G-1.

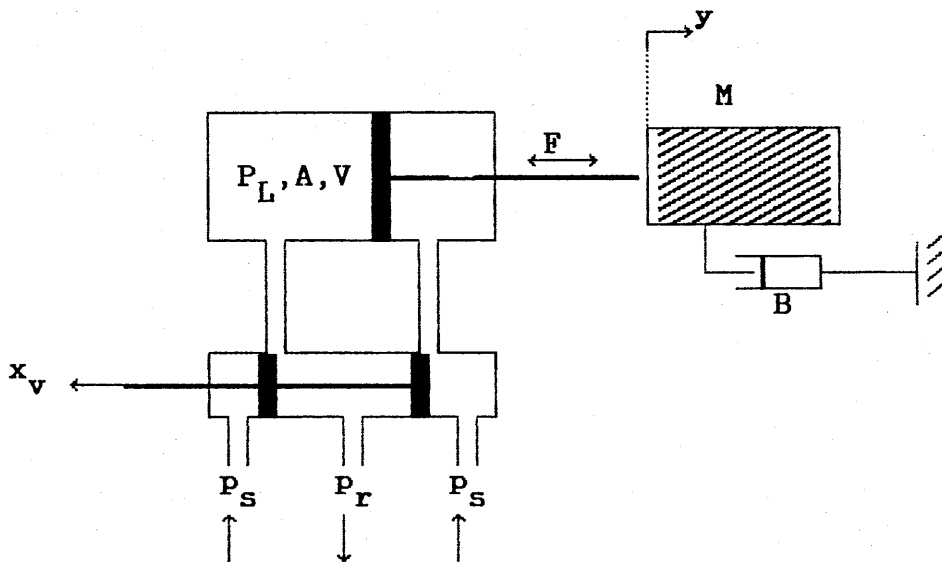


Fig. G.1 Typical Hydraulic Actuator Using a Spool Valve

for;

N	Bulk modulus of Hydraulic fluid (Nm^{-2})
M	Mass (Kg)
F	Force (N)
P	Pressure (Nm^{-2})
A	Effective Area (m^2)
V	Volume (m^3)
Q	flow rate ($\text{m}^3 \text{s}^{-1}$)
x_v	Valve displacement (m)
k	Valve gain
B	Damping coefficient (Nsm^{-1})
C_L	Leakage coefficient

(I) Actuator flow:

Flow into actuator = Volume flow+compressibility flow+Leakage flow

$$Q = A\dot{y} + \frac{V}{2N}\dot{P}_L + C_L P_L \cong A\dot{y} \quad (N \rightarrow \infty, C_L \rightarrow 0) \quad \dots(G.1)$$

(II) Load equation:

$$A P_L = M\ddot{y} + B\dot{y} \quad \dots(G.2)$$

Flow equation:

$$Q = kX_v \sqrt{P_s - P_L} \quad X_v > 0 \quad \dots(G.3)$$

$$Q = kX_v \sqrt{P_s + P_L} \quad X_v < 0$$

From (G.2);

$$P_L = \frac{M}{A}\ddot{y} + \frac{B}{A}\dot{y} \quad \dots(G.4)$$

From (G.1) and using (G.3):

$$P_L = P_s - \left(\frac{A\dot{y}}{kX_v} \right)^2 \quad X_v > 0 \quad \dots(G.5)$$

$$P_L = -P_s + \left(\frac{A\dot{y}}{kX_v} \right)^2 \quad X_v < 0$$

Subst. (G.4) in (G.5);

$$\frac{M}{A}\ddot{y} + \frac{B}{A}\dot{y} = P_s - \left(\frac{A\dot{y}}{kX_v} \right)^2 \quad X_v > 0$$

$$\frac{M}{A}\ddot{y} + \frac{B}{A}\dot{y} = -P_s + \left(\frac{A\dot{y}}{kX_v} \right)^2 \quad X_v < 0$$

so;

$$\ddot{y} = \left[P_s - \left(\frac{A\dot{y}}{kX_v} \right)^2 - \frac{B}{A}\dot{y} \right] \frac{A}{M} \quad X_v > 0 \quad \dots(G.6)$$

$$\ddot{y} = \left[-P_s + \left(\frac{A\dot{y}}{kX_v} \right)^2 - \frac{B}{A}\dot{y} \right] \frac{A}{M} \quad X_v < 0$$

Clearly the relationship in Eqn.(G.6) between $y(t)$ and $x_v(t)$, which is proportional to the input $u(t)$ is non-linear.

If the actuator is lightly loaded, and small perturbations about steady state values are considered, the model can be linearized as follows:[1]

if $P_s \gg P_L$, from Eqns (G.3) and (G.1);

$$Q = kX_v \sqrt{P_s} = CX_v = A\dot{y} \quad (\text{where } C \text{ is a constant})$$

We now consider the behaviour of the combination of two hydraulic actuators connected in series as shown in Fig. G.3. This combination can be modeled as follows (assuming positive inputs);

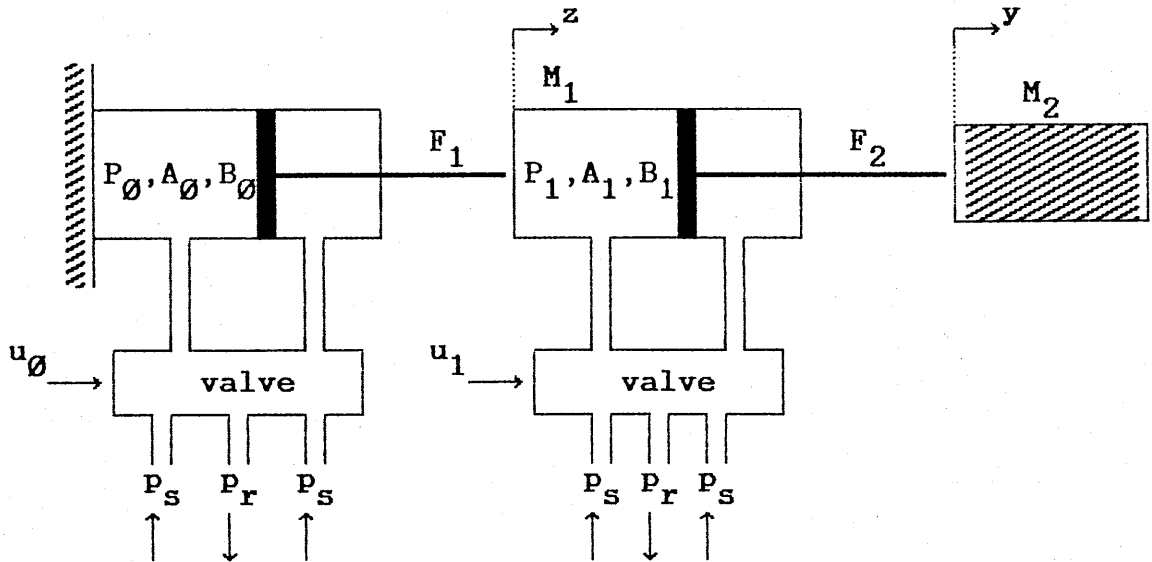


Fig. G-3 Hydraulic System of Two Actuators in Series

(I) Force equations:

$$F_1 = A_0 P_0 - A_1 P_1 = M_1 \ddot{z} + B_0 \dot{z} + B_1 (\dot{z} - \dot{y}) \quad \dots (G.7)$$

$$F_2 = A_1 P_1 = M_2 \ddot{y} + B_1 (\dot{y} - \dot{z}) \quad \dots (G.8)$$

where P_0 and P_1 are differential pressures over pistons.

(II) Flow equations:

$$Q_0 = A_0 \dot{z} + \frac{V_0}{2N} \dot{P}_0 = k_0 u_0 \sqrt{P_s - P_0} \quad \dots (G.9)$$

$$Q_1 = A_1 (\dot{y} - \dot{z}) + \frac{V_1}{2N} \dot{P}_1 = k_1 u_1 \sqrt{P_s - P_1} \quad \dots (G.10)$$

where Q_0 and Q_1 are flows into LHS of cylinders.

Assuming no leakage and as the bulk modulus in Eqns G.9 and G.10 is very large (for a typical hydraulic fluid $N=2 \times 10^9$), the terms $\frac{V}{2N} \dot{P} \rightarrow 0$.

Re-writing the above equations in differential equations form:

$$\dot{y} = (A_1 P_1 - B_1 (\dot{y} - \dot{z})) / M_2 \quad \dots (G.11)$$

$$\dot{z} = (A_0 P_0 - A_1 P_1 - B_0 \dot{z} - B_1 (\dot{z} - \dot{y})) / M_1 \quad \dots (G.12)$$

where;

$$P_0 = \left[1/k_0^2 u_0^2 \right] * (k_0^2 u_0^2 P_s - (A_0 \dot{z})^2) \quad \dots (G.13)$$

$$P_1 = \left[1/k_1^2 u_1^2 \right] * (k_1^2 u_1^2 P_s - (A_1 (\dot{y} - \dot{z}))^2) \quad \dots (G.14)$$

Again, to linearize the model, load pressures are assumed negligible if the actuators are lightly loaded with small perturbations about the valve center and assuming the valve has a square port and no dead zone [1].

REFERENCES:

- [1] J. Lewis, Control systems lecture notes, 1986, DESD, CIT.

A P P E N D I X H

SPECIFICATIONS OF THE HYDRAULIC EQUIPMENT

The hydraulic system designed in this project (Chapter 7) has the following specification:

Potentiometer	$P_1=12\text{cm}, P_2=20\text{cm}, V = \pm 15\text{V}.$
Electric-hydraulic valve	$\pm 15\text{V}.$
Full stroke	6cm
Hydraulic pressure (P_s)	1000 psi = $6.90 \times 10^6 \text{ Pa (Nm}^{-2}\text{)}$
Hydraulic fluid bulk modulus (N)	$2.0 \times 10^9 \text{ Pa}$
Effective area (A)	$0.55\text{in}^2 = 3.5485 \times 10^{-4} \text{ m}^2$
Max output force (F)	$1000\text{psi} \times 0.55 = 550\text{lb} = 2450 \text{ N}$
Load Mass (M)	5Kg
Charge Amplifier	Model QM7R (Env. Equipment).
Accelerometer	Model AQ48 (Env. Equipment).