

THE UNIVERSITY OF HULL

NONLINEAR CONTROL OF AN INDUSTRIAL ROBOT

being a Thesis submitted for the Degree of

Doctor of Philosophy

in the University of Hull

by

James Michael Gilbert BSc.

October 1989

IMAGING SERVICES NORTH

Boston Spa, Wetherby

West Yorkshire, LS23 7BQ

www.bl.uk

BEST COPY AVAILABLE.

VARIABLE PRINT QUALITY

IMAGING SERVICES NORTH

Boston Spa, Wetherby
West Yorkshire, LS23 7BQ
www.bl.uk

MISSING PAGE/PAGES
HAVE NO CONTENT

ABSTRACT

The precise control of a robot manipulator travelling at high speed constitutes a major research challenge. This is due to the nonlinear nature of the dynamics of the arm which make many traditional, linear control methodologies inappropriate. An alternative approach is to adopt controllers which are themselves nonlinear. Variable structure control systems provide the possibility of imposing dynamic characteristics upon a poorly modelled and time varying system by means of a discontinuous control signal. The basic algorithm overcomes some nonlinear effects but is sensitive to Coulomb friction and actuator saturation. By augmenting this controller with compensation terms, these effects may largely be eliminated.

In order to investigate these ideas, a number of variable structure control systems were applied to a low cost industrial robot having a highly nonlinear and flexible drive system. By a combination of hardware enhancements and control system developments, an improvement in speed by a factor of approximately three was achieved while the trajectory tracking accuracy was improved by a factor of ten, compared with the manufacturer's control system.

In order to achieve these improvements, it was necessary to develop a dynamic model of the arm including the effects of drive system flexibility and nonlinearities. The development of this model is reported in this thesis, as is work carried out on a comparison of numerical algorithms for the solution of differential equations with discontinuous right hand sides, required in the computer aided design of variable structure control systems.

ACKNOWLEDGEMENTS

There are many people without whose help, the work described in this thesis would not have been possible. In particular, I would like to thank the SERC for providing funding for the work; the Department of Electronic Engineering for providing the facilities and the technical staff of the department for their help, especially Jim Preston and Dave Glover for the loan of many pieces of equipment.

I would like to thank Tony Wilkinson and Ahmed Parvinmehr for many helpful discussions on various aspects of variable structure control theory and Darwin Caldwell and Gareth Monkman for discussions on the subject of robotics in general. Many thanks are due to Graham Holden for his apparently limitless patience in answering my programming questions and to Ian Gibson, Mike Gunner and Gary Palmer for thoroughly testing the control systems in a practical application. Thanks are also due to UMI and Tim Jones for the supply of technical information on the RTX.

I would like to thank Liselot Hertel for her patience and encouragement during the preparation of this thesis.

Finally and especially, I would like to thank my supervisor, Prof. Paul Taylor for his guidance and encouragement throughout the period of my research.

Summary of Thesis Submitted for PhD. Degree

by

James Michael Gilbert

on

Nonlinear Control of an Industrial Robot

This Thesis considers the development and implementation of Variable Structure Control systems for the control of an industrial robot manipulator. The robot considered is of a low cost construction and, as a result, has many undesirable characteristics not found in more expensive systems. In particular, the joint drive system has a significant degree of flexibility and highly nonlinear frictional behaviour. These effects make traditional, linear control systems less effective than would be hoped.

By a combination of minor hardware enhancements and the development of new variable structure control algorithms, taking account the drive system characteristics, an improvement in speed by a factor of three and an increase in trajectory tracking accuracy by a factor of ten, compared with the manufacturer's controller, have been achieved. The control system was developed, based on a nonlinear dynamic model of the RTX, the derivation of which is described here. Also considered is a comparison of various numerical algorithms for the solution of differential equations with discontinuous right hand sides, used in the computer aided design of variable structure control systems.

CONTENTS

	Page
Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 The Robot Control Problem and Some Solutions	2
1.2 Control of the RTX Robot	5
1.3 Outline of Subsequent Chapters	7
2 The Theory of Variable Structure Control Systems	10
2.1 Introduction to the Theory of Variable Structure Control Systems	10
2.1.1 Historical development of Variable Structure Control Theory	11
2.1.2 Fundamantal Principles of Variable Structure Control Systems	12
2.1.2.1 Sliding and Non-sliding Variable Structure Systems	12
2.1.2.2 Sliding and Reaching Conditions	17
2.1.3 Properties of Systems with Sliding Modes	19
2.1.3.1 Dynamics of the Sliding Mode	19
2.1.3.2 Robustness of Sliding Mode Behaviour	20
2.1.4 Non-ideal Sliding Behaviour	23
2.1.5 Chatter Reduction Techniques	25
2.1.5.1 Compensation Techniques for Chatter Reduction	25

2.1.5.2 Smoothing Techniques for Chatter Reduction	26
2.1.5.3 Combined Smoothing and Compensation Chatter Reduction (Slotine's Algorithm)	30
2.1.6 Variable Structure Control of Nonlinear Systems	34
2.1.6.1 Effect of Input Saturation on Variable Structure Control Systems	34
2.1.6.2 Effect of Coulomb Friction on Variable Structure Control Systems	38
2.2 Variable Structure Control of Robot Manipulators	45
2.3 Selection of Control Parameters in VS Control Systems	48
3 Hardware Enhancements to the RTX	50
3.1 Existing RTX Hardware	51
3.2 Computer Interfaces	55
3.2.1 μ PDP Interfaces	55
3.2.2 Personal Computer Interface	56
3.3 Motor Drive Circuits	58
3.3.1 Selection of Drive Method	59
3.3.2 Design of Linear Current Drive	63
3.4 Position and Velocity Measurement	71
3.4.1 Position/Velocity Measurement Circuit 1	76
3.4.2 Position/Velocity Measurement Circuit 2	80
4 RTX Robot Arm Dynamics	85
4.1 Theoretical Basis for the Dynamic Modelling of the RTX	86
4.1.1 Dynamic model of DC Motor and Current Drive	87
4.1.2 Frictional Behaviour of Joint Drives	91
4.1.2.1 Low Speed Friction	92
4.2.2.2 High Speed Friction	93

4.1.3	Elastic Behaviour of Joint Drive Mechanism	97
4.1.4	Joint Interaction Model of the RTX	102
4.2	Experimental Dynamic Modelling of the RTX	105
4.2.1	Experimental Dynamic Modelling of Motor and Current Drive	107
4.2.1.1	Experiential Measurement of Motor Parameters	107
4.2.1.2	Measurement of Motor and Transmission System Friction	111
4.2.2	Joint Transmission dynamics	118
4.2.3	Measurement of Joint Interaction	122
4.3	Limitations of the Dynamic Model	127
4.4	Conclusions	129
5	A Computer Aided Design Suite For The Design of Nonlinear Controllers For Nonlinear Systems	130
5.1	Existing Computer Aided Design Suite	131
5.2	Miscellaneous Improvements to the CACSD Suite	133
5.3	Enhancements to the CACSD Model Structure	134
5.4	Comparison of Numerical Algorithms for the Solution of Nonlinear Ordinary Differential Equations	136
5.4.1	Single-Step Algorithms	138
5.4.1.1	Variable Step Size Runge-Kutta Algorithm	140
5.4.2	Multi-Step Algorithms	143
5.4.3	The PHAS Algorithm	145
5.4.4	Testing of Numerical Algorithms	147
5.4.4.1	Example Problems for Algorithm Comparison	149
5.4.4.2	Comparison of Algorithms for Example Problems	156

5.4.5 Selection of Numerical Algorithm	161
5.5 Evaluation of the CACSD Suite for the Design of Nonlinear Systems	163
6 Implementation and Testing of Variable Structure Control Systems on the RTX	169
6.1 Simple Variable Structure Control System	171
6.1.1 Computer Aided Design of a Simple VSC System	172
6.1.2 Hardware Implementation of a Simple VSC	174
6.1.3 Software Implementation of a Simple VSC	177
6.2 Variable Structure Control with Boundary Layer	181
6.2.1 Computer Aided Design of a VSC with Boundary Layer	182
6.2.2 Implementation of a VSC with Boundary Layer	183
6.3 Variable Structure Controller Design for Nonlinear Systems	186
6.3.1 VSC with Friction Compensation	186
6.3.1.1 Simple VSC with Friction Compensation	187
6.3.1.2 VSC with Boundary Layer and Friction Compensation	191
6.3.2 VSC with Acceleration Compensation	194
6.3.3 VSC with Boundary Layer and Integral Action	198
6.3.4 VSC with Boundary Layer, Compensation and Integral Action	200
6.4 Variable Structure Control with Piecewise Linear Switching Curves	202
6.4.1 Simple VSC with Piecewise Linear Switching Curve	206
6.4.2 VSC with Boundary Layer	208

6.4.3 VSC with Friction and Acceleration	
Compensation and Integral Action	209
6.5 Control of Joint Position	210
6.5.1 Simple VSC Around Joint Position	211
6.5.2 VSC with Boundary Layer	
Applied to Joint Position	213
6.5.3 VSC with Compensation and Integral Action	214
6.6 Control Using Joint and Motor Mounted Sensors	216
6.6.1 VSC with Boundary Layer Using	
Motor and Joint Mounted Sensors	218
6.6.2 VSC with Piecewise Linear Switching Function	220
6.7 Comparison of Control Algorithms for the RTX	221
6.7.1 Computational Cost of Control Algorithms	222
6.7.2 Speed of Response with	
Variable Structure Controllers	224
6.7.3 End Point Position Accuracy	227
6.7.3.1 Cartesian Position Measurement Techniques	228
6.7.3.2 Repeatability of Point-to-point Movements	230
6.7.3.3 Accuracy of Continuous Path Movements	231
6.8 Selection of Control System	241
7 Evaluation of the Design of Nonlinear Controllers	
for Nonlinear Systems	248
7.1 Computer Aided Design Suite Enhancements	248
7.2 RTX Hardware Development	251
7.3 Evaluation of the Dynamic Model of The RTX	253
7.4 Evaluation of Variable Structure Control Systems	254
8 Conclusions	258

References	261
Appendices	
A1 Motor Drive Power Amplifier Data	A1
A2 High Resolution Optical Encoder Data	A5
A3 74LS2000 Direction Discriminator Data	A10
A4 Dynamics of a Two Link Planar Manipulator	A23
A5 Simulation of a Two Link Planar Manipulator with	
Time Optimal Control	A25
A5.1 Manipulator Dynamics	A25
A5.2 Interaction Nonlinearity Representation	
used in PHAS	A26
A5.3 SIMNON Representation of Dynamics	A34
A6 SIMNON Representation of RTX Link Dynamics	
and VS Control	A37
A7 Additional Results for the Implementation of VS Control	A41
A7.1 Effect of Load on VSC with Piecewise Linear Switching	
Function, Compensation and Integral Action	A41
A7.2 High Gain VSC with Boundary Layer	
Using Joint Sensor	A43
A7.3 VSC with Friction and Acceleration Compensation	
and Integral Action	A44
A7.3.1 Effect of Compensation Terms	
on VSC Using Joint	A44
A7.3.2 Speed of Response Compared with PID Control	A46
A7.4 Effect of Velocity on End Point Tracking Accuracy	
for Various Controllers	A48
A7.5 Effect of Joint Encoder Resolution on	
End Point Tracking Accuracy	A52

CHAPTER 1

Introduction

Increasing use of robots in a wide variety of industrial and domestic environments depends on the solution of a number of problems which have, to date, limited their application to a relatively small number of well structured and highly repetitive tasks. One of these problems is that of obtaining high speed, precise movement under a variety of loads and external influences, without the use of high cost mechanical and actuation systems. Accordingly, this thesis considers the implementation of new control techniques on a low cost industrial robot with the aim of obtaining this type of performance.

The majority of manipulators have dynamic behaviour which results in performance under well established control schemes which is far from ideal. This is particularly true in the case of low cost systems, in which the mechanical configuration cannot be arranged so as to simplify the control problems. There is, therefore, a need for new control algorithms which will overcome the problems inherent in these low cost systems.

As an introduction, this chapter will begin by considering the problems found in the majority of robot manipulators, and those particularly prevalent in low cost systems. Some of the techniques investigated with the aim of overcoming robot control problems will be considered as will their suitability in low cost manipulators.

The particular system considered here is an RTX robot, supplied by UMI. The characteristics of this manipulator will be briefly studied, as they provide the motivation for the work reported here. The scope of this work will then be described, followed by an outline of the contents of subsequent chapters.

1.1 The Robot Control Problem and Some Solutions

The majority of multiple degree of freedom mechanisms have dynamics which are to some extent coupled. In the case of manipulation systems, there are few exceptions to this - only cartesian systems can be considered as having fully decoupled dynamics in the presence of varying loads. Although these robots are useful in some situations, there are many cases in which, serial topology arms having at least some revolute joints provide a better solution in terms of working volume and flexibility to overall size and weight ratios.

These arms, which may be viewed as being at least partially anthropomorphic in nature, have dynamics which are generally highly coupled in a nonlinear manner. It is this undesirable form of behaviour which presents one of the major problems to the control engineer working in this field. A further problem is that the dynamics depend on the mass and form of the load being carried. Although the effect of load may be precalculated in some highly structured environments, or may be determined from the changing performance of the manipulator or from force/torque measurements, these techniques are often costly to implement, and require a good model of the robot dynamics, a model which is often difficult to obtain. In general then, it is desirable that the robot performance should be independent of the load, without regress to load forecasting or direct measurement techniques. This presents a further problem to the control system designer.

These problems are common to many manipulator systems and, although a further problem is also common to most systems, it is particularly severe in the case of low cost systems. This is the problem of nonideal actuator and transmission system behaviour. In high cost mechanisms, the actuators are often considered as being mainly linear with only saturation effects occurring, while the transmission system is considered as being totally rigid. These assumptions are to some extent valid in the case of high cost, high performance robots, but becomes invalid when low cost systems are considered. In such mechanisms, the

actuator and drive system friction is typically nonlinear and accounts for a large proportion of the force/torque generated. Similarly, in the interest of cost savings, transmission systems are often adopted which introduce a significant amount of flexibility. Although these problems are beginning to receive some attention, their full significance to the control engineer has not previously been addressed.

These then are the characteristics of robot systems which make manipulator control a challenge. As the need for fast and precise control of manipulators has increased, so has the need to specifically address these problems. This has led to a large amount of work in this area and we will now briefly review the approaches which have been considered. This, of necessity will be only a very broad summary, but should be sufficient to indicate the areas in which successful solutions have been found and those which require further consideration.

It is not surprising that in early electrically driven manipulator systems, the controllers implemented were taken directly from those applied to servomechanisms, namely PID and relay controllers. Of these, decentralised PID control has remained as the mainstay of robot control and is still used in the vast majority of commercially available systems. For manipulators with relatively linear actuator dynamics which are required to move only at low speeds, PID control provides a solution which is inexpensive to implement and is fairly robust to load changes and external disturbances. Although the technique does not give particularly high speed operation, it is reliable in most situations - an important consideration in industrial applications.

If low speed operation is considered, then the most significant effect of changes in manipulator position, are the resulting variations in effective inertia. A natural way of tackling this problem is to adopt gain scheduled controllers, so as to alter the parameters of a series of linear control systems to suit the prevailing inertia conditions in discrete regions of joint space. A control system of this type is described by Ali [Ali,82] and, although it may be effective for

low speed motions, it does not tackle the problems of coriolis and centripetal forces between joints which become significant at higher speeds.

In addition to changes in inertia, there are a number of other effects which occur in a single link of a manipulator which have a bearing on the controller performance. Many of these problems are considered by Paul [Paul,81] who suggests a series of compensation schemes in which the effect of gravity loading is calculated and removed while feedforward elements are used to eliminate velocity and acceleration errors. This control procedure assumes linear actuators and does not overcome all the problems associated with joint interaction forces.

A number of schemes have been developed in which the inverse of the manipulator dynamics, assuming linear actuators, is introduced in the forward path of the controller so as to form a linear decoupled system, when combined with the manipulator dynamics, around which standard linear controllers may be used (see [Sahba,84] and [Fournier,84] for example). Clearly, in order to be effective the model of the robot dynamics inverted and used in the forward path must be accurate if this technique is to be fully effective and it is difficult to ensure this in the presence of load variations and external disturbances. The simplest form of this algorithm assumes that the actuators are linear, with unlimited force/torque, a situation which is clearly unrealistic, and although a number of developments have been made to overcome this deficiency, the resulting system has a speed of response which is significantly sub-optimal. Also, owing to the neglect of Coulomb friction in the controller design, the resulting movement may have significant errors from the planned trajectory.

Since it has been found difficult to develop controllers which compensate for all the dynamics of a robot, it would seem appropriate to consider a type of control which imposes a specified dynamic behaviour on the system, independent of limited variations in the plant itself. This is the aim of variable structure controllers, in which the state trajectory is maintained on a pre-defined switching surface by means of discontinuous state feedback. While

this is maintained, the dynamics of the system are governed by those of the switching surface, independent of plant parameter changes and disturbances. This type of control system has been considered widely, both alone and in combination with other algorithms, as a way of overcoming the problems caused by interaction dynamics in manipulators [Young,78], [Morgan,85] [Chen,87]. This has proved successful in simulation experiments and when applied to robots with approximately linear actuators [Hirio,84]. To date, little consideration has been given to the behaviour of variable structure controllers when applied to systems with nonlinear actuators, particularly those with Coulomb friction. This has meant that for low cost applications, few results using VSC techniques are available.

The fact that variable structure controllers can result in behaviour which is insensitive to the effects of certain forms of nonlinearity has led to the belief that, with certain developments, the approach may be suitable in the situation of low cost electro-mechanical systems. This then is the motivation for the work reported here, in which variable structure control techniques have been developed and applied to systems which have not performed well when used with better established control methods; namely low cost robot manipulators with significant drive system nonlinearities.

1.2 Control of the RTX Robot

The RTX is a low cost, six degree of freedom robot, with D.C. motors providing the power and timing belts used extensively in the transmission system. As supplied, control of the joints is performed using two micro-controllers, deriving position information from low resolution incremental optical encoders and powering the motors via pulse width modulation, voltage mode drives. The controller implemented on the micro-controllers uses a simple PID algorithm for each joint.

Owing to the low cost nature of this mechanism, the actuator and

transmission systems have characteristics which are far from ideal, in particular, the friction in the motors and transmissions is highly nonlinear and accounts for a significant proportion of the torque generated. As a result, the PID control algorithm must be significantly sub-optimal in performance and was found to be inadequate for many of the tasks required of the robot, in particular for high speed point-to-point movement and trajectory following at all speeds. This inadequacy was found to be particularly severe in tasks involving movement in a horizontal plane using the shoulder, elbow and wrist yaw joints; a type of movement which is frequently required.

The shoulder and elbow joints of the RTX are the ones which suffer the greatest effect from interaction dynamics, while also having the nonlinear actuator behaviour found in other joints, and so it was decided to investigate the use of new control techniques on these joints, and the yaw axis, as they provide the greatest challenge and, if successful, would find an immediate application in which they could be thoroughly tested.

Although the motor drive system and encoder output signals could be obtained by making minor modifications to the existing hardware of the RTX, it was felt that a significant improvement in speed could be obtained by replacing the existing motor drives by linear current drives. Improved controller performance required a higher resolution of position sensing, once again using optical encoders but of a higher resolution than those employed by the manufacturer.

In order to develop new control algorithms it was necessary to have an understanding of the dynamic behaviour of the manipulator and so a large amount of time was spent in developing and testing such a model. The resulting model was found to be of a high order, with many nonlinear elements, and in order to perform simulations in a reasonable time, it was found to be necessary to make alterations to the computer aided control system design suite used. These modifications involved alteration of the system model structure to allow simulation of the interaction dynamics and the implementation

and comparison of a number of numerical algorithms for the solution of nonlinear ordinary differential equations.

The two main areas of scope of this thesis are the development of the dynamic model of the three joints of the RTX considered here and the application of variable structure controllers to these joints. Details of the associated work carried out to support this work will only be described in as far as it aids understanding of these areas.

1.3 Outline of Subsequent Chapters

The remaining chapters of this thesis are divided up as follows:

Chapter two considers the development of the theory of variable structure control systems, from their earliest conception to the present state of the theory. The emphasis throughout is on the implementation aspects of these theories, in particular to systems with significant nonlinearities. A large part of this chapter considers the effects of the forms of nonlinear behaviour found in low cost manipulators on the performance of existing VSC algorithms and goes on to develop new algorithms which overcome these problems.

Major changes to the actuation and sensor systems used on the RTX have been implemented, and these are described in chapter three. This covers the change from low to high resolution optical encoders and the associated counting circuits for position and velocity sensing; the design of linear current mode motor drives and the computer interfaces required to allow their use with both a DEC μ PDP and an IBM compatible personal computer used for control algorithm implementation and testing.

The development of the dynamic model of the RTX is expounded in chapter four. This is divided into the background theory of the model and the experimental derivation of model parameters. The majority of this chapter is concerned with modelling of the actuation and transmission system dynamics, since these have a great influence on the controller performance and have not

been considered elsewhere. The interaction dynamics have been widely studied and so the theory will only be briefly reviewed while the derivation of interaction model parameters will receive more attention since the techniques used are not well established. The major non-ideal effects found in the actuation and transmission system are nonlinear friction in the motors and joints and flexibility in the timing belt drive system. Both these aspects receive detailed attention in this chapter.

Chapter five considers the developments made to a computer aided control system design suite. A number of minor improvements are considered followed by details of the change in model structure to allow the interaction dynamics to be studied. This chapter also describes the work carried out on the implementation and testing of a number of solution algorithms for ordinary differential equations. These are compared with an algorithm designed specifically for the generation of phase portraits which had not previously been fully tested.

The implementation of a large number of variable structure control algorithms is considered in chapter six. The performance of these controllers is compared with a PID algorithm implemented using the same hardware, and in some cases with simulations of the performance using the model described in chapter four. A study of the suitability of different algorithms for this application concludes this chapter.

Chapter seven provides an evaluation of the project, in terms of the developments to the computer aided control system design suite, the modifications to the RTX hardware, the development of the dynamic model and controller design and implementation. The final chapter, chapter eight draws conclusions from the work described.

CHAPTER 2

The Theory of Variable Structure Control Systems

Variable Structure Control (VSC) systems provide a technique which may be applied to linear and nonlinear, time invariant and time varying plants for which a limited knowledge of the dynamics is available and provides performance which is often faster and more robust to parameter variations and disturbances than other, better established control schemes. This improvement in performance is particularly apparent in systems which employ a sliding mode, in which a particular form of dynamic behaviour is imposed on the plant; it will be this type of control which will mainly interest us here.

This chapter will be concerned with the development of the theory of VSC, beginning with the most simple type designed for low order, well modelled systems and progressing to those suitable for more complex, nonlinear systems with poorly modelled, high order dynamics. At all stages the emphasis will be on the suitability of the particular form of controller for application in the field of robotics, although examples from other relevant fields will also be considered. Following the development of the basic theory of VSC systems, a description of the types of controllers implemented on robots will be given, detailing the benefits and failings of each.

2.1 Introduction to the Theory of Variable Structure Control Systems

In this section we shall consider the general theory upon which variable structure control systems are based. We will begin with a brief history of the early development of the principles used in controller design. This will be followed by a discussion of the present state of the theory, beginning with the application of VSC techniques to well modelled linear systems and continuing to

consider the effect of unmodelled dynamics on controller performance. Having identified the problems that these plant characteristics introduce we will consider a number of approaches available to eliminate their effects, while maintaining some of the benefits of VSC systems. The effects of certain types of nonlinear behaviour, particularly common in electro-mechanical systems, will be considered along with various techniques which may be employed to alleviate these effects.

2.1.1 Historical Development of Variable Structure Control Theory

The earliest research work on the theory of variable structure control systems was carried out almost exclusively in the USSR in the early 1960s. Of the early researchers in this field Emel'yanov, Itkis, Barbashin and Utkin and their co-workers were the most prominent. This research was aimed mainly at the development of the mathematical background to VSC systems rather than at their practical implementation. Their efforts resulted in a well developed theory suitable for the design of controllers for high order, time varying systems with some forms of nonlinear behaviour. Certain forms of nonlinearity were not considered, nor were the implementation problems involved with this type of controller. Summaries of the results obtained in this work are given by Itkis [Itkis,76] and Utkin [Utkin,77].

It was not until about 1975 that this work came to the attention of workers in the West. From this date, a large amount of work was carried out, in both East and West, on the application of these theories to the design of controllers in a wide variety of situations. One of the first applications to be considered was that of induction motor control in equipment ranging from CNC machinery to electric vehicles. A summary of these and many other applications are given in [Utkin,83]. More recently, many research workers in Japan have begun looking at the use of VSCs, generally for induction motor control. The prominent figures in this area are Hashimoto et al [Hashimoto,87,88], Dote et al [Dote,87] and Lin [Lin,88].

2.1.2 Fundamental Principles of Variable Structure Control Systems

Variable structure control systems may be divided into sliding and non-sliding types, of which the former will be of the greater interest to us since they possess many desirable properties. We will, in the following sections, consider the difference between these types of control and then go on to describe the situations in which sliding behaviour occurs. Sliding and reaching conditions which are general to all continuous VSC systems will be established.

2.1.2.1 Sliding and Non-sliding Variable Structure Systems

A variable structure control system may be classed as any design in which the structure of the controller changes during the transient process. There are many established control schemes which may be considered to operate in this way; for instance a PID control may only use integral action when the error becomes sufficiently small. There is a further distinction which may be made between non-sliding and sliding controllers. In the former case there are only a small number of changes in structure in each transient (one in the case of the PID control described above) while in the case of a sliding variable structure controller there are a large number (approaching infinity in the ideal case) of rapid changes. In the former the overall behaviour of the system is compounded from large segments of the individual structures while in the latter the behaviour is defined by the way in which the switching between structures occurs with, in the ideal case, no influence from the individual structures.

As an example of a non-sliding variable structure control consider the system defined by figure 2.1. There are two possible system structures, dependant on the position of the switch. Time responses for the two systems are shown in figure 2.2. In the first case (i) a fast transient occurs but the response is undamped and continuous oscillation occurs. In the second case (ii) there is no overshoot but the transient is far slower. By starting a movement

under regime (i), a fast transient is obtained and then, by switching to regime (ii) as the desired point is approached, the system becomes highly damped and and no overshoot occurs, as shown in figure 2.3. The improvement in the transient behaviour of this system is subject to parameter variations and external disturbances.

Figure 2.1 Double integrator plant with zero and non-zero damping

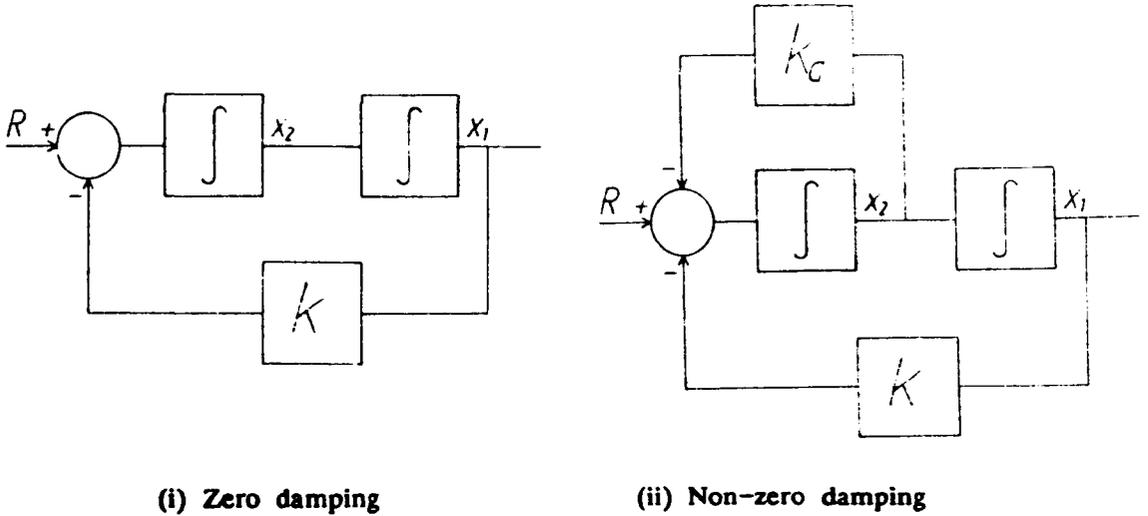


Figure 2.2 Effect of switching velocity feedback in double integrator plant

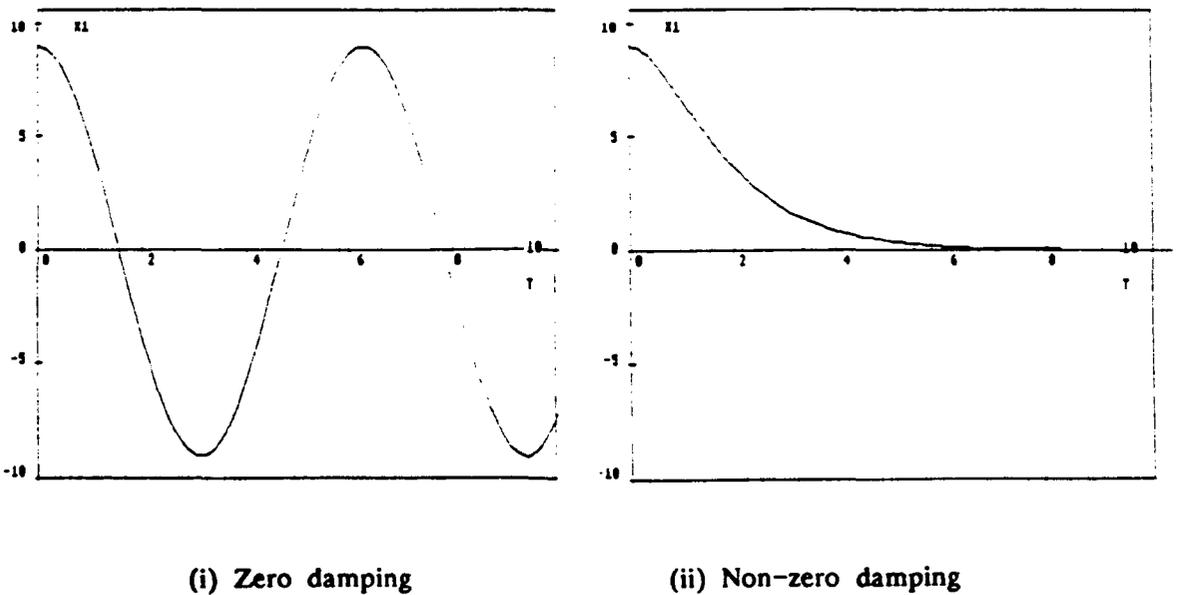
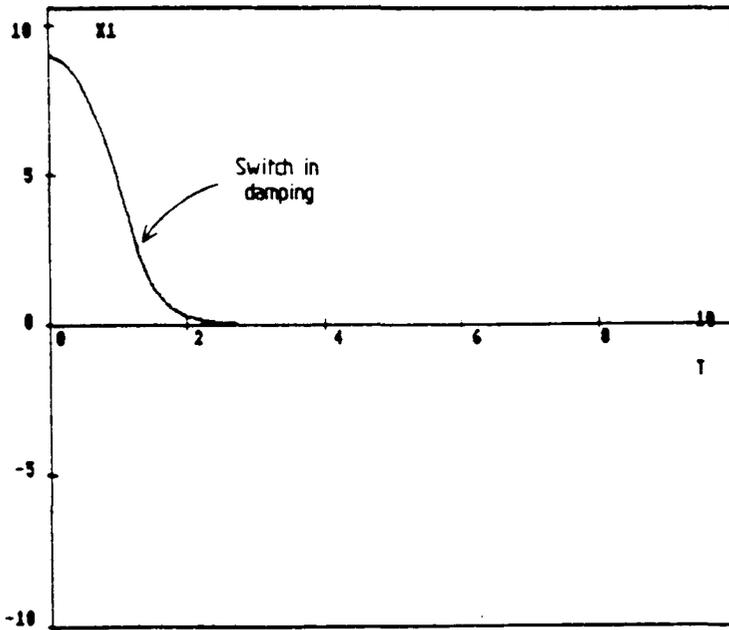


Figure 2.3 Trajectory of system with switched velocity feedback



As an example of a simple sliding mode VSC consider the same plant but with either positive or negative feedback, as shown in figure 2.4. Once again, two types of behaviour are possible as shown in figure 2.5. Neither of these structures gives desirable characteristics, in the first case continuous oscillation occurs and in the latter unstable motion tending to $\pm\infty$.

Figure 2.4 Double integrator plant with switched sign feedback

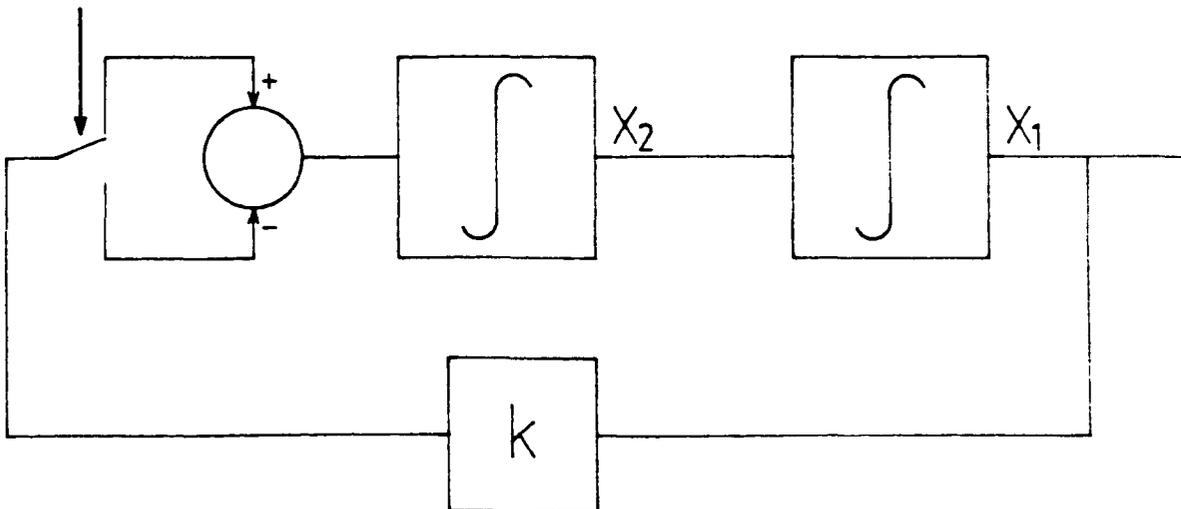
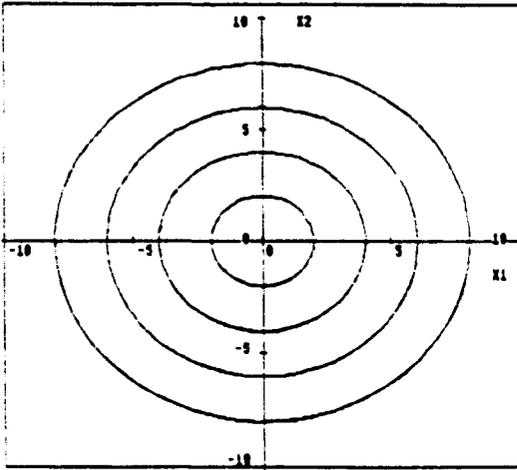
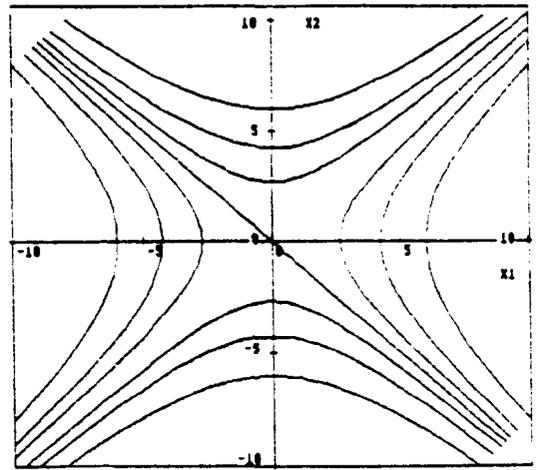


Figure 2.5 Phase trajectory of double integrator plant with positive and negative feedback



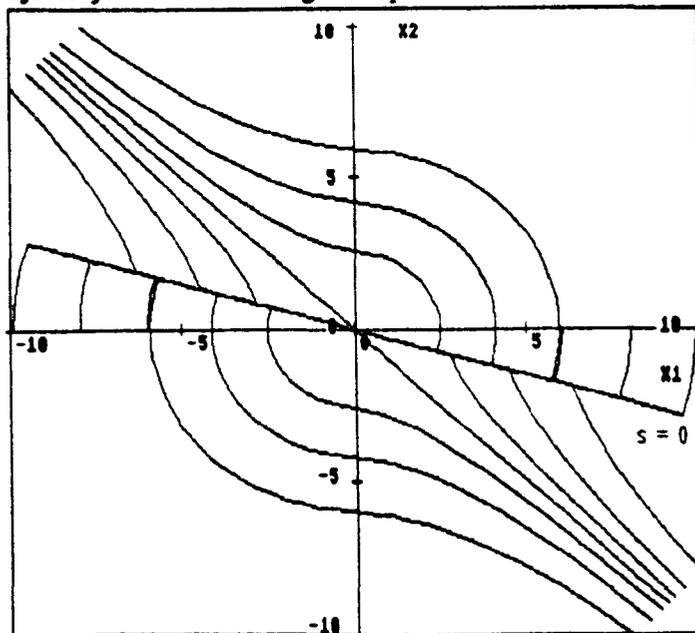
(i) Negative feedback



(ii) Positive feedback

If we define a line through the origin of the phase plane between the $x_2 = 0$ axis and the asymptote of the type (ii) system and switch between the two control regimes as the trajectory crosses this line then on each side of the line the state velocity vector will point towards the line. Switching between the two structures will occur at an infinite speed and the trajectory will travel along the line itself (or within an infinitesimally small region around it), proceeding towards the origin, as shown in figure 2.6.

Figure 2.6 Trajectory of double integrator plant under VSC

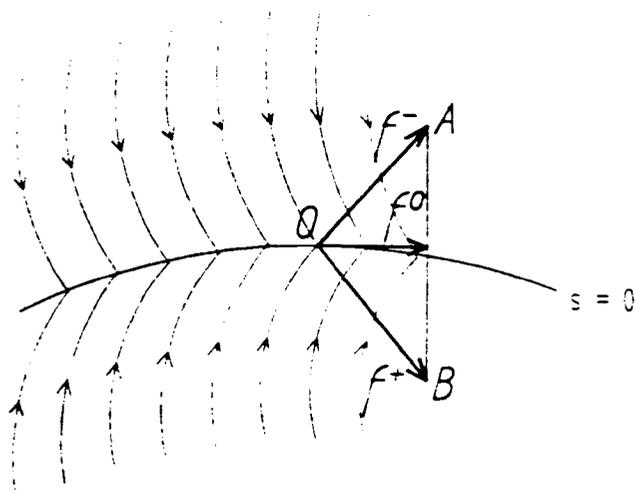


It is this high speed switching of the structures which causes the trajectory to 'slide' along the line, leading to the title of this type of behaviour. Once the sliding mode is established we wish to know how the trajectory will behave. This is most conveniently achieved by use of the idea of equivalent control, introduced by Fillipov [Itkis,76]. Consider the situation illustrated in figure 2.7 in which the switching line is shown, along with the phase velocity vectors, f^+ and f^- on either side, passing through the point O. Joining the ends of these two vectors by the line AB then joining O to this line by the vector f^0 , tangent to the switching line at point O, gives the state velocity vector of the system in the sliding mode. This can be described by:

$$f^0 = \mu f^- + (1-\mu)f^+$$

where $0 < \mu < 1$. μ may be considered as the proportion of time for which each structure of control is required in order to keep the trajectory on the switching line. There is an equivalent control u^0 , which satisfies $u^- < u^0 < u^+$ or $u^- > u^0 > u^+$, which produces the same effect as switching between u^- and u^+ in the ratio $\mu:(1-\mu)$. This equivalent control, although it would control the nominal system in the same manner as the VSC does not provide the disturbance rejection properties of the VSC and is only used in describing the ideal system behaviour.

Figure 2.7 Definition of equivalent control



2.1.2.2 Sliding and Reaching Conditions

It is clear that, in order for sliding behaviour to occur, the state velocity vector on either side of a switching surface must be directed towards that surface. If we define the switching surface to be $s = 0$, where s is a function of the state variables, then it will be seen that this condition may be written as $s\dot{s} < 0$. In this form the condition guarantees at least asymptotic hitting while a stronger condition, $s\dot{s} < -\eta$ where η is an arbitrarily small positive constant, assures hitting in finite time.

Since s is a function of the states, its derivative may be found and putting this into the sliding condition will lead to inequalities involving the controller parameters from which suitable values may be found. As an example of the procedure involved, consider the example given in section 2.1.2.1 where:

$$\dot{x}_1 = x_2 \quad (2.1)$$

$$\dot{x}_2 = b u$$

If we define a switching function to be

$$s = cx_1 + x_2 \quad (2.2)$$

and define the control as:

$$u = \begin{cases} -ax_1 & \text{if } sx_1 > 0 \\ ax_1 & \text{if } sx_1 < 0 \end{cases} \quad (2.3)$$

then

$$\dot{s} = c\dot{x}_1 + \dot{x}_2 = cx_2 - abx_1 \operatorname{sgn}(sx_1) \quad (2.4)$$

This condition covers the whole of the phase plane and it is clear from figure 2.6 that in some regions it is not possible to drive the trajectory towards

the switching line, for instance when approaching the $x_1 = 0$ axis under positive feedback, and so we are only concerned at this stage that, once the trajectory has reached the switching line it should remain on it. Now on the switching line $s = 0 = cx_1 + x_2$ and substituting this into equation 2.4 we obtain:

$$\dot{s} = (-c^2 - ab \operatorname{sgn}(sx_1)) x_1$$

There are four situations which must be considered, depending on the signs of s and x_1 . These are as follows:

If $s > 0$

$$x_1 > 0 \quad \dot{s} = (-c^2 - ab) |x_1| \quad (i)$$

$$x_1 < 0 \quad \dot{s} = (c^2 - ab) |x_1| \quad (ii)$$

If $s < 0$

$$x_1 > 0 \quad \dot{s} = (-c^2 + ab) |x_1| \quad (iii)$$

$$x_1 < 0 \quad \dot{s} = (c^2 + ab) |x_1| \quad (iv)$$

If the inequality, $ss' < 0$ is to be fulfilled these simplify to two conditions, the first given by situations (i) and (iv) and the latter by (ii) and (iii).

$$ab > -c^2$$

and

$$ab > c^2$$

The former condition is less restrictive than the latter and so may be ignored. Using the latter conditions the selection of control parameters, c and b may be made. Using the same procedures sliding conditions may be established for more general, higher order, MIMO and nonlinear plant controllers, as illustrated by Utkin [Utkin,77] and Itkis [Itkis,76].

Having shown that, once established, a sliding regime may be maintained, given suitable control parameters, we must show that a switching surface will be reached from any initial point in the phase plane. Itkis [Itkis,76] shows that the representative point (RP) will reach the switching line provided the system with negative feedback is not aperiodically unstable, i.e. the characteristic equation has no non-negative real roots. This condition is not generally difficult to satisfy for practical systems, given a choice of control parameters.

2.1.3 Properties of Systems with Sliding Modes

Having shown how a sliding mode may be generated and maintained, it remains to show the the type of behaviour that such a regime may introduce. Accordingly, we will discuss the dynamics of a system in a sliding mode and the effect on these dynamics of parameter variations and disturbances.

2.1.3.1 Dynamics of the Sliding Mode

We will begin by considering the simple second order system given by equation 2.1, with the straight switching line defined in equation 2.2. We have shown that subject to appropriate choice of control parameters, the trajectory will remain on the switching line once it is reached and so will follow the equation $s = 0$, i.e.

$$s = 0 = cx_1 + x_2$$

From equation 2.1, we also know that $x_2 = \dot{x}_1$ and so:

$$\dot{x}_1 = -cx_1$$

This is the differential equation of a linear first order system having the solution:

$$x_1(t) = x_1(t_0) e^{-c(t-t_0)} \quad (2.5)$$

where t_0 is the time at which the sliding regime begins and $x_1(t_0)$ is the value of x_1 at that time.

We have obtained a stable, first order, linear response from two systems which were unstable or marginally stable. The time constant of this motion is selected by the designer in the choice of controller parameter c . It will be seen that the controlled system order is one lower than that of the plant itself; this order reduction is also a feature of variable structure systems of higher order and is one of the major benefits of VSC techniques.

When considering the design of a VSC system it is often required that the maximum speed of response be obtained for a given movement. From equation 2.5 it is seen that the time constant of motion once in the sliding regime is equal to $1/c$ and so a large value of c results in a fast response. The desire to increase c does however conflict with the sliding condition which requires that $c < (ab)^{\frac{1}{2}}$. This conflict may be resolved by increasing the gain, b , up to the point where a sufficiently high value of c , and hence speed of response, is obtained. Unfortunately, this is often not possible in practice, for reasons which will be explained in sections 2.1.4 and 2.1.6.1.

2.1.3.2 Robustness of Sliding Mode Behaviour

It will be noted that none of the plant and only one of the control system parameters are present in equation 2.5 and so it is clear that we have imposed the behaviour we required upon the system by our choice of c . The system will remain insensitive to parameter variations, plant nonlinearities and external disturbances as long as the sliding mode is maintained. Itkis [Itkis,76]

shows that for general, higher order systems with bounded time varying parameters and bounded external disturbances VSCs may be designed which give responses which are invariant to these influences.

The trajectory of a variable structure control system is made up of two component parts, the initial reaching phase in which the control signal is continuous or has a finite number of discontinuities and the behaviour is controlled by the dynamics of the subsystem which is operative. In this regime the behaviour is subject to parameter variations and disturbances in the same way as a normal system with the particular control system would be. It is not until the trajectory reaches the switching surface and enters a sliding mode that the behaviour is defined by the switching surface dynamics and becomes insensitive to these disturbances and parameter changes. Once in the sliding mode the input to the system becomes discontinuous and, in the ideal case switches at infinite frequency between two continuous levels.

Clearly the behaviour of the system once in the sliding mode is preferable to that in the reaching phase and this has led to the desire to make the reaching phase as short as possible, with the majority of the transient being in a sliding mode. This has led some researchers to suggest the use of a straight switching line which passes close to the trajectory starting points (the zero velocity axis in a typical second order position control system). This has the effect of greatly increasing the sliding mode time constant and hence slows the entire transient. Tenreiro-Machado [Tenreiro-Machado,88] suggests the use of a curved switching surface which passes through any initial point in the state space and gives a second order overdamped response leading to the origin. In this case the control signal is not discontinuous across the switching line and so robustness to parameter changes is not maintained. A preferable switching line definition adopted by Hashimoto [Hashimoto,87] divides the trajectory into four segments, as illustrated in figure 2.8. These segments are an initial parabolic acceleration; a constant velocity; a parabolic deceleration and a final exponential deceleration.

The switching function is defined by:

$$s_1 = cx_1 + x_2$$

$$s_2 = x_2^2 - \alpha(x_1 - e_0)$$

$$s_3 = x_2 - V$$

$$s_4 = x_2^2 + \beta(x_1 - e_1)$$

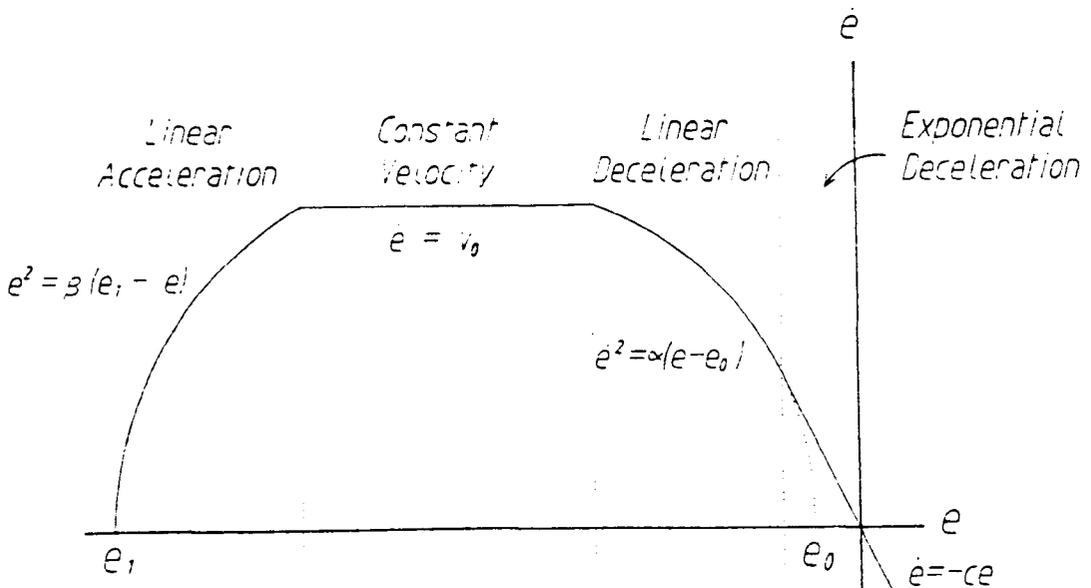
where

$$x_1 = e$$

$$x_2 = \dot{e}$$

where α , β and V are the predetermined deceleration, acceleration and velocity while c is the time constant of the exponential deceleration.

Figure 2.8 Switching curve used to avoid reaching phase



This type of switching function does provide a complete trajectory which is insensitive to parameter variations and disturbances. In order to assure sliding behaviour on all parts of the curve in the presence of these effect however, the curves must be sub time optimal, the degree of sub optimality being dependant on the extent of the disturbing influences.

This technique provides a means of obtaining invariance, but only at the

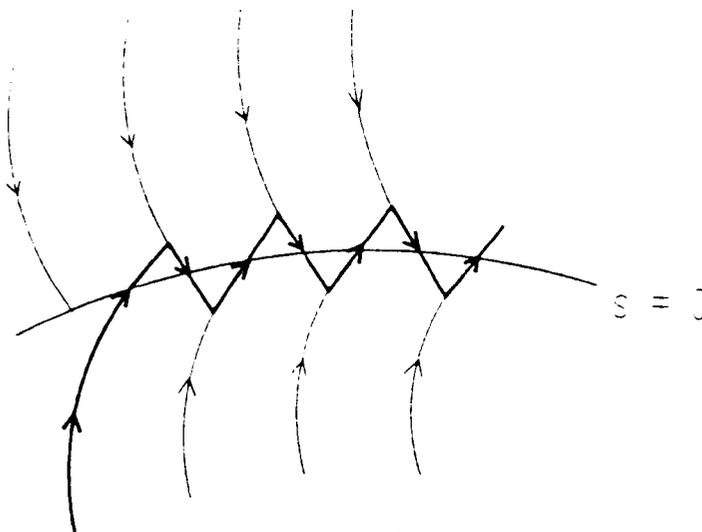
cost of a slower response. It must therefore be decided which is more important and the switching function chosen accordingly.

2.1.4 Non-ideal Sliding Behaviour

The example system given by equation 2.1 is not representative of practical systems upon which we wish to use VSC techniques. Practical systems are not of this simple form and the differences between the true system dynamics and the model used to represent them have a major influence on the VSC performance.

It has been shown that for an ideal variable structure system in which all the dynamics are represented, the RP will remain in an infinitesimal region around the switching line once sliding has begun. In practical systems however this is not the case and unmodelled dynamics and time delays in the control loop result in what is called chattering behaviour. This takes the form of small ripples on either side of the switching line, as illustrated in figure 2.9 which shows the behaviour of a system with a pure time delay in the control loop which might typically be due to computation of the control signal required. This is the simplest form of non-ideal behaviour to analyse but is also one of the most difficult to overcome.

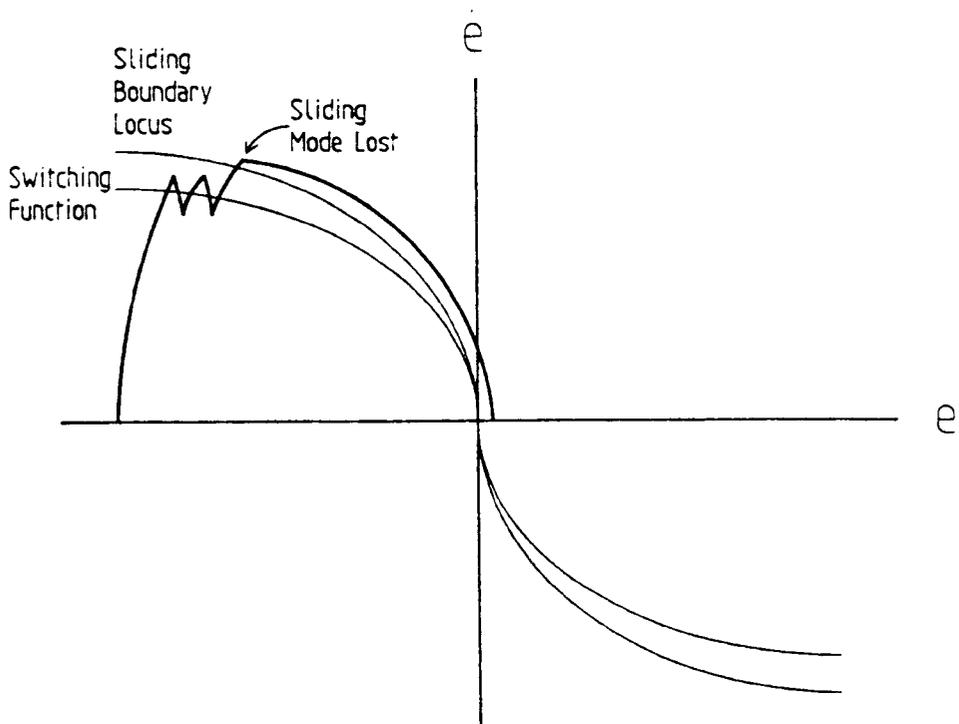
Figure 2.9 Chattering behaviour in variable structure control systems



The precise form of the chatter is dependant on the nature of the unmodelled dynamics and the time delays present. Chatter is undesirable because it involves a large amount of control activity and may excite unmodelled dynamics. In addition, in mechanical systems, this high frequency oscillation may cause excessive gear wear and fatigue.

Chattering behaviour must be taken into account when selecting a suitable switching line. The deviation from the switching surface must be sufficiently small to prevent the RP leaving the region in which the the sliding condition is satisfied. If this is not the case the trajectory may leave the switching line for an extended period and the advantages of sliding motion will be lost, as shown in figure 2.10.

Figure 2.10 Loss of sliding due to chatter



When designing a VSC, it is important to have at least an approximate value for the chatter amplitude. This may be found from the dynamics of the parts of the plant not used directly in the selection of control parameters if these are known, using either simulation techniques or, very approximately, from the state velocity vector and the estimated time constant of these

dynamics. The figures derived should not be relied upon unless the dynamic model is highly accurate since unmodelled dynamics may have a significant effect. It is advisable, in general to perform some experimentation to determine the extent of the chatter. It should be noted that, in general, an increase in controller gain leads to an increase in chatter amplitude, and so it is often advisable to reduce the gain as far as possible, while maintaining sliding behaviour.

Although chatter is unavoidable when the VSC system described by equation 2.3 is applied to a practical system, there are a number of modifications to the basic formulation which allow this behaviour to be completely eliminated. These techniques are the subject of the following section.

2.1.5 Chatter Reduction Techniques

The cause of chatter in variable structure systems is the discontinuous switching of the control signal across some switching surface, causing rapid changes between two continuous controls. The amplitude of the chatter may be reduced by either decreasing the height of this discontinuity or by smoothing the switching to make the control continuous. These two approaches are considered in the following sections.

2.1.5.1 Compensation Techniques for Chatter Reduction

In the simple formulation of a VSC as given by equation 2.2 the whole of the control signal is switched between two levels so that a large discontinuity in the control signal occurs when the switching line is crossed. It is possible to reduce the height of this discontinuity by the use of a control signal constructed of two parts, one continuous and the other discontinuous. In this way a continuous signal is used to control part of the dynamics while the discontinuous part controls the remainder. Typically this would involve a

continuous control function to control the most significant elements of the dynamics, with the effects of the remaining dynamics and external disturbances eliminated using a discontinuous control. The choice of which part of the dynamics to tackle with which part of the control signal is at the discretion of the engineer, subject to the constraints of performance criteria, implementation costs and the significance of parameter variations and disturbances. Often, the least expensive solution is to use a large switched component, since little information is required about the dynamics and the amount of computation required is small. This does however lead to a large amount of chatter and if this cannot be accepted, a controller which compensates for the dynamics more accurately must be implemented.

2.1.5.2 Smoothing Techniques for Chatter Reduction

Rather than attempt to reduce the amplitude of the discontinuity in a VSC, it is possible to introduce a region around the switching line in which a continuous transition between the two states occurs. There are a number of ways in which both the switching region and the smoothing may be defined. An early exposition of this type of control was provided by Slotine [Slotine,83], who proposed using a boundary layer around the switching line of the form shown in figure 2.11 with a control signal consisting of switched state and error feedback and a relay function when outside the band. When within the switching band the control signal is found from a linear interpolation between the two values on either side of the boundary layer, as shown in figure 2.12.

In this way, the discontinuity in the control is removed. The behaviour of the system under this type of control is no longer true sliding since the trajectory is not restricted to the switching line itself but may move around within the band. The deviation of the RP from the switching line is dependant on the boundary layer width, the wider the region the greater the deviation that will occur, while a layer which is too narrow will cause chattering

behaviour to be reintroduced in the presence of unmodelled dynamics. As shown in section 2.1.5.3, a trade off is obtained between the frequency range of unmodeled dynamics and tracking accuracy. Similarly, the desire for improved tracking accuracy leads to greater control activity, resulting in chattering behaviour if very high accuracy is sought.

Figure 2.11 Boundary layer around the switching line

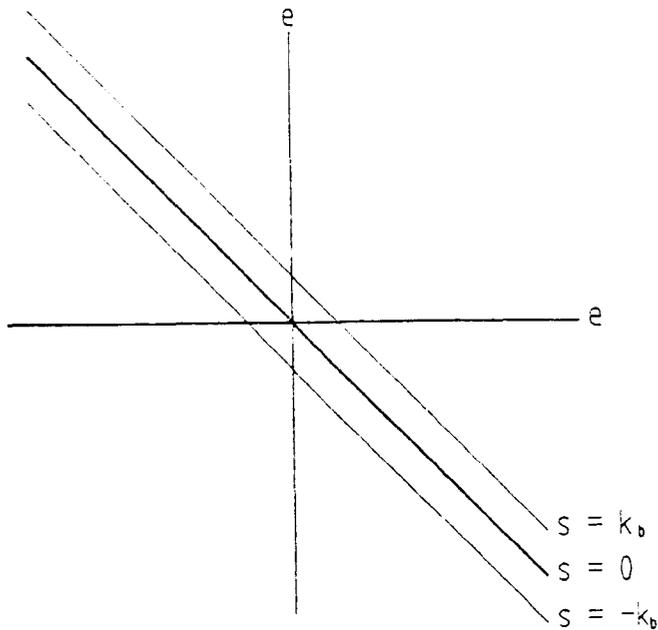
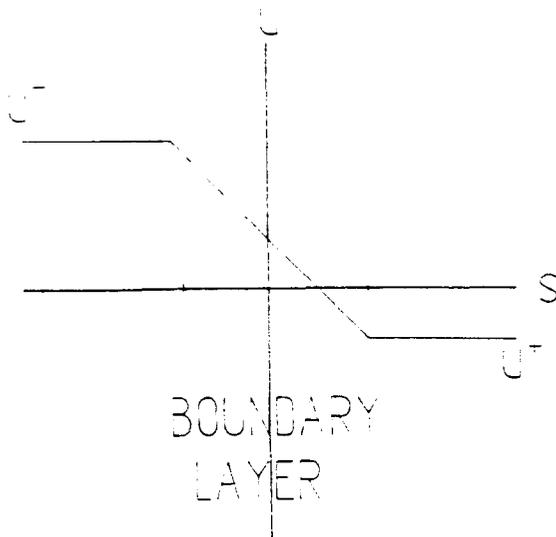


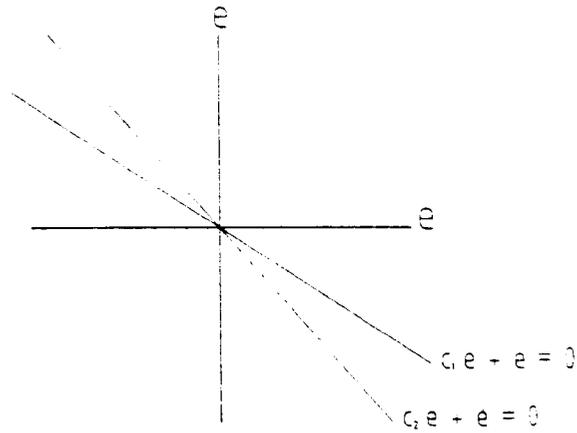
Figure 2.12 Sample interpolation used to smooth control within the boundary layer



There are a number of variations on this idea such as using sliding sectors, as proposed by Zinober [Zinober,88] and Yeung [Yeung,88], where the

phase plane is divided up as shown in figure 2.13, with switched control activity outside the sectors and a smoothed signal within.

Figure 2.13 Phase plane with sliding sectors

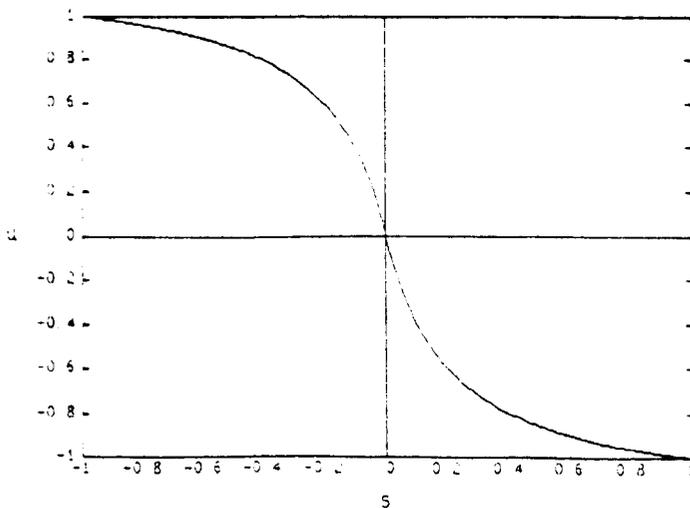


Yeung suggests an interpolation of the form introduced by Slotine and described above, while Zinober uses a control signal found from:

$$u = \frac{-k s}{|s| + \delta}$$

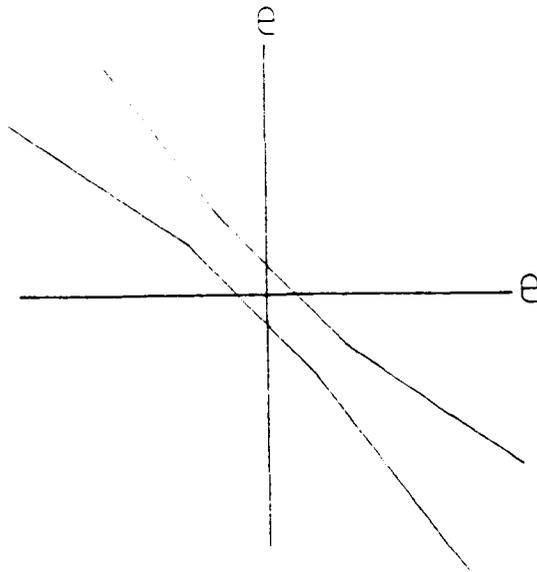
Where k and δ are positive constants. This smoothing function takes the form shown in figure 2.14

Figure 2.14 Smoothing function used by Zinober



It should be noted that as the trajectory approaches the origin in this form of control the sliding sector becomes vanishingly small, returning us to the situation of discontinuous control. In order to rectify this problem a combination of sectors and bands may be adopted in the form shown in figure 2.15

Figure 2.15 Combined sliding sectors and bands



There are similarities between this smoothed VSC and traditional PD control, if certain control system structures are selected. Consider for example the situation where a straight switching line is used, defined by $s = cx_1 + x_2$, and the control input is defined as $k \operatorname{sgn}(sx_1)$ outside the boundary layer with a linear interpolation inside. Within the boundary layer the control signal is thus proportional to s and may be written:

$$u = -ks = -k(cx_1 + x_2) = -kcx_1 - k\dot{x}_1$$

This is the same equation as used in a PD control with a proportional constant of kc and a derivative constant of k . Clearly this is a small subset of the variable structure controllers with smoothing functions which may be investigated and can be improved upon in many respects, but it is conceptually simple and so is often considered.

Compensation and smoothing techniques are not mutually exclusive and indeed are most effective when combined, with compensation techniques used to minimise the discontinuous component and smoothing applied to the remaining switched control. A control structure of this form will be considered in some detail since it involves both forms of chatter reduction technique.

2.1.5.3 Combined Smoothing and Compensation Chatter Reduction (Slotine's Algorithm)

The algorithm suggested by Slotine [Slotine,83] combines the advantages of compensation and smoothing chatter elimination techniques in a way which allows a trade-off between model accuracy and tracking error to be established. The controller will be described for a nonlinear plant with fixed input gain, although the case of time varying gain involves only a simple extension of the theory. The plant is described by the equation:

$$\ddot{x}^{(n)} = f(X,t) + u + d(t)$$

where $X = [x, \dot{x}, \dots, x^{(n-1)}]^T$ is the state vector, u the control input, $f(X,t)$ gives the nonlinear dynamics while $d(t)$ represents external disturbances. $f(X,t)$ is modelled by $\hat{f}(X,t)$ for which the following precision condition is imposed:

$$| f(X,t) - \hat{f}(X,t) | < \Delta f(X,t)$$

also

$$| d(t) | < D(t)$$

If we wish the state to track a specified state, $X_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]^T$, then we may define a tracking error vector, \tilde{X} where

$$\tilde{X} : X - X_d = [x, \dot{x}, \dots, x^{(n-1)}]^T$$

We must also insist, for the moment, that

$$\bar{X}|_{t=0} = 0$$

We may define a time varying switching surface, $S(t) : s(X,t) = 0$, where

$$s(X,t) = \left\{ \frac{d}{dt} + \lambda \right\}^{n-1} \bar{x}, \quad \lambda > 0$$

We then need to choose a control, u which satisfies the sliding condition $s\dot{s} < -\eta$. This control must be discontinuous if precise tracking is to be obtained in the presence of disturbances, although a continuous part can also be used. The control structure suggested is of the form:

$$u = - \hat{f}(X,t) - \sum_{p=1}^{n-1} \left[\begin{matrix} n-1 \\ p \end{matrix} \right] \lambda^p \bar{x}^{(n-p)} - k(X,t) \text{sgn}(s) \quad (2.6)$$

where

$$\left[\begin{matrix} \beta \\ \alpha \end{matrix} \right] = \frac{\beta!}{\alpha! (\beta-\alpha)!}$$

is the number of combinations of β objects taken α at a time. In order to satisfy the sliding condition, the discontinuous part of equation 2.6 must be greater than the total effect of modelling errors and disturbances, i.e.

$$k(X,t) > F(X,t) + D(x,t) + v(t) + \eta$$

where

$$F(X,t) > | \Delta f(X,t) |$$

and

$$v(t) > | x_d^{(n)} |$$

This type of control may be considered as being composed of a continuous

part which compensates for the dynamics as far as the modelling precision will allow, with the remaining dynamics controlled using the discontinuous part, $k(X,t)$.

This controller assures perfect tracking by means of discontinuous control activity. This however leads to chattering in the presence of unmodelled dynamics and so a modification to the algorithm is introduced which removes this behaviour while maintaining tracking to within a prescribed limit. The removal of chatter is achieved by the introduction of a linear smoothing function within a boundary layer around the switching line, as described in section 2.1.5.2. The control signal then becomes:

$$u = - \hat{f}(X,t) - \sum_{p=1}^{n-1} \binom{n-1}{p} \lambda^p \bar{x}^{(n-p)} - k(X,t) \text{ sat} \left[\frac{s}{\lambda^{n-1} \epsilon} \right]$$

where the $\text{sat}()$ function is defined as:

$$\text{sat}(q) = \begin{cases} q & \text{if } |q| < 1 \\ \text{sgn}(q) & \text{if } |q| > 1 \end{cases}$$

Outside the boundary layer the control is the same as equation 2.6 and so the boundary layer is attractive and the trajectory will enter the boundary region. In order to assess behaviour within the band, we must find a relationship for the derivative of s . We note that

$$\dot{s} = x^{(n)} - x_d^{(n)} + \sum_{p=1}^{n-1} \binom{n-1}{p} \lambda^p x^{(n-p)}$$

Within the band the result of the sat function is equal to its argument and so, by substituting $x^{(n)}$ into the equation for \dot{s} , we may write

$$\dot{s} = - k(X,t) \left[\frac{s}{\lambda^{n-1} \epsilon} \right] + (\Delta f(X,t) + d(t) - x_d^{(n)}(t))$$

If we assume that tracking to within a small error, ϵ has been achieved (i.e. the RP is within the boundary layer) and that $\Delta f(X,t)$ and $k(X,t)$ are continuous then we may rewrite this equation as:

$$\dot{s} = -k(X_d, t) \left[\frac{s}{\lambda^{n-1}\epsilon} \right] + (\Delta f(X_d, t) + d(t) - x_d^{(n)}(t) + O(\epsilon))$$

It will be seen that s is the output of a stable first order filter, with dynamics dependant on the desired state and possibly explicitly on time; and input made up of the modelling errors, disturbances and n^{th} order derivative of the desired trajectory. These 'perturbations' are all bounded and so $k(X_d, t)$ may be upper bounded by k_{max} . We may then think of $k_{\text{max}}/\lambda^{n-1}\epsilon$ as the effective 'break frequency' of this filter, while λ is the break frequency of the overall controlled system (the switching line gradient in the second order case). Both λ and $k_{\text{max}}/\lambda^{n-1}\epsilon$ must be significantly below the frequency of unmodelled dynamics and the controller sampling frequency, which in turn limits the best attainable tracking accuracy. If we demand that the dynamics of s are faster than the switching surface dynamics, i.e.

$$\frac{k_{\text{max}}}{\lambda^{n-1}\epsilon} < \lambda$$

we find that the best possible guaranteed tracking accuracy is:

$$\epsilon = k_{\text{max}}/\lambda^n$$

Following this design procedure gives a control system which produces the minimum tracking error, given a certain model precision. This involves compensating for all the known dynamics while controlling the variable and unknown dynamics with a smoothed discontinuous part. This requirement that all known dynamics be compensated for may result in a computationally expensive control algorithm, leading to the need for high cost equipment or low sampling rates, which in turn leads to poorer tracking. It may therefore be more effective to only compensate for those dynamics which are most significant and may most easily be treated. This can lead to a more efficient algorithm with the associated high sampling rate and improved tracking performance. An example of this situation is provided by the robot control

problem in which there are a number of sources of undesirable dynamics. The dynamics of the actuation system generally involve functions of a small number of variables and so may relatively easily be compensated for while the interaction dynamics are of a form, such that compensation requires a large amount of calculation. Depending on the manipulator considered the relative significance of these two forms may vary widely. Although it has generally been assumed that the interaction dynamics must be compensated for, it is often more efficient to only compensate for actuation system dynamics while treating interaction dynamics as disturbances.

2.1.6 Variable Structure Control of Nonlinear Systems

The procedures for the design of VSC systems described above were developed with linear and smooth nonlinear systems in mind and although the robustness characteristics achieved make them insensitive to certain other forms of nonlinearity, there are other types which if not taken into account can result in severe degradation of performance. Two forms of nonlinearity found in robot actuation systems, which have an undesirable effect will be discussed in the following sections along with methodologies which may be adopted to overcome these problems.

2.1.6.1 Effect of Input Saturation on Variable Structure Control Systems

No practical system is capable of producing infinite control activity and if this fact is not taken into account in the design of the VSC it may be found that the sliding condition is not satisfied at all points on the switching line. If a linear system of the form described by equation 2.1 with the controller defined by equation 2.2 is considered then it appears from the sliding condition that any value of c , the switching line gradient, may be selected provided a sufficiently high gain is adopted, allowing any speed of response to be obtained.

However, if we introduce a saturation element in the control path, as shown in figure 2.16, we find that as we increase the gain, the control system approached a relay system of the form shown in figure 2.17.

Figure 2.16 Double integrator plant with control saturation

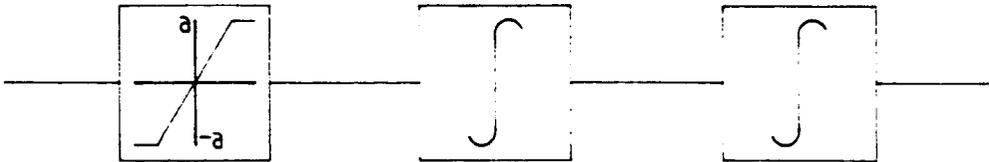
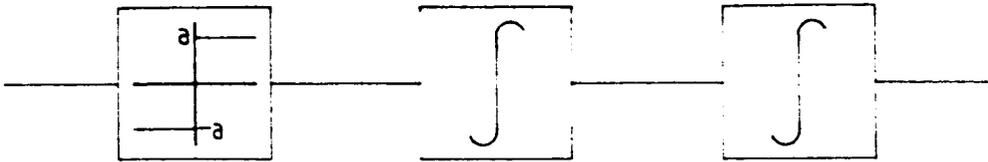
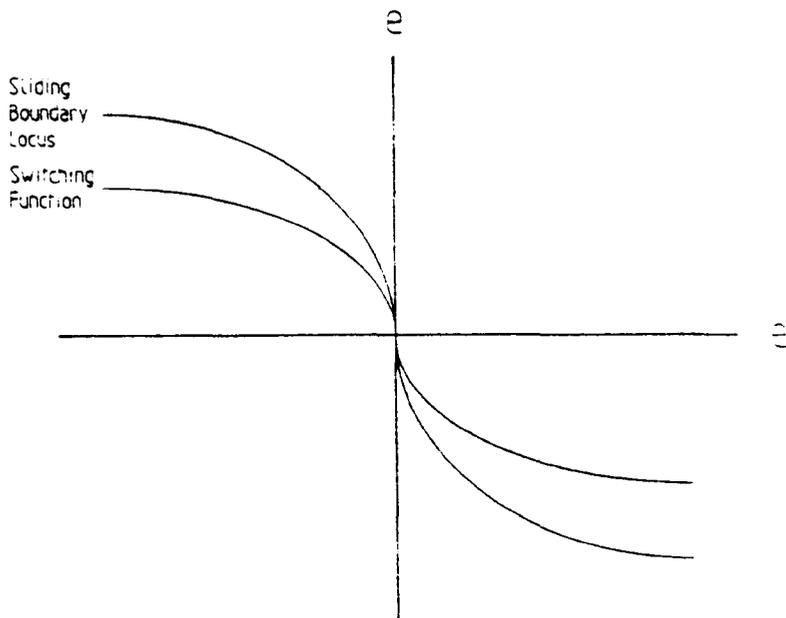


Figure 2.17 Equivalent relay control system for high gain



It is clear from a consideration of time optimal control [Ryan,81] that there is a boundary to the region in which sliding can occur, defined by the sliding boundary locus shown in figure 2.18 [Zinober,75].

Figure 2.18 Sliding boundary locus for relay system



For a VSC system with input saturation it is impossible to obtain sliding behaviour outside the region of sliding for the equivalent relay system. If a controller is designed without consideration of this limiting effect, a switching surface may be selected which passes outside the sliding region, leading to non-sliding behaviour and possible overshoot.

An example of this situation is given by Taylor [Taylor,86] for a double integrator plant with input saturation. A design procedure for the selection of piecewise linear switching curves which overcome this problem while giving near time optimal response for systems of this type with a fixed gain is also given. This algorithm was implemented for second order nonlinear systems by Jobling [Jobling,84] as part of the computer aided design suite described in chapter 5. Unfortunately this procedure cannot easily be extended to higher order systems owing to the difficulty in defining the effect of changing the switching function characteristics on the speed of response.

The switching line in a VSC cannot be such as would give a faster approach to the final position than that obtained by applying the maximum actuation signal to the plant, as in a time optimal control system, and it is this that limits the system performance. Conversely, the fastest response is obtained

from a VSC when the switching line is closest to the optimal control switching function, while still satisfying the sliding condition. This leads to the use of increased gains which, in the limit, become relay functions. An additional effect of this increase in gain is that the amplitude of the chattering behaviour increases. Not only is this undesirable because of the possibility of exciting unmodelled dynamics and causing increased wear, but also because the chatter may cause the RP to leave the sliding region as shown in section 2.1.4. Thus if the chatter can be reduced a switching function closer to the optimal may be selected. Since a large gain must be maintained to allow sliding close to the time optimal switching function but we wish to remove chatter as far as possible it is appropriate to employ the chatter reduction methods described in section 2.1.5. In this way the advantages of high gain and fast response are obtained without the possibility of the sliding mode breaking down due to chatter.

In the majority of practical systems there is some variation of parameters with time and due to external influences and as a result the time optimal switching line is also variable. If a VSC is designed to be near time optimal then the switching line must either be inside the region in which sliding occurs for the worst case plant parameter values, or the line itself must vary with the plant parameters to maintain a feasible sliding system. This has led to interest in the use of adaptive techniques to alter the switching line equation and other control parameters.

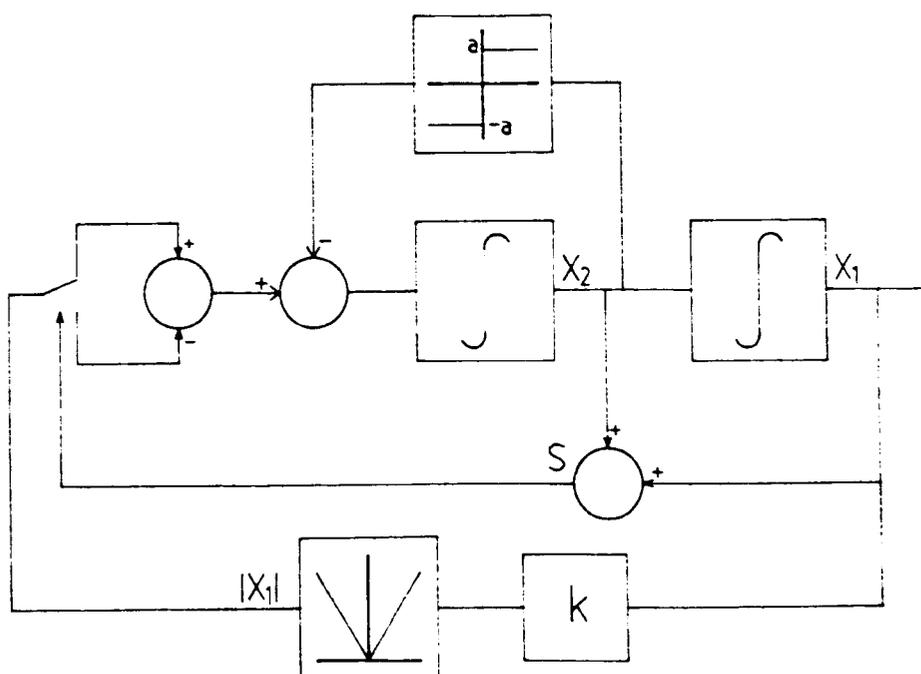
Parvinmehr [Parvinmehr,89] presents a method based on discrete time parameter estimation techniques to determine near optimal control parameters while Zinober [Zinober,80] adopts a scheme of increasing the switching line gradient until the sliding condition is just violated from which point the line is held constant. The latter technique may work in a limited number of cases but is not generally effective since, once sliding is lost, it will not recommence without a change in plant parameters or controller gain. A minor refinement of the algorithm would however make it widely applicable. Any control scheme

which attempts to obtain the time optimal response for a system will always be sensitive to parameter changes and disturbances which may take the RP outside of the sliding region. It is therefore necessary in practical systems, where these problems occur, to aim at a slightly sub-time-optimal response, the degree of sub-optimality dependant on the significance of parameter changes and disturbances.

2.1.6.2 Effect of Coulomb Friction on Variable Structure Control Systems

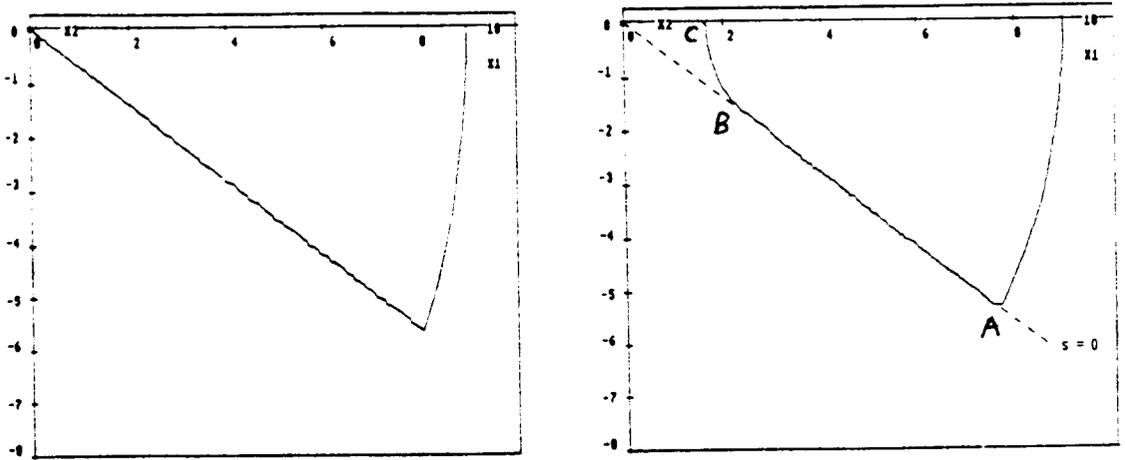
The majority of rotational and translational mechanical systems exhibit frictional behaviour which cannot be characterised as purely viscous. Although the true friction may be of a highly complex form, a coulombic element may generally be identified. This form of friction presents particular difficulties to the control engineer, owing to its discontinuous nature. VSC systems are no exception to this rule, although some improvements may be made over other forms of control. We will begin by considering the case of a simple VSC for a double integrator plant with coulomb friction, as shown in figure 2.19.

Figure 2.19 Double integrator plant with Coulomb friction and VSC



During the initial stages the phase plane trajectory for this system is similar to that for a system without friction, with the RP reaching the switching line and sliding behaviour commencing at point A of figure 2.20. As the trajectory approaches the origin the control input decreases with the error and a point is reached (B) at which this signal becomes less than the Coulomb friction and the trajectory leaves the switching line and moves to the $x_2 = 0$ axis a distance from the origin (point C), where it remains until a change in demand position or a disturbance occurs.

Figure 2.20 Effect of Coulomb friction on VSC response



(i) Zero Coulomb friction

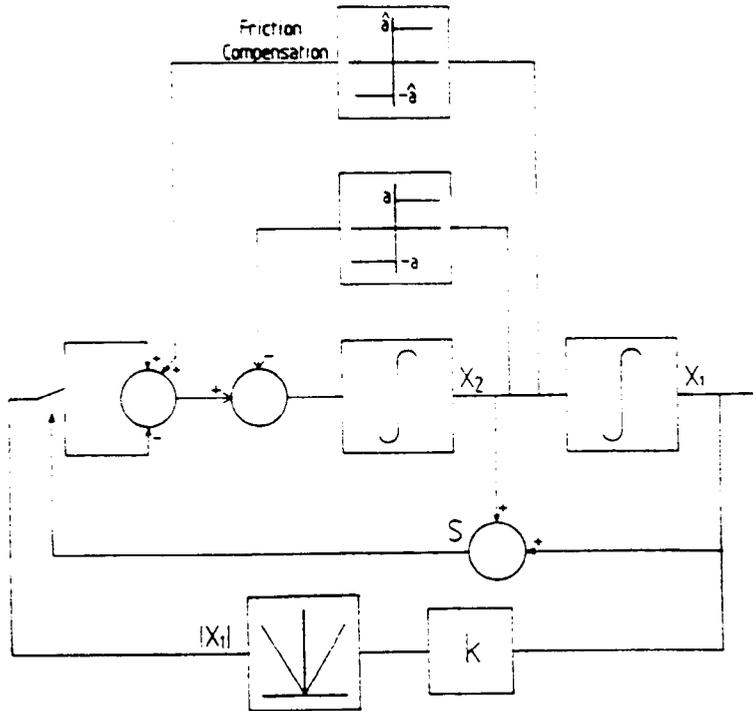
(ii) Non-zero Coulomb friction

The maximum value of this steady state error may easily be calculated and is found to be:

$$|x_{1,max}(t \rightarrow \infty)| = a/k$$

where a is the level of Coulomb friction and k the controller gain. Clearly the steady state error may be reduced by increasing the gain, k but this has the effect of increasing the chatter. It is possible to improve the situation to some extent by including a friction compensation term in the control system, as illustrated in figure 2.21.

Figure 2.21 Variable structure controller with friction compensation



The friction is estimated as $\hat{a}(x_2)$ which may include compensation for other effects, in addition to Coulomb friction. The maximum steady state error then becomes

$$|x_{1,\max}(t \rightarrow \infty)| = |a(0) - \hat{a}(0)|$$

The estimate of Coulomb friction must be a little conservative, being below the minimum expected value. The situation where the estimate is above the true value must be avoided, since this leads to what is effectively positive feedback, with a relay function in the feedback loop. Such a condition will lead to limit cycling about the origin, a highly undesirable situation. If on the other hand the estimate is below the actual value then a steady state error may occur but this is a preferable eventuality. Schemes have been proposed to determine the friction in an adaptive manner and this could prove an effective solution. Friction compensation algorithms has been suggested by Canudas [Canudas,86] and Kubo [Kubo,86] for DC motor and Robot control problems respectively.

An alternative technique which may be employed to overcome the effects of Coulomb friction is to use integral action in addition to the simple VS controller. A modification to Slotine's algorithm (see section 2.1.5.3) has been suggested by Carigan [Carigan,88] in which the switching line equation for a second order system is replaced by:

$$s = \dot{x} + \lambda x + \lambda^2 \int x dt$$

It is assumed in developing this controller that a tracking problem is being considered, so that the initial value of x is zero (or very nearly zero). This will prevent the integral action becoming excessively large during the reaching phase, a problem which might well occur if a large initial error were present.

A more suitable control when considering position control systems, also based on Slotine's algorithm, introduces integral action only when the RP is within the boundary layer around the switching line, which is defined as $s = cx_1 + x_2$. The control signal for the example given in section 2.1.5.3 then becomes:

$$u = -\hat{f}(X, t) - \sum_{p=1}^{n-1} \binom{n-1}{p} \lambda^p \bar{x}^{(n-p)} - k(X, t) \text{sat} \left[\frac{s}{\lambda^{n-1} \epsilon} \right] - k_i(X, t) \int_{t_0}^{\infty} s dt$$

where t_0 is the time the trajectory enters the boundary layer.

Following the procedure outlined in section 2.1.5.3 we may obtain the following expression for the dynamics of s within the boundary region.

$$\begin{aligned} \dot{s} = & -k(X_d, t) \left[\frac{s}{\lambda^{n-1} \epsilon} \right] - k_i(X_d, t) \int_{t_0}^t s dt \\ & + (\Delta f(X_d, t) + d(t) - x_d^{(n)}(t) + O(\epsilon)) \end{aligned}$$

or

$$s = \frac{p [\Delta f(p) + d(p) - x_d^{(n)}(p) + O(\epsilon)]}{p^2 + p \{k(p)/(\lambda^{n-1} \epsilon)\} + k_i(p)}$$

where p is the Laplace operator.

This is the equation of a linear second order filter with a zero at $p=0$ and an input made up of the modeling errors and disturbances. By suitable choice of $k(X_d,t)$ and $k_i(X_d,t)$ the dynamics of this filter may be made stable and, owing to the zero, any steady state disturbance or modeling inaccuracy will be rejected. Coulomb friction may be considered as a disturbance of this form and so this controller removes its detrimental effect from the controlled system.

This is an idealised situation in which the finite sampling rate and the discontinuous nature of the Coulomb friction are not considered and, although the use of integral action may reduce the steady state error, it may also have a destabilising effect, leading to limit cycling around the origin. This problem can be reduced by the use of a dither signal added to the control which effectively smoothes the discontinuity caused by Coulomb friction. This dither must be of such a frequency and amplitude so as to just overcome the Coulomb and static friction without introducing significant movement, typically of approximately the same amplitude as the Coulomb friction with a frequency a little above the cut-off frequency of the plant.

As explained at the end of section 2.1.5.2 the use of a smoothing function in a variable structure system with a boundary layer may result in a system which is equivalent to a simple PD controller. In a similar manner if it is assumed that the compensation term in Slotine's algorithm with added integral action is set to zero and a straight switching line is used then it is easily seen that, if constant parameters are assumed, the control equation becomes:

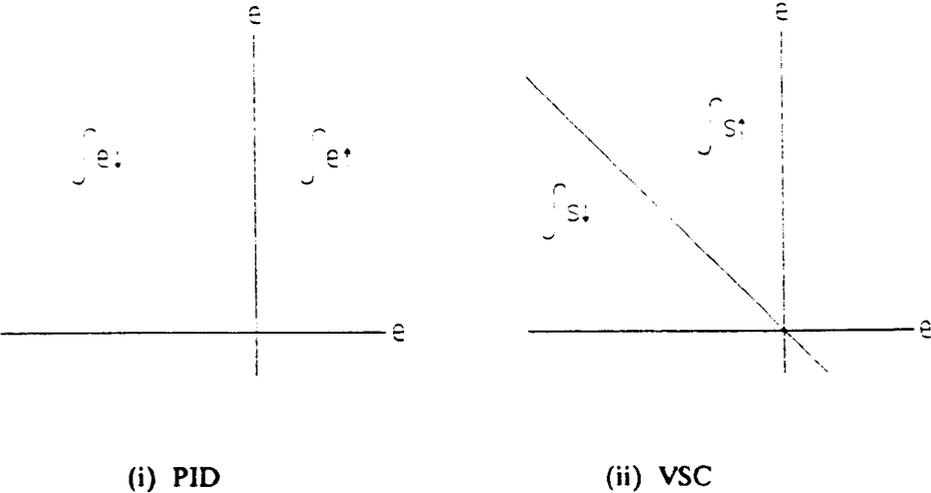
$$u = k_s + k_j \int s dt = k_c x_1 + k \frac{dx_1}{dt} + k_j c \int x_1 dt + k_i \int \frac{dx_1}{dt} dt$$

This may be compared with a PID control system, in which the actuation would be defined according to:

$$u = k_p x_1 + k_i \int x_1 dt + k_d \frac{dx_1}{dt}$$

There are many similarities between these two algorithms but the differences have a significant bearing on the system performance. In the PID algorithm the integral action is initiated at the beginning of a transient, or when a certain error is attained, and will not decrease again until x_1 changes sign, i.e. an overshoot has occurred. In the case of the VSC, since the sign of s may change, the integral of s may both increase and decrease. The behaviour of the integral component of PID and VS controllers in different regions of the phase plane are shown in figure 2.22 for a second order system. In the case of the VSC an overshoot need not occur before the overall integral action decreases. This is particularly important when nonlinear effects like Coulomb friction are present in a system and some integral action is required to remove steady state errors.

Figure 2.22 Region of integral action in VSC and PID control systems



It has been shown that variable structure control techniques provide a useful tool in the development of control systems for a wide variety of situations, particularly when a sliding mode is generated. In its most simple form a VSC system with sliding mode provides the control engineer with the possibility of imposing a desired form of dynamic behaviour on a wide variety

of systems which need not be accurately modelled. This dynamic performance is maintained in the presence varying parameters, nonlinear plant behaviour and external disturbances.

The most simple form of VSC involves a discontinuous change in control input which leads to chatter in practical systems which possess unmodelled dynamics and with controllers having finite time delays. This behaviour is undesirable as it leads to excessive control activity which can excite unmodelled dynamics and may lead to wear and fatigue in mechanical systems. For these reasons it is often required that chatter be removed or reduced in amplitude. This can be achieved by the use of compensation to reduce the height of the discontinuity or by smoothing to make the change continuous.

There are certain forms of nonlinear behaviour which do affect the performance of a VSC system and these must be taken into account in the controller design. Control signal saturation can cause a VSC system designed without consideration of this effect to leave the sliding mode, leading to loss of invariance to parameter variations and disturbances and giving undesirable behaviour such as overshoot. This problem may be largely overcome, without a major degradation of performance by the use of curved or piecewise linear switching functions which results in a response close to the time-optimal behaviour of the plant.

Coulomb friction in mechanical plant under the control of VSC systems can lead to a steady state position error. This may be overcome by the use of friction compensation and integral action. Using these control techniques it is possible to develop VSC systems which give good performance in many practical situations. In the following section the use of these controllers in the field of robotics will be considered.

2.2 Variable Structure Control of Robot Manipulators

Having developed the theory of variable structure control systems, we will now consider the application of these techniques to the challenging problem of robot control. This involves both position and tracking control of a system with highly nonlinear coupled dynamics with unknown and time varying parameters, in the presence of external disturbances.

The dynamics behaviour of robots in general, and the RTX in particular, is considered in detail in chapter 4 in which it is shown that there are two sources of nonlinear behaviour in a typical manipulator. These may be classed as those arising from interaction between the links of the mechanism and those found in the joint actuation systems. The relative significance of these two forms of nonlinearity have a bearing on the most appropriate control strategy. The interaction dynamics are of a complex form, depending on a large number of joint positions and velocities. These dynamics are also dependant on the load carried at the end effector, a variable which may or may not be available for inclusion in the control design. Actuation nonlinearities are, to a first approximation, functions of a single joint velocity and occasionally position; they are also approximately independent of load but are subject to variations in time due to wear and changing lubricant levels.

It is this nonlinear behaviour which gives rise to the difficulty in controlling robots to a high level of precision while attaining high speed operation. The use of variable structure control to overcome these problems has been considered for some time and a large amount of research work has been carried out in this field. The majority of this work has considered only the interaction dynamics of the robot with the actuation nonlinearities neglected altogether or represented as being of a very simple form. This is valid for certain types of robot where the interaction dynamics dominate but this is not always the case and so a major part of this work is involved in the application of VSC techniques to a system having significant actuation system nonlinearities.

The remainder of this section will consider the background to the use of VSCs in robot control problems.

One of the earliest applications of VSC theory to robot control was introduced by Young [Young,78] who used the control law $u = \alpha \text{sgn}(x,s)$ with a straight switching line defined by $s = cx_1 + x_2$. This was applied to a hybrid simulation of the dynamics of a two link planar manipulator, including the effects of inertia variation and coupling but neglecting centrifugal and coriolis forces and assuming a perfect actuation system. The results showed that the technique was applicable to this type of system and produced invariant dynamic behaviour in the presence of certain forms of nonlinearity and disturbances. It also showed the presence of chatter on the switching line due to the sampled nature of the control law implementation.

Slotine [Slotine,83] proposed the addition of a smoothing function in the VSC law to remove chatter and described the design of a controller for robot applications, although once again it was assumed that linear actuators with unlimited torque were available. This was developed [Slotine,84] to include a compensation term for general nonlinear systems rather than robot control specifically.

Up to this time little work had been carried out on the practical application of VSC techniques to real systems and so the problems of non-ideal actuator behaviour was not seriously addressed. Hirio [Hirio,84] introduces a VSC with velocity feedforward applied to a real robot in order to obtain accurate tracking of the computed trajectory. The use of a tracking controller eliminates the problems introduced by the reaching phase present in a position control. Chatter is removed by the use of high switching speeds and an adaptively altered gain in the region of the switching surface. The practical results presented are for a low speed movement but do indicate a significant improvement in performance, compared to that of a PID controller.

Morgan [Morgan,85] suggests the use of a disturbance measurement technique to compensate for external disturbances combined with a VSC to

control any effects not removed by the compensator. The VSC is necessary since the disturbance rejection is based on a greatly simplified version of the inverted manipulator dynamics and is thus subject to errors at certain joint angles and when changing load.

Staszulonek presented work on the implementation of a VSC on a single joint of a robot and compared the performance obtained with that of a PID controller [Staszulonek,86]. It is not clear if the results presented are those for the real system or a simulation but they do not show any effect from actuator nonlinearities, possibly due to the use of low friction bearings in the drive system. The problems of saturation in drive systems was considered for VSC systems, not specifically for robots, by Taylor [Taylor,86].

As an alternative to the interaction compensation technique used within the VSC controller proposed by Slotine and described in section 2.1.5.3, Chen [Chen,87] suggests the use of a decoupling control in a slow outer loop with a fast inner VSC used to assure that the trajectory follows that demanded by the outer loop. In this way the effects of modelling inaccuracy and disturbances on the decoupling control are removed by the VSC. In addition, the calculation of the decoupling control takes account of the limited torque available from the actuators and will not attempt to make a movement which would require excess power. The computation involved in this algorithm is extensive and so it has not been implemented. In simulation studies, the algorithm seems effective although a large amount of chatter is generated. In addition, no account was taken of Coulomb friction leading to possible steady state errors in a practical implementation.

Recently a number of minor modifications to the compensation and smoothing algorithm have been proposed for VS controller by Yeung [Yeung,88], Tenreiro-Machado [Tenreiro-Machado,88] and Pandian [Pandian,88] although few have been tested in practical implementation, generally owing to the computational expense of calculating compensation terms. The practical results which have been presented have all been for systems with relatively low

levels of Coulomb friction and although this may be justified in some cases by the design of robots with low friction drives it is not always the case. Particularly in low cost robots, a significant proportion of the motor torque is absorbed by friction in the joints and so this effect cannot be neglected. Work on the effect of Coulomb friction in systems under VS control systems has been presented by Taylor [Taylor,88], in which a development of Slotine's algorithm, using integral action and friction compensation was used.

It has been shown in this chapter how variable structure control systems may be developed for the control of robot manipulators with both nonlinear interaction dynamics and actuation system nonlinearities. The controllers described should provide fast response and accurate tracking in the presence of parameter variations and external disturbances. In addition some of the algorithms are computationally simple and may easily be implemented.

2.3 Selection of Control Parameters in VSC Systems

It is clear from the preceding sections that the design of a suitable VS control system involves the choice of a large number of parameters and so this section gives some details of the design procedure which may be adopted.

The first design consideration is the type of dynamic behaviour which it is hoped to impose upon the system. This will depend on the particular application and effectively defines the type of switching function adopted. In the case of a robot for instance, if the maximum point to point speed is required then a curved switching function close to the sliding boundary locus may be adopted. If however the maximum invariance to load changes is sought then a multi-section switching function of the type described in section 2.1.3.2 may be preferred at the cost of a loss of speed.

Once a suitable switching function has been selected, the controller gains required to ensure sliding must be found. These may be determined by following a similar procedure to that given in section 2.1.2.2. It may be found

at this point that no gains can be obtained which will guarantee sliding and so it may be necessary to alter the switching function definition.

Having reached this point it is probably wise to simulate the performance of the proposed system in order to determine the degree of chatter which occurs, with appropriate enhancements added to the controller to bring this to an acceptable level.

CHAPTER 3

Hardware Enhancements to the RTX Robot

In this chapter the robot used to test the implementation of control algorithms will be described, as will be the modifications made to the robot to improve measurement and actuation, required for both dynamic modelling and control. The manipulator used in this work is an RTX robot, manufactured by UMI. It is a six degree of freedom device of a SCARA configuration with all joints driven by DC permanent magnet motors. The motors are driven using a voltage mode, pulse width modulation scheme and, after gearing, transmit power to the links via toothed belts. Position information is derived from incremental optical encoders mounted on the motors, pulses from which are fed to a microprocessor, on which control algorithms are implemented. Further details of the original robot hardware are given in section 3.1.

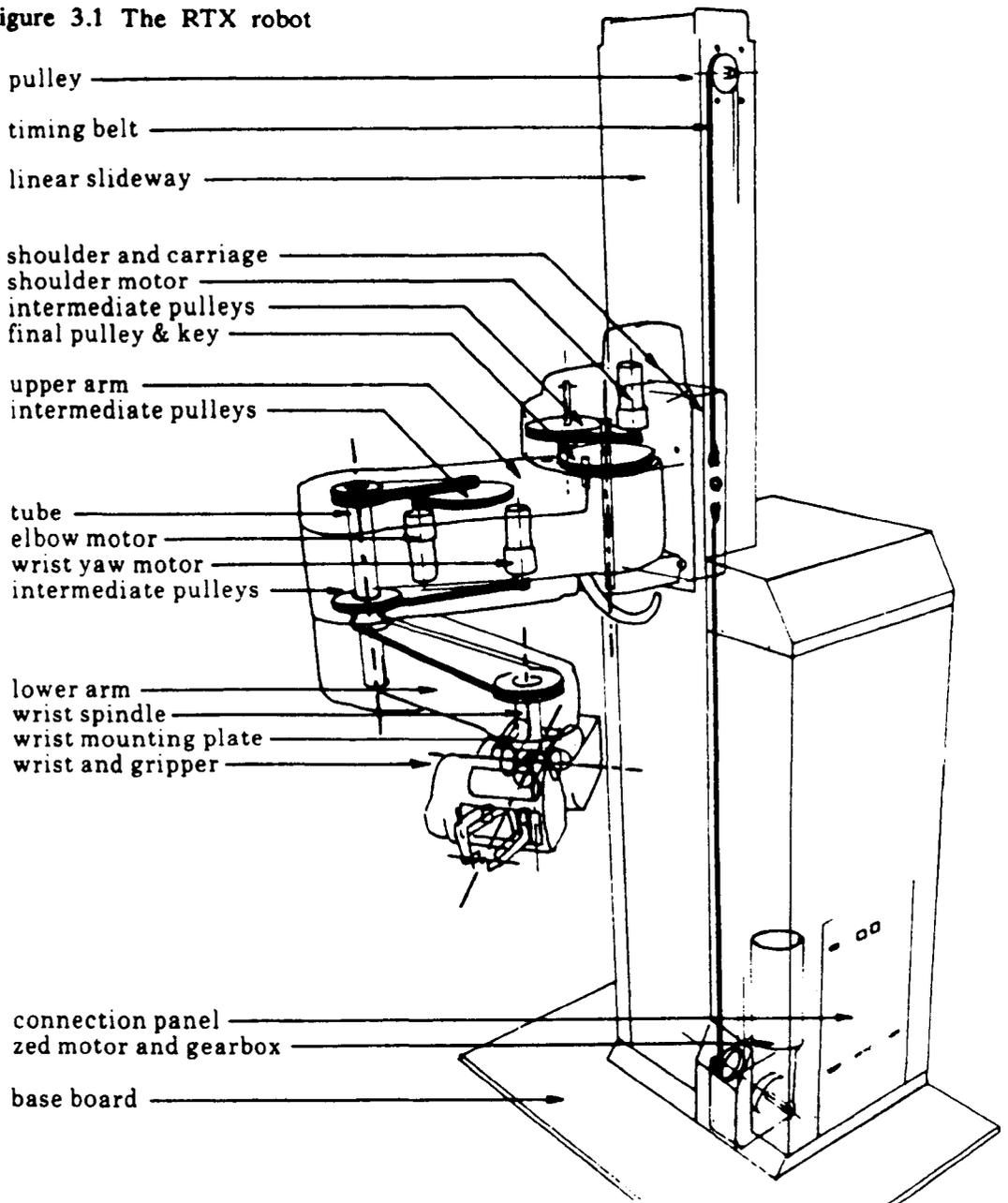
Control and modelling algorithms were developed using a DEC μ PDP computer and an IBM compatible personal computer. Position and velocity measurement circuits and the motor drive system were designed for use with either of these computers and the interfaces required are described in section 3.2.

It was decided that the existing motor drive circuits did not allow the motors to perform as well as possible and so a current mode, linear drive circuit was developed. This increased the maximum velocity of the motors by a factor of approximately two, without increasing the motor power dissipation at low speed. The design procedure employed in the development of this circuit is described in section 3.3. To allow accurate modelling and control of the robot a more accurate position feedback circuit was required. This took the form of high resolution incremental optical encoders which were connected to position and velocity measuring systems as described in section 3.4.

3.1 Existing RTX hardware

The RTX robot is a six degree of freedom manipulator with a Selective Compliance Assembly Robot Arm (SCARA) configuration, of the type first developed by Makino [Makino,80]. The first three links of such a manipulator are shown in figure 3.1.

Figure 3.1 The RTX robot

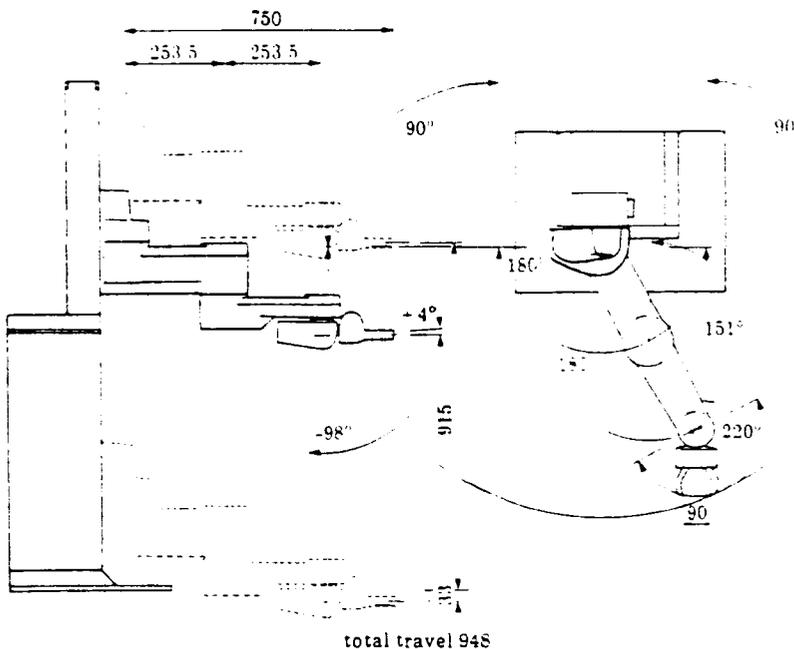


As the name implies, it is intended that the manipulator should have different compliance in different directions, hence a typical arm has low compliance in a vertical direction and relatively high compliance to horizontal

movements. This selective compliance is useful in assembly tasks, for instance when a peg must be inserted into a hole and other generically similar problems. In these circumstances a large downwards force is required while the 'peg' must be allowed some lateral movement to, in effect, find it's own way into the hole. Although selective compliance was the original reason for developing the SCARA configuration it is also found that the kinematics and dynamics of the links are simpler than some other link arrangements having a similar number of degrees of freedom.

The arrangement of the links in the RTX, along with the movement limits on the axes, is shown in figure 3.2. The robot has a vertical aluminium column, defined as being the Z axis, on which is mounted a sliding carriage supporting the rest of the arm. This carriage is driven by a motor mounted at base of the column, via a gearbox and toothed belt, and has a maximum displacement of 948mm.

Figure 3.2 Joint movement limits

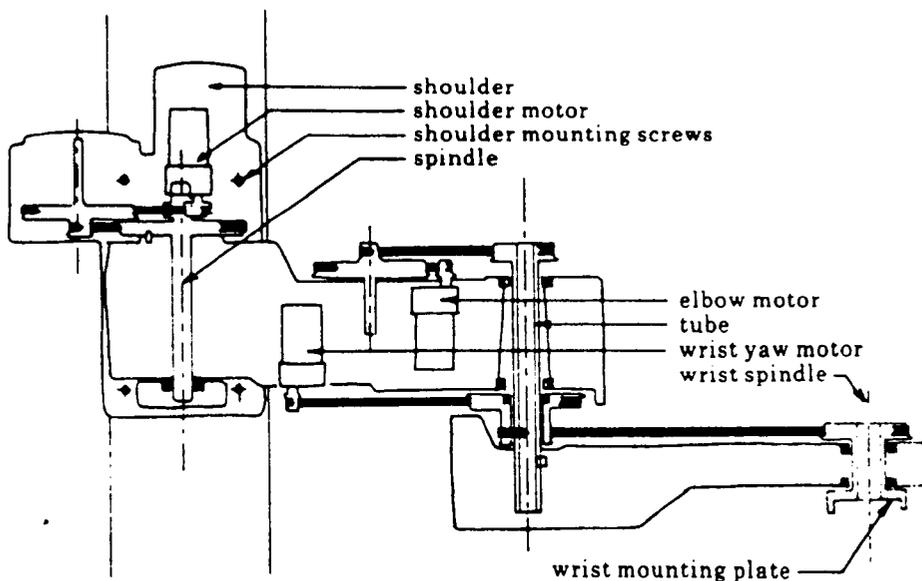


The carriage contains the shoulder motor, gearing and toothed belts to drive the upper link of the arm. The elbow motor is housed in the upper arm moulding and is connected to the elbow joints via a gearbox and toothed belts. The upper and lower arm links are polyurethane mouldings with steel mounting

plates for motors etc. and are both the same length, being either 253.5mm or 254mm depending on the production batch. The gearing of the elbow is arranged to have half the reduction ratio used on the shoulder joint. As a result of this, if the elbow and shoulder motors are run at the same speed the end point of the lower arm moves along a radial line from the shoulder axis.

The drive to the yaw joint passes from a motor mounted in the upper arm, through the elbow joint by means of a toothed belt and pulley mechanism. This is geared so as to maintain the same angle between the yaw joint and a radial line from the shoulder axis, when the yaw motor is stationary, independent of the elbow joint position. The yaw axis connects to a wrist mechanism containing two motors, the outputs of which are combined, with bevel gears, to give roll and pitch movements to the final link, on which is mounted a simple gripper mechanism, also driven by a geared motor. The distance from the yaw axis to the tip of the gripper, when horizontal, is 192mm, giving a maximum reach of 700mm from the shoulder axis. The complete arrangement of drive belts is shown in figure 3.3.

Figure 3.3 Timing belt transmission system

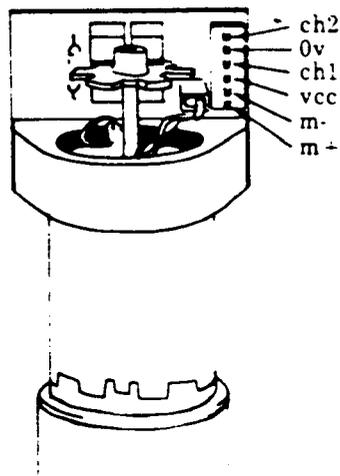


All the motors except that on the Z axis are of the same type, being DC permanent magnet brushed types with integral gearboxes. The reduction ratio in

this gearbox is 20.23:1 while subsequent gearing is used to provide the torques and speeds appropriate to the particular joints. The Z motor is larger than the others, having to support the weight of the rest of the arm, and has an integral gearbox with a reduction ratio of 25:1.

The joint motors are all fitted with low resolution optical encoder discs, having twelve slots per revolution for the Z motor and six slots for all others. The position is sensed using two slotted opto-interrupt switches fixed to the motor body, as shown in figure 3.4.

Figure 3.4 Low resolution motor-mounted encoder



The RTX is controlled using a circuit board containing two 8031 single chip microprocessors. These processors provide most of the required functions, including counting encoder pulses; calculation of motor demands using a simple PID algorithm and generation of pulse width modulation signals used to drive the motors. The processors are also connected, via a serial link to a control computer, generally a PC, which is used to generate joint positions and handle any information returned from the robot processors. The pulse width modulation signals generated by the processors are amplified to 24v and the maximum current increased to 3A using an L298 in the case of the Z motor and 750mA using an L293E for the other motors. These driver chips contain two sets of push-pull transistors in a bridge arrangement to give bi-directional movement.

They also provide current limiting at a level set using external resistors. Thus the mean voltage applied to the motor is set by the processor, independent of the current drawn, provided the current remains below the preset limit.

3.2 Computer Interfaces

The dynamic modelling and control of the robot may most efficiently be done using a computer to apply signals and monitor the response in the case of modelling while the reverse is required when controlling the manipulator. Two of the available types of computer were considered appropriate for this task. The first, a DEC μ PDP running the RT-11 operating system, is sufficiently fast to allow fairly complex algorithms to be run, while performing a large amount of monitoring and data storage. This computer is suitable for algorithm development but it is too expensive to be used on a number of robots and so a less expensive alternative was sought. The robot manufacturer's control system interfaces to a PC and so it was decided that it would be appropriate to use this for the new control algorithms developed. Although not as fast as the μ PDP it could be made to perform quickly enough and provided a solution at a significantly lower price.

In the algorithm development stage, analogue inputs and outputs were required for velocity measurement and motor demand output, in addition to digital input and output for position sensing. Once an appropriate algorithm was devised it was possible to implement it on the PC without analogue input functions, allowing a significant cost saving. The interface circuits used are described in sections 3.2.1 and 3.2.2 for the μ PDP and PC respectively.

3.2.1 μ PDP Interface Circuits

A wide range of interface circuits are available for the PDP range of computers, although they tend to be expensive [Digital,83]. The interface used

for digital I/O, known as the DRV11 gives 16 diode clamped data input lines and 16 latched output lines, all TTL compatible. When using the RT-11 operating system, read and write operations using this interface are straightforward, with a data transfer rate of 40K words per second achievable.

Analogue output is performed using an AAV11-C digital to analogue converter. This provides four channels, each with a resolution of 12 bits, giving an output voltage range selectable between $\pm 10V$ in bipolar mode and 0 to +10V in unipolar mode. Each converter is located at a separate address allowing simple data write operations using either binary for unipolar output or offset binary for bipolar mode.

The analogue input interface selected was an ADV11-C circuit. This gives 16 single ended channels with a resolution of 12 bits and an input range of $\pm 10V$. Software selectable gains of 1,2,4 or 8 are also provided at the input, allowing a wide range of input voltages. The input channels are fed to an analogue multiplexer, the selected channel then being fed to a sample and hold circuit via the switchable gain network. The output of the sample and hold provides the input to the 12 bit A/D converter with the conversion started from a program command, an external event or a realtime clock input. The end of conversion may be used as an interrupt or may be polled, as appropriate. The conversion time of $25\mu S$ allows a system throughput of 25K samples per second.

Using these interfaces all the analogue and digital I/O functions required may be achieved using simple program commands with a very small amount of processor time overhead.

3.2.2 Personal Computer Interfaces

The PC was used in two applications, firstly in the conversion and testing of algorithms developed on the μPDP and secondly in the implementation of algorithms for use on a number of robots. The requirements for these

applications were somewhat different and so two types of interface were used. The first, for conversion and testing work required analogue and digital input and output while the later required only digital I/O and analogue output. By using a purely digital I/O board, with an external digital to analogue converter, in the later instance a significant cost saving was made over the more flexible interface board, an important consideration as this system was to be duplicated many times.

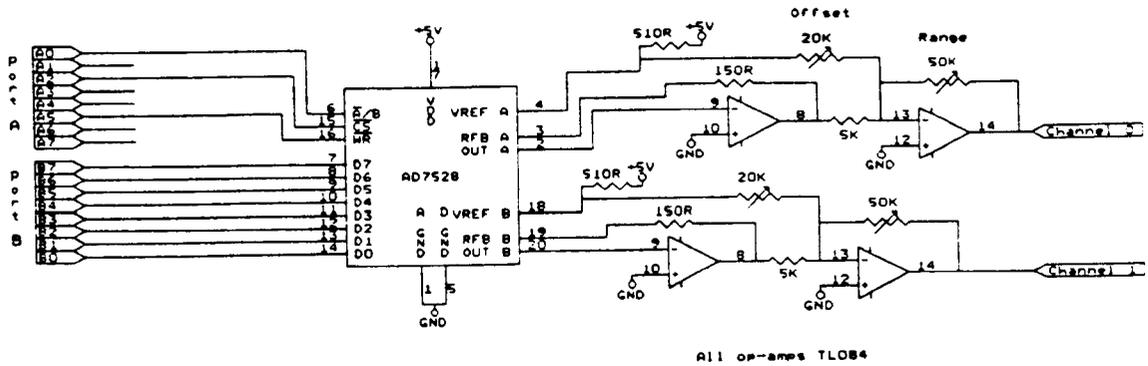
The first interface board, a PC-30A, supplied by Amplicon, provides 24 programmable digital I/O lines, a 16 channel, 12 bit A/D converter, two 12 bit and two 8 bit D/A converters and three programmable timers. These interfaces may be configured to be compatible with the circuits developed for use with the μ PDP and give similar performance, although programming is a little more complex.

The measurement and actuation circuits used for the final implementation of the controller interfaced to a PC-14A board, once again supplied by Amplicon but at about a fifth of the cost. This board provides 48 programmable digital I/O lines, in two groups of 24, and three programmable timers. In practice it was decided that only 24 of the 48 I/O lines would be used for robot control with the remainder available for sensor and control data for other equipment. This also had the advantage of saving a significant cost in cable for connecting the robot and the computer. Both the PC-30A and the PC-14A have selectable base addresses so that the locations of other components may be avoided. The timers on both interfaces may be driven from either an external signal or the system clock and can be used to interrupt the processor, allowing accurate setting of the controller sample rate.

The motor demand signal is obtained from an analogue voltage and so a digital to analogue converter must be connected to the PC-14A interface for this purpose. The circuit devised is based around the AD7528 dual 8 bit converter. It was verified experimentally that the reduction of demand resolution from 12 to 8 bits did not adversely affect the control performance. This was as

expected, given that control system performance is not generally very sensitive to drive system gain. The choice of this device, in particular the need for only an 8 bit digital output, allowed the use of few external components as seen in the simple circuit shown in figure 3.5. This circuit provides two channels, being duplicated for applications requiring more outputs, as seen in section 3.3 where the D/A converter and motor drive circuits are combined.

Figure 3.5 Dual channel D/A converter for demand output



3.3 Motor Drive Circuits

Initially, control of only the shoulder, elbow and yaw joints of the RTX was considered and so the drive system described here was developed for this motor, although it has since been found that only minor alterations are required to make it suitable for the Z-motor.

The choice of motor drive system is very important, if optimal performance is to be obtained from a manipulator system. When DC permanent magnet motors are used the choice of drive circuit affects both the maximum torque and speed of the motor, as well as the motor time constant and power dissipation. The available methods may broadly be divided into linear or pulsed mode, referring to whether the power is applied continuously at the required

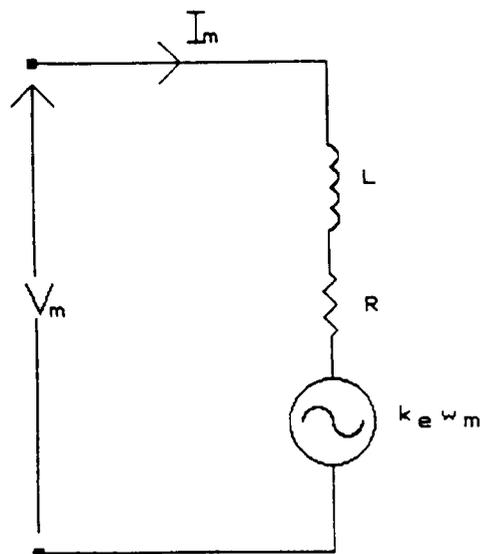
level or in pulses which give an average of the correct value. The second major division is between voltage and current mode, depending on which variable is set by the input signal. Each of these divisions has certain advantages and these are discussed in the following section. Once the drive method has been chosen, various parameters must be selected and the design procedure used to find these is described in section 3.2.2.

3.3.1 Selection of Drive Method

In this section the relative merits of voltage and current mode drives will be considered, followed by a discussion of linear and pulsed drives and finally the drive method selected will be described. Before beginning however, the properties of the DC motor which are pertinent in this context are discussed.

It is shown in section 4.1.1 that the electrical components of a DC permanent magnet motor may be modelled as shown in figure 3.6, where R and L are the winding resistance and inductance respectively; ω_m is the motor velocity and k_e is the back e.m.f. constant while V_m and I_m are the motor voltage and current respectively.

Figure 3.6 Equivalent circuit of a DC permanent magnet motor



The torque produced by the motor is $k_m I_m$ and we may write the motor current in terms of the Laplace operator, s as:

$$I_m(s) = \frac{V_m(s) - k_e \omega_m(s)}{sL + R}$$

A voltage drive system is relatively simple to design requiring that, for demand voltage, V_{in} the motor voltage should be $V_m = k_v V_{in}$ where k_v is the voltage gain of the drive amplifier. For such a system, when the motor is stationary the current will rise exponentially, with a time constant of L/R , towards a maximum value of $k_v V_{in}/R$. Thus in order to limit the maximum current to I_{max} , and hence limit the power dissipation in the motor the motor voltage must be limited to a value of $V_{max} = I_{max}R$.

Once the motor begins to move, a back e.m.f. is generated which opposes the drive voltage. Thus, for a steady velocity of ω_m the current will reach a value of $I_m = (V_m - k_e \omega_m)/R$. As the velocity increases the current and hence the torque are reduced until a point is reached where the torque becomes equal to zero, if friction is neglected, and hence no further increase in speed is possible. This limit occurs at a velocity given by $\omega_{max} = V_{max}/k_e$. If friction is included, a similar limit is reached but at a lower velocity. Thus, for a voltage drive, a high low-speed torque is obtained but the attainable velocity is limited.

When a current drive system is used, it is the motor current rather than voltage which is controlled. Thus for input voltage V_{in} the motor current is given by $I_m = k_i V_{in}$, where k_i is the current gain. Taking the condition when the motor is stationary the drive amplifier will attempt to force a current of $V_{in}k_i$ through the motor immediately. Due to the winding inductance however, this would require an infinite motor voltage. If the drive amplifier has a supply voltage of V_s then, neglecting the small voltage drop in the output stage, the motor voltage will equal V_s and the current will increase exponentially according to $I_m = V_s/R(1 - e^{-(R/L)t})$ until the current approaches that required, from which point the motor voltage reduces to $V_m = V_{in}k_iR$. It will be seen

that this does not impose any limit on the maximum drive voltage as was the case when using a voltage drive. It should also be noted that the initial rate of increase in current and hence torque is higher for the current drive than for the voltage drive by a factor of $V_s/V_{in}k_v$.

As the motor velocity increases the drive voltage will also increase to maintain a current of $k_i V_{in}$ according to $V_m \approx k_e \omega_m + k_i V_{in} R$ until the required voltage reaches the supply voltage. The same problem then occurs as is found in the voltage drive, with the maximum motor velocity being given by $\omega_{max} = V_s/k_e$. In this case however, because the supply voltage is not limited by the low speed power dissipation, this limit may be at a significantly higher level. In fact the supply may theoretically be set at as high a level as desired, without damaging the motor at low speed and allowing any maximum velocity required. There are two main limiting factors on the supply voltage that may be used for the current drive amplifier. Firstly, the power dissipation in the drive is given by $(V_s - V_m)I_m$, with the worst case which occurs for any length of time being $V_s I_{max}$. As the supply voltage is increased, the maximum amplifier power dissipation also increases, requiring higher power and more expensive driving circuits. The second limit on the supply voltage is caused by the fact that the motor commutator becomes more prone to arcing as the drive voltage increases. This reduces the life of the motor and generates electro-magnetic interference. There is also a limit on the speed at which motor bearings may be operated and so any improvements in speed obtained by changing the drive system must be limited.

The second major consideration in the design of a motor drive system is the choice of either linear or pulsed mode drive. In a linear drive, the output voltage or current is proportional to the input demand and remains approximately constant while the input remains fixed. Thus the output is given by:

$$I_m = k_i V_{in} \quad \text{or} \quad V_m = k_v V_{in}$$

for current and voltage drives respectively, where k_i and k_v are the corresponding current and voltage gains.

There are many forms of pulsed mode drive, although the most common form uses pulse width modulation. In this form the drive signal is switched at a fixed frequency, between two levels in such a way that the mean value is proportional to the input demand. In a typical motor drive the output is switched between zero and full power, the duration of the high pulse being proportional to the input demand. Since the drive signal is either fully on or fully off the power dissipation in the drive is minimised. The design of a pulsed mode voltage drive is a straightforward requiring simple timing circuits and power switching devices. Pulsed current drives are rather more difficult to design since measurement of the mean motor current involves some form of filtering which in turn causes stability problems in the current feedback mechanism. Owing to the high speed switching used in pulsed drives they tend to be electrically and acoustically noisy, a problem which is not found with linear drives.

The advantages and disadvantages of the various drive methods described in the preceding paragraphs are summarised in table 3.1.

Having considered the relative merits of the various techniques, in general terms, it remains to describe the particular method chosen, with regard to the particular requirements in this situation. The motors used in the axes of the RTX robot considered here are fairly low power, being rated at 18 watts, with a maximum current of 750mA. Ideally we would like a high maximum speed and a constant, high maximum torque over the speed range.

With these factors in mind, the drive selected was a linear current mode system. The choice of a current drive fulfills the requirements for high speed and torque and, since the motor power level is relatively low, the power dissipation in the drive is not important. Apart from the difficulty of design, a high performance pulsed mode current drive was rejected because of the high level of electrical noise generated which, owing to the close proximity of the

motor and the position sensors, would cause interference on the position measurement system. This type of drive would not give any significant performance advantages over that selected.

Table 3.1 Comparison of motor drive methods

Method	Advantages	Disadvantages
Linear, current mode	High torque at all speeds High maximum speed	High power dissipation at low speed Fairly difficult design
Linear, voltage mode	High low-speed torque Fairly simple design Moderate power dissipation	Low maximum speed
Pulsed, current mode	High torque at all speeds Moderate maximum speed Low power dissipation	Difficult design Noisy
Pulsed, voltage mode	High low-speed torque Low power dissipation Simple design	Low maximum speed Noisy

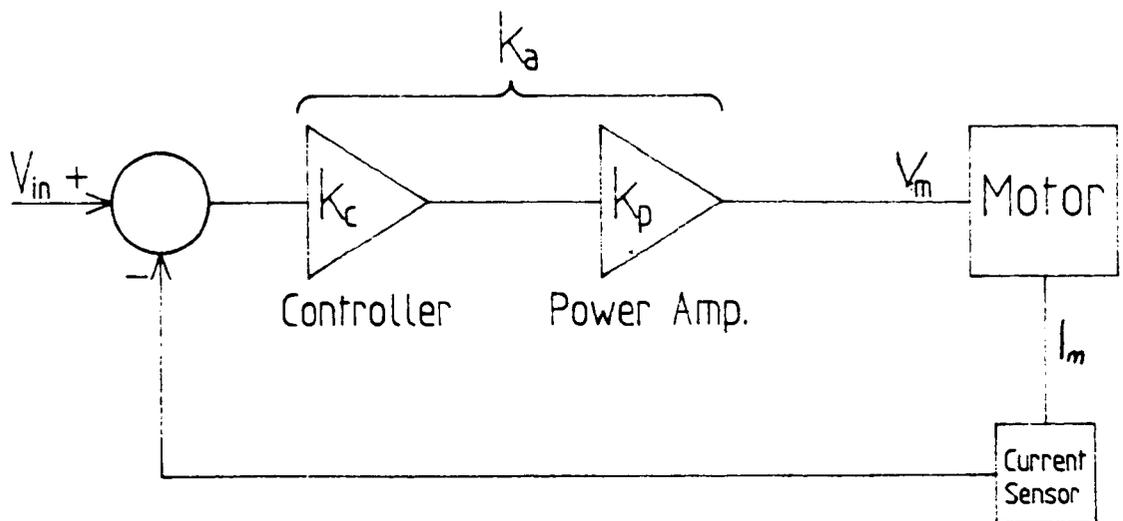
3.3.2 Design of Linear Current Drive

Having decided that a linear, current mode drive is appropriate in this situation the task of designing such a system must be undertaken. The ideal current drive system may be viewed as a control system which forces the desired current through the motor, with a rapid response to changes in demand and independent of disturbing factors, such as the back e.m.f. caused by motor movement. A number of control systems were investigated based on the dynamic model of the motor using root locus and time response information generated with the Hull CACSD suite described in chapter 5. The controllers giving the best performance in simulation were complex and when implemented

in hardware it was found that the behaviour did not agree with that predicted, due to unmodeled amplifier and motor dynamics. In general, these controllers became unstable, giving high frequency oscillation. By a combination of simulation and experimental work, it was found that a simple proportional controller could be made to perform well. The analysis of the system selected will now be given.

A block diagram of the major components of the system is shown in figure 3.7.

Figure 3.7 Schematic diagram of current drive system



The motor current is measured using a simple resistor and the difference between this value and the demand signal is amplified in the two gain stages and fed back to the motor. The requirements for the power amplifier were for a maximum supply voltage of at least 35V and a power dissipation greater than 20 watts. Although such a circuit would not be difficult to design it was more economic to use a commercially available circuit. The unit selected was designed as an audio amplifier but if the decoupling capacitors are removed, a very cost effective DC power amplifier is obtained. The circuit is of a simple design, using two power MOSFETs in a push-pull formation as the power devices, allowing a maximum power dissipation of 75W. The amplifier may be used with a supply voltage of up to 50v and provides a voltage gain of 33 up

to a frequency of about 40kHz. Full details of the circuit are given in appendix 1. A supply voltage of $\pm 40V$ was selected, allowing a high maximum velocity without excessive power dissipation in the drive circuit.

The motor current may be written as:

$$I_m = \frac{V_m - k_e \omega_m(s)}{sL + R_m + R_s}$$

or

$$I_m = \frac{k_{me} (V_m - k_e \omega_m(s))}{s\tau_{me} + 1}$$

where

$$k_{me} = \frac{1}{R_m + R_s} \quad \text{and} \quad \tau_{me} = \frac{L}{R_m + R_s}$$

where R_s is the current sensing resistance. The motor voltage, V_m is:

$$V_m = k_a (V_{in} - R_s I_m)$$

where k_a is the combined gain of the controller and the power amplifier, $k_a = k_c k_p$. The motor current, in terms of the demand voltage, is given by:

$$I_m = \frac{V_{in} k_a k_{me}}{s\tau_{me} + 1 + k_a k_{me} R_s} - \frac{k_e k_{me} \omega_m(s)}{s\tau_{me} + 1 + k_a k_{me} R_s} \quad (3.1)$$

We may rewrite the first term of this equation as:

$$\frac{V_{in} k_d}{s\tau_d + 1}$$

where the steady state gain, k_d is:

$$k_d = \frac{k_a}{R_m + R_s + k_a R_s}$$

while the time constant is:

$$\tau_d = \frac{L}{R_m + R_s + k_a R_s}$$

Considering k_d , we see that, as the gain, k_a is increased, the steady state current approaches V_{in}/R_s . This is the type of behaviour required and, by suitable choice of R_s and scaling of V_{in} the required current range may be obtained. In a similar manner, if we wish to reduce the time constant, τ_d we may increase the gain k_a .

Turning our attention to the second term in equation 3.1 we see that this is effectively a disturbance to the required behaviour caused by the back e.m.f. generated in the motor. This reduces the current in the motor by an amount proportional to the motor speed. It will be seen that the effect of this voltage may be reduced by increasing the gain k_a .

It should be noted that although the above benefits may be obtained by increasing the value of the current sensing resistor, R_s this also increases the power dissipation in this resistor and so any such increases should be limited, it being preferable to increase the gain, k_a .

It has been assumed so far that the dynamics of the motor windings are the only factor affecting performance but this is not in reality the case. In order to prevent high frequency instability excited by noise generated in the motor commutator, it was found experimentally that a pole must be introduced in the power amplifier response at a frequency of about 2kHz. This is achieved by adding a capacitor in parallel with a feedback resistor, within the power amplifier circuit, as detailed in appendix 1. With this addition the transfer function of the power amplifier is changed from the simple gain, k_a to:

$$\frac{k_a}{s\tau_a + 1}$$

where the amplifier time constant, $\tau_a \approx 80\mu S$. Neglecting the effect of the motor back e.m.f. we find that the open loop transfer function of the drive and motor is:

$$G(s) = \frac{I_m(s)}{V_{in}(s)} = \frac{k_a k_{me}}{(s\tau_a + 1)(s\tau_{me} + 1)}$$

From the motor manufacturer's data [Bühler], the constants in the above equations are found to be $\tau_{me} = 0.415\text{mS}$ and $k_{me} = 0.046\text{A/V}$. A number of experiments were carried out on several motors and it was found that more accurate values would be $\tau_{me} = 0.57\text{mS} \pm 0.07\text{mS}$ and $k_{me} = 0.044\text{A/V} \pm 0.003\text{A/V}$ and it was these figures which were used in the computer aided control system design.

When the control loop is closed around this system, the response is typical of a second order system with output feedback, the damping being dependant on the loop gain. The choice of controller gain, k_c , is a compromise between increasing the amount of overshoot to a step change in demand and making the motor current more dependant on the motor speed. The value of gain, selected through simulation and experimental studies, was relatively high at 30V/V . This gives a second order response with angular frequency of $\omega_0 = 31.2\text{krads/s}$ and damping factor $\xi = 0.23$. This degree of damping gives a theoretical peak current overshoot of 47.6% for a step input, compared with a measured value of $50\% \pm 5\%$, while the measured angular frequency was $31.4 \pm 2 \text{krads/s}$. This amount of overshoot is greater than would be hoped for but may only be reduced by decreasing the gain, causing the controller to become more sensitive to the back e.m.f generated in the motor. With this gain the motor current drops by approximately 40mA at the maximum motor speed before the supply voltage of the power amplifier is reached, independant of the demand current.

Using this controller the current settles to within 5% of the final value, for a step input, within 0.45mS. This is much shorter than the mechanical time constant of the motor and the typical sampling interval used in a position control algorithm. It was therefore decided that the demand signal going to the current drive could be filtered using a single pole, with a time constant of 1mS. This has the effect of removing all overshoot from the current response to a step input and increasing the rise time of the current to 1.75mS, without affecting the sensitivity to motor speed. The circuit of the complete current drive circuit is shown in figure 3.8 while the block diagram representation of

the system dynamics is given in figure 3.9.

Figure 3.8 Circuit of current drive system with input filter

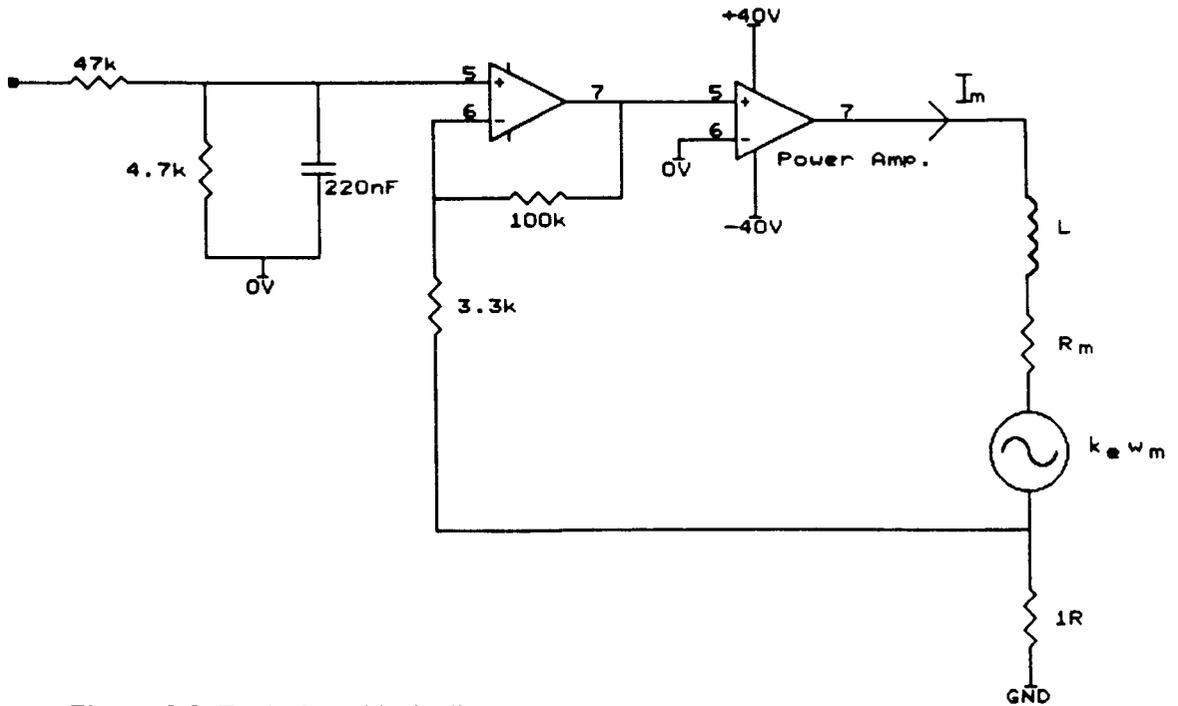
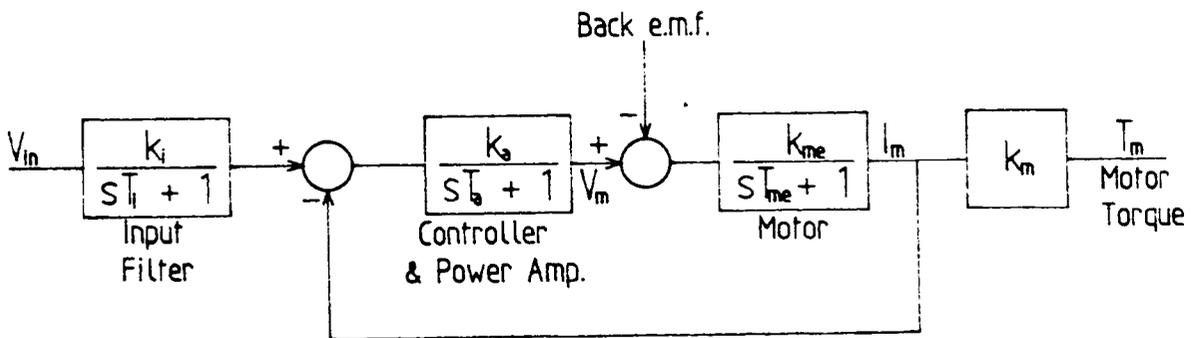


Figure 3.9 Equivalent block diagram



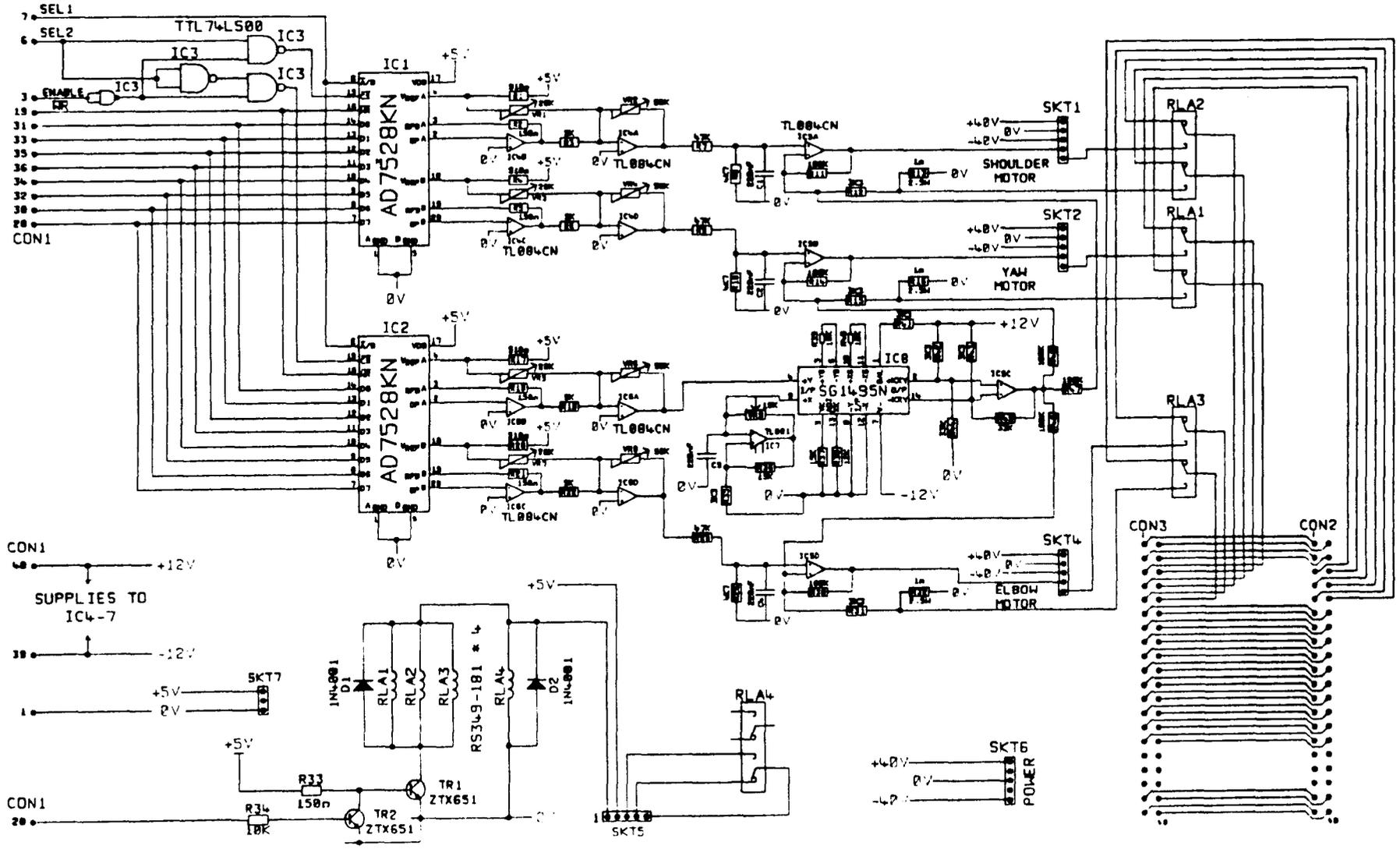
The drive system may be connected to the motor by either external wiring or via the existing wiring from the manufacturer's control board. In order that the original control should still be available it was required that the desired motor drive signal be selectable. This was achieved by means of either a two pole changeover switch, operated by the user, or a two pole changeover relay, controlled from software.

The circuit described above may be interfaced to the μ PDP via the AAV-11C D/A converter, or with the PC using the PC-30A interface. However, when the PC is used with the PC-14A I/O circuit an external D/A converter must be used, as described in section 3.2.2. The circuit developed for this purpose is shown in figure 3.10. This is based around the AD7528 dual 8 bit D/A with appropriate chip select logic that allows the use of two chips on a common 8 bit bus. The circuit also shows the arrangement of relays used to select between the manufacturer's control circuits and those newly developed. These relays are also used in the emergency stop circuit which operates from the original emergency stop switch. When this switch is opened, the motors are connected to the manufacturer's control circuit and the appropriate signal applied to the manufacturer's emergency stop logic. The circuit of figure 3.10 may be connected to ports A and B of the PC-14A interface. Port B gives the output demand data, while port A provides control functions. The individual bits of these ports are defined as in table 3.2.

Table 3.2 Bit definitions for motor demand output

Port	Bit Number	Function
A	0 - 1	Channel Select
	2	Not Used
	3	Position / Demand Select High for posn. I/O, Low for demand o/p
	4,5	Not Used
	6	$\overline{\text{WRITE}}$ Low to write demand data
	7	Not Used
B	0 - 7	Demand data output

Figure 3.10 Multiple axis motor drive circuit



When developing controllers, it was found to be advantageous to add a dither signal to the motor drive current when moving the arm at low speeds. This signal overcomes the static friction and allows smoother movement at low speed than would otherwise be possible. To allow the addition of a dither signal, the circuit based around IC7 and IC8 was developed. IC7 is configured as an astable multivibrator which would normally be used to generate a square wave. By taking the voltage across the capacitor rather than the op-amp output, an exponentially rising and falling voltage is obtained. By selecting the positive feedback resistances, R35 and R36 to give a low threshold voltage, a good approximation to a triangle wave is generated using a small number of components. The amplitude of the dither signal is controlled from the fourth D/A converter, which was not required for motor control. The output voltage of this D/A provides one input of an analogue multiplier, with the other input driven from the triangle wave generated from IC7. The output of the multiplier is buffered using IC5C then added to the control signals fed to the three motor drives. In this way the dither amplitude fed to the axes is controlled from software while the frequency is controlled using VR9. It was found experimentally that the smoothest joint movements were obtained using a dither frequency of approximately 70Hz and an amplitude of 200mA, corresponding to the typical friction level found, as detailed in chapter 4.

The drive circuits described in this section allow an increase in the maximum motor velocity, compared to the manufacturer's control, by a factor of approximately two, without increasing the power dissipation in the motor. In addition the torque generated is smoother and more controllable, without the large amount of electrical noise produced by the manufacturer's drive system.

3.4 Position and Velocity Measurement

The motors of the RTX robot are fitted with low resolution incremental optical encoders, having six slots per revolution, sensed with slotted

opto-interrupt switches with integral amplifier and Schmitt trigger circuits. These give two quadrature square wave from which the relative position may be derived. By counting two edges of one of these signals, the internal controller gives a typical range of 4000 counts over the full range of joint movement. The initial position of the robot is found by driving each link into a predefined end stop.

This arrangement allows the motor position to be found with an accuracy of plus or minus half a slot, corresponding to an end point position error of up to 0.5mm. This is of a similar order to the errors in position resulting from mechanical imprecision caused by drive belt flexibility and backlash. For the eventual controller significantly higher resolution motor position measurement would not give a corresponding improvement in end effector accuracy due to these mechanical limitations. For modelling purposes however a more accurate measurement of motor position was required. This was achieved through the use of high resolution incremental optical encoders as detailed later in this section.

The only way to significantly improve the end point accuracy without making major changes to the mechanical system would be to mount high resolution position sensors on the joints themselves. The cost of fitting such sensors to all robots would be prohibitive and was considered unnecessary since for many of the operations required a high precision was not required. High resolution incremental optical encoders were however fitted to the shoulder and elbow joints of one robot used for modelling and control algorithm development. Encoders with 10800 and 3600 lines per revolution were used, giving a resolution of 0.0083° and 0.025° for the shoulder and elbow joints respectively. These values are approximately four times the corresponding resolutions obtained using the original, motor-mounted encoders.

For a high performance controller to operate satisfactorily it is essential that an accurate velocity measurement be obtainable. With the original motor drive circuits the maximum motor speed results in an encoder output frequency of approximately 1200 counts/s. Thus, for a controller sampling at 60Hz a

resolution of only 5% of full scale is obtained. As the required speed is reduced below about 20% of the maximum, accurate control becomes virtually impossible. Thus it was decided that a more accurate velocity measurement was essential.

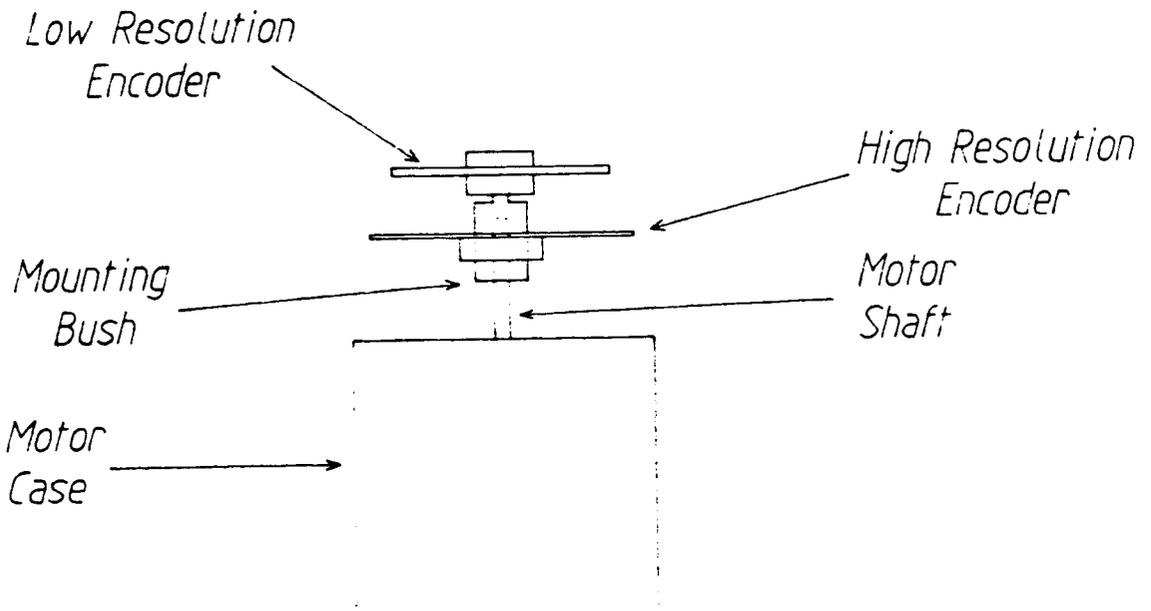
This measurement may most easily be obtained using either a high resolution encoder or tachogenerator mounted on the motor. If an optical encoder is used then the motor speed may be obtained using a frequency to voltage converter driving the input of an analogue to digital converter, connected to the control computer. Alternatively, by tracking the motor position using a suitable counter circuit, the velocity may be obtained by taking the change in position over a known time interval. When a tachogenerator is used it is a simple matter of connecting the output voltage to an analogue to digital converter.

It was found that a tachogenerator would be more expensive than an optical encoder and would add significantly to the motor inertia. It was therefore decided that an optical encoder should be used, particularly as this also gave the possibility of accurate position measurement mentioned above. There are many types of optical encoders available from a number of sources. The particular requirements in this application were that the encoder should not add significantly to the friction or inertia of the motor, but would give a resolution of at least 50 lines per revolution. It was also hoped that the encoder should have reasonably low cost, since many would be required. The most appropriate unit found was supplied by Hewlett Packard and consists of a metal disc, with 500 lines per revolution, and a separate sensor unit with integral signal processing, giving two quadrature, TTL level outputs. This device, which is described in detail in appendix 2, was found to be easy to use both in terms of mechanical alignment and electronic processing.

In order that the manufacturer's control system should still be available, when required, it was decided that the original encoders should be maintained. This was achieved by mounting the high resolution encoders on the rotor shaft

with the original encoder fitted to an extension to the shaft, as shown in figure 3.11. In this way the effect of any eccentricity in the shaft would not be accentuated at the high resolution encoder by the use of a long extension to the shaft. The encoder sensors were mounted on a printed circuit board fixed to the motor body using the collar supplied by the manufacturers, in a similar manner to the original mounting.

Figure 3.11 High and low resolution encoder mounting



The signals derived from the encoders were fed back from the arm via external wiring in the earlier implementation, although in the final version a number of spare wires, fitted by the manufacturer along with the control and sensor wiring, were employed, giving a tidy solution without the possibility of tangling wires and avoiding major mechanical disassembly. It was found that the optical encoder sensors could not be used with cable over about 1m long and so TTL buffers were connected close to the sensors to provide sufficient current to drive the longer cable required to reach the counter circuit.

Having derived the encoder signals it remains to obtain the position and velocity values. Firstly, the position monitoring system will be discussed, followed by the velocity measurement techniques. In the position measurement scheme some form of counter circuit is required which keeps track of the

number of encoder pulses received, counting up or down according to the relative phase of the two signals. It is a relatively simple matter to design such a system using one of the common logic families, as detailed in the hardware control system description given in section 6.1.2.

A more appropriate solution, in terms of simplicity of design and reliability, was to use a number of SN74LS2000 direction discriminator chips, described in appendix 3. This device provides most of the functions required in a position tracking system and is easily interfaced to a bus structure, allowing simple expansion. In order to operate, the IC requires a 5 volt supply, the two encoder pulse trains and a clock, which must be at least 4 times the encoder pulse frequency. The internal 16 bit counter, which may be expanded using extra chips, is connected to a single 8 bit bi-directional bus, the upper or lower byte selected with the appropriate logic signal. The counter is clocked on a single, double or quadruple counting scheme, allowing a resolution of one quarter of the encoder line spacing. Using 500 line per revolution encoders in single count mode gives a typical range of 200 000 counts for the full joint movement. This is greater than can be contained in a single sixteen bit counter and so two devices must be cascaded for each channel. This also allows the quadruple counting scheme to be used, giving a typical resolution of 1:800 000 of the full movement.

This device was employed in two types of circuit, the first, built in a card system, was initially designed to interface to a DEC μ PDP although, by the use of an little extra circuitry an IBM compatible PC could also be used. The second circuit was designed to interface to the IBM PC and, when combined with a motor drive circuit board described in section 3.3.2, provided all the interfacing circuitry required for the controller. Both of the circuits allow the position of up to four encoders to be monitored, each with a 32 bit counter. The input and output data for each of the channels are connected to common 8-bit buses, with the channel and byte selected using a set of control bits.

3.4.1 Position/Velocity measurement circuit 1

The input, output and control functions in this circuit are handled through two, sixteen-bit input and output buses with the functions defined as in table 3.2. The output port is divided into two halves, the lower eight bits providing counter data when the position is initialised while the upper eight bits provide channel select and control functions.

The input port is similarly split into two with the lower eight bits used for reading the position data. Bits eight to eleven give the direction of rotation of the encoder. These four bits are combined with the output from a frequency to voltage converter, read via an A/D converter, which gives a measure of the motor speed. This allows the motor velocities to be found, as described below.

Table 3.2 Definition of input/output bits for position/velocity circuit 1

Bit Number	Input Port Function
0 - 7	Position data input
8 - 11	Direction of rotation of encoders 0 - 3
12 - 15	Not used

Bit Number	Output Port Definition
0 - 7	Counter initialisation data output
8 - 10	Encoder counter select
11	Enable
12	Byte select High for least significant byte, low for most significant byte.
13	$\overline{\text{READ}}$. When low data from counter presented on data bus.
14	$\overline{\text{WRITE}}$. When low data from data bus loaded into counter.
15	Not used

The circuit diagram of the first position and velocity measurement system constructed is shown in figure 3.12. The circuit is split up onto six boards, connected via a common backplane, and consists of four encoder counter boards, one for each channel, a channel select and control board and a velocity measurement board which is also used to buffer the encoder and data bus signals. The circuit is of a straightforward design for the most part although some comment on the velocity measurement system is required.

The absolute value of velocity is obtained using an LM2917 tachometer chip. This is effectively a frequency to voltage converter, with certain features which make it particularly suitable for this application. The component values were selected to give linear operation over the full range of motor speeds. The input count is taken from the exclusive OR of the two encoder phases, effectively doubling the frequency of a single signal. This then passes through a level shifter at the input which keeps the mean input level at zero volts, the threshold level of the clock input. The timing components connected to pins 2 and 3 of the IC are selected as a compromise between the speed of response to a change in motor speed and the level of ripple at low speeds. The value of C_5 also affects the speed range over which the circuit operates. With the component values selected, the output voltage is given by $V_O = 70\mu\text{V}/\text{Count}/\text{s}$, up to a speed of approximately 87k lines/s, giving a voltage of 6V. From a speed of 5kcounts/s up to the maximum speed, the output voltage increases linearly with speed to within 2.5%. When a step change in motor speed was simulated, using a voltage controlled oscillator with a step change in control voltage, it was found that the output responded with a time constant of 0.8mS, significantly shorter than the mechanical time constant of the motor. The output of the IC, at pins 4 and 7 has a 40mV pk-pk ripple superimposed on it at all motor speeds attainable, limiting the dynamic range to 1:300. A high order low pass filter was added at the output, with the intention of removing this ripple. It was found that although the ripple was reduced, the noise generated by this circuit was approximately 100mV and so it was removed.

Figure 3.12 Position/velocity circuit 1

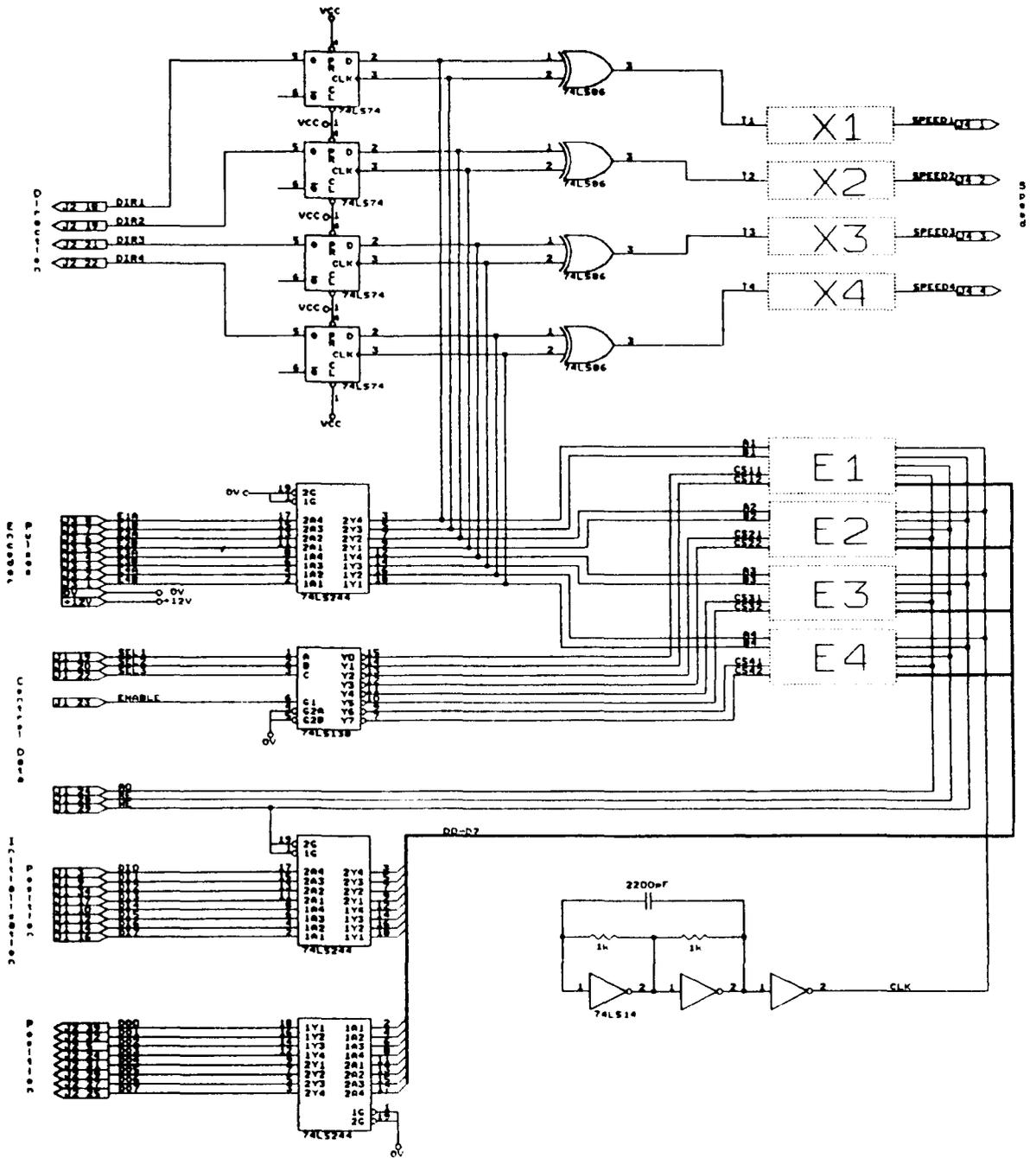
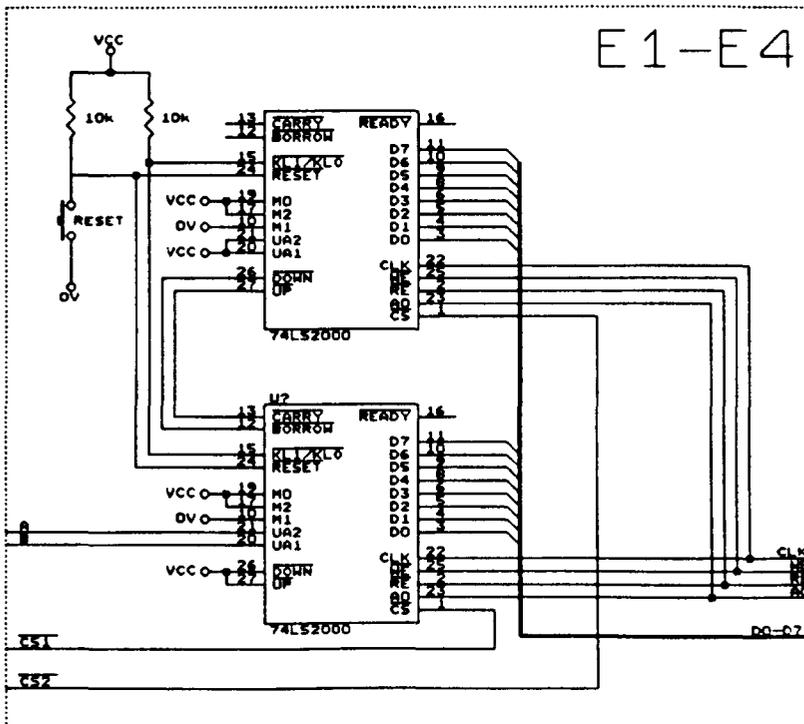
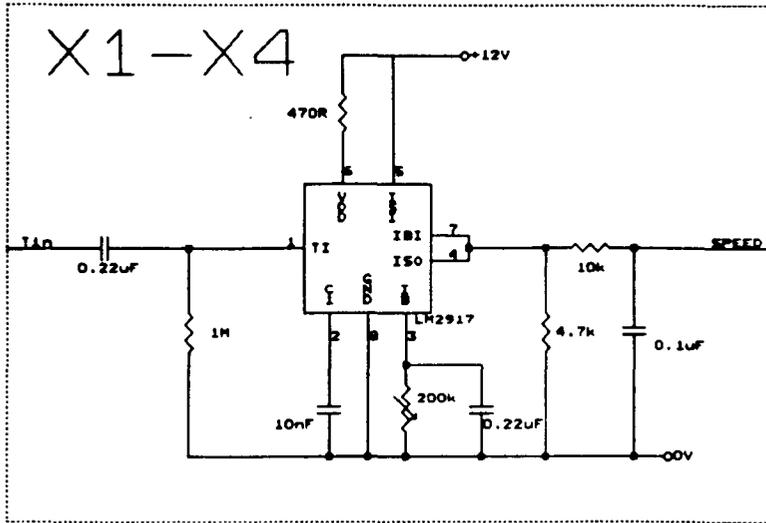


Figure 3.12 Position/velocity circuit 1 (Cont.)



Having obtained a measure of the absolute speed of the motor it remains to find the direction of rotation. This is most easily achieved using a D type flip-flop with the data driven from one encoder channel and the clock from the other. Once the speed and direction signals are obtained they are read into the control computer using an analogue to digital converter and bits 8 to 11 of the input port, as detailed above. The velocity measurement obtained using this circuit is of a similar accuracy to that obtainable using a tachogenerator at significantly lower cost and without greatly increasing the motor inertia.

3.4.2 Position/Velocity Measurement Circuit 2

The design of the second position measurement system, intended for connection to a PC via a PC-14A interface, is similar to that previously described, with the differences being due to the limited I/O capabilities of the interface chosen. Since analogue input circuitry was not available using this interface, an alternative velocity measurement system was required. The method selected used one of the counters on the PC-14A interface, driven from the system clock, to interrupt the processor at a fixed time interval, equal to the controller sample interval, with the velocity found from the difference in position at successive interrupts.

The circuit, which is shown in figure 3.13 was constructed on a single printed circuit board, designed to fit inside the case of the RTX, parallel to the manufacturers control board and the motor drive board described in section 3.4. Since the same 24 I/O lines must be used both for position measurement and motor demand output, the port bit definitions are rather more complex than on the μ PDP interface, with all the bits being used. The three ports of the PC-14A are selected so that ports A and B are outputs, with A providing control functions and B giving output data, while port C is defined as input, for reading position data. The individual bits are used as defined in table 3.3.

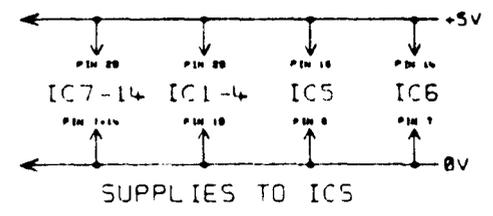
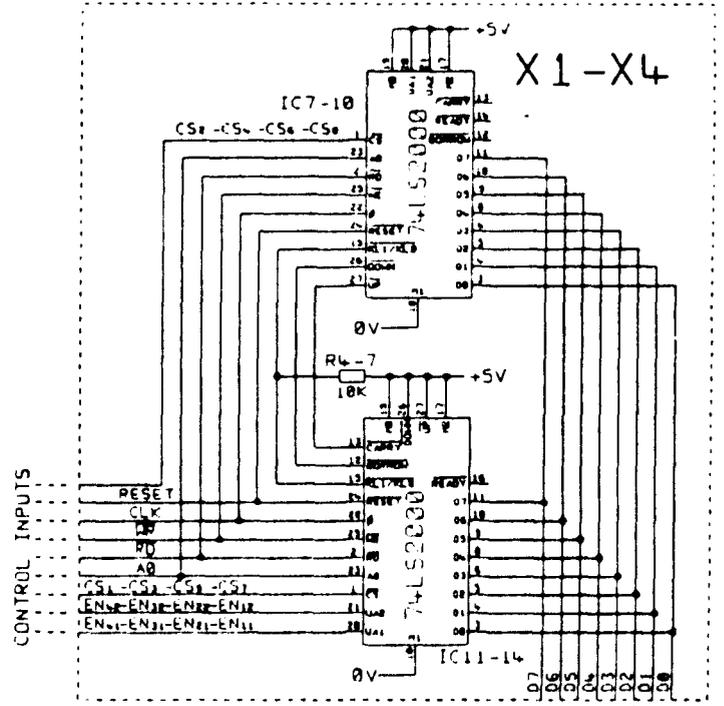
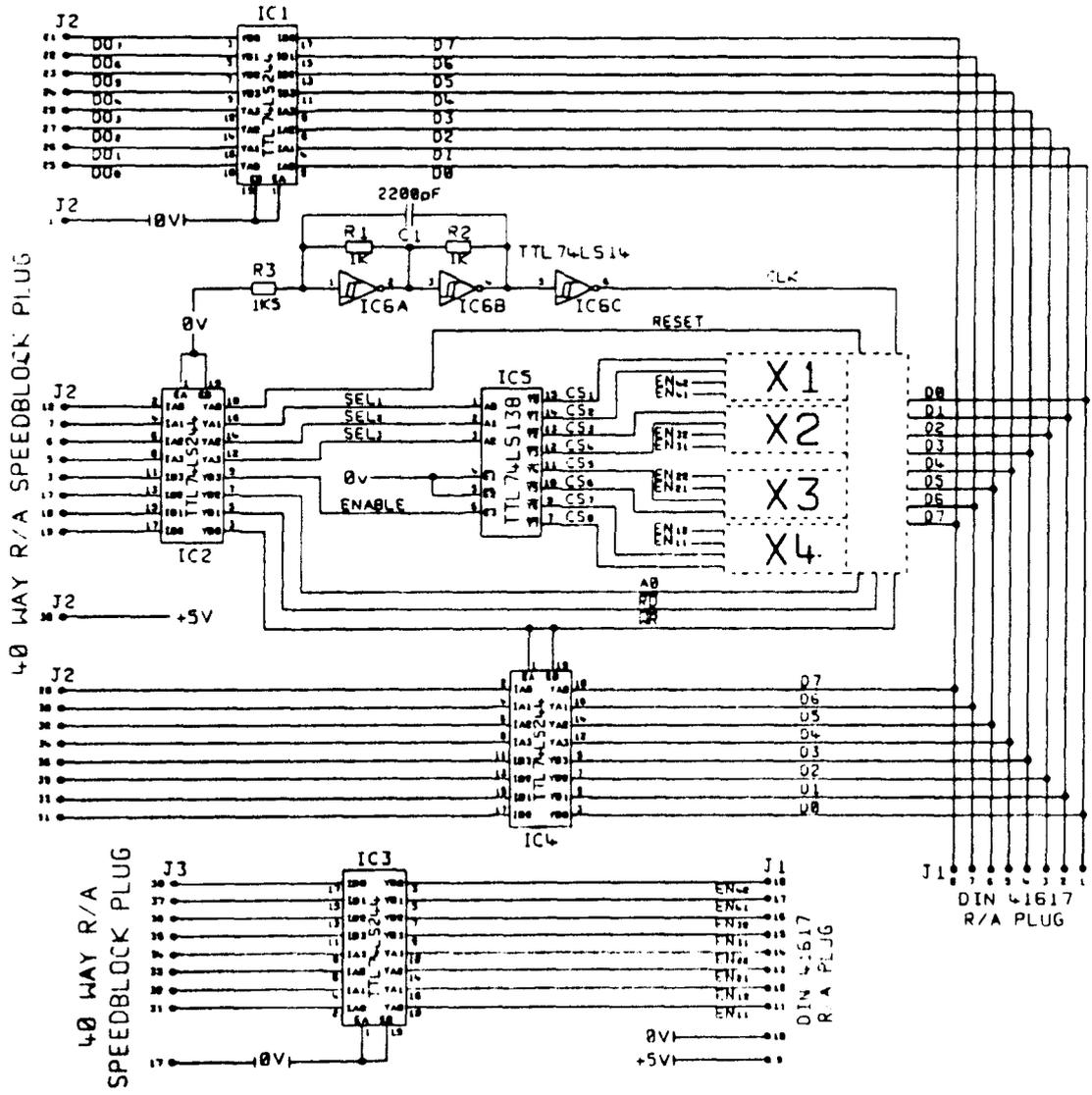


Figure 3.13 Position/velocity measurement circuit 2

Table 3.3 Definition of I/O bits for Personal Computer interface

Port	Bit Number	Function
A	0 - 2	Channel Select
	3	Position / Demand Select High for posn. I/O, Low for demand o/p
	4	Byte Select High for least significant byte, Low for most significant byte
	5	$\overline{\text{READ}}$ When low, read position data
	6	$\overline{\text{WRITE}}$ When low, write position data
	7	Controller Select Low for UMI control High for external control
B	0 - 7	Counter init. data / Demand output
C	0 - 7	Position data input

It will be noted that all eight bits of port A are used to provide essential functions. This did not leave any free to provide the reset pulse to the counter circuits. This function was previously achieved manually by means of an easily accessible switch but this would not be satisfactory in the final implementation and so some other method was required. It would be possible to use one bit of the second set of ports but this was felt to be wasteful as it would require a significant increase in the wiring between the PC and robot. The method selected used one of the spare timer outputs on the PC-14A interface. These outputs were connected to the robot via the ribbon cable already in use for position and demand signal data and, with suitable software could be made to provide a single reset pulse, as required.

The modifications made to the hardware of the RTX robot described in this chapter allow data collection for accurate dynamic modelling of the arm and the design and evaluation of control algorithms. The motor drive system itself allows a twofold improvement in maximum motor speed while the position

and velocity measurement schemes give a significant improvement in accuracy over the existing hardware. With these modifications, it is believed that the performance has been improved as far as is possible without major alteration to the mechanical construction of the manipulator.

CHAPTER 4

RTX Robot Arm Dynamics

When designing controllers for any dynamical system it is vital to have some knowledge of the system dynamics. In addition, a greater understanding of the dynamics allows a controller to be designed which better fulfills a specified criterion, whether that criterion involves dynamic performance, implementation or running cost or any combination of these. This increase in understanding requires the expenditure of time and often money and, in practical situations, a point is reached at which the improvement in understanding of the system dynamics is not worthwhile in terms of the improvement in system behaviour when a control system based on the model is implemented.

The dynamic behaviour of a robot arm is generally complex, involving many types of nonlinearity which make both dynamic modeling and control a challenging subject for research. The nonlinearities found in a typical robot arm may be divided into two categories; those found in the actuation and drive mechanisms and those arising from variations in the link positions and velocities. The relative significance of these two categories affects, to a large extent, the most appropriate control strategy for a particular robot.

In order to develop control systems for the RTX robot a dynamic model was derived which includes the effects of both types of nonlinearity. As explained in chapter 3, the control of three of the joints of the RTX is considered in this work and so a dynamic model is derived only for these joints, although the same procedures would be followed if other axes were considered. It should be noted that the interaction nonlinearities found in the joints considered are far more significant than those found in other joints.

The development of the dynamic model may be split into three sections; the first part will consider the dynamics of the joint motors and current drive

circuits which are common to all the axes considered. Following this, the elastic behaviour of the timing belt drive mechanisms will be dealt with. The effects of interactions between joints and of variations in load will then be investigated and all aspects combined to give a model of the complete system.

The control system implemented by the manufacturers of the RTX is based on a linear model of the arm, taking the worst case inertia parameters into consideration. Since this controller was developed, a model of the interaction dynamics has been obtained by the manufacturers but the elastic and nonlinear behaviour of the actuation system has not previously been investigated. In accordance with this, the majority of the work described here concerns the actuation and drive system characteristics.

The theoretical background to the model will be given first, followed by details of the experimental procedures used to determine model parameters.

4.1 Theoretical Basis for the Dynamic Modeling of the RTX

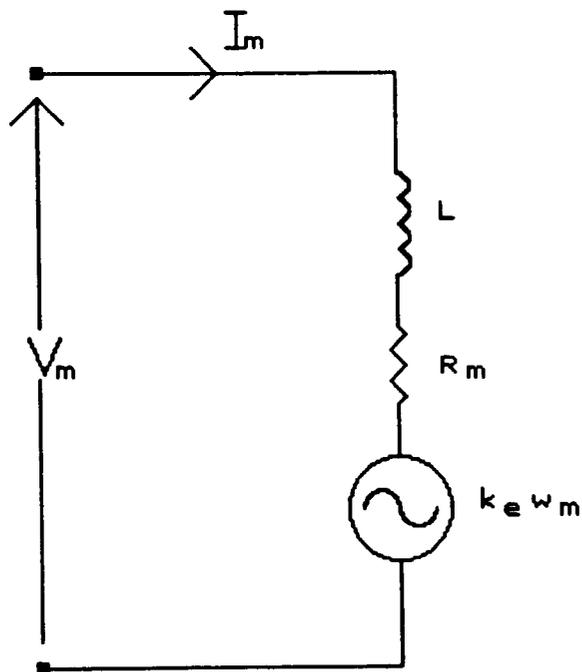
The dynamic modeling of any practical dynamic system involves some degree of compromise between the accuracy of the model and the amount of work involved in determining it. In a real system there is virtually no limit to the complexity of a fully realistic representation and it is up to the engineer to decide when a model is sufficiently accurate for a particular application. The model derived in this chapter is more detailed than would normally be used in the design of a robot control system, mainly in the description of the nonlinear and elastic behaviour of the drive system. This complexity is required because the nonlinearities in the drive system are far more significant in the RTX than in the majority of other robots. In this section the theoretical background will be given for those aspects of behaviour embodied in the dynamic model.

4.1.1 Dynamic Model of D.C. Motor and Current Drive

The dynamics of the joint actuation system are composed of three subsystems which may initially be treated separately. The motor itself can be considered in terms of the electrical and mechanical subsystems while the current drive circuit may be developed once the dynamics of the motor are understood. The design of this circuit is largely dealt with in section 3.3, although the behaviour of the complete actuation system is considered here.

The motors used to drive the axes considered are all of the same model, being a direct current, permanent magnet, brushed type. The electrical components of this type of motor may be represented as in figure 4.1 where V_m and I_m represent the motor voltage and current respectively; R_m and L the rotor winding resistance and inductance and V_b the back e.m.f. generated in the windings, equal to the motor angular velocity, ω_m , multiplied by the constant, k_e .

Figure 4.1 Equivalent circuit of a DC permanent magnet motor



The relationship between motor voltage and current is given by:

$$I_m(s) = \frac{V_m(s) - k_e \omega_m(s)}{sL + R_m}$$

The torque generated by the motor is proportional to the winding current and depends on the relative position of the windings and the poles of the permanent magnet. This relationship is complex and involves many parameters which cannot easily be measured experimentally. A good approximation to the overall behaviour of the motor in the normal operating range is obtained if it is assumed that the generated torque is proportional to the motor current while the torque acting on the load is equal to this, less the frictional torque. The acceleration of the motor is thus given by:

$$\theta_m(s)s^2 = \frac{k_m I_m(s) - T_f(s) - T_o(s)}{J_m}$$

where θ_m is the motor angular position; k_m the motor torque constant; $T_f(s)$ the friction of the motor and gearbox, which is a function of the motor velocity; $T_o(s)$ is the output torque transmitted to the load and J_m is the inertia of the motor and gearbox.

When investigating the dynamics of the motor we may consider both the motor and any components attached directly to it as a single unit by lumping the inertial and frictional elements of each part. In this way the effective inertia, J_m , may be considered as the rotor inertia plus the inertia of the gearbox elements multiplied by the square of the appropriate gear ratios. Similarly the effective friction, $T_f(s)$, is the sum of motor and gearbox friction with corresponding gear ratio multipliers.

From this model it is possible to design a suitable current drive system, as described in section 3.2.2. Combining the dynamic equations of the drive system with those of the motor results in the following equation:

$$\theta_m(s)s^2 = \frac{k_i k_a k_m k_{me} V_{in}(s)}{J_m (s\tau_a + 1)(s\tau_{me} + 1) + k_a k_{me}} (s\tau_i + 1)$$

$$- \frac{k_e k_{me} (s\tau_a + 1) \omega_m(s)}{J_m (s\tau_a + 1)(s\tau_{me} + 1) + k_a k_{me}} - \frac{T_f(s) + T_o(s)}{J_m}$$

k_i - Input filter gain

k_m - Motor torque constant

k_a - Current drive gain

k_{me} - Motor electrical gain

k_e - Motor back e.m.f. constant

τ_a - Drive amplifier time constant

τ_i - Input filter time constant

τ_{me} - Motor electrical time constant

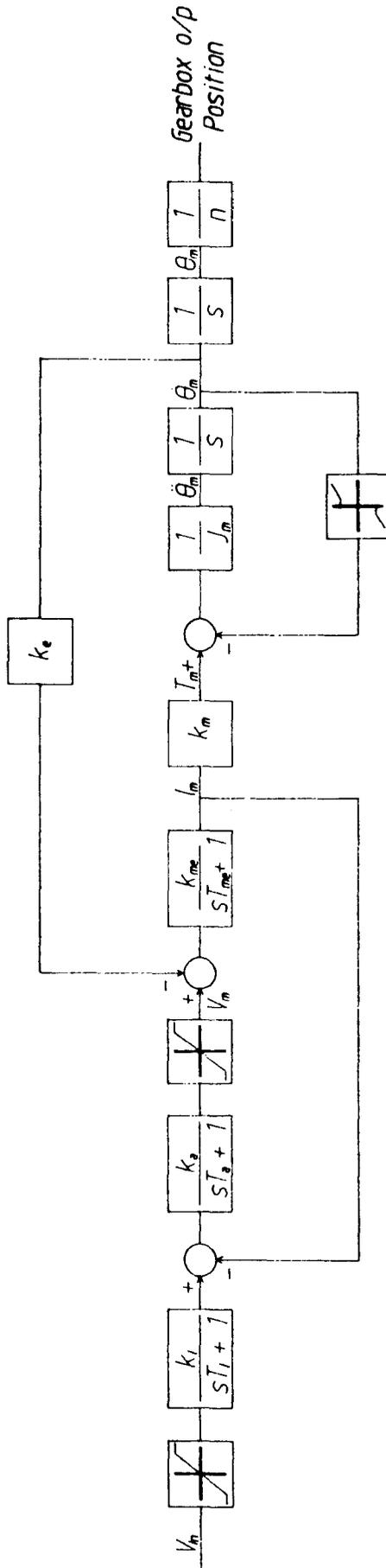
T_f - Motor friction

T_o - Output torque to load

J_m - Motor Inertia

This is the small signal model of the motor and drive system. In addition to the nonlinear friction, there are two other nonlinearities which must be included if the dynamic model is to give an accurate representation of the system behaviour. The first is a saturation element at the input to the current drive circuit which limits the input to $\pm 10V$ and hence limits the motor current to $\pm 500mA$. The second source of nonlinear behaviour is the limited supply voltage used for the current drive. This was set to $\pm 40V$ which, when the voltage drop in the drive amplifier is accounted for, limits the motor voltage to $\pm 37V$. The model including these nonlinearities is represented in block diagram form in figure 4.2

Figure 4.2 Nonlinear dynamic model of a motor and current drive



4.1.2 Frictional Behaviour of Joint Drives

The frictional torque present in the motor and gearbox is a function of the motor velocity. It is often assumed that the friction in a servo system may be represented purely as viscous friction or in some cases a combination of viscous and Coulombic effects. Although such a representation may be sufficiently accurate when friction contributes only a small amount to the overall behaviour; when it is a major factor, a more realistic representation is required. The form of the friction is highly complex with many different processes operating in the motor and in each stage of the gearbox. A further source of friction is the contact between the graphite brushes and the rotating commutator. The study of friction is a large field of continuing research of which only a few salient results will be considered here.

The motor considered here uses journal bearings throughout, with the motor and high speed gears mounted on steel shafts running in bronze bushes while the low speed gears use steel shafts and bushes. The gearbox uses spur gears made of either steel or paxolin, the latter used in the high speed, low torque section. The bearings and gears are lubricated using a viscous grease.

As shown in section 4.1.3, the majority of the flexibility in the transmission to the joints is due to the timing belt in the second stage, closest to the joint, and so the first stages of the belt drive may be considered as part of the motor - gearbox combination. The pulley used between the two stages is, once again, mounted on a journal bearing with the plastic pulley rotating on a steel shaft. In each of the journal bearings and at the interface between gears there are several types of behaviour which may occur, dependant on the speed of motion and the transmitted torque.

4.1.2.1 Low Speed Friction

When parts of lubricated journal bearings move at low speeds the lubricant film between the parts is broken down and the two surfaces come into contact. The frictional behaviour at low speeds is therefore similar to that for unlubricated contacts. Here we are only interested in the behaviour of metal-metal contacts while contact between non metals, and elastomers in particular, depend on significantly different processes.

The first explanation for observed frictional behaviour between metals at low speeds was proposed by Amontons and de la Hire in 1699 [Dawson,79]. They believed that the surface asperities of the two parts meshed with one another so that relative motion was only possible if the asperities of the upper surface were lifted above those of the lower. This accounts for the static friction found in practice but not the energy dissipated in sliding.

There have been many other explanations proposed since this date but that proposed by Bowden in 1950 [Bowden,50] is now generally accepted. The mechanism of friction proposed involves welding, shearing and ploughing behaviour. The rough surface of one part comes into contact with that of the other part at a series of points and at each contact, a weld is formed due to the normal force acting on the parts. When a tangential force is applied the material around these welds must be sheared if motion is to occur.

There may in addition be some plastic flow in the region of contact. Plastic flow in the materials has two effects, both of which increase the friction between the parts; firstly the contact area between the parts increases as a tangential load is applied and secondly work hardening occurs in the region of the junction, making shearing more difficult. The behaviour when plastic flow occurs is similar, on a macroscopic level, to that found with shearing [Moore,75] and so need not be considered separately.

The second component of friction at low speeds is due to ploughing. In this process the asperities of the harder material form grooves in the bulk of

the softer material, with the softer material displaced to the sides of the groove. The difference in hardness may be due to the material used or possibly to work hardening of the asperities following plastic flow. It seems likely that this term will be of less significance than the shearing term in materials of similar hardness [Bowden,50].

There have been many attempts to characterise the dependence of friction on speed, mostly derived empirically, but that which most adequately describes the behaviour found is of the form:

$$\mu = (a + bV)\exp(-cV) + d$$

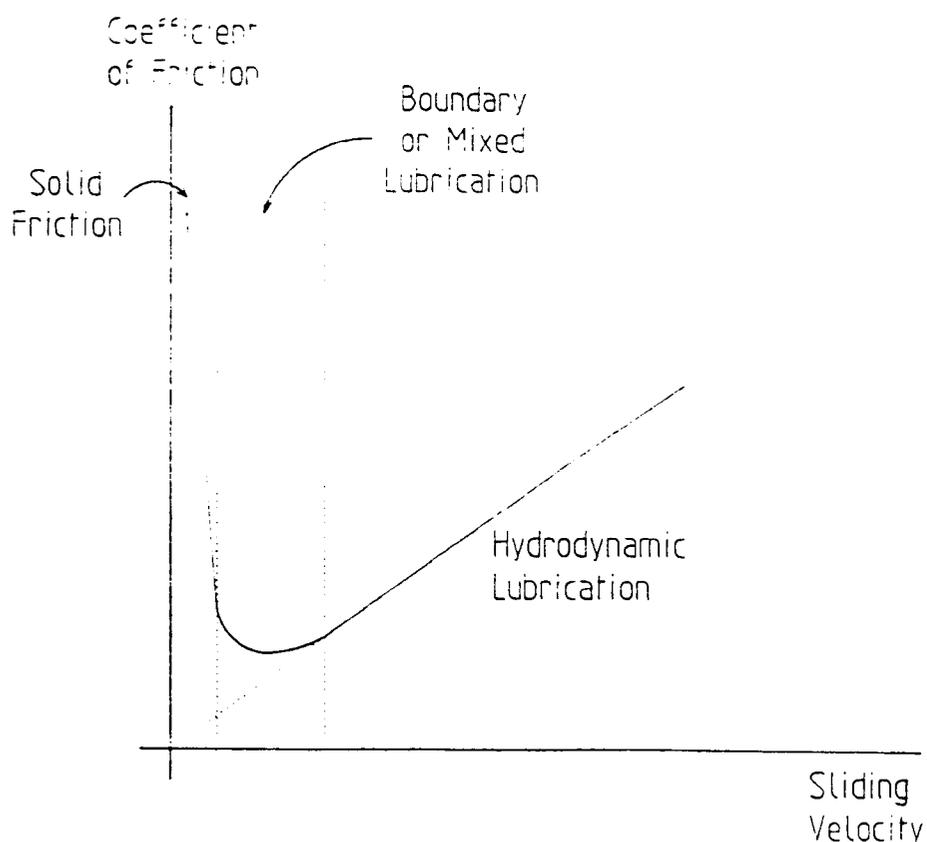
where μ is the coefficient of friction, equal to the ratio of frictional to normal force; V is the sliding velocity and a, b, c and d are constant for given materials and load conditions [Moore,75]. This indicates that at low speeds μ increases with increased speed but drops again as higher speeds are reached. It is also found that the value of μ decreases for an increase in load, at a given sliding speed. It is unlikely, at the speeds considered in this instance, that a significant reduction in μ would occur before the effects of lubrication became dominant. The behaviour of bearings at these higher speeds will be considered in the next section.

4.1.2.2 High Speed Friction

In the previous section it was assumed that the speeds of relative motion were sufficiently low that a lubrication film could not be established and so the frictional behaviour was governed by metal-metal contacts. As the speed of motion is increased however, lubricant is forced into the channel between the moving parts and forms a film which prevents contact between the moving parts. This type of behaviour is known as hydrodynamic lubrication and friction in this regime is controlled by the viscosity of the lubricant and so an increase

in speed results in an approximately proportional increase in friction. This assumes that a constant and uniform lubricant film is produced with no contact between the moving parts. This is not in reality the case and there is a range of speeds at which a combination of solid and hydrodynamic friction occurs. In this case the surfaces are, for the most part, separated but some contact between surface asperities still occurs. This type of behaviour, known as mixed or boundary lubrication, is highly complex, depending on the adsorption and physico-chemical interactions which determine the exact nature of the solid contact. It is in this mode of contact that the lubricity or 'oiliness' of the lubricant becomes significant. The behaviour under conditions of mixed lubrication can be seen as a combination of the properties of solid and hydrodynamic friction so that the overall friction for a wide range of speeds takes the form shown in figure 4.3 [Moore,75].

Figure 4.3 Variation of the coefficient of friction with sliding speed for fixed normal force



As the loading on the bearing increases the velocity at which fully hydrodynamic lubrication begins increases, due to the increased load on the lubricant film which tends to force lubricant out of the loaded region.

At low speeds a negative coefficient of friction is seen and it is this which causes stick-slip phenomena to occur. As an example of this behaviour, if a block is in contact with a moving base and is under the restraining influence of a spring, as shown in figure 4.4, then, if the spring is initially relaxed, the static friction will cause the block and base to move at the same velocity. This will extend the spring, causing a force on the block in the opposite direction to the movement. A point is eventually reached at which this force exceeds the static friction and the block begins to move relative to the base. As soon as this movement begins, the friction drops and the speed of relative motion increases. Movement continues until the force exerted by the spring becomes less than the frictional force and the block comes to rest, relative to the base. This cycle of sticking and slipping is then repeated. The position of the block is as shown in figure 4.5 as is the coefficient of friction between the members [Bowden,50].

Figure 4.4 Spring - mass system in which stick - slip may occur

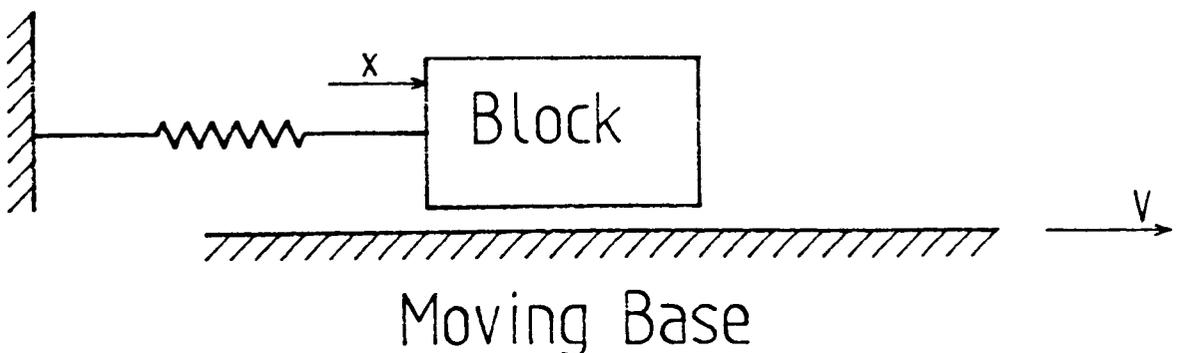
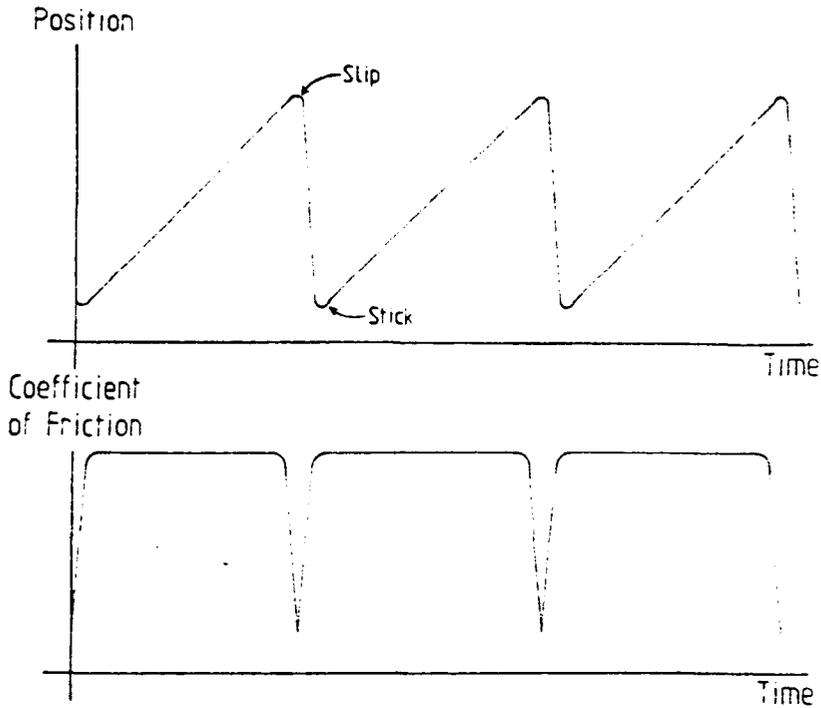


Figure 4.5 Position and coefficient of friction under stick-slip conditions



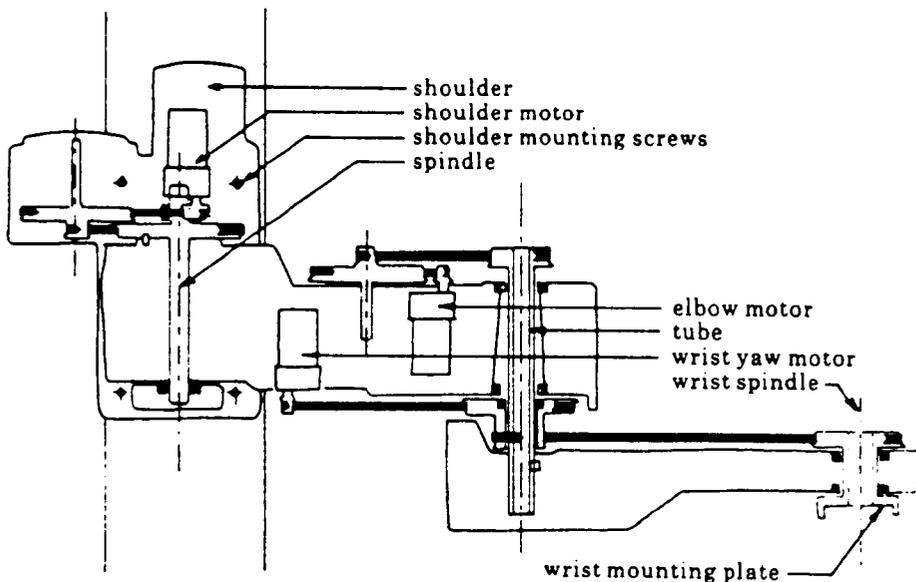
A similar type of behaviour often occurs in servo systems having this type of frictional behaviour when proportional control is used to force the position to track a varying demand. As the position error increases the actuation signal also increases up to the point at which slipping occurs. The position then moves past the point required and stops. The same process then begins again, giving a repeated stick - slip characteristic. If proportional - integral control is used on a system of this type, instability may occur in the form of limit cycling of the position around the desired value.

Although the frictional behaviour of any particular stage of the drive system may be estimated to within an order of magnitude from the above considerations, the combined effect of all sources of friction cannot easily be found and is indeed liable to change due to wear and other external influences. It is thus necessary to measure the friction in each joint experimentally while being guided in these experiments by the general trend of behaviour expected.

4.1.3 Elastic Behaviour of Joint Drive Mechanism

Timing belts provide a low cost and simple means of transmitting power from the motors to the joints of the robot. They do however introduce a significant degree of flexibility between motor and joint. This elasticity has a major effect on the dynamics of the arm and, since the arm position is measured at the motor, it means that the sensed position is not always correct, leading to significant end point inaccuracy.

Figure 4.6 Timing belt transmission used in the RTX

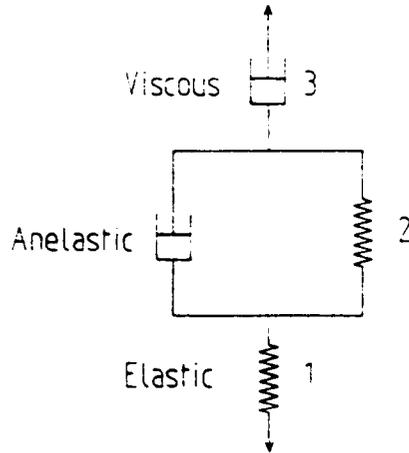


The arrangement of motors and belts used in the RTX is shown in figure 4.6 from which it may be seen that the power for the three joints considered here passes through two belts each. The belts used in the first stage transmit low torques at high speeds and are constructed of polyurethane with steel tensioning members while those in the second stage convey higher torques at lower speeds and use rubber with nylon fabric reinforcement. The effect of applying an external stress to the two types of belt is significantly different. In the steel reinforced belt the strain follows an essentially Hookeian relationship, the strain being proportional to the applied stress. In the case of the fabric reinforced belt the behaviour is far more complex, depending on the strain rate

and the previous stresses applied.

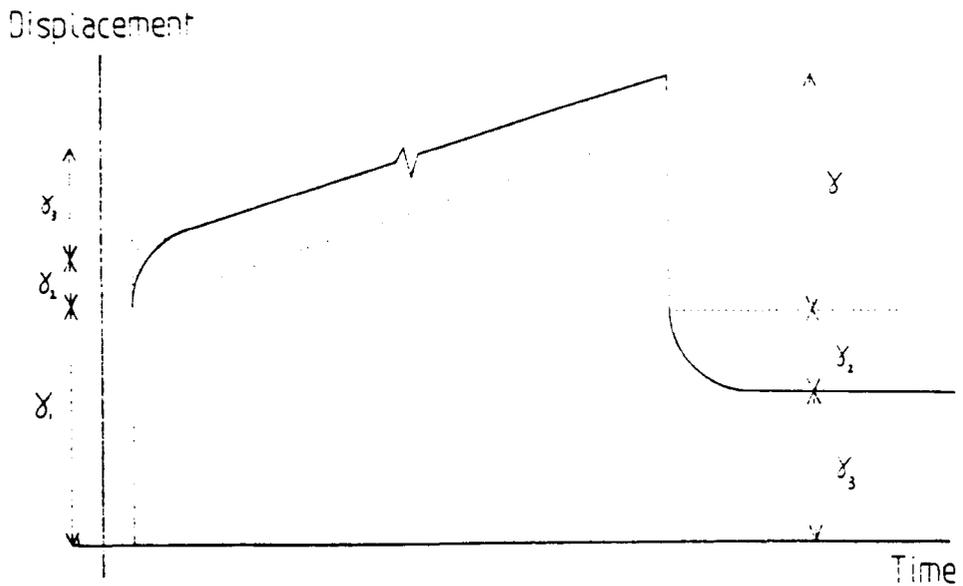
The analytical model of a fabric reinforced rubber belt depends on the properties of both the rubber matrix and the reinforcing fibres. A useful model, although it involves many simplifications, can be represented as the spring - dashpot system shown in figure 4.7. [Hayden,65]

Figure 4.7 Viscoelastic model of a polymeric material



When a fixed load, F is applied to the material the strain response is of the form given in figure 4.8 [Van Vlack,75].

Figure 4.8 Viscoelastic displacement of a polymeric material



It can be seen that the total extension γ at a given time t is:

$$\begin{aligned}\gamma &= \gamma_1 + \gamma_2 + \gamma_3 \\ &= F/G_1 + (F/G_2) (1 - \exp(-tG_2/\eta_2)) + tF/\eta_3\end{aligned}$$

where G_i is the shear modulus and η_i the viscosity of element i .

It may be noted that while the elastic and anelastic deformation, γ_1 and γ_2 , are recoverable, the viscous component, γ_3 is not. If a sufficiently large load were applied to a belt such that viscous behaviour occurred then permanent deformation would take place. In practice therefore the applied load must remain below this threshold.

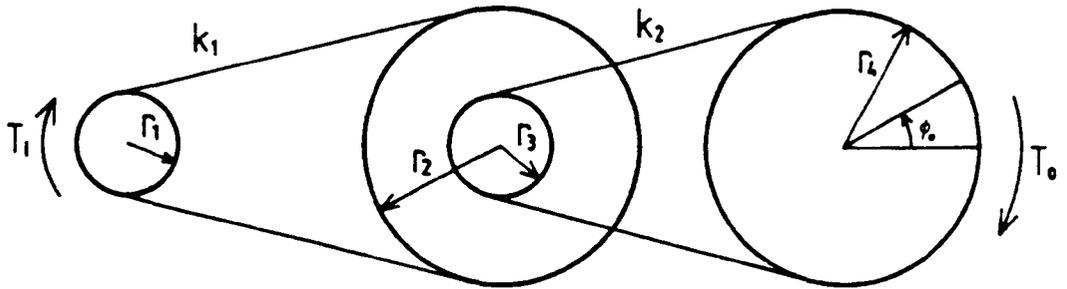
If we look at the situation when a belt of this type is extended by an amount $\gamma(s)$ and we seek the restoring force generated in the belt, it can be seen from above that, neglecting the viscous component, the restoring force is given by:

$$F(s) = \frac{(s\tau_b + 1) \gamma(s)}{k_{s2}(s\tau_b + 1) + k_{s1}}$$

where k_{si} is Young's modulus for element i and τ_b is the time constant of anelastic displacement, given by η_2/G_2 .

If we consider the simplified situation, in which two belts which obey Hooke's law are fitted to a series of pulleys as shown in figure 4.9, we require the relationship between the torque at the output stage and the rotation of the final pulley, when the first pulley is fixed; i.e. how φ_0 depends on T_0 .

Figure 4.9 Schematic diagram of belt transmission system



It is a simple matter to show that in the static case the angular deflection is given by:

$$\varphi_0 = \frac{T_0}{r_2^2} \left\{ \frac{1}{K_2} + \left[\frac{r_3}{r_2} \right]^2 \frac{1}{K_1} \right\}$$

where K_1 and K_2 are the spring constants of the first and second belts, equal to $1/(k_{s1} + k_{s2})$ in each case. The behaviour is the same as that for a single belt system but with the spring constant modified to take account of both elements. It should be noted that in the joints of the RTX considered here, the ratio r_3/r_2 is typically 0.2 and so the effect of elasticity in the first belt is significantly less than that of the second belt, providing the spring constants are of a similar order.

The viscoelastic behaviour of the individual belts may be combined in a similar manner and when the model of the motor and current drive system is included, we obtain the block diagram representation of a single joint of the robot shown in figure 4.10.

4.1.4 Joint Interaction Model of the RTX

In the previous section it was assumed that the inertia of a link of the robot was constant and that no external forces were applied to the joint. This is not the case for the RTX or indeed for most other robots. In reality the effective inertia of a particular link depends on the positions of the subsequent links in the kinematic chain and on the load being held by the end effector. External forces are exerted on a link by the motion of other links in the chain due to centripetal and coriolis effects and to torque coupling. In addition many robots suffer from the effects of gravity loading. Interaction forces may be minimised by choice of suitable kinematic configurations. This often makes the robot less efficient in other respects, typically reducing it's working volume for a given weight of arm. Such designs are also not robust to variations in load. In the case of the RTX, little attempt has been made to reduce these interaction effects and so it is important that they be modeled. The manufacturers of the RTX have derived the form of the interaction nonlinearities but the individual parameters have not previously been found [UMI].

The dynamics of a kinematic chain may be derived in many ways; the simplest method however uses Lagrange's equation. It is found [Paul,81] that the link dynamic equations for an n degree of freedom manipulator take the form:

$$J(\theta) \ddot{\theta} = C(\dot{\theta}, \theta) + G(\theta) + T \quad (4.1)$$

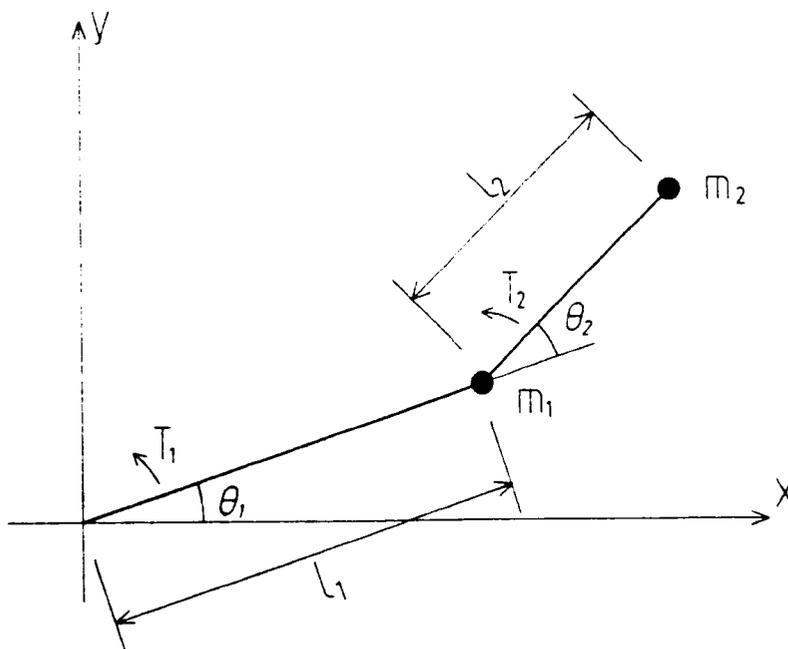
where θ is an n vector of joint angles/displacements and $J(\theta)$ is an $n \times n$ inertia coupling matrix. The three terms on the right hand side are n vectors of which $C(\dot{\theta}, \theta)$ represents the effects of centripetal and coriolis forces/torques; $G(\theta)$ gravity loading and T the input forces/torques.

This equation represents the full nonlinear interaction dynamics for a

general n joint system. In a practical robot there are many terms in equation 4.1 which are zero or very small because significant interactions do not occur between all axes. In the case of the RTX the Z axis has no effect on the remaining axes and is only affected itself by the gravity loading term. The Roll and Pitch axes have very little effect on the remaining joints due to the low radius of gyration of the parts they move. Similarly the wrist unit is approximately balanced about the Yaw axis and since the distance from this axis to the end effector is small there are relatively small interaction effects with other joints. This assumption becomes invalid if a heavy load is carried at a large distance from the end effector. This situation does not generally occur however, due to the limited load carrying ability of the RTX.

It will therefore be assumed that the interaction dynamics with overriding significance are those between the shoulder and elbow joints. Since these joints operate in a horizontal plane there is no gravity loading effect and so $G(\theta)$ is zero. This simplified mechanism is shown in figure 4.11 where l_1 and l_2 are the link lengths and m_1 and m_2 the equivalent lumped masses of the links and wrist and end effector units.

Figure 4.11 Representation of the shoulder and elbow links on the RTX



Applying Lagrange's equation, it is straight forward to derive the equation of motion for this system and it is shown in appendix 4 that the equation may be written:

$$\begin{bmatrix} l_1^2(m_1+m_2)+m_2l_2^2+2m_2l_1l_2\cos(\theta_2) & m_2l_2^2+m_2l_1l_2\cos(\theta_2) \\ m_2l_2^2+m_2l_1l_2\cos(\theta_2) & m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} - \begin{bmatrix} m_2l_1l_2\dot{\theta}_2(2\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_2) \\ -m_2l_1l_2\dot{\theta}_1^2\sin(\theta_2) \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

It should be noted that the input torques to the joints, T_1 and T_2 , are the torques at the end of the belt drive mechanism and not those generated by the motors.

By inverting the inertia coupling matrix we can obtain the joint accelerations:

$$\ddot{\theta} = J(\theta, \dot{\theta})^{-1} C(\theta, \dot{\theta}) + J(\theta, \dot{\theta})^{-1} T$$

where:

$$J^{-1} = \Omega \begin{bmatrix} m_2l_2^2 & -m_2l_2^2-m_2l_1l_2\cos(\theta_2) \\ -m_2l_2^2-m_2l_1l_2\cos(\theta_2) & (m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos(\theta_2) \end{bmatrix}$$

where:

$$\Omega = \frac{1}{(l_1l_2)^2m_2(m_1+m_2)\sin^2(\theta_2)}$$

It is clear from the preceding sections that the dynamic model of a practical robot manipulator is highly complex, involving many nonlinear effects. Although some of the parameters of the arm may be found either from manufacturers data or simple measurements there remain a large number that

cannot be treated in this way. It is therefore necessary to perform a number of experiments if an accurate model is to be obtained. All the experimental work involved in deriving a dynamic model of the RTX is described in the following section.

4.2 Experimental Dynamic Modelling of the RTX

It has been shown that the dynamics of a typical robot are highly complex and nonlinear in nature. Identification of dynamical systems is a widely researched field and there are many approaches available, see [Doebelin,80] and [Eykhoff,74] for example. These may essentially be divided into three categories; analytical, statistical and *ad hoc* methods. The first of these techniques considers each individual component and mathematically describes its relationship to the others. It is dependant on a good understanding of the relationship between elements and knowledge of the parameters of each element. By making assumptions based on experience, it is often possible to obtain a reasonable system model, although this should be compared with experimental data if possible. Analytical methods can incorporate nonlinear system behaviour in the model provided it is of a known form. The construction of analytical models is generally time consuming and depends for its success on a good understanding of all the significant processes; such an understanding is sometimes not possible.

By comparison with analytical methods, statistical identification methods demand far less *a priori* knowledge of the system under test, often requiring only an estimate of the system order. They typically depend on sequences of input-output data to construct the best estimate of the true system. Although statistical methods were first applied to linear systems there are algorithms which can identify a limited range of nonlinear systems [Billings,87 and Chen,88]. Algorithms for the identification of nonlinear systems present a compromise between the amount of *a priori* knowledge available about the

system and the computation involved in determining system structure and parameters. In other words, if little is known about the system, particularly the position and form of the nonlinearities, then a very large amount of computation is required while if the system is largely understood then relatively little computation is involved. The results obtained using statistical algorithms do not generally provide much insight into the internal behaviour of the system.

Ad hoc methods use a combination of analytical and experimentally derived results for a number of subsystems within the plant to construct a suitable model. In this way, those parts which are well understood are treated analytically while experimental procedures are used to gain insight into the parts which are not. The procedure depends on an understanding of the way in which the subsystems interact without necessarily a full knowledge of the workings of each of the subsystems. Those subsystems which are not understood are investigated using suitable sets of input-output data, possibly using statistical methods, until a model with the required accuracy is obtained. In this way a model of the complete system may be built up. It is worthwhile checking that the behaviour of a series of subsystems performs as expected to allow changes to be made before the full model is developed. This may be an iterative process of constructing a subsystem model, simulating the model behaviour and comparing the results with experimental data and modifying the model appropriately. Clearly there are practical systems for which a division of this kind is not possible but where it is, *ad hoc* techniques often prove to be the most effective.

In the case of the RTX a good understanding of the behaviour of the major components is possible since the overall system may be broken down into a number of relatively simple subsystems, as shown in the previous sections. The type of dynamics expected in these subsystems is known although the parameters may not be. It was therefore proposed that the parameters of the dynamics of each subsystem be found by whatever method was considered most appropriate, with the resulting models compared with experimental results using

simulation techniques. Once a sufficiently accurate model had been derived for a set of subsystems, the overall behaviour of that set was compared with experimental results. The subsystems described in section 4.1 will now be considered in turn although these are themselves subdivided in some cases. In this way a complete dynamic model of the RTX may be constructed.

4.2.1 Experimental Dynamic Modeling of Motor and Current Drive

Consideration of the dynamics of a single joint actuation system will be split into two parts, the first concerning the parameters of the motor itself while the second involves the frictional behaviour of the motor and drive system.

4.2.1.1 Experimental measurement of motor parameters

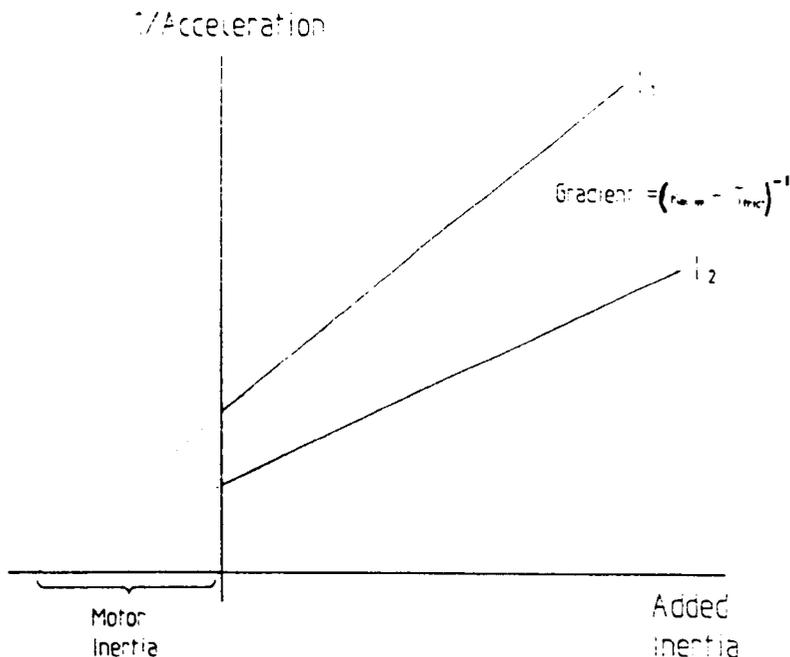
It was shown in section 4.1.1 that the dynamics of the RTX motors, when used with the current drive circuit derived in section 3.3, may be described by a third order equation with the three time constants and three gains, defined by the parameters of the input filter, drive amplifier and motor windings. Each of these constants may be measured directly by applying pulses of known amplitude to the input of the subsystem and recording the output. In the case of the motor winding parameters a relatively low amplitude signal must be used to avoid loading of the of the signal generator. In using a low signal level, it is assumed that this subsystem is linear. This is justified however by results described later using the current drive circuit to provide large amplitude signals.

The remaining unknown parameters in the motor–drive model are the rotor inertia, the motor torque and back e.m.f. constants and the frictional torque. The measurement of the frictional behaviour will be considered in section 4.2.2 but the measurement of rotor inertia and torque constant will now be described.

There are many techniques available for calculating and measuring the

inertia and torque constant. The method used here was selected because it allows both values to be calculated from a single set of measurements. The technique consists of applying a known, constant current to the motor windings and measuring the resultant acceleration with a series of loads having known inertias attached to the rotor. The measurements were made when the current was first applied since in this region the current and acceleration remain approximately constant. By plotting the inverse of acceleration against the added inertia it is possible to determine the torque constant from the gradient of the line and by extrapolating back to the inverse acceleration axis, the rotor inertia is found as shown in figure 4.12. During the acceleration, the velocity dependant friction would change. However, by performing these tests with the motor detached from the rest of the arm, the friction was minimised and, as may be seen from figure 4.14, it does not vary by a great deal. By performing tests at a number of current levels, the remaining friction torque may be eliminated from the calculations.

Figure 4.12 Calculation of torque constant and rotor inertia



The calculation of acceleration from the position measurement using the repeated backward difference method is subject to a significant amount of noise

due to quantization of the position measurement, typically 15% peak-to-peak at the maximum acceleration using a sampling rate of 500Hz. To reduce this noise, a first order recursive digital filter was used to smooth the acceleration estimate obtained, without having a significant effect on the relatively slowly changing acceleration. In this way the noise was reduced to typically 3.5% peak-to-peak under the same conditions. This method was used to determine the torque constant; however, for the calculation of rotor inertia it is not necessary to know the true value of acceleration provided a quantity proportional to acceleration is available. It is thus possible to use the time taken to reach a fixed velocity as a measure of acceleration, thus avoiding the noise problem described earlier.

The back e.m.f. constant of the motor is an important parameter since it defines the maximum speed attainable for a given drive voltage. The technique used to measure it here is a by-product of the friction measurements described in the next section. This involves driving the motor at a number of fixed speeds while monitoring the voltage and current driving the motor. At a steady speed the motor voltage is made up of two components; one, proportional to the motor current, accounts for power dissipated in the motor windings and in overcoming friction, while the second, proportional to motor velocity, corresponds to the back e.m.f. generated. Since the motor current, winding resistance and frictional torque are known, the dissipative terms may be calculated and the back e.m.f. found from the measured motor voltage. In practice the dissipative term is of little significance and so this method gives highly accurate results. Using these techniques the parameters given in table 4.1 were obtained for RTX serial number 070 and are compared with values derived from manufacturer's data [Bühler] and component values:

Table 4.1 Motor and drive system parameters

<u>Parameter</u>	<u>Analytic</u>	<u>Experimental</u>	<u>Units</u>
Input filter time constant, τ_j	0.98 \pm 0.2	1.06 \pm 0.04	ms
Input filter gain, k_j	49 \pm 2	49 \pm 2	mV/V
Drive amplifier time constant, τ_a	73 \pm 4	80 \pm 4	μ s
Current loop gain, k_a	990	990	V/V
Motor electrical time constant, τ_{me}	0.45	0.57 \pm 0.07	ms ¹
Motor electrical gain, k_{me}	46	44 \pm 1	mA/V
Motor back e.m.f. constant, k_e ²	0.489	0.63 \pm 0.015	V/r/s
Motor torque constant, k_m ²	0.489	0.481 \pm 0.009	Nm/A
Rotor inertia, J_m ²	3.68	3.07 \pm 0.06	$\times 10^{-4}$ Kgm ²
Steady state gain, k_d	46	43 \pm 1	mA/V

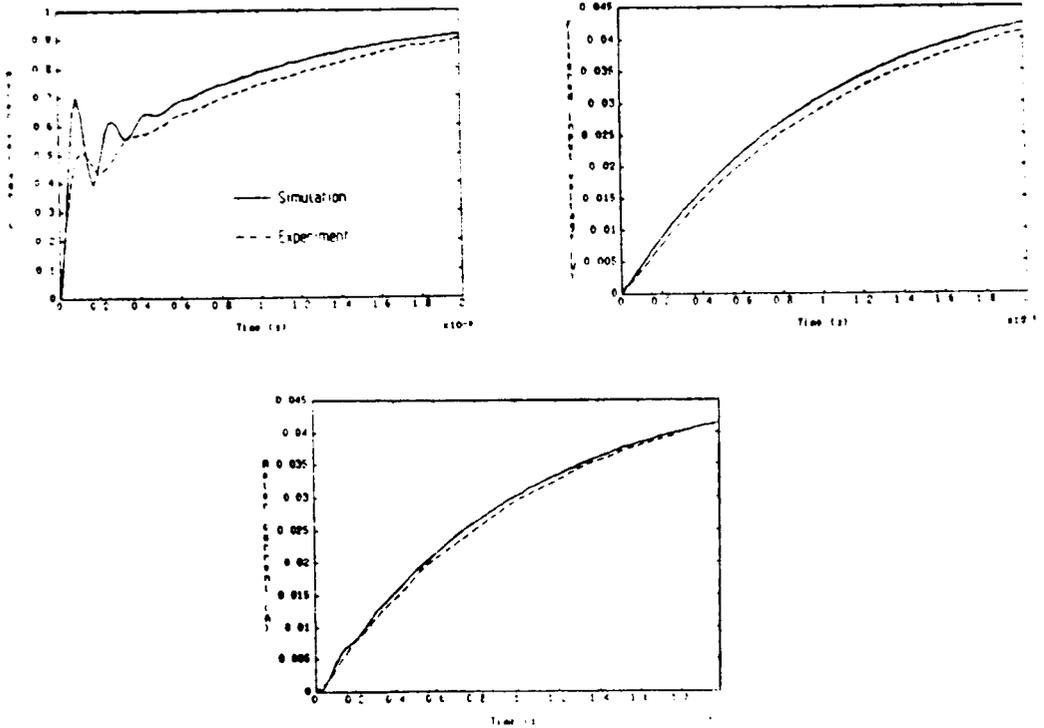
¹ Range of values for all motors tested. For motor used in remaining tests

$$\tau_{me} = 0.6\text{ms} \pm 0.04\text{ms.}$$

² Parameter values referred to gearbox output

With the input filter in place, the step responses of the motor and current drive circuit for a 1V step change in input, derived from experiment and simulation are as shown in figure 4.13.

Figure 4.13 Simulated and experimental step response of motor and drive circuit



The agreement between the simulated response and that measured using a digital storage oscilloscope is good in respect to the filter output voltage and the motor current. The agreement in motor voltage is generally good, except in the initial dynamics. This is due to unmodelled motor and amplifier dynamics and disturbances. The effect of any disturbance is apparent, due to the low damping of the control loop, a situation which would only be remedied by a reduction in gain. These have little effect on the overall system behaviour since the time scale of this behaviour is of the order of $500\mu\text{s}$.

4.2.1.2 Measurement of motor and transmission system friction

The most appropriate technique for friction measurement depends on the type of mechanism involved, the level of friction expected and the speed range of interest. A number of methods are presented by Moore [Moore,75] but these are more appropriate for measurement of friction between two components

rather than for a complete system. When friction measurement specifically for robotic applications is considered a number of procedures are given by Armstrong [Armstrong,88].

For the measurement of breakaway friction a gradually increasing torque may be applied to the stationary arm until motion just begins. This type of measurement is sensitive to external mechanical and electrical disturbances but gives a reasonably accurate result. At low velocities a stiff velocity control loop may be used to maintain a constant joint velocity while monitoring the torque required to maintain the set speed. This technique suffers from the effect of the negative coefficient of viscous friction at these speeds which makes accurate control difficult. Finally, for high speed measurements, Armstrong uses open loop control torques, precalculated to give a certain velocity profile, the difference between expected and measured velocity is then due entirely to the frictional torques. These techniques were applied to a Unimation PUMA 560 arm with results similar to those given here, although it should be noted that the friction terms in the PUMA are far less significant than in the RTX.

The method used in this instance is similar to the velocity control loop used by Armstrong but uses PI control in the velocity loop with integral anti-windup. When compared with a simple proportional control, this drastically reduces the steady state velocity error and gives improved velocity tracking at low speeds where a negative friction coefficient is present. Using this control, implemented on a PC with a sampling rate of 250Hz, it was found that acceptable control could be obtained at a large range of speeds, although performance was degraded at low speeds as expected, with stick-slip behaviour occurring below 0.4 r/s in the elbow joint. This velocity is 0.3% of the maximum for that joint. A common way to overcome the problems of low speed control in the presence of static friction involves the use of a dither signal, added to the control, which keeps the system in continuous, high frequency motion. This would not be appropriate in this instance however, since the effect for which we are looking would be masked by the dither signal.

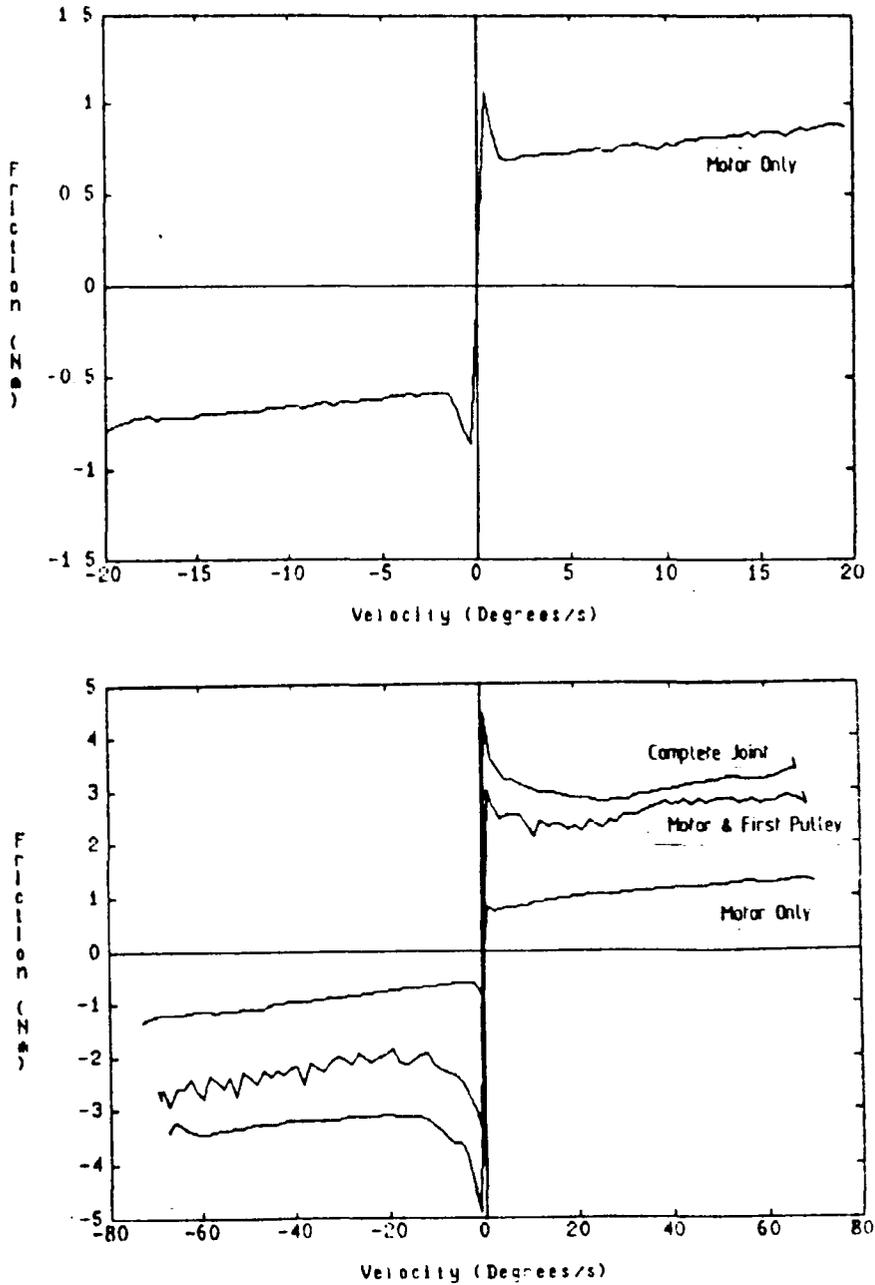
The friction measurement was automated to the extent that tests could be performed over a range of velocities, with a given velocity increment. At each of the set velocities, the mean and variance of the actual velocity, the control signal, the motor current and voltage were stored for later plotting. Out of a total of 250 samples at any set speed, only the last 150 were used in the statistical analysis, so that the acceleration period would not be included. This results in measurements starting 0.4 seconds after the application of the step, allowing sufficient acceleration time, as seen in figure 4.19. Between each of the friction measurements the joint was initialised to the limit of motion in the opposite direction to that of the test so that all readings were over the same region of operation. It was found that, for a given speed, there were some variations in the value of friction at different joint angles. Since these changes were less than 5% of the frictional torque, they were not included in the model derived.

The relationship between friction and velocity for each stage of the shoulder drive system are shown in figure 4.14. The individual components of friction were found by disconnecting the drive from those parts not of interest. Thus the motor friction was found by performing the tests with the first belt removed while only the second belt was removed when measuring the friction in the motor and first pulley. It should be noted that the velocities marked are those at which the joint itself would rotate, where it connected.

It is interesting to note that, with the motor alone, the friction is almost wholly that expected for a hydrodynamic lubrication scheme, with boundary lubrication behaviour only occurring below speeds of approximately $1.7^{\circ}/s$, referred to the joint axis. With the first pulley attached it is seen that mixed lubrication behaviour continues up to a velocity of $15^{\circ}/s$ at the joint axis. This is caused by the relatively low speed of the first pulley, compared with that of the motor itself. With all parts of the transmission system included it is seen that boundary lubrication behaviour continues up to $30^{\circ}/s$. It should also be noted that the maximum absolute value of static friction is 4.8Nm compared

with a maximum drive torque of 10.6Nm. This large relative frictional component constitutes one of the major problems in controlling the RTX, particularly at low speeds.

Figure 4.14 Frictional torque in the shoulder joint of the RTX as a function of speed



The frictional behaviour of the elbow joint is of a similar form to that found for the shoulder axis, as is seen from figure 4.15. An added consideration in the case of the elbow axis is that, since the drive to the yaw

joint passes through this axis, an additional element of friction is caused by this drive. The extent to which this occurs is seen in figure 4.16 where the friction found in the complete elbow axis is plotted against speed, with and without the yaw axis drive connected.

Figure 4.15 Frictional torque in the elbow joint of the RTX

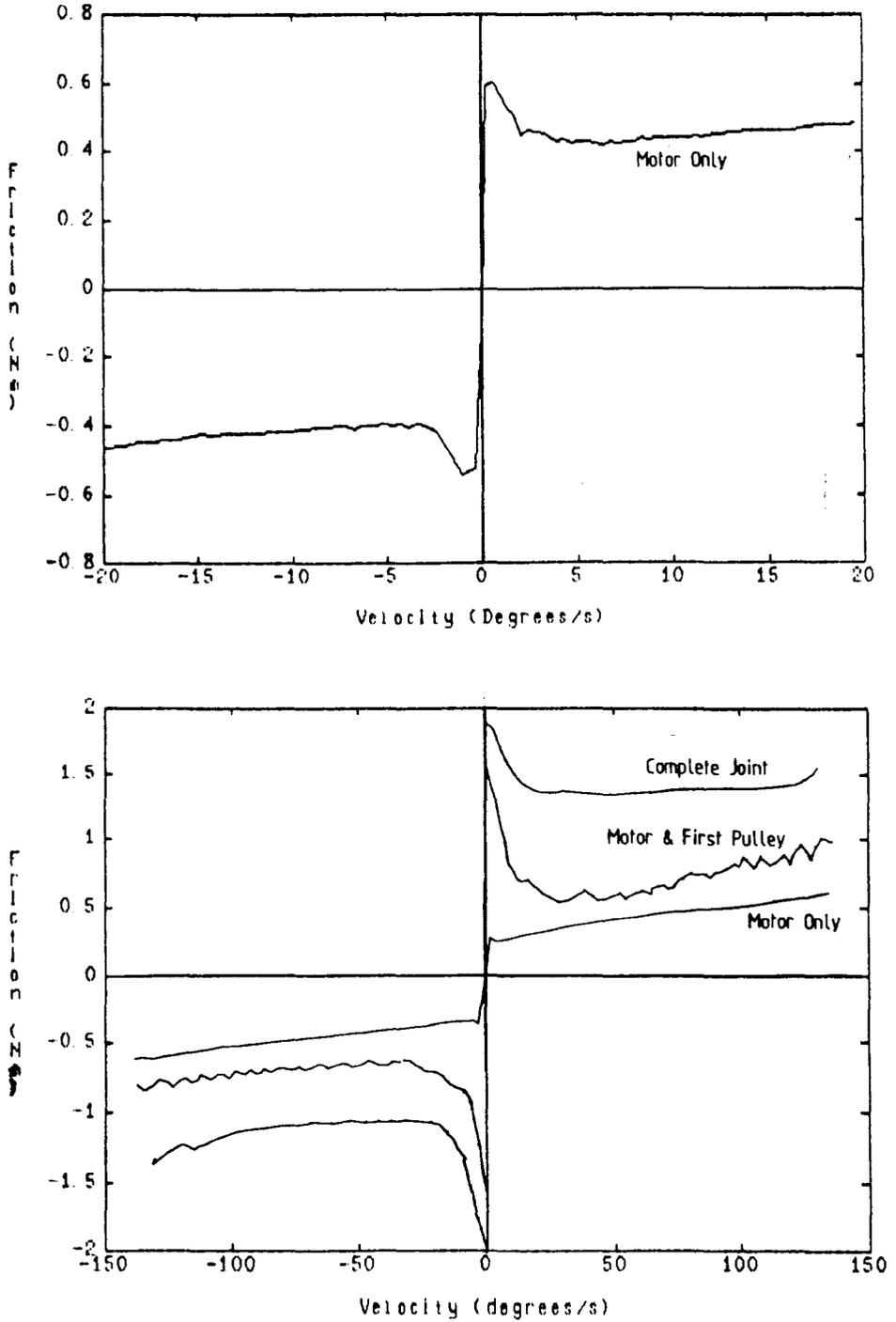
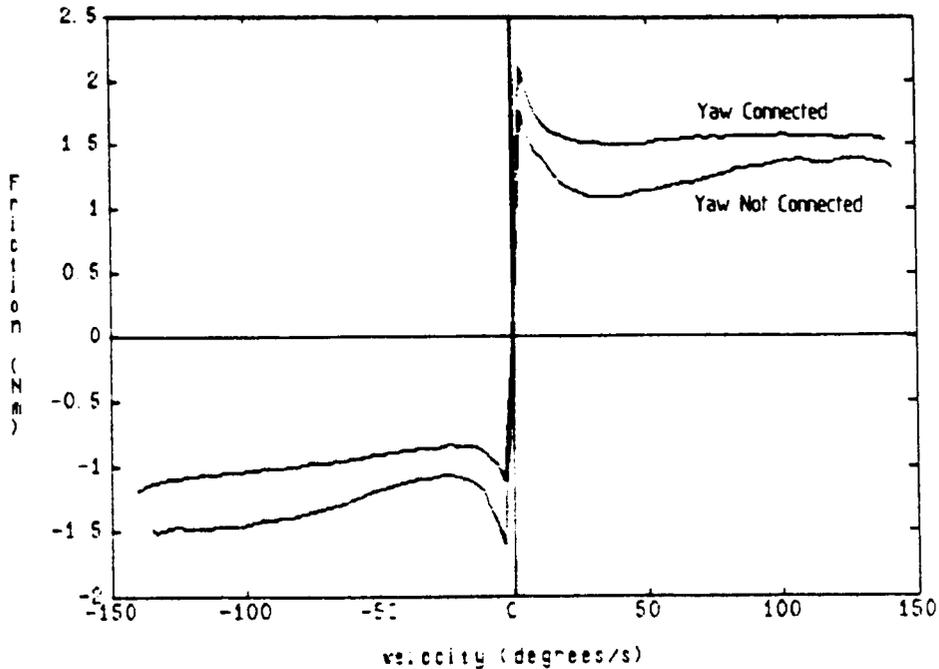


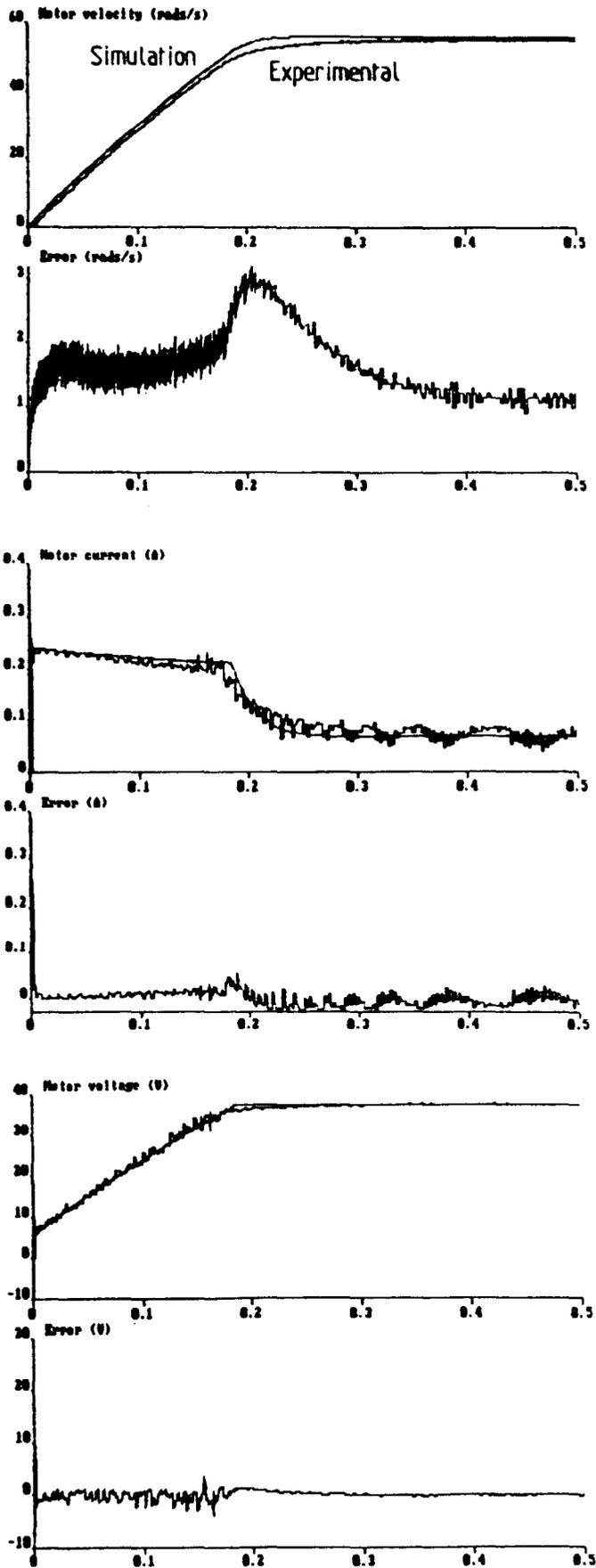
Figure 4.16 Dependence of elbow friction on presence of yaw axis drive



It was expected that some variation in joint friction would be found with changing arm configurations and end effector loads, due to the changing bearing loads. It was however found from experiments that a maximum change of 10% occurred over a wide range of load and position conditions. The limited size of this variation was probably due to the fact that a large proportion of joint friction is found in the motor and first pulley, compared with that in the joint itself, and is therefore independent of the joint bearing load. In addition the joint axes use ball bearings which have a frictional torque characteristic which is less dependant on load than that in simple journal bearings.

Before proceeding to find a dynamic model of a complete link of the RTX it is worthwhile comparing the experimental and simulated open loop behaviour of the joint motors, when disconnected from the joint. This provides a check on the parameter estimates before the complication of drive mechanism flexibility is included. Accordingly a number of simulations were performed using both the CACSD suite described in chapter 5 and the SIMNON, nonlinear simulation package [Simnon,86]. Such a comparison for a step input signal of 10V, using SIMNON to generate the model response, is shown in figure 4.17.

Figure 4.17 Open loop step response of RTX motor and current drive



There is close agreement between the experimental and simulated response for most of the test. The difference which does occur is around the point where the motor voltage approaches the limiting value of 37V, imposed by the power supply. It was believed that this difference was due to the loading of the power supply which caused the voltage to drop, thus limiting the motor speed. As the motor current dropped the supply voltage would gradually increase, allowing a gradual increase in motor velocity. This explanation was investigated experimentally and it was found that the supply voltage dropped from 37V to typically 35V when the full current was drawn and then increased again as the current dropped, in time with the increase in motor velocity. This problem could be overcome by the use of a voltage regulating circuit in the power supply. This would have a high power dissipation and would add considerably to the cost of the drive system. This was not considered worthwhile since the problem does not significantly affect system performance and only poses a problem in this model validation work. With the exception of this difference it is seen that the model derived corresponds well with the experimental results.

4.2.3 Joint transmission dynamics

The belt drive transmission system adopted in the RTX uses two types of belt, the first obeys Hooke's law and has a modulus of elasticity which may be obtained from stress - strain measurements, while the second may be approximated by a model of the form given in figure 4.7. The individual parameters in this model may be derived from a plot of displacement against time for a fixed applied load, as shown in figure 4.8. The equipment available did not allow this type of measurement; it could however give a plot of force against time for a fixed displacement change. It is straightforward to obtain the parameters required from a plot of this type, as shown in figure 4.18 which illustrates the changing forces generated in a 100mm length of fabric reinforced belt for a 1mm length change. By fitting a curve of the correct form to the

data shown in figure 4.18 we obtain the parameter values given in table 4.2. As expected, η_3 is infinite, indicating that the viscous component of extension is zero.

Figure 4.18 Variation in force for fixed belt extension

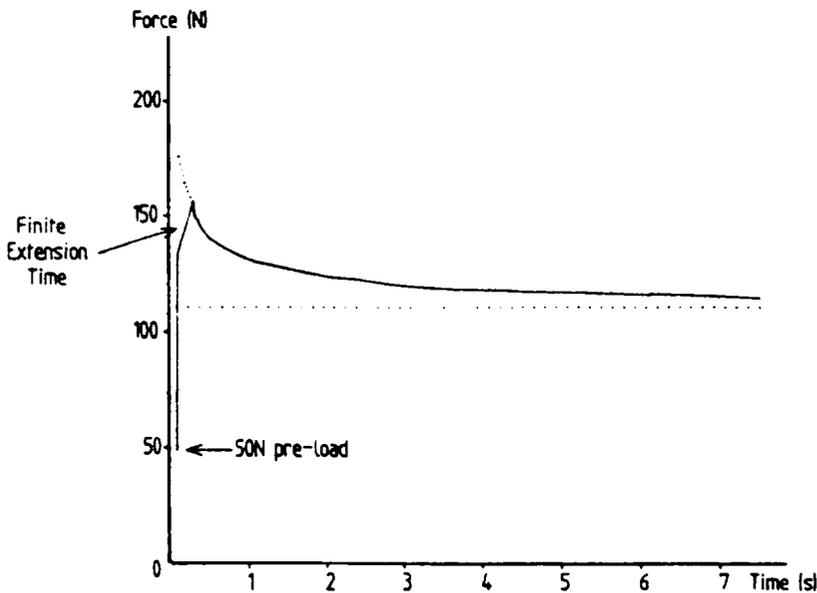


Table 4.2 Parameters of viscoelastic belt model

$$G_1 - 286 \text{ N/\%}$$

$$G_2 - 189 \text{ N/\%}$$

$$\eta_2 - 240 \text{ N/\%s}$$

$$\eta_3 - \infty$$

The elasticity of the second belt was found to be approximately half that in the first and so the extension in the first belt may be neglected when the gearing effect is taken into account. The overall behaviour is thus found from the length of belt between the last two pulleys and the radius of the last pulley; the parameters in this relationship are given in table 4.3. These values were validated by measuring the difference in angle between motor and joint, measured using the joint mounted encoders, during a movement.

The remaining parameters required in formulating dynamic models of the

individual links are the inertias of each link and the effective inertia of links further down the kinematic chain. The method used to measure inertia is described in section 4.2.4 along with the values found. For the single link models we will assume that only one link is able to move, the inertia being determined from the inertia of the link itself and the positions of subsequent links in the chain.

We may now construct a dynamic model of a general link, with the parameters dependant on which of the links is considered. This general model is given in figure 4.10 while the parameter values for the shoulder, elbow and yaw joints are given in tables 4.2 and 4.3. For the inertias, a range of values are given corresponding to the maximum and minimum values for all positions of the unloaded arm. The actual value for a given arm configuration may be found using the equations given in section 4.2.4.

A number of simulations were carried out using the model given in figure 4.10 and the parameters given in tables 4.1 and 4.3 and the results compared with those found from experiments performed under the same conditions. These verified that the model predicted the measured system behaviour with reasonable accuracy, although it did not have all the characteristics of the experimental results.

One difficulty encountered in deriving these results was determining the exact initial conditions. Since there may be a difference between the motor and joint position due to the static friction in each bearing and the elasticity and backlash in the drive mechanism; and with the limited initialisation possible, the precise joint position is not known. This discrepancy between motor and joint position may be greatly reduced by the application of a dither signal, of suitable amplitude and frequency, to the motor to overcome the static friction and so bring the two parts into closer alignment. The particular dither frequency and more especially amplitude used must be precisely chosen so as to overcome the static friction without causing significant joint movement. This is made more difficult by the drive elasticity which absorbs much of the dither

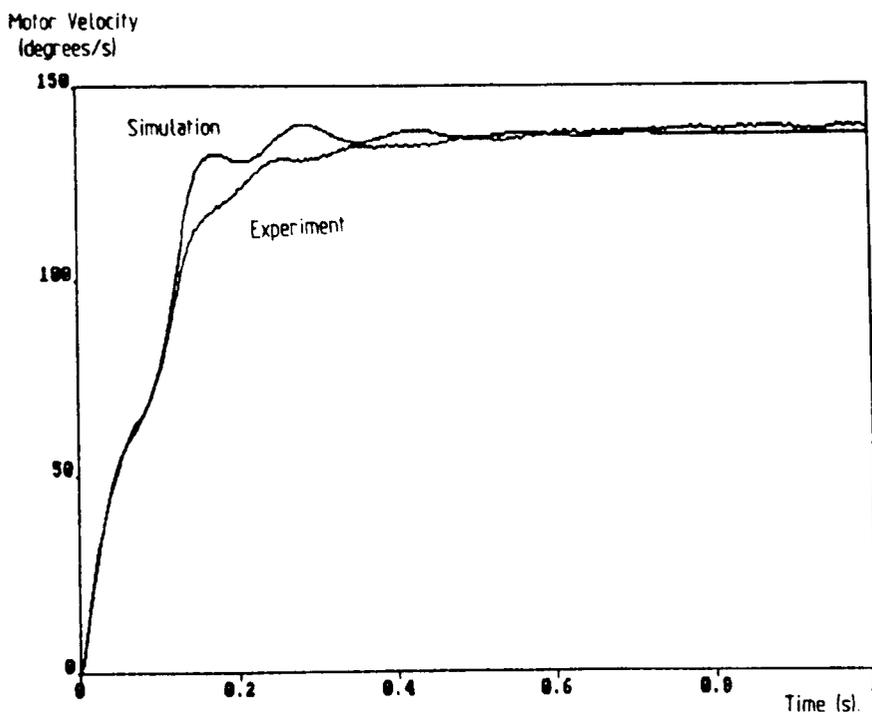
oscillation applied. It was found that the dither signal which gave the most movement towards the aligned position, for the elbow joint, had a frequency of 10Hz and an amplitude of 4Nm. Similar results were found for the remaining joints.

As an example of simulated and experimental results, Figure 4.19 shows the open loop velocity response of the elbow motor, with the gripper removed from the joint, to an input step change of 10V. The velocities indicated correspond to the motor but are scaled to give the equivalent arm velocities. It is seen that there is a close agreement between simulated and measured response, except in the region in which power supply loading occurs, as explained in section 4.2.2.

Table 4.3 Model parameters for RTX links

<u>Parameter</u>	<u>Shoulder</u>	<u>Elbow</u>	<u>Yaw</u>	<u>Units</u>
Current drive and motor parameters	As Table 4.1			
Belt elastic spring constant, K_{S1}	0.0018	0.008	0.012	Nm/Rad
Belt anelastic spring constant, K_{S2}	0.0028	0.012	0.018	Nm/Rad
Belt time constant, τ_b	1.27	1.27	1.27	s
Link inertia, J_1	96 - 556	99 - 120	22.1	$\times 10^{-3} \text{Kgm}^2$

Figure 4.19 Measured and simulated velocity response of elbow joint to a step input



4.2.4 Measurement of Joint Interaction

In section 4.1.4, it was shown that the interaction dynamics of a robot arm may be equated to those of a series of links with end point masses independent of arm position, as in figure 4.11. These masses cannot be found by direct measurement and must be calculated from the link masses and inertias. Deriving these values can be approached either by considering the components of which the arm is made and adding their individual masses and inertias; or by measuring the total mass and inertia of a complete link.

The manufacturers have provided mass and inertia data derived by the first method, with some experimentally measured values [UMI]. These values were found for an early version of the arm and since then alterations have been made to some parts. These changes cannot be allowed for reliably since the manufacturers found significant differences between calculated and measured values. Alterations have also been made, by the author, to the particular arm

considered here, including the addition of extra sensors and removal of the original gripper. It was therefore decided to measure the true inertias of the arm upon which the control designs were to be tested.

There are several ways of measuring the inertia of an object, the method most suitable depending, to a large extent, on the nature of the object. The method adopted here for measuring link inertias may be divided into two parts; firstly, the centre of mass must be found by suspending the link from two fixed points, with the supporting force at each point measured. From these forces the position of the centre of mass may be found, relative to the suspension positions. If the arm is then allowed to swing about a fixed pivot point, under the influence of gravity, the moment of inertia about the pivot may be calculated from the oscillation frequency and the distance of the pivot from the centre of mass. The inertia about the centre of mass may then be calculated using the parallel axis theorem. It was found to be worthwhile repeating this procedure using a number of pivot points to give an indication of the reliability of the results obtained.

A further method was adopted to measure both the centre of mass and inertia. This consisted of suspending the link from two fixed points and measuring the two oscillation frequencies. From these frequencies, both the center of mass and the inertia may be calculated. This method was only used for verification since the calculated results are relatively sensitive to frequency measurement inaccuracies.

Using the earlier method the inertias measured were consistent over a number of tests and were approximately in agreement with the estimates derived from the manufacturer's data. Repeating this procedure for each of the links considered here, the values given in table 4.4 were found.

Table 4.4 Link masses and inertias

Link	Effective inertia about joint axis (kgm ²)	Effective mass on axis (kg)	Distance of centre of mass from axis (mm)
Wrist ¹	0.00225	1.45	21
Lower arm	0.0277	1.55	151
Upper arm	0.0637	2.24	136

¹ Wrist unit does not include the gripper and has the gripper mounting plate in the initialisation position.

Using the data given in table 4.3 the parameters in the interaction dynamic equation for the shoulder and elbow joints (equation 4.1) may be completed and we obtain:

$$m_1 = 0.99\text{Kg}$$

$$m_2 = 3.0\text{Kg}$$

$$l_1 = 0.254\text{m}$$

$$l_2 = 0.151\text{m}$$

It is assumed that there is no load at the end effector. If this is not the case the values of m_2 and l_2 must be adjusted according to the equations:

$$m_2 = 3.0 + m_l$$

$$l_2^2 = (0.068 + J_l)/m_2$$

where m_l is the mass of the load and J_l is the inertia of the load with respect to the elbow axis. Changes in yaw angle have a minor effect on the interaction

model and these may be treated in a similar manner to changes in load if the yaw is fixed. The behaviour of the yaw axis was not included in the model since the effects of this joint are small compared to those caused by movement in the elbow and shoulder axes and even less significant when compared with the effects of drive system nonlinearities.

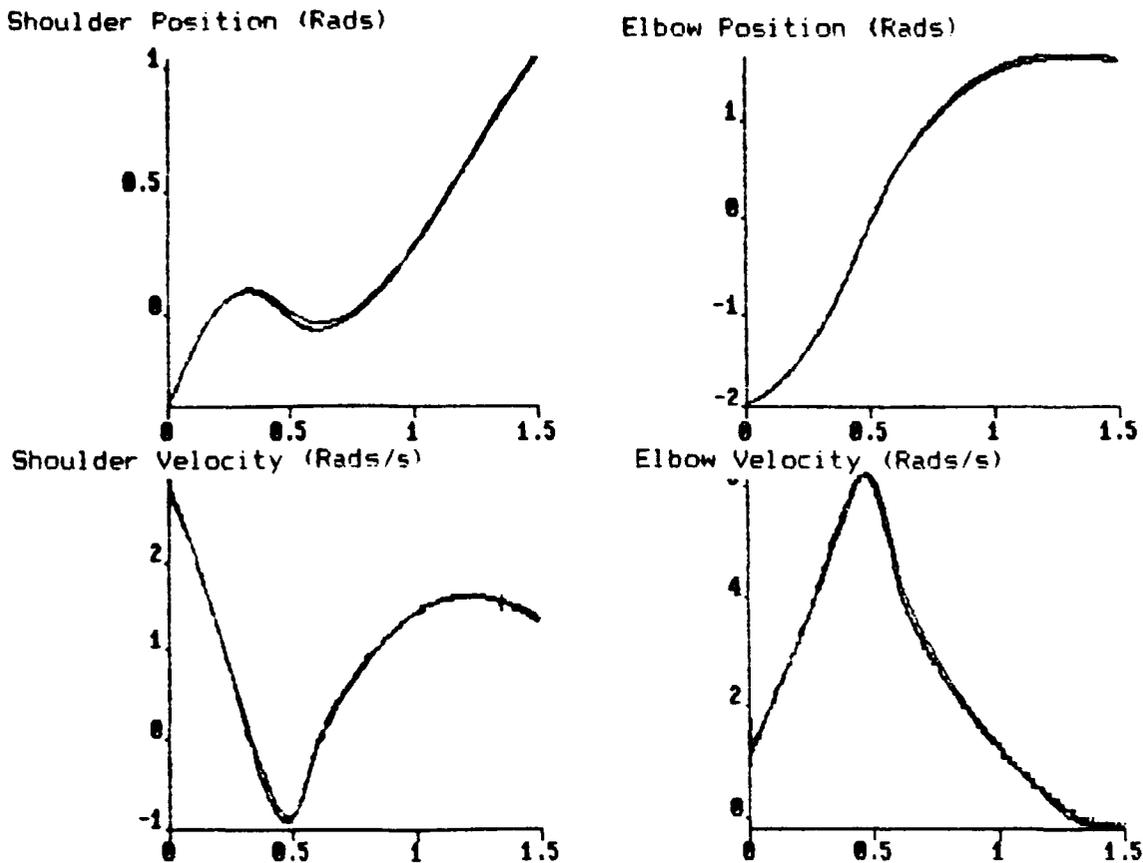
To verify the interaction dynamic equations and the derived parameters, a number of experiments were performed on the elbow and shoulder joint of the RTX. These experiments involved removing the drive belts from the axes of interest so that the joints were free to move with a minimum of friction, allowing the normally relatively small interaction effects to become significant. There remained some effects from the joint friction and so approximate level of friction torques were found by measuring the joint acceleration with the other joints fixed and no applied torques. The experiments involved giving one joint an initial velocity 'by hand' and monitoring the subsequent motion. It was found that, with the belts removed, the interaction effects became significant. By comparing the trajectories measured with those found from simulation the parameters in the interaction model were verified. As an example of such tests the results given in figure 4.20 show the simulated and measured positions and velocities of the shoulder and elbow joints in free motion, having been given initial velocities. It will be seen that there is a close correspondence between the results.

The interaction and drive system nonlinearity models may be combined by introducing the effects of interaction into the single link model. Since each of the links has essentially the same dynamic model, with only the parameters changing, a diagram of the complete system model would be largely repetitious and so will not be given.

As an indication of the relative significance of interaction forces and drive system nonlinearities, consider the centripetal and coriolis forces acting on the shoulder and elbow joints of the RTX when moving at the maximum velocity attainable with the enhanced hardware. For the shoulder joint, it can be seen

from section 4.1.4 that the joint torque resulting from centripetal effects is given by $m_2 l_1 l_2 \dot{\theta}_2^2 \sin(\theta_2)$ while the coriolis effect is $2m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_2)$. Given that the maximum joint velocities are $\dot{\theta}_{1,max}=1.22\text{rads/s}$ and $\dot{\theta}_{2,max}=2.44\text{rads/s}$, the maximum centripetal and coriolis effects cause a joint torque of 0.68Nm each, or a total of 1.37Nm. The centripetal force on the elbow joint under the same conditions is 0.17Nm. These figures are for an unloaded robot but the effect of an added load would not be large. These values should be compared with the joint friction torques which are, on average, about 3.5Nm and 1.4Nm for the shoulder and elbow joints respectively while the maximum motor torques are 10.6Nm and 5.3Nm respectively.

Figure 4.20 Simulated and experimental responses of the shoulder and elbow joints in free motion



4.3 Limitations of the Dynamic Model

The model of the RTX robot derived in the preceding sections is highly complex and although it gives a reasonable accurate representation of the true system it does not include all facets. As stated at the beginning of this chapter there is virtually no limit to the possible detail that can be included in the model of a real dynamic system and a point is reached at which the model is judged to be sufficiently accurate for its intended purposes.

The most significant effect which is not included in the model derived above is backlash in the timing belt transmission system. The form of this backlash is highly complex involving the nonlinear frictional forces between the belt and the pulley, the flexibility in the belt teeth, the belt speed and the transmitted force. In addition the behaviour is significantly affected by unpredictable influences, such as mechanical vibration, which cannot easily be measured. For this reason a reliable model of the backlash behaviour cannot be derived.

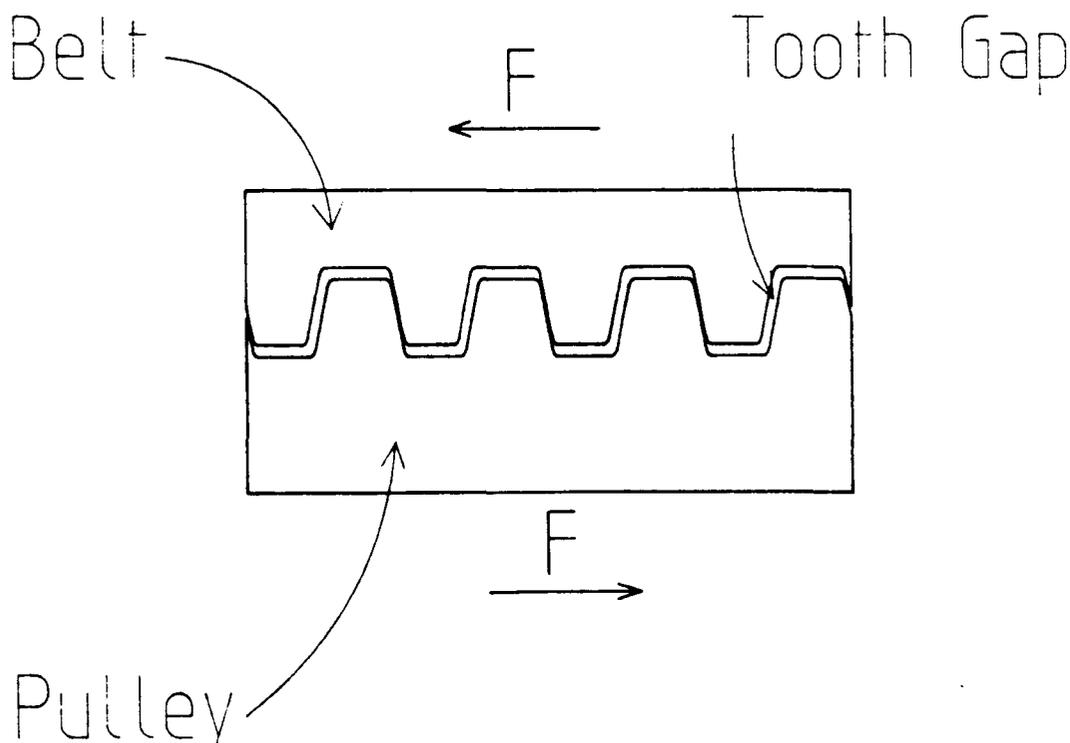
Since, in the standard RTX, the true arm position is not measured and the backlash effects cannot be allowed for in the control system, it is worthwhile considering methods which might be adopted to reduce the effect.

The teeth on the drive belt are trapezoidal in cross section with corresponding recesses in the pulley. In order to allow for flexing of the belt to fit around small pulley sizes, some gap must be left between belt and pulley teeth when using larger radius pulleys. The typical situation is shown in figure 4.21.

A number of methods may be adopted to improve the backlash behaviour. The most obvious is to reduce the tooth gap by using a closer tolerance belt. This cannot be accomplished satisfactorily when a large gear ratio is required because of the different tooth size required at each pulley. Alternatively the initial belt tension may be increased and so increase the force holding the belt to the pulley, thus increasing the friction between belt and pulley. Unfortunately

this also increases the load on the bearings and hence the joint friction.

Figure 4.21 Meshing of toothed belt and pulley showing tooth gap



The most effective solution to this problem would be to adopt a different design of tooth which avoids the tooth gap found with trapezoidal teeth. One such design involves the use of semi-circular cross section teeth which automatically mesh sufficiently to remove the tooth gap and so eliminate backlash.

With the existing belt drive system using trapezoidal teeth the tooth gap is typically 0.25mm which corresponds to an angle of 0.29° in the shoulder joint or 0.64° in the elbow joint. These unknown offsets correspond to an end point position error of approximately 2.5mm. Slipping between the belt and pulley typically occurs at around 75% of the maximum motor torque and so any attempt to overcome backlash effects by the use of a dither signal would require a signal of this amplitude. Owing to their elasticity, high frequency torques are largely absorbed in the belts and so a low frequency dither signal would be required but this in turn would excite arm movement and so cannot be used.

Other effects are present in the real system, including flexing in the Z-column and carriage and flexing of the links, but the time required to measure them is not justified by their relative importance.

4.4 Conclusions

The system model derived in the previous sections predicts behaviour close to that found in practice, given the limited knowledge of the test conditions. It is clear that a model of the size of that derived here is too large to be used in the design of control systems, hence the full model was only used to simulate behaviour in the final stages of algorithm testing. For the purposes of controller design a number of simplifications must be made to the model. The validity of reducing the model complexity depends to a large extent on the type of controller being considered. Similarly the effect of errors in parameter estimates depends on the nature of the controller and this provides a useful test of control system robustness which would not be possible for a robot with more easily modeled dynamics. The system model used in the development of each of the control schemes considered will be given with the description of the control algorithm, in chapter 6.

CHAPTER 5

A Computer Aided Design Suite For the Design of Nonlinear Controllers For Nonlinear Systems

In this chapter a Computer Aided Control System Design (CACSD) Suite for the analysis and design of linear and nonlinear controllers for nonlinear systems will be described. The design suite is part of the Hull University Computer Aided Design Suite [Taylor,84] which is based on the UMIST CACSD suite and complements other packages already available for control system analysis and design. The package described here allows the analysis of nonlinear systems with linear and nonlinear controllers, in phase space and by time responses.

The existing package will be described first, followed by details of a number of minor alterations made to improve the general utility of the program. The model structure used in the package has been augmented to make it more suitable for the design of controllers for robotic systems and these changes are expounded in section 5.3. The algorithm used for solving differential equations implemented in this program was developed by Jobling [Jobling,84] specifically for the generation of phase trajectories and although it appeared to perform well it had not been thoroughly tested and compared with other, better established algorithms. Tests were performed to compare this algorithm with a number of other methods and the results of these are described in section 5.4. The overall performance of the enhanced CACSD suite, when applied to problems of nonlinear control for robotic applications, is evaluated in section 5.5.

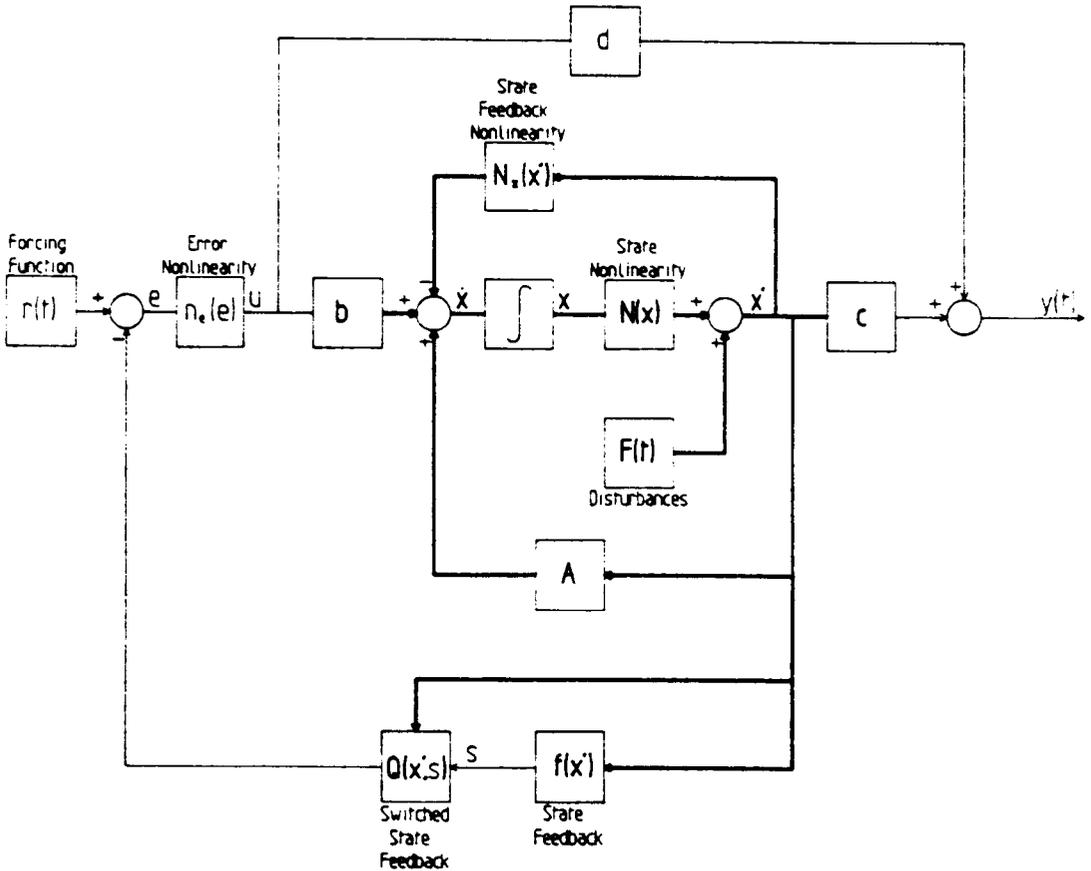
5.1 Existing Computer Aided Design Suite

The existing package, generally known as PHAS, was developed by Jobling [Jobling,84] as part of a doctoral research project. The aim of this work was to develop routines to aid in the evaluation of nonlinear controllers for linear and nonlinear plants, the design work to be carried out using phase plane and time response techniques. It was also intended that this package should become part of the Hull CACSD suite, requiring that, as far as possible, system information should be interchangeable between the different packages within the suite. The package is presently run on a MicroVAX system under the VMS operating system, although much of the software was originally written in FORTRAN 4 on a PDP 10 or PDP 11. The software has since been converted to DEC FORTRAN which was also used in the development of the software described here.

The structure of the dynamic model is based on the single input single output, state space form, augmented with a number of nonlinear elements as shown in figure 5.1. The familiar A,b,c and d matrices used may be obtained from other system descriptions using other parts of the design suite.

Nonlinear systems may be represented using the error nonlinearity, $n(e)$, state nonlinearity matrix, $N(x)$, and nonlinear state feedback matrix $N_x(x)$. The error nonlinearity, $n(e)$ is a scalar function, $N(x)$ is a diagonal matrix while any elements of the $N_x(x)$ matrix may be defined. Each element in these nonlinear blocks may consist of one of a wide variety of functions selected from a series of menus, including linear, piecewise linear and nonlinear forms, in addition to null, constant and functions of time. If the required function cannot be found within these menus then it is possible to specify sets of points for piecewise linear or polynomial fits.

Figure 5.1 Original model structure used in CACSD suite



To facilitate the design of linear and nonlinear controllers blocks $f(x)$ and $Q(s)$ are included. The first of these allows any linear combination of functions of a single states to be fed back to the system input or, for switching controllers, into the switching block, $Q(s)$, which then provides switched state feedback.

An input to the system may be provided using $r(t)$ while measurement disturbance may be added with the $F(t)$ block. Both these elements allow a wide range of functions of time to be selected from a menu, including step, ramp and sinusoidal types.

Data entry is carried out in an interactive manner using many of the same routines used in other parts of the suite, with default values available at most points. The default system is linear, autonomous and time invariant. Once entered the system data is stored on disc, for later retrieval and modification.

The algorithm used in the generation of state trajectories was developed primarily for design in the phase plane and is described in detail in section 5.4.3. The parameters required for the algorithm to operate are entered at the setting up stage and are stored on disk so that the same conditions may easily be used in a series of simulations. This involves entering upper and lower limits for each of the states and a time limit. A set of initial conditions may be entered or, for second order systems a number of initial conditions may be automatically generated at fixed spacing in the phase plane. The maximum and minimum search radii and an accuracy tolerance coefficient are required and these may be either explicitly entered or suitable values generated from the limits set on the state variables.

If the state trajectory leaves the area of interest or the time variable exceeds the set limit then the simulation is halted. It is also stopped if a singular point is encountered or if a numerical error occurs in any of the routines. Each of these conditions is indicated by an appropriate message and the program proceeds with the next set of initial conditions until all trajectories have been calculated at which point the plotting programs are called.

Once a simulation is complete the resulting states may be plotted against time, in pairs to give phase-plane trajectories or in groups of three using one of a number of transforms, giving for example an isometric projection. All system files, including state trajectory data may be saved in a user readable form on disc or printed out for closer inspection.

5.2 Miscellaneous Improvements to the CACSD Suite

A number of developments have been made to the PHAS package to increase the ease of use and to improve the speed with which results are generated. These have been made possible by improvements in the computer hardware (using VAX and microVAX in place of a PDP11). It is now possible to consider more complex systems, including robot manipulators, which required

unrealistic computation time on earlier computers. One of the major factors which has allowed greater speed is the introduction of a virtual filing system developed by [Sawyer,86]. This alleviates the excessive disc operations required on computers with less main memory. This makes data entry faster and increases the speed of computation for a typical simulation by a factor of ten.

With these improvements other limitations in the existing system became more apparent in the work on controller design for robot manipulators. The first of these to be rectified was the limitation of the number of system states to five. This represented the limit of what would allow practical calculation times. This limit was increased to twenty, this being considered the maximum which could be assimilated by the operator. With this increase it became necessary to allow data within matrices to be modified element by element. This had not been necessary earlier since the entry of a five by five matrix is not too arduous. This improvement involved a simple alteration to the linear data entry routines although those for nonlinear data required more adaptation.

5.3 Enhancement of the CACSD Model Structure

A major alteration to the package was made necessary by the nature of the dynamics of a robot manipulator. In order to study these dynamics accurately some changes must be made to the structure of the system model used.

It is shown in section 4.1.4 that the link interaction dynamics of a two link planar manipulator are given by a set of highly nonlinear equations with terms having the form:

$$\frac{(a + b \sin(q_2)) (\dot{q}_1 + \dot{q}_2)^2}{c + d \sin^2(q_2)}$$

where q_1 and q_2 are the two joint angles and a, b, c and d are constants for a given manipulator with a fixed load. In some cases, the numerator will not include both \dot{q}_1 and \dot{q}_2 but this form covers most cases.

often be the case that a system will have fewer nonlinear feedback paths than the system order, allowing a saving in computation time over a system represented as having an equal number of linear and nonlinear feedback paths. In addition, by introducing two or more parallel feedback paths using L , N_x and M , it is possible to introduce nonlinear behaviour made up of a sum of standard forms. The L and M matrices are entered and stored in the same manner as the other linear data elements and have default values which give a system with no nonlinear feedback.

The types of function required in $N_x(x)$ are not presently available in the selection menus as they are specific to this type of problem and have a more complex structure than the functions available. The inclusion of this type of function in the menus was considered but to be sufficiently general it would require a large amount of extra software and would be unwieldy to use. The functions involved are, however, well behaved over the range of parameters found in real systems and so it is possible to obtain a good approximation to the true function using piecewise linear or polynomial approximations. The disadvantage of this method is that for a single change in parameter the complete set of piecewise linear or polynomial data must be recalculated. This problem was partially eased by the use of programs to generate the appropriate data for each nonlinear term in N_x , given a set of physical parameters. The resulting data points may be incorporated into a file which is then used to emulate the normal keyboard input. Further details of this program are given in appendix 5.

5.4 Comparison of Numerical Algorithms for the Solution of Nonlinear Ordinary Differential Equations

Since the solution to differential equations is not, in many cases possible by analytical means, some approximate method must be employed. Generally this involves the use of numerical algorithms which take a series of steps,

making an approximation to the solution at each step. The methods available may generally be classified as single-step or multi-step algorithms. In the former, approximation of the value of the solution at the next time increment is made only on the basis of a single previous value. Multi-step algorithms on the other hand use two or more previously calculated values to approximate the new value of the solution. The PHAS algorithm, designed specifically for the generation of phase trajectories will also be investigated. Each of these categories will be briefly discussed and their relative merits considered. Throughout the initial description, it will be assumed that the step length in the approximation remains constant. This need not be the case and procedures for varying the step length will be considered later.

Although many of the algorithms considered here are widely used and their relative merits investigated for linear, and a subset of nonlinear problems [Hull,72] they have not previously been tested on the types of nonlinear equations found in discontinuous control systems. The PHAS algorithm has not been thoroughly tested and so this work provides new information on the performance of this algorithm compared to more common techniques for both linear and nonlinear equations, as well as giving an assessment of the performance of other algorithms for the type of problems found in nonlinear control.

Before considering the solution methods a few preliminary details must be given. A scalar problem will be considered here, although the extension to the vector case is a simple one which will be detailed later. The differential equation considered is of the form:

$$\frac{dx}{dt} = f(x, t)$$

with initial conditions given by:

$$x(t_0) = x_0$$

with the solution required over the interval:

$$t \in [t_0, t_f]$$

We thus require a value for the solution at times $t_0, t_0 + h, t_0 + 2h, \dots$ and so, if we have the value x_i at time t_i we then seek the value x_{i+1} at time t_{i+1} .

5.4.1 Single-step Algorithms

The most basic, single step method, Euler's method [Fröberg,69], although it is rarely used in practice does provide a conceptually simple starting point. Applied to the differential equation described above the approximation for x_{i+1} is given by:

$$x_{i+1} = x_i + hf(x_i, t_i)$$

It will be seen that this approximation assumes that the value of $f(x,t)$ remains constant over the interval $t \in [t_i, t_{i+1}]$. Clearly this is not, generally the case in practical problems. If however, the step size, h is reduced the approximation improves. At each step the error introduced using Euler's method is of the order of h .

The aim of more sophisticated single step algorithms is to obtain a better approximation to the behaviour of $f(x,t)$ over $t \in [t_i, t_{i+1}]$ and so increase the order of the single step error. The first method is to consider the area under the curve $f(x,t)$ over this time interval as a trapezium, rather than a rectangle, as in the Euler method. Before this area can be found, we must obtain an approximation for x_{i+1} using Euler's method. Denoting this estimate k , the resulting equation for x_{i+1} then becomes:

$$x_{i+1} = x_i + \frac{1}{2} h [f(x_i, t_i) + f(k, t_{i+1})]$$

where

$$k = x_i + h f(x_i, t_i)$$

This algorithm is known as the Improved or Modified Euler approximation [Morris,74] and gives a single step error of $O(h^2)$.

The Improved Euler approximation uses evaluations of the right-hand side of the differential equation at time, t_i and an estimate of the value at t_{i+1} . A logical extension to this method is to make estimates of the right-hand side at other points in the interval $t \in [t_i, t_{i+1}]$. If estimates are also made at $t = t_i + \frac{1}{2} h$, then we may obtain the fourth order Runge-Kutta algorithm [Johnson,82], in which the value of x is updated according to:

$$x_{i+1} = x_i + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4] \quad (5.1)$$

where

$$k_1 = h f(x_i, t_i)$$

$$k_2 = h f(x_i + \frac{1}{2} k_1, t_i + \frac{1}{2} h)$$

$$k_3 = h f(x_i + \frac{1}{2} k_2, t_i + \frac{1}{2} h)$$

$$k_4 = h f(x_i + k_3, t_i + h)$$

This algorithm, which gives a single step error of $O(h^4)$, is sufficiently accurate for many applications and is widely used in practice. Each step does require four evaluations of the right-hand side of the differential equation and so can be computationally expensive.

5.4.1.1 Variable Step Size Runge-Kutta Algorithm

The choice of the most appropriate step size in a numerical algorithm of this kind is a difficult one, depending on the type of system considered and the accuracy required. The precision attained must be balanced against the time required for computation. In the methods described above there is no indication of the accuracy obtained and so it is natural to err on the side of caution and use an unnecessarily small step size. Even so, erroneous results may be generated for certain types of equation which are not particularly amenable to numerical solution, such as stiff systems of equations. It is therefore desirable that some indication of accuracy be obtained at each step of the algorithm. An obvious way to do this is to perform an iteration using the standard step size, h and compare the result with that obtained from two iterations with a step size of $h/2$. This gives a good indication of the accuracy but requires 11 function evaluations at each step for a fourth order Runge-Kutta algorithm.

A preferable method uses the fact that the local truncation error, τ for a k^{th} order Runge-Kutta method is given by $\tau = O(h^k)$. By taking the difference between the results of say a fourth and a fifth order Runge-Kutta algorithm with the same step size it is possible to find an approximation to the local truncation error in the fourth order result. Once such an error estimate is available it is natural to develop some means of altering the step length to suit the particular situation and so limit the error to some preset maximum. In this way a step length should be reached which is 'optimal', i.e. the largest step which keeps the local error below some specified value. By using certain pairs of approximations with some common time increments it is possible to reduce the number of function evaluations below that required when using two half steps. One popular combination was developed by Fehlberg [Johnson,82] and uses the equations:

$$x_{i+1} = x_i + h \left\{ \frac{25}{216} k_1 + \frac{1408}{2565} k_2 + \frac{2197}{4104} k_3 - \frac{1}{3} k_4 \right\}$$

$$\tilde{x}_{i+1} = x_i + h \left\{ \frac{16}{135} k_1 + \frac{6656}{12825} k_2 + \frac{28561}{56430} k_3 - \frac{9}{30} k_4 + \frac{2}{33} k_5 \right\}$$

(5.2)

The first of these equations is the fourth order approximation while the second is fifth order. The values of k are calculated according to:

$$k_1 = f(x_i, t)$$

$$k_2 = f \left\{ x_i + \frac{h}{4} k_1, t + \frac{h}{4} \right\}$$

$$k_3 = f \left\{ x_i + h \left\{ \frac{3}{32} k_1 + \frac{9}{32} k_2 \right\}, t + \frac{3h}{8} \right\}$$

$$k_4 = f \left\{ x_i + h \left\{ \frac{1932}{2197} k_1 - \frac{7200}{2197} k_2 + \frac{7296}{2197} k_3 \right\}, t + \frac{12h}{13} \right\}$$

$$k_5 = f \left\{ x_i + h \left\{ \frac{439}{216} k_1 - 8k_2 + \frac{3680}{513} k_3 - \frac{845}{4104} k_4 \right\}, t + h \right\}$$

$$k_6 = f \left\{ x_i + h \left\{ -\frac{8}{27} k_1 + 2k_2 - \frac{3544}{2565} k_3 + \frac{1859}{4104} k_4 - \frac{11}{40} k_5 \right\}, t + \frac{h}{2} \right\}$$

This pair of formulae requires only six function evaluations compared with ten if an unrelated pair of fourth and fifth order equations were used. An estimate of the local truncation error is given by $\tilde{x}_{i+1} - x_{i+1}$, from which we may define:

$$e = \frac{\tilde{x}_{i+1} - x_{i+1}}{h}$$

We then wish to select h so that:

$$|e| \approx \epsilon \frac{|x_i|}{t_f - t_0}$$

where ϵ is maximum relative error acceptable and $[t_0, t_f]$ is the time interval over which the solution is sought. We may call h 'optimal' if this condition is

satisfied. If this is not the case then we may use e to make a new guess at the optimal by noting that

$$e = \frac{\tilde{x}_{j+1} - x_{j+1}}{h} = M h^4 + O(h^5)$$

for a constant M . If we define a new step size γh which is the optimal size then, since e is proportional to h^4 ,

From this we obtain:

$$\gamma^4 |e| \approx \epsilon \frac{|x_j|}{t_f - t_0}$$

$$\gamma \approx \left[\frac{\epsilon |x_j|}{|e| (t_f - t_0)} \right]^{\frac{1}{4}}$$

Thus if we have taken a step of length h we may calculate e and can say that the step length should have been γh . If γ is greater than unity then we can say that the step was successful and we will use a step of say $h_1 = 0.8\gamma h$, where the factor of 0.8 is incorporated to avoid a value of h_1 which is just greater than the optimal since this would result in the error condition not being fulfilled at the next step. If the value of γ is less than unity then the step must be rejected and another step tried with a length calculated as above. To avoid very large changes in step size an upper and lower limit must be set on γ , typically limiting the ratio of successive values of h to a factor of five.

Problems occur with this form of update as $|x_j|$ and hence γ approach zero. To overcome this a mixed, absolute/relative update may be used so that the correction equation becomes:

$$\gamma = \left[\frac{\epsilon_r |x_j| + \epsilon_a}{|e| (t_f - t_0)} \right]$$

where ϵ_a and ϵ_r are the absolute and relative error tolerance respectively.

This form of algorithm may easily be modified for systems of first order equations in which case the Runge-Kutta equations take a vector form, as do

the error estimates with the smallest value of γ used to update the value of h , so that the variable with the largest error controls the step size. In this case the absolute error tolerance coefficient, ϵ_a may be made a vector with elements appropriate to the range of values expected for a particular variable.

5.4.2 Multi-step Algorithms

In contrast with the methods described in section 5.4.1, multi step methods make use of information from earlier steps to estimate the next point. The behaviour of $f(x,t)$ in the interval $[t_i, t_{i+1}]$, and so the value of x at t_{i+1} is thus estimated from the previously computed values of x at $t_i, t_{i-1}, t_{i-2}, \dots$. A common group of multi-step algorithms are known as predictor-corrector methods since they involve the use of two equations, the predictor equation which gives an estimate of x at t_{i+1} , based on previously calculated values of x , and the corrector equation which uses past values of x and the estimate of x_{i+1} , to improve the estimate of x_{i+1} . The corrector equation may be applied in an iterative manner until the approximations to x_{i+1} , is sufficiently accurate.

A simple example of this type of method is given although, once again this would not often be used without some improvement. The prediction equation used to obtain a first estimate is Euler's equation, i.e.

$$x_{i+1}^{(0)} = x_i + h f(x_i, t_i)$$

while the n^{th} iteration towards x_{i+1} , is carried out according to:

$$x_{i+1}^{(n)} = x_i + \frac{1}{2} h [f(x_i, t_i) + f(x_{i+1}^{(n-1)}, t_{i+1})]$$

This simple example is, in fact, a single-step method since it does not involve use of x_{i-1} , and earlier estimates but it does illustrate the principle involved in multi-step, predictor-corrector methods. A more practically useful predictor-corrector algorithm is due to Adams-Moulton [Conte,72] and uses four

previous evaluations of $f(x,t)$. If we define

$$f_i = f(x_i, t_i)$$

then the prediction equation is:

$$x_{i+1}^{(0)} = x_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

while the correction formula is:

$$x_{i+1}^{(n)} = x_i + \frac{h}{24} [9f(x_{i+1}^{(n-1)}, t_{i+1}) + 19f_i - 5f_{i-1} + f_{i-2}]$$

In order to start this algorithm four values of $f(x,t)$ are required. These are generally found using the fourth order Runge-Kutta technique. It will be noted that only a single evaluation of the right-hand side of the differential equation is required at each iteration and so, provided no more than three iterations are required at each step, a computational saving will be made, compared with a fourth order Runge-Kutta algorithm. Another feature of predictor-corrector algorithms is that they allow a measure of the accuracy of the approximation to be made by comparing the results of successive iterations. It is thus possible to vary the step length in a manner similar to that described above. This change in step size does however require the evaluation of new starting points and so if the step length is repeatedly changed this may be more time consuming than a single step, variable step length algorithm. If the variation between two consecutive step lengths is limited to a ratio of say two then it may be possible to use some of the previously computed values of x but this will only partly alleviate the problem.

5.4.3 The PHAS Algorithm

The methods of solving differential equations described above all take discrete time steps and although this step size may be variable, time is treated as a reference against which other variables are computed. This generally results in a smooth time response but there are occasions when this is not the most important criterion. When investigating the behaviour of second order systems for instance, the phase plane may provide more useful information than time responses, and so the smoothness of the phase trajectories is also important.

An algorithm intended specifically for the generation of phase trajectories, although it also produces data for time responses, has been developed by Jobling [Jobling,84] as part of the Hull Computer Aided Control System Design Suite.

Consider, initially a time invariant system of first order equations given by:

$$\dot{x} = f(x, t)$$

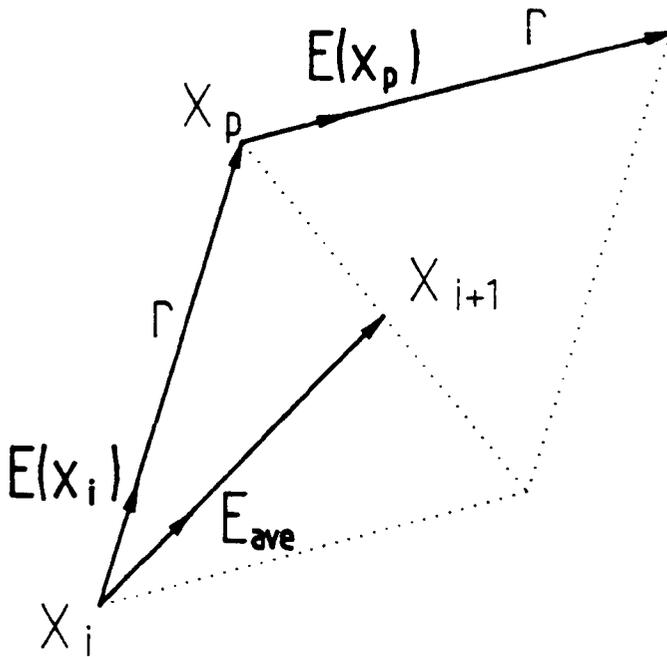
Starting at the point x_i in phase space, we may define a unit vector $E(x_i)$ in the direction of the state velocity vector at this point, $f(x_i)$ as shown in figure 5.3.

If we project from this point a distance r in the direction of $E(x_i)$, we reach a point x_p at which point the velocity vector is $f(x_p)$ with corresponding unit vector $E(x_p)$. If the two vectors $E(x_i)$ and $E(x_p)$ are aligned to within a specified tolerance then the point x_{i+1} is taken to be at point q shown in figure 5.3, which is defined by:

$$x_{i+1} = x_i + r E_{ave}$$

$$E_{ave} = (E(x_i) + E(x_p))/2$$

Figure 5.3 Generation of update in PHAS algorithm



Thus the trajectory is extended a distance r in the mean direction of the derivatives at x_i and x_p .

If the two vectors are not in sufficiently close alignment then the search radius, r is reduced by a factor of two and the procedure repeated until the condition is fulfilled or until some specified minimum radius is reached. At the next step the search radius will revert to the original, maximum value. In this way a complete phase trajectory is built up until either one of the states leaves the area of interest or the trajectory reaches a singular point, a condition which is detected by checking if the modulus of $f(x_i)$ becomes less than some specified limit, ϵ .

In addition to generating phase trajectories this algorithm may be used to produce time responses if time is considered as an extra state which has a constant derivative of unity. If this is the case the algorithm remains unchanged except that the condition for detecting a singular point must be modified to $|f(x_i)| - 1 < \epsilon$.

5.4.4 Testing of Numerical Algorithms

The choice of the most suitable algorithm for solving differential equations depends on the nature of the equations being considered and the accuracy of the results required. The development of the above methods is based on the assumption that the differential equation obeys the Lipschitz condition, i.e.

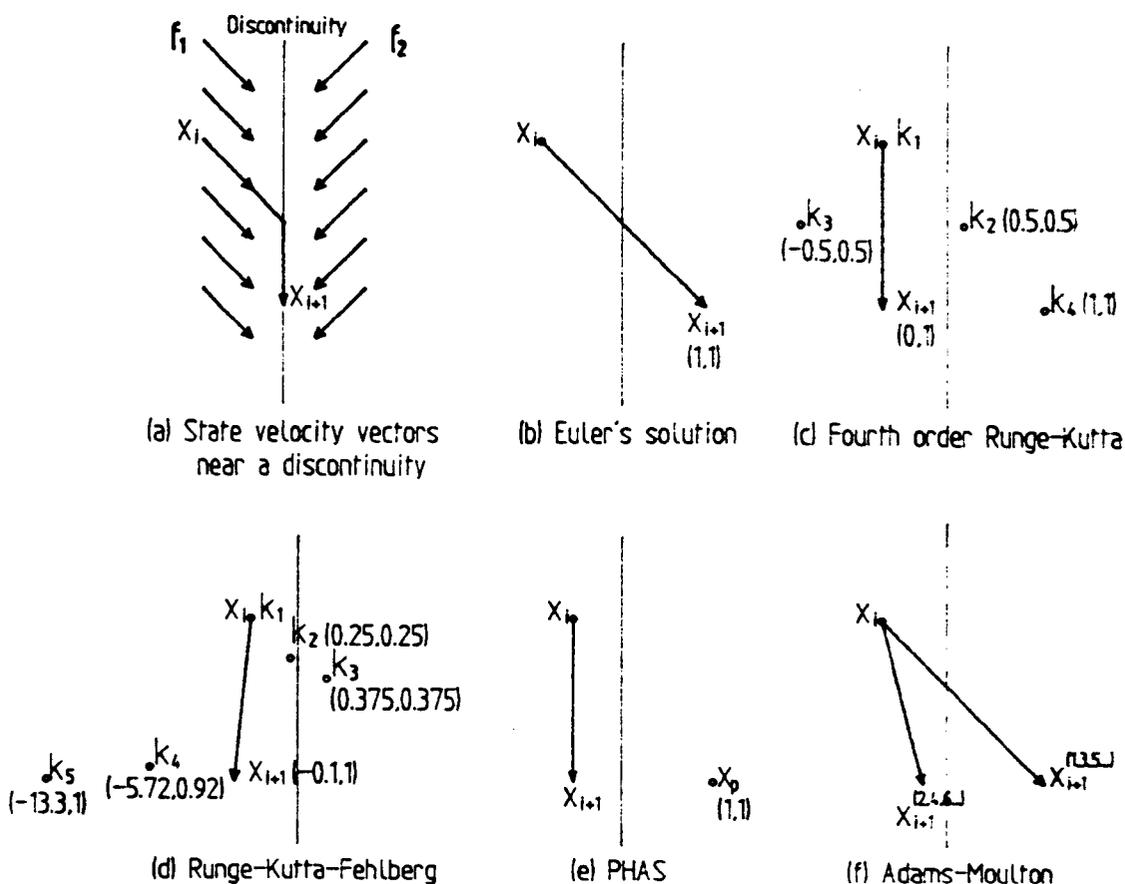
$$|f(x_1, t_1) - f(x_2, t_2)| < L |t_1 - t_2|$$

for some constant, L . This may not be the case in practice and, indeed such discontinuous right-hand sides may be introduced intentionally, in a control system for instance, to obtain particular behaviour. The algorithms given above will perform differently when the system considered is in violation of this condition, and this affects the selection of the most appropriate method for solving such differential equations. A fixed step length algorithm will proceed with the usual step length and, when a discontinuity is encountered, will make the next step in a direction which is some combination of the direction vectors on either side of the discontinuity.

A typical situation, which will be used to illustrate the effect of a discontinuity on various of the algorithms discussed, is shown as an isocline plot and phase plane trajectory in figure 5.4(a). In the case of Euler's equation, the trajectory is updated as shown in figure 5.4(b), where x_i is the starting point and x_{i+1} the subsequent point generated. For the more complex, fourth order Runge-Kutta algorithm described by equation 5.1 the behaviour is shown in figure 5.4(c), where the points x^1 , x^2 , x^3 and x^4 indicate the points at which k_1 to k_4 are evaluated. Here the direction of the next step depends on the position of the starting point x_i relative to the discontinuity but a typical result would be that illustrated. When the Fehlberg equations (5.2) are used the situation can become worse, as shown in figure 5.4(d), since the weighting on the terms of the equation are more diverse and can produce larger errors than

occur with the basic Runge-Kutta algorithm which has more evenly weighted terms. For the PHAS algorithm the situation is illustrated in figure 5.4(e) where E_{ave} is the direction of the next step.

Figure 5.4 Behaviour of numerical algorithms in the vicinity of a discontinuity



When a multi-step algorithm is used and this type of discontinuity is encountered the direction of the subsequent steps depends on the past approximations and so the trajectory is slow to change direction, as shown in figure 5.4(f), where a fourth order Adams-Moulton algorithm is considered. It is clear that this type of algorithm is liable to generate larger errors than any of those considered previously. It should also be noted that in this example, after the second iterations, the corrector equation gives the same result each time it is applied. Thus, if the difference between subsequent iterations is taken as a measure of the accuracy the results generated using this algorithm will,

incorrectly, appear very precise, even though the solution is poor.

As the difference in the direction of the derivatives increases, the problems described above are heightened.

The errors introduced when a discontinuity is encountered are of the order of h and so may only be reduced by a significant decrease in h with the resulting increase in computation time. Clearly this reduction in step size need only be made in the vicinity of the discontinuity and so a variable step size algorithm proves useful here. In order to minimise the computation time it is important that the algorithm used to alter h is efficient when a discontinuity is encountered, without significantly adding to the computation burden when the step size is only varying slightly.

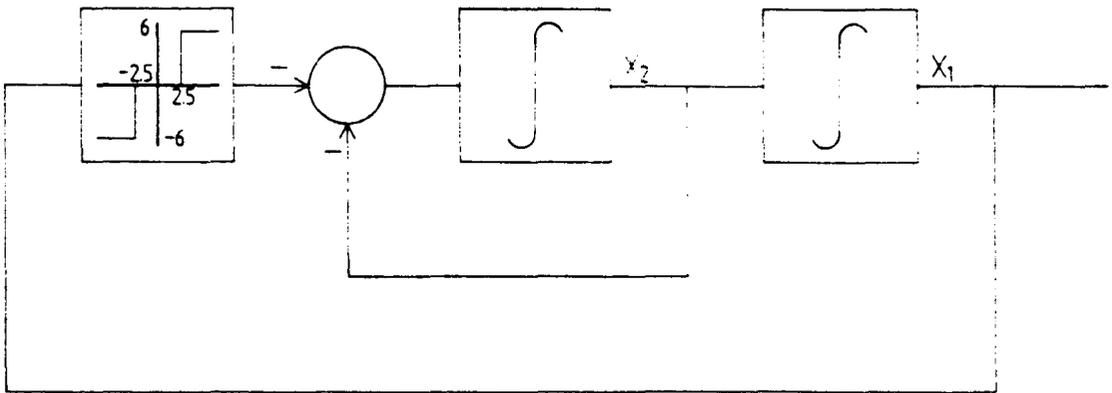
5.4.4.1 Example Problems for Algorithm Comparison

In this section a number of example problems are considered which were used to test the performance of some of the algorithms described above. The examples considered are sufficiently simple to allow an understanding of the solution process to be gained and they possess analytic solutions so that the results obtained may be compared with the true solutions.

Example 1.

The first example is that of a simplified model of a DC motor with position feedback consisting of a relay with dead-band, as illustrated in figure 5.5.

Figure 5.5 Motor with relay control system



The state space description of this system is:

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

$$\dot{x}_2 = -x_2 + U$$

$$U = \begin{cases} -6 \operatorname{sgn}(x_1) & |x_1| > 2.5 \\ 0 & |x_1| \leq 2.5 \end{cases}$$

With initial conditions:

$$x_1(0) = 8 \quad x_2(0) = 0$$

the analytic solution, shown in figure 5.6, is split into two parts, with a single crossing of the discontinuity: the first section is for $x_1 > 2.5$ and the remainder for $x_1 \leq 2.5$. The transition between these stages occurs at $t_1 = 1.74139$.

For $0 < t < t_1$,

$$x_1 = 14 - 6(t + e^{-t})$$

$$x_2 = -6(1 - e^{-t})$$

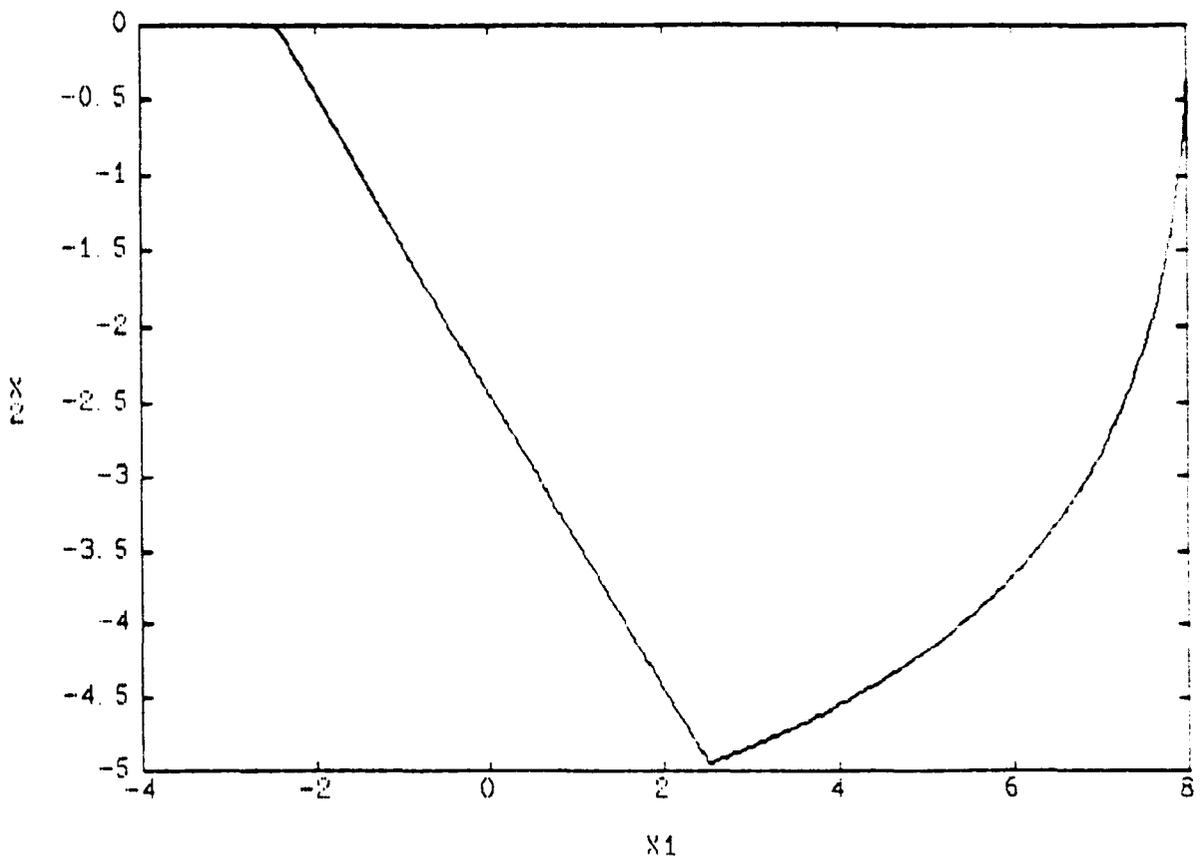
and for $t > t_1$,

$$x_1 = 2.5 - c (1 - e^{-(t-t_1)})$$

$$x_2 = -c e^{-(t-t_1)}$$

where $c = x_2(t_1) = 4.94834$

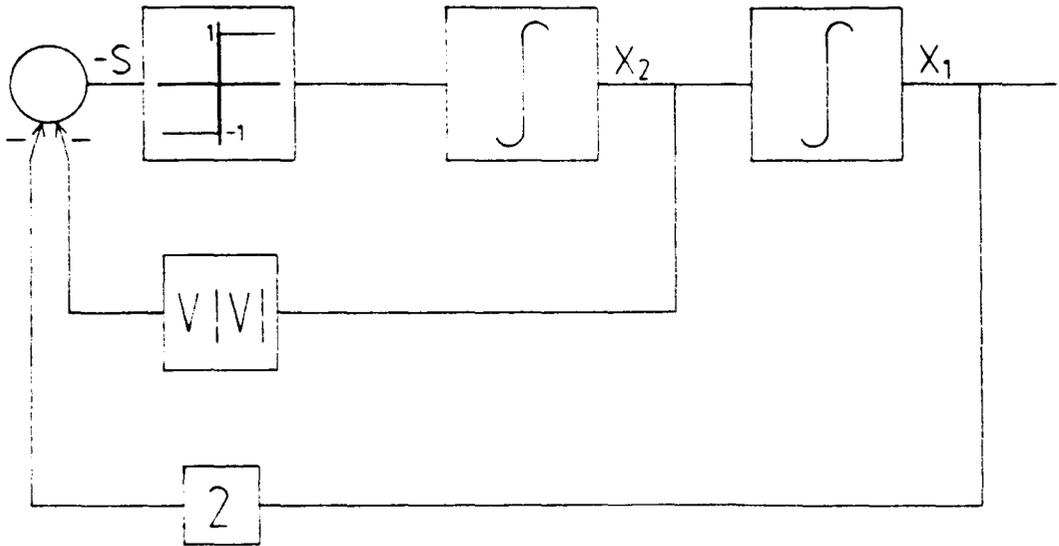
Figure 5.6 Analytic solution to example problem 1



Example 2.

The second example problem considers a double integrator plant with an ideal relay in the forward path and $V|V|$ feedback designed to give a time optimal response. The block diagram of the system is shown in figure 5.7.

Figure 5.7 Double integrator with $V|V|$ feedback



The nonlinear feedback defines a switching line in phase space given by the equation $s = 2x_1 + x_2|x_2| = 0$ and the system response depends on the position of the state point, relative to this line so that the dynamic equations are given by:

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

$$\dot{x}_2 = -1 \operatorname{sgn}(s)$$

The response, for initial conditions $x_1(0) = 20$, $x_2(0) = 0$, is split into three sections given by:

for $0 < t < t_1$, where $t_1 = \sqrt{20}$

$$x_1 = 20 - \frac{1}{2} t^2$$

$$x_2 = -t$$

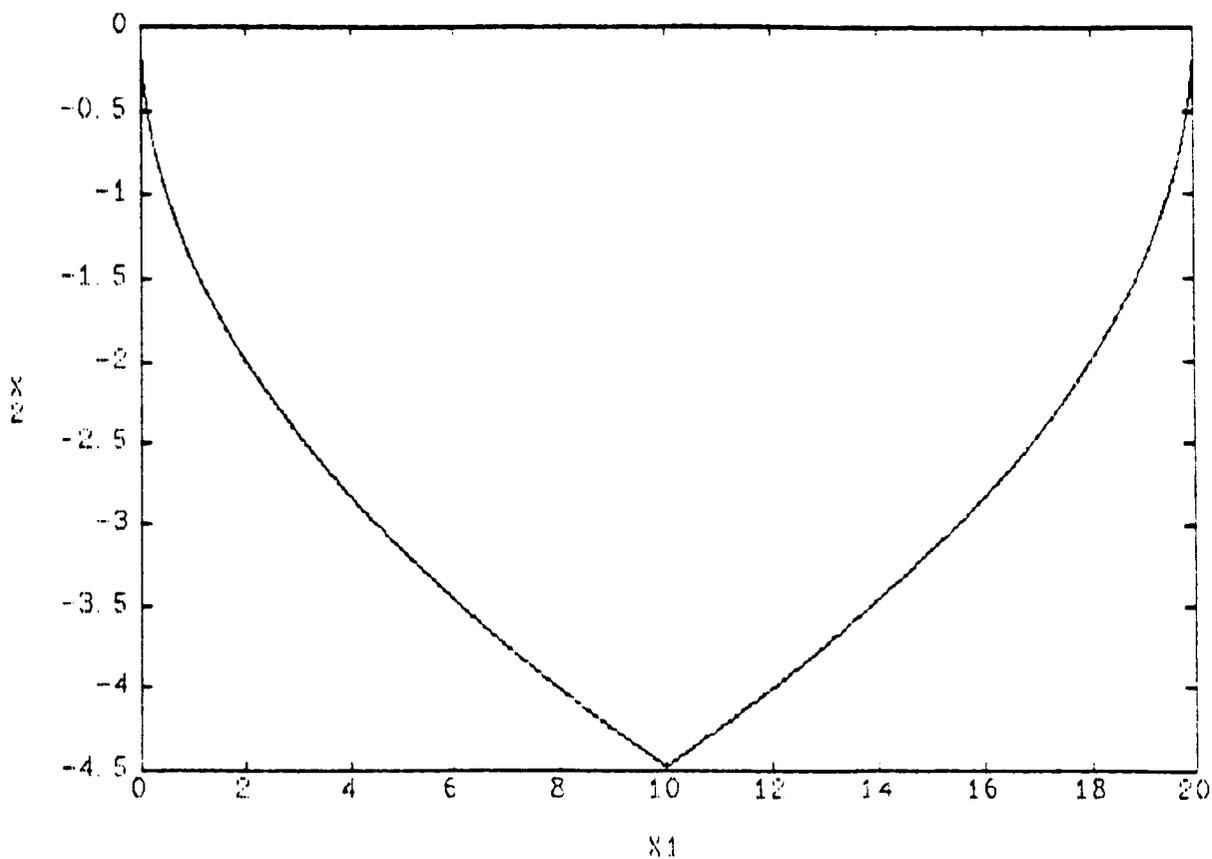
for $t_1 < t < t_2$, where $t_2 = 2\sqrt{20}$

$$x_1 = 40 - 2\sqrt{20}t + \frac{1}{2}t^2$$

$$x_2 = -2\sqrt{20} + t$$

for $t > t_2$ the state point remains at the origin. This system is unrealistic in that a real system would form a limit cycle around the origin, due to the finite switching time of the relay. The analytic phase plane trajectory for this system is shown in figure 5.8.

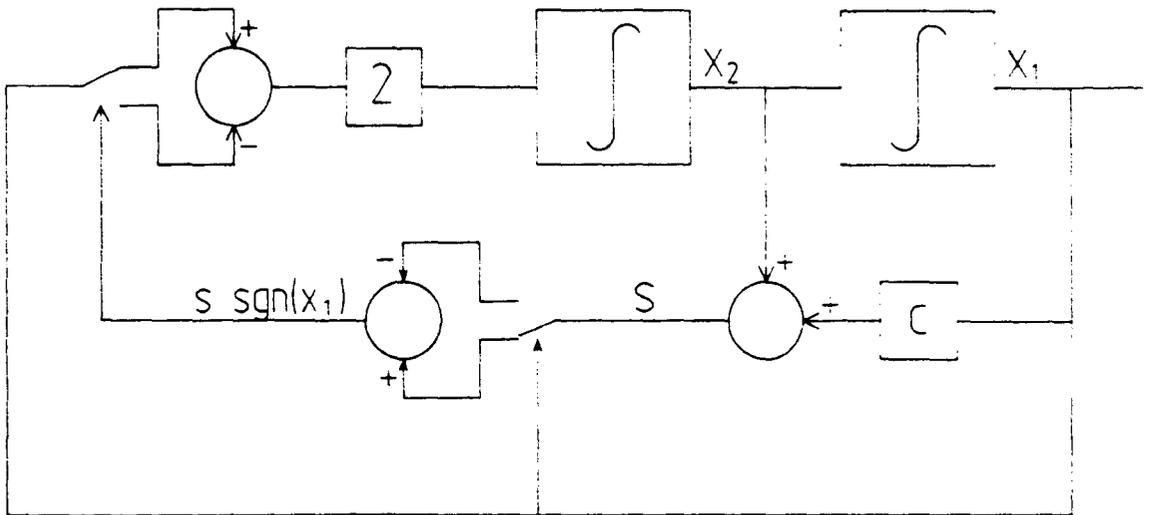
Figure 5.8 Analytic solution to example problem 2



Example 3.

The third example involves a simple second order system with a variable structure controller using a straight switching line, as illustrated in figure 5.9.

Figure 5.9 Second order system with variable structure control system



The plant dynamics are given by:

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

$$\dot{x}_2 = -x_2 + U$$

$$x_1(0) = 8 \quad x_2(0) = 0$$

while the controller is defined by:

$$U = \begin{cases} -2x_1 & \text{for } s > 0 \\ 2x_1 & \text{for } s < 0 \end{cases}$$

where the switching line, s is given by:

$$s = x_1 + x_2$$

During the reaching phase, before the trajectory reaches the switching line, the response is given by:

$$x_1 = 16 \sqrt{2/7} e^{-\frac{1}{2}t} \sin(\omega t + \alpha)$$

$$x_2 = -32/\sqrt{7} e^{-\frac{1}{2}t} \sin(\omega t)$$

where $\omega = \frac{1}{2}\sqrt{7}$ and $\alpha = \cos^{-1}\left\{\frac{1}{2\sqrt{2}}\right\}$

The trajectory reaches the switching line at $t = t_1$, where:

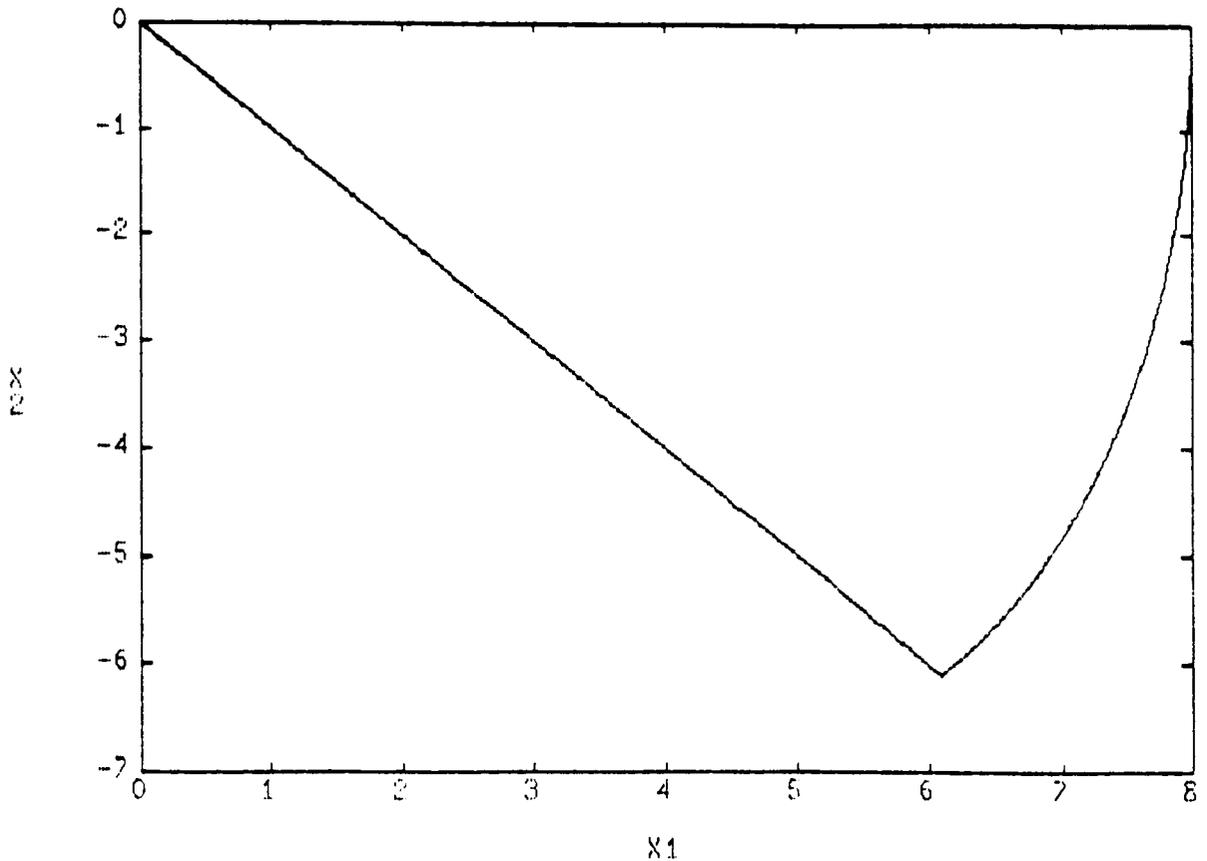
$$t_1 = \frac{2}{\sqrt{7}} \tan^{-1}\left\{\frac{\sqrt{7}}{3}\right\} = 0.5463$$

From this point onward the trajectory follows the switching line and the state is defined by:

$$x_1 = -x_2 = x_1(t_1) e^{-(t-t_1)}$$

where $x_1(t_1) = 6.0877$. The analytic phase plane trajectory for this system is shown in figure 5.10

Figure 5.10 Analytic solution to example problem 3



5.4.4.2 Comparison of Algorithms for Example Problems

In order to find the relative merits of the different types of algorithm described earlier in this section, a number were implemented in the environment of the PHAS package. The methods selected were a fourth order, fixed step size Runge-Kutta algorithm, given by equation 5.1, since this is commonly used in other packages; a variable step length fourth/fifth order Runge-Kutta algorithm as described in section 5.4.1.1 and the PHAS algorithm described in section 5.4.3, with the option of either fixed or variable step size. Multi-step algorithms were not considered here as they are unlikely to perform well in systems with discontinuous right hand sides for the reasons given in section 5.4.2.

For the fixed step length algorithms a range of step sizes were used, giving a useful range of accuracies. With the variable step size algorithms a range of maximum and minimum step lengths were used, along with a range of accuracy specifications. All algorithms kept a record of the number of times the

right hand side of the differential equation was evaluated, as a way of comparing the computation time required for the different methods. To investigate the performance of the algorithms the results at each step of the solution were compared with the analytic solutions with the maximum and mean absolute errors between the estimated and analytic value of each state stored to disc. The results at each iteration were also be stored in a log file to allow more detailed analysis.

Each of the example problems will be considered in turn with the results for the fixed step size algorithms considered first, since these are the simplest to investigate. The variable step size algorithms will then be considered.

For the first example problem, during the first section of the response, before reaching the discontinuity, the fourth order Runge-Kutta algorithm gave an error of the order of h^4 for moderate step lengths. The Fehlberg equations also gave an error of $O(h^4)$ but lower than those for the standard Runge-Kutta algorithm by a factor of approximately 2 for a given step length. To attain the same accuracy using the Fehlberg algorithm would thus require a step size of $2^{1/4}h_4$ or $1.19h_4$, where h_4 is the step size used in the fourth order Runge-Kutta equations. The calculation of these equations requires 6 as opposed to 4 function evaluations at each step so that a comparison in terms of the number of evaluations indicates that the standard fourth order equations are more accurate by a factor of $1.5/1.19 \approx 1.25$. For small step lengths the errors occurring were larger than $O(h^4)$ due to truncation errors. For this section the errors using the PHAS algorithm were of the order of h^2 . These results were as expected, corresponding with the order of the approximation equations.

Once the discontinuity is reached, the errors generated using all algorithms increases sharply and becomes of the order of h , although the errors for the Runge-Kutta algorithms were between two and four times larger than those for the PHAS algorithm.

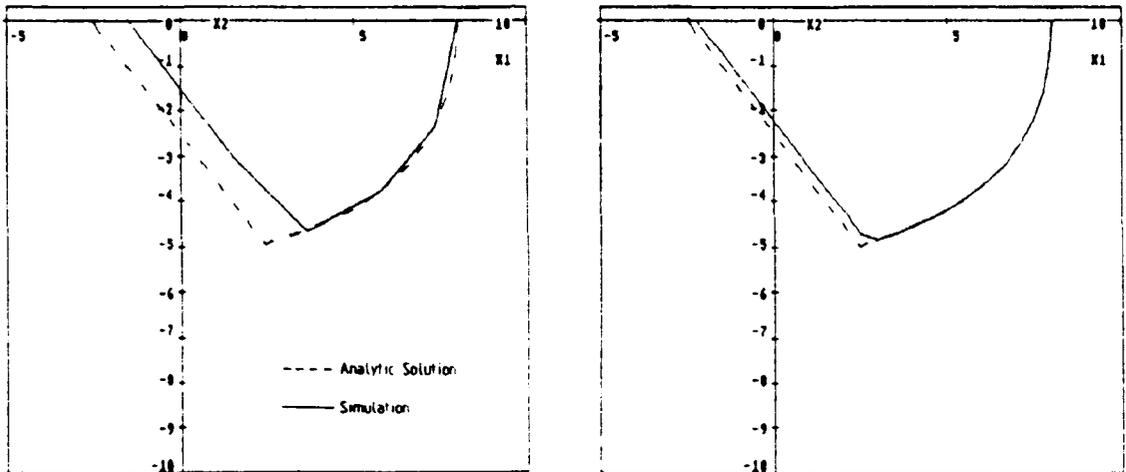
Both the variable step size PHAS and Runge-Kutta algorithms behaved well on the first part of the response, the mean error being controlled by the

tolerance set, independent of the minimum search radius provided this was set below a certain level. The Runge-Kutta algorithm was more accurate for a given number of function evaluations for most minimum step sizes although this became less so when the minimum step size was increased. For the second section of the response the accuracy obtained using both algorithms was proportional to the preset tolerance for the majority of minimum step sizes, with the PHAS algorithm being more accurate by a factor of approximately five. For low preset tolerance the minimum step size became more important causing the algorithms to behave somewhat differently. The PHAS algorithm continued to give an accuracy dependant on the set tolerance independent of the minimum step size although this was at the cost of many more function evaluations. The results for the Runge-Kutta algorithm became very dependant on the minimum step size but did not require a disproportionately large number of function evaluations. The difference in behaviour between the algorithms is due to the different step size adaptation mechanisms employed. The Runge-Kutta algorithm calculates the optimal step size at each step and provided this is correct the minimum possible number of steps will be used for a given problem. The PHAS algorithm however, begins with the maximum step size and gradually decreases this until a sufficiently accurate result is obtained. Thus, for a problem which requires a small step size not only are more steps required but each step also requires many more function evaluations.

When comparing the fixed and variable step size algorithms for this example it is found that both types of algorithms behaved similarly for the first section, when the required step size was large. However for the second section the variable step size algorithms became more accurate by a factor of approximately ten for the same number of evaluations. The effect of including time as an extra state in the PHAS algorithm is seen in figure 5.11. Here phase plane plots are shown for data from fixed step length PHAS and Runge-Kutta algorithms which required the same number of function evaluations. It is clear that the response for the PHAS procedure is far

smoother, even though the data points were of a similar accuracy for both routines.

Figure 5.11 Comparison of Runge-Kutta and PHAS algorithms showing improved smoothness



(i) Runge-Kutta

(ii) PHAS

For the second example problem the response is split into three sections. The first, before the trajectory reaches the the switching line is a simple double integrator with a constant input and as a result the errors in x_2 generated by all the algorithms were small while for x_1 , the errors were zero. For the Runge-Kutta algorithms, when a step length of greater than 0.2s was used the errors generated were lower than the machine resolution so that they appeared to be zero. It should be remembered however that this was only for the first few steps and could not be relied upon to continue. Once the step length was reduced the errors increased due to the increasing number of steps required, each of which has the possibility of incorporating truncation error. The PHAS algorithm gives results with a error of the order of h^2 for large step sizes until the truncation errors become significant. For small step sizes the errors produced using PHAS were a factor of five smaller than those from the

Runge-Kutta algorithms for the same number of function evaluations.

In the second section of the response the errors generated are dependant on the accuracy of the step in which the switching line is crossed, since from this point on the motion is similar to that found in the first section. In all cases the errors were of the order of h with PHAS giving the smallest errors, followed by the standard Runge-Kutta and Fehlberg equations with errors of two and five times those for PHAS respectively. The final section of the response should be a limit cycle about the origin with infinite frequency and zero amplitude, however due to the finite step size used the limit cycle will have an amplitude of the order of h . This was found to be the case with all the algorithms, and once again the PHAS equation gave the smallest errors with those for the Runge-Kutta algorithms approximately two times larger for a given number of function evaluations.

When the variable step size algorithms are considered it is found that in the first section of the response the errors generated were significantly larger than those obtained with the fixed step methods, the errors being at least ten times larger for the same number of evaluations. The behaviour in the second and third sections was similar for each of the algorithms. The PHAS algorithm gave errors approximately five times smaller than their fixed step size equivalent in each section. The Runge-Kutta algorithm however gave errors of the same order as the fixed step version and involved many more function evaluations in the final section of the response.

The final example consists of two sections, the first a damped second order response is followed by a sliding response which should ideally follow a straight line in the phase plane to the origin. Due to the switching behaviour in the latter section the response is bound to deviate from the ideal, when a finite step size is used. In the first section the errors are of the order of h^2 and h^4 for the PHAS and Runge-Kutta algorithms respectively, with both forms of Runge-Kutta algorithm resulting in similar errors for a given number of function evaluations. Once the sliding regime is begun the errors increase

significantly for all algorithms and become of the order of h . The PHAS algorithm gave the smallest errors although the difference between the algorithms was small.

The performance of the variable step size algorithms for the third problem was not as good as that for fixed step methods in each of the sections. In the first section the errors for the variable step algorithms were of the order of 100 times larger than their fixed step counterparts while in the latter part the errors were approximately two times larger for the same number of function evaluations. Of the two variable step algorithms the Fehlberg procedure performed better in the first section while PHAS gave more accurate results in the second part, although in each case the difference was small.

5.4.5 Selection of Numerical Algorithm

It was shown in the previous section that the algorithm used to solve nonlinear differential equations has a major effect on the speed of computation and the accuracy of the results. It will also be noted that the most appropriate algorithm depends on the nature of the problem considered and whether the results are to be plotted as phase plane trajectories or as time responses. The choice of method is less critical when linear and smooth nonlinear systems are investigated. In this case the most important consideration is the accuracy of the result required. In general, if only moderate accuracy is required, then the PHAS algorithm is preferable, particularly when a phase plane plot is required. If however a high accuracy is required then the higher order Runge-Kutta algorithm is superior, requiring a smaller reduction of step length for a given accuracy. The small step size used means that the trajectories are smooth whether plotted in phase plane or time response. Although it may be possible to find the most appropriate step size for a particular problem and hence obtain results with the minimum computation, this is not usually the case and, indeed the step size may need to alter as the solution proceeds and so the

variable step size algorithms are generally preferable for this type of problem.

When equations with non-smooth nonlinear feedback elements are considered, the selection of numerical algorithm becomes more critical. For our purposes these types of equation may be divided into two categories, firstly those with a small number of discontinuities which trajectory reach, cross and leave on the other side. The second category may be classed as those which involve sliding modes, i.e. those in which the derivatives on either side of the discontinuity point towards it so that the trajectory effectively travels along the discontinuity. In the former case the majority of the solution is for a smooth equation and so the selection of algorithm depends on the same conditions as in the previous section although a variable step size algorithm must be used if the solution is to be obtained without reducing the step size for the whole solution for the sake of a few points close to the discontinuity. The second example problem shows that for this type of system the PHAS algorithm is between two and five times more accurate than the others tested here, for the same number of function evaluations. This effect is due to the fact that each step using the PHAS algorithm will be in a direction between the directions of the derivatives on either side of the discontinuity while the high order equations used in the Runge-Kutta algorithms allow a step in a completely different direction, as indicated by the example in section 5.4.4. The step size adaptation algorithm described in section 5.4.1.1 for the Runge-Kutta algorithm may also become inefficient when a discontinuity is encountered since the difference between the fourth and fifth order equations may not give a good measure of the local truncation error, since the Lipschitz condition is violated.

When the system considered involves a sliding motion the fixed step size algorithms give an error of the order of the step size, with the PHAS algorithm being more efficient than the Runge-Kutta algorithms, since this requires only two function evaluations at each point, as opposed to four or six. In this case, since the discontinuity is a stable manifold, the trajectory will always slide along the discontinuity, with only the amplitude of the chatter

depending on the accuracy of the algorithm. If a variable step size algorithm is employed then, provided the step size adaptation algorithm operates correctly, the step size will be reduced until the preset minimum is reached. The results obtained using this step size should give a similar accuracy to that obtained using their fixed step size counterparts, however the computation involved in the adaptation algorithm will have been wasted. Since the adaptation procedure used for the PHAS algorithm is not efficient when a significant reduction in step size is required, it is not particularly suitable in this type of problem, with the Runge-Kutta method being superior. Hence, if the sliding mode constitutes a large portion of the trajectory a fixed step algorithm is preferable, with the PHAS algorithm being the best of those considered here, while if the sliding mode is only a small part of the solution a variable step size Runge-Kutta method may be better.

In accordance with these findings it is intended that the algorithms described here will continue to be made available to the user of PHAS, as will the option of using either fixed or variable step lengths.

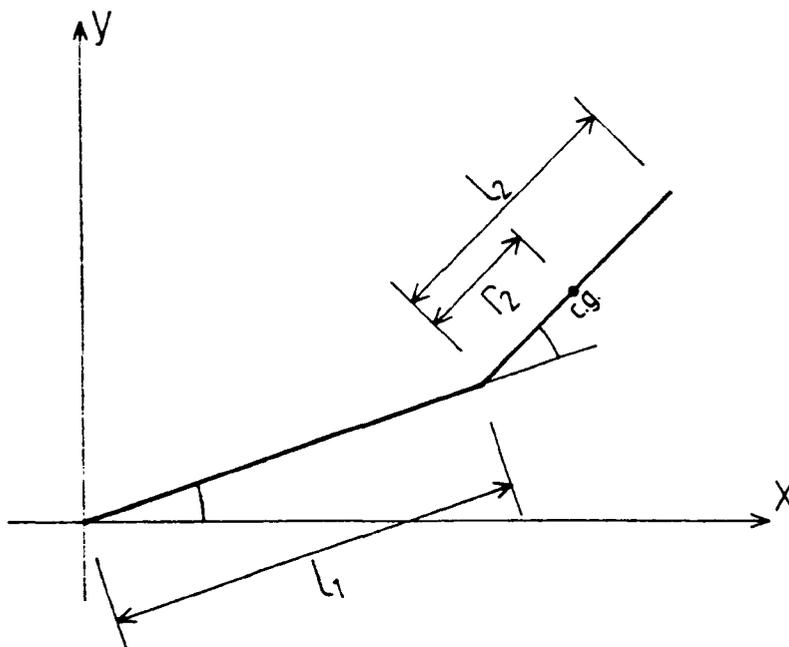
5.5 Evaluation of the CACSD Suite for the Design of Nonlinear Systems

Having implemented the changes to the Computer Aided Control System Design Suite described in the previous sections, it is necessary to go on to test the performance of the system when used for a problem involving significant nonlinear behaviour. Although the dynamic model of the RTX robot developed in chapter 4 gives a reasonably accurate prediction of experimental results, it is not so accurate as to be useful in testing the performance of the CACSD suite, since any differences found could be due either to modelling errors or errors generated in the solution algorithms. It was therefore decided to compare the results obtained using the suite with published results and with those obtained using SIMNON for a problem having well understood results.

The problem selected involves the optimal control of a 2 degree of

freedom planar robot, as shown in figure 5.12.

Figure 5.12 Two D.o.F. planar robot used in optimal control investigation



This example, investigated by Geering [Geering,86], involves significant interaction terms as well as the discontinuous, saturated control signals required in time optimal motions and so gives an indication of the performance of the design suite in all important aspects. In addition, since the control is effectively open-loop, any errors generated will not be reduced by feedback and so will have a noticeable effect.

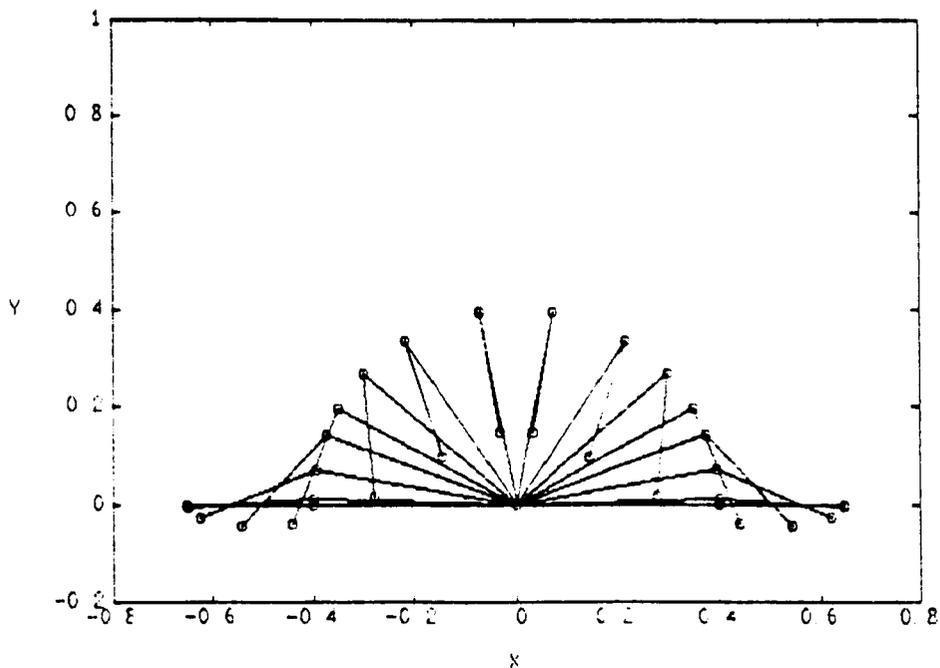
The nonlinear differential equations used to describe the robot behaviour, given in appendix 5, are presented in the original paper and were compared with those derived using Lagrange's equations, as described in chapter 4.1.4. These equations may be entered directly into SIMNON but in order to be compatible with the PHAS model structure, they must be divided up into a number of separate functions of the joint positions and velocities. There are a number of ways in which this may be achieved although the differences between them are minor. The form selected is described in appendix 5, as are the functions involved.

As explained in section 5.3, the form of these equations is not available as a standard form within the suite and so must be represented as either a piecewise linear function or as a power series. A number of tests were performed to compare the accuracy of each representation over the range of argument values required. Representing the least smooth of the functions as a set of 24 points for a piecewise linear curve resulted in a maximum error between this function and the true value of 1.3% of the full range of function values. For the same function, a number of different order polynomials were investigated and it was found that the maximum attainable accuracy was 1.1% of full range. This required a 10th order fit, while higher order polynomials gave larger errors due to truncation effects. The PHAS package allows a maximum of 7th order polynomials, for which the error was found to be 4.6%. Although either the maximum polynomial order or the number of data points for a piecewise linear fit could have been increased to improve this accuracy, it was felt that this would make the data entry task too time consuming and so the 24 point, piecewise linear fit was adopted.

A program was developed to construct a file containing the discrete data points in the appropriate format for the PHAS package to read, as if from the keyboard, along with the other system information. The representation of the system used in SIMNON is given in appendix 5.

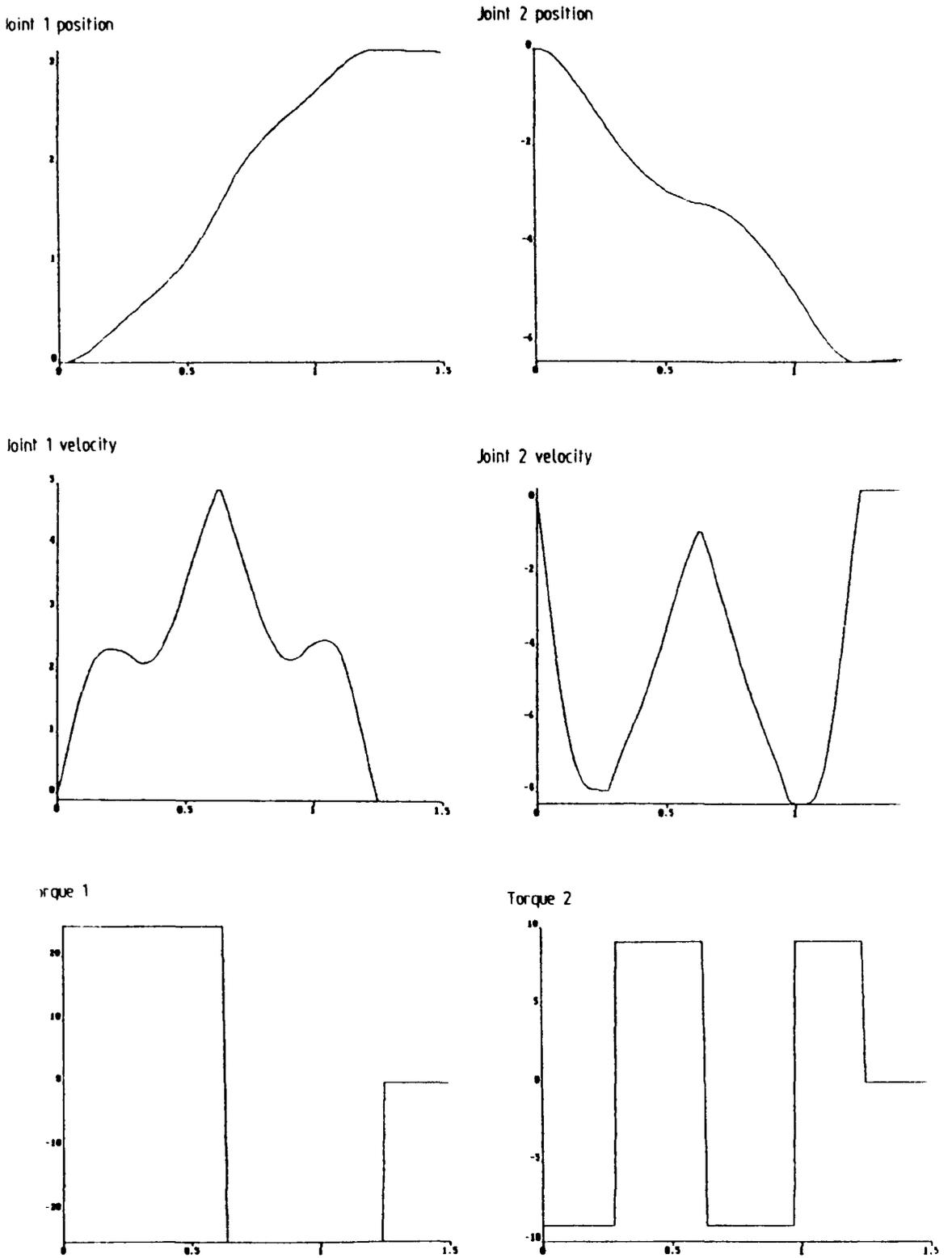
The motion considered here is from the extreme of the reach in one direction to the opposite extreme, requiring a rotation of 180° in joint 1 and 360° in joint 2 with zero initial and final velocities, as illustrated in figure 5.13. The time optimal control which achieves this involves one switch of the joint 1 torque and three switches of the joint 2 torque during the transient, as shown in figure 5.14, along with the time evolution of the four states.

Figure 5.13 Cartesian representation of required time optimal movement



The behaviour of the system was simulated using both SIMNON and PHAS and the results compared with those generated using SIMNON, with the error tolerance set to 10^{-5} times the default value. Using SIMNON running on an 8086 based PC compatible with numerical co-processor, with the default error tolerance, produced the results in approximately one quarter of the time required by PHAS running on a MicroVAX 2. The resulting errors in the state trajectories generated were approximately 25% larger using PHAS than SIMNON with the default error tolerance, having a mean of 0.9% of full scale for PHAS and 0.7% for SIMNON. Thus it is seen that even though the interaction dynamics are represented by piecewise linear data giving a maximum error of 1%, the resulting simulation produces trajectory information sufficiently accurate for the majority of applications, particularly when it is considered that, in the majority of instances, the use of feedback will significantly reduce the size of the errors produced.

Figure 5.14 Evolutin of states during time optimal movement



CHAPTER 6

Implementation and Testing of Variable Structure Control Systems on the RTX

A number of variable structure control systems have been implemented and tested on the RTX robot. The control structure and details of the methods of application will be described in this chapter along with an evaluation of the performance of each type of controller.

The general trend of the algorithms described in this chapter will be from simple to more complex, starting with a simple variable structure control loop around the joint motors and proceeding to consider the developments to this algorithms, required to give performance improvements in the presence of non-ideal behaviour. Following this, details will be given of the implementation of similar control algorithms, but using position information derived from joint mounted sensors alone and from both joint and motor mounted encoders. The chapter will conclude with a summary of the performance of each type of algorithm and a discussion of the selection of the most suitable control system for a particular application.

The standard against which the new controllers will be judged is a PID control, since this is the most commonly used control algorithm in robotics and is the one implemented by the manufacturers of the RTX. Clearly a comparison between the manufacturer's controller using the original hardware and a newly developed algorithm using the improved motor drive circuits and position sensors would not reveal a great deal about the properties of the control algorithms themselves. In order to provide a better basis for a comparison, a PID control was implemented using the same hardware as used in testing the other algorithms. The performance of the variable structure algorithms tested will be compared to that of the PID implemented on the same hardware, unless

otherwise stated. The performance will, where possible, be compared for both point-to-point and continuous path movements since both types of operation are required in practice.

Software versions of many of the control algorithms were tested using both the μ PDP with an implementation of 'C' supplied by Real Time Systems and on the Victor personal computer using Turbo Pascal Version 4, supplied by Borland. In the majority of cases positions were derived from the high resolution encoders mounted on either the motors or the joints themselves, as appropriate. The motors were driven using the current drives, connected to the computers via the interfaces described in section 3.2.

For the purposes of comparison, the results presented here will be those derived using the PC, since all controllers were not implemented on the μ PDP. The PC compatible used for the majority of the work was based on the 8086 processor, running at a clock speed of 4.77MHz, although a few requiring slightly more processing were derived using an 8086 based machine running at 7.16MHz. When using the PC, it was found that, in order to obtain a sufficiently high sampling speed, integer arithmetic was required throughout, with suitably scaled parameters derived from the real values entered (with the equivalent real values returned so that the effect of integer quantization could be noted).

The amount of computation required for the algorithm implementation is clearly an important factor in the selection of the most appropriate controller. The computation times involved may be found from the number of times a certain operation (in this case incrementing an integer counter) may be performed in the slack time between interrupts. The figures derived in this way are dependant on the efficiency of the coding used and, in many situations, certain operations (particularly multiplication and division of an integer by a power of two, used for scaling purposes) were replaced by faster machine code routines embedded in the Turbo Pascal program. For this reason, the times obtained do not give a reliable indication of algorithm speed, and so will not

be considered in detail. A more appropriate figure is obtained by considering the number of each operation required. These values will be given in section 6.7, where the performance of the different algorithms is compared.

Throughout the development of the control systems described in this chapter, the control parameters for each algorithm were selected through a sequence of simulation and experimental trials. This generally involved the determination of an approximate range of values which would give stable behaviour using a simulation, while fine tuning of the control was done on the robot itself. The first parameter selected was the switching function definition, since this gives defines the system behaviour when sliding, followed by the controller gain parameters, chosen to maintain sliding with the minimum of chatter.

6.1 Simple Variable Structure Control System

One of the most basic forms of variable structure control algorithm, designed for a second order system uses a control signal proportional to the position error multiplied by the sign of the switching function, which is a linear combination of position and velocity errors. Thus, for a system represented as:

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = f(e_1, e_2) + bu$$

where e_1 is the position error and e_2 the velocity error, the switching function is:

$$s = ce_1 + e_2 \quad (6.1)$$

and the control signal is defined as:

$$u = \begin{cases} -\dot{e}_1 & \text{if } se_1 > 0 \\ \dot{e}_1 & \text{if } se_1 < 0 \end{cases}$$

A controller of this type was implemented on the elbow and shoulder joints of the RTX using the position and velocity signals derived from the motor mounted encoders. Two forms were investigated, the first using a purely hardware control system used the low resolution motor mounted encoders and a pulse width modulation voltage mode motor drive circuit. The second implementation used software to generate the control algorithm and used the high resolution motor mounted encoders and the linear, current mode drive circuit described in chapter 3. These control systems will be described in sections 6.1.2 and 6.1.3, following details of the computer aided selection of the control system parameters described in section 6.1.1.

6.1.1 Computer Aided Design of a Simple VSC

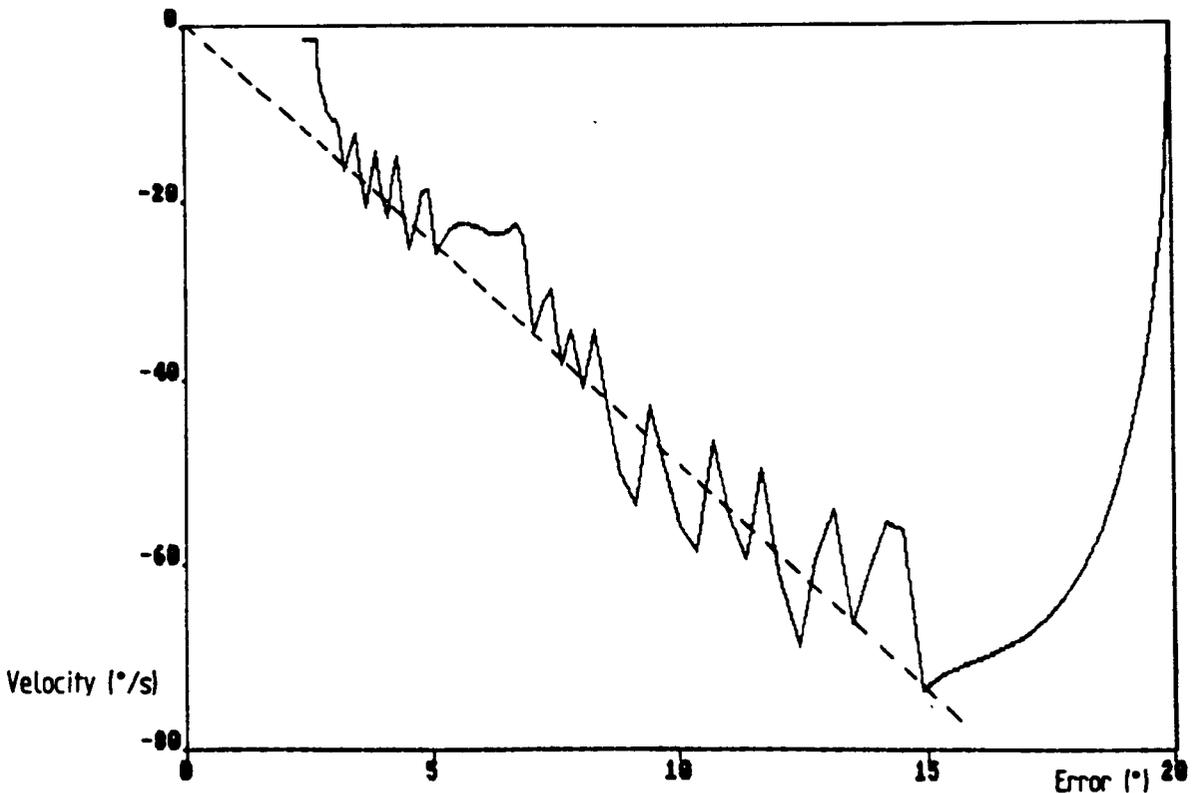
Having used SIMNON in the testing of the dynamic model of the RTX, as described in chapter 4, it is a simple matter to apply a variety of control algorithms to the plant model obtained. These controllers may be represented as either continuous or discrete time and so give a good indication of the likely performance when a real control system is implemented in either discrete or continuous form, and can be used to investigate the effect of changes in sample rate in the case of discrete time control systems. A number of control strategies were investigated using this technique, although they will not all be described in detail.

The plant model used in these simulations is the full single link nonlinear model with friction in the motor and joint taken from a file containing experimentally measured values. The continuous time model of the robot is stored in one file, while the discrete time controller definition is contained in a separate file, with the two linked using a connection definition file. This division allows rapid changes to be made to the joint dynamics and the

controller definition and keeps each section of the code to a manageable size. Further details of a typical set of files are given in appendix 6.

The simulation of the performance of a simple variable structure controller applied to the model of the elbow joint of the RTX is straightforward and although the computation time is quite long (typically 15 minutes for a step change in position demand) the resulting phase plane trajectories give a good correspondence with those found in practice. A typical result is shown in figure 6.1 for a controller with switching line coefficient, $c = 5s^{-1}$ and a gain, $\Psi = 1V/^\circ$, with a sampling frequency of 150Hz. These control system parameters were selected, through experiments, so as to give a fairly rapid response, without pushing the performance to the limits of what was achievable, so that less effective algorithms could be included in comparisons. The sample rate was selected to be as high as possible for the computationally most strenuous algorithm, so that all algorithms could be compared with an equal sample rate.

Figure 6.1 Simulated phase plane response for simple VSC



The reaching and sliding phases can be clearly distinguished, as can the chattering behaviour and the steady state error due to Coulomb friction. There is a significant steady state error, indicating the large amount of friction present in the drive system. This could be reduced by an increase in gain but only at the cost of increased chatter.

It will be seen that the velocity does not fall to exactly zero near the origin but continues at approximately $2 \dot{/s}$. This is due to the quantized representation of the frictional behaviour used in storing the data. This data is converted, using a linear interpolation, to give a friction value of between 0 and 0.50Nm in the velocity range 0 to $2.8 \dot{/s}$. With the reduced friction approximation at this velocity, some movement is still possible.

6.1.2 Hardware Implementation of a Simple VSC

At an early stage of the investigation into the suitability of variable structure control systems for the RTX, before the enhanced hardware was developed, it was decided that a hardware implementation of such a controller would provide a useful indication of the problems which might be encountered at a later stage. Accordingly a controller was implemented in hardware, deriving position and velocity signals from the low resolution encoders mounted on the motor and driving the motor via a pulse width modulation voltage mode drive circuit.

The control algorithm was implemented using analogue circuitry so as to allow rapid alterations to the control parameters by means of variable resistances. The encoder signals were used to drive a series of up-down counters which provided the input to a digital to analogue converter so as to give a voltage proportional the position and, by means of a differencing amplifier, the position error. The velocity signal was derived using a tachometer circuit with a direction discriminator and switched gain amplifier of the type described in section 3.4. Owing to the low encoder resolution, a filter having a

time constant of approximately 50mS was required to smooth the speed signal, introducing a significant phase shift into the measured velocity.

A schematic diagram of the control system is shown in figure 6.2 while the circuit used to implement it is shown in figure 6.3.

Figure 6.2 Schematic Diagram of the Hardware Implementation

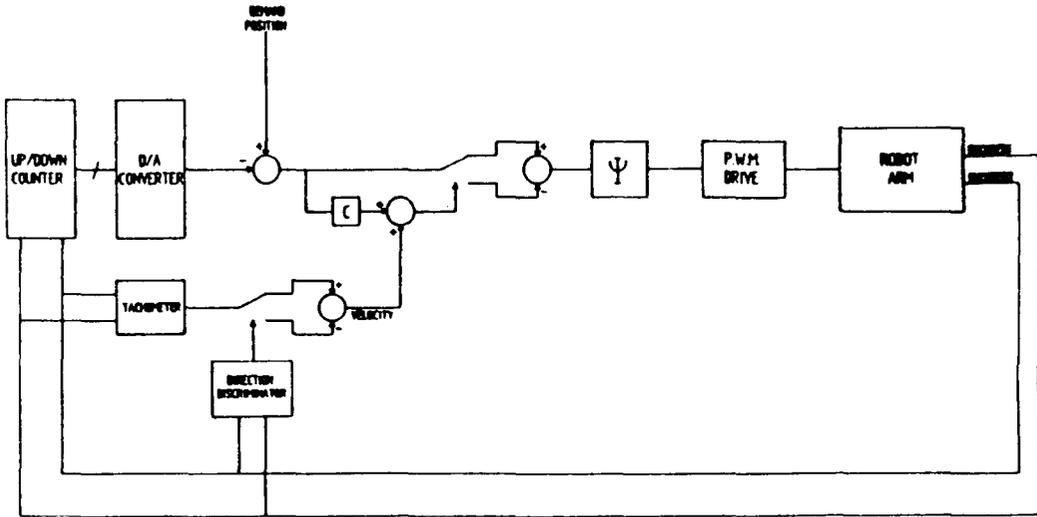
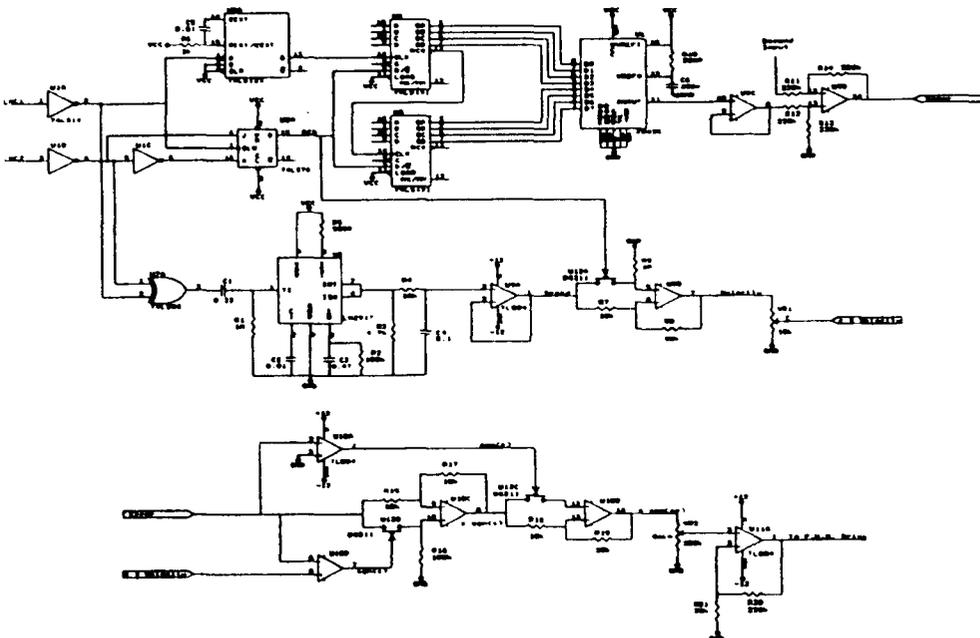


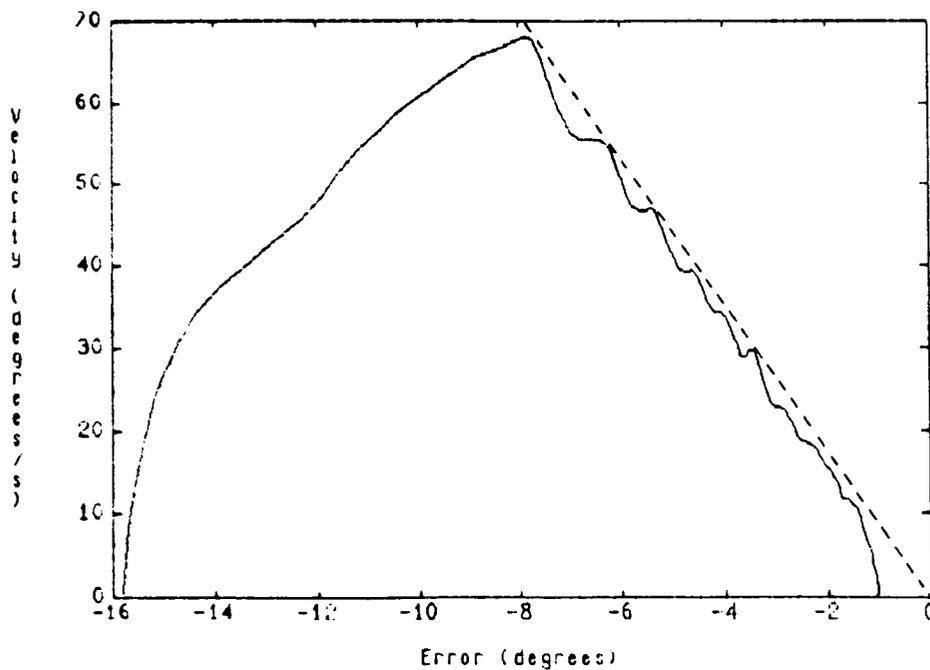
Figure 6.3 Circuit Diagram of the Hardware Implementation



The control parameters are c , the switching line gradient, and Ψ , the controller gain; these are defined by VR_1 and VR_2 which were implemented using resistance boxes, to allow rapid variation.

The behaviour of this control system when subject to a step change in demand position was investigated for a number of switching line gradients and controller gains. A typical response for the elbow joint is shown in figure 6.4 for a switching line gradient of $8.8s^{-1}$ and a gain of $1.23V/\dot{\cdot}$. These parameters were selected, using simulation and experimental procedures, to give a moderate level of performance. The expected pseudo-sliding behaviour is clear, as is the chatter along the switching line. From the actuation signal it is easy to distinguish the reaching and sliding phases of the movement. It will also be seen that a significant steady state position error occurs, typically 0.95° . This is due to the Coulomb friction present in the motor and drive system, and is one of the major problems with this type of control system.

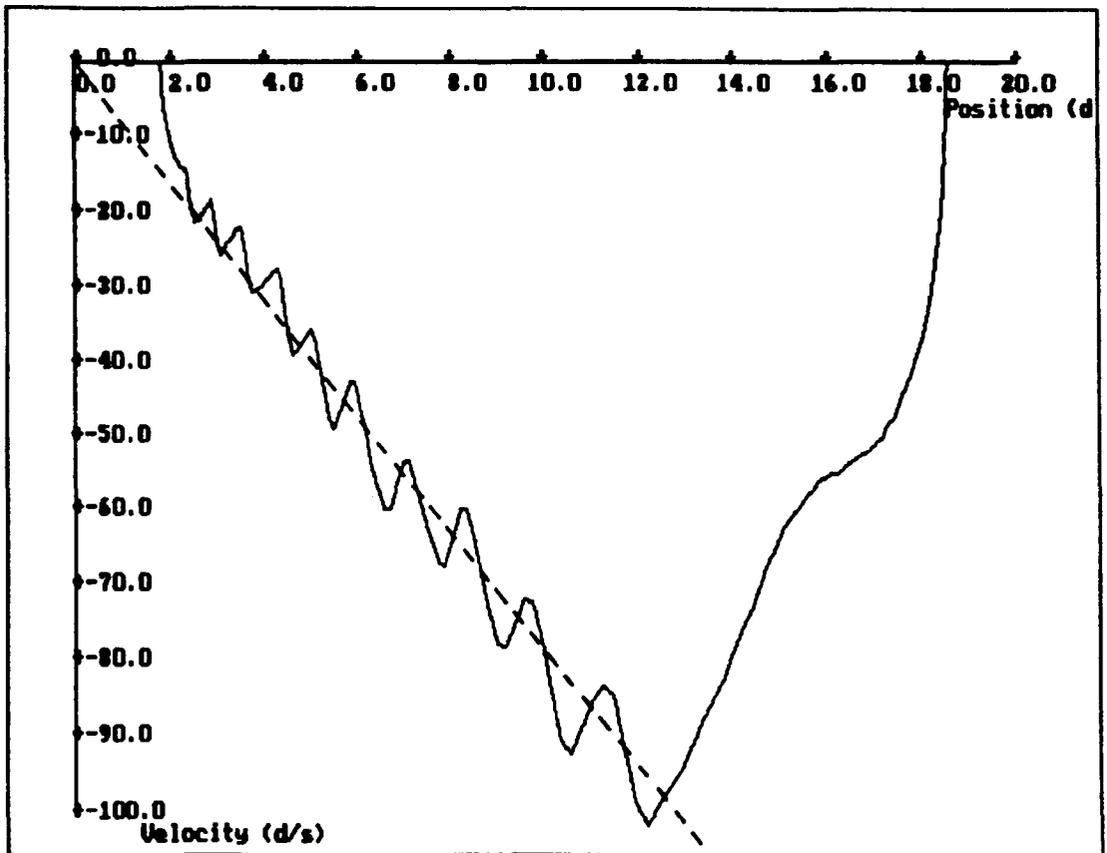
Figure 6.4 Response of the elbow joint with hardware VSC



6.1.3 Software Implementation of a Simple VSC

Having shown that a hardware implementation of a variable structure controller was at least partially successful, it was decided to implement a similar controller in software, using the computer system described earlier in this chapter. A typical responses for the elbow joint, with a controller having switching line coefficient, $c = 8s^{-1}$, a gain of $\Psi = 1.0V/^\circ$ and a sample rate of 400Hz, when subject to a 20° step change in demand position is shown in figure 6.5, from which it can be seen once again that pseudo-sliding behaviour occurs.

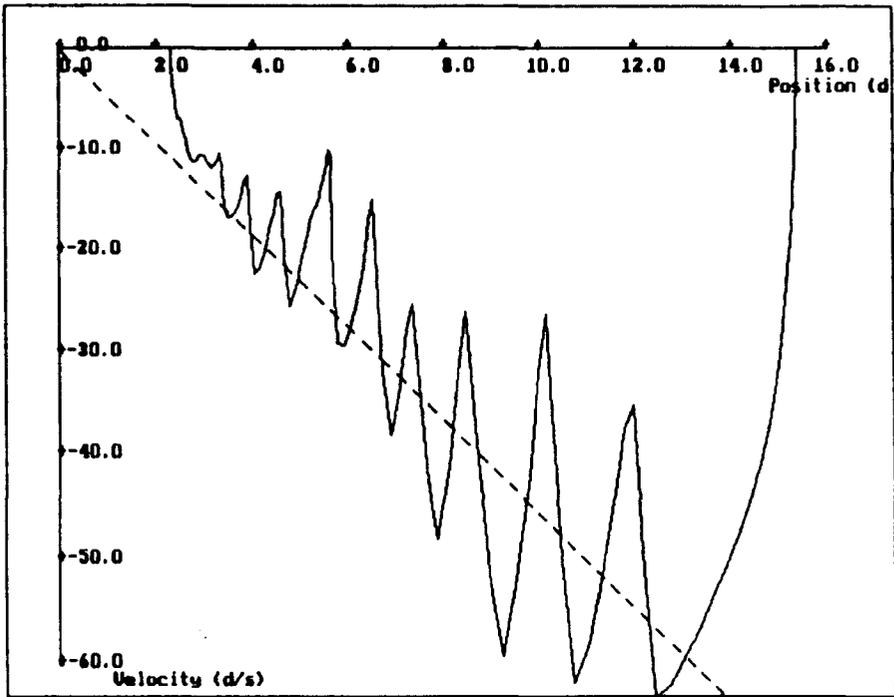
Figure 6.5 Response of a simple VSC implemented in software



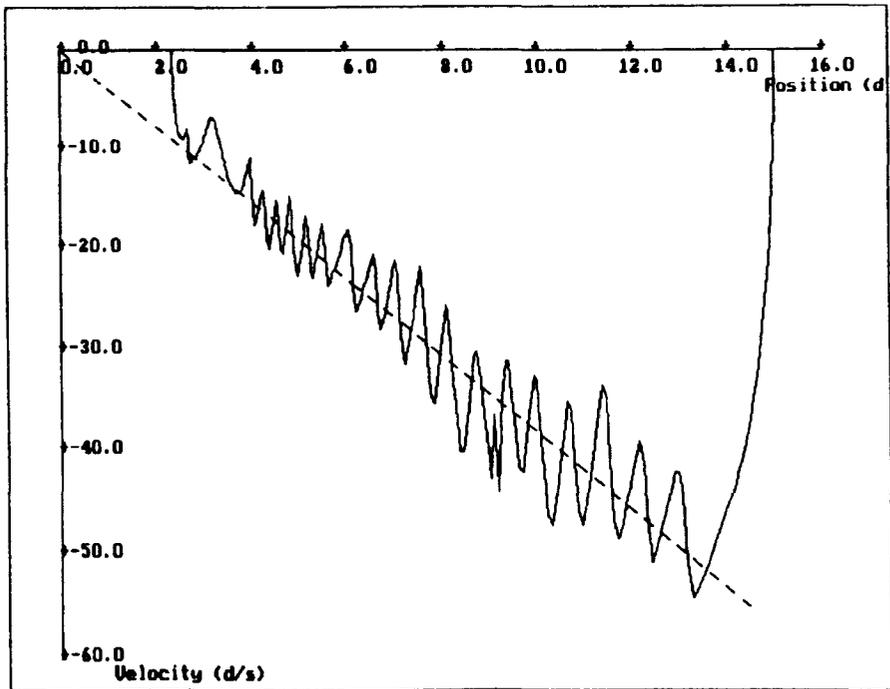
It is worthwhile noting the effect of varying the sampling interval on the response. Figure 6.6 shows the response of the elbow joint using first a controller having $c = 4.69s^{-1}$ and a sample rate of 200Hz and then one having

$c = 3.91s^{-1}$ and as sample rate of 500Hz, both with a gain of $0.93V/^\circ$. Clearly there is a significant reduction in chatter in the latter case, due to the increased sample rate. To verify the accuracy of the simulation described in section 6.1.1, the simulated and experimental responses of the elbow joint are shown in figure 6.7 for a choice of control parameters which cause a significant amount of chatter. The effect of any inaccuracies in the system model used in simulation, particularly high frequency dynamics, are accentuated by the high frequency switching present in the sliding mode and so a very close agreement would not be expected. It will however be seen that there is agreement in the overall behaviour and particularly in the switching line following, and hence speed of response, and the steady state position error.

Figure 6.6 Effect of varying sampling rate in a simple VSC

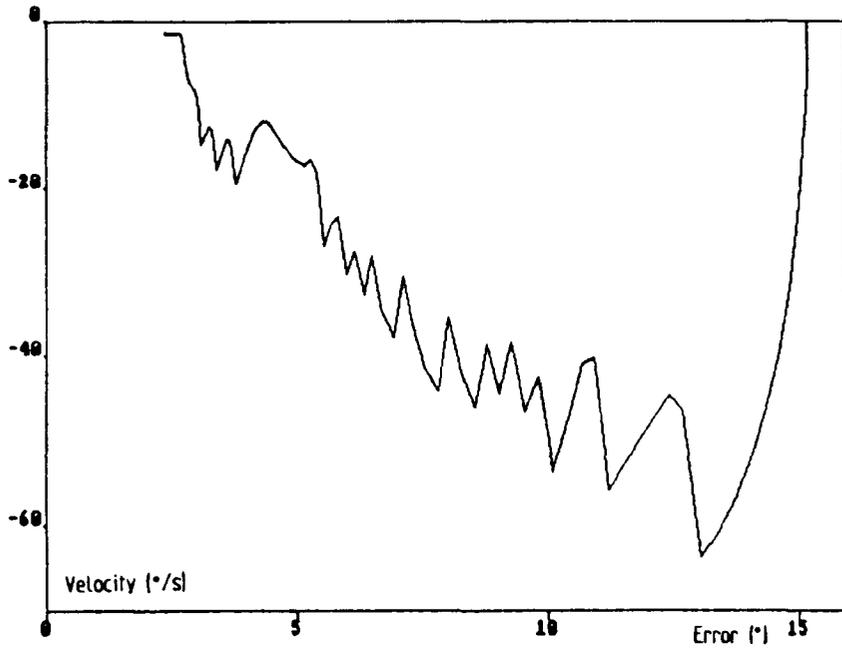


(i) 200Hz Sample Rate

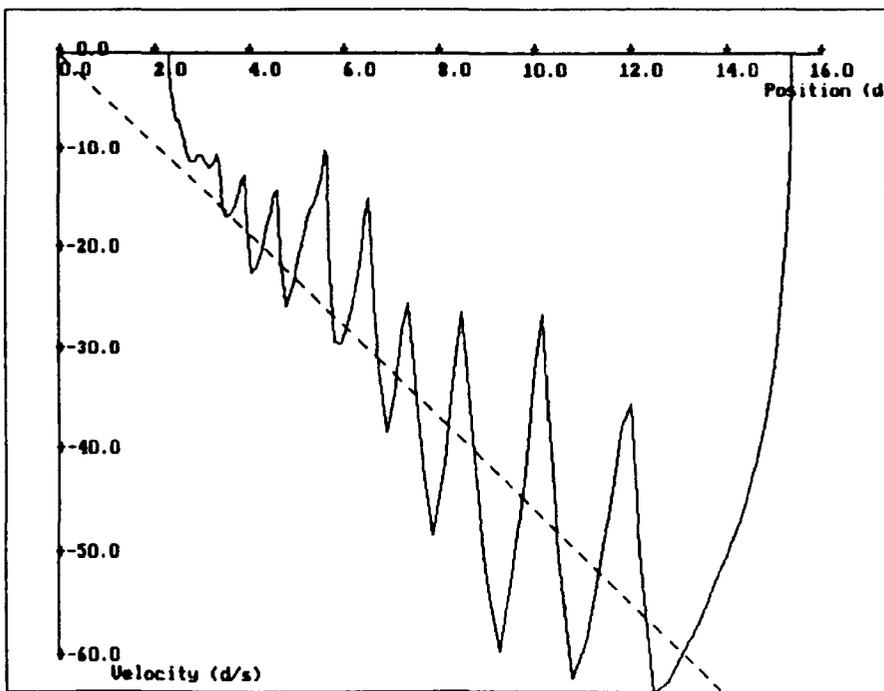


(ii) 500Hz Sample Rate

Figure 6.7 Comparison of simulation and experimental results for simple VSC



(i) Simulation



(ii) Experimental

6.2 Variable Structure Control with Boundary Layer

It was seen in the previous section that when a simple variable structure control system is employed on the RTX robot, a significant amount of chatter occurs around the switching line and, associated with this, is a large amount of control activity. Both of these features are undesirable and it is hoped, in designing the control system described in this section, that these effects will be eliminated.

The improvement to the basic VSC algorithm depends on the use of a boundary layer around the switching line in which the control signal undergoes a smooth transition from one structure to the other, rather than the abrupt change used in the simple VSC. As explained in section 2.1.5, there are a number of variations on the basic idea of variable structure control with boundary layer. The most simple form of this type of control employs a bang-bang control regime when the R.P. is outside the boundary layer and a control proportional to the distance from the switching line when it is within. The control signal is thus defined as:

$$u = \begin{cases} -l \operatorname{sgn}(s) & \text{if } |s| > k_b \\ k_s s & \text{if } |s| < k_b \end{cases}$$

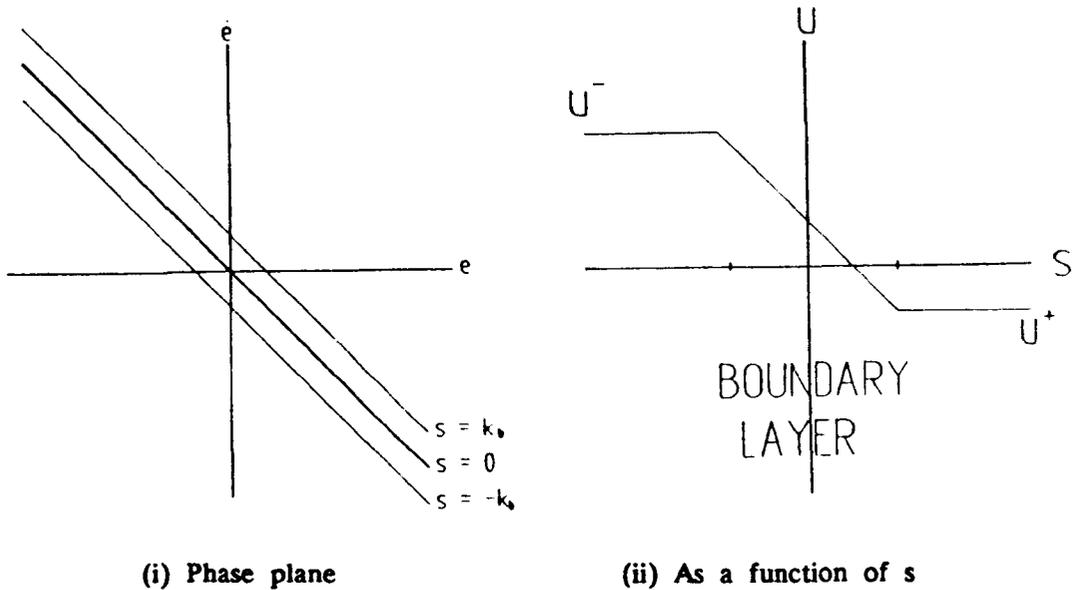
where the switching function, s is defined by equation 6.1, given in section 6.1. k_b represents the width of the boundary layer, k_s is a constant gain and l is a constant, usually set to the maximum control amplitude permissible. This form of control may be represented as shown in figure 6.8 which shows the control in terms of the phase plane position and the value of s .

It may be noted that if the equation for the control signal within the switching band is rewritten in terms of the position and velocity errors we obtain:

$$u = k_s (ce + \dot{e}) \quad \text{if } |s| < k_b$$

which is the same as that for a PD control system with the constants $k_p = k_s c$ and $k_d = k_s$. Although this control is effectively the same as a PD control this formulation gives an alternative insight which can be useful when designing controllers in the phase plane. This form of controller is also the basis from which a number of developments may be made.

Figure 6.8 Definition of the control signal in a VSC with boundary layer

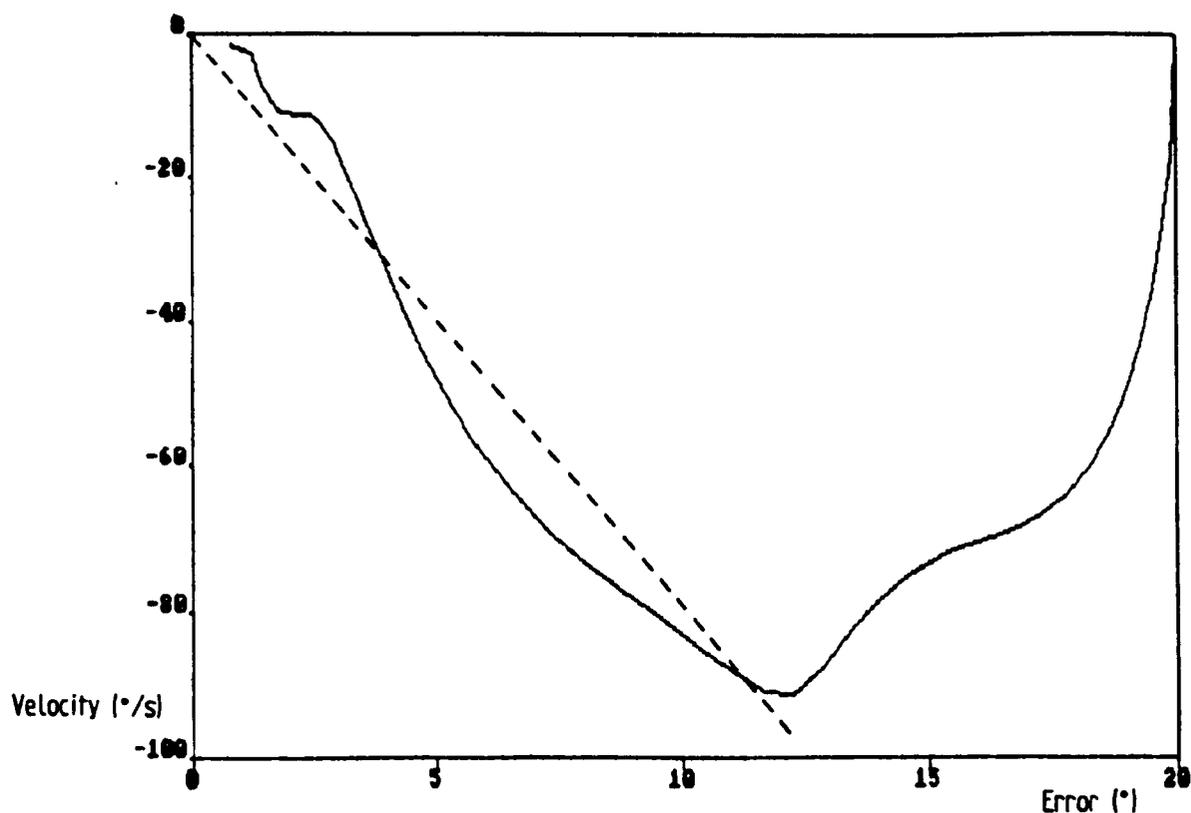


6.2.1 Computer Aided Design of VSC with Boundary Layer

Using the same procedure as outlined in section 6.1.1, a VSC with boundary layer was investigated using SIMNON. A typical result for a controller having $c = 8.0s^{-1}$, $k_b = 20.0s^{-1}$ and $k_s = 0.5V/s^{-1}$, applied to the elbow joint model, is shown in figure 6.9

It is seen that the chatter has been eliminated and the trajectory proceeds smoothly towards the origin, in the vicinity of the switching line, but falls 1.1° short, due to the Coulomb friction in the system.

Figure 6.9 Simulated performance of elbow joint for VSC with boundary layer

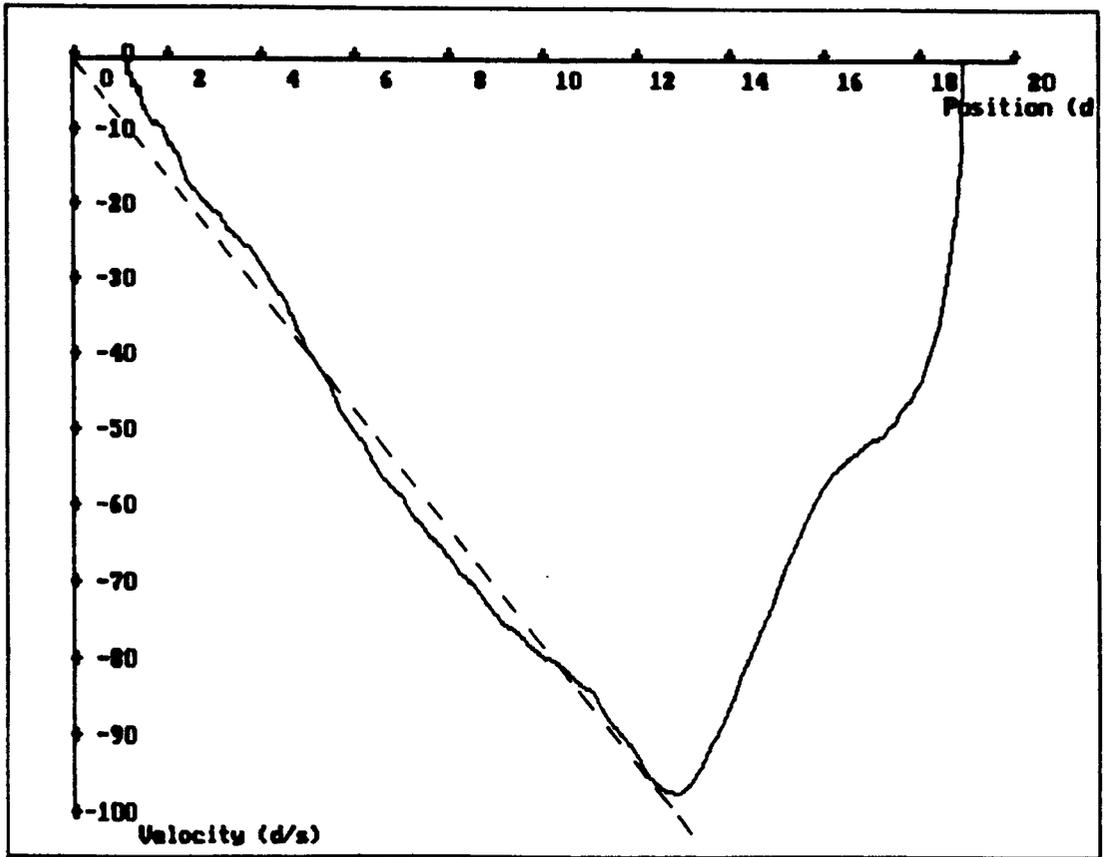


6.2.2 Implementation of a VSC with Boundary Layer

A control system based on this formulation was implemented and tested on both the shoulder and elbow joints of the RTX. The response of the elbow joint to a step change in position demand is shown in figure 6.10 for control parameters $c = 8\text{s}^{-1}$, $k_b = 20\text{s}^{-1}$ and $k_s = 0.50\text{V}/\text{s}^{-1}$. The trajectory is as expected, with the RP entering the boundary layer and remaining within, but deviating from the switching line itself.

Once again, due to the coulomb friction in the joints, there remains a steady state position error at the end of each movement, equal to approximately 0.5° and 1.0° for the shoulder and elbow joints respectively. These errors could be reduced by increasing the gain, k_s but a large increase leads to behaviour similar to the chattering found in the simple variable structure control.

Figure 6.10 Response of Elbow joint under VSC with boundary layer



It is worthwhile noting the effect of changes in plant parameters on the behaviour of this type of controller. Figure 6.11 shows the behaviour of the shoulder joint of the RTX with a controller having $c = 23.4s^{-1}$, $k_D = 4^\circ/s$ and $k_S = 1V/s^{-1}$. The three responses shown are for elbow angles of 0° , 90° and 135° and it can be seen that the behaviour during the reaching phase is significantly different but, when the sliding regime is entered, the trajectories become far more similar. These two forms of behaviour can also be seen when the position errors are plotted against time, as in figure 6.12.

Having investigated the performance of frequently studied variable structure control algorithms when applied to a system with significant nonlinearities, and having established the detrimental effects that these nonlinearities have, we will go on to consider techniques which may be adopted to eliminate these problems. This is the subject of the following section.

Figure 6.11 Effect of elbow angle on shoulder response

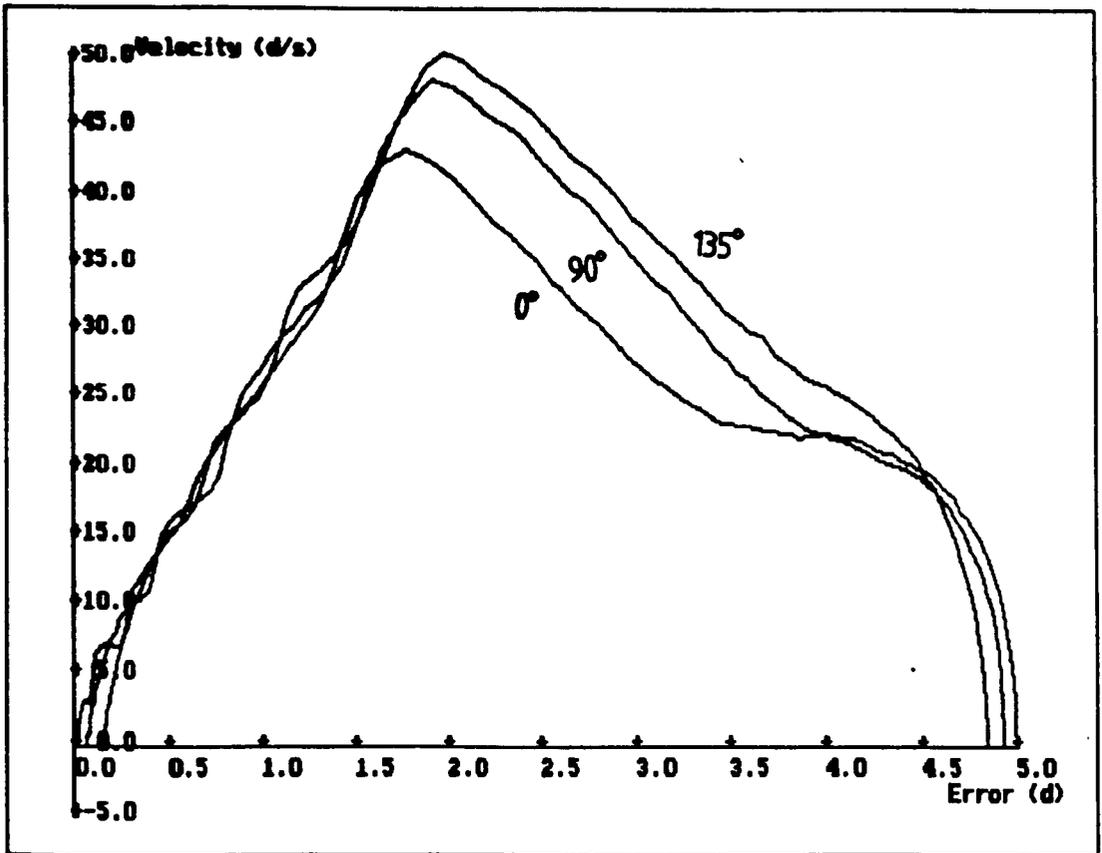
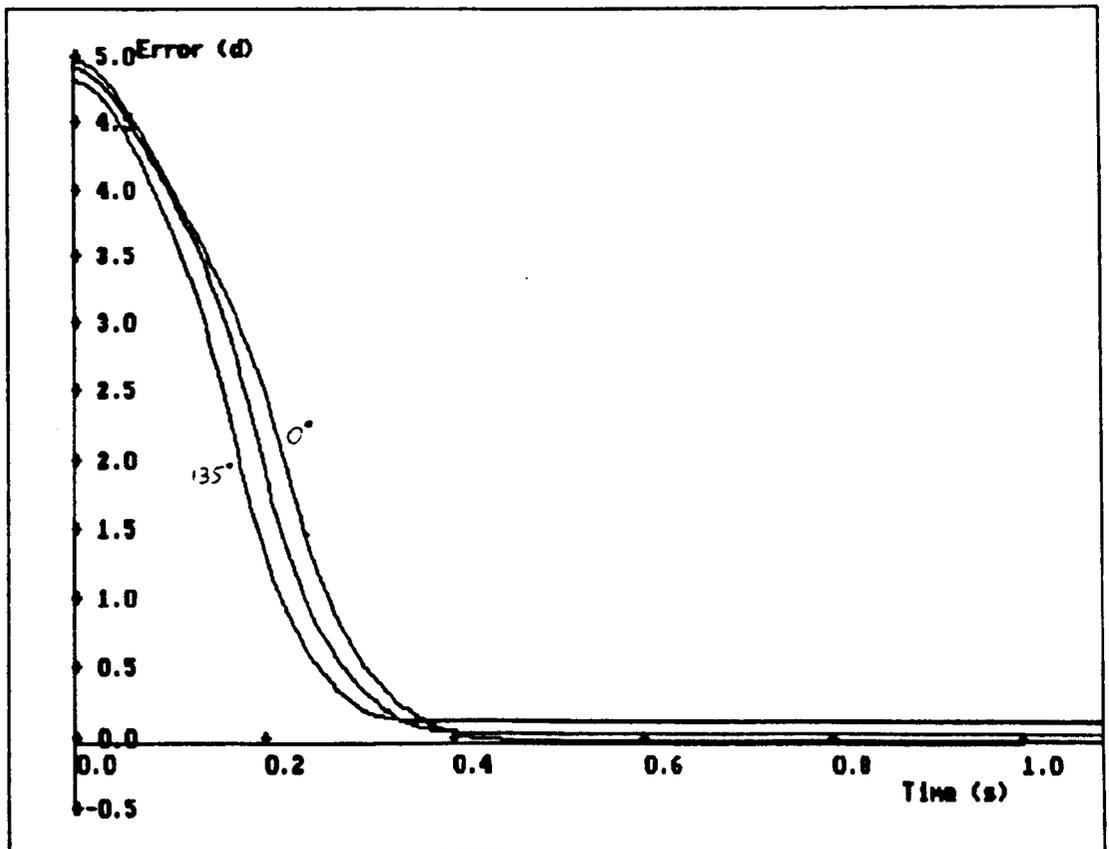


Figure 6.12 Effect of elbow angle on shoulder error response



6.3 Variable Structure Controller Design for Nonlinear Systems

The control systems described in the previous sections were developed without a great deal of consideration for the nonlinearities generally found in electro-mechanical systems, namely input saturation and Coulomb friction. In this section we will consider the implementation of controllers with integral action and friction compensation, mechanisms which provide a solution to the problems caused by Coulomb friction. A further development of the basic algorithm will be considered, which reduces the deviation of the trajectory from the switching function due to the acceleration and deceleration necessary to maintain the required trajectory. These techniques may be used individually or may be combined. Friction compensation will be considered first, followed by acceleration compensation and then the use of integral action will be investigated. The effect of using all these techniques will then be considered.

6.3.1 VSC with Friction Compensation

The variable structure controller with friction compensation considered in this section is based on the algorithm suggested by Slotine and described in section 2.1.5.3. The aim of this algorithm, in its original form, was to compensate for all the known dynamics of the plant (although only nonlinearities caused by joint interactions were considered and linear actuators were assumed) while eliminating the effects of unknown and varying dynamics by the use of a discontinuous control, added to the continuous control used in compensation.

For a dynamical system as complex as the RTX the full compensation control signal would be require extensive calculation, even neglecting the joint interaction behaviour, and this could not be achieved in real time. It was therefore decided to compensate only for the most significant components of the overall dynamics. This results in a relatively large discontinuous control

signal but, owing to the time varying and load dependant nature of the interaction dynamics, a large discontinuous signal would be required in any case.

As shown in section 4.2.1.2, friction in the joint transmission system of the RTX is of a highly nonlinear form and amounts to typically 35% of the maximum motor torque. Friction is the most significant, non-ideal element in the dynamics and is also the most straightforward to compensate for. Accordingly it was decided to implement a variable structure control system with friction compensation.

6.3.1.1 Simple VSC with friction compensation

The first controller to be considered is a basic VS controller, augmented with the inclusion of a friction compensation term. The control signal may be defined as:

$$u = \begin{cases} -\Psi x_1 - \hat{a}(x_2) & \text{if } sx_1 > 0 \\ \Psi x_1 - \hat{a}(x_2) & \text{if } sx_1 < 0 \end{cases}$$

where $\hat{a}(x_2)$ is the model of the true frictional relationship, $a(x_2)$. It is assumed here that the friction is a function of velocity only and not of joint position. Although this is not strictly true, the variation in friction with position is less than that which occurs with velocity and time, as shown in section 4.2.1.2. For the reasons explained in section 2.1.5.1, the estimate of the friction should take a lower value than the true friction, particularly at low velocities, in order to avoid limit cycle behaviour around the origin. The difference between the friction model and the true relationship governs the possible controller gains which will still ensure sliding, a poorer model requiring a larger gain, with the associated problems of chatter. In addition a poor model will lead to a relatively large steady state error, for a given gain.

Due to the variability of the friction relationship and the difficulty of obtaining a reliable model, it was decided to represent the friction as $\hat{a}(x_2) = f$

$\text{sgn}(x_2)$ where f is a constant selected to give a close correspondence with the measured relationship. Thus the frictional behaviour measured as described in chapter 2 and the model would typically take the form show in figure 6.13.

Using this form of friction compensation in a control loop around the elbow joint of the RTX, with the friction compensation level set at 3.4V, corresponding to 1.77Nm, and a gain of 1.0V/°, the trajectory shown in figure 6.14 was produced. This may be compared with the response using the same control system parameters but with no friction compensation, shown in figure 6.15.

Figure 6.13 Frictional relationship at elbow joint and simple friction model

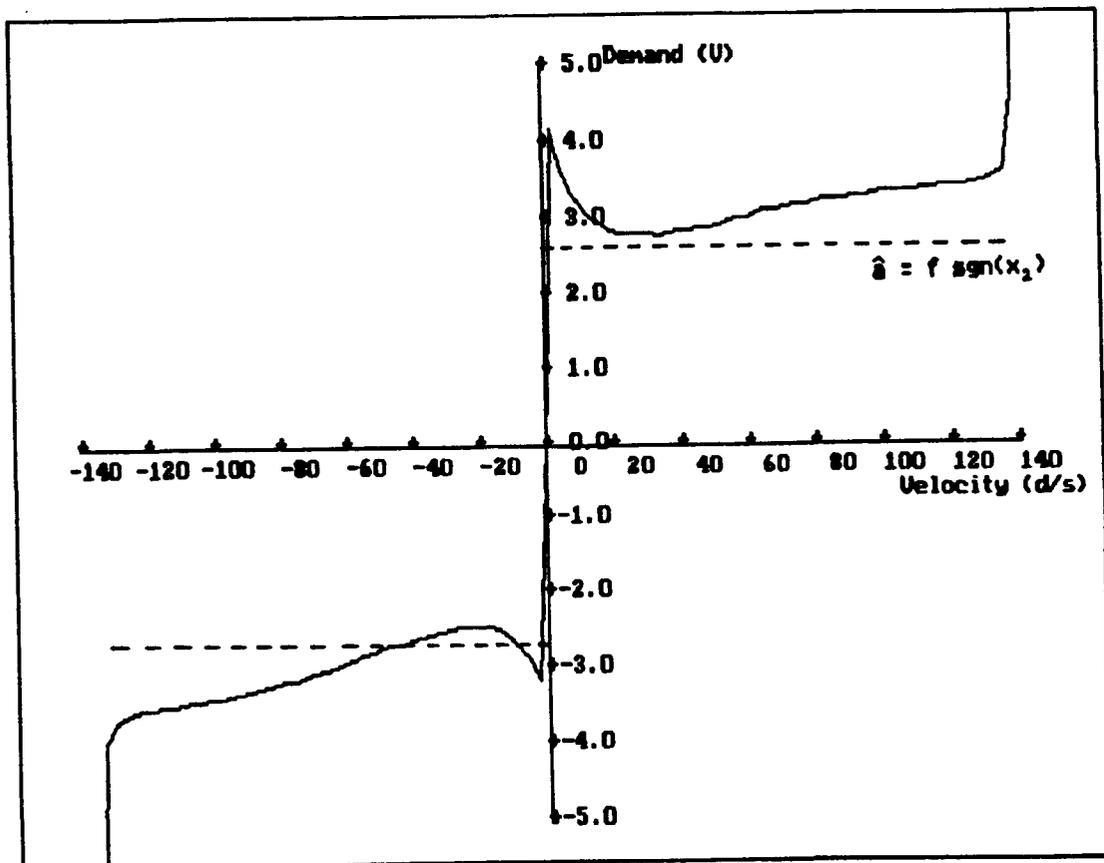


Figure 6.14 Response of Elbow joint with friction compensated controller

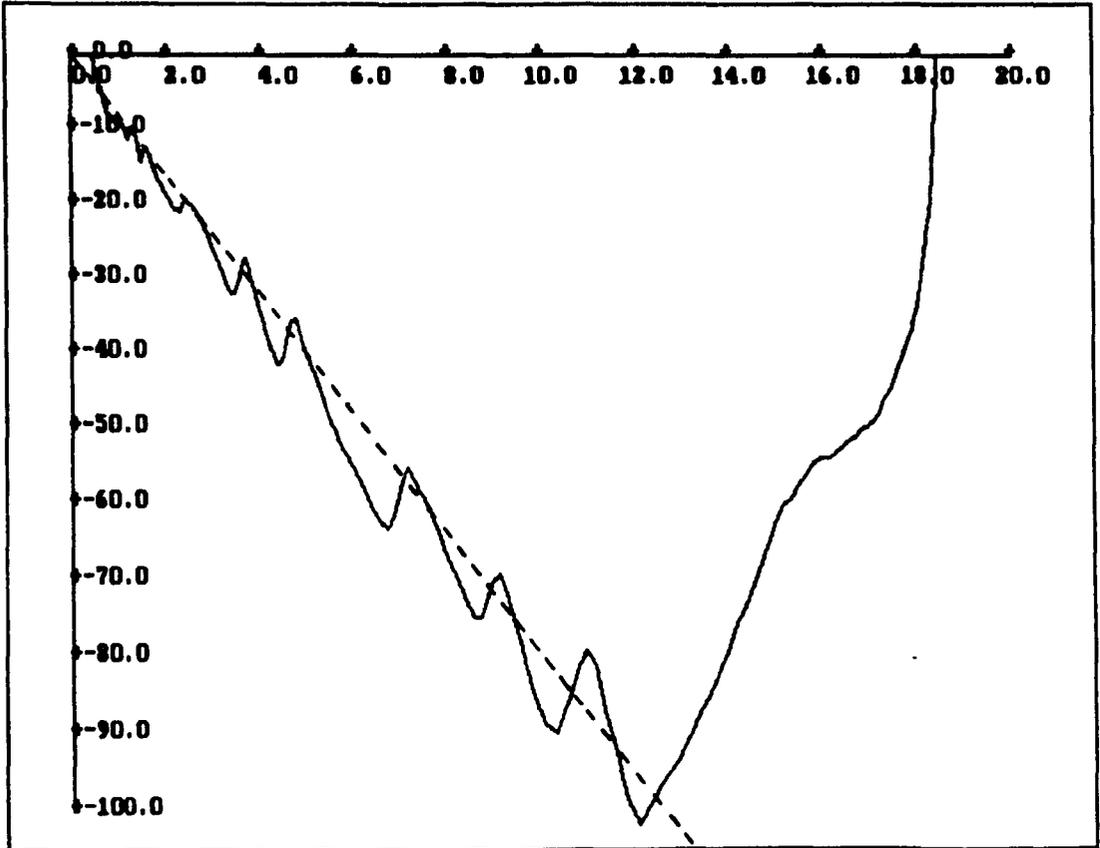
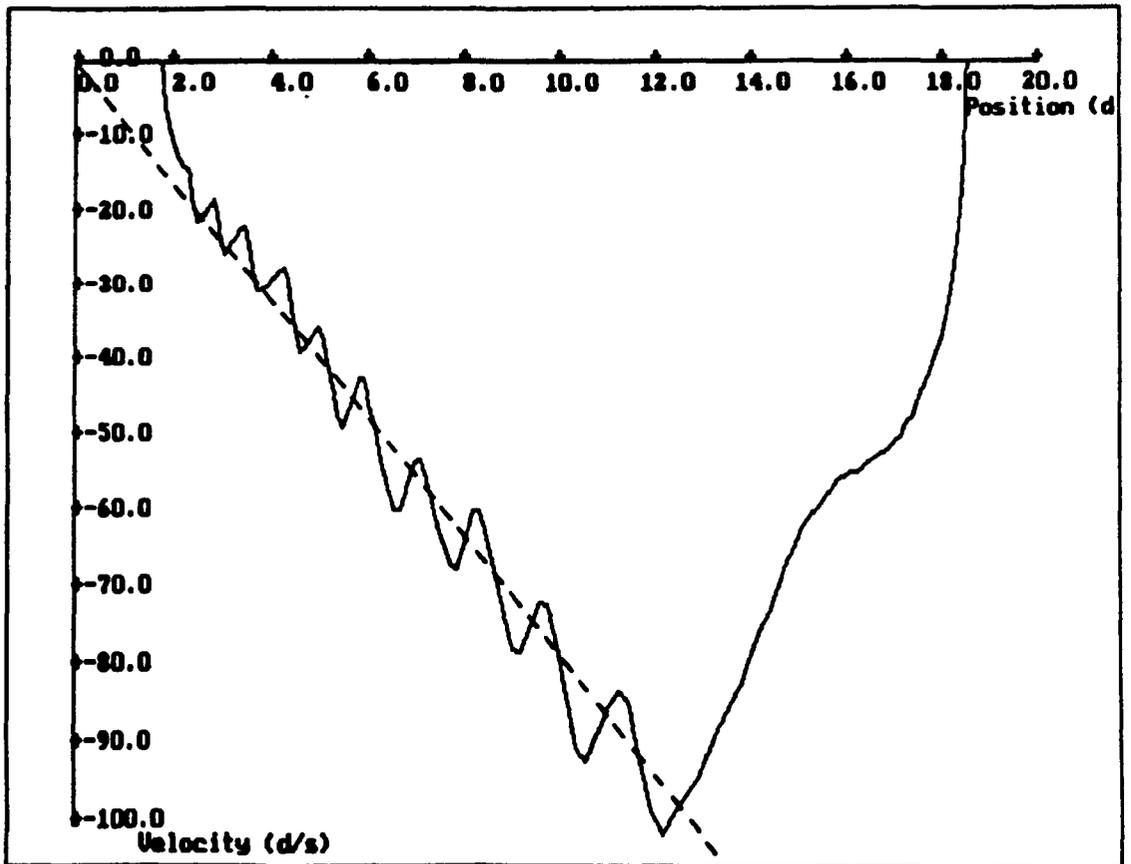


Figure 6.15 Response of Elbow joint without friction compensation



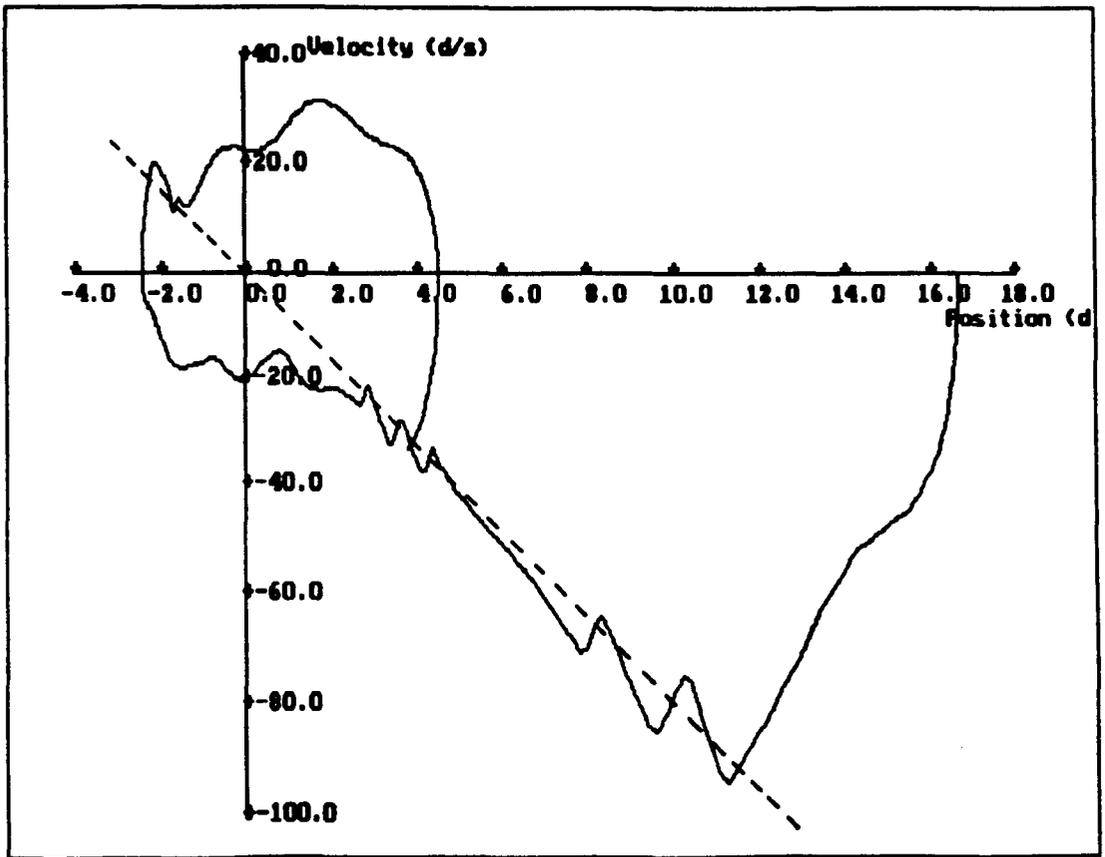
It is clear that chatter still remains with the friction compensated controller but the steady state error has been reduced to 0.47° in figure 6.14, compared with 1.72° for the system with no friction compensation, shown in figure 6.15.

It is worthwhile noting the different gradients of the trajectories under the control structures with and without friction compensation. In Figure 6.15, representing the case without friction compensation, the sub-trajectories when under positive feedback make a larger angle with the switching line than in the case with compensation, where these sub-trajectories make an obtuse angle, as seen in figure 6.14. This may be taken as an indication that the sliding condition is nearer to violation in the friction compensated case than in the uncompensated situation.

This point is further illustrated in figure 6.16 which shows the situation where the friction compensation is raised to a level of 3.9V or 2.03Nm which is above the measured value of friction. It will be seen that sliding ceases in the region of 6° to 8° position error, is briefly re-established but then breaks down completely, leading to large overshoot and the beginnings of limit cycle behaviour.

The reason for this highly undesirable behaviour is that, when the position error approaches zero, the control signal without friction compensation must be zero, independent of the velocity. When the friction compensation is included, this is the only signal acting on the system. Hence, if the friction compensation is above the true friction, the link will accelerate, rather than decelerating as required. It is clear then that some modification to the control algorithm is required, if the steady state error is to be removed, without the risk of overshoot. One such modification is the inclusion of a boundary layer around the switching line.

Figure 6.16 Response for simple VSC with excess friction compensation



6.3.1.2 VSC with Boundary Layer and Friction Compensation

Although the use of friction compensation is partially effective in removing steady state position errors, when used as part of a simple VSC system it does not cause a large reduction in the chatter amplitude. In order to achieve this chatter reduction we may adopt the smoothing technique described in section 6.2 while including friction compensation to reduce the steady state error.

Using the same nomenclature as in section 6.2 and 6.3.1.1, we may write the control signal used in this form of control as:

$$u = \begin{cases} -1 \operatorname{sgn}(s) & \text{if } |s| > k_b \\ k_s s - \hat{a}(x_2) & \text{if } |s| < k_b \end{cases} \quad (6.2)$$

Once again we will consider the greatly simplified friction model, given by $\hat{a}(x_2) = f \operatorname{sgn}(x_2)$. It should be noted that, since the linear part of the control signal is no longer a linear interpolation between two values which are less than or equal to the input saturation level, but are augmented by the friction

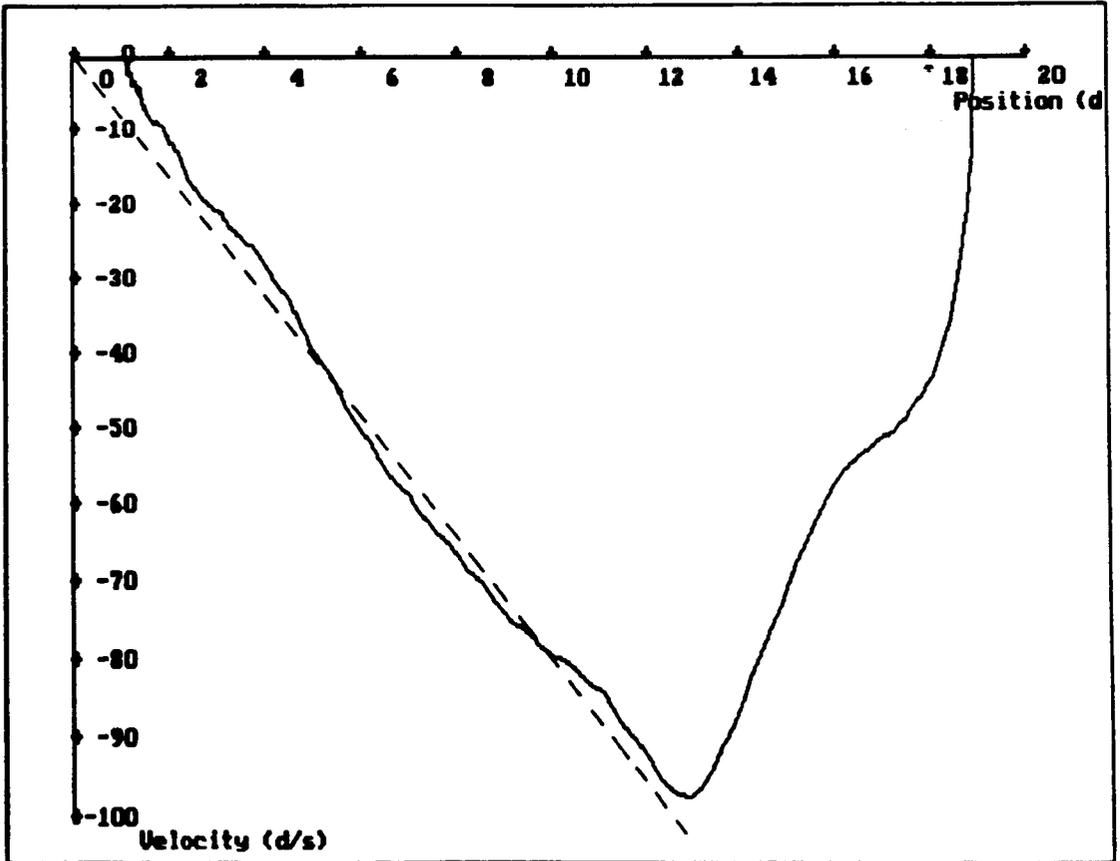
compensation term, the control described by equation 6.2 may give a value of u above the saturation value. Thus the control signal must be limited according to:

$$u' = \begin{cases} u_{\text{sat}} \operatorname{sgn}(u) & \text{if } |u| > u_{\text{sat}} \\ u & \text{if } |u| \leq u_{\text{sat}} \end{cases}$$

where u_{sat} is the input saturation level.

A number of experiments were carried out using this algorithm, involving a variety of control parameters and friction compensation levels. For comparison purposes, the response using a controller without friction compensation is shown in Figure 6.17. This is for the elbow joint of the RTX with a controller having a switching line constant of $c = 8 \text{ s}^{-1}$ as in section 6.3.1.1, a boundary layer width of 20 s^{-1} and a gain of $k = 0.50 \text{ V/s}^{-1}$.

Figure 6.17 Response for controller with boundary layer and no friction compensation



It will be seen that the trajectory remains close to the switching line for the majority of the trajectory, without chatter, but leaves the switching line as the origin is approached, leaving a steady state error of 1.2° . By introducing friction compensation at a level of 4.9V, equivalent to a torque of 2.54Nm, this response is modified to that shown in figure 6.18.

The modified system does not remain as close to the switching line as in figure 6.17 for reasons which will be explained shortly, but the trajectory does approach the origin much more closely, giving a steady state error of 0.06° . It is worth considering, at this stage, the effect of using a friction compensation above the true friction level. This situation is illustrated in figure 6.19, in which a compensation level of 7.33V or 3.82Nm was used.

Figure 6.18 Response for controller with boundary layer and friction compensation

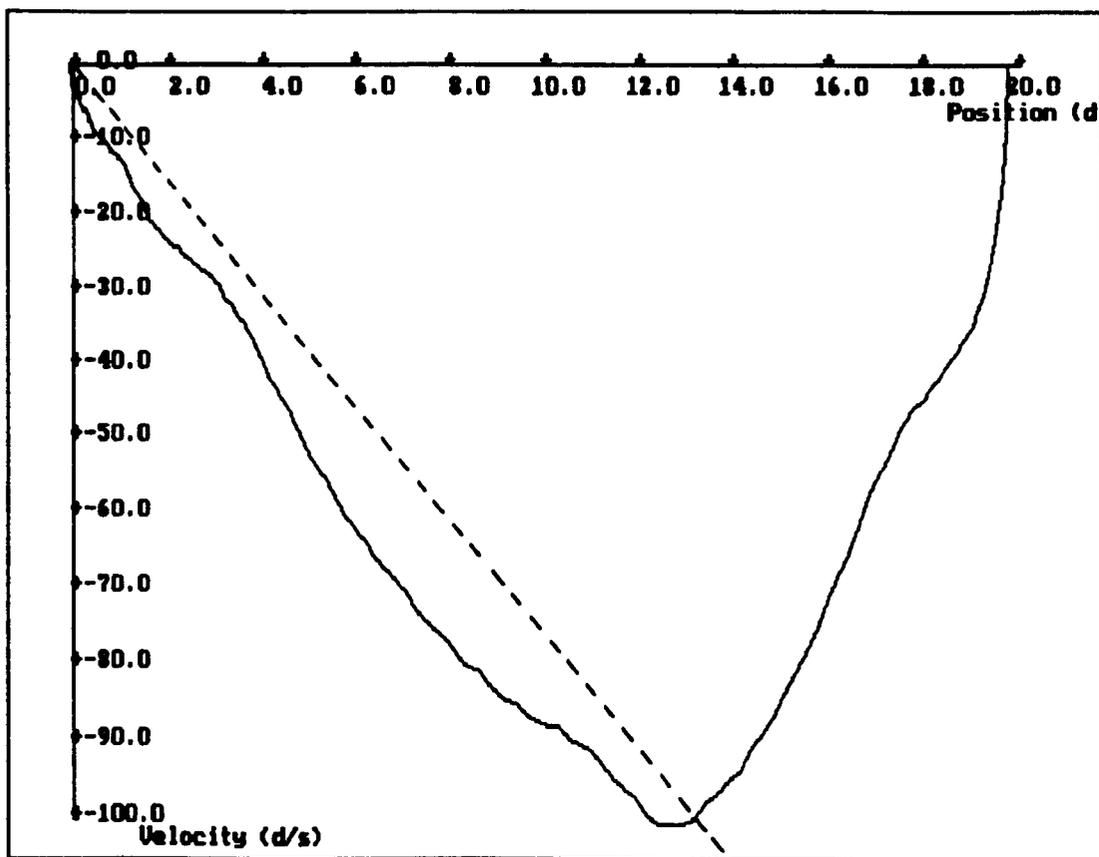
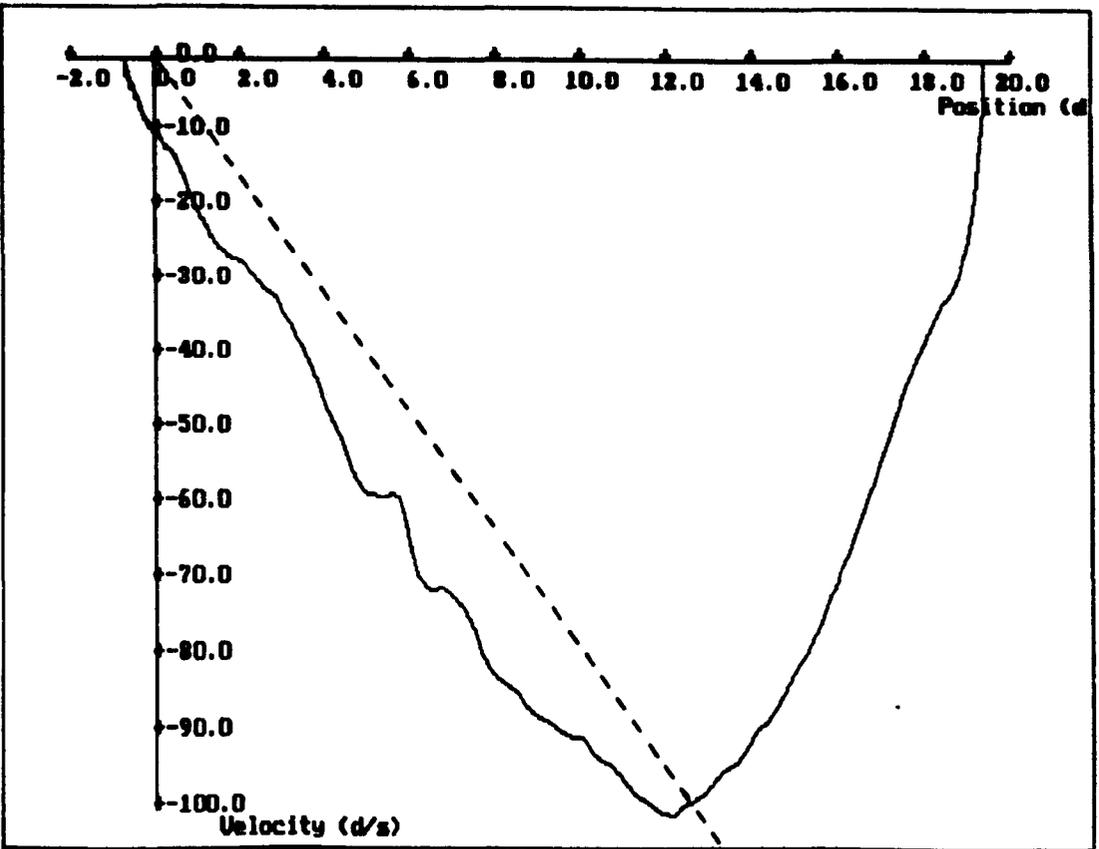


Figure 6.19 Response for controller with over-compensation of friction



This trajectory is seen to overshoot the origin by $0.68'$ but there is no oscillation of the form found with the simple VS controller. Thus, although the overshoot may be undesirable, the effect of a poor friction estimate is not as bad as in the case of a simple VSC.

6.3.2 VSC with Acceleration Compensation

The reason for the deviation of the trajectory from the switching line shown in figure 6.17 is that, in order for the RP to follow the switching function, a deceleration must be induced into the system. This fact is not accounted for in the controller definition and so, when on the switching line, if the friction compensation level is correct, the total effective torque will be zero and a constant velocity will be maintained. In order to attain the required deceleration, the trajectory must deviate from the switching line by an amount which causes the required torque to be applied. In the case of zero friction compensation, illustrated in figure 6.17, the deceleration due to friction is sufficient to keep the trajectory in the vicinity of the switching line for the

majority of the transient, although as the origin is approached, the required deceleration drops and so the friction causes an excess deceleration, leading to the position shortfall. When friction compensation is included the deviation of the trajectory from the switching line is large, particularly at high velocities where the greatest deceleration is required.

In a second order system, it is a simple matter to determine the acceleration required to maintain the trajectory on the switching line, given the gradient of that line. The acceleration is given by:

$$\frac{de_2}{dt} = \frac{de_2}{de_1} \frac{de_1}{dt} = -ce_2$$

If we consider a simplified second order plant model, in which the effect of nonlinear friction has been eliminated by the compensation mechanism described previously, the plant dynamics are described by

$$\dot{e}_1 = e_2$$

$$\dot{e}_2 = bu$$

then the required acceleration is caused by an input of:

$$u = -ce_2/b$$

Adding this factor into the existing control definition, we obtain:

$$u = \begin{cases} -1 \operatorname{sgn}(s) & \text{if } |s| > k_b \\ k_s s - \hat{a}(e_2) - ce_2/\hat{b} & \text{if } |s| \leq k_b \end{cases} \quad (6.3)$$

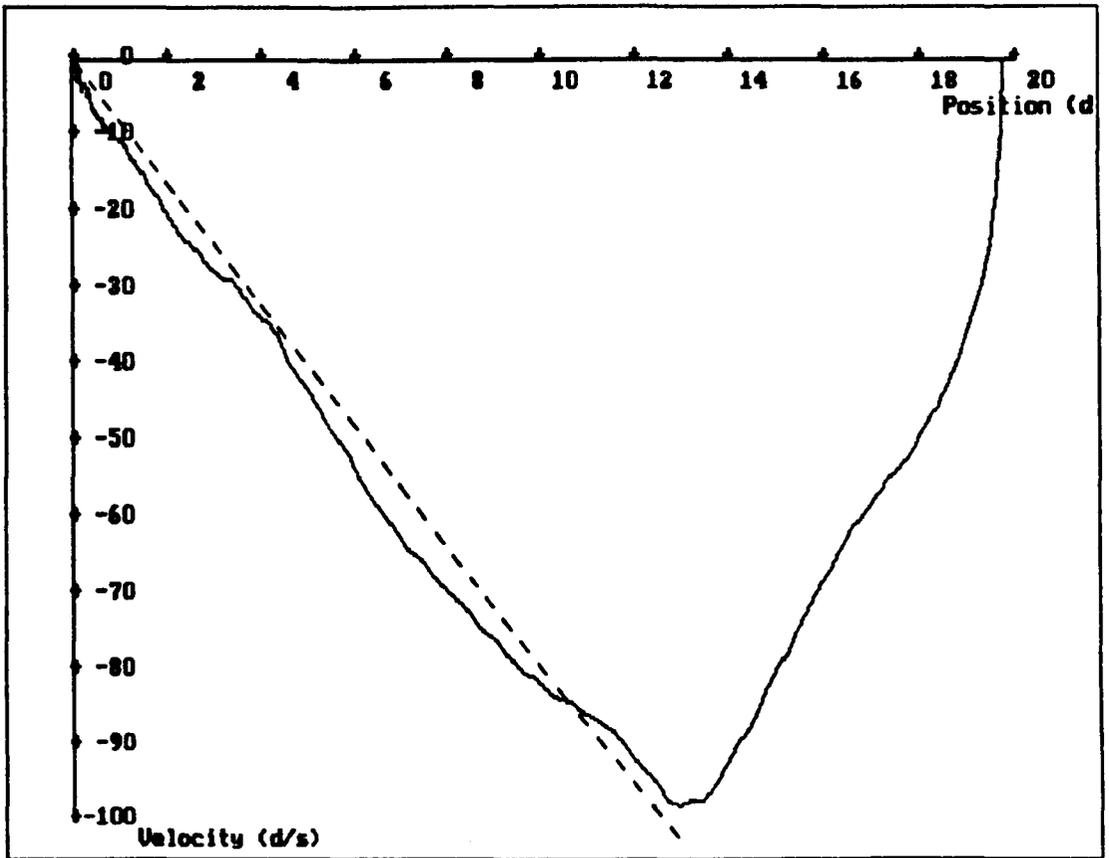
where \hat{b} is an estimate of the input gain, b .

Once again, this control signal must be limited to prevent overdriving of the motors as described in section 6.3.1.2.

Applying this algorithm to the elbow joint of the RTX, with the common

parameters the same as those used to generate figure 6.18 and a value of $1/\hat{b}$ of $0.006V/s^{-2}$, the result shown in figure 6.20 is obtained. From this it will be seen that a significant reduction in the deviation from the switching line has been obtained, although some deviation remains, due to the simplified nature of the friction model used.

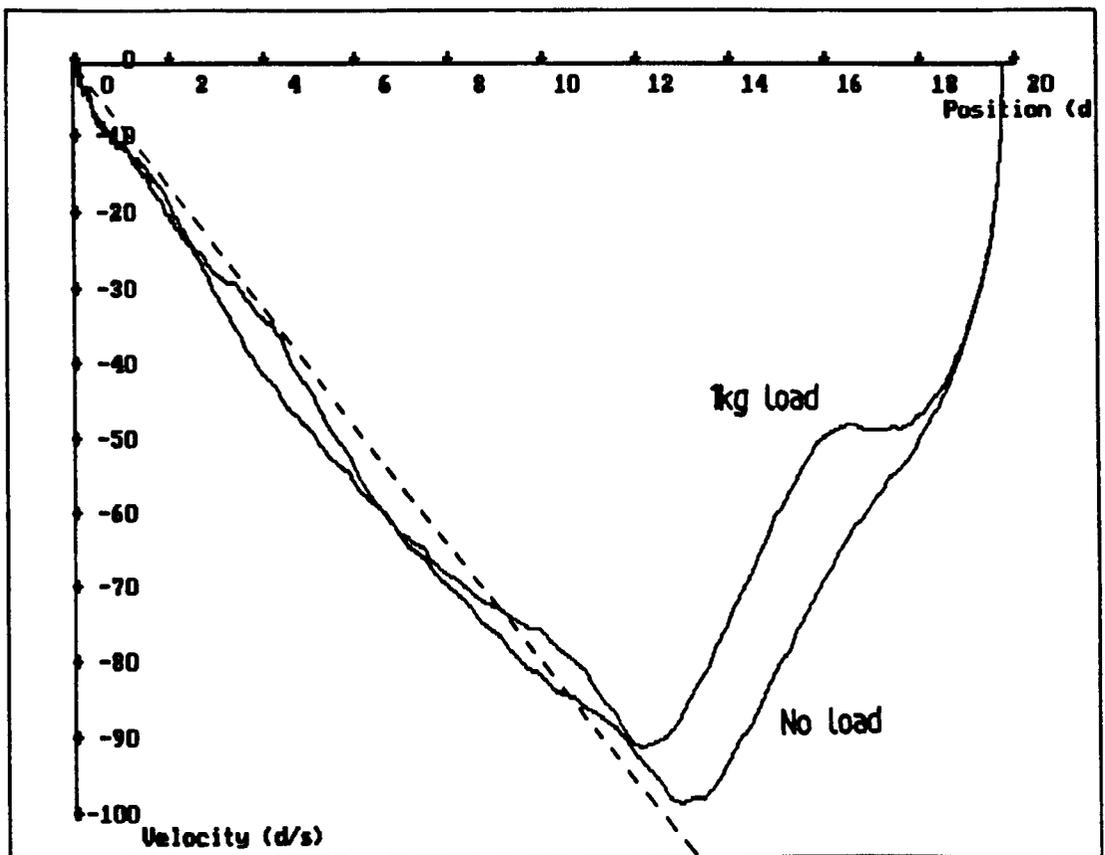
Figure 6.20 VSC with boundary layer and friction and acceleration compensation



It would be possible to reduce the switching line deviation further in this situation by the use of a more accurate friction model, with friction values stored as a piecewise linear function of velocity. There is however a limit to the extent of these improvements, due to the variable nature of the robot dynamic parameters due to configuration and load changes. In particular, the acceleration compensation algorithm given by equation 6.3 assumes a constant input gain, b . Clearly this is not the case and in practice, some variation in this parameter will occur. In the absence of any reliable estimate of this parameter, some error in the acceleration compensation scheme is unavoidable.

As an instance of this problem, the experiment illustrated in figure 6.20 was repeated with a 1kg load attached to the gripper mounting plate. The resulting phase trajectory is shown in figure 6.21, along with the result for no load. It is seen that the added mass has a significant effect on the dynamics in both the reaching phase and the pseudo-sliding phase. The maximum overall deviation from the switching line remains approximately the same but the position of maximum deviation varies due to the different reflected torques from the arm.

Figure 6.21 Effect of load mass on the friction and acceleration compensated controller



Although friction compensation provides a means of eliminating a significant proportion of the steady state error caused by Coulomb friction and acceleration compensation eliminates some of the deviation from the switching line due to the required deceleration any compensation scheme must have limited success in the presence of varying and unknown dynamics and

disturbances. It is therefore necessary to consider alternative disturbance and modeling error rejection techniques. One such method, the use of integral action, is the subject of the following section.

6.3.3 VSC with Boundary Layer and Integral Action

As described in section 2.1.6.2, integral action may be added to the algorithm for variable structure control with a boundary layer, in such a way that a second order stable filter is introduced into the dynamics of s with an input constituted of the effective modeling error and external disturbances. Due to a zero in the numerator of this filter, any steady state disturbances and modeling errors are rejected.

A simple version of algorithm, in which dynamic compensation is not included, may be written as:

$$u = \begin{cases} -1 \operatorname{sgn}(s) & \text{if } |s| > k_b \\ k_s s + k_i \int s dt & \text{if } |s| \leq k_b \end{cases}$$

A controller of this form was implemented on the elbow joint of the RTX. Initially, the same switching line definition and gain as used in the previous two sections were adopted to allow a comparative evaluation of the effect of integral action.

As seen in figure 6.22, the addition of a moderate amount ($k_i = 37.2V/^\circ$) of integral action leads to a reduced deviation from the switching line, following the initial overshoot, and a significant reduction in steady state error (0.01° instead of 0.87°) compared with the same system without integral action. It will also be seen that an increase in activity around the origin occurs; an undesirable feature not found with the friction compensation described in section 6.3.1.2.

If the integral constant is increased to $k_i = 74.4V/^\circ$ oscillation around the switching line is caused, as shown in figure 6.23. The steady state error is small (0.003°) but clearly this type of behaviour is undesirable.

Figure 6.22 VSC with boundary layer and integral action ($k_i = 37.2V/^\circ$)

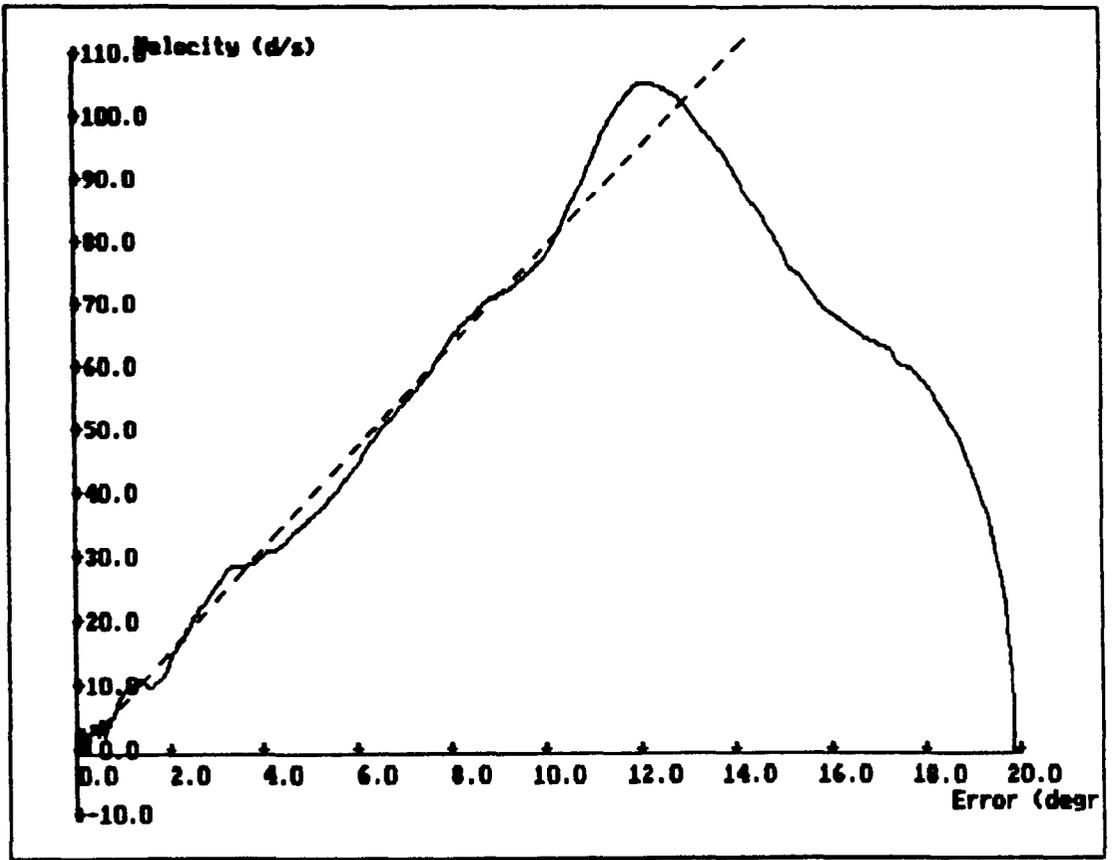
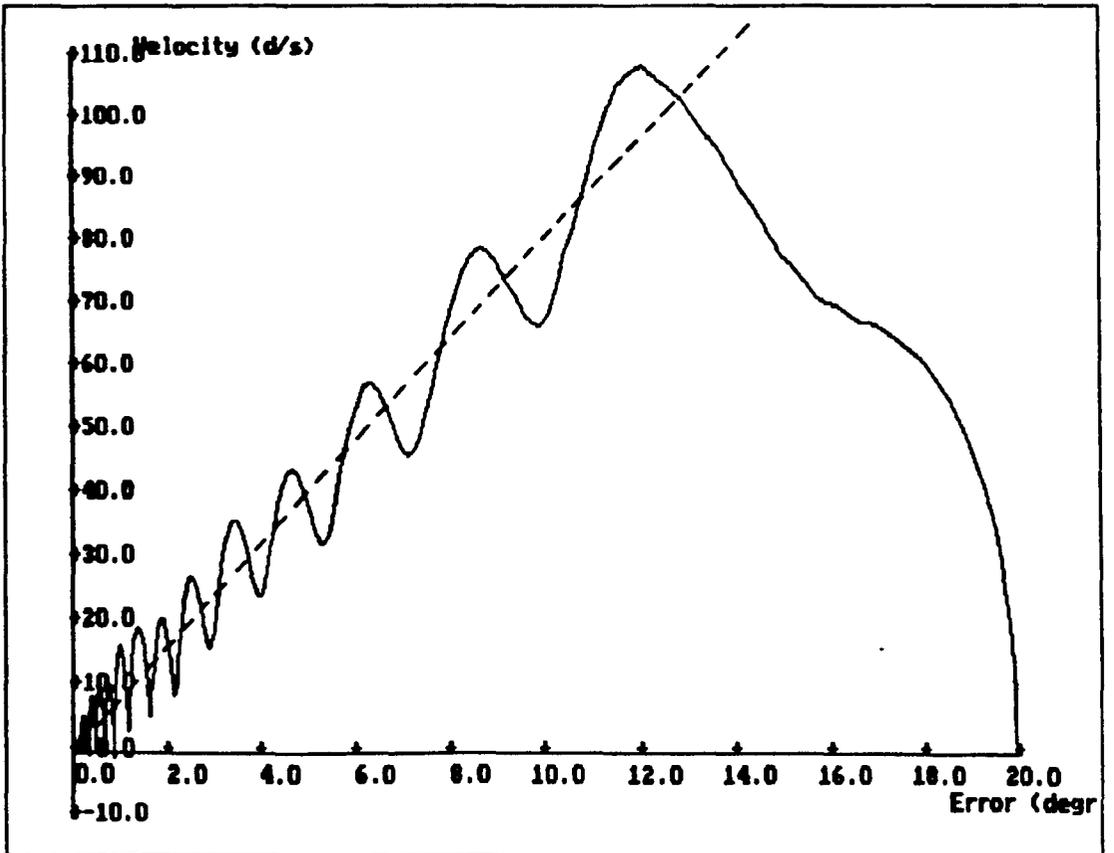


Figure 6.23 VSC with boundary layer and integral action ($k_i = 74.4V/^\circ$)

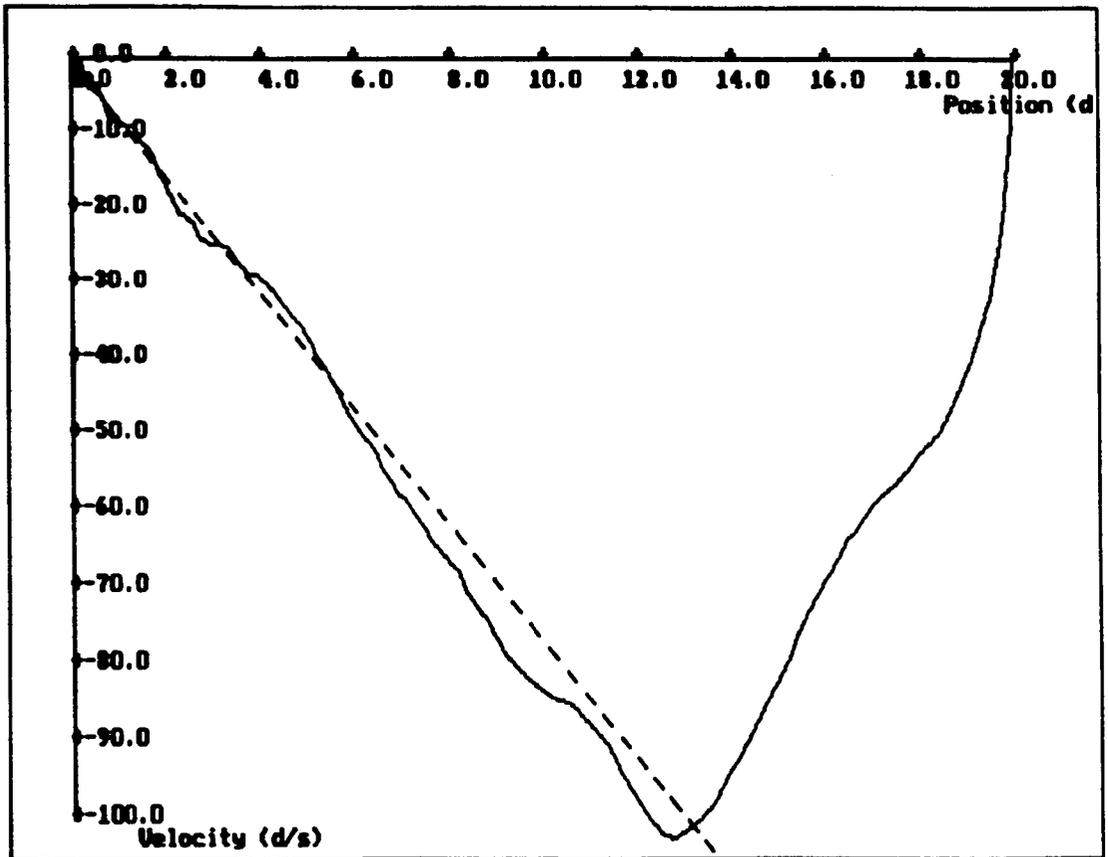


The addition of integral action to the control algorithm provides a means of removing the effects of steady disturbances and reducing the effect of Coulomb friction. It does however have a destabilising effect which means that the use of this component must be limited. This in turn means that the time taken for the correcting action to be effective is longer than would be hoped, making it ineffective in overcoming the problems caused by higher frequency disturbances. This is seen in Figure 6.22, in which the integral action does not begin drawing the trajectory towards the switching line until it has passed it for a significant time. Thus although integral action does give some significant benefits, it is not wholly effective in overcoming the problems of a simple VSC with boundary layer. It is therefore necessary to consider the effect of combining integral action with the compensation techniques described in sections 6.3.1 and 6.3.2.

6.3.4 VSC with Boundary Layer, Compensation and Integral Action

It is a simple matter to combine the effects of the compensation and integral algorithms described in the previous sections and, in the case of the compensation terms, the same parameters may be adopted. For the integral action, it is no longer necessary to adopt as high a component as required when used alone, since the disturbance and nonlinear effects it is used to overcome are to a large extent removed by the compensation terms. This has the advantage that the high activity in the region of the origin, described in section 6.3.3, is mostly eliminated. When implemented on the RTX, the results for a step input change are of the form shown in figure 6.24. This result, obtained using the elbow joint, is for the case of 4.4V friction compensation, $0.006\text{V}/\text{s}^2$ acceleration compensation and an integral component of $20.0\text{V}/\text{s}$, while the remaining parameters were the same as used in previous sections.

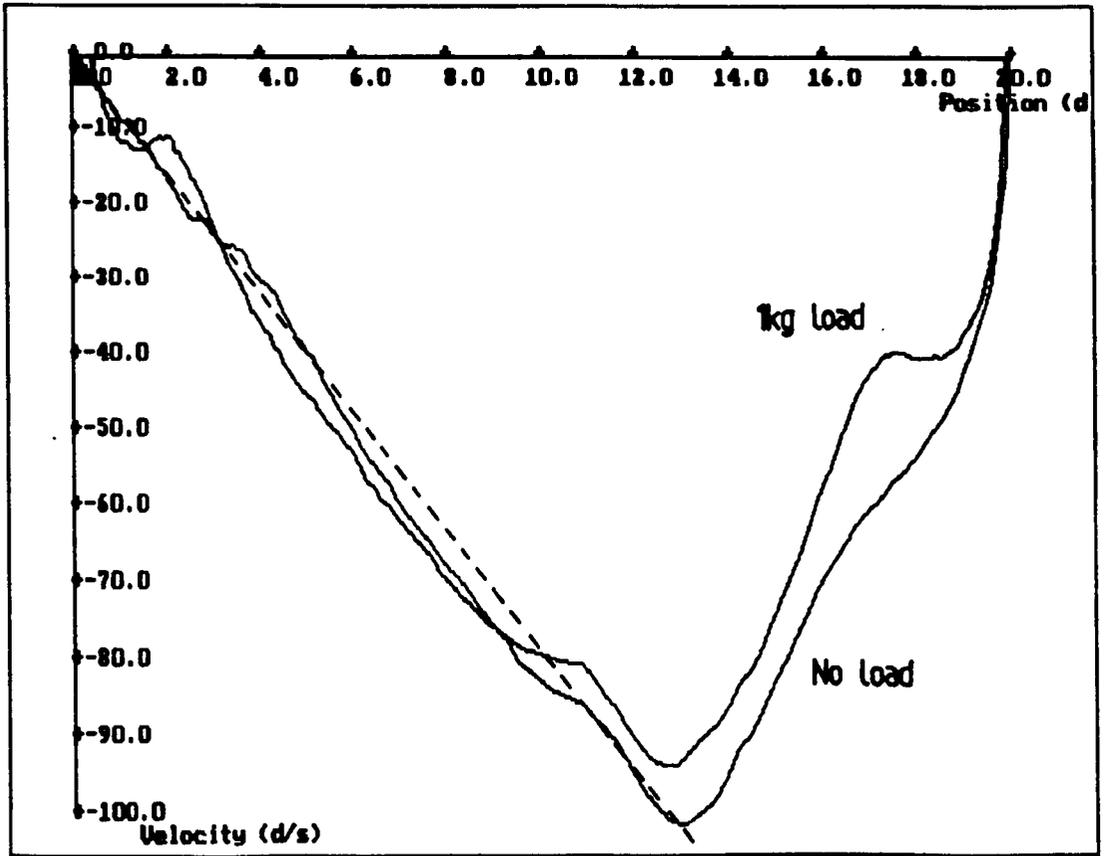
Figure 6.24 VSC with compensation and integral action



Clearly the deviation from the switching line has been further reduced while the steady state error has been lowered to 0.0045° . Since the control signal adopted here is largely composed of compensation terms, with only a small integral term used to eliminate any additional effects, it is natural that the response should be sensitive to changes in those elements in the system which are being compensated for. This point is illustrated in figure 6.25, in which the same control system was employed for the robot with and without a 1kg load fitted.

Clearly the performance is somewhat degraded, due to this change in plant parameters, but remains better than that achieved with the previously described algorithms.

Figure 6.25 VSC with compensation and integral action, with 1kg load



It should be noted that the results described in the previous sections involved fixed control parameters, so as to provide a common basis for comparing performance, rather than being optimised for each form of control. It is possible to obtain a better performance, in terms of speed of response and switching line tracking, by a careful selection of parameters to suit the particular form of algorithm, as seen in section 6.7.2. This was not done in all cases since the optimised parameter values for one algorithm would not give stable control for another and so could not be used for comparison purposes.

6.4 Variable Structure Control with Piecewise Linear Switching Curves

It will be noted from the previous sections that the phase trajectories of the RTX joints under a given form of VS control differ widely between the case of no load and when a load is applied. The most significant difference occurs during the reaching phase, in which the system is driven with the

maximum input torque and gives the full acceleration possible. During this phase, the response is highly dependant on plant parameter changes. As explained in section 2.1.3.2, this lack of invariance during the reaching phase may be undesirable in some situations, and so a number of approaches have been considered to avoid this behaviour.

The most promising is the use of a curved or piecewise linear switching function, starting from the initial position and going to the origin, so as to form a single curve along which sliding may occur. This curve may take many forms, but the most generally applicable type consists of a region of linear acceleration, a constant velocity phase, a linear deceleration and a final exponential deceleration section, as described in section 2.1.3.2. An algorithm of this form was implemented on the RTX in a way that allowed variation of each of the switching line parameters individually. It would be possible to calculate the equation of the switching function as a part of the control algorithm but, owing to timing constraints, it was considered more appropriate to pre-calculate the switching line definition and store the data as two arrays, one of switching line gradients, c , and the other of offset velocities, v_{off} , as functions of position error, so that the value of s may then be calculated from:

$$s = c(e) e + \dot{e} + v_{\text{off}}(e)$$

This method of storage results in a piecewise linear switching function which, by making the position quantization level sufficiently small, (0.1° , 0.2° and 0.15° for the shoulder, elbow and yaw joints respectively) gives a close approximation to a smooth curve. The gradient and offset values for each of the sections of the curve shown in figure 6.26 may be calculated using the equations given in table 6.1.

Figure 6.26 Piecewise linear switching curve

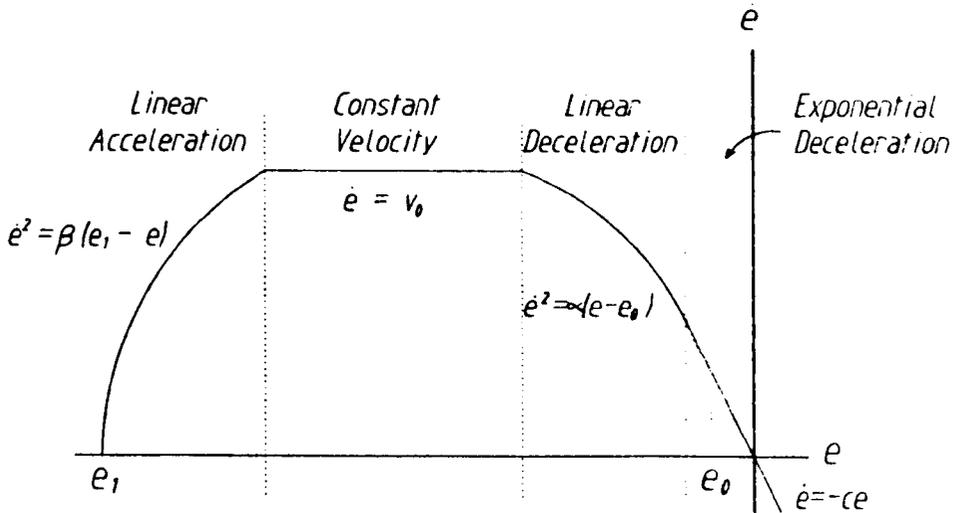


Table 6.1 Calculation of gradients and offsets for piecewise linear switching curve

Region	Behaviour	c	Offset velocity
1	Exponential decel.	c_0	0
2	Linear deceleration	$\sqrt{\alpha}/(4(e - e_0))$	$\sqrt{\alpha}(e - e_0) - ec(e)$
3	Constant Velocity	0	v_0
4	Linear acceleration	$-\sqrt{\beta}/(4(e - e_1))$	$\sqrt{\beta}(e - e_1) - ec(e)$

It should be noted that the absolute value of e_0 should be made a little larger than the absolute value of the initial position error since, if this is not the case, a small disturbance may move the RP to a region in which the switching function is not defined, leading to unpredictable behaviour. This adjustment need only be of the order of 0.1° to assure that the correct sliding behaviour begins.

The calculation of the gradient and offset tables was carried out in such a way that, if there were any overlap between regions, the following order of precedence would be followed:

- 1 - Exponential deceleration
- 2 - Linear deceleration
- 3 - Linear acceleration
- 4 - Constant velocity

Thus, the exponential deceleration would always be present, while the remaining parts would only be used if required to complete the curve.

This form of switching function was tested with each of the forms of VSC described in previous sections, with the same control and switching line parameters, to allow comparison between the different control algorithms. The switching curve parameters, α and β used were selected to give a fast acceleration and deceleration while still being maintainable under worst case loading conditions; v_0 was chosen to give a speed about the middle of the attainable speeds while c_0 was selected to be compatible with previous tests and the limit of the linear switching function region was chosen, through experiments, to give a smooth final approach to the origin. The values selected were:

$$c_0 = 8s^{-1}$$

$$e_c = 2^\circ$$

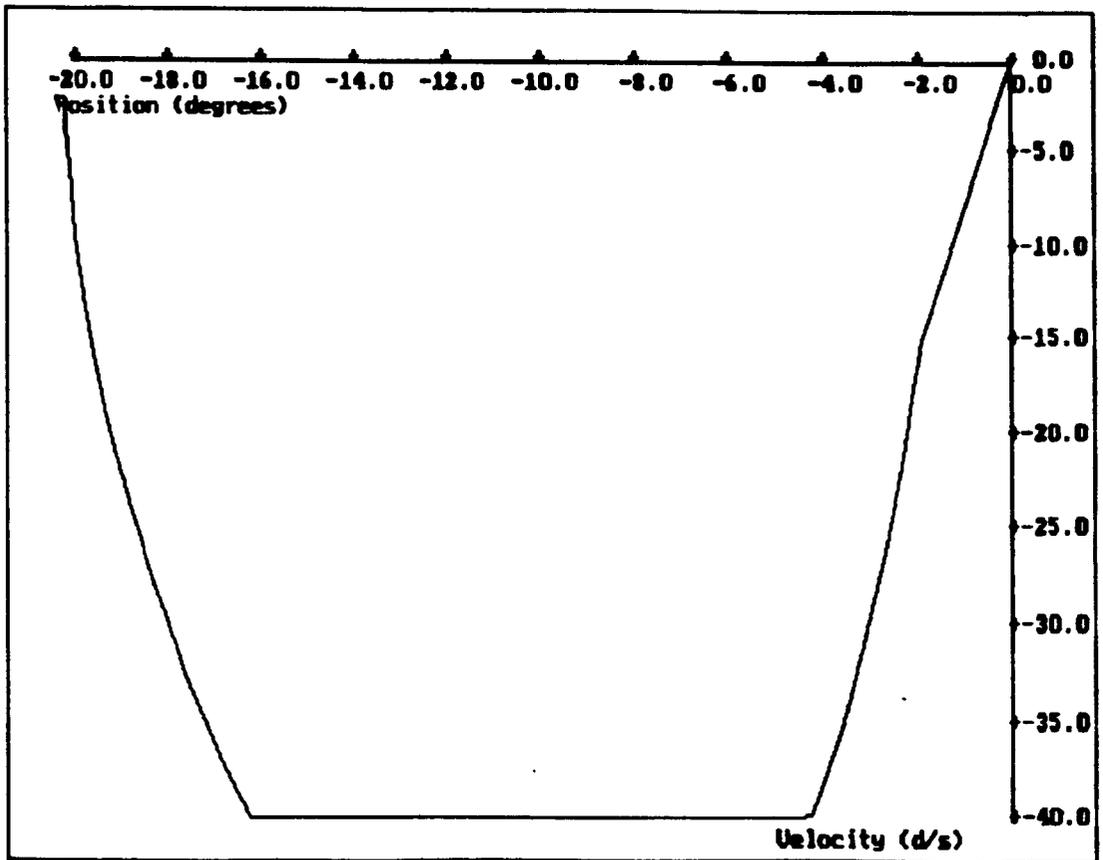
$$\alpha = 600^\circ s^{-2}$$

$$v_0 = 40^\circ s^{-1}$$

$$\beta = 400^\circ s^{-2}$$

For an initial error of 20° , this gives a switching function of the form shown in figure 6.27.

Figure 6.27 Example switching function definition



6.4.1 Simple VSC with Piecewise Linear Switching Curve

The simple VSC algorithm defined in section 6.1 was applied to the case of a piecewise linear switching function. Since the actuation signal is proportional to the error and since a large initial error is present, for much of the trajectory the system is effectively under bang-bang control. This, as expected, results in a large chatter amplitude for the majority of the transient, as shown for the elbow joint in figure 6.28.

It is worthwhile noting the effect of plant parameter changes, when using this form of control. As an example, figure 6.29 shows the velocity profile of the elbow joint with no load and with a 1kg load at the gripper. It will be seen that the behaviour is not invariant to parameter changes, as would be the case in an ideal VSC. The presence of pseudo-sliding, rather than true sliding leads to this loss of invariance.

Figure 6.28 Simple VSC with piecewise linear switching function

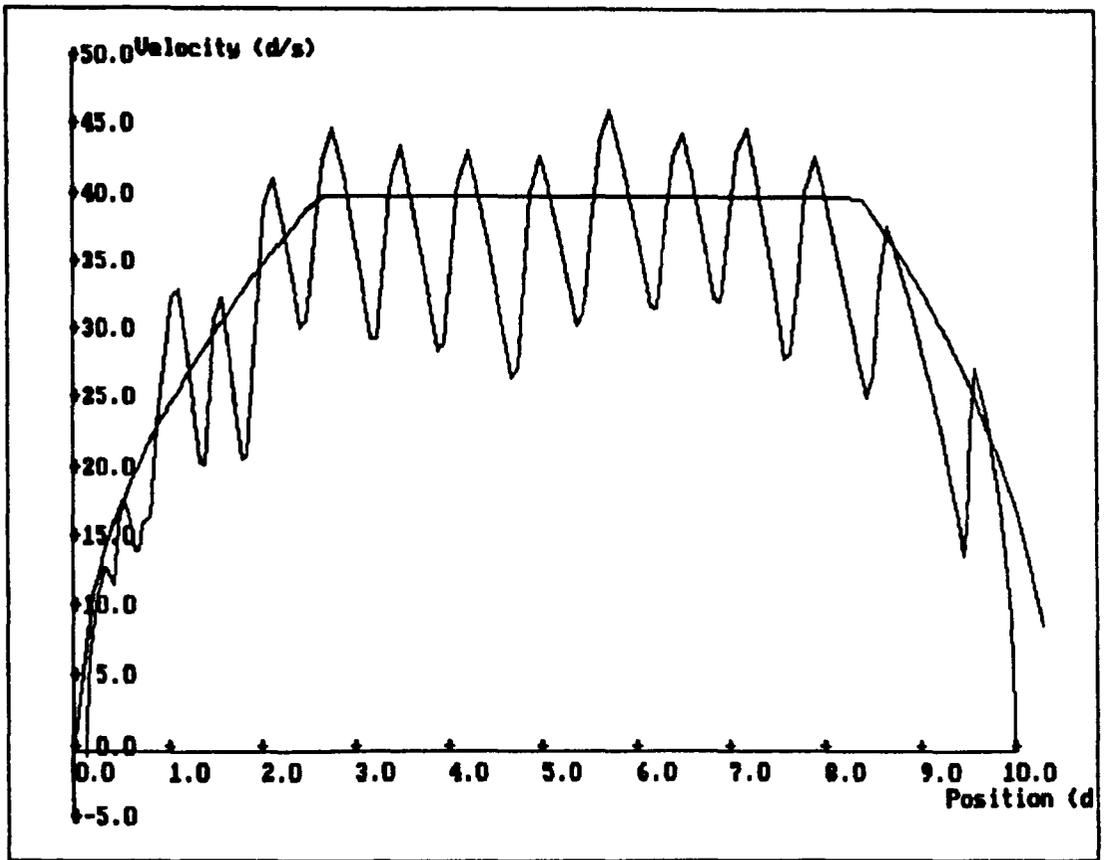
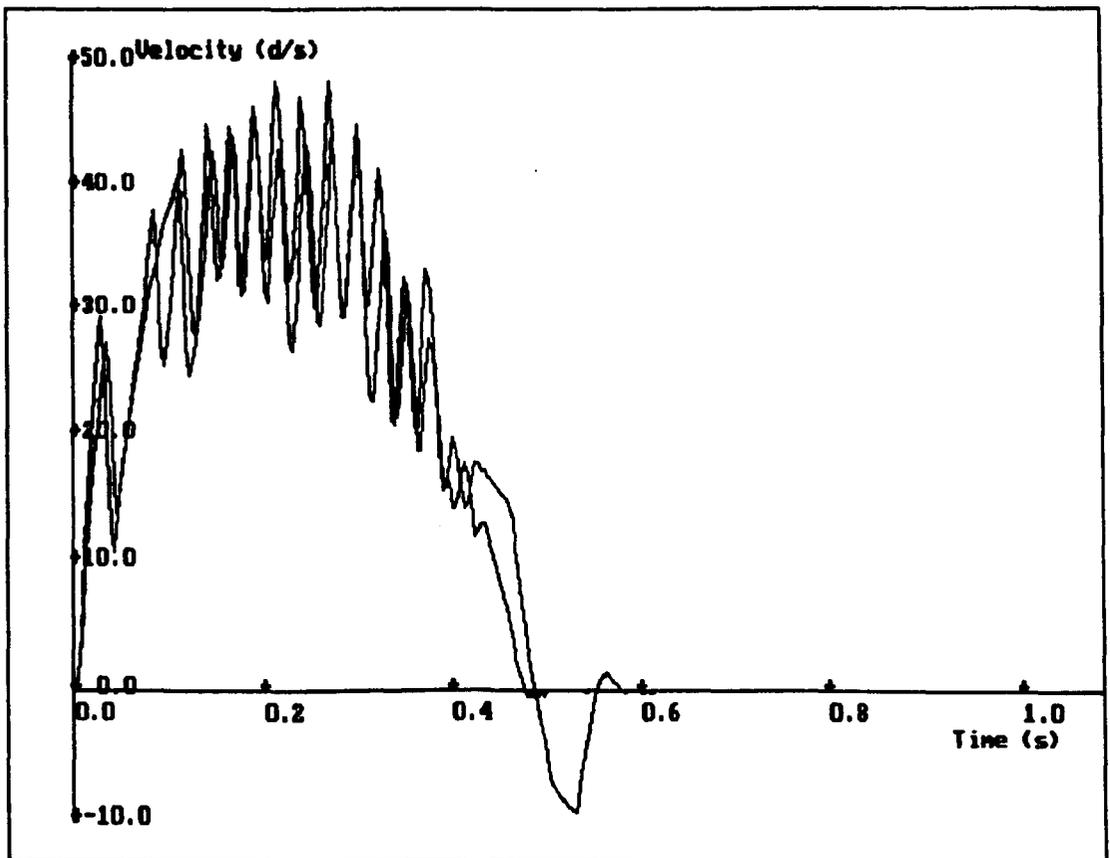


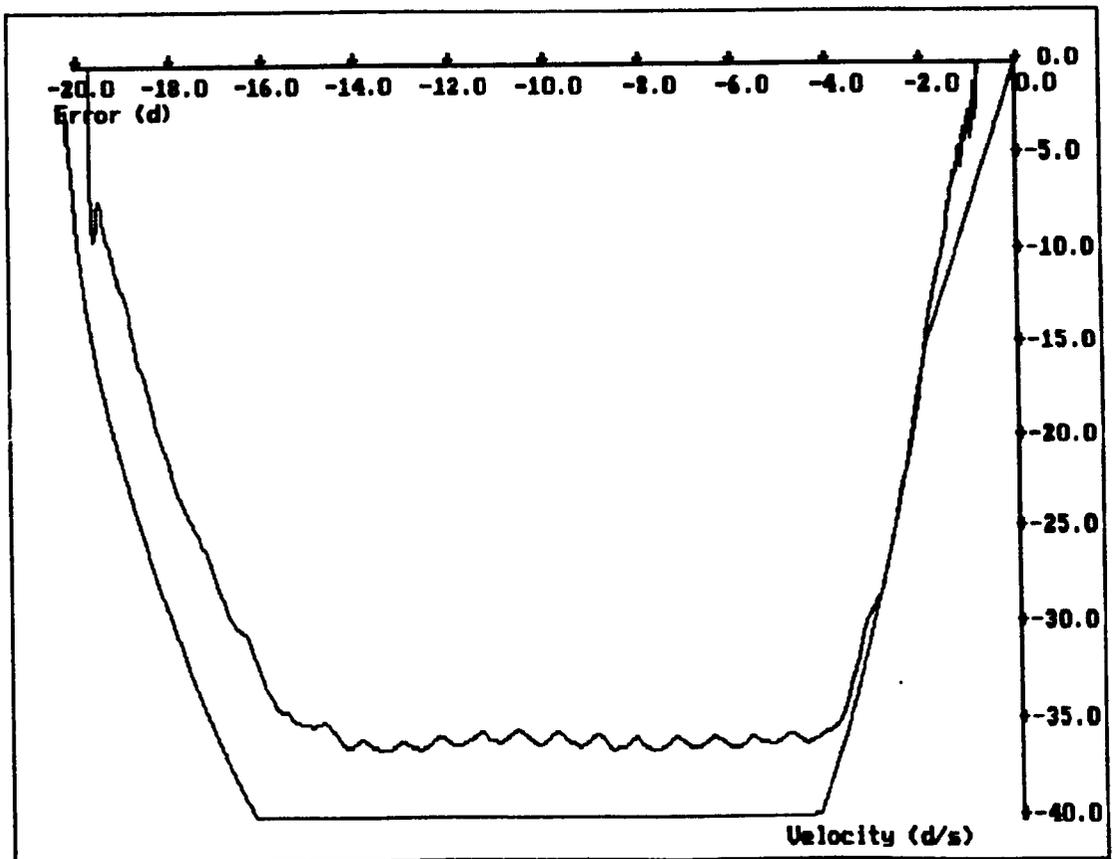
Figure 6.29 Behaviour of simple VSC with added load



6.4.2 VSC with Boundary Layer

With the addition of a boundary layer, and the modification of the algorithm to the form given in section 6.2, the actuation signal becomes proportional to the distance from the switching line, rather than to the position error, and so the problem of input saturation for large initial errors is avoided, with the effect of reducing chatter. For the tests described in this section and for the controllers developed from this, the width of the boundary layer was set to 17°s^{-1} , with a gain of 0.58V/s^{-1} . Using the same switching function definition as in section 6.4.1 results in a phase trajectory of the form shown in figure 6.30.

Figure 6.30 VSC with boundary layer and piecewise linear switching curve



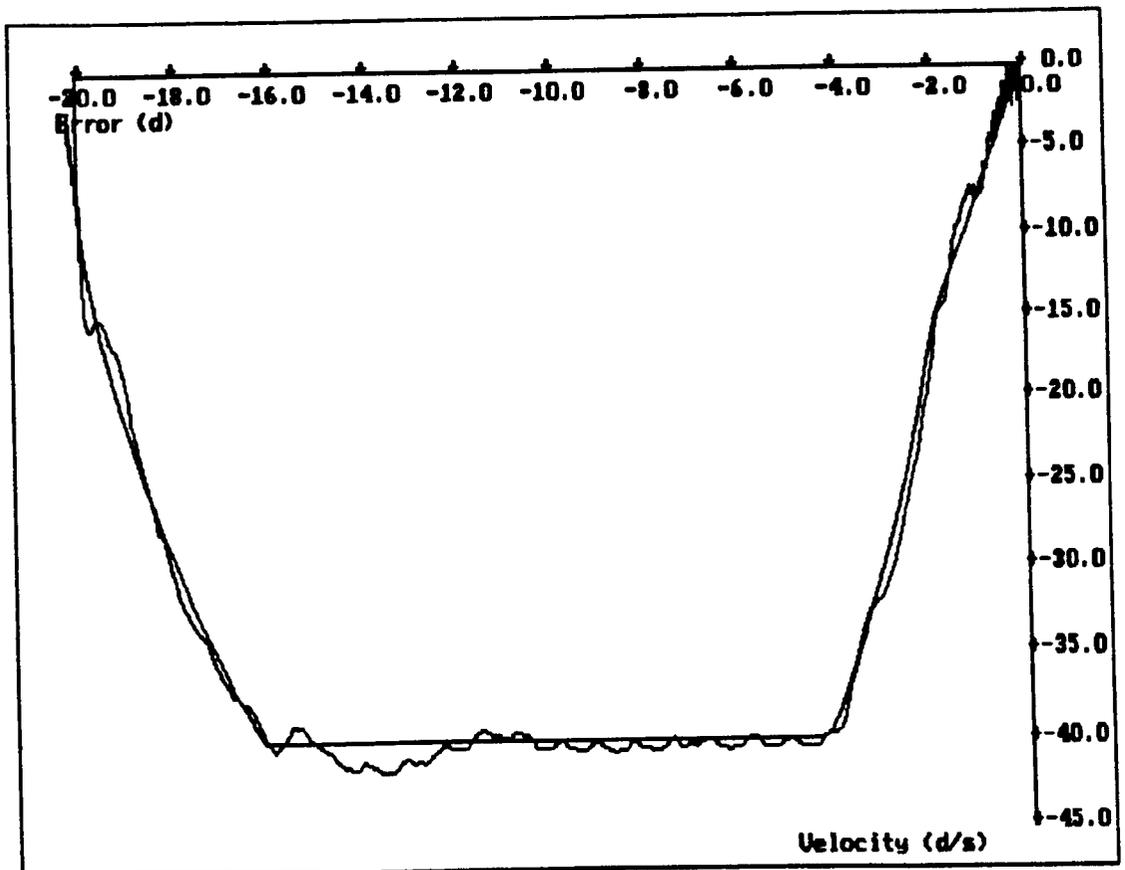
It will be seen that the trajectory follows the switching line more closely than in the case of the simple VSC, but that the velocity is always of a lower value than that required. This is due to the friction in the joint which balances

the torque generated in the motor at a given distance from the switching line. This error is unavoidable with this form of controller, although it may be reduced by an increase in gain or by the inclusion of friction compensation or integral action, as described in the following section.

6.4.3 VSC with Friction and Acceleration Compensation and Integral Action

Using the control described in section 6.3.4, with friction and acceleration compensation and integral action, combined with a piecewise linear switching function, a number of tests were performed on the elbow joint of the RTX. Adopting the same control parameters as in section 6.3.4 and the switching function definition used in section 6.4.1, the resulting phase plane trajectory is as shown in figure 6.31.

Figure 6.31 VSC with compensation and integral action



By combining all the components, it is possible to obtain good switching line following and improved robustness to parameter changes, as shown in appendix 7.1

It should be noted that the results given so far have been for controllers placed around the motor mounted position/velocity sensors and it has been assumed that the true arm position has followed the motor position sufficiently closely. Clearly this is not always the case and, if large accelerations are introduced, a significant position difference can occur. The reason for this emphasis on the control of motor position is that the accurate sensing of joint position and, more particularly, velocity cannot be achieved without the use of high cost sensors, which may not be justified in all situations. It is however worthwhile investigating the problems of controlling the joint position, rather than that of the motor. This is the subject of the following section.

6.5 Control of Joint Position

There are two major problems encountered when considering the closure of a control loop around the joint rather than the motor. The first problem is that only a relatively low position resolution is available when moderately priced sensors are considered. The resolution of 0.025° , available from the encoder mounted on the elbow joint of the RTX, is sufficient for position measurement but when this is differentiated to obtain velocity, a very poor resolution is obtained. For instance, if a 400Hz sample rate is used, then a backward difference differentiation gives a velocity resolution of $10^\circ/\text{s}$, approximately 7% of the maximum velocity. Although this may be reduced slightly by the use of more complex filtering techniques, a significant improvement cannot be obtained without the introduction of a large phase lag.

The second problem encountered is that caused by the flexibility in the joint transmission system. The torque applied to the arm may be considered as being dependant on the difference between motor and joint position, rather

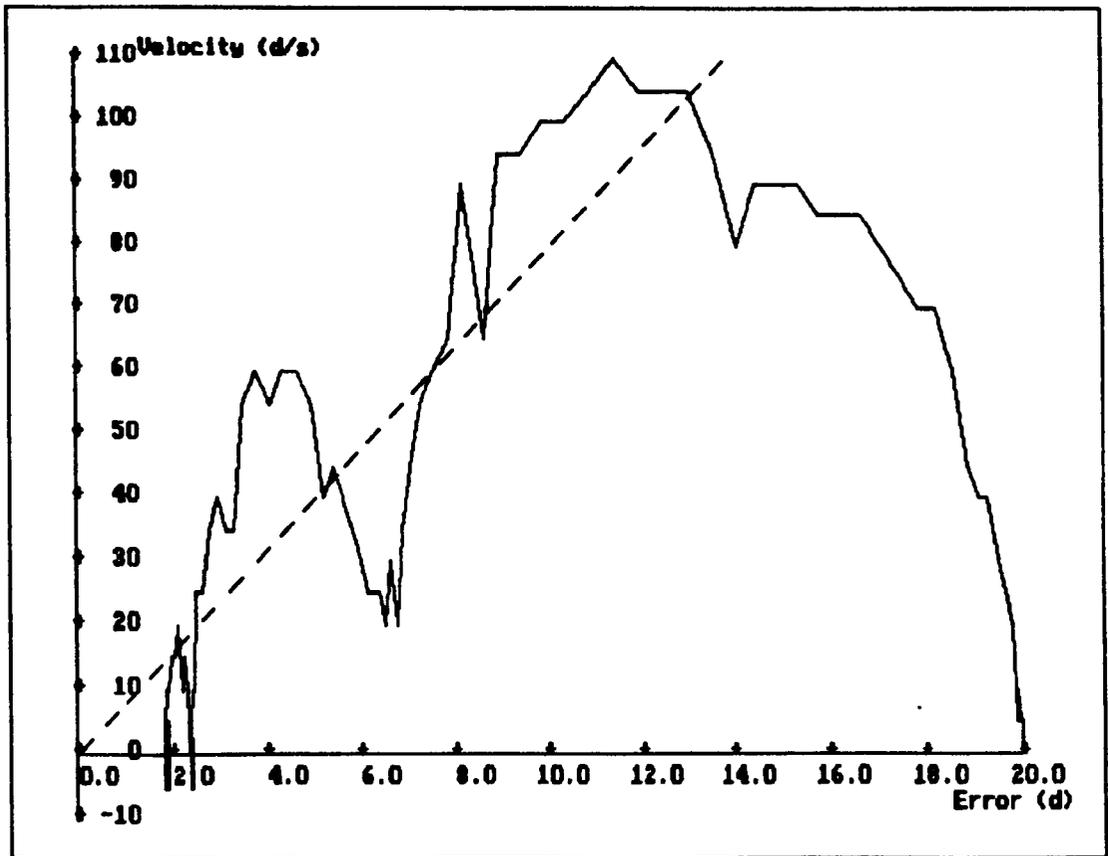
than on the torque generated in the motor. This effect, as noted by Wallenberg [Wallenberg,88], often leads to limit cycle behaviour.

Notwithstanding the preceding problems, a number of forms of variable structure control have been applied to the RTX, using joint position and velocity rather than those of the motor. Fortunately there is sufficient damping in the transmission system and in the joint itself to allow some form of control to be obtained, although it will be seen that the performance is, in some respects, inferior to that obtained when using a control loop around the motor. These controllers will be described in the following sections.

6.5.1 Simple VSC Around Joint Position

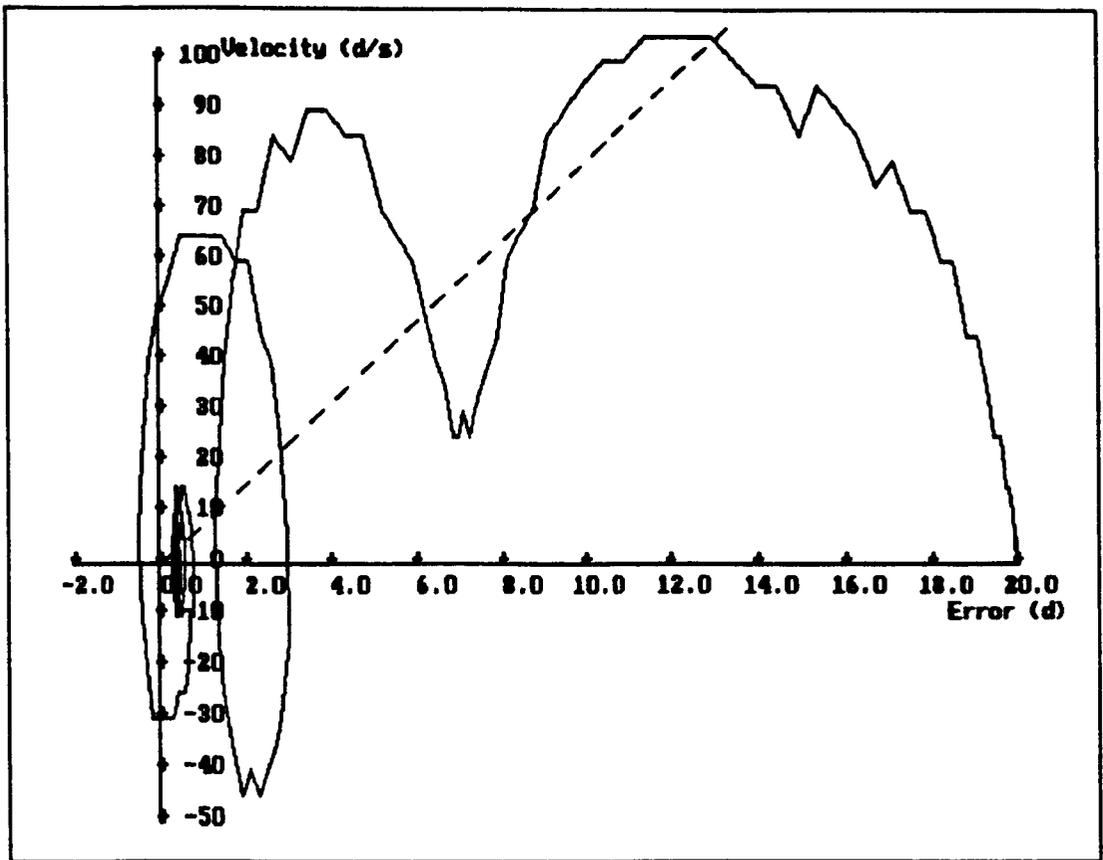
A simple VSC may be applied to the arm, with position and velocity derived from the joint mounted sensor, in the same manner as described in section 6.1.3, with suitable scaling factors to allow precise setting of control parameters, without the possibility of overflow. A typical response to a 20° step change in demand input for a system with control parameters $c = 8s^{-1}$, $\psi = 0.6V/^\circ$ and a sample rate of 200Hz is shown in figure 6.32. It will be seen that pseudo-sliding behaviour does occur, but with a significantly higher degree of chatter than was the case with feedback from the motor mounted encoder. The other problems described in section 6.1.3 also remain here; there is a steady state error of 1.8° and the control activity is large.

Figure 6.32 Simple VSC around joint sensor



The problem found when applying any form of discontinuous control to a system with significant drive flexibility is that the flexing of the drive allows a position difference to be established between the motor and the joint; the size of this difference being dependant on the amplitude of the discontinuity, a large discontinuity leading to a large position difference, and it is this position difference which leads to the degradation of performance observed here. As the gain of a VSC is increased, so the amplitude of the discontinuity increases for a given error leading to further reduction in performance. This may be seen in figure 6.33 in which a gain of $4.0V/^\circ$ was used. The large discontinuity in control has led to oscillatory behaviour.

Figure 6.33 High gain VSC leading to oscillatory behaviour

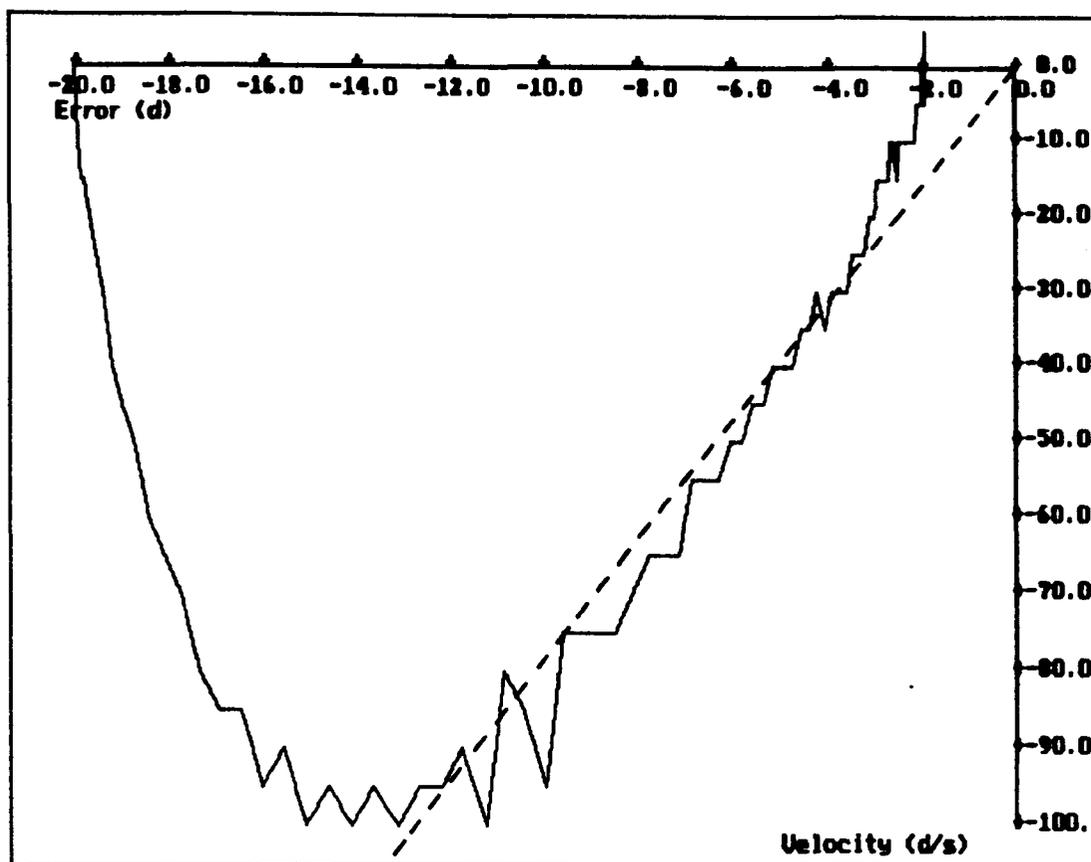


The problems induced by the use of discontinuous control may be alleviated to some extent by the use of smoothing techniques as described in section 6.2.

6.5.2 VSC with Boundary Layer Applied to Joint Position

Adding a boundary layer to the switching line in the manner described in section 6.2 results in a significant improvement in performance, as illustrated in figure 6.34, which shows the result for the elbow joint with control parameters $c = 8\text{s}^{-1}$, $k_b = 67\text{s}^{-1}$ and $k = 0.15\text{V/s}^{-1}$ and a sampling rate of 200Hz.

Figure 6.34 VSC with boundary layer

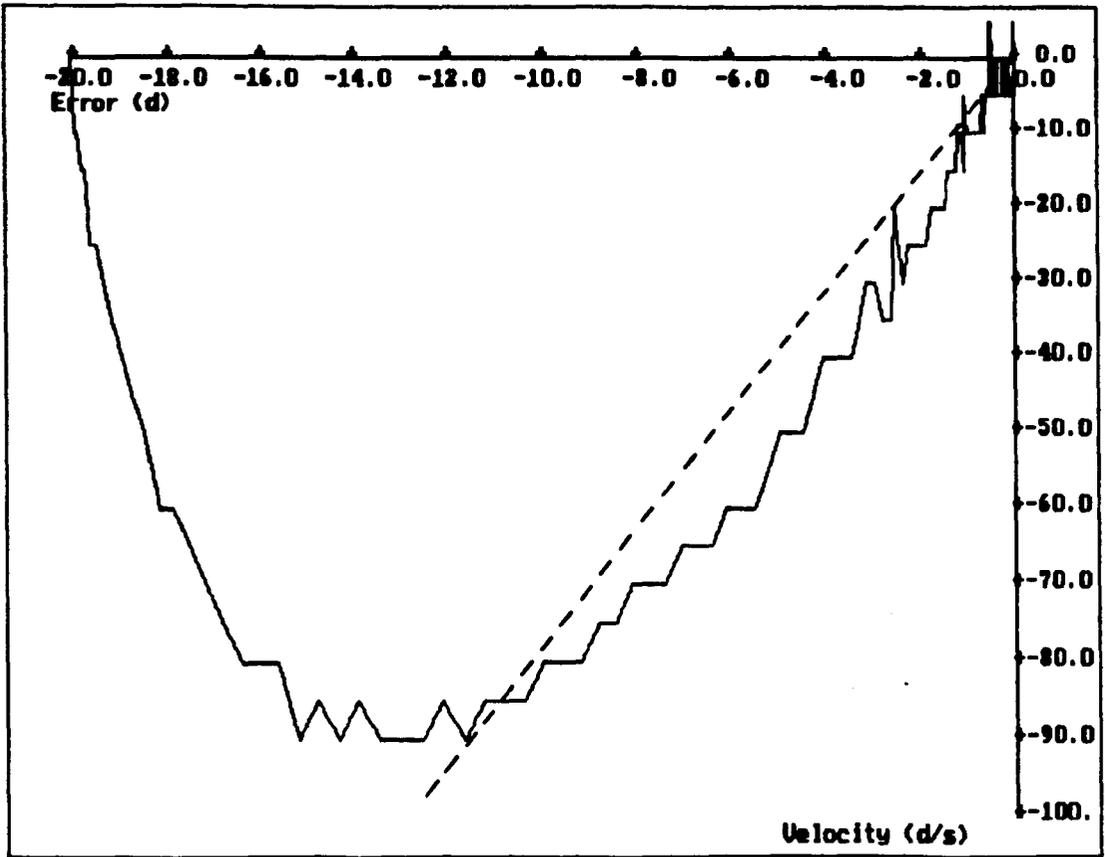


The deviation of the trajectory from the switching line has been significantly reduced by the removal of the discontinuous control signal, without an increase in the steady state error. Once again, the use of a large controller gain results in undesirable behaviour, as seen in appendix 7.2.

6.5.3 VSC with Compensation and Integral Action

In much the same manner as described in sections 6.3 and 6.4, friction and acceleration compensation and integral action may be used to enhance the performance of the VSC applied to the control of joint position. It is found that similar performance improvements are achieved, as illustrated in figure 6.35, in which the same compensation parameters as found suitable for motor position control were used, along with a gain of $0.15V/\dot{s}^{-1}$ and an integral constant of $1V/\dot{\cdot}$.

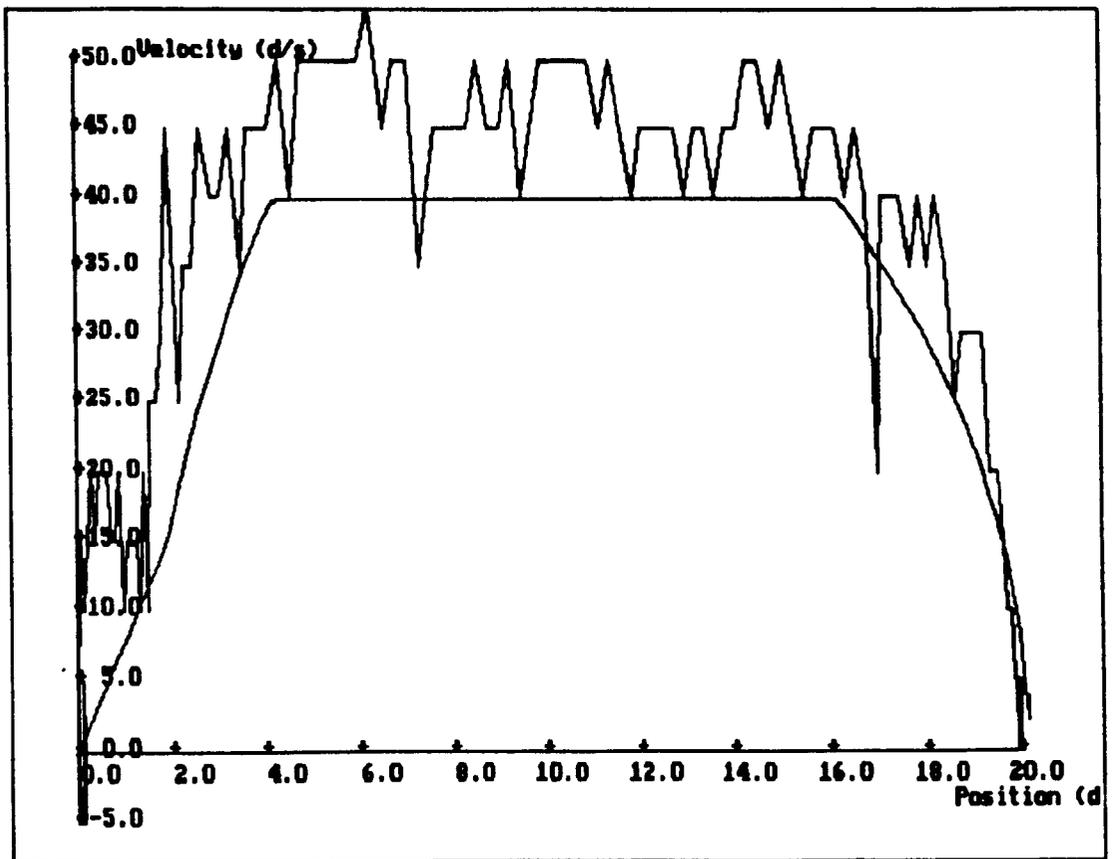
Figure 6.35 VSC with friction and acceleration compensation and integral action



The final form of the controller, represented by the result shown in figure 6.35 gives close switching line following and a steady state error of 0.05° , corresponding to two encoder counts.

As a means of improving the robustness of the complete movement, a piecewise linear switching function of the type described in section 6.4 may be adopted. When combined with the enhanced VSC algorithm, this produces results of the form shown in figure 6.36 from which it is seen that switching line tracking is within 10°s^{-1} for the majority of the trajectory; a figure which corresponds to two times the resolution of the measurement. Clearly this performance is far from ideal but could not be significantly bettered without improved velocity sensing.

Figure 6.36 Joint control using VSC with piecewise linear switching curve



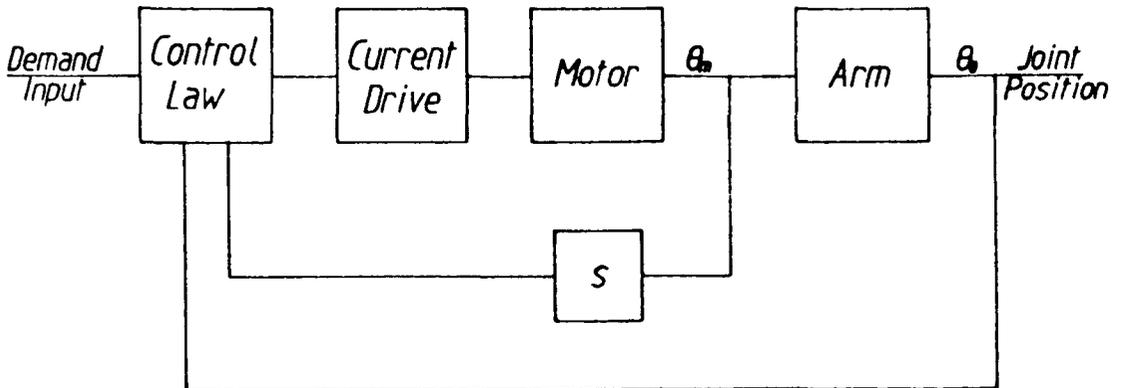
6.6 Control Using Joint and Motor Mounted Sensors

From the preceding results, it will be clear that control using the motor mounted position sensor gives good velocity control but poor end point position accuracy, while using the joint mounted sensor allows good position accuracy but does not allow precise velocity control, due to the limited sensor resolution. It is therefore desirable that the advantageous properties of each control scheme be combined in a single control system.

For reasons explained previously, when a steady velocity or acceleration is induced in the motor, a position difference between the motor and the joint occurs, suggesting that a control system taking data from both sensors would not be effective, if the transmission flexibility were not explicitly considered. It should be noted however that, given constant velocity or acceleration, the steady state position difference is fixed, provided the link dynamics remain unchanged. Under these conditions, the joint velocity will be approximately

equal to the motor velocity. Taking account of this, it was considered worthwhile to investigate the behaviour of a control system using the joint position and the motor velocity. A schematic diagram of the variable structure controller adopted is shown in figure 6.37.

Figure 6.37 Variable structure control system using motor and joint mounted sensors



This form of control may be viewed in two perspectives, firstly it may be considered as a straightforward VSC with a boundary layer, in which any discrepancy between the motor and joint velocities is neglected, with the inherent robustness of the VSC algorithm minimising any detrimental effects caused by the disparity. Alternatively, it can be considered that the position error as measured at the joint is used to generate a velocity demand which the controller forces the motor to match. In the later case, since the velocity profile generated from the position error is a smooth function, the velocity difference between motor and joint does not undergo any rapid changes, justifying the assumption the the two velocities are approximately equal.

It is a simple matter to convert the control software already written to operate in this manner, only requiring changes to the software controlling the reading in of position and the calculation of velocity, with minor modifications to the scaling constants used. It was found that the control parameters developed using the control system operating around the motor mounted sensors

continued to give good performance when adapted to this configuration, although limited tuning was carried out to give further improvements.

The performance of the elbow joint of the RTX with this sensor arrangement was investigated for several forms of variable structure control algorithm with boundary layer described in previous sections and the resulting performance will now be summarised. In assessing the performance of the various forms of control investigated, a problem is encountered in determining the exact joint behaviour, owing to the poor velocity measurement resolution. Thus the proximity of the trajectory to the switching line cannot be found precisely, although a reasonable idea of the overall performance can be obtained.

The initial investigation of this controller involved the use of a straight switching line with boundary layer and a control signal proportional to the deviation of the trajectory from this line, and proceeded to consider the inclusion of friction and acceleration compensation and integral action.

6.6.1 VSC with Boundary Layer using Motor and Joint Mounted Sensors

The simplest form of control without compensation or integral action, defined by:

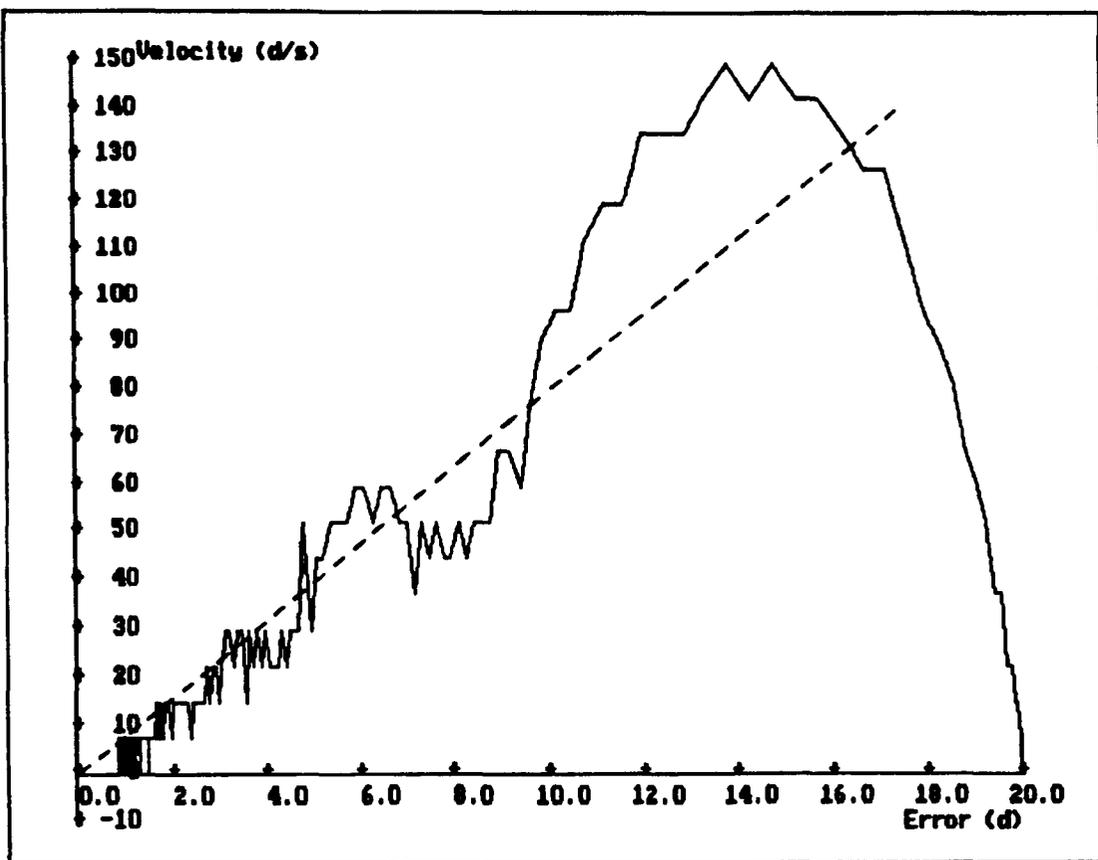
$$u = \begin{cases} -1 \operatorname{sgn}(s) & \text{if } |s| > k_b \\ k_s s & \text{if } |s| \leq k_b \end{cases}$$

$$s = c(e_j)e_j + \dot{e}_m$$

where e_j is the joint position error, defined by $e_j = x_d - x_j$, where x_j is the joint position; and \dot{e}_m is the motor velocity error, defined by $\dot{e}_m = \dot{x}_d - \dot{x}_m$, where \dot{x}_m is the motor velocity.

Using this form of control, applied to the elbow joint, results in a response of the form shown in figure 6.38.

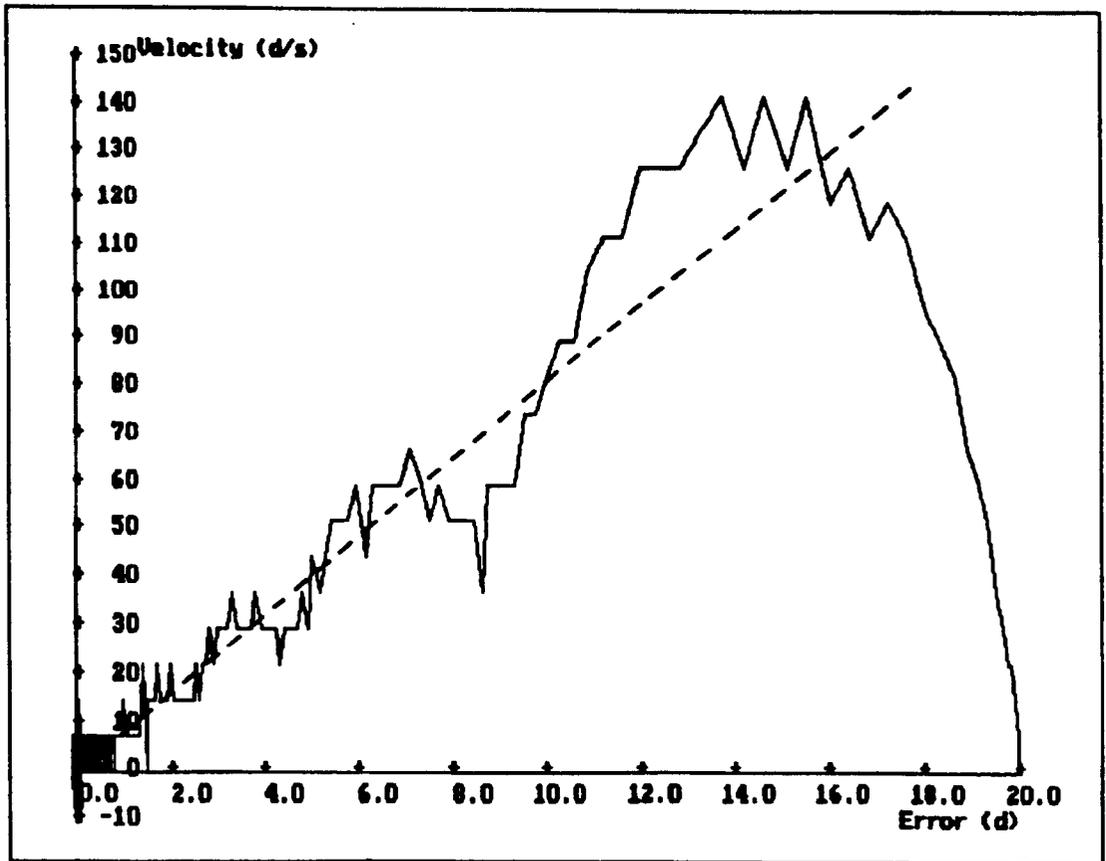
Figure 6.38 VSC with boundary layer using joint and motor mounted sensors



It will be noted that the trajectory remains in the vicinity of the switching line but has a far greater deviation from it than is the case when using either the motor or joint mounted sensor alone. This relatively large deviation might be expected, given that the control system undergoes an abrupt change of structure from bang-bang to a smooth control action when the trajectory first reaches the boundary layer and that this change leads to a similarly rapid variation in the relative positions of the motor and joint. Following this large initial deviation, the trajectory follows the switching line more closely until, as the origin is approached, Coulomb friction leads to the position shortfall experienced previously with this form of control algorithm.

Figure 6.39 shows the effect of adding friction and acceleration compensation and integral action to the basic algorithm. The improvements in performance found are similar to those obtained when these control algorithms were used with the motor mounted sensor.

Figure 6.39 VSC with compensation and integral action



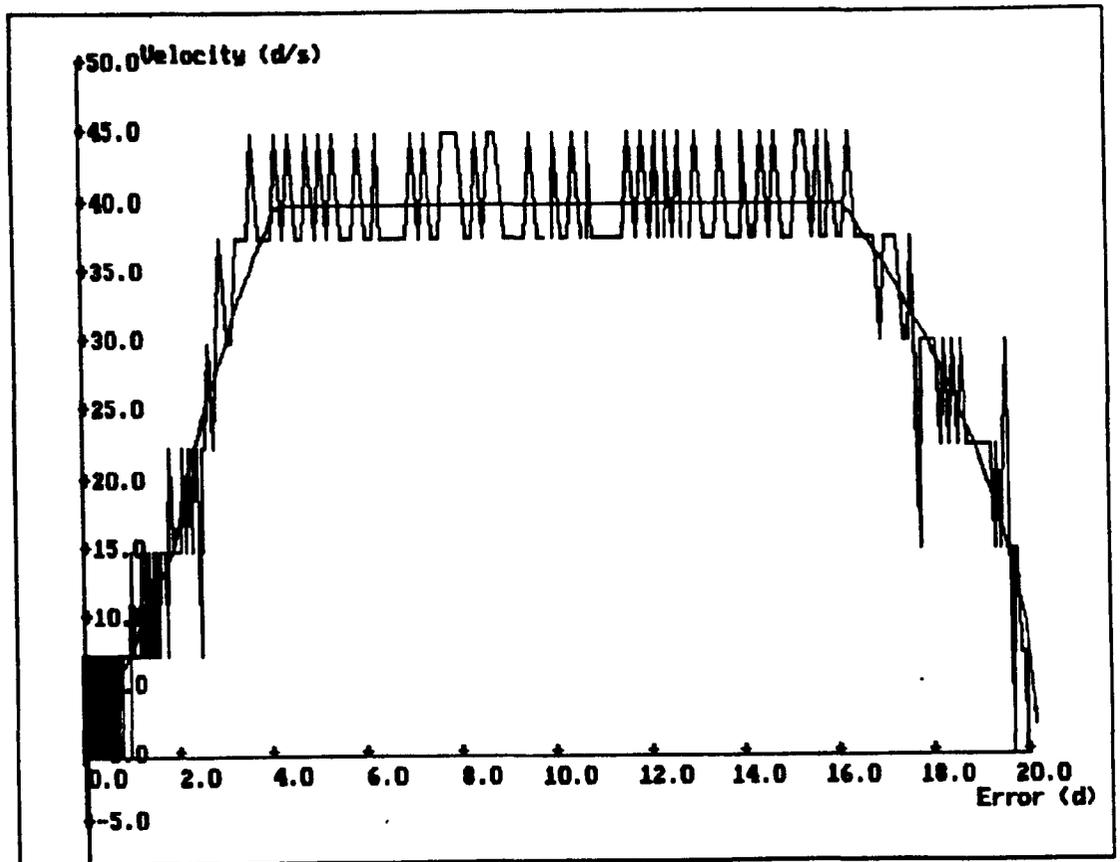
6.6.2 VSC with a Piecewise Linear Switching Function

It was noted in the previous section that a relatively large deviation of the trajectory from the switching line occurs following the transition between the two control structures. This is caused by the rapid change in the required acceleration which occurs at this point, leading to a rapid change in position difference. One approach which may be adopted to overcome this problem is that of using a piecewise linear switching function of the form described in section 6.4. In this regime, the changes in acceleration are relatively small, leading to reduced position disparity and improved switching line tracking. A number of experiments were carried out using this type of switching function with each of the variations to the basic control signal definition described in the previous section and in section 6.4.3. It was found that each development of the algorithm gave similar performance benefits to those described in section 6.4.3, with the response of the elbow joint for a controller with a friction and

acceleration compensation and integral action being of the form shown in figure 6.40.

It will be seen that the trajectory follows the switching function closely, given the poor velocity measurement resolution, and gives less deviation from the switching function than was the case when a straight switching line was used, owing to the relatively small changes in acceleration required in this case.

Figure 6.40 VSC with a piecewise linear switching function



6.7 Comparison of Control Algorithms for the RTX

In the preceding sections, a number of control algorithms have been applied to the control of the elbow joint of the RTX and the resulting behaviour investigated. The performance of each type of controller has not however been compared with each other in quantitative terms. It is the intention in this section to present such a comparison between these controllers

and a PID controller for a variety of movements. The comparison will be divided into four sections, considering the computational effort required; the speed of response for the movement of a single joint; the accuracy and repeatability of point-to-point movements involving the elbow and shoulder joints and the tracking accuracy for continuous path movements. The performance in each of these respects will be considered in turn in the subsequent sections and this will be followed by an overall summary and recommendations on selection.

It should be noted that in preceding sections, the control parameters were not adjusted to give the best possible performance in all cases, since this would not allow an evaluation of the effect of individual terms in a control algorithm. It is therefore necessary to re-tune each controller to give the best performance before carrying out the tests reported in this section.

6.7.1 Computational Cost of Control Algorithms

As indicated in the introduction to this chapter, the computational cost of the control algorithms considered here may be evaluated by determining the amount of slack time between interrupts. This method of evaluation gives a useful indication of the computation time required for a given coding of the algorithm and has frequently been used in the development of algorithms. It does however give an indication of the efficiency of the algorithm coding, rather than of the algorithm itself, and so a modification to the software (replacing certain operations with machine code routines for instance) may give an increased amount of slack time, without any change to the algorithm. In addition, this method does not take account of the relative speed of the hardware on which the program is run. For these reasons it was decided that a more useful indication of an algorithm's efficiency would be obtained by a comparison of the number of times each form of arithmetic operation is required. The figures for each of the components in the algorithms considered

in the preceding sections are collated in table 6.2. The number of operations for a given control scheme may be found by adding the components for each element used. Thus, for a VSC with boundary layer and acceleration compensation requires a total of 2 + 1 add/subtracts, 2 + 2 multiply/divides and one conditional statement.

Table 6.2 Number of operations required in control algorithms

Control Algorithm Element	Operation Type		
	Add/Subtract	Multiply/Divide	Others
PID	3	3	-
Simple VSC	2	2	1 sign
VSC with Boundary Layer	2	2	1 Conditional
Friction Comp.	1	1	1 sign
Acceleration Compensation	1	2	-
Integral Action	1	1	-
Piecewise Linear Switching Curve	1	1	2 Table Lookups

These figures represent the computation involved in the control of a single axis, using position information derived from a single encoder. In the case where multiple joints are considered, the increase in computation is a multiple of that for a single joint, with only a single element for the interrupt handling overhead, which is only required once for each interrupt, independent of the number of joints. This overhead time is 0.18ms on an 8086 based computer running at 4.77MHz. When the control algorithm requires both joint and motor position or velocity information, the input routine must be repeated twice. Each of these position readings takes 0.46ms on the same computer.

6.7.2 Speed of Response with Variable Structure Controllers

An important consideration in the evaluation of a control algorithm for a robot manipulator is the speed with which the controller drives the arm to the required position, generally with the proviso that any overshoot of the required position be extremely small. When developing a control system with this in mind, it is vital that the nonlinear nature of the robot, and in particular the limited forces/torques available, be considered since, if these factors are neglected and a linear system model is adopted, it is possible to design a control system which will theoretically provide any speed of response required. Clearly this is not in practice possible and the behaviour of such a system will in practice be far from ideal, often with overshoot and possibly instability.

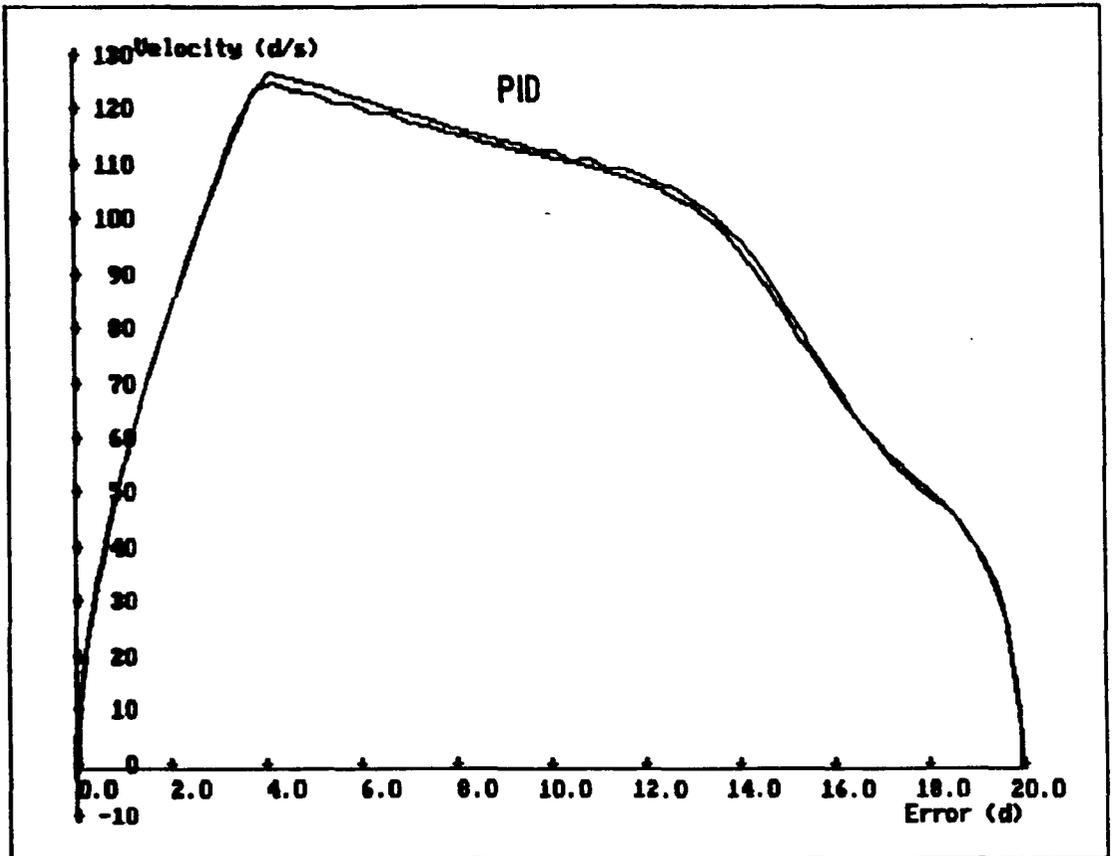
It has been shown [Ryan,81] that for a single input single output, time invariant system with bounded inputs, the fastest speed of response is obtained when the control signals are extremal. In addition, for a second order system, the optimal motion is obtained using a single switching of the control signal from one limit to the other during the transient.

Although an optimal control system could be determined, following the procedure described by Geering [Geering,86], and augmented to overcome the effects of nonlinear friction, such a system would be highly sensitive to parameter and load changes. It is clear from this however that, in order to attain the highest speed of response possible, the nonlinear nature of the RTX must be considered and that a near optimal response will be one which reaches the desired point using a control signal which is saturated for the majority of the transient and has a small number of switches. These facts were used in tuning the control systems described previously so that, by looking at the control signal used, the optimality could be estimated.

The movement for which the speed of response was compared involved the elbow joint in a rotation of 20° , with the yaw axis at an angle of zero degrees to the radial line through the shoulder axis and with no gripper or other

additional load. Using this example movement, the PID controller and the VSC with a straight switching line and boundary layer were tuned to give the fastest response without overshoot. The resulting phase trajectories are shown in figure 6.41 while the control signals are shown in appendix 7.3.

Figure 6.41 Response of PID and VSC with boundary layer tuned for maximum speed



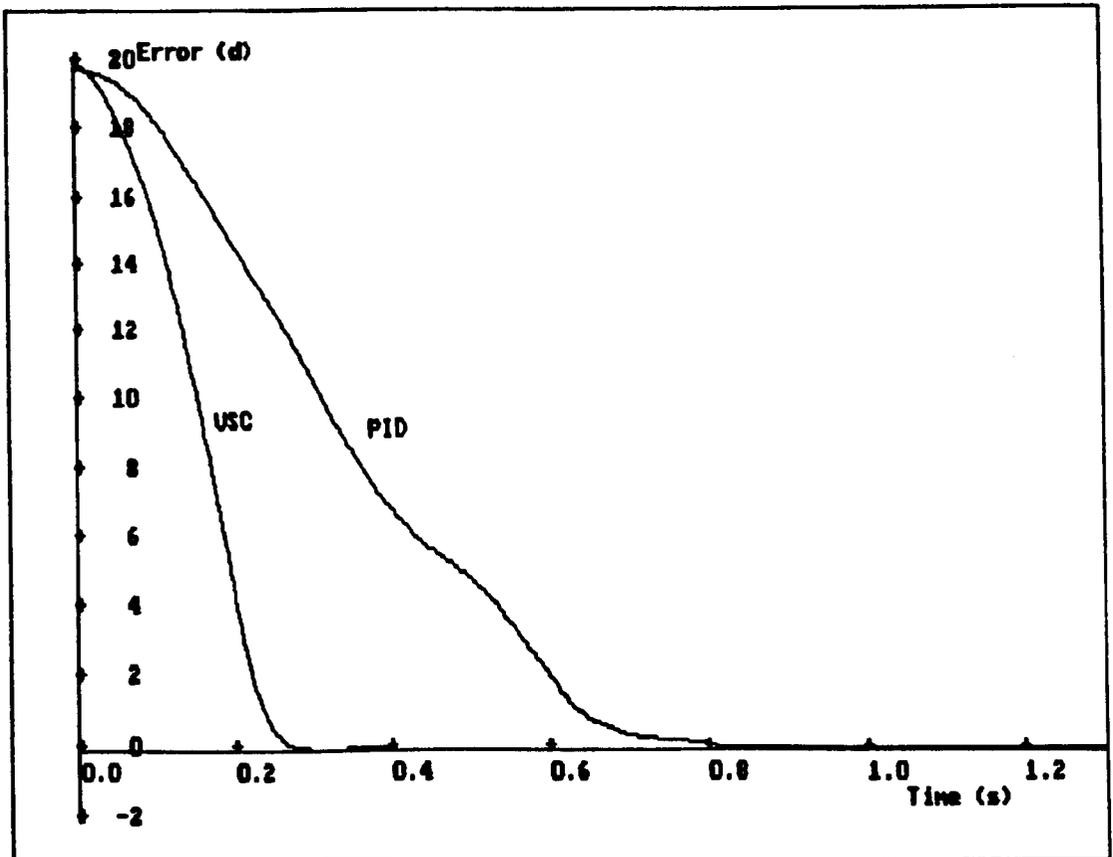
It is clear that the two trajectories are very similar, with the VSC being only marginally faster in this case, although in other tests this was not so, the PID control being faster by a similarly small margin. The variations in speed are due to changes in the initial joint angle, relative to the motor angle, dependant on the previous movement carried out. It should be noted in the case of the VSC that the trajectory deviates from the switching line by a large amount and, in fact leaves the boundary layer.

From a consideration of optimal control theory it is clear that if a simplified, double integrator model is used to represent the plant dynamics, the

time optimal controller will involve a curved switching line. With this in mind, a VSC with boundary layer using a piecewise linear switching function was investigated. It was found, as seen in appendix 7.3, that the phase trajectory was virtually the same and the time response no faster than when a straight switching line was adopted.

Although it has been shown that the difference in performance between PID and VSC algorithms is minimal when tuned for a particular movement, it is worthwhile comparing the speed of response of the elbow joint using the VSC algorithm and the modified motor drive and sensor hardware with that attained by the manufacturers using the original hardware system. A comparison is shown in figure 6.42 from which it is clearly seen that the control system developed has resulted in a speed improvement by a factor of three for a typical movement.

Figure 6.42 Comparison of speed using VSC and manufacturer's PID control



6.7.3 End Point Position Accuracy

Throughout the work described so far, the accuracy of movement has been judged from the difference between the position as sensed at either the motor or the joint and that required; and it has been shown that this error can be reduced to a very small amount, typically of the order of 0.005° , by the use of appropriate control strategies. Although this is a necessary activity and, when the joint mounted encoders are used, provides a reasonably accurate measurement of end point position accuracy, there are a number of effects which can lead to end point errors which cannot be measured in this way.

The main sources of end point position errors, when the motor mounted sensors are used, are flexibility in the timing belt transmission system and backlash between these belts and the pulleys. These effects can result in a significant difference between the true joint angle and the angle measured at the motor (up to 2.0° due to flexing and 2.5° due to backlash in the elbow joint) and although the former problem mainly occurs during the transient, when the force transmitted by the belt is greatest, owing to the static friction in the joint, a steady state error can also be introduced from this source. Errors may also be introduced when the robot is initialised, since each joint is driven to an end stop, introducing a significant degree of flexing in the transmission, before the motor position is set. Due to variability in the backlash and the static friction characteristics, a large variation in the initialisation position can occur (up to 3mm).

Each of these problems are alleviated when the position sensors are mounted on the joints themselves but there are still further problems which cause inaccuracy in this situation. These problems are caused by flexing of the Z-column and the carriage on which the arm is mounted. Although mainly causing errors during the transient stage, some inaccuracy can remain in the steady state position.

Due to the inaccuracy of the initialisation procedure, a test of the absolute

accuracy of the robot would give more information about the accuracy of the initialisation than that of the control algorithm itself and so it was decided that a measure of the repeatability would be more informative in this instance. In order to measure the repeatability, a number of techniques were considered as described in the following section.

6.7.3.1 Cartesian Position Measurement Techniques

A number of schemes to allow accurate measurement of the cartesian position of the robot end point have been investigated and a suitable technique established. A number of the possible solutions to this problem will be considered here. Ideally the measurement system should be non-contact so as to avoid any effect on the robot dynamics and should give a high measurement speed to allow evaluation of the dynamic behaviour.

A number of optical techniques have been considered and a system using a videcon camera connected to a MicroVAX with a frame store developed. The MicroVAX was used to determine the centre of area of a light source mounted on the end effector and, when running real time allowed a sample rate of 7Hz to be attained or, by recording the video signal to tape before processing, this could be increased to 25Hz. Owing to the nonlinear nature of the camera and lens system, the resulting measurements were not sufficiently accurate and although they could be linearised, this would require precise setting up and would increase the computation time significantly.

A system which would provide a higher speed of sampling was investigated, based on an X-Y optical position sensor which could be made to provide voltages proportional to the X and Y co-ordinates of a light spot falling on it's surface. The sampling rate using this device is only limited by the signal to noise ratio of the sensor and the subsequent electronics. Once again, the optics and sensor used were nonlinear and so the system was not fully developed. A number of optical tracking systems are commercially

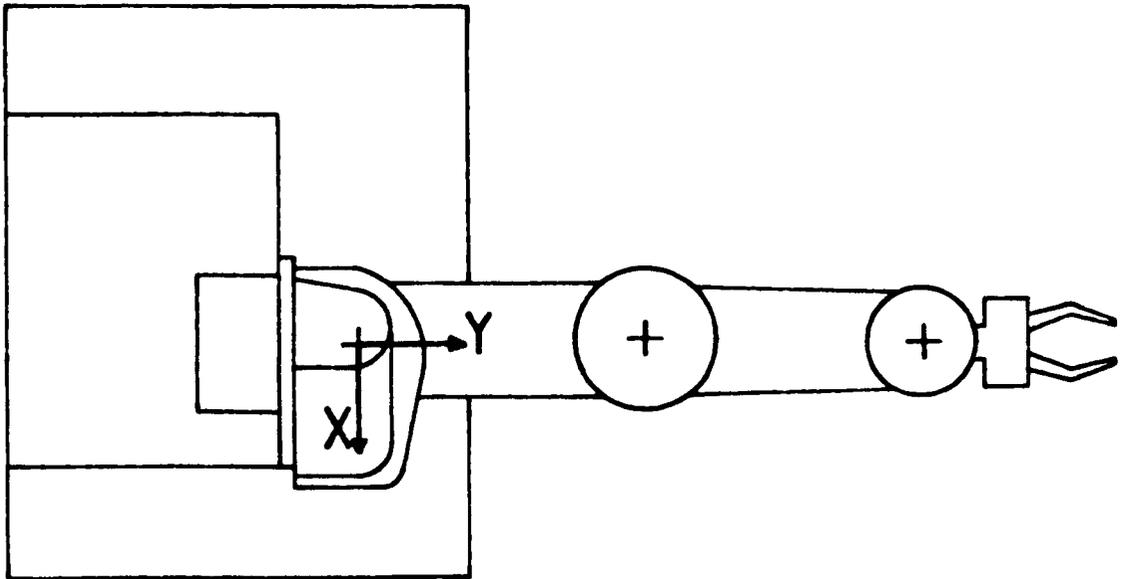
available, including the Watsmart system supplied by Northern Digital Inc. and Robotest available from Polytec, but were not considered cost effective in this instance.

Two measurement schemes were eventually used, both dependant on the planar nature of the movements considered. The first technique involved a solenoid, mounted on the end effector with the plunger caused to vibrate with a vertical movement of approximately 4mm, at a known frequency. By attaching the solenoid to a pen, or by placing a sheet of carbon paper below it, a trace of the end effector path could be obtained. Photographically enlarging the resulting plot allowed the positions to be accurately measured in both X and Y co-ordinates. The sample rate attainable using this method was limited to 20Hz by the solenoid dynamics; also, the evaluation of the trajectory was time consuming, particularly when a high dot rate was used.

A preferable method and the one used to derive the results presented here, consisted of a digitizing tablet placed below the robot with the cursor mounted on the end robot effector. By simulating the pressing of a button on the cursor by means of a by-pass circuit, the cartesian position of the end effector could be obtained and transmitted via a serial line to a logging computer. Driving the cursor button by-pass from a signal generator enabled by the control computer for the duration of the transient, a record of the transient behaviour could be obtained with an accuracy of 0.1mm at a sample rate of up to 150Hz. The static position accuracy could be assessed in the same manner.

In order to investigate the cartesian position accuracy, a suitable co-ordinate frame must be established. The choice of co-ordinates is somewhat arbitrary; that shown in figure 6.43 is used throughout the subsequent description.

Figure 6.43 Co-ordinate frame used in cartesian accuracy measurements



Both point-to-point and continuous path accuracy measurements were performed with the arm at a distance of 327mm from the highest point of its travel.

6.7.3.2 Repeatability of Point-to-point Movements

The repeatability of the RTX was assessed using a number of control algorithms and sensor configurations at several points in the working envelope. For each test, the required position was approached from a point 50 mm or 100mm distant in the positive and negative X and Y directions, thus giving four separate movements. The demand positions were (0,200), (0,400), (300,300) and (-300,300), thus giving an indication of the effect of position on the repeatability. The resulting measurements are presented in table 6.3 as the maximum and mean absolute deviation from the mean position in X and Y directions and radially.

Table 6.3 Point-to-point repeatability for various controllers

Control Algorithm / Position Sensor Placement	See Part	Deviation from Mean (mm)					
		X		Y		Radial	
		Max.	Mean	Max.	Mean	Max.	Mean
Original PID	-	0.38	0.14	1.00	0.43	1.03	0.47
PID	-	2.30	0.84	2.35	1.09	2.58	1.48
VSC / Motor	6.3.4	1.80	0.66	1.57	0.77	2.31	1.10
VSC / Joint+Motor	6.6.1	0.27	0.08	0.47	0.22	0.48	0.25

From the figures presented in table 6.3, it is clear that although changes to the control algorithm can give some improvement in repeatability, in order to achieve high accuracy, the position sensors must be mounted at the joints rather than the motor. This hardware modification leads to a typical improvement in end point position repeatability of a factor of four, compared with that achieved using the same algorithm but with the motor mounted sensors.

The figures for the manufacturer's control system appear good compared with those for the control systems using the motor mounted sensors. The reason for this difference is that, owing to the lower acceleration attained using the manufacturer's motor drive system, the stiction between the belt and pulley is not broken, and the effect of backlash is not encountered. If the arm were required to manipulate a heavy load or apply force to an object as part of an assembly task, a lower repeatability could be expected, since this stiction would be broken. The figures given here should be compared with the manufacturer's claimed repeatability of 0.5mm.

6.7.3.3 Accuracy of Continuous Path Movements

The end point accuracy of continuous path movements may be measured in two ways, by using either the joint mounted encoders or the digitizing

tablet. When using the digitizing tablet the problem of accurate initialisation is once again encountered, making it difficult to relate the co-ordinate frames of the digitizer and the robot. The joint mounted encoders provide a measure of the end point position accurate to within approximately 0.2mm in the static case. The dynamic position measurements obtained using these sensors may be verified using the data from the digitizer to give the end point tracking accuracy to a similar precision.

The measurements performed involved movement at a range of speeds and in different directions, so as to assess the overall performance. The results presented here are based on the following movements:

Number	Start Position (mm)		End Position (mm)		Velocity (mm/s)
	X	Y	X	Y	
1	0	200	0	400	50
2	0	400	0	200	50
3	0	200	0	260	20
4	0	200	0	400	100 ¹
5	100	300	-100	300	50

¹ The velocity of this movement was reduced to 80mm/s in the case of the manufacturer's controller since the higher velocity could not be attained.

Tests 1 and 2 may be considered typical of the operations required from the robot, being in the centre of the working volume and at a moderate speed. By demanding the same trajectory but in different directions, it was hoped that any effects caused by backlash would be noticable. Operations 3 and 4 are similar to 1 and 2 but were designed to give an indication of the effect of speed on the tracking accuracy. The final test was intended to give an indication of the performance when moving in a different direction and, owing to the crossing of the X = 0 axis, requires the elbow joint to stop and reverse in the middle of the trajectory.

These operations were performed using the manufacturer's PID control; a PID algorithm using the enhanced hardware and the VSC algorithms, with friction and acceleration compensation and integral action, deriving position and velocity signals from the motor alone, as described in section 6.3.4, and using joint position and motor velocity information, as detailed in 6.6.1. In all cases the end point positions were derived using both the high resolution joint mounted encoders and the digitizing tablet and the results from each source compared.

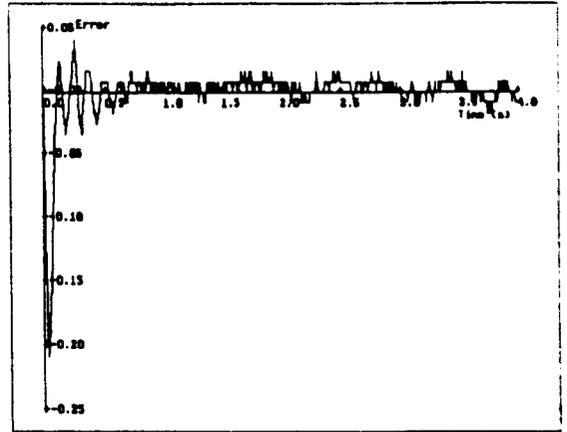
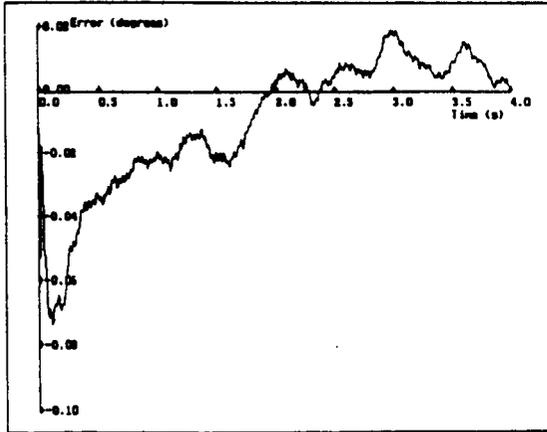
A number of the individual results are worthy of comment, since they give an indication of the sources of some of the errors found and these will be presented before a summary of the overall performance is given.

The result for the VSC systems for the first movement are shown in figure 6.44 in terms of the angular errors, as measured using the sensor used in the control loop.

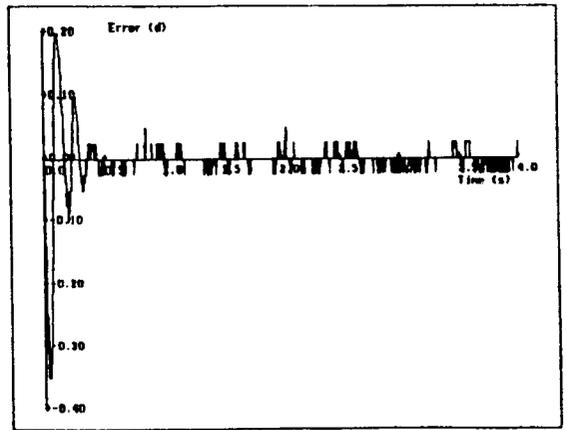
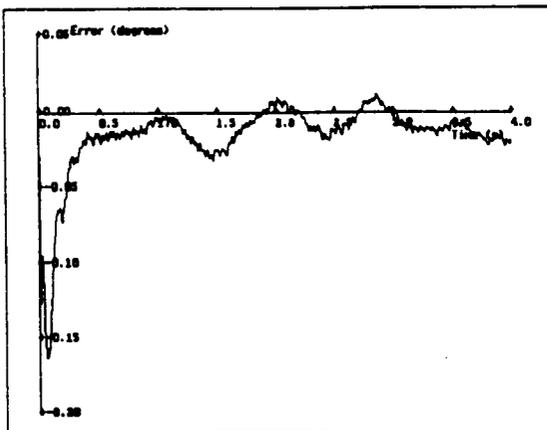
It will be seen that in both cases a relatively large error occurs during the initial acceleration phase, typically 0.20° for the shoulder and 0.35° for the elbow in the case of the joint position based controller and 0.075° and 0.16° when the motor position is employed. The errors reduce to 0.0083° and 0.025° for the joint controller and 0.02° and 0.03° using the motor position once this is over. The large initial error is a result of the infinite acceleration required to follow the trajectory. This error could be reduced by the introduction of a constant acceleration phase at the beginning of each movement. In the case of the joint mounted encoders, the position error is nearing the resolution limit of the encoder, as may be seen from the step-like nature of the result. The larger errors in the acceleration phase for the joint position based controller are due to the flexibility of the drive system which absorbs the initial movement of the motor, allowing the joint to lag behind. The control systems thus perform well in terms of the measured position. This should be contrasted with the results shown in figure 6.45 which are for the same tests but with the cartesian errors plotted.

Figure 6.44 Measured joint angular errors for VSC systems using motor and joint mounted sensors

Shoulder Error



Elbow Error



(i) Motor Position Control

(ii) Joint Position Control

It will be seen that in both cases a relatively large error occurs during the initial acceleration phase, typically 0.20° for the shoulder and 0.35° for the elbow in the case of the joint position based controller and 0.075° and 0.16° when the motor position is employed. The errors reduce to 0.0083° and 0.025° for the joint controller and 0.02° and 0.03° using the motor position once this is over. The large initial error is a result of the infinite acceleration required to follow the trajectory. This error could be reduced by the introduction of a

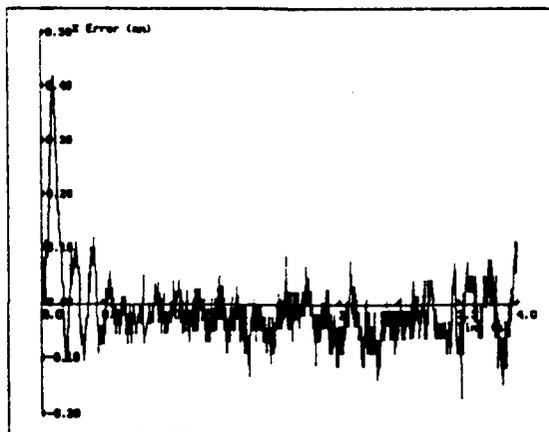
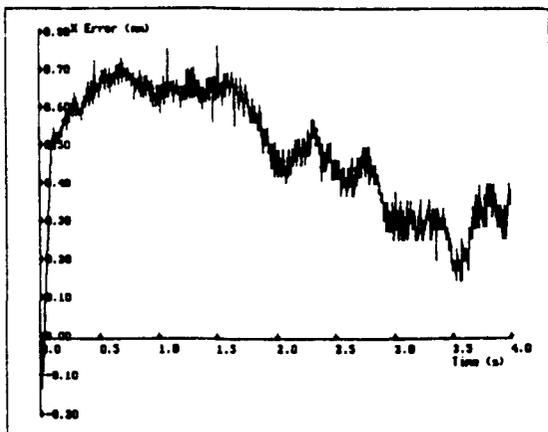
constant acceleration phase at the beginning of each movement. In the case of the joint mounted encoders, the position error is nearing the resolution limit of the encoder, as may be seen from the step-like nature of the result. The larger errors in the acceleration phase for the joint position based controller are due to the flexibility of the drive system which absorbs the initial movement of the motor, allowing the joint to lag behind. The control systems thus perform well in terms of the measured position. This should be contrasted with the results shown in figure 6.45 which are for the same tests but with the cartesian errors plotted.

Clearly, the errors for the control system using motor position are significantly larger than those for the joint-mounted encoder system. For the control system using only the motor-mounted sensor, the maximum position errors following the initial deviation are of the order of 0.8, while when joint position and motor velocity are used the maximum errors fall to approximately 0.2mm. It should also be noted that the position error during the acceleration phase is reduced from 1mm to 0.4mm when the joint positions are used. These figures may in turn be compared with the cartesian position errors obtained using the manufacturer's PID control system and the PID algorithm using enhanced hardware, shown in figure 6.46.

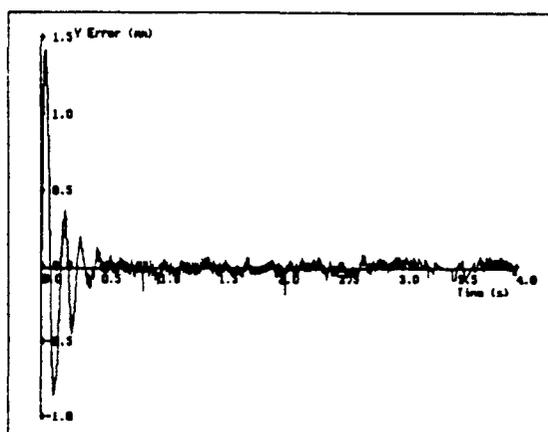
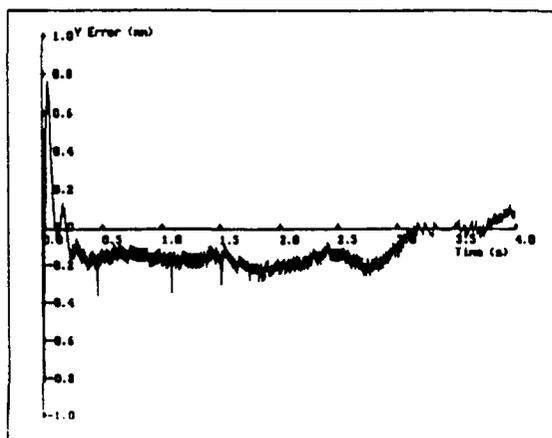
In this case the position errors have a maximum value of 2mm during the acceleration phase remains large throughout the transient. It will be noted from the results for the manufacturer's controller that a periodic signal is superimposed on the error trajectories. This is caused by mechanical resonances which are excited by the control system and by stick-slip phenomena due to static friction in the joints, a problem overcome by the use of friction compensation in the VS controllers. This effect is quite noticeable when observing the robot.

Figure 6.45 Cartesian position errors for VSC systems using motor and joint position

X Error



Y Error

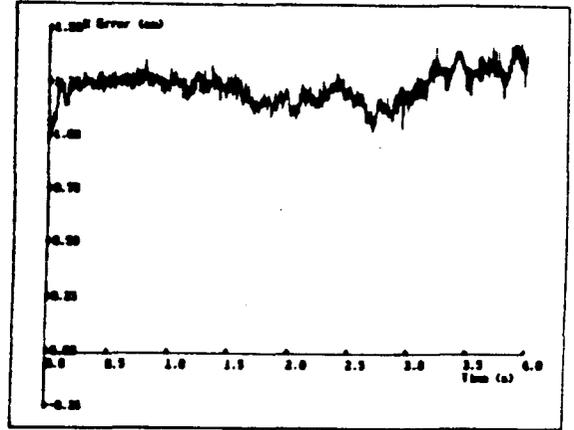
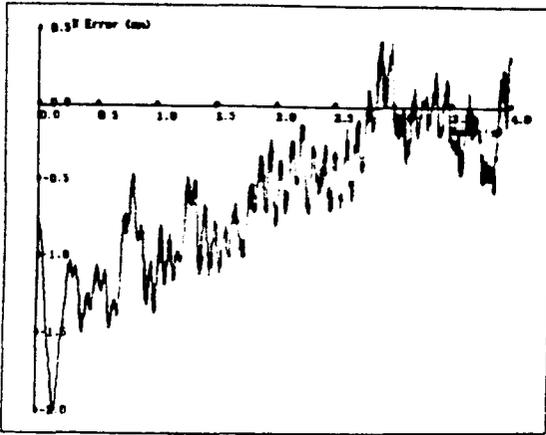


(i) Motor Position Control

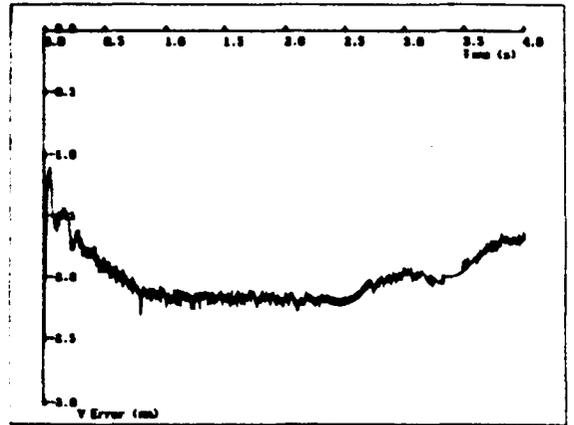
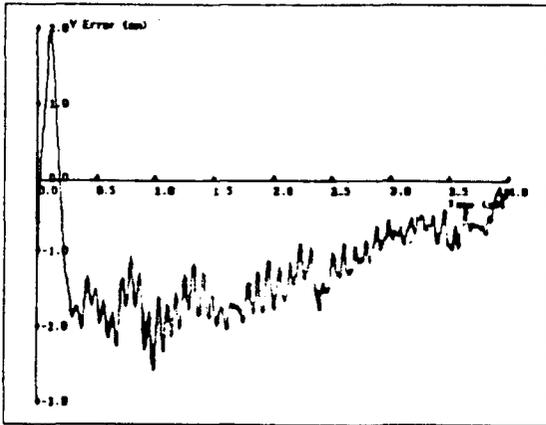
(ii) Joint Position Control

Figure 6.46 Cartesian position errors for PID control systems

X Error



Y Error



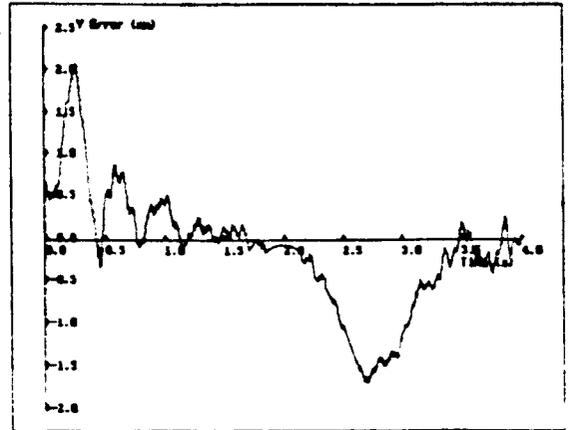
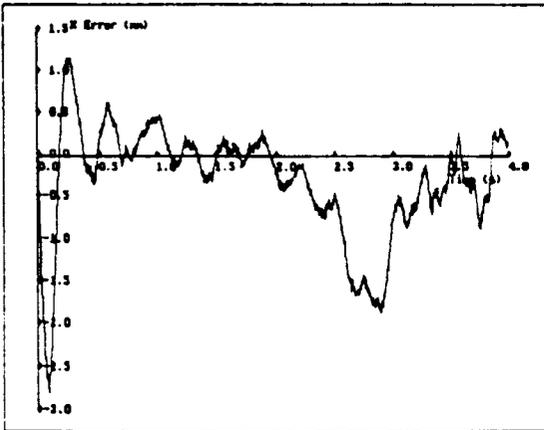
(i) Manufacturer's PID

(ii) PID using enhanced hardware

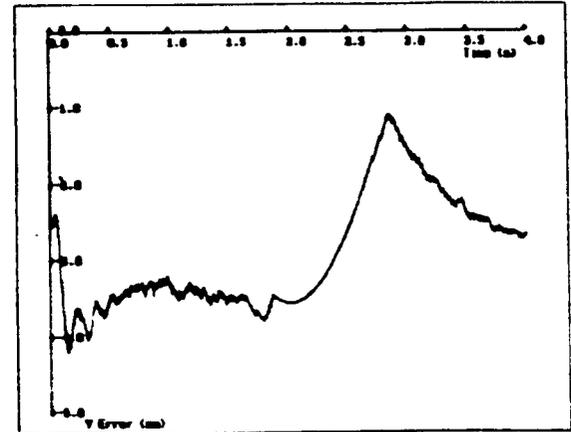
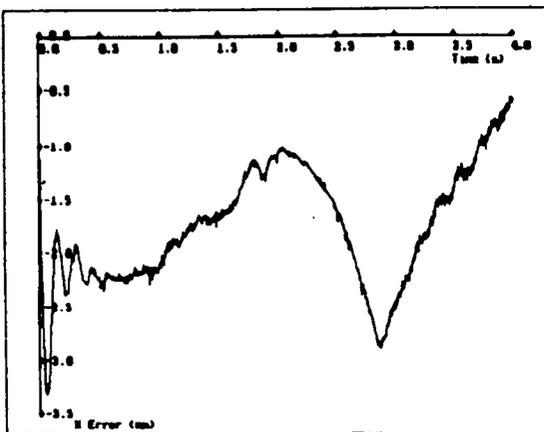
Figure 6.47 Cartesian position errors when following type 5 trajectory

X Error

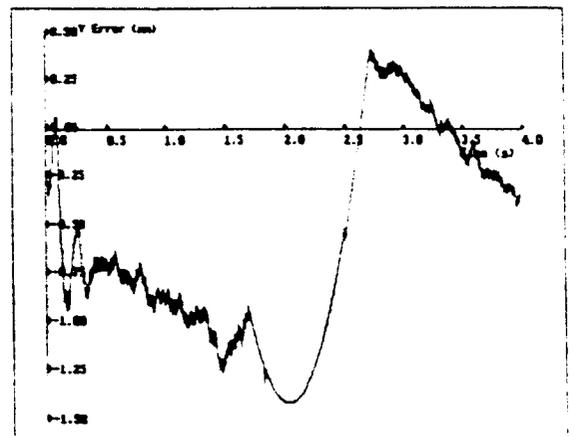
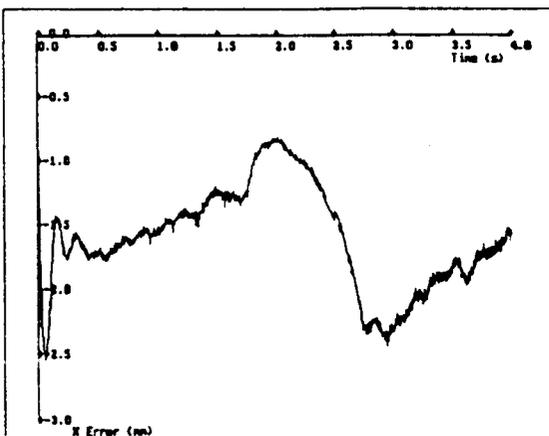
Y Error



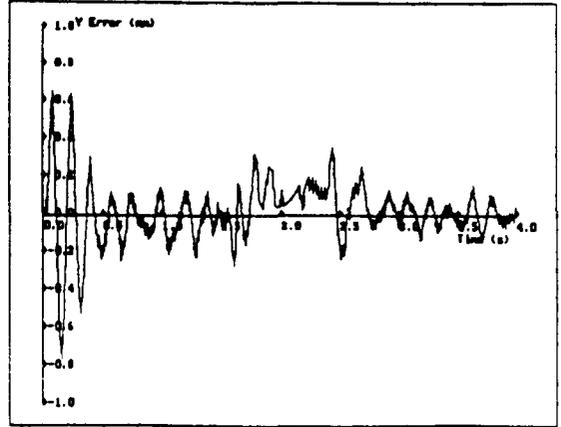
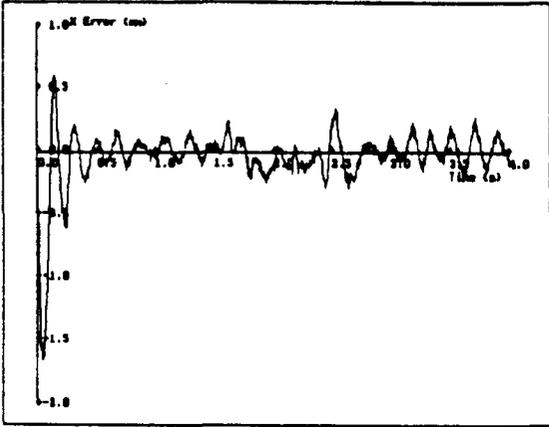
(i) Manufacturer's PID Control



(i) PID Control using enhanced hardware



(iii) Motor Position Based VSC



(iv) Joint/Motor Based VSC

The relatively large end point position errors observed for the controllers deriving position information from the motor mounted sensors are due to the flexibility and backlash in the belt drive system and errors introduced in the initialisation procedure. The effects of backlash, flexibility and Coulomb friction may be seen in figure 6.47 which shows the cartesian position errors for the three control schemes when the fifth trajectory is to be followed.

In the case of the controllers using motor position, a large error is present for the duration of the trajectory and changes form following the 2s point. At this point in time the direction of movement of the elbow joint changes so that the effects of the non-ideal drive mechanism and the joint stiction are accentuated at this point. In the case of the joint position based controller the effect of this is far less significant.

The effect of velocity on the tracking error was investigated for each control system by causing the robot to follow trajectories 3, 1 and 4, which are the same type of path but at velocities of 20, 50 and 100mm/s respectively. As expected, the tracking errors, particularly those in the acceleration phase, increase at the higher speeds. The results of these experiments are given in appendix 7.4.

The errors recorded for each of the trajectories, when performed by the four control schemes are summarised in table 6.4 in terms of the mean values

for the first 0.5s of the trajectory, the acceleration phase, and for the remainder of the response. The values given in this table are the maximum absolute error, given by:

$$e = \max (|x_i - \hat{x}_i|^2 + |y_i - \hat{y}_i|^2)^{\frac{1}{2}}$$

and the mean absolute errors, defined by:

$$e = \frac{1}{N} \sum_1^N (|x_i - \hat{x}_i|^2 + |y_i - \hat{y}_i|^2)^{\frac{1}{2}}$$

where \hat{x}_i and \hat{y}_i are the required x and y positions at sampling instant i, and N is the number of samples considered.

It can be seen that the VSC algorithm using the high resolution motor mounted encoders and current mode drives leads to an improvement in mean tracking accuracy by a factor of approximately 30% when compared with the manufacturer's PID controller and approximately 60% when compared with a PID algorithm using the same hardware. This is the case for both the initial, acceleration phase and for the remainder of the trajectory. This may be compared with the results for the VSC system using position information derived from the joint mounted encoders. In this case the mean error is reduced to less than a tenth of that for the manufacturer's system for the latter part of the trajectory but, during the acceleration phase no significant improvement is obtained. This is due to the unreasonable demand trajectory which would require infinite initial acceleration. In both of the VSC systems this acceleration cannot be achieved and so a similar maximum error occurs, although the joint position based controller reduces this error much more quickly, leading to a lower mean error.

Table 6.4 End point tracking errors for control systems (mm)

For $t < 0.5S$

Traject. Number	Manf. PID		PID		VSC / Motor		VSC/Joint+Mot	
	Max.	Mean	Max.	Mean	Max	Mean	Max.	Mean
1	2.65	1.93	2.77	2.06	1.05	0.42	1.44	0.36
2	2.69	1.56	3.48	2.92	2.46	2.09	1.66	0.49
3	2.12	1.35	2.96	1.24	0.66	0.39	0.72	0.27
4	3.59	1.60	2.32	0.76	2.28	0.55	3.14	0.77
5	2.83	1.47	4.52	4.18	2.53	1.86	1.65	0.58
Mean	2.78	1.58	3.21	2.23	1.80	1.06	1.72	0.49

For $t > 0.5S$

Traject. Number	Manf. PID		PID		VSC / Motor		VSC/Joint+Mot	
	Max.	Mean	Max.	Mean	Max	Mean	Max.	Mean
1	2.25	1.36	2.67	2.39	0.82	0.27	0.19	0.05
2	2.25	1.47	3.45	2.99	2.22	1.72	0.31	0.06
3	1.98	1.13	1.80	1.56	0.41	0.20	0.25	0.09
4	2.24	1.51	1.49	1.25	0.40	0.26	0.26	0.09
5	2.34	0.65	4.18	3.36	2.44	1.79	0.46	0.13
Mean	2.34	1.22	2.72	2.31	1.26	0.85	0.29	0.08

6.8 Selection of Control System

It has been shown that the adaptation of the position measurement and motor drive hardware and the control algorithms used in the RTX can lead to a significant improvement in performance in terms of speed of response and position accuracy for both point-to-point and continuous path movements. The cost of implementing these changes may not be justified in a small number of situations where speed and accuracy are not important, but in the majority of cases the improvement in performance is sufficient to justify at least some of

the enhancements described here.

The improvement in the speed of response by a factor of approximately three is almost entirely due to the adoption of the current mode motor drive system. This allows a maximum speed approximately twice that possible with the voltage drive used by the manufacturer and, since the motor torque is controlled, a rapid and smooth acceleration and deceleration may be obtained, without risk of motor damage due to overheating. In order to obtain adequate control at the higher speeds attained in this way, an improved velocity measurement system was required, as was an adaptation to the control system. It was seen in section 6.7.2 that the difference in performance between the PID and VS control algorithms, when tuned for maximum speed for a given movement with fixed load, was minimal, with the performance being governed by the current saturation level rather than the control system. This is to be expected in a controller which produces a saturated actuation signal for the majority of the transient, as was the case with both forms of control. Thus, once again, the improvement in speed performance was due to the development of new hardware rather than changes to the control system.

When the positioning accuracy of the robot is considered, it is found that the choice of control algorithm had a far more significant effect on performance, although the changes in hardware were also necessary for these controller enhancements to be effective. In terms of the point-to-point position repeatability, a slight deterioration in performance was seen when using the VSC algorithm using motor position compared with the manufacturer's controller. This was due to the higher torques generated using the VSC system which cause the drive system imperfections to become more apparent than in the manufacturer's PID controller. If the robot were required to apply force to its environment, the PID algorithm would suffer similar effects and so no real advantage is obtained compared with the VSC. The performance when using motor position in this mode of operation is thus limited by the design of the transmission system rather than the control algorithm. This could be improved

by adopting the drive system modifications described in section 4.3. If the joint position based controller is considered, an improvement in repeatability by a factor of two is obtained, when compared with the manufacturer's controller, although this improvement could be expected to increase in the presence of large external disturbances.

When the trajectory tracking performance is compared, it is seen that the motor position based VSC gave 30% lower end point position errors than the manufacturer's PID system and 60% lower compared with the PID algorithm using the enhanced hardware. This was due to the smoothness of response obtained using the friction compensation technique to prevent the stick-slip behaviour which occurred with the original controller. The original control performance was found to be very jerky, leading to the backlash effects not found with the VSC. Once again, using the joint position and motor velocity led to a far more significant reduction in tracking error, in this case by a factor of ten.

It is clear that the adoption of a VSC algorithm with friction compensation based on the motor position gives significant performance improvements when combined with the enhanced hardware. The greatest improvements are obtained however when the joint position and motor velocity are combined and used with a VSC algorithm.

This latter option gives the greatest performance improvement but is also the most expensive to implement, requiring high resolution encoders for the joint and motor. In particular, the 10800 lines/rev and 3600 lines/rev encoders mounted on the shoulder and elbow constitute a large component of the implementation cost. With this in mind, a number of tests were performed to investigate the effect on position accuracy of using lower resolution joint mounted encoders. The reduced resolution system was simulated by simply masking off a number of the least significant bits of the position information used in the control loop, while retaining the full precision for evaluating the performance. Both point-to-point and continuous path movements were

investigated using the same procedures as described in sections 6.7.2.2 and 6.7.2.3, although continuous path tests were only performed using the type 1 trajectory, since the errors found for this trajectory were typical of the overall performance. The resolution levels considered were set to 1, $\frac{1}{2}$, $\frac{1}{3}$ and $\frac{1}{4}$ times the maximum value for the respective encoders, giving equivalent resolutions of 10800, 5400, 2700 and 1350 lines/rev for the shoulder and 3600, 1800, 900 and 450 lines/rev for the elbow. The resulting figures for point-to-point repeatability are shown in table 6.5, while those for trajectory tracking are given in table 6.6. Plots of position error against time for the trajectory tracking tests are shown in appendix 7.5.

Table 6.5 Point-to-point repeatability with various sensor resolutions

Relative Resolution / (Shoulder, Elbow Resolution)	Deviation from Mean (mm)					
	X		Y		Radial	
	Max.	Mean	Max.	Mean	Max.	Mean
1 (10800, 3600)	0.27	0.08	0.47	0.22	0.48	0.25
$\frac{1}{2}$ (5400, 1800)	0.25	0.11	0.40	0.15	0.41	0.21
$\frac{1}{3}$ (2700, 900)	0.38	0.11	0.45	0.22	0.47	0.25

Table 6.6 Effect of sensor resolution on tracking accuracy

Relative Resolution / (Shoulder, Elbow Resolution)	Tracking Error (mm)			
	$t < 0.5$		$t > 0.5$	
	Max.	Mean	Max.	Mean
1 (10800, 3600)	1.44	0.36	0.19	0.05
$\frac{1}{2}$ (5400, 1800)	1.33	0.35	0.27	0.08
$\frac{1}{3}$ (2700, 900)	1.36	0.36	0.35	0.12
$\frac{1}{4}$ (1350, 450)	1.59	0.41	0.52	0.21

From tables 6.5 and 6.6, it is clear that the effect of the small changes in encoder resolution considered have a minimal effect on the end point repeatability while the effect on trajectory tracking accuracy is far more significant, with the mean accuracy for the latter part of the trajectory being approximately four times better in the case of the full resolution sensors, compared with the $\frac{1}{2}$ resolution result. It will also be noted that the tracking errors between the unity and $\frac{1}{2}$ resolution cases are similar, suggesting that increasing the sensor resolution beyond 10800/3600 lines/rev would not give a further performance improvement with this control scheme. It would therefore be recommended that a control system be adopted, based on the VSC algorithm described using motor velocity and joint position, sensed with a resolution of typically 5000 and 2500 lines/rev for the shoulder and elbow respectively.

The most appropriate sensor system for a particular application will depend on the accuracy required and the particular cost/accuracy trade-off. It should be noted that even with the lowest resolution simulated here, the tracking error is superior to that achieved with the motor mounted sensors. If encoder resolutions of the order of 5000/2500 lines/rev are required, the most cost effective solution would be to employ lower resolution sensor (typically 500 lines/rev) driven from the joints via anti-backlash gearing of the appropriate ratio to give similar effective resolution (typically 10:1 and 5:1 for shoulder and elbow respectively). Since the moment of inertia of an encoder disc of this resolution is small ($< 10^{-7} \text{kgm}^2$) the effective inertia added to that of the arm will be insignificant compared with that of the arm itself. Also, since the torque required to overcome friction and accelerate such a small disc would be minimal, the antibacklash gearing used could be of a small size and so would not affect the joint dynamics. The encoder mounted on the motor need not have a resolution as high as that used here, with a sensor of 100 lines/rev giving sufficient velocity resolution at a little lower cost.

To improve the overall accuracy of the RTX, a more accurate initialisation procedure is required. This could be achieved using the moderate resolution

sensor and antibacklash gearing system described, if the encoder with an index channel and an additional low resolution sensor, such as an optical interrupt switch, were combined to give a unique position indication.

The motor drive circuit should be left unchanged since it has performed well throughout and the cost savings which could be made by redesign would not justify the time required.

CHAPTER 7

Evaluation of the Design of Nonlinear Controllers for Nonlinear Systems

In the preceding chapters, the design of nonlinear control systems for plants which are themselves nonlinear has been considered with reference to a particular set of control methodologies (Variable Structure Controllers) applied to a specific nonlinear system (the RTX robot). This has involved the study of the dynamic modelling of the robot arm, the enhancement of a computer aided design suite to make it suitable for the robot dynamics, the development of new hardware for the robot and the implementation and testing of a range of controllers for a number of movement types. Although results pertinent to the particular subject considered have been given in each chapter, it remains to discuss these results within the framework of the overall project. In this chapter we will consider the work carried out on the enhancement of the computer aided design suite, the development of new robot hardware, the determination of a dynamic model of the RTX and the design of new control systems within this framework.

7.1 Computer Aided Design Suite Enhancements

The alterations made to the CACSD suite covered a number of aspects, having effects of varying significance for the user, from simply allowing linear matrix data to be modified rather than re-entered, to changes in the model structure and the solution algorithm. A number of changes were made as a result of improvements in the hardware available, allowing a virtual filing system to be operated which in turn made the use of larger model structures a

practical proposition. Increasing the number of system states from five to twenty allows the majority of general systems to be considered and enables the investigation of the full nonlinear model of two links of the RTX.

The modifications made to the system model structure described in section 5.3 were shown in section 5.5 to allow the simulation of the interaction dynamics of a two link planar robot to a reasonable accuracy within an acceptable time scale. Although the time involved was a little longer than that achieved using SIMNON, the benefits of having other parts of the CACSD suite available as part of the same package may outweigh this deficiency. In particular, the system representation interchange routines prove useful, as does the possibility of adding user implemented code to the existing routines.

While studying the application of discontinuous control schemes to dynamical systems, it was found that the numerical solution algorithm implemented became inefficient in circumstances where a non-smooth nonlinearity was present. This prompted the investigation of a number of alternative algorithms and a comparison between them for this type of system. This also provided the opportunity of comparing the performance of the PHAS algorithm with other better established methodologies for a variety of system types, a comparison which had not previously been thoroughly made. The algorithms compared are typical of those used in general simulation packages, being a fixed step size fourth order Runge-Kutta and a fourth/fifth order Runge-Kutta-Fehlberg with variable step length. The PHAS algorithm was also tested using both fixed and variable step size. All these algorithms used a single step mechanism. Multi-step procedures were not considered viable, for the reasons given in section 5.4.4.

It was found that the most appropriate algorithm in a particular instance depends on the nature of the differential equation considered and the accuracy of solution sought. In general, for differential equations with smoothly varying right hand sides, all algorithms performed as expected, with fixed step algorithms giving errors proportional to the square of the step size in the case

of the PHAS algorithm and the fourth power of the step size for the Runge-Kutta methods. Thus, if a relatively large error is acceptable, the PHAS algorithm is faster in computation, while if greater accuracy is sought, the fourth order Runge-Kutta method becomes preferable. It was noted that the PHAS algorithm gave a smoother phase plane trajectory than the Runge-Kutta result because of the different interpretation of the time variable; in the case of PHAS as an extra state, and in the Runge-Kutta method as a reference against which the states evolve.

The variable step size versions of the algorithms performed well in this situation with the errors produced being predictable from the error tolerance set, provided a reasonable step length was allowed. These algorithms were however more expensive computationally than their fixed step counterparts, for a given result accuracy. This reduction in efficiency, by a factor of typically 50%, may well be tolerable in a general purpose package, given the benefits of improved reliability. Of the variable step size algorithms tested, the PHAS algorithm proved more computationally efficient than the Runge-Kutta-Fehlberg algorithm for smooth equations since in the former case the number of function evaluations per step is smaller and only a small amount of step length adaptation is required.

The relative efficiency of the algorithms becomes noticeably different when differential equations with discontinuous right hand sides are considered. Here the fixed step length algorithms give errors of the order of the step length, with the PHAS algorithm being typically five times more efficient for the same errors as the higher order algorithms. The decision of whether or not to use a variable step algorithm, and which one to adopt, depends to a large extent on the number of discontinuities found in the trajectory considered. If only a small number of discontinuities are encountered, a variable step size algorithm is most appropriate, since the step size will be reduced at the discontinuity without significantly adding to the computation time of the continuous parts of the trajectory. In this case the variable step size PHAS algorithm is most suitable,

although the difference between this and the Runge-Kutta-Fehlberg method is small. If a large number of discontinuities are crossed by the trajectory, as found for instance in a sliding mode control system, the large amount of step size adaptation required makes the variable step length algorithms less efficient than fixed step length routines giving the same accuracy of results. The variable step length PHAS algorithm was particularly inefficient in this situation, owing to its simple step length adaptation procedure.

These comments may be summarised by saying that for systems with few or no discontinuities the variable step length PHAS algorithm should be used, while if a large number of discontinuities are encountered, the fixed step length PHAS algorithm or the variable step length Runge-Kutta-Fehlberg method should be adopted. In accordance with this, both PHAS and the Runge-Kutta-Fehlberg algorithms remain available within the PHAS package with the option of either fixed or variable step length in both cases. A further development of the package might involve algorithms which select the most appropriate algorithm in a particular situation depending for instance on the presence or absence of sliding behaviour. This could either be implemented as an advisory system to aid the user or could perform the operation automatically, possibly during the simulation.

7.2 RTX Hardware Development

It would be true to say that the most significant improvements in the performance of the RTX were achieved through the development of new hardware for position and velocity measurement and for the motor drive system. In particular, the adoption of a current drive system, in place of the voltage drive provided by the manufacturer, resulted in a two-fold increase in the maximum joint velocity, without increasing the power dissipated in the motor and, owing to the improved control of torque achieved, allowed a further improvement in response speed to give an overall increase by a factor of three.

This increase in speed could not be combined with good control without the superior position and velocity measurement achieved through the use of high resolution optical encoders mounted on the motors in place of the low resolution sensors used by the manufacturer. The 500 line/rev encoders used in this instance provided a higher resolution than necessary but were the most cost effective option readily available at the time. If it proved less expensive to adopt sensors with a resolution of 50 or 100 lines/rev, it appears that this could be done without any significant loss of control performance.

It was shown in section 6.7.3 that a significant improvement in position accuracy for both point-to-point and continuous path movements may be achieved by the use of high resolution sensors mounted on the joints themselves. Obtaining joint position using encoders with resolutions of 10800 and 3600 lines/rev on the shoulder and elbow joints respectively, combined with velocity information derived from the high resolution motor mounted encoders, gave an improvement in mean end point repeatability by a factor of between two and six when compared with control systems using motor position information alone, and an improvement of up to fifteen times when compared with the manufacturer's system. Similarly, when comparing the performance for smooth trajectory following, it was found that using these joint mounted sensors gave an improvement in tracking accuracy by a factor of between ten and fifteen times compared with the motor position based controllers for the majority of the trajectory. It was also noted, in section 6.8, that although the cost of the encoders used was relatively high, a limited reduction in encoder resolution could be implemented with only a slight loss of accuracy. It was also suggested that, by adopting a lower resolution sensing system attached to the joint via anti-backlash gears, the same end point accuracy could be achieved at a greatly reduced cost. It was therefore concluded that the most effective hardware arrangement would be to have moderate resolution sensors on both the motors and the joints, with the joint sensors driven via anti-backlash gears. The new current drive system should be left unchanged.

An alternative approach which could be adopted to improve the position accuracy of the RTX, without the use of joint mounted sensors, would be a modification to the belt drive system used. By using larger radius pulleys where possible, the effect of elasticity and backlash would be proportionately reduced. Using belts with a different tooth shape could also reduced backlash. These ideas are further described in section 4.3 and although they would give some improvement, they would not give such good results as those obtained using joint mounted sensors, and would involve greater implementation cost.

7.3 Evaluation of the Dynamic Model of the RTX

The dynamic modeling of any practical system involves some degree of compromise since there are always additional effects which could be considered which would lead to a more accurate representation of the true behaviour. It is therefore up to the engineer to decide at what point the cost and time involved in expanding the system representation ceases to be worthwhile in terms of the improvement of model accuracy. In the case of the shoulder, elbow and yaw joints of the RTX, a great many effects have been considered, as described in chapter 4, leading to a significant improvement in behaviour prediction compared with other published models.

The major improvements to the model described in that chapter concern the effects of nonlinear friction in the joints and the elastic behaviour of the joint transmission system. These aspects of the behaviour have a very significant effect on the overall performance of the RTX and have not previously been considered. Thus, although the model derived in chapter 4 was seen to be limited in it's accuracy, it represents a significant improvement over previous representations available.

A number of known limitations of the model derived were not considered in detail because the time involved could not be justified at this stage. The most significant of these effects are backlash in the timing belt drive system

and twisting of the Z-column and sliding carriage. The former effect was not considered because of the highly unpredictable nature of the friction between the belts and the pulleys, making any reliable model very difficult to obtain, while in the second case, the degree of twisting could not be measured without a large amount of instrumentation which was not available. Both these effects are the subject of on-going research which, it is hoped, will lead to further improvements in the dynamic model of the RTX, as will work on the remaining axes not considered here.

7.4 Evaluation of Variable Structure Control Systems

A large number of control algorithms have been implemented on at least one joint of the RTX and the resulting performance investigated for a variety of movements. These ranged from the well established PID algorithm to novel variable structure algorithms deriving position and velocity information from different points in the drive chain and including well established VSC techniques and a number of enhancements to the basic methodology, using friction and acceleration compensation and integral action.

Various hardware configurations have also been adopted, involving both high and low resolution motor-mounted and high resolution joint-mounted position and velocity sensors, and both voltage and current mode motor drive circuits. Control algorithms were implemented using both hardware and software. Although some of the variations in hardware provide significant performance benefits, these do not give an indication of the success of the control algorithm adopted. Accordingly, it is necessary to compare algorithms implemented using the same hardware for position and velocity measurement and for joint actuation. The arrangement considered will be that of high resolution motor-mounted position and velocity sensors and current mode motor drive system, with the control algorithm implemented in software.

The variable structure algorithms may be divided between those that adopt

a discontinuous control signal and those having some form of smoothing to avoid the chatter inherent in discontinuous control systems. It was shown in chapter 6 that both forms of control could be successfully applied to the shoulder and elbow joints of the RTX, leading to the type of dynamic behaviour expected.

In the case of the simple VSC with a discontinuous control signal, a large amount of chatter was observed due to the significant nature of the unmodeled dynamics present in the system. The problem of chatter was particularly apparent when a curved switching function was adopted with the aim of removing the reaching phase, so as to make the whole response invariant to parameter changes. It was found that a large amount of chatter occurred in the initial stages of the trajectory when the discontinuity in the control signal was greatest. The high amplitude of chatter resulted in a response which was highly susceptible to the effects of load changes, showing that the hoped for invariance was not achieved. In addition to these problems, a large steady state error was found, and although this could be reduced by the use of friction compensation, the effect of poor friction modeling was either a substantial steady state error or overshoot and limit cycling.

When variable structure control systems with a boundary layer were considered, it was found that the chattering behaviour was eliminated provided the gain was set below a certain level. The switching line following characteristics were, however, fairly poor and a steady state error similar to that found with the simple VSC occurred. Both of these problems were largely due to the significant level of Coulomb friction present in the system and could be overcome to some extent by the use of friction compensation. It was also noted that over-estimation of the friction did not cause the limit cycling problems which occurred with the simple VSC in similar circumstances. For this reason, the compensation level could be set closer to the measured friction value without fear of the undesirable behaviour found in the simple VSC.

In order to attain good switching function following, it was necessary to

add acceleration compensation to the friction compensated controller. This gave a significant improvement in the performance, but was not entirely robust to changes in load. This problem, as well as the problem of steady state errors due to inaccurate friction compensation, were largely overcome by the addition of integral action to the algorithm. This had the effect of reducing any problems caused by poor modeling and parameter variations, and resulted in a controller giving accurate and rapid response with good invariance to parameter changes. The result was better than that achieved using integral action alone since, in the former case, the level of integral gain did not need to be set as high as when only integral action is used. This reduces the destabilising effect which a large amount of integral action caused. It was therefore considered that all these elements of the control algorithm should be included in order to maintain the good controller performance achieved.

The maximum speed of response attained using the control algorithm with compensation and integral action for a particular movement was the same as that attained using the PID algorithm within the limits imposed by the variable nature of the dynamics of the RTX. This response is close to the time optimal behaviour and involves a saturated actuation signal for the majority of the transient.

The repeatability of the VSC and PID algorithms using the enhanced motor drive system and motor position sensing for point-to-point movements was similar, with the VSC being only slightly more accurate. With the algorithm developed to employ joint position and motor velocity, the accuracy of both point-to-point and continuous path movements improves drastically, compared with the motor position based controllers.

Thus the most effective of the control algorithms considered here, in terms of speed of response and accuracy, employs motor velocity and joint position in a VSC with boundary layer having friction and acceleration compensation and integral action.

CHAPTER 8

Conclusions

It has been shown in the preceding chapters that the shoulder, elbow and yaw joints of the RTX constitute a highly complex nonlinear system having many features which make traditional linear control system design procedures ineffective. It was noted that the most significant forms of non-ideal behaviour were due to Coulomb friction in the motors and joints and flexibility in the timing belt transmission system. These characteristics make it difficult to obtain an accurate model of the RTX and this leads to problems in controller design and parameter selection. The inclusion of nonlinear friction and drive system flexibility in the dynamic model described in chapter 4 give a significant improvement in modeling accuracy compared with other available models of the RTX. This improvement in model accuracy has led to improvements in control performance, particularly through the use of friction compensation in the variable structure control system.

In order to study the full dynamics of the RTX and similar robots, it was necessary to enhance the computer aided control system design suite and it was shown in chapter 5 how this could be achieved. Chapter 5 also provided useful results concerning the performance of various numerical algorithms for the solution of differential equations, a comparison which had not previously been made. It was shown that the PHAS algorithm provided significant benefits when compared with better established methods for a number of types of dynamic system, in particular those in having discontinuous right hand sides, as is the case with variable structure control systems.

By combining improvements to the actuation and position/velocity measurement hardware used in the RTX with the development of new control algorithms it was possible to realise significant performance improvements

compared with the manufacturer's controller in terms of speed of response and position accuracy. The improvement in speed of response for point-to-point movements by a factor of three was largely due to developments in the motor drive electronics, while the improvement in end point accuracy by a factor of between two and fifteen was due to both hardware and control algorithm enhancements.

These improvements were achieved at a fraction of the cost of the original robot and make the RTX effective in many applications for which it would not otherwise be suitable.

References

Chapter 1

Ali, A. and Taylor, P.M. "Nonlinear Control of an Industrial Robot" Proc. IEEE Conf. on Applications of Adaptive and Multivariable Control, Hull 1982.

Chen, N. and Dwyer, T.A.W. "Single Step Optimisation of Manipulator Maneuvers with Variable Structure Control" Proc. American Control Conf., Vol. 1, p. 107-112, 1987.

Fournier, S.J. and Schilling, R.J. "Decoupling of a Two-Axis Robotic Manipulator using Nonlinear State Feedback: A Case Study" Int. Journal of Robotics Research Vol. 3, No. 3, p. 76-86, 1984.

Hirio, M.; Hojo, M.; Hashimoto, Y.; Ade, K. and Dote, Y. "Microprocessor-Based Decoupled Control of Manipulator Using Modified Model Following Method with Sliding Mode" Proc. IECON, p. 405-409, 1984.

Morgan, R.G. and Özgüner, Ü. "A Decentralized Variable Structure Control Algorithm for Robotic Manipulators" IEEE Journal of Robotics and Automation, Vol. RA-1, No. 1 p. 57-65, 1985.

Paul, R.P. "Robot Manipulators - Mathematics, Programming and Control" MIT Press, Cambridge MA, 1981.

Sahba, M. and Mayne, D.Q. "Computer-Aided Design of Nonlinear Controllers for Torque Controlled Robot Arms" IEE Proc. Part D, No. 1, p. 8-14, 1984.

Young, K-K.D. "Controller Desing for a Manipulator using Theory of Variable Structure Systems" IEEE Proc SMC. Vol. SMC-8, No. 2, P. 101-109, 1978.

Chapter 2

Canudas, C.; Åström, K.J. and Braun, K. "Adaptive Friction Compensation in DC Motor Drives" Proc. IEEE Conf. on Robotics and Automation, p. 1556-1561, 1986.

Carignan, C.R. and Akin, D.L. "Tracking and Station Keeping for Free-Flying Robots using Sliding Surfaces" Proc IEEE Conf. on Robotics and Automation, p. 969-974, 1988.

Chen, N. and Dwyer, T.A.W. "Single Step Optimisation of Manipulator Maneuvers with Variable Structure Control" Proc. American Control Conf., Vol. 1, p. 107-112, 1987.

Dote, Y.; Shinojima, M. and Hoft, R.G. "Digital Signal Processor Based Novel Variable Structure Control for Robot Manipulators" Proc. IECON '87, p. 175-179, 1987.

Hashimoto, H. "Variable Structure Strategy for Motion Control Systems - Applications to Electrical Machines" Proc. IECON '87, p. 159-165, 1987.

Hashimoto, H.; Yamamoto, H.; Yanagisawa, S. and Harashima, F. "Brushless Servo Motor Control Using Variable Structure Approach" IEEE Trans. Industrial Applications, Vol. IA-24, No. 1 p. 160-170, 1988.

Hirio, M.; Hojo, M.; Hashimoto, Y.; Ade, K. and Dote, Y. "Microprocessor-Based Decoupled Control of Manipulator Using Modified Model Following Method with Sliding Mode" Proc. IECON, p. 405-409, 1984.

Itkis, U. "Control Systems of Variable Structure" John Wiley, 1976.

Jobling, C.P. and Taylor, P.M. "A Computer Aided Design Suite for the Design of Nonlinear Feedback Control Systems using the Phase Plane" Proc. IASTED Conf. Telecon '84, 1984.

Kubo, T.; Anwar, G. and Tomizaka, M. "Application of Nonlinear Friction Compensation to Robot Arm Control" Proc. IEEE Conf. on Robotics and Automation, p. 722-727, 1986.

Lin, J-L. and Lin, W-T. "Application of Variable-Structure System in DC-Motor Velocity Control" Modelling, Simulation and Control (France), Vol. 15, No. 2, p. 37-52, 1988.

Morgan, R.G. and Özgüner, Ü. "A Decentralized Variable Structure Control Algorithm for Robotic Manipulators" IEEE Journal of Robotics and Automation, Vol. RA-1, No. 1 p. 57-65, 1985.

Parvinmehr, A.; Greenough, R.D.; Wilkinson, A.J. and Jenner, A. "Discrete Variable Structure Control for an Active Anti-Vibration System", Proc. IFAC ACASP '89, Glasgow p. 547-552, 1989.

Ryan, E.P. "Optimal Relay and Saturated Control System Synthesis" P. Peregrinus, 1981.

Slotine, J-J.E. and Sastry, S.S. "Tracking Control of Non-Linear Systems using Sliding Surfaces, with Application to Robot Manipulators" Int. Journal of Control, Vol. 38, No. 2, p. 465-492, 1983.

Slotine, J-J.E. "Sliding Controller Design for Non-Linear Systems" Int. Journal of Control, Vol. 40, No. 2, p. 421-434, 1984.

Staszulonek, A. and Van-Brussel, H. "Inertially Decoupled, Sliding Mode Controller Design for Trace and Pick-Up Robot" 16th ISIR, p. 457-467, 1986.

Taylor, P.M. and Gilbert, J.M. "Computer Aided Design of Variable Structure Controllers for Robots with Significant Drive Nonlinearities" Proc. 4th IFAC Symposium on CACSD, Beijing, 1988.

Taylor, P.M. and Jobling, C.P. "Computer Aided Design of Piecewise Linear Switching Surfaces for Nonlinear Systems" Proc. IMC Workshop on Computer Aided Control System Design, Salford p. 37-41, 1986.

Utkin, V.I. "Variable Structure Systems with Sliding Modes" IEEE Trans. on Automatic Control, Vol. AC-22, No. 2, p. 212-222, 1977.

Utkin, V.I. "Variable Structure Systems: Present and Future", Automatic and Remote Control (USA), Vol. 44, No. 9, p. 1105-20, 1983.

Yeung, K.S. and Chen Y.P. "A New Controller Design for Manipulators using the Theory of Variable Structure Systems" IEEE Proc. on Automatic Control, Vol. AC-33, No. 2, p. 200-206, 1988.

Young, K-K.D. "Controller Desing for a Manipulator using Theory of Variable Structure Systems" IEEE Proc SMC. Vol. SMC-8, No. 2, P. 101-109, 1978.

Zinober, A.S.I. "Adaptive Relay Control of Second Order Systems" Int. Journal of Control, Vol. 21, No. 1, p. 81-98, 1975.

Zinober, A.S.I. "Adaptive Variable Structure Systems", Proc. 3rd IMA Conf. on Control Theory, p. 343-360, 1980.

Zinober, A.S.I. "Variable Structure Model-Following Control for a Robot Manipulator" Proc. 4th IFAC Symposium on CACSD, Beijing, p. 175-180, 1988.

5

Chapter 3

Bühler Private communication.

Digital "PDP-11 Microcomputer Interfaces Handbook" Digital Equipment Corporation, 1983.

Makino, H. and Furuya, N. "Selective Compliance Assembly Robot Arm" Proc. Assembly Automation Conference, Brighton 1980.

Chapter 4

Armstrong, B. "Friction: Experimental Determination, Modeling and Compensation" Proc. IEEE. Conf. on Automation and Control, p. 1422-27, 1988.

Billings, S.A. and Voon, W.S.F. "Piecewise Linear Identification of Non-linear Systems" Int. Journal of Control, Vol. 46, No. 1, p. 215-235, 1987.

Bowden, F.P. and Tabor, D. "Friction and Lubrication of Solids, Volumes 1 and 2" Oxford University Press 1950 (Vol. 1) and 1964 (Vol. 2)

Bühler Private communication

Chen, S. and Billings, S.A. "Prediction-error Estimation Algorithm for Non-linear Output-affine Systems" Int. J. Control Vol.47 No.1 1988 p309-332

Doebelin, E.O. "System Modeling and Response" Wiley, 1980

Dowson, D. "History of Tribology" Longman, 1979

Eykhoff, P. "System Identification" Wiley, 1974

Hayden, H.W. Moffat, W.G. and Wulff, J. "The Structure and Properties of Materials - Volume 3, Mechanical Properties" Wiley, 1965

Moore, D.F. "Principles and Applications of Tribology" Pergamon Press, 1975

SIMNON "Simnon User's Guide" Lund Institute of Technology, 1986

UMI Private communication

Van Vlack, L.H. "Materials Science for Engineers" Addison-Wesley, 1970

Chapter 5

Conte, S.D. and DeBoor, C. "Elementary Numerical Analysis: An Algorithmic Approach", McGraw-Hill, 1980.

- Fröberg, C-E. "Introduction to Numerical Analysis 2nd Edition", Addison-Wesley, 1969.
- Geering, H.P.; Guzzella, L.; Hepner, S. and Onder, C. "Time Optimal Motions of Robots in Assembly Tasks", IEEE Trans. in Automatic Control, Vol. AC-31, No.6, p. 512-518, 1986.
- Hull, T.E. et al "Comparing Numerical Methods for O.D.E.s" SIAM Journal of Numerical Analysis, Vol. 9 p603-637, 1972.
- Jobling, C.P. and Taylor, P.M. "A Computer Aided Design Suite for the Design of Nonlinear Feedback Systems Using the Phase Plane", Proc. IASTED Conf. TELECON '84, 1984.
- Johnson L.W and Reiss, R.D. "Numerical Analysis 2nd Edition" Addison-Wesley, 1982.
- Morris, W.D. "Differential Equations for Engineers and Applied Scientists", McGraw-Hill, 1974.
- Sawyer, A.N. "CACSD Filing System" University of Hull, Dept. of Electronic Engineering, Internal Report No. 79/89, 1986.
- Taylor P.M. "The Hull Control System Design Suite", Proc. INSTMC Workshop on Computer Aided Control System Design, Brighton, U.K., p. 73-78, 1984.

Chapter 6

Geering, H.P.; Guzzella, L.; Hepner, S. and Onder, C. "Time Optimal Motions of Robots in Assembly Tasks", IEEE Trans. in Automatic Control, Vol. AC-31, No.6, p. 512-518, 1986.

Ryan E.P. "Optimal Relay and Saturated Control System Synthesis" P. Peregrinus, 1981.

Wallenberg, A. and Åström, K.J. "Limit Cycle Oscillations in High Performance Robot Drives", Proc. IEE Conf. Control '88, p. 444-449, 1988.

APPENDIX 1

Motor Drive Power Amplifier Data

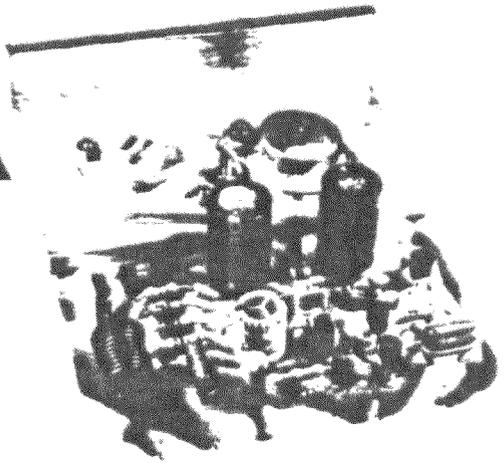
The MOSFET power amplifier used to drive the joint motors is described in the following pages. In order to operate as a DC amplifier, two decoupling capacitors must be short circuited. These are C_1 , the input decoupling capacitor and C_3 , a feedback decoupling capacitor. For the reasons described in section 3.3.2, it is necessary to introduce a pole in the amplifier characteristics at a frequency of 2kHz. This is most easily achieved by adding a 2200pF capacitor in parallel with the feedback resistor, R_7 . The remainder of the circuit was left unchanged.

MOSFET AMPLIFIER

An incredible Hi-Fi Amp that's virtually bomb-proof — like the best valve amps

Specification

Power output:	> 75W RMS into 4Ω > 50W RMS into 8Ω 650mV RMS for rated output
Sensitivity:	47%±
Input impedance:	44-0-44V DC, 2A
Power supply:	20Hz to 20kHz virtually flat
Frequency response:	10Hz to 40kHz ±1dB 20Hz to 20kHz < 0.005% 1kHz < 0.002%
Total harmonic distortion:	120dB
Signal-to-noise ratio:	



by Dave Goodman

Power MOSFETs are a relatively new addition to the range of semiconductor devices available. Small signal MOSFETs have been around for some years, mostly finding uses in high frequency applications, but it was found difficult to make MOSFETs with gate-to-drain voltages greater than 30V (most are rated at 20V) and with high current capabilities such as would be required in power amps. The high voltages possible with power MOSFETs are achieved by separating the gate and drain layers with a layer of ion implanted silicon dioxide. In addition, a field plate is provided on the source, near the gate. These two measures prevent electric field concentration which at high voltages would otherwise destroy the gate. The high current capability is achieved by using a comb-shaped structure for the drain and source regions.

Figure 1 shows the connection configuration for the MOSFETs used in this amplifier. Although a TO3 package is used to give excellent heat dissipation, MOSFETs are far superior to bipolar transistors in their response to high temperatures. As a bipolar transistor heats up for a given voltage, the current through it becomes greater, i.e. it has a positive temperature characteristic. If the temperature were allowed to continue to rise thermal runaway would ensue and the transistor would be destroyed. A MOSFET, however, has a negative thermal characteristic. As the transistor becomes hotter the current tends to decrease so

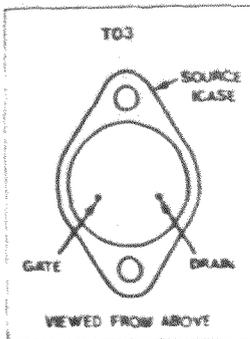


Figure 1: Power MOSFET package

power MOSFETs are most unlikely to be destroyed due to high temperatures.

Power MOSFETs also have a far wider frequency response than bipolar power transistors, so that a very wide and extremely flat frequency response can be obtained without any complicated circuitry. Figure 3(a) shows the typical output characteristics of a power MOSFET for gate-to-source voltages (V_{GS}) from 1V to 10V in 1V steps. Figure 3(b) shows the remarkably low total harmonic distortion generated by this amplifier. It is scarcely measurable even with the best test equipment available and certainly far below the minimum audible level.

Circuit Description

TR1 and TR2 form a static differential input buffer amplifier. The bias current for each transistor being set to 0.5mA. The 2SA872 transistor is used because it has a very low noise



Figure 2: MOSFET symbols

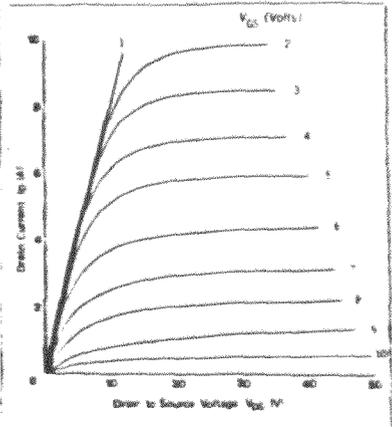


Figure 3(a): Typical output characteristics

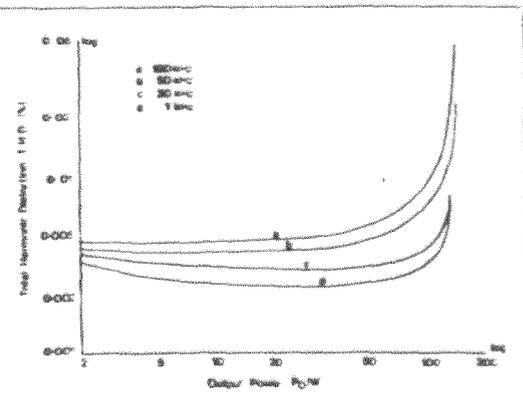


Figure 3(b): Harmonic distortion graph

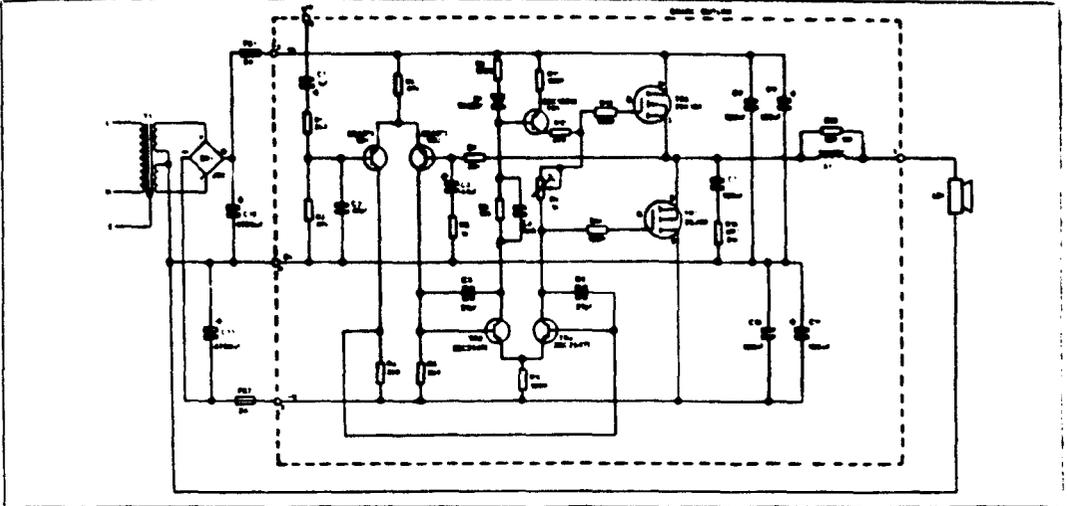


Figure 4. Circuit diagram for single channel amplifier

output but can handle high voltages. TR3 and TR4 form a 'current mirror' to give a high open loop voltage gain. TR5 acts as a constant-current load and this low-noise, high-gain, class A amplifier stage is all that is required to drive the power MOSFETs TR6 and TR7. The transistors in the driver stage need to have a high voltage durability, high F_T and low C_{gs} . They also have to supply sufficient power to charge and discharge the gate to source capacitance of the power MOSFETs. In this case a bias current of around 50mA is sufficient to ensure adequate power is available at all frequencies and power levels.

The input impedance of the amplifier is set by R2, to 47k, and C2 bypasses any RF signals present at the input. The amplifier has a gain of 33, and this is set by R7 and R6, via decoupling capacitor C3. R13 and R14 improve the stability at high frequencies by reducing the effective gate load capacitance. C7 and R15 are a Zobel network which, in conjunction with R16 and L1, ensures excellent stability into reactive loads at high frequencies.

Construction

Fit the five Veropins, labelled 1 to 5, to the PCB and solder. Fit and solder diode D1 taking care that it is the right way round. Fit and solder all the resistors except R16, and all the capacitors, taking care with the polarity of the electrolytic ones. C1, C3, C9 and C11 (refer to Figure 5). Scrape or burn the enamel off one end of the piece of enamelled copper wire

and solder it to one lead of R16, close to the body of the resistor. Now wind the wire tightly around the resistor ten times to form L1, as shown in Figure 6. Do not cut the wire, but hold it tightly and scrape off the enamel where it will touch the other lead out wire of the resistor, then wrap it around the lead and solder. Fit this composite component to the PCB and solder. Fit and solder the preset to the PCB, then the transistors (TR1-5).

Make the heatsink bracket shown in Figure 7 (Note that this is available ready-made and is included in the kit supplied by Maplin Electronic Supplies Ltd). The mounting bracket fits to the component side of the PCB as shown in the photograph. Align it with the holes in the PCB and put one bolt through the centre hole from underneath using a 6BA nut, bolt and shakeproof washer. Referring to Figure 8, place a nylon bush in each of the four large holes in the bracket, smear both faces of both mica washers with Thermopack silicone grease and place these in position. Mount the two power MOSFETs ensuring that TR6 (2SK133) is fitted closest to the coil L1. Put in the 6BA bolts to hold the transistors from underneath and secure them using nuts and shakeproof washers. Solder the bolt heads to the track on the PCB. Finally solder the drain and gate pins to the PCB and re-check all component positions, polarisations and solder joints.

Power Supply

The PSU (T1, BP1, C12 and I3

PARTS LIST

Amplifier

Resistors — all 1% 0.0W metal film unless specified

R1	2k2		
R2,3	47k	2 of	(R2,3)
R4,5	3k9	2 of	(R4,5)
R6	1k		
R7	33k		
R8,10,11,13,14	100R	5 of	(R8,10,11,13,14)
R9	12k 1/4W STD		
R12	47R		
R15	4R7 3W 1/4W		
R16	10R 1W carbon		
RV1	10k 5.0M. Preset 1k		

Capacitors

C1	1µF 100V PC elec.		(C1)
C2	47pF ceramic		(C2)
C3	47µF 63V PC elec.		(C3)
C4	6nF poly-carbonate		(C4)
C5,6	27pF ceramic	2 of	(C5,6)
C7	100nF poly-carbonate		(C7)
C8,10	100nF polyester	2 of	(C8,10)
C9,11	100µF 63V PC elec.	2 of	(C9,11)

Semiconductors

D1	1N4001		(D1)
TR1,2	2SA877	2 of	(TR1,2)
TR3,4	2SC2547E	2 of	(TR3,4)
TR5	2SA1085E		(TR5)
TR6	2SK133		(TR6)
TR7	2SJ46		(TR7)

Miscellaneous

L1	Enamelled copper wire 18SWG 1m		(L1)
	T03 mounting kit	2 of	(T03)
	Thermopack		(T03)
	Printed circuit board		(PCB)
	Mounting bracket		(BR1)
	Pin 2141	5 of	(P2141)
	Bolt 6BA 1m	5 of	(B6A)
	Nut 6BA	5 of	(N6A)
	Shakeproof washer 6BA	5 of	(WS6A)

Power supply

C12,13	4700µF 63V Cap elec.	2 of	(C12,13)
BR1	Bridge J02		(BR1)
T1	Transformer 32-0-32V 2A		(T1)
FS1,2	Fuse 2A 20mm	2 of	(FS1,2)
	Fuseholder	2 of	(FH1,2)

Test components

	100R 5W 1/4W	2 of	(R100)
	Fuse 250mA 20mm	2 of	(F250)

Note: A complete kit (KWS1) of all the parts listed under Amplifier is available for just £11.49 inc. VAT and P&P from Maplin Electronic Supplies Ltd. The kit does not include the power supply or test components.

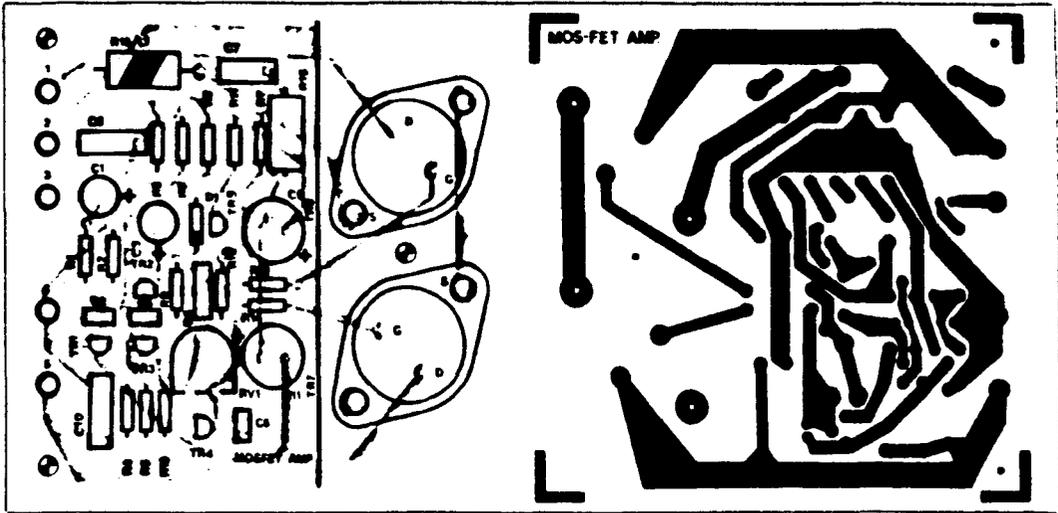


Figure 5 PCB component overlay and track layout

and FS1 and 2) will produce approximately 44.0-44V DC for a stereo pair use a 4A transformer at 32.0-32V instead of a 2A type. Alternatively, toroidal transformers rated at 35.0-35V could be used, 160VA for a single amp and 300VA for a stereo pair. If the transformer voltage is increased to 40.0-40V and TR6 and 7 replaced by a 2SK134 and 2SJ48 respectively output powers in excess of 75W RMS into 8 ohms are possible.

Figure 9 shows how simple it is to parallel the output transistors to achieve even higher powers. Using the higher voltage and transistor types just mentioned power levels in excess of 125W RMS into 4 ohms are possible with a 1V RMS input signal if this circuit is used.

Setting Up

With no speaker connected and fuses not inserted check that the voltage across C12 is approximately 45V ($\pm 5V$) and that the voltage across C13 is the same. Switch off and short C12 and C13 in turn with a resistor (e.g. one of the test resistors). Now connect FS1 and FS2 via 100R 5W resistors, to pins 2 and 5 respectively. Connect 0V to pin 4. Check with a multimeter set to the highest resistance range that there is no connection between the MOSFET cases and the mounting bracket. Turn RV1 fully clockwise.

Insert 250mA fuses for test purposes as FS1 and FS2 and switch on again. If either fuse blows or any component gets excessively hot switch off immediately. If all is well connect a

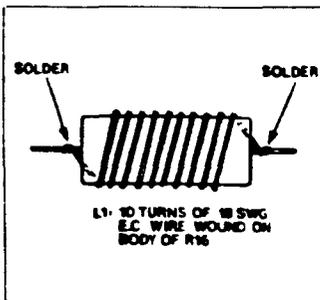


Figure 6 Making the inductor L1

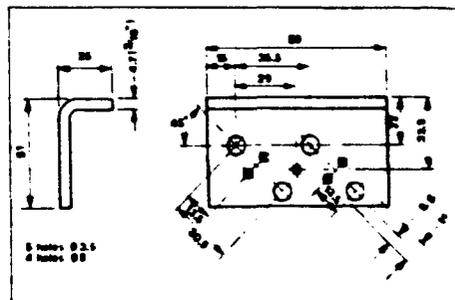


Figure 7 Mounting bracket

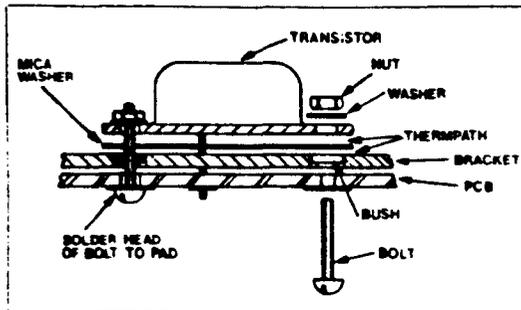


Figure 8 Method of mounting MOSFETs

DC voltmeter between pin 1 and pin 4. The meter should read about 0V (not more than $\pm 100mV$). Switch off and remove the two 100R resistors. Connect FS2 directly to pin 5 and connect a multimeter switched to about 100mA DC between FS1 and pin 2 (+ve lead to fuse and -ve lead to pin 2). Switch on again and rotate RV1 slowly until the meter reads 50mA. Leave for 10 minutes and

re-adjust.

Switch off, disconnect the meter and connect FS1 direct to pin 2. The mounting bracket must now be bolted to a good sized heatsink or a substantial chassis. Finally connect a loudspeaker to pin 1. Note that the speaker negative terminal must be returned to the 0V in the power supply and not to pin 4 of the amplifier. We recommend making the negative

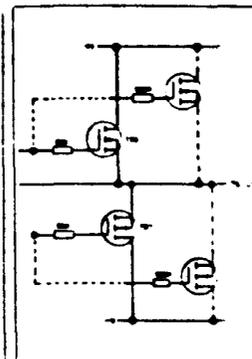


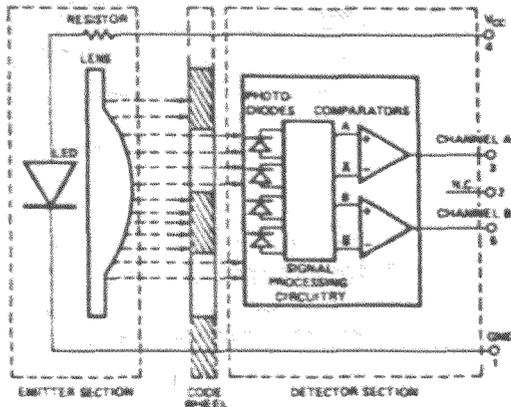
Figure 9 Parallel output connections

tag of C12 the only 0V point with more than one wire attached. Replace the 250mA fuses with 2A types and connect an input between pins 3 and 4. Note that if you use a toroidal transformer you will have to use anti-surge fuses and the test fuses used should be 500mA rating.

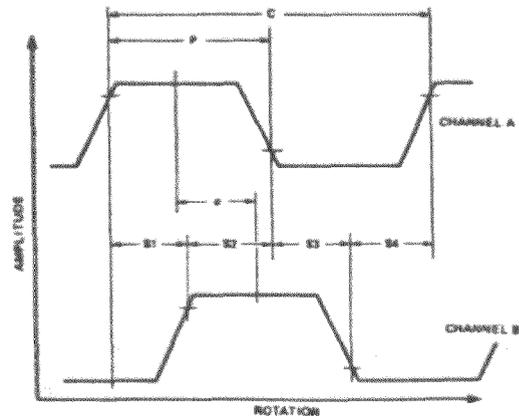
APPENDIX 2

High Resolution Optical Encoder Data

Block Diagram



Output Waveforms



Theory of Operation

The HEDS-9100 is a C-shaped emitter/detector module. Coupled with a codewheel it translates the rotary motion of a shaft into a two-channel digital output.

As seen in the block diagram, the module contains a single Light Emitting Diode (LED) as its light source. The light is collimated into a parallel beam by means of a single lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC consists of multiple sets of photodiodes and the signal processing circuitry necessary to produce the digital waveforms.

The codewheel rotates between the emitter and detector, causing the light beam to be interrupted by the pattern of spaces and bars on the codewheel. The photodiodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the codewheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photodiode outputs are then fed through the signal processing circuitry resulting in A, \bar{A} , B and \bar{B} . Two comparators receive these signals and produce the final outputs for channels A and B. Due to this integrated phasing technique, the digital output of channel A is in quadrature with that of channel B (90 degrees out of phase).

Definitions

Count (N) = The number of bar and window pairs or counts per revolution (CPR) of the codewheel.

1 Shaft Rotation = 360 mechanical degrees
= N cycles

1 cycle (c) = 360 electrical degrees ($^{\circ}$ e)
= 1 bar and window pair

Pulse Width (P): The number of electrical degrees that an output is high during 1 cycle. This value is nominally 180° e or $\frac{1}{2}$ cycle.

Pulse Width Error (ΔP): The deviation, in electrical degrees, of the pulse width from its ideal value of 180° e.

State Width (S): The number of electrical degrees between a transition in the output of channel A and the neighboring transition in the output of channel B. There are 4 states per cycle, each nominally 90° e.

State Width Error (ΔS): The deviation, in electrical degrees, of each state width from its ideal value of 90° e.

Phase (ϕ): The number of electrical degrees between the center of the high state of channel A and the center of the high state of channel B. This value is nominally 90° e for quadrature output.

Phase Error ($\Delta\phi$): The deviation of the phase from its ideal value of 90° e.

Direction of Rotation: When the codewheel rotates in the direction of the arrow on top of the module, channel A will lead channel B. If the codewheel rotates in the opposite direction, channel B will lead channel A.

Optical Radius (R_{op}): The distance from the codewheel's center of rotation to the optical center (O.C.) of the encoder module.

Absolute Maximum Ratings

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Storage Temperature	T_s	-40		100	$^{\circ}$ C	
Operating Temperature	T_o	-40		100	$^{\circ}$ C	
Supply Voltage	V_{cc}	-0.5		7	Volts	
Output Voltage	V_o	-0.5		V_{cc}	Volts	
Output Current per Channel	I_o	-1.0		5	mA	

Recommended Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Temperature	T	-40		100	°C	
Supply Voltage	V _{CC}	4.5		5.5	Volts	Ripple $\le 100\text{mVp-p}$
Load Capacitance	C _L			100	pF	3.3 k Ω pull-up resistor
Count Frequency	f			100	kHZ	Velocity (rpm) x N 60

Note: The module performance is guaranteed to 100 kHz but can operate at higher frequencies.

Encoding Characteristics

Encoding Characteristics over Recommended Operating Range and Recommended Mounting Tolerances. These characteristics do not include codewheel contributions.

Parameter	Symbol	Typ.	Case 1 Max.	Case 2 Max.	Units	Notes
Pulse Width Error	Δp	7	30	40	elec. deg.	
Logic State Width Error	Δs	5	30	40	elec. deg.	
Phase Error	$\Delta \phi$	2	10	15	elec. deg.	

Case 1. Module mounted on tolerances of $\pm 0.13\text{mm}(0.005")$.

Case 2. Module mounted on tolerances of $\pm 0.38\text{mm}(0.015")$.

Electrical Characteristics

Electrical Characteristics over Recommended Operating Range, typical at 25°C

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Supply Current	I _{CC}		17	40	mA	
High Level Output Voltage	V _{OH}	2.4			Volts	I _{OL} = 40 μ A Max.
Low Level Output Voltage	V _{OL}			0.4	Volts	I _{OL} = 3.2 mA
Rise Time	t _r		200		ns	C _L = 25 pF
Fall Time	t _f		50		ns	R _L = 11 k Ω pull-up

Note:

1. For improved performance in noisy environments or high speed applications, a 3.3 k Ω pull-up resistor is recommended.

Recommended Codewheel Characteristics

The HEDS-9100 is designed to operate with the HEDS-5100 series codewheel. See ordering information and specifications at the end of this data sheet.

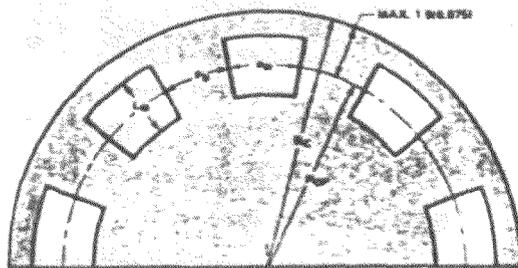


Figure 1. Codewheel Design

Codewheel Options

CPR (N)	OPTICAL RADIUS (mm (inch))
96	11.00 (0.433)
100	11.00 (0.433)
192	11.00 (0.433)
200	11.00 (0.433)
256	11.00 (0.433)
360	11.00 (0.433)
400	11.00 (0.433)
500	11.00 (0.433)
512	11.00 (0.433)

Parameter	Symbol	Min.	Max.	Units	Notes
Window/Bar Ratio	a _w /a _b	0.7	1.4		
Window Length	L _w	1.8(07)	2.3(09)	mm (inch)	
Absolute Maximum Codewheel Radius	R _c		R _{OP} + 1.9(0.075)	mm (inch)	Includes eccentricity error

Mounting Considerations

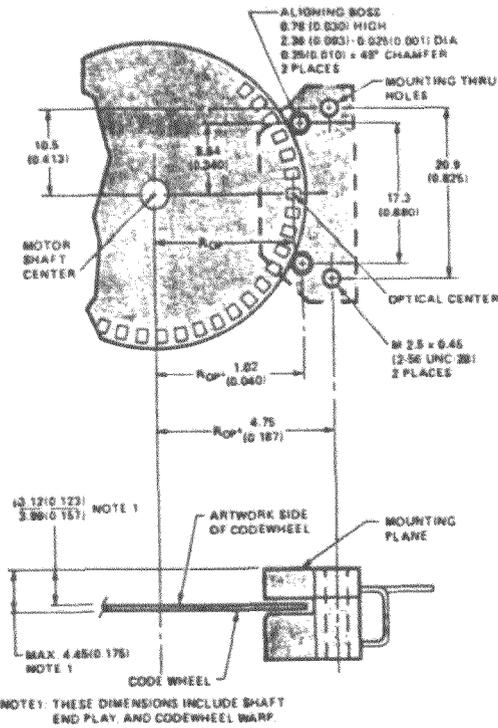


Figure 2. Mounting Plane Side A.

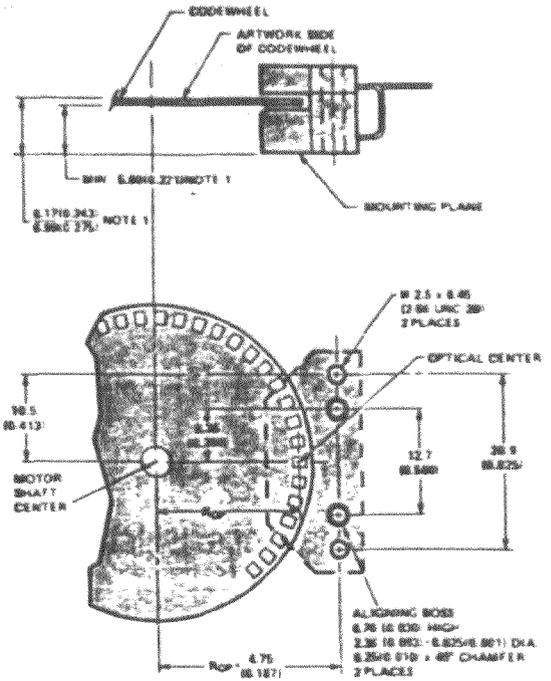


Figure 3. Mounting Plane Side B.

Connectors

Manufacturer	Part Number	Mounting Surface
AMP	103686-4 840442-5	Both Side B
Berg	85036-032 with 4825X-000 terminals	Both
Molex	2695 series with 2759 series terminals	Side B

Figure 4. Connector Specifications

Ordering Information

HEDS-9100 Option D10

HEDS-5100 Option 7

Resolution (Cycles per Revolution)		Shaft Diameter	
K - 96 cpr	G - 360 cpr	01 - 2 mm	11 - 4 mm
C - 100 cpr	H - 400 cpr	02 - 3 mm	14 - 5 mm
D - 192 cpr	A - 500 cpr	03 - 1/8 in.	12 - 6 mm
E - 200 cpr	I - 512 cpr	04 - 5/32 in.	
F - 256 cpr		05 - 3/16 in.	
		06 - 1/4 in.	

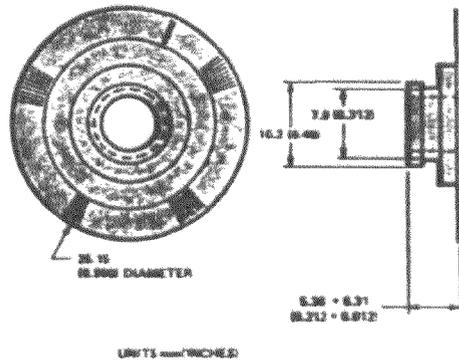


Figure 5. HEDS-5100 Codewheel

APPENDIX 3

74LS2000 Direction Discriminator Data

THCT2000 Incremental Encoder Interface

Data sheet

Available from

Quarndon

Quarndon Electronics (Semiconductors) Ltd
Black Lane, Derby DE3 3ED. Telex: 37163
Tel: (0332) 32851 Fax: (0332) 360922



**TEXAS
INSTRUMENTS**

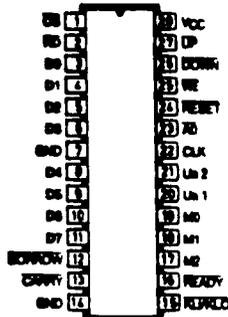
THCT 2000

INCREMENTAL ENCODER INTERFACE

FEATURES

- INTERFACES mechanisms to data bus
- DIRECTION DISCRIMINATOR identifies and measures forward or backward rotation or direction
- PULSE-WIDTH MEASUREMENT
- FREQUENCY MEASUREMENT
- CASCADABLE 16-BIT COUNTER
- TTL COMPATIBLE
- 8-BIT PARALLEL 3-STATE BUS
- SIMPLE WRITE AND READ PROCEDURE

Dual-In-Line Pin-Out



APPLICATIONS

The THCT2000 enables mechanical devices to be interfaced with microprocessors. It may be used in many diverse applications, including robotics, tracker balls (or mouse), lathes or tooling machines, automobiles, conveyor belts and transport mechanisms.

ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE

Parameter	Min	Max	Unit
Supply Voltage, Vcc		7	V
All Input Voltages		6	V
Operating Free-air Temperature Range	0	70	°C
Storage Temperature Range	-65	150	°C

DESCRIPTION

The THCT2000 INCREMENTAL ENCODER INTERFACE can determine the direction and displacement of a mechanical device based on two input signals from transducers in quadrature. In addition, it can measure a pulse width using a known clock rate, or a frequency over a known time interval. It includes a 16-bit counter which may be used separately. The THCT2000 may be cascaded to provide accuracy greater than 16-bits, and is designed for use in many types of microprocessor-based systems.

ARCHITECTURE

There are four major functional blocks in the THCT2000, as shown in Figure 1. The measurement and mode control logic generates up or down count pulses (I1 and I2) from the quadrature signals Ua1 and Ua2, the clock input, and the mode controls (M0, M1 and M2). The control logic provides common microprocessor interface signals. The control logic provides common microprocessor interface signals. The output register latches the value of the counter, and the output multiplexer allows the processor to read either the upper or lower byte in the register or to preload the counter.

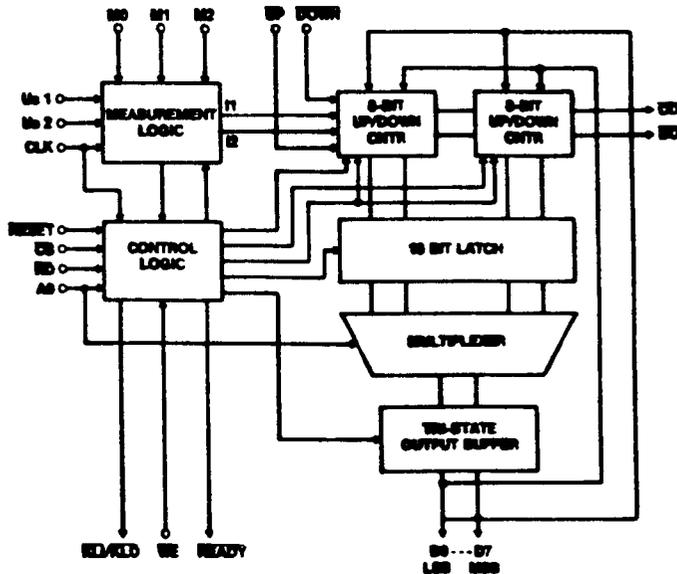


Figure 1

OPERATION

The eight modes of operation of the THCT200 are summarized in Table 1.

Mode	M2	M1	M0	Mode Description
0	0	0	0	Counter 16-bit up/down counter (includes direction discriminator)
1	0	0	1	Direction Discriminator Single count pulse synchronous with U _{h1} rising in forward direction and U _{h1} falling in backward direction.
2	0	1	0	Single count pulse synchronous with U _{h2} rising in forward direction and U _{h2} falling in backward direction.
3	0	1	1	Double count pulse synchronous with U _{h1} rising and falling
4	1	0	0	Double count pulse synchronous with U _{h2} rising and falling
5	1	0	1	Quadruple count pulse synchronous with all edges.
6	1	1	0	Pulse Width Measurement U _{h1} is gate signal U _{h2} is high for up counting and low for down counting. Count is synchronous with rising dock.
7	1	1	1	Frequency Measurement U _{h1} is frequency signal to be measured U _{h2} is gate signal of known time interval. Count is synchronous with rising edge of U _{h1}

Table 1

MODE 0: 16-BIT UP/DOWN COUNTER MODE

In this mode the THCT200 may be used as a four 16-bit up/down counter with cascade capability. This is operated using the UP and DOWN inputs. Setting UP and DOWN low at the same time is prohibited.

The status of the outputs of the counters are transferred to a 16-bit latch. The contents of this 16-bit latch are multiplexed on an 8-bit parallel data bus (D₀...D₇), so strobes to microprocessors using RD and CS.

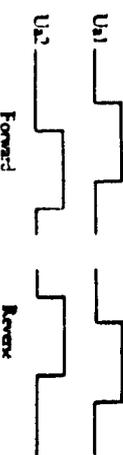
A₀ is the control input for the multiplexer. A high level at this input transfers the least significant byte to the data outputs, and a low level transfers the most significant byte.

To load the up/down counter, the WR, CS, and A₀ signals are used. The load operation should start with the LSB going up or the MSB in the correct sequence. The counter is cleared using the RESET signal.

Cascading to a bus is possible using the inputs UP, DOWN, and the outputs BORROW, CARRY, and RITZDO.

MODES 1-8: DIRECTION DISCRIMINATOR MODES

The quadrature signals U_{h1} and U_{h2} identify forward or backward directions. If U_{h1} leads U_{h2}, the forward direction is indicated and the counter will count up; if U_{h1} lags U_{h2}, the reverse direction is indicated and the counter will count down.



U_{h1} and U_{h2} are both stored in the first of a pair of counter-type flip-flops on the clock falling edge, and transferred to the next on the clock rising edge. By comparing the status of the four flip-flops and detecting the mode inputs, the up or down count pulses are generated at the rising edge of CLK; see Figures 2 and 3.

MODES 1 to 5 define which edge of the quadrature signals will be counted in accordance with Table 1.

The clock frequency should be at least four times greater than the frequencies of the quadrature signals; this will eliminate problems resulting from timing jitter in the transducer signals, and will allow the quadruple counting mode to be used. The frequency of the quadrature signals, U_{h1} and U_{h2}, may be calculated from the relationship:

$$f = \frac{\text{shaft speed}}{\text{resolution of transducer}}$$

DIRECTION DISCRIMINATOR UP CLOCK

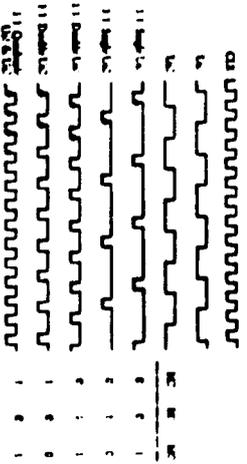


Figure 2: Pulse diagram - Direction Discriminator

DIRECTION DISCRIMINATOR DOWN CLOCK

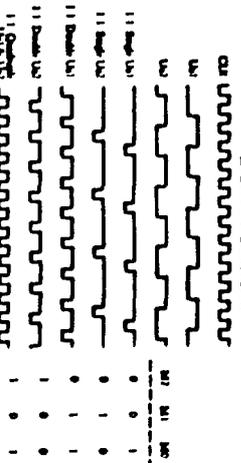


Figure 3: Pulse Diagram - Direction Discriminator

MODE 6: PULSE WIDTH MEASUREMENT MODE

In this mode, U₁ acts as a gate, and is the pulse width to be measured. Synchronized with the clock edge after a low to high transition of U₁, counting begins at the input clock frequency. Similarly, synchronized with the clock edge after a high to low transition of U₁, counting is disabled; the value in the counter is loaded into the output register; KIDPO is pulled low; and then the counter clears. See figure 4. If U₂ is held high, the counter will count up, and if U₂ is held low, the counter will count down.

The counter can be preloaded by activating \overline{CS} , \overline{WE} , and A0; this must be done while U₁ is low. The output register should be read by activating \overline{CS} , \overline{RD} , and A0 after U₁ has fallen and before the next preload takes place.

The KIDPO signal may be used as an interrupt to indicate to the processor when the output register has been loaded. In both the pulse width and frequency modes, the output register will not be loaded via \overline{CS} and \overline{RD} , but by the falling edge of U₁, or by pulling KIDPO low (only if in cascade mode).

PULSE WIDTH MEASUREMENT

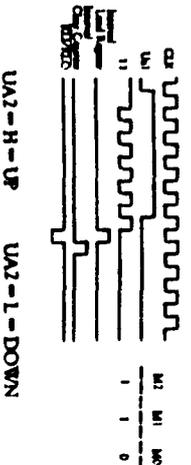


Figure 4: UA2 = H = UP UA2 = L = DOWN

Figure 4

In the pulse width mode, the maximum time that can be measured is:

$$T_{max} = 2(T\phi) \quad (\text{Accuracy is } + \text{ or } - T\phi)$$

MODE 7: FREQUENCY MEASUREMENT MODE

In Mode 7, U₁ is the signal of unknown frequency to be measured; U₂ is a gate signal of known width. A low to high transition of U₂ enables counting at the frequency of U₁. When the gate (U₂) falls, counting is disabled; the value of the counter is loaded into the output register; KIDPO is pulled low; and then the counter is cleared. See Figure 5. The data in the output register may be read at anytime.

FREQUENCY MEASUREMENT

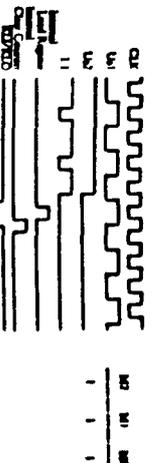


Figure 5

The maximum frequency that can be measured is:

$$F_{max} = PV/2$$

RESET OPERATION

A reset is generated by pulling the RESET pin low. This will clear the counter to zero, reset the D flip-flops at the inputs of the quad register signals (U₁ and U₂), and clear the latches that inhibit the load register pulse. To avoid a spurious count after a reset, the U₁ and U₂ inputs should be held to the values indicated in Table 2 during and just after the reset pulse.

MODE	U ₁	U ₂
0	X	X
1-5	H	H
6-7	L	L

Table 2

CASCADING DEVICES

The KIDPO pins of all cascaded THCT2000's should be tied together, so that all of the devices load their registers at the same time. When the 'master' generates a pulse for either THCT2000A, KIDPO on the 'master' works as an output, and KIDPO on the 'slaves' work as inputs. The CARRY output of one device should be tied to the DP input of the next device in the cascade; similarly, BORROW should be connected to DOWN. See System Application.

PIN DESCRIPTION

Pin name	Pin number	I/O	Description
\overline{CS}	1	Input	Chip Select. A low level at this input enables this device.
\overline{RD}	2	Input	Read. When this and \overline{CS} are active (low), the data from the output register will be present on the data bus.

Pin name	Pin number	I/O	Description
DDO7	3, 4, 5, 6, 8, 9, 10, 11	Input/ Output	Data Bus Buffer: 8-bit Bidirectional buffer with tri-state outputs connected to the microprocessor system.
BORROW	12	Output	Counter subflow signal
CARRY	13	Output	Counter overflow signal
RD/WD	15	Input/ Output	Cascade load input/cascade load output. Open collector output requiring always an external pull-up resistor.
READY	16	Output	When active (low) indicates to the processor that it may complete its read or write operation. Ready falling edge is synchronous with clock. Open collector output, requiring external pull-up resistor only if output is used.
M2, M1, M0	17, 18, 19	Input	Mode select inputs (see Table 1)
Ua1, Ua2	20, 21	Input	Measuring input signals
CLK	22	Input	Clock. Used for internal synchronization and control timing. Must be applied in all modes.
A0	23	Input	Byte select. A high level selects the least significant byte. A low level selects the most significant byte.
RESET	24	Input	Device reset. When active (low), the control logic is reset to a known state and the counter is cleared.
WE	25	Input	Write enable. When this and CS are active (low), the data that is on the data bus is loaded into the counter.
DOWN	26	Input	Cascade input for counting down
UP	27	Input	Cascade input for counting up
Vcc	28		Power supply voltage: +/- 10%
GND	2, 14		Ground

READ OPERATION

When in MODES 0 to 5 the contents of the counter can be read at any time by pulling CS and RD low. The most significant byte may be selected by setting A0 low, and the least significant byte may be read by setting A0 high. This will cause a load output pulse to be generated and RD/WD will go low during the next low clock pulse. READY will also go low as the clock goes low, and will stay low until CS and/or RD go high. The load output register pulse stores the current value of the counter in a 16-bit latch; register and A0 directs the proper byte through a multiplexer to the outputs. CS and RD also enable the 3-state outputs. See Figure 6.

The output registers will be latched immediately if RD/WD is pulled low. Every time CS and RD go low following RD/WD going low, and until a RESET occurs, the load output register is inhibited; thus the value stored in the register can be read.

There are three situations when the load output register pulse will be inhibited:

- 1) After a reset has occurred, the first time CS and RD go low the value of A0 will set a latch within the device. From that time until the next reset, A0 must be that same value in order to trigger the load output register pulse during a read operation. If A0

is the opposite of the value stored in the latch, the load pulse will be inhibited, and a pulse will not be generated until a new read cycle is initiated with A0 properly applied. Holding A0 at the inhibiting value will also keep RD/WD from loading the output register. Thus, the high and low bytes should always be read in the same order. This is to control unwanted load output register pulses in a cascaded operation.

- 2) Every time CS and RD go low after RD/WD has been pulled low externally, and until a reset occurs, the load output register pulse is inhibited to allow the data loaded during the RD/WD pulse to be read.
- 3) The load output register pulse via CS and RD will be disabled during the pulse width and frequency measurement modes.

WRITE OPERATION

A number may be preloaded into the counter by pulling CS and WE low while using A0 to direct the value on the data bus to the selected byte of the counter. This will cause READY to go low on the next falling clock edge, and remain low until CS and WE go high. See Figure 7.

THCT2000
ELECTRICAL CHARACTERISTICS OVER RECOMMENDED OPERATING FREE-AIR
TEMPERATURE RANGE (UNLESS OTHERWISE NOTED)

SYMBOL	DESCRIPTION	TEST CONDITIONS	MIN	TYP	MAX		
V_{CC}	SUPPLY VOLTAGE		4.5	5	5.5		
V_{IH}	HIGH LEVEL INPUT VOLTAGE	$V_{CC} = 4.5 - 5.5 \text{ V}$ at all temps	2		V_{CC}	V	
V_{IL}	LOW LEVEL INPUT VOLTAGE	$V_{CC} = 4.5 - 5.5 \text{ V}$ at all temps	0.0		0.8	V	
V_{IK}	INPUT CLAMP VOLTAGE	$V_{CC} = \text{MAX}$, $I_{IK} = -20 \text{ mA}$ @ all temps	-0.1	-0.9	-1.5	V	
V_{OH}	HIGH LEVEL VOLTAGE	$V_{CC} = \text{MIN}$, $V_{IL} = 0 \text{ V}$, $V_{IH} = V_{CC}$	$I_{OH} = -20 \mu\text{A}$ from V_{CC} all temps	V_{CC}			V
			$I_{OH} = -4 \text{ mA}$ 25°C	3.86	4.3		V
			$I_{OH} = -4 \text{ mA}$ @ Temp	3.76	4.1		V
V_{OL}	LOW LEVEL OUTPUT VOLTAGE	$V_{CC} = \text{MIN}$, $V_{IL} = 0 \text{ V}$, $V_{IH} = V_{CC}$	$I_{OL} = 20 \mu\text{A}$ all temps		10	100	mV
			$I_{OL} = 4 \text{ mA}$ 25°C		180	320	
			$I_{OL} = 4 \text{ mA}$ 0/70°C -40/85°C		210	400	
I_{OZ}/I_1	STATE (HIGH IMPEDANCE STATE) OUTPUT CURRENT D0-D7†	$V_{CC} = \text{MAX}$ $V_{OZ} = -V_{CC} \text{ MAX}$ $V_{IL} = 0 \text{ V}$ $V_{IH} = V_{CC}$	25 °C		10	±600	mA
			@ Temp		0.1	±6.0	µA
I_{OFF}	OPEN DRAIN LEAKAGE CURRENT	$V_{CC} = \text{MAX}$ $V_O = 0 \text{ V} - V_{CC}$ $V_{IL} = 0 \text{ V} - V_{CC}$	25 °C		10	±500	mA
			@ Temp		0.10	±100	µA
I_L	INPUT LEAKAGE CURRENT (except D0-D7)	$V_{CC} = \text{MAX}$ $V_I = 0 \text{ V} - V_{CC}$	25 °C		10	±100	mA
			@ Temp		0.05	1.0	µA
I_{CC}	SUPPLY CURRENT	$V_{CC} = \text{MAX}$ D0-D7 INPUT MODE OUTPUTS OPEN	Stand by		0.5	100	µA
			AVG @ 3MHz		5		mA

+ All typical values are at $V_{CC} = 5 \text{ V}$, $T_a = 25^\circ\text{C}$

† I_{OZ} on bidirectional pins includes input leakage for input buffers as well as high impedance leakage current.

THCT2000
TIMING REQUIREMENTS OVER RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
t_{C1}	CLOCK Cycle time, duty 50%	150			ns
t_{C2}	Pulse width: low CLOCK	75			ns
t_{RES}	Pulse width: RESET input low	100			ns
t_{MUD}	Maximum frequency, UP or DOWN, input duty 50%	3	3.5		MHz
t_{MUD}	Pulse width: UP or DOWN input low	50			ns
t_{W1}	Pulse width: RD/KL0 input low	50			ns
t_{WRD1}	Pulse width: RD input low (MODE = 6 & 7)	80			ns
t_{WRD2}	Pulse width: RD input low (MODE = 0 to 5)		t_{C1}		ns
t_{DR}	Time between two read cycles (LSB and MSB)	0			ns
t_{WWE}	Pulse width: WE input low	50			ns
t_{DR}	Time between two read cycles (LSB and MSB)	0			ns
t_{SE}	Set up time: DATA prior to *WE	50			ns

TMCT2000
TIMING REQUIREMENTS OVER BROADERED OPERATING CONDITIONS

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS
QUS	Set up time, CS and RD low before CLK falling edge	30			ns
QUA	Set up time, A0 prior to WE and CS low	30			ns
QUC	Set up time, DP or DQEN, rising edge before CLK falling edge	40			ns
QUJ	Set up time, Ua1 or Ua2 prior to CLK falling edge	35			ns
QUA	Set up time, DATA prior to WE	hold			ns
QUJ	Set up time, Ua2 enable before CLK falling edge	35			ns
QUK	Set up time, Ua1 or Ua2 rising edge before CLK falling edge	35			ns
QUJ	Set up time, A0 enable before CS and RD low after reset	10			ns
QUK	Set up time, Ua1 or Ua2 falling edge	35			ns
QU	Set up time, BIST high prior to CLK falling edge	25			ns
QUK	Set up time, DP or DQEN, rising edge prior to K1/K1D0 (output) falling edge	80			ns
QCR	Hold time, DATA after WE	10			ns
QCV	Pulse width, Ua1 setup high (MODE = 0)		min. 2 t _{IC1}		ns
QCV	Pulse width, Ua2 setup high (MODE = 7)		min. 2 t _{IC1}		ns
QCV	Pulse width, Ua1 setup low (MODE = 0)		min. 2 t _{IC1}		ns
QCV	Pulse width, Ua2 setup low (MODE = 7)		min. 1 t _{IC1}		ns
QUA	Address hold time after WE or CE high	10			ns
QUJ	Ua1 or Ua2 hold time after CLK falling edge	20			ns
QCA	DQD7 hold time after A0 change	15			ns
QUA	Ua1 high hold time after CLK falling edge	20			ns
QUA	Ua2 hold time after CLK falling edge	20			ns

TMCT2000
SETTING CHARACTERISTICS V_{cc} = MIN; ALL TEMPERATURES FREQ = 1 MHz;
RL = 1kΩ CL = 35pF (50pF bidirectional pin)

SYMBOL	PARAMETER	TEST CONDITIONS (See appendix A (Test pattern))	MIN	TYP	MAX	UNITS
QCC	Access time, RD and CLK to data output valid (MODE = 0 to 5)	FROM CLK ↓ mode 1 DATA >FD TO X01		100	155	ns
QCC	Access time, RD to data output valid (MODE = 0-5 2nd byte; MODE = 6-7; burst type)	FROM A0 ↓ mode 1 DATA >AA - >A5 FROM RD ↓ mode 6 DATA >FF - >00 >00 - >FF		40	60	ns
QCC	Preparation delay, RD, WE or CS, inactive to READY	FROM CS ↓		20	40	ns
QCC	Preparation delay, CLK ↑ to READY low			40	60	ns

TMCT2000
SWITCHING CHARACTERISTICS, $V_{CC} = \text{MIN}$ ALL TEMPERATURES FREQ = 1 MHz
RL = 1KΩ CL = 30pf (30pf bidirectional pin)

	PARAMETER	TEST CONDITIONS See appendix A (Test pattern)	MIN	TYP*	MAX	UNITS
t_{CLC}	Propagation delay UP or DOWN rising edge to CARRY or BORROW rising edge	FROM UP 1 TO CARRY 1		70	120	ns
t_{CLL}	Propagation delay CLK rising edge to CARRY or BORROW rising edge	FROM CLK 1 TO BORROW 1 FROM CLK 1 TO CARRY 1		120	205	ns
t_{CCL}	Propagation delay CLK falling edge to REPELO falling edge			80	120	ns
t_{CCL}	Propagation delay CLK rising edge to CARRY or BORROW rising edge	FROM CLK 1 TO CARRY 1		90	160	ns
t_C	Enable time RD and CS low to DQD ¹	FROM RD 1 DATA >ZZ ->AA		40	60	ns
t_{OL1}	REPELO low output pulse width			t_{L1}		ns
t_{OL2}	CARRY or BORROW low output pulse width			t_{L2}		ns
t_{OL3}	CARRY or BORROW low output pulse width			t_{L3}		ns
t_{DC}	Release time DATA after RD, CS	FROM CS ² DATA >01 ->ZZ	0	25	45	ns

*Typ values 20°C $V_{CC} = 5V$

READ

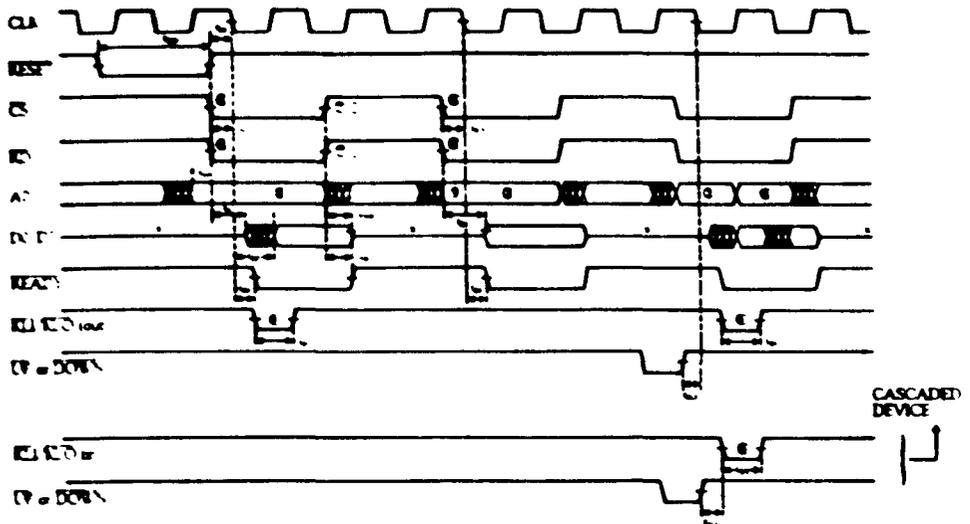


Figure 6

- | | |
|--|---|
| <p>Notes: 1. Either or both may go high</p> <p>2. Source generating output latch load pulse</p> <p>3. Source not generating output latch load impulse</p> <p>4. As an output</p> <p>5. As an input</p> | <p>6. Whichever is later</p> <p>7. Whichever is earlier</p> <p>8. First byte (high or low)</p> <p>9. Second byte (opposite to first byte)</p> <p>10. B should not change while clock is low</p> |
|--|---|

WRITE-CYCLE

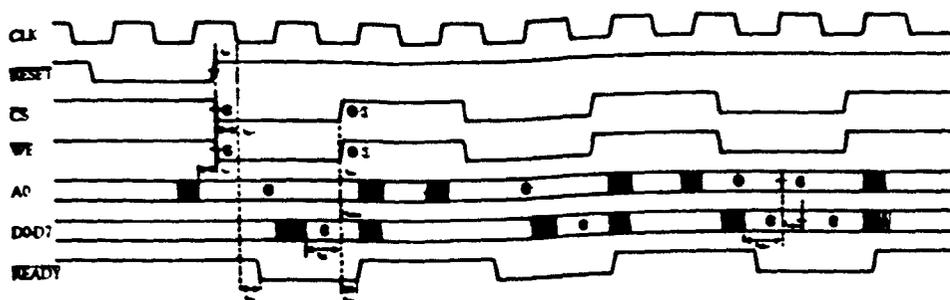


Figure 7

Notes: ① Either or both may go high:

- ② State generating output latch load pulse
- ③ State not generating output latch load pulse
- ④ As an output
- ⑤ As an input

⑥ Whichever is later

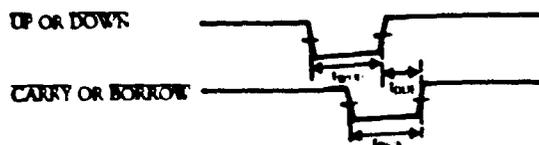
⑦ Whichever is earlier

⑧ First byte (high or low)

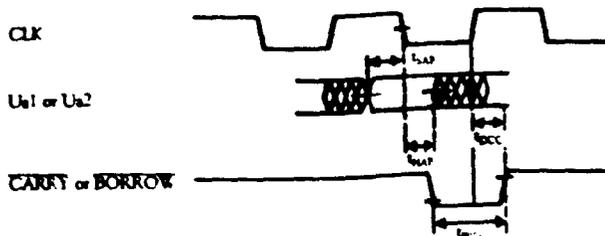
⑨ Second byte (opposite to first byte)

⑩ B should not change while clock is low

ALL MODES



MODE 1-5



MODE 6-7

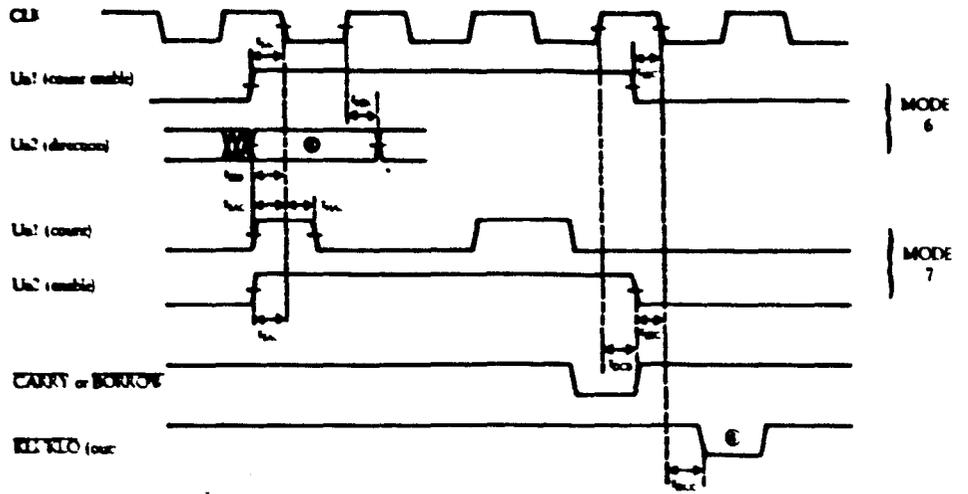
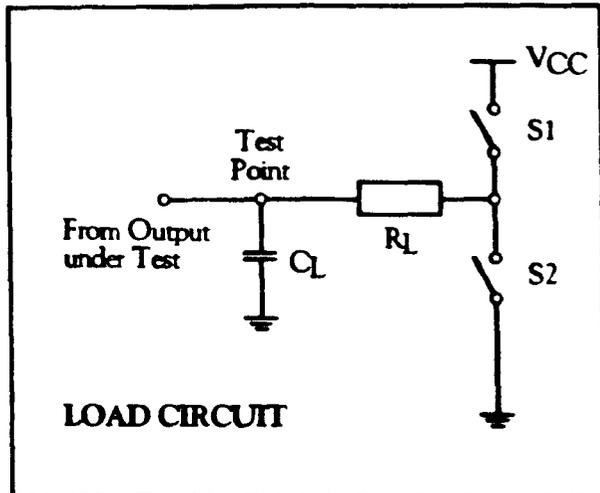
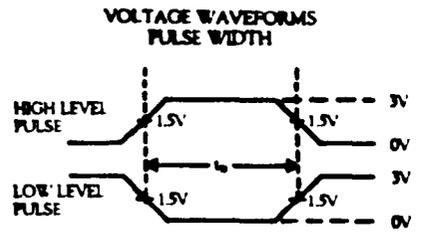
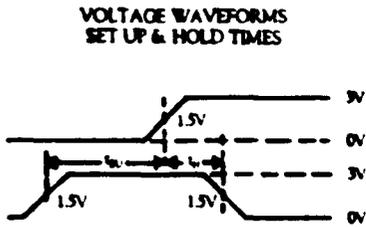


Figure 8

PARAMETER MEASUREMENT INFORMATION

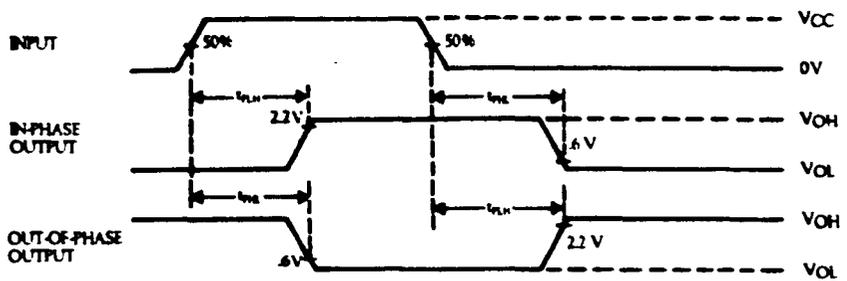


INPUTS

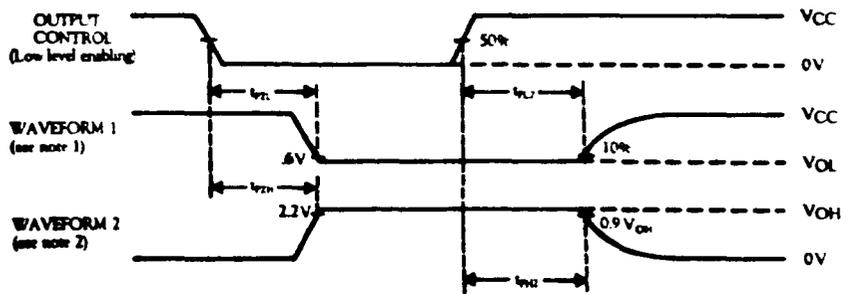


OUTPUTS

VOLTAGE WAVEFORMS PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS ENABLE AND DISABLE TIME, THREE STATE OUTPUTS

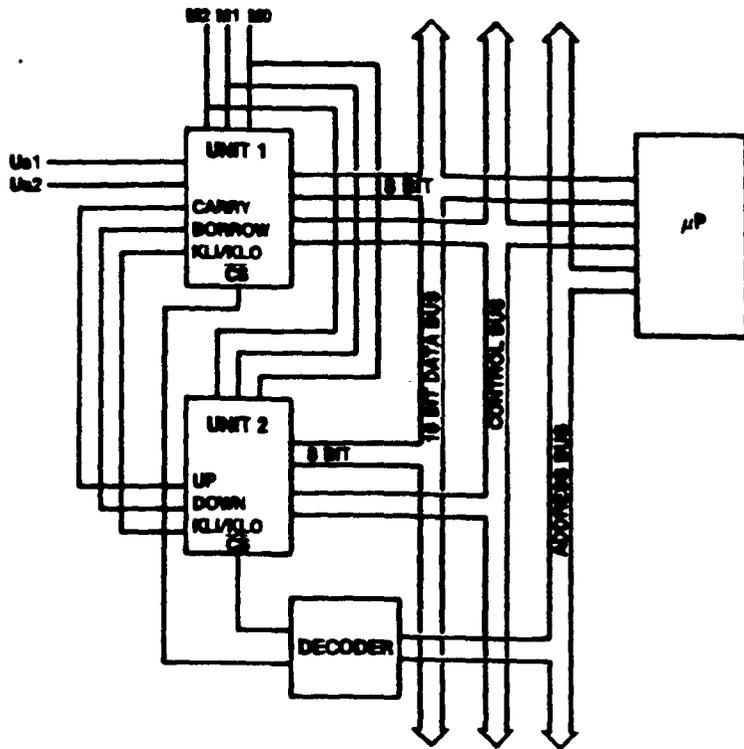


Note 1: Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.

Note 2: Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.

* in the examples above the phase relationships between outputs and inputs have been chosen arbitrarily.

SYSTEM APPLICATION



Appendix 4

Dynamics of a Two Link Planar Manipulator

Following the notation used in figure 4.11, the following procedure may be followed in order to obtain the interaction dynamics of a two link planar manipulator operating in a horizontal plane. The cartesian positions of the two lumped masses are:

$$x_1 = l_1 \cos(\theta_1)$$

$$y_1 = l_1 \sin(\theta_1)$$

$$x_2 = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y_2 = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

The total kinetic energy of this system is given by:

$$k = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2)$$

$$k = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1^2 \cos(\theta_2) \\ + \frac{1}{2} m_2 l_2^2 \dot{\theta}_1 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_2) + m_2 l_2^2 \dot{\theta}_1 \dot{\theta}_2$$

$$\frac{\delta k}{\delta \theta_1} = m_1 l_1^2 \dot{\theta}_1 + m_2 l_1^2 \dot{\theta}_1 + 2m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_2) + m_2 l_2^2 \dot{\theta}_1 \\ + m_2 l_1 l_2 \dot{\theta}_2 \cos(\theta_2) + m_2 l_2^2 \dot{\theta}_2$$

$$\frac{\delta k}{\delta \theta_2} = m_2 l_2^2 \dot{\theta}_1 + m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_2) + m_2 l_2^2 \dot{\theta}_1$$

$$\begin{aligned}
T_1 &= \frac{d}{dt} \left\{ \frac{\delta k}{\delta \dot{\theta}_1} \right\} - \frac{\delta k}{\delta \theta_1} = \ddot{\theta}_1 (l_1^2(m_1+m_2) + 2m_2 l_1 l_2 \cos(\theta_2) + m_2 l_2^2) \\
&\quad + \ddot{\theta}_2 (m_2 l_1 l_2^2 \cos(\theta_2) + m_2 l_2) \\
&\quad - m_2 l_1 l_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) \\
T_2 &= \frac{d}{dt} \left\{ \frac{\delta k}{\delta \dot{\theta}_2} \right\} - \frac{\delta k}{\delta \theta_2} = \ddot{\theta}_1 (m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2)) \\
&\quad + \ddot{\theta}_2 m_2 l_2^2 \\
&\quad + m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2)
\end{aligned}$$

Representing this equation in matrix form, we obtain:

$$\begin{bmatrix} l_1^2(m_1+m_2)+m_2 l_2^2+2m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2+m_2 l_1 l_2 \cos(\theta_2) \\ m_2 l_2^2+m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \\
- \begin{bmatrix} m_2 l_1 l_2 \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_2) \\ - m_2 l_1 l_2 \dot{\theta}_1^2 \sin(\theta_2) \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

APPENDIX 5

Optimal Control Problem used to Test PHAS

A5.1 Manipulator Dynamics

The dynamics of the two degree of freedom planar manipulator considered in section 5.5 may be described by the following equations:

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

$$\dot{x}_2 = \frac{\theta_7(u_1 - u_2 + \theta_6(x_2+x_4)^2 \sin(x_3)) - \theta_6(u_2 - \theta_6 x_2^2 \sin(x_3)) \cos(x_3)}{\theta_7 \theta_5 - \theta_2^2 \cos^2(x_3)}$$

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

$$\dot{x}_4 = \frac{(\theta_5 + \theta_6 \cos(x_3))(u_2 - \theta_6 x_2^2 \sin(x_3)) - (\theta_7 + \theta_6 \cos^2(x_3))(u_1 - u_2 + \theta_6(x_2+x_4)^2 \sin(x_3))}{\theta_7 \theta_5 - \theta_2^2 \cos^2(x_3)}$$

where:

θ_1 is the mass moment of inertia of link 1 with respect to axis 1,

θ_2 is the mass moment of inertia of link 2 with respect to axis 2,

θ_3 is the mass moment of inertia of the end effector with respect to the wrist axis,

$$\theta_4 = \theta_2 + m_3 l_2^2$$

$$\theta_5 = \theta_1 + (m_2 + m_3) l_1^2$$

$$\theta_6 = (r_2 m_2 + l_2 m_3) l_1$$

$$\theta_7 = \theta_3 + \theta_4$$

r_2 is the distance between the centre of gravity of the second link and the second axis

The robot parameters considered were:

$$\begin{aligned}
 \theta_1 &= 1.6 \text{ kgm}^2, & \theta_2 &= 0.43 \text{ kgm}^2, & \theta_3 &= 0.01 \text{ kgm}^2, \\
 l_1 &= 0.4 \text{ m}, & l_2 &= 0.25 \text{ m}, & r_2 &= 0.125 \text{ m}, \\
 m_2 &= 15 \text{ kg}, & m_3 &= 6 \text{ kg}, \\
 M_{\varphi, \max} &= 25 \text{ Nm}, & M_{\psi, \max} &= 9 \text{ Nm}
 \end{aligned}$$

A5.2 Interaction Nonlinearity Representation used in PHAS

In order to represent this system in the PHAS package, the nonlinear elements must be divided up so as to give a number of functions of single variables. It will be noted that the control signals are themselves multiplied by some function of the states before being fed into the integrator block and so rather than have these entering the system from $r(t)$, which is in any case a scalar value, they are generated at the output of the state nonlinearity block. By making states x_5 and x_6 ramp up at unity velocity, effectively equal to time, these may be transformed to give the appropriate switching signals using piecewise linear functions in $N(x)$.

The expressions for \dot{x}_2 and \dot{x}_4 may be split up into functions of x_3 , x_2 , $(x_2 + x_4)$, u_1 , u_2 and $(u_1 - u_2)$. This may be achieved by using the linear pre-combination matrix L given by:

$$L = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Multiplying the state vector by this matrix gives an eight element vector which may be called $l = [l_1, l_2, \dots, l_8]^T$. This is then fed into the nonlinear

feedback matrix, $N_x(l)$ which is defined as:

$$N_x = \begin{bmatrix} n_{11} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ n_{21} & 0 & n_{23} & 0 & 0 & 0 & 0 & 0 \\ n_{31} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ n_{41} & n_{42} & 0 & 0 & 0 & 0 & 0 & 0 \\ n_{51} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ n_{61} & n_{62} & 0 & 0 & 0 & 0 & 0 & 0 \\ n_{71} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ n_{81} & 0 & n_{83} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Where 0 represents a null element, 1 indicates a unity multiplier and n_{ij} represents a scalar function, given by:

$$n_{11} = \frac{\theta_7}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{21} = \frac{\theta_7 \theta_6 \sin(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{23} = l_3^2$$

$$n_{31} = - \frac{\theta_6 \cos(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{41} = \frac{\theta_6^2 \cos(l_1) \sin(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{42} = l_2^2$$

$$n_{51} = \frac{\theta_5 + \theta_6 \cos(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{61} = - \frac{(\theta_5 + \theta_6 \cos(l_1)) \theta_6 \sin(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{62} = l_2^2$$

$$n_{71} = - \frac{\theta_7 + \theta_6 \cos(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{81} = - \frac{(\theta_7 + \theta_6 \cos(l_1)) \theta_6 \sin(l_1)}{\theta_7 \theta_5 - \theta_6^2 \cos^2(l_1)}$$

$$n_{82} = l_2^2$$

The result of the first four rows of N_x are summed and fed back to \dot{x}_2 , while the remaining four constitute \dot{x}_4 . This form is defined by the linear post-combination matrix, M , defined as:

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

The A matrix has the simple form:

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

The B matrix is used to generate unity input to \dot{x}_5 and \dot{x}_6 from a step input signal from $r(t)$, and so has the form:

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

while the other linear matrices, C and D may all be considered as being null.

Using this system description, with the controller switching times defined in the state nonlinearity matrix, $N(x)$, the performance of the robot with the time optimal control may be simulated in the usual way.

In order to generate the data used in the piecewise linear feedback functions, the program shown in listing A5.1 was written, and for the robot parameter values given produced the data shown in table A5.1.

Listing A5.1 Program used to generate piecewise linear feedback data

{ Routine to calculate the values of nonlinear functions used by
PHAS in simulation of optimal controller for 2 link planar manipulator.
See "Time optimal motions of robots in assembly tasks"
by Geering, H.P et al in IEEE AC-31 No.6 p512-518 for details of system}

USES

crt,dos;

CONST

th1 = 1.6; (Inertia of link 1 w.r.t. first axis m²kg)
th2 = 0.43; (Inertia of link 2 w.r.t. second axis m²kg)
th3 = 0.01; (Inertia of hand w.r.t. hand position m²kg)
l1 = 0.4; (Length of link 1 m)
l2 = 0.25; (Length of link 2 m)
r2 = 0.125; (Distance from axis 2 to C of M of link 2 m)
m2 = 15.0; (Mass of link 2 kg)
m3 = 6.0; (Mass of hand and load kg)

VAR

th4,th5,th6,th7 : real; (Derived constants)
fin : real; (Input to functions)
n : ARRAY [0..10,0..100] OF real;
(Output of functions)
firMin,firMax,finStep : real; (Range and increment of
function input)
denominator : real; (Common denominator of all
functions)
StepNumber,NumberOfSteps : integer; (Function input number)

```

nNumber          : integer; ( Number of functions )
OutFile          : string [20]; ( file name for data
                                output )
OFile            : text;
StoreParam       : char;

```

```
BEGIN
```

```
( Calculate derived constants )
```

```
th4 := th2 + 12*12*m3;
```

```
th5 := th1 + 11*11*(m2+m3);
```

```
th6 := 11*(r2*m2 + 12*m3);
```

```
th7 := th3+th4;
```

```
write (' Output file name ');
```

```
readln (OutFile);
```

```
assign (OFile,OutFile);
```

```
rewrite (OFile);
```

```
( Print out parameters )
```

```
writeln;
```

```
writeln ('          Parameter values used ');
```

```
writeln ('          _____ ');
```

```
writeln;
```

```
writeln (' theta1 ',th1:6:4,'   theta2 ',th2:6:4,'   theta3 ',th3:
        6:4);
```

```
writeln (' 11      ',11:6:4 ,'   12      ',12:6:4 ,'   r2      ',r2:
        6:4 );
```

```
writeln (' m2      ',m2:6:4 ,'   m3      ',m3:6:4);
```

```

writeln;

writeln;

write (' Do you require parameters in output file ');

readln (StoreParam);

IF ((StoreParam = 'y') OR (StoreParam = 'Y')) THEN
BEGIN

    writeln (OFile);

    writeln (OFile,'          Parameter values used ');

    writeln (OFile,'          _____ ');

    writeln (OFile);

    writeln (OFile,' theta1 ',th1:6:4,'   theta2 ',th2:6:4,'
                theta3 ',th3:6:4);

    writeln (OFile,' l1      ',l1 :6:4,'   l2      ',l2:6:4 ,'   r2
                ',r2:6:4 );

    writeln (OFile,' m2      ',m2 :6:4,'   m3      ',m3:6:4);

    writeln (OFile);

    writeln (OFile);

END;

write (' Enter minimum input value ');

readln (finMin);

write (' and maximum input value ');

readln (finMax);

write (' Enter number of steps ');

readln (NumberOfSteps);

FOR StepNumber := 0 TO NumberOfSteps DO
BEGIN

    fin := finMin + StepNumber * (finMax-finMin) / NumberOfSteps;

    denominator := th5*th7 - th6*th6*cos(fin)*cos(fin);

    n[1,StepNumber] := th7/denominator;

```

```

n[2,StepNumber] := th7*th6*sin(fin)/denominator;
n[3,StepNumber] := -th6*cos(fin)/denominator;
n[4,StepNumber] := th6*th6*cos(fin)*sin(fin)/denominator;
n[5,StepNumber] := (th5 + th6*cos(fin))/denominator;
n[6,StepNumber] := -(th5 + th6*cos(fin))*th6*sin(fin)
                    /denominator)
n[7,StepNumber] := -(th7 + th6*cos(fin))/denominator;
n[8,StepNumber] := -(th7 + th6*cos(fin))*th6*sin(fin)
                    /denominator;

END;

FOR nNumber := 1 TO 8 DO
BEGIN
    FOR StepNumber := 0 TO NumberOfSteps DO
    BEGIN
        fin := firMin + StepNumber * (finMax-finMin)
                / NumberOfSteps;
        writeln (OFile,fin:6:4,' ',n[nNumber,StepNumber]:6:4);
    END;
    writeln (OFile);
END;

close (OFile);

END.

```

Table A5.1 Piecewise linear feedback data for given robot parameter values

Parameter values used

theta1 1.6000 theta2 0.4300 theta3 0.0100

11 0.4000 12 0.2500 r2 0.1250
m2 15.0000 m3 6.0000

Angle	n ₁₁	n ₁₂	n ₁₃	n ₁₄	n ₁₅	n ₁₆	n ₁₇	n ₁₈
0.00	0.3671	0.0000	-0.6081	0.0000	2.8425	0.0000	-0.9753	0.0000
0.26	0.3480	0.1215	-0.5568	0.1945	2.6748	-0.9341	-0.9049	-0.3160
0.52	0.3047	0.2056	-0.4371	0.2949	2.2912	-1.5459	-0.7418	-0.5005
0.78	0.2603	0.2484	-0.3051	0.2911	1.8895	-1.8030	-0.5654	-0.5395
1.05	0.2273	0.2656	-0.1884	0.2202	1.5716	-1.8368	-0.4157	-0.4858
1.31	0.2079	0.2711	-0.0894	0.1165	1.3548	-1.7663	-0.2973	-0.3876
1.57	0.2016	0.2722	-0.0003	0.0004	1.2273	-1.6568	-0.2019	-0.2725
1.83	0.2078	0.2711	0.0888	-0.1158	1.1761	-1.5341	-0.1190	-0.1553
2.09	0.2271	0.2657	0.1878	-0.2196	1.1945	-1.3973	-0.0394	-0.0460
2.35	0.2601	0.2486	0.3043	-0.2908	1.2787	-1.2221	0.0442	0.0422
2.62	0.3044	0.2059	0.4363	-0.2952	1.4161	-0.9581	0.1319	0.0893
2.88	0.3478	0.1222	0.5563	-0.1954	1.5604	-0.5482	0.2085	0.0732
3.14	0.3671	0.0008	0.6081	-0.0013	1.6262	-0.0035	0.2410	0.0005
3.40	0.3482	-0.1209	0.5574	0.1935	1.5619	0.5422	0.2092	0.0726
3.66	0.3049	-0.2052	0.4379	0.2946	1.4180	0.9540	0.1330	-0.0895
3.92	0.2606	-0.2483	0.3058	0.2914	1.2801	1.2195	0.0452	-0.0431
4.19	0.2274	-0.2656	0.1891	0.2208	1.1951	1.3955	-0.0384	0.0448
4.45	0.2080	-0.2711	0.0899	0.1172	1.1759	1.5325	-0.1181	0.1539
4.71	0.2016	-0.2722	0.0008	0.0011	1.2262	1.6554	-0.2008	0.2711
4.97	0.2078	-0.2711	-0.0882	-0.1151	1.3527	1.7651	-0.2960	0.3863
5.23	0.2270	-0.2658	-0.1871	-0.2191	1.5684	1.8364	0.4141	0.4848
5.49	0.2599	-0.2488	-0.3035	-0.2906	1.8850	1.8045	-0.5634	0.5393
5.76	0.3041	-0.2063	-0.4355	-0.2954	2.2861	1.5509	-0.7396	0.5017
6.02	0.3476	-0.1228	-0.5557	-0.1964	2.6710	0.9439	-0.9033	0.3192
6.28	0.3671	-0.0016	-0.6081	-0.0026	2.8424	0.0128	-0.9753	0.0042

A5.3 SIMNON Representation of Dynamics

This system may be represented in SIMNON using the form given in listing A5.2, where t_1 to t_7 represent the θ values and $s1i$ and $s2i$ the i^{th} controller switching time. Sine and cosine functions are only calculated once to save computation time and increase the readability of the file.

Listing A5.2 SIMNON representation of robot with time optimal controller

CONTINUOUS SYSTEM opti

"

" Simulation to test optimal control of a two link planar manipulator.

" Details as given in Time-Optimal Motions of Robots in Assembly Tasks

" By Geering, H.P. et al in IEEE trans. AC-31 No.6

"

STATE tme x1 x2 x3 x4 " theta1, thetaldot, theta2, theta2dot

DER dtme dx1 dx2 dx3 dx4

" Plant Constants

t1 : 1.6 " Inertia of link 1 w.r.t. joint 1

t2 : 0.43 " Inertia of link 2 w.r.t. joint 2

t3 : 0.01 " Inertia of hand and load w.r.t. hand

l1 : 0.4 " Link 1 length

l2 : 0.25 " Link 2 length

r2 : 0.125 " Distance of C of G of link 2 from joint 2

m2 : 15 " Mass of link 2

m3 : 6 " Mass of hand and load

mt : 25 " Max torque at joint 1

mp : 9 " Max torque at joint 2

t4 = t2 + l2*l2*m3 "Other plant constants

t5 = t1 + l1*l1*(m2+m3)

t6 = l1 * (r2*m2+l2*m3)

t7 = t3 + t4

" Differential equations

c3=cos(x3)

s3=sin(x3)

" Common elements in differential equations

dp1 = t7*(u1-u2+t6*(x2+x4)*(x2+x4)*s3)-t6*(u2-t6*x2*x2*s3)*c3

dp2 = (t5+t6*c3)*(u2-t6*x2*x2*s3)

dp3 = dp2-(t7+t6*c3)*(u1-u2+t6*(x2+x4)*(x2+x4)*s3)

dtme = 1

dx1 = x2

dx2 = dp1/(t7*t5-t6*t6*c3*c3)

dx3 = x4

dx4= dp3/(t7*t5-t6*t6*c3*c3)

" Input switching times

s11 : 0.626 " Input 1 switching times

s12 : 1.252

s13 : 1.252

s21 : 0.278 " Input 2 switching times

s22 : 0.626

s23 : 0.974

sf : 1.252 " Stop time

" Controller definition

" u1 = (+,-,+,-) mt switching at s11,s12,s13

" u2 = (-,+,-,+) mp switching at s21,s22,s23

u11 - if tme<s11 then mt else if tme<s12 then -mt else mt

u1 - if tme<s13 then u11 else if tme<sf then -mt else 0

u21 - if tme<s21 then -mp else if tme<s22 then mp else -mp

u2 - if tme<s23 then u21 else if tme<sf then mp else 0

END

vin : 9.77

```
dvf = -vf/ti + ki*vin/ti      " Input filter dynamics
vdi = 0.05*vin - im          " Input to voltage amplifier

dvds = -vds/tid + kv*vdi/tid  " Vd=(Kv Vdi/tid)/(S + 1/tid) Drive
                                voltage
vd = IF vds<-vdsat THEN -vdsat ELSE IF vds>vdsat THEN vdsat ELSEvds
                                " Amplifier saturation
vb = ke * vm                 " Vb = Ke Vm   Motor back e.m.f
dim = -r*im/l + (vd-vb)/l    " Im = ( (Vd - Vb)/L ) / (S + R/L)
                                Motor current
kf = - km * FUNC (1,vm*180/(21.7*3.142)) " Friction torque - read
                                    from file
tm = km * im                 " Tm = Km Im   Motor torque
dvm = (tm - kf - rtorq )/j   " vm = ((Tm - f(vm) - Tl)/J)/S
                                Motor velocity

dpm = vm
posm = pm                    " Motor output position
vel = vm
curr = im
volt = vd
```

END

Link Dynamics

CONTINUOUS SYSTEM elbow

" Model of elbow joint of RTX, with yaw fixed.

```

STATE armpos armvel armtor
DER darmpos darmvel darmtor
INPUT motorpos motorvel
OUTPUT reftor          "torque reflected back to motor

ks1 : 0.008  " Elastic spring constant of drive belts
ks2 : 0.012  " Anelastic spring constant of drive belts
tb  : 1.27   " Time constant of drive belt
nb  : 21.666 " Reduction ratio of belts
jarm : 0.0994 " Inertia of forearm, wrist and gripper
km  : 0.489  " Motor torque constant - same as in edrive

tdtor1 = (1/(ks2*tb))*( - (ks1+ks2)*armtor)
tdtor  = tdtor1 + (1/(ks2*tb))*((motorpos/nb-armpos)
          + (tb*motorvel/nb-armvel))
          " Belt dynamics
darmtor = tdtor          " Torque at joint
reftor  = armtor / nb    " Reflected torque
darmvel = armtor/jarm - km*nb*FUNC(2,armvel*180/3.142))/jarm
          " Joint acceleration =
          (generated - friction torque)
          / inertia
darmpos = armvel        " joint position

END

```

Controller Definition

```
DISCRETE SYSTEM vsc
```

" Discrete time simple Variable structure controller

INPUT vel pos

OUTPUT dem

TIME t

TSAMP ts

" CONTROL PARAMETERS

c : 5 " Switching line gradient

g : 1 " Controller gain

dt: 0.005 " Sample Interval

" DEMAND PARAMETERS

pdem : 20

vdem : 0

" CONTROLLER EQUATIONS

e = pdem - pos

edot = vdem - vel

s = c * e + edot

demr = IF s*e > 0 THEN g*e ELSE -g*e

dem = IF demr>10 THEN 10 ELSE IF demr<-10 THEN -10 ELSE demr

ts = t+dt

END

APPENDIX 7

Additional Results for the Implementation of VS Control

In addition to the results given in Chapter 6, there are a number of results which, although not critical to an understanding of the control system implemented, do give useful information. Such results will be given in this appendix.

A7.1 Effect of Load on VSC with Piecewise Linear Switching Function, Compensation and Integral Action

It was noted in section 6.4.3 that good switching function following could be obtained for the nominal system while the results given in figures A7.1 and A7.2 show that addition of a 1kg load to the end effector does not have a significant effect on this behaviour in terms of either the phase plane or velocity profile.

Figure A7.1 Effect of load on phase trajectory

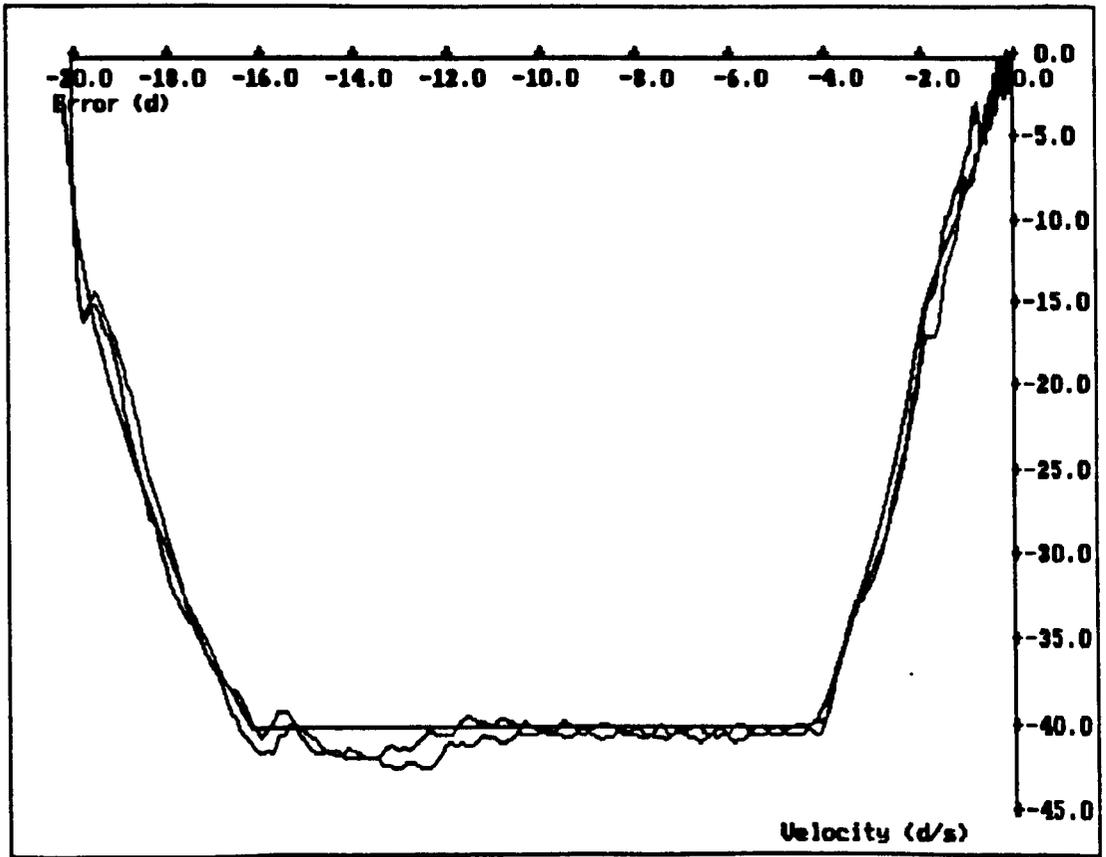
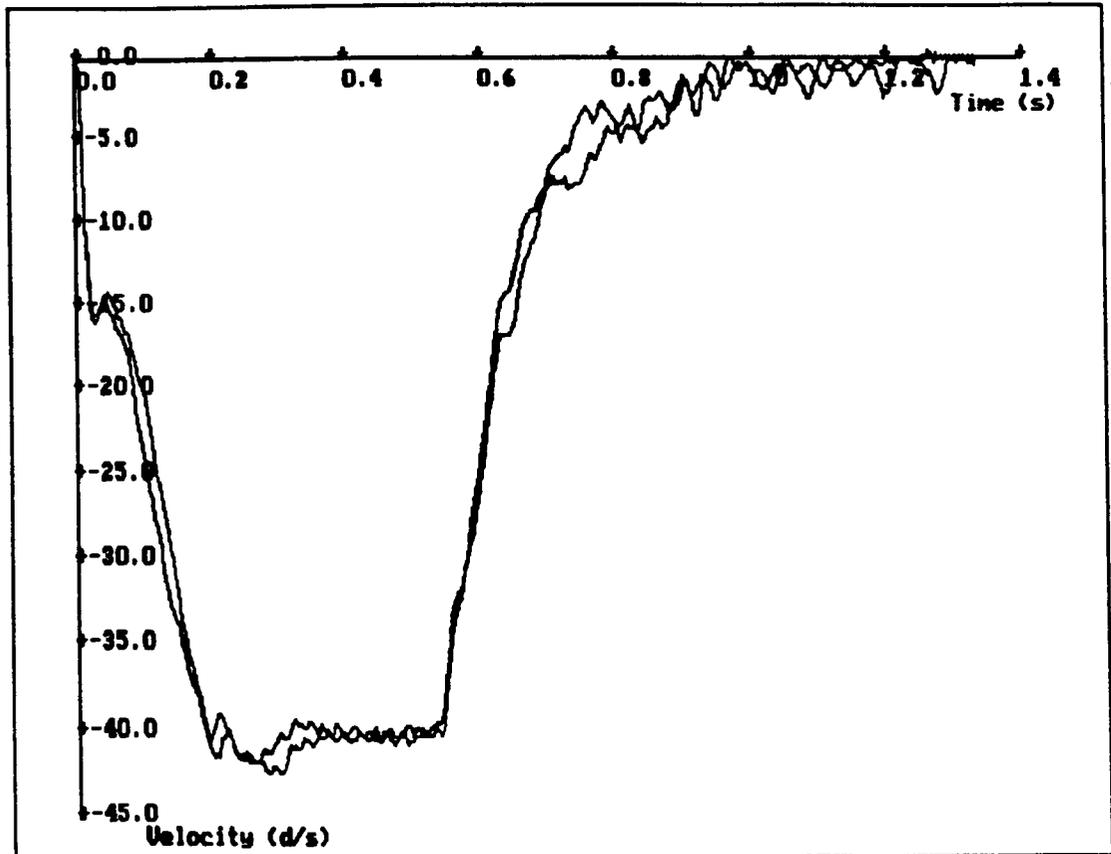


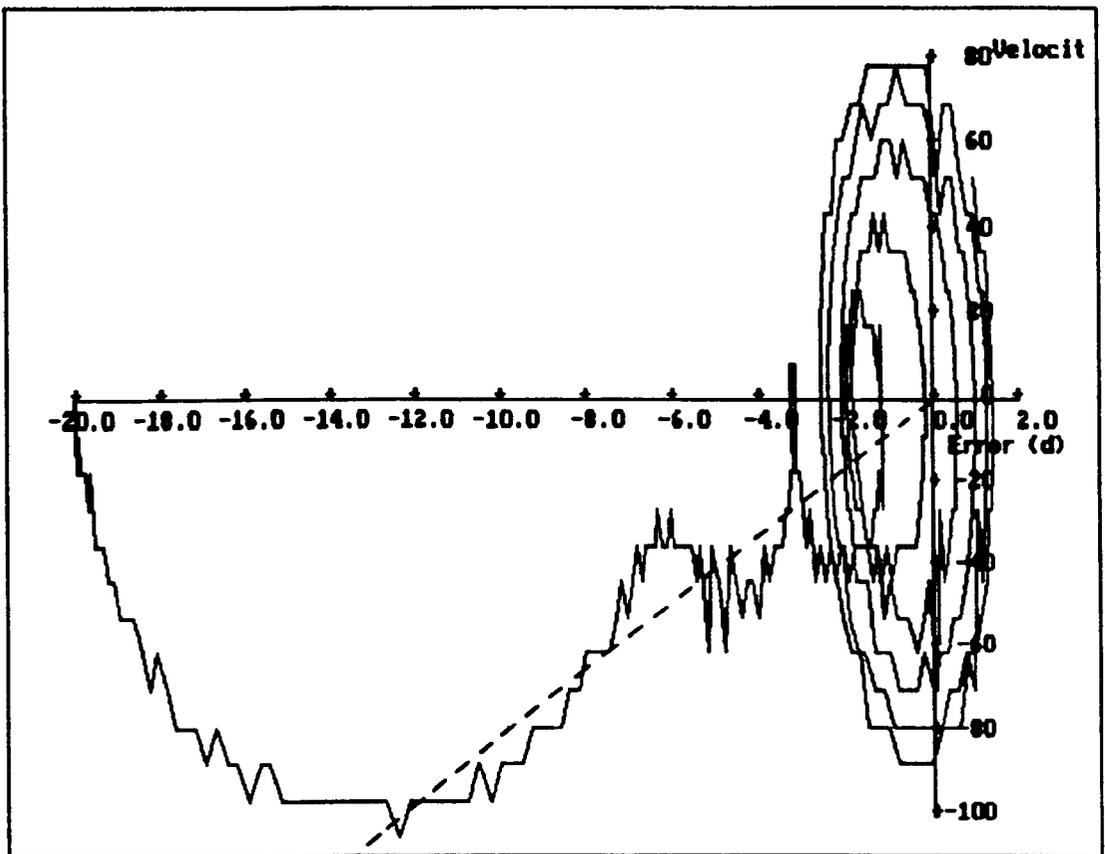
Figure A7.2 Effect of load on velocity profile



A7.2 High Gain VSC with Boundary Layer Using Joint Sensor

It was noted in section 6.5.1 that, when using a simple VSC to a system deriving position and velocity information from the joint mounted sensor, that a high gain resulted in undesirable behaviour in the form of oscillation around the origin. In the same way, when using a VSC with boundary layer, the use of a high gain leads to limit cycling around the origin as seen in figure A7.3

Figure A7.3 High gain VSC with boundary layer showing limit cycle behaviour



A7.3 VSC with Friction and Acceleration Compensation and Integral Action

The use of compensation and integral action within a VSC with boundary layer was seen, in chapter 6, to give significant benefits when compared with the basic VSC algorithm. Two sets of results concerning this type of controller are given in the following sections.

A7.3.1 Effect of Compensation Terms on VSC Using Joint Position

As an illustration of the effect of adding compensation terms to a VSC using joint position and velocity, as in section 6.5.3, figure A7.4 shows the effect of friction compensation while A7.5 shows the effect of friction and acceleration compensation.

Figure A7.4 VSC with boundary layer and friction compensation

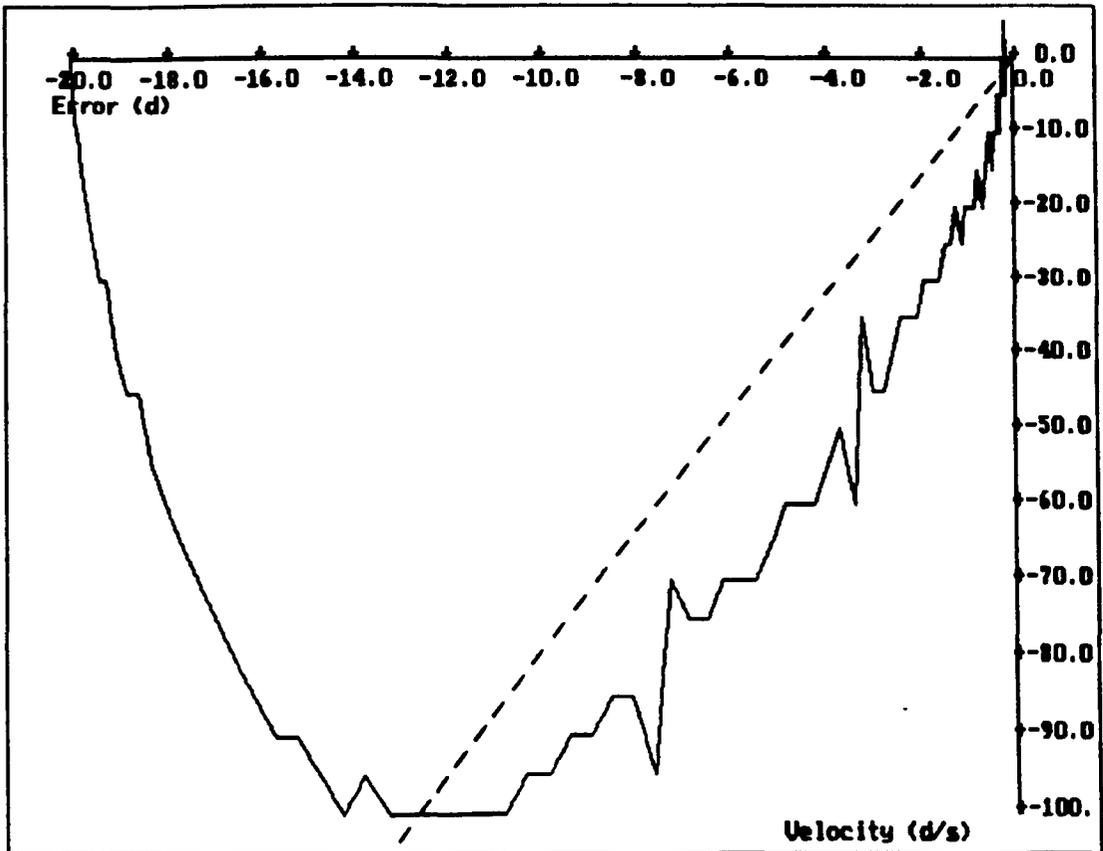
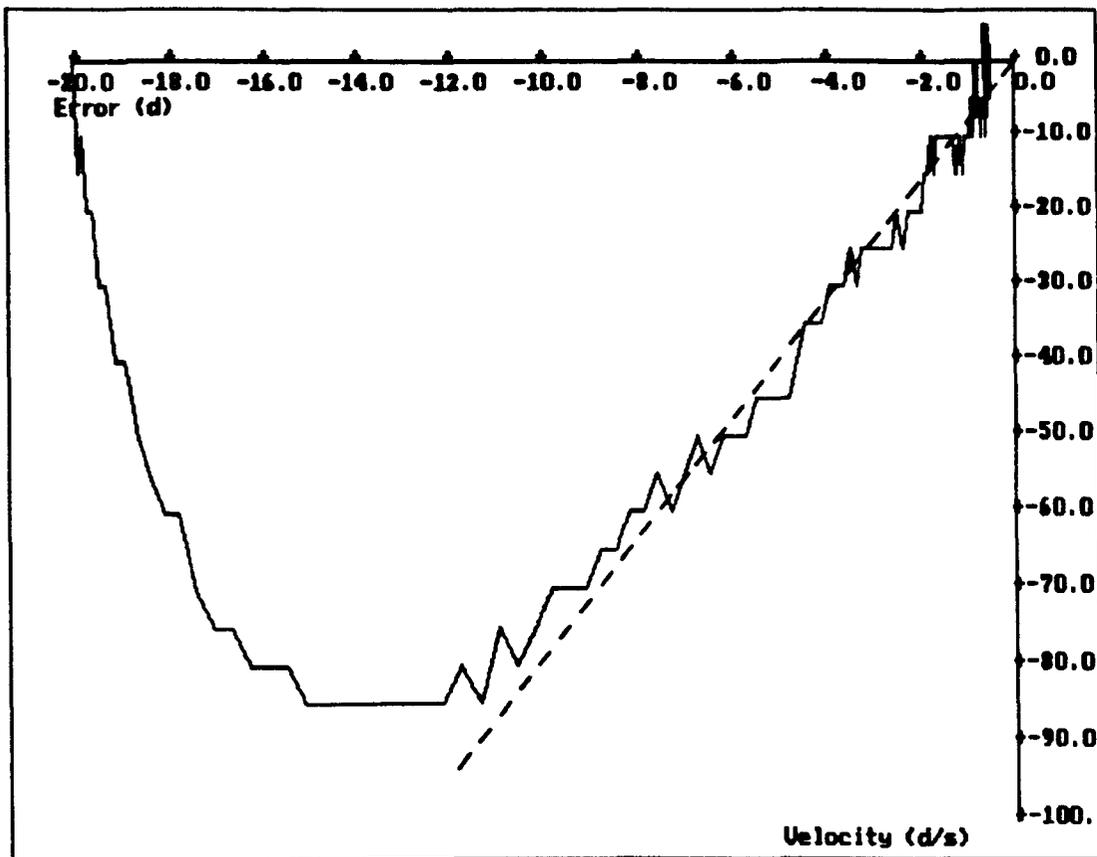


Figure A7.5 VSC with friction and acceleration compensation



A7.3.2 Speed of Response Compared with PID Control

It was seen in section 6.7.2 that the speed of response of a VSC and a PID control, both tuned for the maximum speed, was very similar. The result shown in figure A7.6 indicates that the actuation signal used in each case is similar to the form expected for an optimal control scheme, i.e. saturated for the majority of the transient with only a single change of state.

The effect of using a piecewise linear switching function on the speed optimised VSC is shown in figure A7.7

Figure A7.6 Control signals for speed optimised VSC and PID

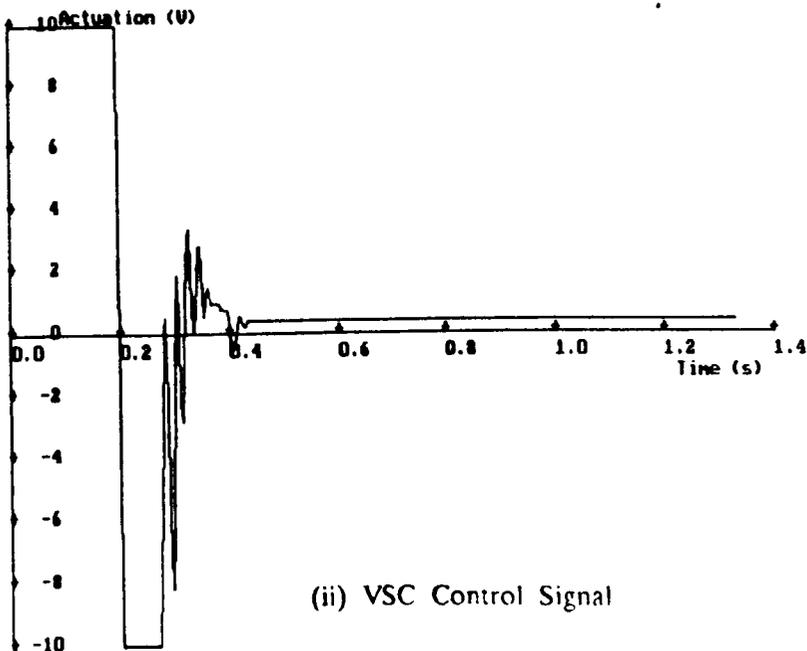
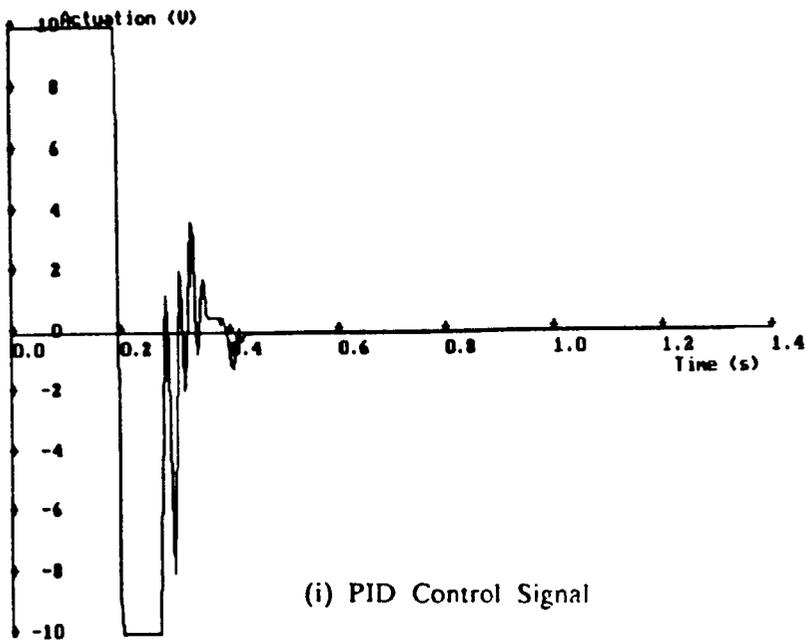
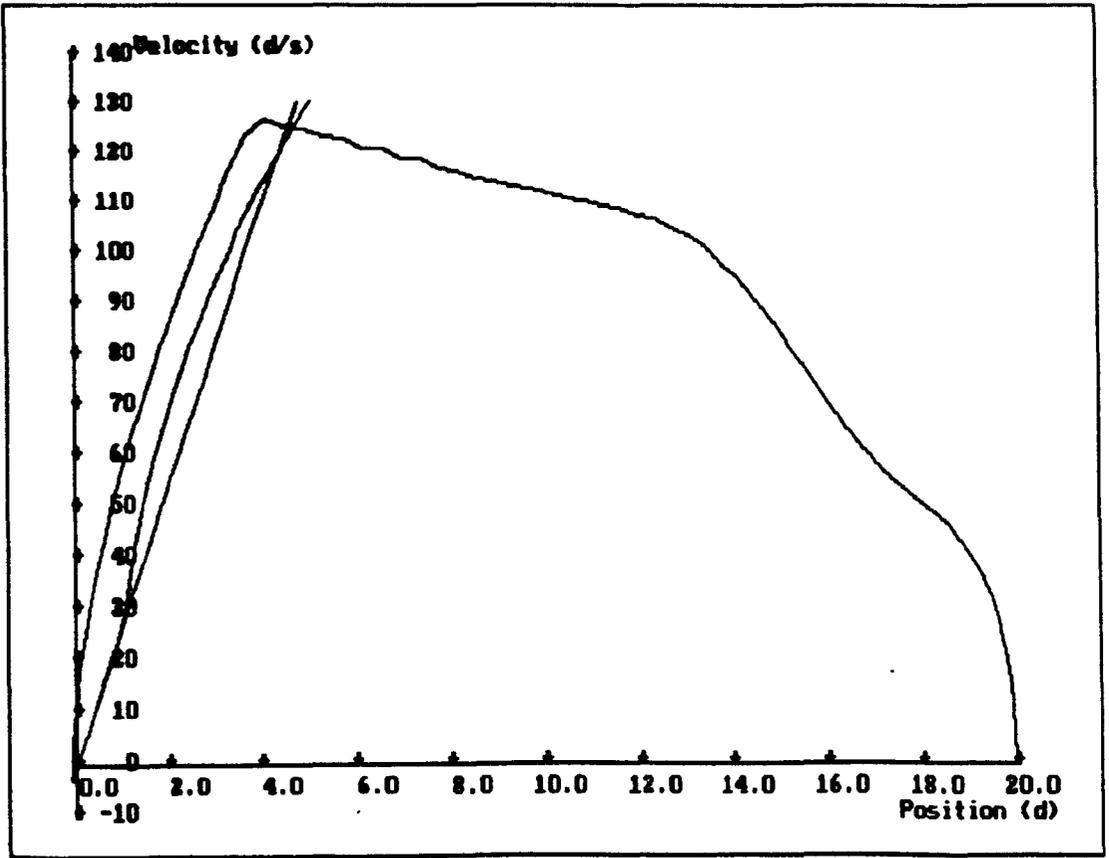


Figure A7.7 Effect of piecewise linear switching curve on speed optimised VSC



A7.4 Effect of Velocity on End Point Accuracy for Various Controllers

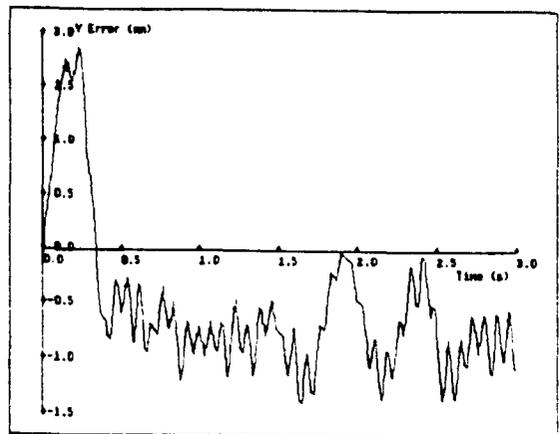
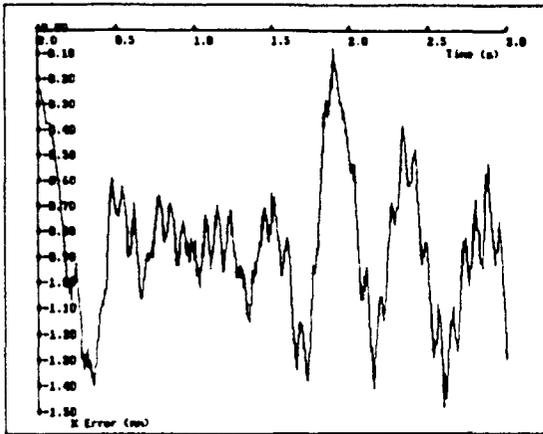
In section 6.7.3.3, the robot was forced to follow a number of trajectories at a number of speeds using different control algorithms. The resulting cartesian errors for the manufacturer's PID control and motor and joint/motor based VSCs are shown in figure A7.8 for velocities of 20, 50 and 100mm/s.

Figure A7.8 Effect of velocity on end point tracking accuracy

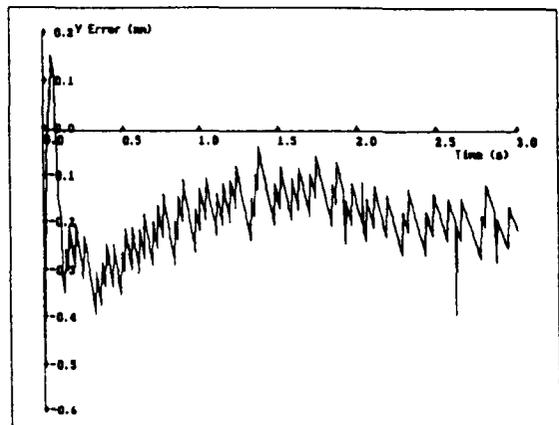
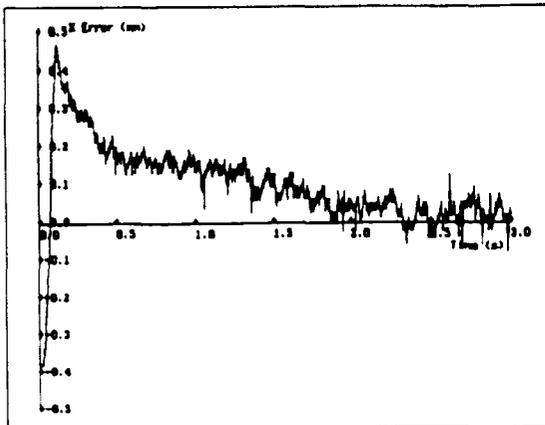
20mm/s

X Error

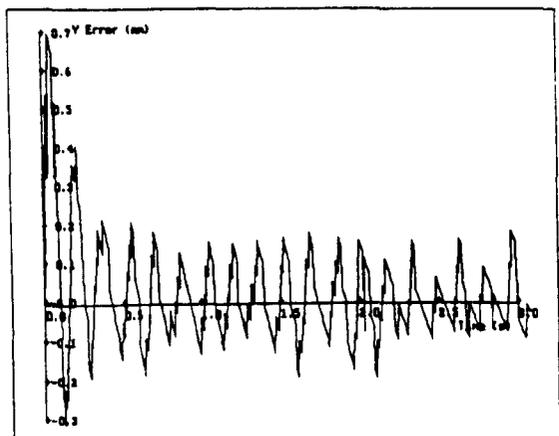
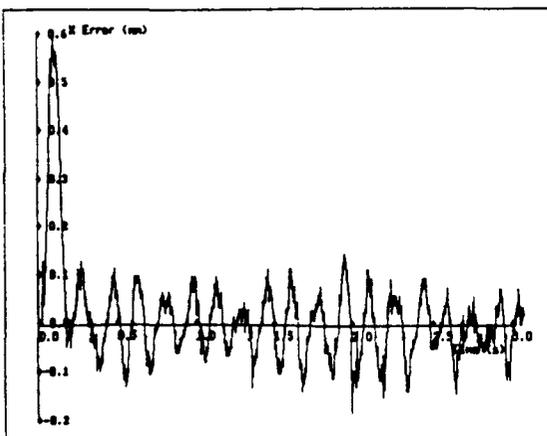
Y Error



Manufacturer's PID Control



Motor Position Based VSC

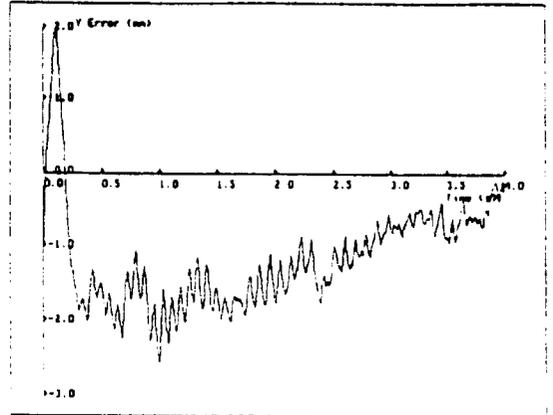
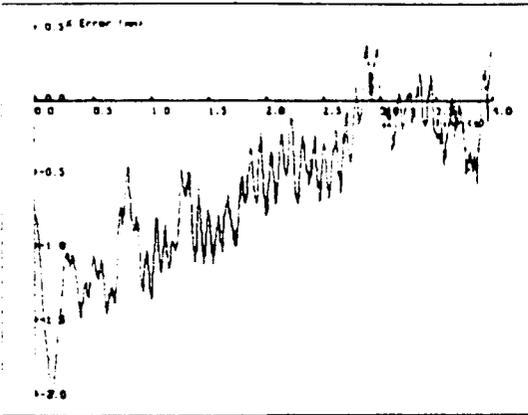


Motor/Joint Based VSC

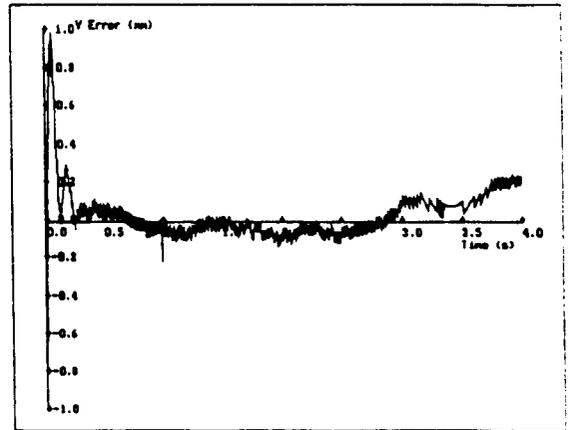
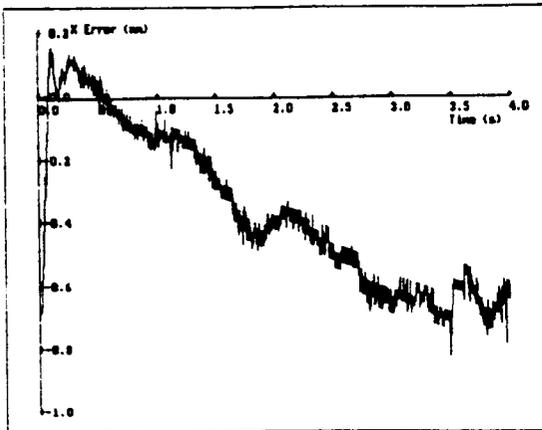
50mm/s

X Error

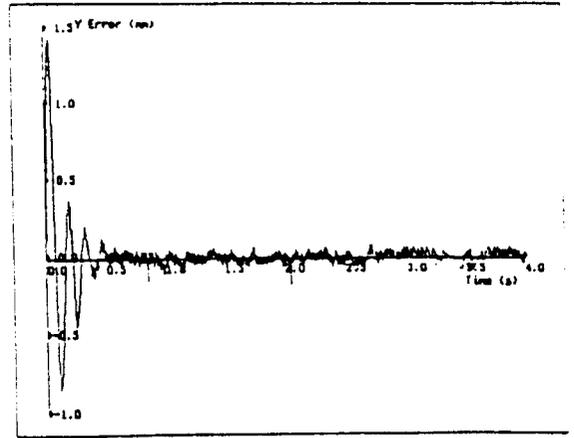
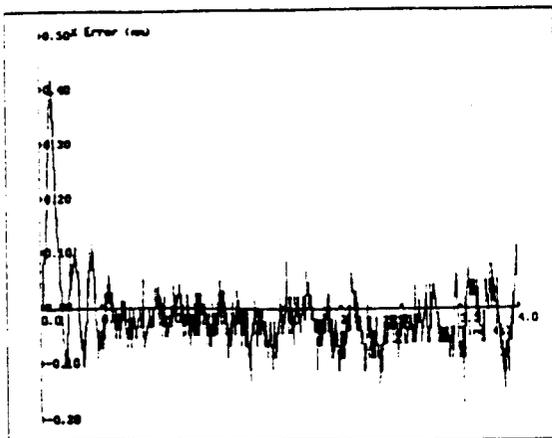
Y Error



Manufacturer's PID Control



Motor Position Based VSC

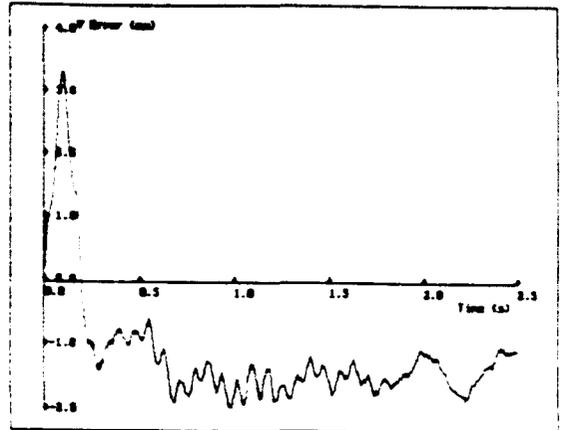
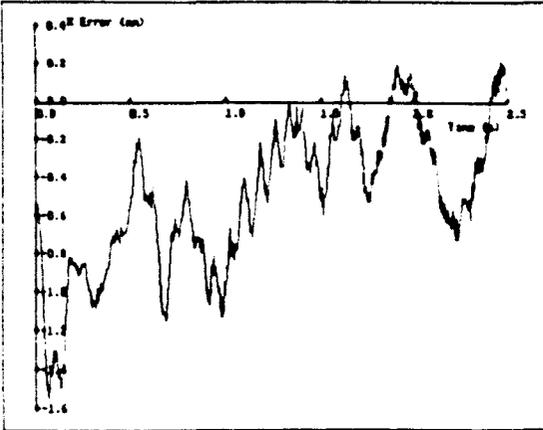


Motor/Joint Based VSC

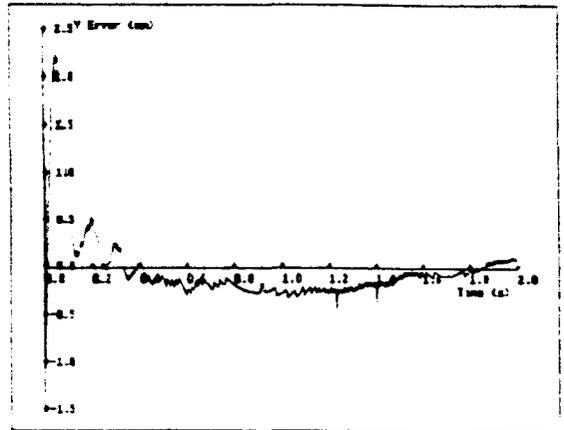
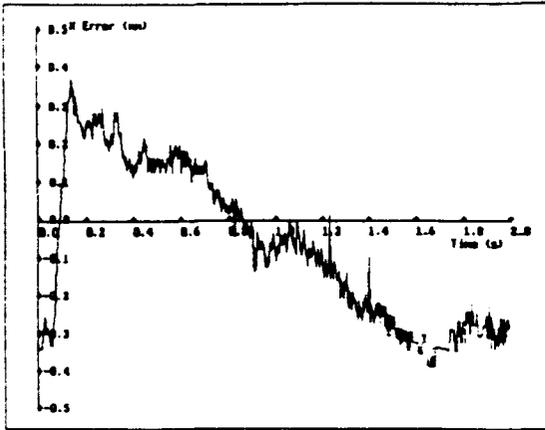
100mm/s

X Error

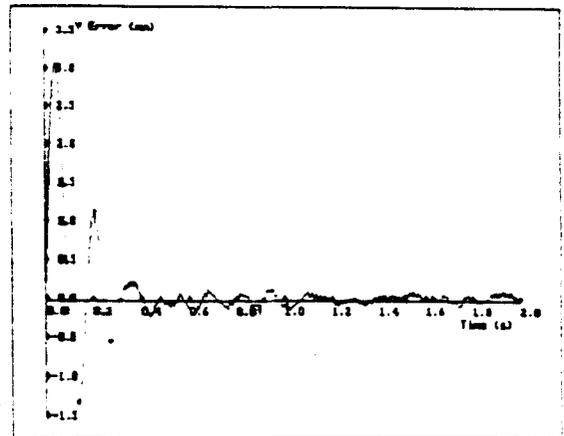
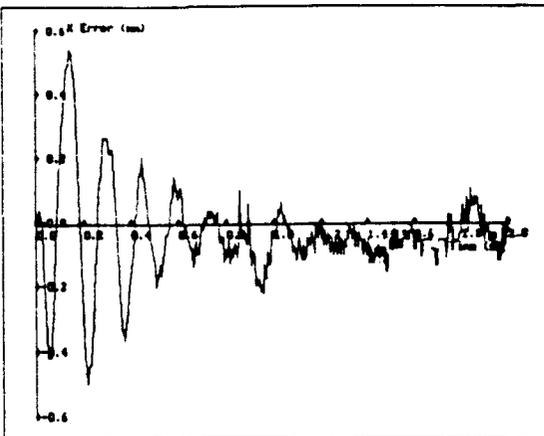
Y Error



Manufacturer's PID Control



Motor Position Based VSC

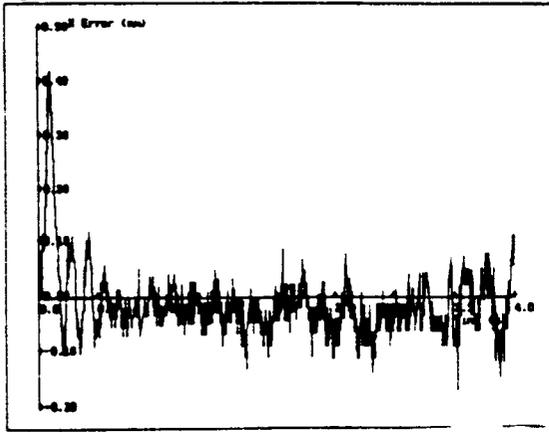


Motor/Joint Based VSC

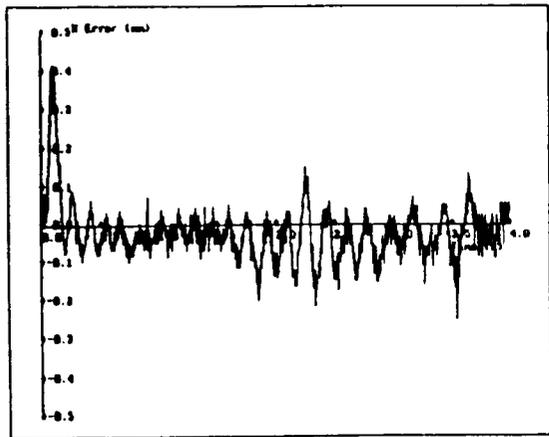
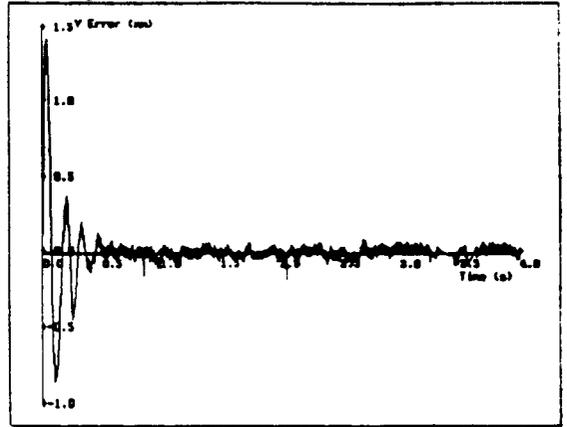
A7.5 Effect of Joint Encoder Resolution of End Point Tracking Accuracy

It was seen in section 6.8, that lower resolution in joint mounted position sensors could be used while still maintaining a reasonable level of control accuracy. The results shown in figure A7.9 show the effect of changes in encoder resolution on the tracking accuracy for a typical trajectory.

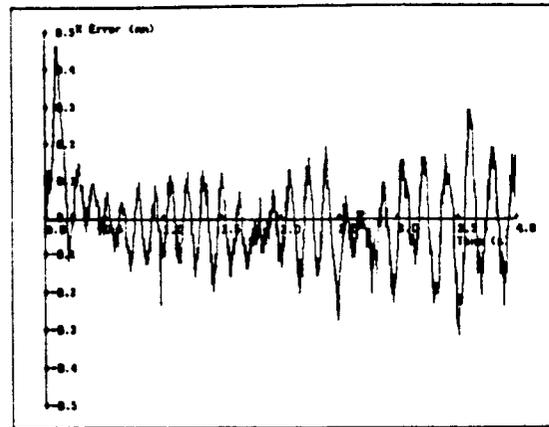
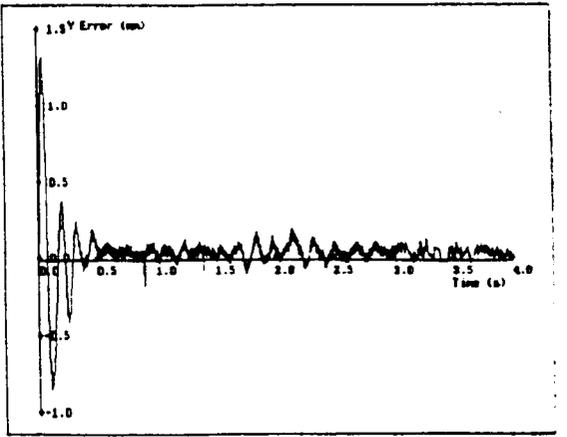
Figure A7.9 Effect of encoder resolution on tracking accuracy



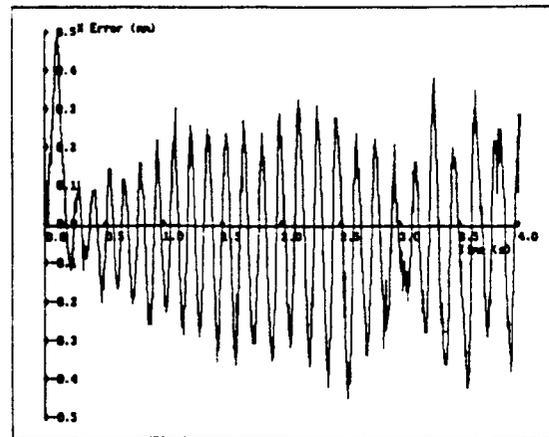
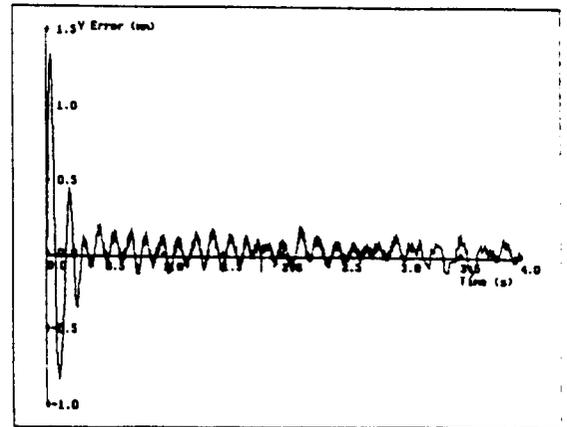
Unity Resolution (10800,3600)



Half Resolution (5400,1800)



Quarter Resolution (2700,900)



Eighth Resolution (1350,450)

